SAPIENZA
UNIVERSITÀ DI ROMA

CONFRONTO TRA MODELLI PREVISIVI PER IL MERCATO IMMOBILIARE: APPLICAZIONE AL CASO DEI PREZZI DEGLI IMMOBILI DI MADRID

Facoltà di Scienze Statistiche
Laurea Magistrale in Scienze Statistiche

Romeo Silvestri
Matricola 1944738

Relatore
Prof.ssa Cecilia Vitiello

Correlatore
Prof. Marco Alfò

Anno Accademico 2021/2022

Tesi non ancora discussa

CONFRONTO TRA MODELLI PREVISIVI PER IL MERCATO IMMOBILIARE: APPLICAZIONE AL CASO DEI PREZZI DEGLI IMMOBILI DI MADRID

Tesi di Laurea Magistrale. Sapienza Università di Roma

Questa tesi è stata composta con LaTeX e la classe Sapthesis.

Email dell'autore: silveromeo98@gmail.com

Indice

Capitolo 1

Introduzione

Il mercato immobiliare è una parte vitale dell'economia di un paese che si occupa della costruzione, della gestione e della compravendita di beni immobili che possono essere adibiti ad una moltitudine di scopi.
In questo contesto, la capacità di valutare e prevedere correttamente il prezzo di vendita degli immobili è diventata una priorità per i professionisti del settore e per gli investitori. Il prezzo è infatti spesso difficile da stabilire perché è influenzato da una serie di fattori sia interni che esterni all'immobile stesso, come la sua dimensione, lo stato di manutenzione, la posizione geografica e la situazione economica. Per questo motivo i modelli statistici sono un utile strumento a supporto delle finalità previsive.
Nell'ambito di questa tesi si comparano quindi i diversi modelli previsivi per il mercato immobiliare per stabilire il valore delle abitazioni nel modo più accurato possibile. In particolare, si presterà attenzione alla componente spaziale, poiché la posizione geografica dell'immobile risulta essere un elemento chiave.
Si utilizzeranno sia modelli previsivi tradizionali che spaziali. I modelli regressivi tradizionali sono stati ampiamente impiegati in quest'ambito e tengono conto della collocazione geografica attraverso l'introduzione di apposite variabili. D'altra parte, i modelli spaziali incorporano le informazioni sulla vicinanza geografica delle osservazioni direttamente nel metodo di costruzione ed esprimono l'intensità di queste relazioni esplicitamente attraverso appositi parametri. E' dunque interessante andare a studiare e confrontare questi due diversi approcci.
Un altro problema importante è rappresentato dalla gestione dei dati mancanti. In questa tipologia di problemi, i dati mancanti sono spesso presenti in grandi quantità, soprattutto per quanto riguarda le informazioni sulle caratteristiche interne all'immobile. Il corretto trattamento dei dati mancanti è quindi parte integrante dell'analisi. Ciò comporta la necessità di utilizzare delle tecniche di imputazione adeguate a stimare i valori mancanti.
La tesi si concentrerà sullo sviluppo di un'applicazione per il mercato immobiliare di Madrid. Questa città rappresenta un'area geografica estremamente interessante con un settore vario, ampio e in costante evoluzione. Oltretutto, quest'area di studio è caratterizzata dalla disponibilità di dati di alta qualità provenienti dai principali portali immobiliari spagnoli. Tali immobili sono infatti descritti da numerose informazioni e, differentemente dai portali di altri paesi, viene spesso indicato l'indirizzo preciso dell'abitazione.

Capitolo 2

Dati Mancanti

I metodi della statistica tradizionale sono stati sviluppati per analizzare insiemi di dati rappresentabili e trattabili sotto forma matriciale come prodotto di unità e variabili. I dati mancanti (o missing data) sono in questo senso un problema che complica l'analisi statistica dei dati raccolti in molte discipline.
In un contesto di regressione, gestire adeguatamente i dati mancanti migliora l'efficacia dei modelli che verranno successivamente applicati ed è quindi fondamentale per ottenere risultati predittivi accurati. Avere a disposizione i dati completi permette difatti di descrivere meglio la relazione tra le variabili esplicative e la variable risposta.
I dati mancanti sono inoltre diffusamente presenti quando viene analizzato il mercato immobiliare e, in particolare, nel caso in cui si voglia trattare il problema della determinazione dei prezzi degli immobili. Spesso, infatti, i dati sul mercato immobiliare sono gestiti da privati o da agenzie immobiliari che non sono a conoscenza di parte delle informazioni sulle proprietà. O, ancora, possono trascurare delle informazioni che sono in realtà determinanti al fine di associare un valore di vendita all'immobile.
L'analisi dei dati mancanti è quindi parte integrante del problema previsionale che si sta affrontando ed è necessario conoscere i metodi di trattamento dei dati mancanti per poter scegliere adeguatamente quello più adatto alle situazioni specifiche.

2.1 Tipologia di Dati Mancanti

Il processo che regola le probabilità che determinano la mancanza dei dati (missingness) è chiamato meccanismo dei dati mancanti o meccanismo di risposta.
In genere il meccanismo di risposta non è sotto il controllo degli sperimentatori, ma si fanno ipotesi su di esso. La validità dell'analisi dipende dal fatto che queste ipotesi siano valide per i dati in questione. Il modello del processo prende invece il nome di modello dei dati mancanti o modello di risposta.
Un primo sistema di classificazione dei dati mancanti è stato formulato da Donald B. Rubin nel 1976. In particolare, si possono avere dati di tipo "Missing Completely at Random" (MCAR), "Missing at Random" (MAR) o "Missing not at Random" (MNAR) a seconda del modello di risposta sottostante.
Per descrivere le varie tipologie di dati mancanti, si utilizzerà un insieme di assunzioni comuni e una conseguente specifica notazione. In questa parte della trattazione non si distinguerà tra variabili esplicative e variabile risposta. Infatti, la gestione della

missingness non influenza l'analisi di regressione.
Si indica con $X = \{x_{ij},\ i = 1, \ldots, n,\ j = 1, \ldots, p\}$ la matrice $n \times p$ contenente i valori dei dati sulle p variabili per le n unità statistiche del campione o della popolazione di riferimento. Viene poi introdotta la matrice indicatrice di risposta $R = \{r_{ij},\ i = 1, \ldots, n,\ j = 1, \ldots, p\}$ di dimensione $n \times p$ che denota la presenza o l'assenza di un determinato dato. Se il valore x_{ij} viene osservato, r_{ij} assume un valore pari a 1, mentre se x_{ij} risulta mancante, r_{ij} è pari a 0.
Si assume che la matrice indicatrice di risposta R sia completamente nota, ovvero di essere a conoscenza della posizione precisa di ogni singolo dato mancante. E' importante specificare che quando gli indicatori di risposta sono nulli, questi mascherano i valori reali, ovvero si suppone per vera l'esistenza di tutti i dati.
In questo scenario $X = \{X_{oss}, X_{mis}\}$, dove X_{oss} è l'insieme dei dati osservati, mentre X_{mis} è l'insieme dei dati mancanti. La matrice X rappresenta quindi l'insieme dei dati ipoteticamente completi.
La distribuzione di R può dipendere da X tramite il disegno campionario o per aleatorietà. Questa relazione è descritta dal modello di risposta $P(R|X_{oss}, X_{miss}, \psi)$, dove ψ indica l'insieme dei parametri che governano tale modello.

2.1.1 Missing Completely at Random

Se la probabilità di essere mancante è la stessa per tutte le osservazioni, si dice che i dati sono completamente mancanti aleatoriamente (MCAR). Ciò implica che il meccanismo dei dati mancanti sia indipendente dai dati osservati e da quelli mancanti.
In termini formali i dati sono detti MCAR se:

$$P(R = 0|X_{oss}, X_{miss}, \psi) = P(R = 0|\psi) \tag{2.1}$$

dove la probabilità di essere mancante dipende solo da alcuni parametri ψ, ovvero dalla probabilità comune di essere mancante.

Un esempio si ha quando viene estratto un campione casuale da una popolazione, in cui ogni individuo ha la stessa probabilità di essere incluso nel campione. I dati non osservati degli individui della popolazione che non sono stati inclusi nel campione sono di tipo MCAR.
Per questo tipo di dati possiamo ignorare molte delle complessità che sorgono a causa della presenza dei dati mancanti, ad esclusione della perdita di informazione. Infatti, la caratteristica fondamentale degli MCAR è che i dati osservati possono essere considerati come un campione casuale dei dati completi. Di conseguenza, i momenti e la distribuzione congiunta dei dati osservati non differiscono dai corrispondenti momenti e dalla rispettiva distribuzione congiunta dei dati completi.
Si hanno conseguentemente importanti implicazioni. In primo luogo, i "casi completi" (unità statistiche prive di dati mancanti, si differenziano dai dati completi) possono essere considerati come un campione casuale della popolazione di riferimento; inoltre, qualsiasi metodo di analisi che produca misure inferenziali valide in assenza di dati mancanti, produrrà misure inferenziali valide, anche se non efficienti, quando l'analisi è limitata ai soli casi completi. Questo modo di procedere viene spesso definito analisi del caso completo.

In secondo luogo, le stesse conclusioni valgono per tutte le unità indipendentemente dal pattern dei dati mancanti. La distribuzione condizionata di X_{oss} per queste unità coincide difatti con la distribuzione di X per la popolazione di riferimento.
Infine, la distribuzione di X_{mis} per le unità con un qualsiasi pattern di dati mancanti combacia con la distribuzione di X per i casi completi.
Di conseguenza, così come nell'analisi del caso completo, anche tutti i dati osservati disponibili possono essere utilizzati per ottenere stime inferenziali valide. In generale, tutti i metodi di analisi che producono misure inferenziali valide in assenza di dati mancanti, produrranno misure inferenziali valide anche quando l'analisi si basa sui soli dati osservati o anche quando è limitata ai casi completi.
Sebbene sia facilmente trattabile, il modello di risposta sottostante spesso non è realistico perché si basa su ipotesi troppo restrittive.

2.1.2 Missing at Random

Se la probabilità di essere mancanti è la stessa solo all'interno dei gruppi denotati dai dati osservati, allora i dati sono mancanti aleatoriamente (MAR). In altre parole, se i soggetti sono stratificati sulla base di valori simili per i dati osservati, la mancanza è semplicemente il risultato di un meccanismo casuale che non dipende dai valori dei dati non osservati.
Formalmente, i dati sono MAR se:

$$P(R = 0|X_{oss}, X_{mis}, \psi) = P(R = 0|X_{oss}, \psi) \tag{2.2}$$

dove la probabilità di essere mancante dipende dai dati osservati e dai parametri del modello, ma non dagli specifici valori mancanti che si sarebbero dovuti osservare.

Un esempio di tipo MAR si verifica quando si estrae un campione casuale da una popolazione, dove la probabilità di inclusione dipende da qualche proprietà nota.
L'ipotesi MAR può essere considerata anche come una generalizzazione dell'ipotesi MCAR e spesso risulta essere più plausibile di quest'ultima in molte applicazioni.
E' essenziale analizzare alcune proprietà degli MAR. Innanzitutto, poiché il meccanismo dei dati mancanti dipende da X_{oss}, la distribuzione di X in ciascuno degli strati definiti dai modelli di risposta differisce dalla distribuzione di X nella popolazione di riferimento. Questo implica che i casi completi formeranno un campione distorto della popolazione di riferimento e la distribuzione condizionata di X_{oss} per le unità con un qualsiasi modello di risposta non coincide con la distribuzione di X per la popolazione obiettivo.
Pertanto, le misure inferenziali basate sui casi completi o sui dati osservati sono necessariamente distorte. Tuttavia, la distribuzione condizionata dei dati mancanti X_{mis}, dati i valori osservati X_{oss}, è uguale alla distribuzione condizionata delle osservazioni corrispondenti ai casi completi per lo stesso condizionamento, a patto che i casi completi abbiano gli stessi valori di X_{oss}.
Pertanto, quando i dati sono MAR, i valori mancanti possono essere validamente imputati utilizzando i dati osservati e un modello corretto per la distribuzione di X.
Poiché il tipo MCAR è uno caso speciale di MAR, valgono le stesse affermazioni appena esposte sull'imputazione.

Per questi motivi, nella pratica, i moderni metodi di imputazione generalmente partono dall'ipotesi che i dati mancanti siano di tipo MAR.

2.1.3 Missing not at Random

Se la probabilità di essere mancante dipende da effetti sconosciuti a colui che conduce l'analisi, allora si è in presenza di dati mancanti non aleatoriamente (MNAR).
In altri termini, il modello di risposta non è semplificabile ed è esprimibile come:

$$P(R = 0|X_{oss}, X_{mis}, \psi) \tag{2.3}$$

quindi la probabilità di essere mancante dipende dai dati osservati, dai dati mancanti stessi e dai parametri del modello.

Questa situazione si può verificare ad esempio quando nel campionamento di una popolazione, la probabilità di risposta dipende da caratteristiche non note o che sfuggono dal processo di misurazione. La gestione dei dati mancanti di tipo MNAR può risultare molto complessa. Solitamente per analizzare questi dati è necessario raccogliere più informazioni sulle cause delle mancate risposte o eseguire delle analisi di sensitività specifiche.
Questo è dovuto al fatto che quando i dati sono MNAR, quasi tutti i metodi di analisi standard non sono validi. Ad esempio, i metodi basati sulla verosimiglianza che ignorano il meccanismo dei dati mancanti producono stime distorte.
Per ottenere stimatori validi, si possono utilizzare dei modelli congiunti per i dati osservati e per il meccanismo dei dati mancanti. I tre approcci principali basati sul modello sono la selezione, la mistura di pattern e i modelli a parametri condivisi. In alternativa, nel tempo sono stati introdotti dei metodi che non richiedono la specificazione della distribuzione congiunta dei dati osservati.

2.1.4 Verifica del tipo di Missingness

In molti contesti può essere utile dover verificare il tipo di dato mancante. Il processo di verifica non è sempre possibile o intuitivo.
Per verificare l'ipotesi di MCAR contro quella di MAR sono stati proposti numerosi test statistici, benché non siano molto diffusi e utilizzati nella pratica. Tutti questi test hanno come presupposto che i dati non siano di tipo MNAR.
Il metodo più semplice per valutare l'MCAR consiste nell'utilizzare una serie di test t a due a due indipendenti per confrontare i sottogruppi dei dati mancanti. Questo approccio separa i casi mancanti da quelli completi su una particolare variabile ed esamina le differenze medie del gruppo su altre variabili del set di dati. Il meccanismo MCAR implica che i casi completi debbano essere in media uguali a quelli con valori mancanti. Di conseguenza, un test t non significativo dimostra che i dati sono MCAR, in caso contrario, suggerisce che i dati sono MAR o MNAR.
Un altro possibile approccio prende il nome di test di Little. Questo test è un'estensione multivariata dell'approccio del test t che valuta simultaneamente le differenze medie su ogni variabile dell'insieme dei dati. A differenza dei test t univariati, la procedura di Little è un test globale a confronti multipli per la verifica di MCAR che

si applica all'intero set di dati. Nonostante questi test siano efficaci per verificare l'assunzione di MCAR, non forniscono elementi per identificare le potenziali variabili dipendenti dal meccanismo dei dati mancanti.
Una verifica oggettiva che non può essere fatta è quella tra MAR e MNAR. Questo è dovuto al fatto che le informazioni necessarie per un tale test risultano mancanti.

2.2 Ignorabilità

I modelli per i dati mancanti dipendono esplicitamente da alcuni parametri ψ che sono in genere sconosciuti e non sono di interesse per l'analisi.
In ambito parametrico, il focus dell'analisi è invece incentrato su dei parametri di interesse θ legati alle caratteristiche dei dati. Nella regressione parametrica θ corrisponde all'insieme dei coefficienti di regressione e/o alle statistiche di sintesi della variabile risposta. L'importanza pratica della distinzione tra i diversi tipi di dato mancante consiste nel fatto che essa definisce le condizioni in cui è possibile stimare accuratamente i parametri θ senza la necessità di conoscere ψ.
I dati effettivamente osservati sono X_{oss} e R, mentre la funzione di densità congiunta $f(X_{oss}, R|\theta, \psi)$ di X_{oss} e R dipende sia dai parametri d'interesse θ che dai parametri ψ. La funzione di verosimiglianza di θ e ψ è proporzionale alla loro densità congiunta:

$$L(\theta, \psi|X_{oss}, R) \propto f(X, R|\theta, \psi) \tag{2.4}$$

dove $L(\cdot)$ è la funzione di verosimiglianza.

Il meccanismo dei dati mancanti è ignorabile per l'inferenza basata sulla verosimiglianza se la funzione di verosimiglianza non dipende da X_{mis}. Esploreremo meglio le implicazioni sulla teoria della verosimiglianza nella sezione 2.3.2.

E' importante sapere che in ogni caso questo proprietà è valida se vengono rispettate le seguenti condizioni:

- i dati mancanti sono del tipo MAR (o MCAR);
- i parametri θ e ψ sono distinti, ovvero lo spazio congiunto dei parametri (ψ, θ) è il prodotto dello spazio dei parametri di θ e dello spazio dei parametri di ψ.

La prima condizione è in genere più rilevante, in quanto la condizione sui parametri è difficile da verificare, ma risulta in quasi tutte le casistiche intuitiva e ragionevole.

Nella costruzione dei modelli di imputazione è di particolare interesse definire la distribuzione a posteriori dei dati mancanti condizionata ai dati osservati e al meccanismo dei dati mancanti. Tale distribuzione viene indicata con $P(X_{mis}|X_{oss}, R)$. Nel caso in cui il meccanismo di risposta sia ignorabile, si può dimostrare che la distribuzione a posteriori non dipende dal meccanismo di risposta stesso, ovvero:

$$P(X_{mis}|X_{oss}, R) = P(X_{mis}|X_{oss}) \tag{2.5}$$

Questo implica che la distribuzione dei dati X sia la stessa tra il gruppo di dati osservati e quello dei dati mancanti.:

$$P(X|X_{oss}, R = 1) = P(X|X_{oss}, R = 0) \tag{2.6}$$

Di conseguenza, se il modello dei dati mancanti è ignorabile, si può formulare la distribuzione a posteriori a partire dai dati osservati e utilizzare il modello in questione per l'imputazione.
Al contrario, se il meccanismo dei dati mancanti non è ignorabile, non è possibile trarre conclusioni sulla distribuzione a posteriori e quindi utilizzare il modello associato ad essa.

2.3 Trattamento dei Dati Mancanti

La presenza dei dati mancanti implica necessariamente un trattamento mirato. Dopo aver assunto o verificato il tipo dei dati mancanti bisogna infatti valutare la possibilità di procedere con l'imputazione delle informazioni non presenti nel dataset. In letteratura esistono numerose metodologie per gestire i dati mancanti. In questo testo presenteremo diversi tipi di approcci e ne studieremo le proprietà statistiche.

2.3.1 Metodi Naïve

I metodi più comunemente utilizzati per trattare la missingness ignorano durante l'analisi i dati mancanti, rimuovono le osservazioni incomplete o consistono in forme semplici di imputazione, ossia metodi tramite i quali i dati assenti sono effettivamente sostituiti per consentire un'analisi completa dei dati.
Questi approcci generalmente non sono molto affidabili e spesso conducono a stime distorte e non valide per effettuare previsioni.

Listwise Deletion

L'approccio che veniva un tempo frequentemente utilizzato per trattare i dati mancanti nella maggior parte delle analisi statistiche si riduce semplicemente alla cancellazione di tutte le osservazioni che li presentano. Questo approccio è chiamato Cancellazione dalla Lista ("*Listwise Deletion*") o equivalentemente, come sopra anticipato, analisi del caso completo, in quanto utilizza solo le osservazioni (o i casi) in cui tutte le variabili sono state osservate.
La Listwise Deletion è un metodo molto semplice per gestire i dati mancanti, poiché non richiede alcun calcolo o manipolazione dei dati e permette di ottenere una soluzione rapida per tale problema. Questa procedura può portare d'altra parte ad una elevata eliminazione delle osservazioni e ad una conseguente perdita di informazione non trascurabile.
Sotto l'ipotesi di un meccanismo di risposta di tipo MCAR, con la Listwise Deletion tutte le proprietà che sono valide per il modello statistico utilizzato rimarranno valide. Specificatamente, a livello parametrico le stime non risulteranno distorte. Inoltre, poiché con la Listwise Deletion la numerosità campionaria diminuisce, gli errori standard prodotti dai modelli di regressione applicati saranno maggiori rispetto agli errori standard che sarebbero stati prodotti nel caso in cui i dati mancanti fossero stati osservati. Ciò nonostante, le stime degli errori standard risultano non distorte. Come visto precedentemente, l'ipotesi di MCAR è però spesso poco realistica.
Se i dati mancanti non sono MCAR, la Listwise Deletion non restituirà stime affidabili dei parametri, poiché la relazione tra le variabili da cui dipende il meccanismo

dei dati mancanti e la variabile su cui sono presenti i dati mancanti implicherà una distorsione nei parametri indipendentemente dal tipo di analisi che viene realizzata. Nell'ambito dell'analisi della regressione, la Listwise Deletion possiede però alcune caratteristiche uniche che la rendono interessante in alcuni contesti. In alcuni casi può fornire stime analoghe alle procedure più sofisticate. Se i valori mancanti sono infatti associati soltanto alla variabile risposta (e non alle variabili esplicative), la Listwise Deletion è equivalente a livello di proprietà ai metodi di imputazione multipla per la determinazione dei coefficienti di regressione. Le quantità che dipendono dalla corretta distribuzione marginale della variabile dipendente, come la media o il coefficiente di determinazione R^2, richiedono però l'ipotesi di MCAR.
Esistono anche casi in cui il metodo in esame può superare gli altri metodi per la gestione dei dati mancanti. Il primo caso speciale si verifica quando la probabilità di missingness non dipende dalla variabile risposta, mentre il secondo caso si presenta con l'applicazione della regressione logistica quando la probabilità di mancare dipende dalla sola risposta o da un'unica esplicativa.
In sintesi, la Listwise Deletion non possiede buone proprietà per nessuno dei tre tipi di dato mancante, ad esclusione di casi studio molto specifici. Per questo motivo risulta un metodo sconsigliabile per la maggior parte delle analisi.

Pairwise Deletion

Un'alternativa al caso precedente è quella di utilizzare un metodo che elimina solo parzialmente le osservazioni. Questo approccio per la gestione dei dati mancanti è chiamato Cancellazione a Coppie ("*Pairwise Deletion*") o analisi del caso disponibile ed è un metodo per gestire i dati mancanti che prevede la conseguente applicazione dei modelli lineari e non è adatta ad altre situazioni.
Il metodo stima le medie e la matrice di varianze e covarianze utilizzando tutti i dati disponibili. Pertanto, le medie e le varianze per le singole variabili sono stimate impiegando i dati osservati per quella variabile, mentre le covarianze tra le variabili sono stimate a due a due utilizzando tutte le osservazioni con dati disponibili per entrambe le variabili.
Successivamente, la matrice delle statistiche descrittive di sintesi viene utilizzata con una metodologia specifica per l'analisi lineare che si vuole compiere, come il modello regressivo lineare o il modello per l'analisi della varianza.
Come nel caso della Listwise Deletion, i parametri ottenuti dalla Pairwise Deletion sono non distorti sotto l'ipotesi di MCAR. Quando i dati sono MAR, la Pairwise Deletion produce invece stime soggette a distorsione.
La stima degli errori standard in un contesto di regressione con la Pairwise Deletion è tuttavia molto complessa, poiché ogni matrice di covarianza può essere teoricamente stimata con un numero differente di osservazioni. Poiché gli errori standard sono una funzione della dimensione del campione, è impossibile stimare correttamente questi errori. Un'ulteriore complicazione è dovuta al fatto che le matrici di covarianza e di correlazione ottenute con questo metodo non sono necessariamente definite positive.
E' importante osservare che nel caso in cui la correlazione tra le variabili è bassa, la Pairwise Deletion offre delle stime più efficienti della Listiwise Deletion. Al contrario, se la correlazione è elevata, risulta meno efficiente.

In sintesi, la Pairwise Deletion rappresenta un piccolo miglioramento rispetto alla Listiwise Deletion a livello teorico, in quanto fa uso di tutti i dati disponibili, tuttavia rimane insoddisfaciente.
Inoltre, sebbene non sia impegnativa dal punto di vista computazionale, la Pairwise Deletion è un metodo che molto spesso risulta complicato da utilizzare in molti contesti applicativi.

Imputazione con Media, Moda e Mediana

Il metodo di imputazione più semplice consiste nel sostituire i valori mancanti con la media, la moda o la mediana della specifica variabile per cui il dato è mancante. Per questo motivo, tale metodo prende anche il nome di imputazione della media, moda o mediana non condizionata.
Si evidenzia che la media può essere utilizzata soltanto per le variabili quantitative, la mediana per le variabili quantitative o qualitative ordinali, infine la moda per tutti i tipi di variabili.
Questo metodo modifica la distribuzione delle variabili in molteplici modi e ha conseguenze importanti nelle analisi successive.
L'imputazione della media, ad esempio, riduce la variabilità dei dati, in quanto lo stesso valore di centralità viene assegnato per tutti i valori mancanti. Questo porta ad una sottostima della varianza per la suddetta variabile e altera la relazione di covarianza con le altre variabili presenti. L'imputazione della media porterà quindi a stime distorte per quasi tutti i parametri diversi dalla media, oltre al fatto che produrrà errori standard più bassi a causa della diminuzione della variabilità. Se i dati non sono MCAR, anche la stima della media può risultare distorta.
Simili conclusioni si possono trarre con l'imputazione di moda o mediana.
Ad esempio, l'imputazione di numerosi valori con la mediana (e anche con la media) può trasformare una distribuzione unimodale in bimodale.
L'imputazione con media, moda e mediana costituisce una soluzione rapida al problema dei dati mancanti e non richiede un costo computazionale intensivo. In ogni caso, non avendo buone proprietà, questo metodo di imputazione viene sconsigliato per tutti i tipi di analisi.

2.3.2 Metodi di Massima Verosimiglianza e Bayesiani

I metodi basati sulla funzione di verosimiglianza o sulla statistica bayesiana non sono veri e propri metodi di imputazione, ma si basano invece sui dati disponibili.
Più precisamente, per questi metodi l'imputazione viene definita implicita, perché, nonostante i valori di imputazione non siano esplicitati, si suppone la loro esistenza.
In generale, i metodi per la gestione dei dati mancanti basati sulla verosimiglianza presuppongono l'esistenza di un modello parametrico per i dati completi e, nei casi in cui si ipotizza che la missingness non sia ignorabile, richiedono anche un modello parametrico per il meccanismo dei dati mancanti.
Con i dati mancanti, l'inferenza di verosimiglianza si basa sulla funzione di verosimiglianza dei dati disponibili:

$$\begin{aligned} L(\theta, \psi | X_{oss}, R) &= c \times \int f(X, R|\theta, \psi) dX_{mis} \\ &= c \times \int f(X_{oss}, X_{mis}, R|\theta, \psi) f(R|X_{oss}, X_{mis}, \psi) dX_{mis} \end{aligned} \tag{2.7}$$

dove c è un fattore che non dipende da (θ, ψ).

Sotto l'ipotesi di ignorabilità del meccanismo dei dati mancanti si ottiene:

$$\begin{aligned} L(\theta, \psi | X_{oss}, R) &= c \times \int f(X_{oss}, X_{mis}|\theta) f(R|X_{oss}, \psi) dX_{mis} \\ &= c \times f(X_{oss}|\theta) f(R|X_{oss}, \psi) \end{aligned} \tag{2.8}$$

La stima di massima verosimiglianza (MLE) di θ (e ψ) è il valore che massimizza $L(\theta, \psi|X_{oss}, R)$. In campioni di grandi dimensioni, la MLE ha una distribuzione approssimativamente normale con matrice di covarianza data dall'inverso della matrice di informazione osservata. Gli errori standard che ne conseguono tengono adeguatamente in considerazione del fatto che alcuni dati siano mancanti.
Nell'approccio bayesiano, infatti, ai parametri (θ, ψ) viene assegnata una distribuzione a priori, mentre l'inferenza per θ (e ψ) si basa sulla sua distribuzione a posteriori, ottenuta moltiplicando la verosimiglianza dei dati osservati per la distribuzione a priori di (θ, ψ).
Le stime puntuali di (θ, ψ) possono essere ottenute come misure di sintesi della distribuzione a posteriori. L'incertezza sulle stime puntuali può essere espressa in termini di deviazioni standard a posteriori o di intervalli di credibilità basati sui quantili della distribuzione a posteriori.
Il calcolo di questi intervalli solitamente prevede l'estrazione dalla distribuzione a posteriori tramite simulazione di Markov Chain Monte Carlo (MCMC). Per campioni con numerosità campionaria ridotta l'inferenza derivante è particolarmente sensibile alla scelta della distribuzone a priori.
Un'importante distinzione si deve fare quando si ipotizza l'ignorabilità o la non ignorabilità del meccanismo dei dati mancanti. Quando si ipotizza che la mancanza sia ignorabile, i metodi basati sulla verosimiglianza (e in modo simile per i metodi bayesiani) imputano implicitamente i valori mancanti trattando e stimando i parametri per $f(X|\psi)$. In questo caso la verosimiglianza si basa esclusivamente sulla distribuzione marginale dei dati osservati e le stime si ottengono massimizzando la funzione di verosimiglianza. Il contributo di verosimiglianza per ciascuna osservazione è $f(X_{oss}|\psi)$. In un certo senso, i dati mancanti sono validamente previsti dai dati osservati attraverso il modello per il valore atteso condizionato $E(X_{mis}|X_{oss}, \psi)$ e il modello per la covarianza.
In questa forma di imputazione i parametri sono stimati massimizzando la verosimiglianza con algoritmi appositi come l'algoritmo EM o l'algoritmo di Newton-Raphson. Particolarmente efficiente è l'algoritmo EM (Expectation-Maximization) che risulta spesso essere l'implementazione standard.
L'EM è infatti un algoritmo iterativo che si sviluppa in due fasi. Nella prima fase (fase di Expectation) si attua l'imputazione dei valori mancanti con i rispettivi valori attesi condizionati, date le risposte osservate e le stime dei parametri dell'iterazione

precedente. Successivamente, in seconda fase (fase di Maximization) si massimizzano le probabilità per i "dati completi" che ne sono risultati.
Pertanto, quando la mancanza è ignorabile, l'inferenza basata sulla verosimiglianza non richiede la specificazione del meccanismo dei dati mancanti, ma richiede ipotesi distributive complete su X. Inoltre, l'intero modello per $f(X_{oss}|\psi)$ deve essere specificato correttamente. Qualsiasi errata specificazione del modello per la covarianza produrrà, in generale, stime distorte della media della variabile dipendente.
Se il meccanismo dei dati mancanti non è ignorabile, è necessario impiegare un modello congiunto dei dati (modelli di selezione o di mistura di pattern) e le misure inferenziali tendono a divenire estremamente sensibili alle ipotesi del modello che non sono verificabili.

2.3.3 Metodi di Ponderazione

La ponderazione è un insieme di metodi per ridurre la distorsione quando la probabilità di inclusione delle unità nel campione varia.
Con questi metodi, la sottorappresentazione (o sovraraprresentazione) di alcuni gruppi di unità tra i dati osservati viene presa in considerazione e corretta. Questo avviene spesso nelle indagini campionarie dove i rispondenti non hanno la stessa probabilità di essere inclusi.
Nel tempo sono stati proposti numerosi approcci che prendono generalmente il nome di metodi a propensione ponderata o a probabilità inversa ponderata (IPW).
In quest'ottica le osservazioni sono ponderate in base ai pesi di progetto, i quali sono inversamente proporzionali alla probabilità di essere selezionati nell'indagine.
In caso di dati mancanti, i casi completi vengono ripesati in base ai pesi di progetto e aggiustati per contrastare gli effetti di selezione prodotti dalle mancate risposte.
Questo metodo è ampiamente utilizzato nella statistica sociale.
L'implementazione è relativamente semplice in quanto è necessario un solo set di pesi per tutte le variabili incomplete. Tuttavia, scarta i dati mediante Listwise Deletion e non può gestire le osservazioni parziali.
Le forme derivanti per esprimere la varianza dei pesi di regressione e delle correlazioni tendono a essere complesse o non esistenti. A differenza dei metodi pesati per i disegni campionari, i pesi sono stimati dai dati osservati e vengono solitamente fissati.
Le implicazioni di questa procedura non risultano tuttora ben definite.
A partire dal ventunesimo secolo, si è registrato un maggiore interesse per le procedure di ponderazione robuste. Uno di questi metodi viene detto metodo doppiamente robusto e richiede la specificazione di tre modelli: il primo modello è quello di interesse per i dati completi, il secondo modello è quello di risposta e il terzo ed ultimo modello è un modello congiunto per i predittori e per l'esito.
La doppia proprietà di robustezza afferma che se uno tra il secondo e il terzo modello è errato (ma non entrambi), le stime del primo modello sono ancora coerenti e utilizzabili al fine dell'analisi.
Concludendo, i metodi di ponderazione sono più adatti per trattare i pattern di dati mancanti monotoni, ovvero quando la mancanza di una certa variabile X_j per un'unità implica che tutte le variabili X_k che seguono ($k > j$) o che la precedono ($k < j$) siano mancanti per tutte le altre unità. Sebbene infatti i metodi di pon-

derazione possano essere applicati a modelli di dati mancanti non monotoni, nella pratica questi sono più complessi da formulare e difficili da implementare.

2.3.4 Metodi di Regressione Parametrica

Un'ampia gamma di metodi di imputazione utilizzati è quella dei modelli di regressione parametrica. Questi modelli sono molto diffusi per le analisi di regressione classiche, ma trovano qui spazio anche per il trattamento dei dati mancanti. Nel seguito andiamo quindi a mostrare alcuni di questi modelli.

Modello di Regressione Lineare

Nell'ambito del trattamento dei dati mancanti, l'utilizzo dei modelli di regressione lineare prende spesso il nome di imputazione della media condizionata. Questo serve per mettere in relazione questo metodo con l'imputazione della media non condizionata. Un modo per migliorare l'imputazione della media consiste infatti nell'imputare la media della variabile condizionata ai valori osservati sulle altre variabili tramite il modello di regressione in questione.
A differenza dei metodi Naïve per i dati mancanti, l'imputazione per regressione lineare produce stime corrette dei parametri sia per gli MCAR che per gli MAR, a condizione che le variabili che influenzano la mancanza dei dati negli MAR siano incluse nei modelli statistici utilizzati.
Inoltre, non solo consente di utilizzare tutti i dati disponibili, ma sfrutta inoltre questi utlimi per migliorare le imputazioni dei dati mancanti.
L'aspetto negativo dell'imputazione per regressione lineare è il fatto che le medie condizionate per i valori mancanti rafforzeranno le relazioni tra le variabili, poiché tutti i valori si trovano sulla retta di regressione. Questo porterà a una sottostima della variabilità dei dati e degli errori standard.
L'imputazione per regressione lineare rimane di fatto un metodo relativamente facile da gestire ed implementare, ma non costituisce la scelta primaria per l'imputazione dei dati mancanti. E' da sottolinare infine che questo modello può funzionare soltanto in presenza di variabili quantitative o codificate come tali.

Modelli di Regressione Lineare Generalizzati

In presenza di variabili qualitative si può ricorrere ad una classe di modelli più ampia rispetto ai modelli di regressione lineare. I modelli lineari generalizzati estendono infatti il concetto di regressione lineare introducendo una famiglia distributiva comune. Si rimanda all'apposita sezione del capitolo 3 per un trattamento formale completo. In ogni caso, a seconda della natura della variabile viene associato un determinato modello. Per le variabili qualitative dicotomiche viene generalmente utilizzato il modello di regressione logistica, per le variabili qualitative multicategoriche si ricorre al modello di regressione logistica multinomiale, e, infine, per le variabili qualitative multicategoriche ordinate si utilizza il modello di regressione logistica multinomiale ordinato (o modello di probabilità proporzionale).

Modello di Regressione Stocastica

Il problema principale dell'imputazione con il modello di regressione lineare è il fatto che i valori che vengono sostituiti risultano essere di tipo deterministico e producono così necessariamente sempre lo stesso insieme di valori. La stima di una retta di regressione, tuttavia, è sempre associata ad alcuni errori residui intorno alla retta di regressione.
Un modo naturale per gestire il problema delle imputazioni deterministiche è quindi quello di aggiungere un termine di errore casuale a ogni valore imputato a partire dalla distribuzione normale, dove la deviazione standard per la distribuzione normale è presa dall'errore standard residuo del modello.
Questo metodo di generazione delle imputazioni casuali dalla media condizionata migliora leggermente la situazione, aumentando la varianza delle variabili e producendo errori standard più elevati in un contesto di regressione. Tuttavia, le varianze e gli errori standard saranno ancora troppo ridotti rispetto alla controparte reale.
Il motivo per cui gli errori standard e le varianze rimangono troppo piccoli è dato dal fatto che, in un contesto di regressione, l'incertezza delle stime non dipende solo dai residui, ma anche dall'incertezza dei parametri stimati.
Poiché i parametri di regressione, in base al Teorema del Limite Centrale, seguono una distribuzione approssimativamente normale se la dimensione del campione è elevata, o una distribuzione normale esatta se i residui sono distribuiti normalmente, è possibile generare le imputazioni estraendo casualmente i parametri tenendo conto di un termine di errore. In questo modo si ricrea l'incertezza associata alla regressione e si minimizza il problema della sottostima delle varianze e degli errori standard.
Un aspetto negativo di questo approccio è che gli stimatori dei parametri non mostrano delle buone proprietà di efficienza. Questo problema può essere risolto utilizzando tale metodo in un contesto di imputazione multipla che verrà descritto successivamente.
L'imputazione con il modello di regressione stocastica è più complicata di quella con il modello regressivo lineare, ma rimane comunque molto soddisfacente dal punto di vista computazionale.
In sintesi, la regressione stocastica risulta essere un metodo valido per il trattamento dei dati mancanti grazie alle sue proprietà statistiche di qualità.

Altri Modelli di Regressione Parametrica

Oltre ai metodi parametrici appena esposti esistono numerosi altri modelli di regressione parametrica che vengono utilizzati solitamente per tipi di variabile o casi applicativi specifici. Presentiamo nel seguito una panoramica non esaustiva dei modelli che sono impiegati in base a ciascuna particolare situazione.

Per le variabili di conteggio si utilizzano i seguenti modelli:

- Modello Logistico Multinomiale Ordinato
- Modello di Poisson
- Modello Binomiale Negativo

Questi modelli tengono in considerazione il fatto che le variabili di conteggio assumono valori discreti non negativi. L'approccio che si adotta può essere quello di trattare la variabile di conteggio come una variabile qualitativa ordinale e utilizzare quindi un modello lineare generalizzato (il primo modello nella lista).
Alternativamente il secondo e il terzo modello elencato possono essere implementati nella loro forma base o nella versione che tiene conto dell'inflazione degli zeri. In particolare, il modello basato su una distribuzione binomiale negativa risulta particolarmente flessibile e adatto a gestire i casi di sovradispersione dei dati non uguagliando il valore della varianza alla media del modello come nel caso della distribuzione di Poisson.

Per i dati semi-continui, dove si osserva un'elevata concentrazione di valori in un punto (solitamente lo zero) e una distribuzione continua nei valori restanti, si procede con una procedura a due fasi. In primo luogo si determina se il valore da imputare sia uno zero, e successivamente, nel caso non lo sia, si estrae un valore per la parte continua. Spesso per la parte discreta si impiega un modello logistico, mentre per la parte continua un modello normale dopo un'opportuna trasformazione.

Un'altro particolare tipo di situazione si presenta quando i dati sono censurati o troncati. Per queste casistiche si ricorre spesso ad analisi di sopravvivenza che esaminano attentamente il problema in esame. Per i dati censurati due tra i metodi più diffusi per trattare i dati mancanti sono il Modello del Set di Rischio e il Modello di Kaplan-Meier

Come si è visto la varietà dei casi particolari può essere molto diversificata. Spesso invece di ricorrere a modelli parametrici si preferisce semplificare il processo di imputazione tramite metodi di matching che verranno esposti nella sezione dedicata.

2.3.5 Metodi di Regressione Non-Parametrica

I metodi non-parametrici trovano frequentemente spazio nell'imputazione dei dati mancanti. Molti dataset contengono infatti delle relazioni tra variabili di tipo non lineare e complesse strutture di interazione che non sono facili da cogliere con dei modelli parametrici.
Il grande vantaggio dei metodi non-parametrici è la loro flessibilità nell'adattarsi a relazioni molto diverse tra loro. I modelli applicati sono il più delle volte basati su criteri di vicinanza o sugli alberi decisionali e possono essere impiegati sia in contesti quantitativi che qualitativi.
Il k-nearest neighbours (KNN) fa uso di una determinata metrica di distanza per associare, e quindi imputare, i dati mancanti con il più vicino tra i k gruppi di dati. Solitamente il KNN si applica ai dati standardizzati ed è efficiente a livello computazionale con tutti i tipi di dati.
Tra i metodi basati sugli alberi di regressione o di classificazione (CART) spicca in particolare l'albero decisionale semplice e il modello Random Forest.
E' importante non confondere l'imputazione dei dati mancanti tramite i metodi basati sugli alberi con l'applicazione dei corrispondenti modelli sui dati disponibili. Per quest'ultima situazione, i modelli possono funzionare ignorando i soli valori

mancanti o applicando degli aggiustamenti appositi (come ad esempio l'uso di categorie dedicate o di split surrogati). Questi metodi non vengono però trattati in questa sede. I veri e propri modelli per dati mancanti che sono stati citati verranno meglio trattati nel capitolo relativo ai modelli. Risulta difatti facile riadattare i modelli di regressione non parametrici al trattamento dei dati mancanti.

2.3.6 Metodi Hot Deck

I metodi di imputazione "*Hot Deck*", nelle loro forme più semplici, erano spesso utilizzati nelle analisi statistiche.
La loro caratteristica contraddistintiva consiste nell'imputare i valori mancanti di un'unità tramite l'appoggio ad un'unità simile secondo le caratteristiche osservate.
Il termine "hot deck" faceva riferimento all'archiviazione dei dati su schede perforate e indica che i donatori di informazioni, ovvero le unità da cui sono prese le imputazioni, provengono dallo stesso set di dati dei destinatari (le unità che ricevono le imputazioni). La lista di unità era "hot" in quanto era in corso di elaborazione.
Questi metodi sono contrapposti ai metodi di tipo "*Cold Deck*", dove i donatori di informazioni vengono presi da set di dati esterni al dataset di riferimento.
Una distinzione che viene fatta in alcuni studi è quella tra metodi hot deck deterministici, in cui l'associazione tra donatore e destinatario è univoca e basata su una particolare metrica (in questo senso il KNN può essere visto anche come metodo hot deck), e metodi hot deck aleatori, dove il donatore effettivo è estratto in maniera casuale da un insieme di potenziali donatori.

LOCF e BOCF

Il Last Observation Carried Forward (LOCF) e il Baseline Observation Carried Forward (BOCF) sono forme elementari di metodi hot deck deterministici.
Il loro utilizzo si rivolge principalmente ai dati longitudinali o di tipo caso-controllo.
Sostanzialmente, quando mancano più valori in successione, il metodo cerca l'ultimo valore osservato nel caso della LOCF o il valore considerato di base in condizioni normali per il BOCF.
Per il LOCF le unità vengono quindi dapprima ordinate secondo uno specifica variabile rilevante (che può essere ad esempio il tempo), o set di variabili rilevanti, e successivamente si associano i valori mancanti delle unità con quella immediatamente precedente che presenta i valori necessari all'imputazione.
L'uso del BOCF avviene, ad esempio, in uno studio sul dolore cronico in cui, quando un paziente si ritira dal trattamento, può essere ragionevole supporre che il dolore torni al livello di base e che il paziente, a lungo termine, non ne tragga beneficio.
Analizzando le proprietà statistiche, è stato dimostrato che questi sono metodi conservativi e si può osservare una distorsione delle stime in entrambe le direzioni anche in presenza di MCAR. Per tali motivi il LOCF e il BOCF non vengono più impiegati come approccio primario per la gestione dei dati mancanti, a meno che le ipotesi su cui si basano non siano scientificamente giustificate.
Nel corso del tempo sono stati proposti metodi alternativi volti ad aggiustare le limitazioni del LOCF o del BOCF. Ad esempio, il Last Rank Carried Forward (LRCF) è una versione migliorata e non-parametrica del LOCF basata sui ranghi.

Nonostante ciò, tali metodi risultano molto settoriali e meno efficienti rispetto ai principali metodi di imputazione.

Predictive Mean Matching

Il Predictive Mean Matching calcola il valore previsto della variabile con valore mancante in base ad un modello di imputazione specificato.
Per ogni valore mancante, il metodo forma un piccolo insieme di donatori potenziali a partire da tutti i casi completi che hanno valori più vicini al valore da prevedere per il dato mancante. La vicinanza tra le osservazioni è espressa tramite un preciso criterio di vicinanza. Sono possibili diverse metriche per definire la distanza tra i casi. Il valore previsto deve rappresentare un riassunto di un numero di informazioni rilevanti che mettono in relazione le variabili con quella da imputare. Una volta definita la metrica, un donatore viene estratto aleatoriamente tra i candidati e il valore osservato del donatore servirà per imputare quello mancante. L'ipotesi fondamentale è che la distribuzione del valore mancante per i donatori potenziali sia la stessa rispetto a quella dei dati osservati.
Sia $\hat{x}_i$ il valore riassuntivo previsto dal potenziale donatore i-esimo, con $i = 1, ..., n_0$ e $\hat{x}_j$ il valore riassuntivo previsto dall'osservazione con valori mancanti j-esimo, con $j = 1, ..., n_1$, allora i possibili metodi per estrarre un donatore consistono in:

- Selezionare come donatore effettivo il candidato i-esimo con distanza $\mid \hat{x}_i - \hat{x}_j \mid$ minima. Questo metodo è deterministico e non ottimale.
- Fissare un valore soglia η ed eleggere come donatori potenziali le unità con $\mid \hat{x}_i - \hat{x}_j \mid < \eta$. Dopodiché viene estratto casualmente un donatore effettivo tra i candidati e si prendono i suoi valori come riferimento per l'imputazione.
- Selezionare un numero fissato di donatori potenziali d (in genere 3, 5 o 10) con distanze $\mid \hat{x}_i - \hat{x}_j \mid$ minime ed estrarre casualmente un donatore effettivo.
- Selezionare un donatore effettivo in base a delle probabilità che dipendono in modo proporzionale da $\mid \hat{x}_i - \hat{x}_j \mid$.

Tra i metodi elencati, si preferisce di norma per semplicità impostare il numero di donatori potenziali. Il numero da determinare d può influenzare significativamente i risultati in base alla numerosità del campione dei dati. Generalmente un numero basso di donatori d comporta un maggior numero di duplicati. Invece un numero elevato allevia il problema dei duplicati, ma introduce una distorsione dovuta alla perdita della qualità delle associazioni. E' consigliabile aumentare il valore d proporzionatamente alla numerosità campionaria per raggiungere risultati inferenziali migliori. E' anche possibile utilizzare un metodo che permette di impostare il numero di donatori potenziali in modo adattivo in base ai dati osservati. Si rimanda al lavoro di Schenker e Taylor del 1996 per ulteriori dettagli.
Il predictive mean matching risulta essere particolarmente versatile e robusto rispetto alle trasformazioni delle variabili. Il metodo può essere utilizzato per tutti i tipi di varibili, ma possiede proprietà statistiche migliori quando si vanno a trattare variabili quantitative continue e discrete. Le imputazioni si basano su valori osservati e sono, di conseguenza, più realistiche rispetto a valori fuori range. Per costruzione,

le imputazioni al di fuori dell'intervallo dei dati osservati non si verificano, evitando così imputazioni prive di significato.
Il modello sottostante è implicito, il che significa che non è necessario definire un modello esplicito per la distribuzione dei valori mancanti. Per questo motivo, il Predictive Mean Matching risente meno dell'errata specificazione di un modello rispetto alla maggior parte dei metodi di imputazione.

2.4 Imputazione Multipla

Fino ad ora è stato dato implicitamente per scontato di trovarsi in un contesto di imputazione singola. L'imputazione singola si ha infatti quando l'assegnazione viene applicata una ed una sola volta, indipendentemente dal fatto che il metodo sia di tipo deterministico o aleatorio.
L'insieme dei metodi basati sull'imputazione multipla ha il compito di fornire più stime dei valori mancanti e di combinarli tra di esse. Come nei casi precedentemente descritti, un'assunzione importante è l'ignorabilità del meccanismo dei dati mancanti, anche se si possono ottenere buoni risultati in presenza di MNAR.
L'approccio sottostante è quello bayesiano ed essenzialmente fornisce una soluzione naturale per derivare metodi di stima dei parametri tali da prendere in considerazione molteplici fonti di incertezza. Nel contesto dei dati incompleti, è auspicabile incorporare all'inferenza un'incertezza aggiuntiva per la mancanza di alcuni dati.
Distinguendo dal parametro θ utilizzato per la verosimiglianza, si indica con β un parametro d'interesse nel caso bayesiano. Poiché gli unici dati osservati sono X_{oss} e R, un'analisi bayesiana per un parametro d'interesse β richiede il calcolo della distribuzione a posteriori $P(\beta|, R)$.
In linea di principio, si potrebbe sempre eseguire un'analisi bayesiana diretta per produrre la distribuzione a posteriori, ma in pratica ciò potrebbe richiedere dei procedimenti complessi. Tuttavia, una semplice applicazione standard della probabilità condizionata mostra che la distribuzione a posteriori può essere espressa in modo da separare il problema dei dati mancanti dal modello di analisi dei dati di interesse:

$$P(\beta|X_{oss}, R) = \int P(\beta|X_{oss}, X_{mis})P(X_{mis}|X_{oss}, R)dX_{mis} \tag{2.9}$$

Questo integrale deve essere considerato come una somma nel caso di X_{mis} discreti. Tale rappresentazione si basa sul fatto che l'integrale rappresenta la distribuzione a posteriori per β dato un set di dati completo. L'intuizione chiave dell'imputazione multipla consiste nel vedere se si possa trovare un modo per generare o imputare un campione di m valori $X_{mis}^{(k)}$, con $k = 1, ..., m$, dalla distribuzione predittiva per i dati mancanti, cioè dalla distribuzione $P(X_{mis}|X_{oss}, R)$.
In tal caso la distribuzione a posteriori può essere approssimata tramite la media della distribuzione a posteriori dei dati completi valutata per ciascuno dei valori imputati X_{mis}:

$$P(\beta|X_{oss}, R) \approx \frac{1}{m}\sum_{i=1}^{m} P(\beta|X_{oss}, X_{mis}^{(k)}) \tag{2.10}$$

Sotto l'ipotesi di ignorabilità, il valore atteso del parametro d'interesse β si può

ottenere applicando le regole di Rubin dei valori attesi iterati e si esprime come:

$$E(\beta|X_{oss}, R) = E(E(\beta|X_{oss}, X_{mis})|X_{oss}, R) \tag{2.11}$$

mentre la varianza di β è:

$$\begin{aligned} Var(\beta|X_{oss}, R) =& E(Var(\beta|X_{oss}, X_{mis})|X_{oss}, R) \\ &+ Var(E(\beta|X_{oss}, X_{mis})|X_{oss}, R) \end{aligned} \tag{2.12}$$

Queste quantità possono essere stimate a partire dai dati imputati approssimando i relativi integrali. Si ottiene così il valore atteso di β approssimato:

$$E(\beta|X_{oss}, R) \approx \frac{1}{m}\sum_{i=1}^{m}\hat{\beta}^{(k)} = \hat{\beta}^{mis} \tag{2.13}$$

e quello della varianza approssimata:

$$\begin{aligned} Var(\beta|X_{oss}, R) &\approx \frac{1}{m}\sum_{i=1}^{m}\hat{V}^{(k)} + \frac{1}{m-1}\sum_{i=1}^{m}(\hat{\beta}^{(k)} - \hat{\beta}^{mis})^2 \\ &= \bar{V} + B = \hat{V}^{mis} \end{aligned} \tag{2.14}$$

Si nota che $B = \frac{1}{m-1}\sum_{i=1}^{m}(\hat{\beta}^{(k)} - \hat{\beta}^{mis})^2$ è una stima della varianza tra le imputazioni del parametro di interesse β, dove $\hat{\beta}^{(k)}$ è la stima del parametro ottenuta dal k-esimo set di dati completo e $\hat{\beta}^{mis}$ è la stima del valore atteso approssimato dei parametri.

In sintesi, queste quantità offrono valide approssimazioni per i primi due momenti della distribuzione a posteriori per valori grandi di m. Per piccoli valori di m la formula della varianza deve includere un termine aggiuntivo, proporzionale a $\frac{1}{m}$, per riflettere l'incertezza nel valore di β^{mis} come stima della vera media a posteriori. Questo porta alla stessa espressione di cui sopra per la media a posteriori, mentre come approssimazione per la varianza a posteriori si ottiene:

$$\hat{V}^{mis} = \bar{V} + \left(1 + \frac{1}{m}\right)B \tag{2.15}$$

Date queste approssimazioni a posteriori dei primi due momenti, si possono ottenere intervalli di credibilità e test statistici assumendo che $(\hat{\beta}^{mis} - \beta)/\sqrt{V^{mis}}$ segua una distribuzione normale standard, per un numero elevato di imputazioni m, o una distribuzione t di Student per un numero di imputazioni qualsiasi.
Ad esempio, per m grande, si ottiene il seguente intervallo di credibilità di livello $(1-\alpha)$ per β:

$$IC_{1-\alpha}(\beta) = \hat{\beta}^{mis} \pm z_{1-\alpha}\sqrt{\hat{V}^{mis}} \tag{2.16}$$

con $z_{1-\alpha}$ che rappresenta il quantile di livello $(1-\alpha)$ di una distribuzione normale standard.

Andiamo ora ad offrire una panoramica sommaria dei passaggi che vanno a comporre

il procedimento per l'ottenimento di imputazioni multiple:

1. Imputazione: si generano un numero m di copie del dataset incompleto e si utilizza una procedura appropriata comune per imputare i valori mancanti in ciascuna di queste copie. Poiché non si conoscono i veri valori mancanti, i valori imputati in ogni copia possono essere in generale diversi l'uno dall'altro.

2. Analisi: Per ogni copia imputata, si esegue, applicando i relativi modelli, l'analisi standard che sarebbe stata eseguita in assenza di valori mancanti e si memorizzano le stime dei parametri di interesse insieme ai loro errori standard stimati o alla matrice di varianze e covarianze. La stima dei parametri ottenuta dal k-esimo set di dati completo, con $k = 1, ..., m$, è indicata con $\hat{\beta}^{(k)}$ e la sua varianza stimata come $\hat{V}^{(k)}$.

3. Pooling: si mettono in comune i risultati degli m dataset imputati attraverso le regole di Rubin. In primo luogo si ottiene una stima combinata dei parametri, come ad esempio la media delle m stime singole, denotata ciascuna con $\hat{\beta}^{mis}$. Poi si calcola l'errore standard per questa stima come radice quadrata della seguente varianza combinata $\hat{V}^{mis}$. Infine l'inferenza a imputazione multipla procede nel modo consueto a partire da questi risultati, formando statistiche test e intervalli di credibilità come sopra descritto.

2.4.1 Strategie per l'Imputazione Multipla

Per applicare la tecnica dell'imputazione multipla esistono numerose diverse strategie. In genere non esiste una strategia che risulti assolutamente migliore delle altre. Un aspetto fondamentale rimane il bilanciamento tra proprietà statistiche e costo computazionale di ciascun metodo. Si vogliono quindi qui esplorare tre fra le più diffuse ed efficienti strategie esistenti.

Data Augmentation

L'idea generale dell'imputazione attraverso la Data Augmentation, o incremento dei dati, consiste nell'utilizzare simulazioni Markov Chain Monte Carlo (MCMC) per estrarre imputazioni casuali da una distribuzione multivariata a posteriori. L'estrazione dei valori direttamente dalla distribuzione multivariata è tuttavia difficile a causa della mancanza di dati.
Invece di fare riferimento direttamente alla distribuzione, il processo può essere semplificato. Aggiornando la distribuzione dopo ogni estrazione e generando una sequenza, o una catena, di queste imputazioni, è possibile infatti approssimare la distribuzione multivariata effettiva e utilizzarla per l'estrazione delle imputazioni. In sostanza, questa procedura prevede di specificare innanzitutto una distribuzione multivariata che si presume descriva accuratamente i dati e un insieme di parametri iniziali $\dot{\beta}^{(0)}$. Solitamente viene utilizzata una distribuzione normale multivariata.

Successivamente i dati vengono aggiornati in due step di estrazione:

1. Imputation Step: $\dot{X}_{mis}^{(t)} \sim P(X_{mis}|X_{oss}, \dot{\beta}^{(t-1)})$

2. Posterior Step: $\dot{\beta}^{(t)} \sim P(\beta|\dot{X}^{(t)}_{mis}, X_{oss})$

dove X_{mis} sono i dati effettivamente mancanti, $\dot{X}^{(t)}_{mis}$ sono i dati imputati, o incrementati, t-esimi per X_{mis} e $\dot{\beta}^{(t)}$ sono i parametri della distribuzione estratti t-esimi.

Questi step sono ripetuti fino alla convergenza della distribuzione e, conseguentemente, i dati mancanti vengono imputati per formare il dataset completo. Questa procedura viene applicata m volte al fine di generare m diversi dataset completi.
Affinché le imputazioni siano indipendenti, è necessario effettuare un adeguato numero di iterazioni per ogni imputazione così da garantire che la dipendenza temporale della catena di Markov sia eliminata.
La Data Augmentation ha il vantaggio di imputare i dati in maniera aleatoria dalla corretta distribuzione approssimata dei valori mancanti. D'altra parte non esiste un criterio di convercenza per la catena di Markov, forzando l'algoritmo ad un numero arbitrario di iterazioni. Inoltre, è necessario esplicitare una distribuzione multivariata per i dati, il che può non sempre risultare immediato, e un valore iniziale per il parametro $\dot{\beta}^{(0)}$. Per quest'ultimo solitamente si utilizza un algoritmo di pre-elaborazione di tipo Expectation-Maximization.
Concludendo, questa strategia di imputazione risulta avere delle ottime proprietà, ma richiede delle assunzioni distributive forti e un costo computazionale elevato.

Specificazione Completamente Condizionale

La Specificazione Completamente Condizionale (Fully Conditional Specification) è un metodo alternativo alla Data Augmentation che non richiede l'esplicitazione di una distribuzione multivariata dei dati.
Questo metodo elabora invece la distribuzione congiunta dei dati implicitamente attraverso un insieme di distribuzioni condizionate in cui ogni singola variabile è condizionata a tutte le altre. Ciò consente di specificare distribuzioni condizionate diverse per ogni variabile e, di conseguenza, significa che è possibile specificare distribuzioni appropriate per ogni tipo di dati.
La versione più diffusa dell'algoritmo di base è l'imputazione multipla attraverso equazioni concatenate (MICE), ovvero un metodo MCMC per imputare i dati attraverso le distribuzioni condizionate di tutte le variabili.
Per molti versi MICE assomiglia alla Data Augmentation, tranne per il fatto che effettua estrazioni solo per una variabile alla volta, condizionata alle altre, e ripete la procedura per tutte le variabili in gioco.

In particolare, si parte dall'imputazione per una variabile j-esima dei valori $\dot{X}^{(0)}_j$ e si procede con i seguenti step:

1. Estrazione dei Parametri: $\dot{\beta}^{(t)}_j \sim P(\beta^{(t)}_j|X^{oss}_j, \bar{X}^{(t)}_{-j})$

2. Estrazione dei Valori: $\dot{X}^{(t)}_j \sim P(X^{mis}_j|\bar{X}^{t}_{-j}, \dot{\beta}^{(t)}_j)$

dove X^{oss}_j sono i dati osservati per la j-esima variabile, X^{mis}_j sono i dati effet-

tivamente mancanti per la variabile j-esima e $\bar{X}_{-j}^{(t)}$ sono i dati completi a seguito dell'imputazione a meno della j-esima variabile.

Questo algoritmo mantiene le stesse proprietà della Data Augmentation e non necessita della specificazione esplicita della distribuzione multivariata. Il costo computazionale rimane in ogni caso elevato, ma, nonostante tutto, si dimostra essere uno degli algoritmi più efficienti per l'imputazione multipla dei dati mancanti.

Ricampionamento Bootstrap

Il metodo di ricampionamento di tipo Bootstrap viene spesso implementato a seguito dell'algoritmo Expectation-Maximization (EM), il quale, a differenza della Data Augmentation, risulta essere un passaggio praticamente obbligatorio al fine di mantenere delle buone proprietà statistiche.
L'algoritmo EM è stato sviluppato come metodo iterativo per calcolare le stime di massima verosimiglianza per i parametri in distribuzioni in cui tali parametri non potevano essere stimati direttamente a causa dei dati mancanti. Esso ha lo scopo di massimizzare la verosimiglianza per il parametro β.
La convergenza di tale algoritmo è deterministica, rappresentando quindi un notevole vantaggio rispetto ad altre strategie di imputazione multipla.

L'algoritmo si avvia con la stima iniziale dei parametri $\dot{\beta}^{(0)}$ tramite Listwise Deletion e prosegue con i seguenti step:

1. Expectation Step: imputazione di $\dot{X}^{(t)}$ come $\dot{X}^{(t)} = E(X_{mis}|X_{oss}, \dot{\beta}^{(t-1)})$

2. Maximization Step: stima di $\dot{\beta}^{(t)}$ massimizzando il valore atteso della log-verosimiglianza del parametro, ovvero $G(\beta|\bar{X}^{(t)}) = E(logL(\beta|\bar{X}^{(t)}))$

Basandosi su una procedura deterministica, quest'algoritmo non presenterebbe variabilità nell'imputazione multipla dei dati mancanti. Per questo motivo viene introdotto il processo di ricampionamento bootstrap.
Inizialmente viene infatti estratto un campione con sostituzione dai dati osservati, dopodiché, applicando nuovamente l'algoritmo EM, si introduce un fattore di incertezza alle stime che viene poi utilizzato per generare il dataset completo tramite imputazione. Chiaramente questa procedura viene ripetuta m volte.
I risultati ottenuti con il Ricampionamento Bootstrap sono in generale leggermente meno soddisfacenti rispetto ai metodi precedenti; d'altra parte, offre un costo computazionale decisamente ridotto.

2.4.2 Numero di Imputazioni

Uno dei vantaggi dell'imputazione multipla è che può produrre stime efficienti con intervalli di credibilità corretti con un basso numero di set di dati imputati.
Un suggerimento è quello di utilizzare un numero di imputazioni m tra 3 e 5 per quantità moderate di dati mancanti. Spesso numerosi studi portano invece ad indicare un numero di imputazioni consigliato più elevato, compreso tra 20 e 100.

Vogliamo ora offrire un possibile approccio per la scelta del numero di imputazioni. Per iniziare si definisce una quantità di interesse Q che si potrebbe calcolare se osservassimo l'intera popolazione. Esempi di quantità d'interesse sono la media, la varianza o i coefficienti di regressione della popolazione.
In linea teorica si potrebbe calcolare Q solo se i dati della popolazione fossero completamente noti. L'obiettivo dell'imputazione multipla è quello di trovare una stima $\hat{Q}$ non distorta. Con il termine non distorsione, o correttezza, s'intende che la media di $\hat{Q}$ su tutti i possibili campioni X della popolazione sia uguale a Q.

In altri termini, significa che:

$$E(\hat{Q}|Y) = Q \tag{2.17}$$

Nel caso dell'imputazione multipla, supponendo che $\hat{Q}_k$ sia la k-esima stima di Q, con $k = (1, ..., m)$, allora la stima combinata di Q è:

$$\bar{Q} = \frac{1}{m} \sum_{k=1}^{m} \hat{Q}_k \tag{2.18}$$

Definiamo poi la varianza totale (di imputazione) di $\bar{Q}$ come:

$$T = \bar{U} + \left(1 + \frac{1}{m}\right) B \tag{2.19}$$

dove:

- $U = \frac{1}{m} \sum_{k=1}^{m} \bar{U}_k$ è la media delle varianze $\bar{U}_k$ dei dati completi, ovvero la varianza relativa al fatto che si prenda in considerazione un campione aleatorio piuttosto che osservare l'intera popolazione; questa è la misura statistica convenzionale di variabilità.
- $B = \frac{1}{m-1} \sum_{k=1}^{m} (\hat{Q}_k - \bar{Q})(\hat{Q}_k - \bar{Q})^T$ é la stima standard non distorta della varianza tra le m stime dei dati completi, in altre parole è la varianza aggiuntiva causata dalla presenza di valori mancanti nel campione.
- $\frac{B}{m}$ é la varianza aggiuntiva della simulazione causata dal fatto che Q stesso è stimato per un numero finito di imputazioni m.

Un elemento chiave dell'imputazione multipla è la presenza dell'errore di simulazione per $\bar{Q}$ e T. Si può dimostrare che un numero sempre più elevato di imputazioni ci porterebbe sempre più vicini all'eliminazione dell'errore di simulazione di T.
In altri termini, impostando $m = \infty$ si ha che $T_\infty < T_m$. Le varianze totali sono, inoltre, legate dalla relazione:

$$T_m = \left(1 + \frac{\gamma_0}{m}\right) T_\infty \tag{2.20}$$

dove γ_0 è la vera frazione di dati mancanti della popolazione.

Questa quantità è pari alla frazione attesa di osservazioni mancanti se X è una

singola variabile.
Ad esempio, per $\gamma_0 = 0,2$ (una sola variabile con il 20% di valori mancanti) e $m = 5$ si ha che la varianza calcolata è $T_m = 1 + \frac{0.2}{5} = 1,04$, ovvero risulta più grande del 4% rispetto alla varianza ideale T_∞. L'intervallo di confidenza corrispondente risulterebbe $\sqrt{1,04} = 1,02$, vale a dire il 2% più ampio rispetto all'ideale.
Aumentando il numero di imputazioni m a 10 o a 20, si osserverebbe un intervallo di confidenza rispettivamente del 1% e del 0,5% rispetto all'ideale. Un incremento del numero di imputazioni non porterebbe quindi ad un grande miglioramento in termini di errore di simulazione, bensì ad un costo computazionale decisamente maggiore.
Un criterio di selezione di m potrebbe quindi basarsi sul raggiungimento di una prefissata ampiezza dell'intervallo di confidenza.
In letteratura, esistono altri approcci per la scelta del numero di imputazioni che si basano per lo più sugli intervalli di confidenza, sulla potenza dei test statistici e, in generale, su tutte le quantità che fanno riferimento ad m.

2.5 Selezione del Metodo di Imputazione

Confrontare i vari metodi di imputazione può risultare un problema a livello teorico non indifferente. Non è infatti possibile conoscere la controparte reale dei valori che vengono imputati. Si potrebbe pensare che un metodo di selezione possa essere basato sulle classiche misure di accuratezza indicate solitamente per valutare i modelli previsivi. I metodi che verrebbero selezionati non risulterebbero corretti in quanto una maggior capacità previsiva di un metodo non corrisponde spesso ad un miglior approccio per imputare i dati mancanti.
Metriche come la radice dell'errore quadratico medio (RMSE), l'errore medio assoluto (MAE) o l'errore medio assoluto percentuale (MAPE) tenderebbero, infatti, a favorire a prescindere degli specifici metodi di imputazione, non tenendo conto in maniera esaustiva delle loro proprietà. In particolare l'RMSE predilige metodi basati sull'imputazione della media condizionata, l'MAE favorisce i metodi che imputano tramite la mediana condizionata e l'MAPE quelli che imputano sfruttando la moda condizionata.
Un esempio si può avere quando si va a confrontare l'imputazione attraverso il modello di regressione lineare con l'imputazione attraverso il modello di regressione stocastica. In questo caso, le metriche tradizionali indicheranno molto probabilmente la regressione lineare come migliore metodo di imputazione nonostante presenti proprietà statistiche peggiori.
Una proprietà desiderabile per i metodi di imputazione è la loro capacità di preservare le distribuzioni congiunte e marginali dei dati. A tal fine si introduce un criterio di selezione del metodo di imputazione più oggettivo e statisticamente più consistente che prende il nome di Imputation Score.

Imputation Score

Il criterio di selezione basato sugli Imputation Scores (o I-Scores) fa riferimento alle medie condizionate osservate dei dati. L'I-Score è un criterio di selezione ideale per i metodi di imputazione quando:

- l'obiettivo dell'imputazione è quello di riprodurre fedelmente la vera distribuzione dei dati;
- non è possibile avere accesso ai dati completi;
- non si vogliono mascherare artificialmente le osservazioni durante la fase di valutazione come avviene per i criteri di selezione classici.

Per superare la difficoltà nel poter osservare i soli dati incompleti, l'I-Score fa utilizzo delle proiezioni geometriche casuali sullo spazio delle variabili per ridurre la dimensionalità dei dati. Questo criterio è applicabile soltanto a dati quantitativi discreti o continui. Per questo risulta necessario pre-codificare i dati in modo adeguato.
Lo score più elevato viene assegnato al metodo di imputazione che meglio riproduce la distribuzione condizionata dei dati osservati. E' stato empiricamente dimostrato che con i dati veri si otterrebbe generalmente lo score massimo. Di conseguenza, il metodo di imputazione che propone dei valori più vicini alla controparte reale sarà catalogato come il migliore.
La correttezza di questa procedura è dimostrata sotto l'ipotesi di MCAR, ma anche nel caso di MAR con assunzioni leggermente più restrittive.
L'I-Score teorico può essere stimato nella sua versione principale come rapporto di densità, il quale prende il nome di Density Ratio I-Score.

Gli I-Scores si rifanno al concetto di Proper Scores, ma con le dovute differenze. Sia $(\Omega, \mathcal{A}, \mathbf{P})$ lo spazio di probabilità sottostante su cui si denotano tutti gli elementi aleatori necessari all'analisi e $\mathcal{P}$ una collezione di misure di probabilità su R^n dominate da alcune misure σ-finite μ.
Si definisce con P la distribuzione osservata dei dati X con valori mancanti e, in maniera analoga, $P^* \in \mathcal{P}$ si riferisce alla vera distribuzione di X indicata con X^*.
Similmente si denota con P^M, con supporto $\mathcal{M}$, la distribuzione del vettore aleatorio di non-risposta M in $\{0,1\}^n$ (i suoi valori sono opposti alla matrice indicatrice di risposta R e m rappresenta una sua realizzazione) per X.
Inoltre, per un sottoinsieme $A \subseteq \{1, ..., n\}$ e per un vettore aleatorio X, o una osservazione $x \in R^n$, si indica con X_A (o x_A) la proiezione definita su quel sottoinsieme di indici.
Si ha allora che (P, P^*) forma una tupla in cui P deriva da P^* e P^M e $\mathcal{H}_P \subset P$ è l'insieme delle distribuzioni di imputazione compatibili con P, ovvero:

$$\mathcal{H}_P = \{H \in P : h(o(x,m)|M=m) = p(o(x,m)|M=m), \forall : m \in \mathcal{M}\} \qquad (2.21)$$

dove $o(x,m)$ è la controparte osservata di x corrispondente alla realizzazione m, mentre $h(\cdot)$ e $p(\cdot)$ sono le distribuzioni di H e $\mathcal{P}$.

Chiaramente si ha che $P^* \in \mathcal{H}_P$, di conseguenza, la vera distribuzione P^* può essere vista come un'imputazione compatibile.
La funzione $S_{NA}(\cdot, P) : \mathbf{R}^n \to \mathbf{R}$, definita come $S_{NA}(H,P) = E_{Y \sim H}(S_{NA}(X,P))$ è il valore atteso di $X \sim H$. Se tale funzione rispetta la condizione:

$$S_{NA}(H,P) \leq S_{NA}(P^*,P) \qquad (2.22)$$

per qualsiasi distribuzione di imputazione $H \in \mathcal{H}_P$, allora prende il nome di I-Score.

Il metodo di imputazione che restituirà il valore di I-Score più elevato, risulterà il metodo migliore e quello da selezionare.
La stima degli I-Scores attraverso Density Ratios (o rapporti di densità) permette di aggirare il problema della mancata osservazione di P^* in maniera efficiente utilizzando appunto delle proiezioni aleatorie su uno spazio delle variabili di dimensione ridotta. Ogni proiezione è scelta aleatoriamente secondo una determinata distribuzione di probabilità K con supporto $\mathcal{A}$.
Dato allora un set di indici di proiezione $A \in \mathcal{A}$ e un pattern di dati mancanti $M_A \sim P_A^M$ sulla proiezione corrispondente $X_A \sim H_{M_A}$, si definisce la funzione:

$$S_{NA}^*(X_A, P_A|M_A) = log\left(\frac{p_A(X_A|M_A = 0)}{h_{M_A}(X_A)}\right) \tag{2.23}$$

dove $p_A(X_A|M_A = 0)$ è la densità della parte completamente osservata di P proiettata su A e $h_{M_A}(X_A)$ è la distribuzione di un'imputazione H, dato il pattern di dati mancanti m_A sulla proiezione X_A.

Il Density Ratio I-Score della distribuzione delle imputazioni H è:

$$S_{NA}^*(H, P) = E_{A\sim K, M_A\sim P_A^M, X_A\sim H_{M_A}}(S_{NA}^*(X_A, P_A|M_A)) \tag{2.24}$$

Si rimanda all'articolo in bibliografia di Jeffrey Naf e Meta-Lina Spohn del 2021 per una trattazione teorica completa.

Capitolo 3

Modelli Previsivi

Per trattare la componente spaziale nei problemi relativi al mercato immobiliare, è possibile utilizzare dei modelli previsivi di regressione che non possiedono esplicitamente all'interno della formulazione dei parametri spaziali. Si può invece tener conto della spazialità definendo delle nuove variabili associate ai diversi sottomercati spaziali di una determinata area.
In questo capitolo, si vuole quindi stabilire la natura delle variabili spaziali e presentare la teoria relativa ai modelli statistici principalmente utilizzati per la previsione dei prezzi. Infine, si esporrà una strategia per selezionare il modello basata su degli indicatori di accuratezza delle previsioni.

3.1 Variabili Spaziali

Un valido approccio per trattare gli eventuali effetti spaziali in un insi eme di dati tramite modelli previsivi non spaziali consiste nell'introdurre variabili spaziali che esprimono la collocazione geografica delle unità statistiche. Nel nostro caso le variabili indicheranno se un'unità è situata in una determinata regione all'interno della città, ovvero andranno a definire dei sottomercati spaziali.
Ispirandoci al lavoro di Bor-Ming Hsieh del 2012, esponiamo tre metodi basati sull'identificazione di alcuni tipi di sottomercato spaziale dei prezzi delle abitazioni. Il primo metodo prende in considerazione solo l'indirizzo dell'immobile, mentre gli altri due metodi esaminano anche le caratteristiche dell'abitazione.
I sottomercati spaziali hanno il compito di determinare delle zone secondo le quali esistono degli immobili i cui prezzi sono in relazione tra loro. Avranno quindi in questo senso il ruolo di variabili esplicative all'interno dei modelli tradizionali applicati.

Aree Amministrative

Una semplice tecnica che può essere utilizzata per tenere in considerazione gli effetti spaziali si serve della conoscenza relativa all'area amministrativa di appartenenza dell'immobile. Una differente collocazione amministrativa di un'abitazione implica molto spesso un diverso valore di vendita sul mercato. Solitamente nelle zone più centrali di una città hanno sede infatti attività sociali ed economiche maggiori

e, conseguentemente, i prezzi di transazione mantengono un livello più elevato. Al contrario, le aree periferiche saranno meno ambite sul mercato immobiliare determinando in genere un minore prezzo di vendita.
Nelle varie città del mondo ci sono diversi livelli di confini amministrativi a seconda della dimensione delle aree che si prendono come riferimento. Scegliere un corretto livello di suddivisione territoriale risulta quindi importante al fine dell'applicazione dei modelli. Si consiglia di selezionare un livello corrispondente ad un adeguato potere amministrativo. Maggiore è la rilevanza dell'area amministrativa, maggiori sono gli effetti sul mercato immobiliare. Accade spesso che in una determinata area si fissi il prezzo al metro quadro degli immobili o che vengano stabiliti dei prezzi soglia di riferimento da parte di coloro che possiedono la responsabilità del lotto.
Nei modelli statistici queste considerazioni si riflettono sul processo di selezione del modello e sulla variazione probabilistica dei risultati.

Cluster Spaziali

La Cluster Analysis può essere impiegata per selezionare e raggruppare insiemi di immobili con caratteristiche omogenee. Gli immobili di uno stesso cluster andranno quindi a formare un possibile sottomercato abitativo spaziale.
La variabile indicante il cluster svolgerà il ruolo di predittore nei modelli statistici che si applicheranno. Le variabili che vengono utilizzate per la costruzione dei cluster devono essere scelte tra le esplicative ed essere rilevanti per la determinazione di un sottomercato. Nell'articolo di riferimento originale sono state analizzate le seguenti variabili: la superficie costruita, la superficie del lotto, l'anno di costruzione, la larghezza della strada per accedere al lotto e la distanza dal centro città.
Per la finalità di questa ricerca, si ritiene che alcune di queste variabili siano ridondanti o non esaustive al fine di descrivere un determinato sottomercato. Per questo motivo si propone un altro sistema di variabili simile al precedente. Per descrivere le caratteristiche dell'immobile si consiglia di utilizzare la superficie costruita, il numero di bagni, il numero di stanze e il piano d'ingresso. Queste variabili hanno la caratteristica di essere numeriche, di conseguenza i modelli per il clustering applicabili sono più semplici. Osservando la differenza con le variabili dell'articolo, si nota che la superficie del lotto sia infatti ridondante, data la presenza della superficie costruita. Inoltre, il numero di bagni, di stanze e il piano d'ingresso appaiono delle caratteristiche più significative rispetto alle precedenti.
Alle variabili appena indicate, si suggerisce di aggiungere le coordinate geografiche dell'immobile a sostituzione della distanza dal centro città. Le coordinate servono ad indicare l'esatta posizione dell'immobile all'interno dell'area di studio.
Alcuni dei modelli di clustering particolarmente noti per la loro efficienza sono quelli a mistura finita con componenti Gaussiane. L'algoritmo con cui vengono generalmente implementati è quello di tipo EM e la selezione del modello avviene tramite il criterio di informazione Bayesiana (BIC). Si rimanda al libro di Nizar Bouguila e Wentao Fao del 2020 per gli aspetti teorici legati ai modelli a mistura finita con componenti Gaussiane.

Cluster LISA

Il metodo di clustering basato sull'Indicatore Locale di Associazione Spaziale (LISA) rileva, a differenza dei precedenti, se esista una dipendenza spaziale significativa dei prezzi degli immobili in un determinato confine.
Per definizione il LISA è un qualsiasi indicatore che soddisfa le seguenti due condizioni:

- per ogni osservazione fornisce un'indicazione dell'entità del raggruppamento spaziale con i valori misurati sulle osservazioni vicine;
- la somma dei valori dell'indicatore per tutte le osservazioni è proporzionale a un indicatore globale di associazione spaziale.

Prenderemo in esame il LISA calcolato a partire dall'indice di Moran locale, il quale misura l'autocorrelazione spaziale tra le osservazioni. Nel caso trattato il LISA analizza la concentrazione spaziale dei prezzi delle case in base alla posizione geografica e ai valori assunti dalla variabile dipendente. Tratteremo più approfonditamente gli aspetti teorici legati all'indice di Moran e all'indicatore LISA nel capitolo successivo.
L'aspetto fondamentale è che i risultati del LISA possono essere clusterizzati e sono utili per identificare i sottomercati spaziali dei prezzi delle abitazioni.
Le informazioni combinate consentono infatti di classificare i sottomercati spaziali come cluster che si differenziano in base alla correlazione spaziale interna.
Si osserveranno quindi delle zone che presentano dei prezzi degli immobili elevati, altre zone con prezzi degli immobili bassi e, infine, zone con prezzi che tendono in una delle due direzioni senza favorirne una specifica.

3.2 Regressione Parametrica

Nella regressione parametrica i modelli sono costruiti introducendo uno o più parametri di interesse. La forma dei modelli e le relative assunzioni sono completamente specificate a priori. Prenderemo in esame dei modelli parametrici lineari e derivati.
In ambito immobiliare la regressione parametrica è ampiamente impiegata per specificare dettagliatamente i singoli effetti delle esplicative sulla risposta. Inoltre, forniscono una buona precisione anche con un numero minimo di osservazioni.

3.2.1 Modello Lineare

Il modello di regressione lineare multipla risulta essere il modello di base più utilizzato per molti tipi di analisi statistiche, tra cui la descrizione dei fattori che influiscono sui prezzi. Caratteristiche fondamentali che lo contraddistinguono sono l'interpretabilità e la facilità con cui può essere formulato.
Il modello per la variabile risposta si presenta nella seguente forma:

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij} + \epsilon_i \,, \;\; i = 1, ..., n \tag{3.1}$$

con:

- Y_i : variabile dipendente (o risposta) quantitativa continua per l' i-esima osservazione, con $i = 1, ..., n$;

- X_{ij} : variabile indipendente (o esplicativa) j-esima, con $j = 1, ..., p$, per l' i-esima osservazione, con $i = 1, ..., n$;
- β_0 : intercetta del modello, corrisponde al valore atteso di Y quando tutte le variabili esplicative sono nulle;
- β_j : coefficiente angolare (o coefficiente di regressione) per la variabile esplicativa X_j, con $j = 1, ..., p$;
- ϵ_i : errore statistico per l' i-esima osservazione, con $i = 1, ..., n$. Si assume che gli errori siano tra loro indipendenti ed identicamente distribuiti secondo una distribuzione normale di media nulla e varianza costante. In altri termini, si assume che $\epsilon_i \sim N(0, \sigma^2)$.

I parametri del modello sono l'intercetta β_0 e i coefficienti di regressione β_j.
Questi ultimi sono stimati minimizzando la somma dei quadrati dei residui:

$$S(\beta) = S(\beta_1, ..., \beta_p) = \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2 \tag{3.2}$$

di conseguenza, i parametri stimati risultano essere:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}\, S(\beta) \tag{3.3}$$

L'approccio utilizzato prende il nome di Metodo dei Minimi Quadrati Ordinari o Ordinary Least Squares (OLS) e i parametri stimati che ne conseguono corrispondono alle stime di massima verosimiglianza dei parametri β_j fissata la varianza σ^2.
Gli stimatori ai minimi quadrati hanno inoltre proprietà di ottimalità garantite dal teorema di Gauss-Markov.

Il modello di regressione lineare possiede le seguenti assunzioni:

1. Linearità: la relazione tra le variabili indipendenti e la variabile dipendente è di tipo lineare nei parametri, di conseguenza, non si possono applicare delle trasformazioni sui parametri, ma soltanto delle trasformazioni sulle variabili indipendenti.

2. Normalità: la variabile dipendente segue una distribuzione normale. Questa assunzione non è vincolante, poiché gli stimatori che si otterrebbero senza l'ipotesi di normalità hanno buone proprietà. I risultati inferenziali (intervalli di confidenza e verifiche di ipotesi) non possono però essere ottenuti in maniera semplice, anche se, tuttavia, per un numero elevato di osservazioni n gli stimatori risultano asintoticamente normali e restituiscono buoni risultati.

3. Omoschedasticità: la varianza della variabile dipendente è costante, ovvero $Var(Y_i) = \sigma^2$, con $i = 1, ..., n$. Come per la non normalità, gli stimatori dei minimi quadrati ordinari (OLS) hanno comunque buone proprietà, ma non c'è soluzione al problema relativo alle procedure inferenziali (anche con n grande). Quest'assunzione risulta quindi maggiormente vincolante rispetto alla precedente.

4. Indipendenza: le osservazioni della variabile dipendente sono tra loro linearmente indipendenti. Forme diffuse di dipendenza tra variabili sono l'autocorrelazione temporale per le serie storiche e l'autocorrelazione spaziale per i dati spaziali.

5. Indipendenza tra Variabili Esplicative: le variabili esplicative X_j sono tra loro linearmente indipendenti. In caso di dipendenza tra esplicative si parla di collinearità o multicollinearità.

Una procedura fondamentale per il modello regressivo lineare è la diagnostica del modello. I metodi diagnostici hanno l'obiettivo di verificare le assunzioni del modello eseguendo delle specifiche analisi grafiche e dei test appositi.
Le analisi vengono spesso condotte sui residui del modello di regressione, ovvero sulle differenze tra i valori osservati e i valori stimati della risposta.

3.2.2 Modello Lineare Generalizzato

I modelli lineari generalizzati sono una famiglia di modelli che vanno ad estendere il concetto di modello lineare. La loro caratteristica fondamentale è l'appartenenza della variabile risposta ad una specifica famiglia di dispersione esponenziale.
In termini di osservazioni sulla risposta si può esprimere la famiglia distributiva come segue:

$$p(y_i|\theta_i,\phi) = exp\left\{\frac{\theta_i y_i - b(\theta_i)}{a_i(\phi)} + c(y_i,\phi)\right\}, \;\; i = 1,...,n \tag{3.4}$$

con $\theta_i \in \Theta \subseteq R$, $a_i(\phi) > 0$.

Il parametro θ_i è detto parametro naturale, mentre ϕ è detto parametro di dispersione. Le funzioni $a(\cdot)$ e $b(\cdot)$ si riferiscono rispettivamente a ψ e θ_i e contribuiscono anch'esse alla determinazione della specifica distribuzione della risposta.

Per questa tipologia di modelli le assunzioni sottostanti sono analoghe a quelle del modello lineare, ciò nonostante risultano leggermente più flessibili.
In particolare, vengono mantenute le ipotesi di indipendenza sia per la variabile risposta che per le esplicative.

Le altre assunzioni possono essere così espresse:

1. Linearità : $g(E(Y_i)) = g(\mu_i) = \eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}$
dove $g(\cdot)$ è una funzione liscia invertibile nota che prende il nome di funzione di legame (o link function), mentre η_i è il predittore lineare i-esimo.
L'ipotesi di linearità ricade quindi non più sulla risposta, ma sulla funzione di legame.

2. Distribuzione : $Y_i = DE_1(\mu_i, a_i(\psi)v(\mu_i))$
La variabile dipendente segue una particolare distribuzione appartenente alla famiglia di dispersione esponenziale come espresso dall'equazione (3.4).

I parametri β_j del modello sono stimati attraverso i minimi quadrati pesati iterati. L'algoritmo che svolge quest'operazione è detto anche metodo di Newton-Raphson. Gli stimatori risultanti hanno, come nel caso del modello lineare, buone proprietà statistiche di ottimalità.
I metodi diagnostici del modello lineare generalizzato sono simili a quelli del modello lineare con dei dovuti aggiustamenti. Analizzeremo più a fondo questo aspetto nella parte applicativa.

Poiché nella nostra applicazione si costruirà un modello di regressione per una variabile considerabile come quantitativa continua (il prezzo di vendita), andiamo a presentare un modello lineare generalizzato apposito.
Nel caso in questione si suppone che la variabile risposta segua una distribuzione Gamma del tipo $Y_i \sim Ga(\alpha, \lambda_i)$. Questa distribuzione si può anche esprimere in termini di famiglia esponenziale come segue:

$$p(y_i|\lambda_i, \alpha) = exp\Big\{-\lambda_i y_i + \alpha \ln \lambda_i\Big\} \frac{y_i^{\alpha-1}}{\Gamma(\alpha)}\ ,\ \ y_i, \lambda_i, \alpha > 0 \tag{3.5}$$

In particolare, risulta che il valore atteso è $E(Y_i) = \mu_i = \frac{\alpha}{\lambda_i}$, la varianza $Var(Y_i) = \frac{\mu_i^2}{\alpha}$, il parametro naturale $\theta_i = -\frac{1}{\mu_i} = -\frac{\lambda_i}{\alpha}$ e il parametro di dispersione $\phi = \frac{1}{\alpha}$.
I parametri α e λ_i sono quelli tipici di una distribuzione Gamma e rappresentano rispettivamente il parametro di forma e di scala.

3.2.3 Modelli con Regolarizzazione

La regolarizzazione (o shrinkage) è una tecnica che ha l'obiettivo di ridurre il problema dell'overfitting di un modello.
Molto spesso accade, infatti, che il modello selezionato costruito sui dati di training non si adatti adeguatamente ad un nuovo dataset. Questo è dovuto al fatto che si debba trovare un bilanciamento tra la distorsione e la variabilità del modello.
Un modello non distorto che si adatta perfettamente ai dati di training avrà frequentemente difatti delle capacità previsive ridotte sul dataset di test.
La regolarizzazione introduce in questo senso una penalità durante l'applicazione del metodo di stima dei parametri. Questa tecnica si può applicare ad una moltitudine di modelli, ma faremo riferimento ai modelli lineari e lineari generalizzati.
Esistono diversi tipi di regolarizzazione. Le più frequenti sono la Ridge e il Lasso. Spesso si parla infatti di Regressione Ridge o Regressione Lasso. Verrà inoltre presentato un tipo di regolarizzazione che unisce il concetto di queste due procedure che prende il nome di Elastic Net.
Passaggio fondamentale per l'applicazione dei metodi di regolarizzazione risulta la standardizzazione delle variabili in fase di pre-processing in modo da dare a tutte le variabili lo stesso peso. In seguito all'applicazione dei metodi di regolarizzazione si possono successivamente ritrasformare le variabili nella loro scala originale.

Ridge

La Regressione Ridge è un metodo di regolarizzazione del modello che viene utilizzato soprattutto per analizzare dati che soffrono di multicollinearità. Questo metodo si basa su una metrica di tipo $L2$, in quanto penalizza per il quadrato del valore dei coefficienti di regressione del modello.
Nel caso, ad esempio, del modello di regressione lineare, la funzione di costo basata sui minimi quadrati assume la seguente forma:

$$S(\beta) = \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}\beta_j^2 \tag{3.6}$$

dove λ è la costante di restringimento che viene fissata.

Se λ è nullo si ricade nel caso della regressione lineare con stime basate sul metodo di stima OLS. Maggiore è λ, minore risulterà la variabilità del modello a discapito di una maggiore distorsione. Si osserva infatti che, i valori delle stime dei parametri β saranno più piccoli, e quindi meno rilevanti per la determinazione della risposta, tanto più elevato risulterà il valore della costante di restringimento.

Lasso

La Regressione Lasso si basa su una regolarizzazione di tipo $L1$. A differenza della Ridge, il Lasso ha la funzione di eseguire, oltre alla regolarizzazione, anche una selezione delle variabili esplicative.
Ner il modello di regressione lineare, la funzione di costo per il Lasso si presenta come segue:

$$S(\beta) = \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \mid \beta_j \mid \tag{3.7}$$

con λ costante di restringimento fissata.

Si può notare che la penalizzazione avviene attraverso il valore assoluto dei coefficienti di regressione. Come nel caso della Ridge, all'aumentare di λ aumenta la distorsione e diminuisce la variabilità del modello. Per valori sufficientemente elevati di λ, la regressione Lasso inizierà a restringere i coefficienti di regressione a tal punto da rendere quelli meno rilevanti nulli. Se il valore di λ tende ad ∞ verrà selezionato il modello nullo privo di alcuna variabile esplicativa.

Elastic Net

L'Elastic Net è un metodo di regolarizzazione che unisce la Regressione Ridge con la Regressione Lasso in un unico modello. Questo metodo si basa quindi sia sulla metrica $L1$ che su quella $L2$.
Per il modello lineare, la funzione di costo è:

$$S(\beta) = \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda_1 \sum_{j=1}^{p} \mid \beta_j \mid + \lambda_2 \sum_{j=1}^{p}\beta_j^2 \tag{3.8}$$

con λ_1 e λ_2 costanti di restringimento fissate riferite rispettivamente alla norma $L1$ e a quella $L2$.

Spesso, per motivi di efficienza e di costo computazionale, la funzione di costo si presenta nella seguente forma semplificata:

$$S(\beta) = \frac{\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p}\beta_j x_{ij})^2}{2n} + \lambda(\frac{1-\alpha}{2}\sum_{j=1}^{p}\beta_j^2 + \alpha\sum_{j=1}^{p} \mid \beta_j \mid) \tag{3.9}$$

con λ costante di restringimento fissata comune e α costante che regola il rapporto tra l'influenza della norma $L1$ ed $L2$.

Si osserva che, se la costante α è pari a 0 si rientra nel caso della Regressione Ridge, mentre se α è 1, si ha una Regressione Lasso.
L'Elastic Net risulta quindi un'evoluzione più flessibile rispetto ai primi due metodi di regolarizzazione e può controllare sia la selezione delle variabili che il problema di multicollinearità. La costante α rende, d'altra parte, il modello più complesso e difficile da implementare.

3.3 Regressione Non-Parametrica

Nella regressione non-parametrica la forma dei modelli non viene specificata a priori, ma viene invece determinata in base ai dati osservati. Il termine non-parametrico non è riferito all'assenza di parametri, ma al fatto che il numero dei parametri è flessibile e non fissato a priori. Talvolta i parametri dei modelli non-parametrici sono detti iperparametri. Un vantaggio rispetto alla regressione parametica è data dalla ridotta quantità di assunzioni sottostanti i modelli e dalla minore sensibilità ai valori anomali. D'altra parte, la regressione non-parametrica può essere poco precisa in presenza di poche osservazioni e i risultati possono essere meno facili da interpretare. In particolare, modelli che discuteremo sono il K-Nearest Neighbours e il MARS.

3.3.1 Modello K-Nearest Neighbours Regressivo

Il k-nearest neighbours (KNN) è un semplice modello non-parametrico basato sulle distanze. Spesso il KNN viene impiegato per gestire problemi di classificazione, ma il suo utilizzo trova spazio anche per analisi di regressione.
La distanza tra le osservazioni è la misura di similarità su cui si basa questo modello. Esistono molteplici tipi di distanza. Quelle più diffuse per le variabili quantitative continue sono la Distanza Euclidea e la distanza di Mahalanobis, mentre per variabili categoriche si usa solitamente la Distanza di Hamming.

L'algoritmo di implementazione del KNN poggia su alcuni semplici passaggi:

1. Scelta di k: si fissa un numero a priori di gruppi k a cui può appartenere ogni singola osservazione.

2. Calcolo delle Distanze: per ogni osservazione si calcolano le distanze multidimensionali basate sulle variabili esplicative rispetto a tutte le altre osservazioni del dataset. In questo modo si vanno a formare dei vettori di distanza per ciascuna osservazione.

3. Ordinamento: per ciascuna osservazione si ordinano i vettori delle distanze appena calcolate in ordine crescente.

4. Selezione: da ciascun vettore si selezionano le prime k distanze.

5. Stima: si stima il valore della variabile risposta per ciascuna osservazione, calcolando la media (o mediana) delle risposte delle k osservazioni con distanze selezionate.

Poiché il modello KNN è basato sulle distanze, è preferibile standardizzare le variabili in fase di pre-processing per evitare distorsioni dovute alla diversa scala di misura. Nonostante questo modello sia relativamente semplice e abbia un costo computazionale generalmente ridotto, spesso riesce a superare per capacità previsive molti tra i modelli più complessi. Questo modello può in definitiva essere utilizzato per avere delle stime dei prezzi di vendita in maniera rapida senza troppi compromessi.

3.3.2 Modello MARS

Il modello Multivariato Adattivo di Regressione con Splines (MARS) può essere visto come un'estensione dei modelli lineari che prevede componenti di non linearità e interazioni tra variabili.

Il modello si presenta come segue:

$$Y_i = a_0 + \sum_{m=1}^{M} a_m B_m(X_{i1}, ..., X_{ip}) \,, \;\; i = 1, ..., n \tag{3.10}$$

con a_0 intercetta del modello, a_m coefficiente costante dell'm-esima funzione base, B_m funzione di base m-esima e M numero totale di componenti.

La funzione di base può assumere due forme particolari:

- Hinge Function : è una funzione a gradini definita come

$$h(x_j) = \max(0, x_j - c) \tag{3.11}$$

  dove x_j è una variabile esplicativa, c è il punto di taglio (cut point) che divide l'intervallo di definizione in due segmenti, e α è un parametro di forma. La hinge function ha un valore di 0 per $x_j < c$ e un valore crescente per $x_j \geq c$.

- Spline : è una funzione curva che può essere del tipo

$$s(x_j) = \sum_{k=1}^{K+1} \beta_k h_k(x_j) \tag{3.12}$$

dove β_k sono i coefficienti della spline, $h_k(x)$ sono le hinge functions che descrivono i segmenti della spline e K è il grado della spline, ovvero il numero di hinge functions che la compongono. In generale, una spline può avere più di due punti di taglio, ovvero più di tre segmenti, e permette di includere dei termini di interazione tra le variabili del modello. Il gredo massimo delle splines determina il grado del modello.

Chiaramente il tipo di hinge function e di spline descritto rappresenta uno dei tanti possibili modi con cui è possibile esprimerle. Quelle presentate sono infatti delle forme molto semplici di funzione di base, ma ne esistono numerose altre volte ad esprimere delle relazioni più complesse.

L'algoritmo alla base del modello MARS è composto da due fasi:

1. Forward Step: a partire dal modello con sola intercetta, per ogni variabile esplicativa, si aggiungono ripetutamente delle funzioni di base trovando ad ogni passo quella che riduce maggiormente la radice dell'errore quadratico medio.
L'algoritmo valuta quindi tutte le possibili combinazioni di funzioni di base. Ogni nuova combinazione di funzioni di base consiste in un termine già presente nel modello moltiplicato per una nuova hinge function o spline. Per l'aggiunta di ciascuna funzione di base si devono testare tutte le combinazioni possibili dei termini che la vanno a formare. Questo processo continua finché la variazione dell'errore residuo non è troppo piccola per continuare o fino al raggiungimento di un numero massimo di termini. Quest'ultimo è impostato a priori. Si ripete il processo per tutte le variabili, cercando di aggiungere una nuova funzione di base per ciascuna variabile. Per calcolare il coefficiente di ciascuna funzione di base, si applica una regressione lineare sui termini già noti.
La ricerca delle combinazioni della Forward Step può essere accelerata tramite metodi euristici per la selezione dei termini esistenti.

2. Backward Step: per evitare l'overfitting, si seleziona un modello provando a rimuovere le funzioni di base meno efficaci. I modelli annidati vengono confrontati utilizzando un particolare criterio di selezione. Questo passo viene detto anche fase di Pruning del modello.

Il modello MARS risulta quindi un'evoluzione più flessibile del modello di regressione lineare che mantiene comunque un buon livello di interpretabilità e offre una selezione automatica delle variabili del modello. Le funzioni di base vengono usate per approssimare la relazione non lineare tra le variabili, tuttavia, il modello richiede una procedura di selezione delle variabili rilevanti e delle loro interazioni che può essere computazionalmente costosa in presenza di un grande numero di esplicative.

3.4 Regressione con Alberi

I modelli basati sugli alberi possono essere utilizzati sia per la classificazione che per la regressione (talvolta vengono indicati come CART). Fanno parte della classe dei modelli non-parametrici, ma sono rilevanti a tal punto da essere spesso trattati

come categoria a sé stante. Tutti questi modelli fanno riferimento alla struttura ad albero della teoria dei grafi. L'albero è un grafo non orientato, connesso e aciclico. In altri termini, dato un insieme di nodi ed archi di un grafo, questo risulta essere un albero se ciascun arco non presenta alcun verso e, se presi due nodi qualsiasi, questi sono connessi da uno ed un solo percorso.
Esistono modelli basati sugli alberi di diverse tipologie e complessità. Spesso, per molte applicazioni, molteplici alberi vengono messi in relazione e combinati al fine di formare modelli più sofisticati (metodi ensemble).

3.4.1 Albero Decisionale

L'albero decisionale di regressione, noto anche come albero di regressione, è il più semplice tra i modelli basati su una struttura ad albero.
La struttura sottostante è infatti quella di un grafo ad albero univoco in cui ogni nodo interno è etichettato con una tra le variabili esplicative a disposizione. Gli archi che si dipartono da un nodo indicano una selezione tra i possibili valori della variabile esplicativa, oppure conducono ad un nodo decisionale subordinato ad un'altra variabile. Ogni foglia dell'albero è associata ad un valore per la risposta. Un albero viene costruito dividendo l'insieme di partenza, che costituisce il nodo radice dell'albero, in sottoinsiemi che vanno a comporre i nodi figli successivi.
La suddivisione si basa su un insieme di regole di suddivisione. Questo processo viene ripetuto su ogni sottoinsieme di nodi ricorsivamente (partizionamento ricorsivo) fino a quando il sottoinsieme di un nodo ha tutti gli stessi valori della variabile risposta, quando la suddivisione non aggiunge più valore alle previsioni o quando si raggiungono le condizioni per applicare un criterio di stop pre-impostato. Questo processo è di tipo top-down in quanto parte dal nodo radice fino a giungere ai nodi foglia.
A livello di variabili, l'insieme dei valori delle esplicative $X_1, ..., X_p$, ovvero lo spazio dei predittori, viene diviso in K regioni distinte R_k non sovrapposte. Queste regioni a livello teorico possono avere una forma qualsiasi, ma frequentemente, per semplicità di costruzione ed interpretazione, sono rettangoli multiidimensionali. Alle osservazioni appartenenti ad una stessa regione R_k sarà associato uno stesso valore previsto, corrispondente alla media delle variabili risposta per le osservazioni in quella regione. Le regioni vengono scelte in modo tale da minimizzare ad ogni passo una determinata metrica (ad esempio l'RMSE per la regressione) in maniera ricorsiva come sopra descritto, la quale svolge il ruolo di funzione di costo. Poiché la logica applicata è quella top-down, la divisione delle regioni viene fatta in modo da preferire un albero migliore al passo immediatamente successivo, piuttosto che un albero migliore a livello globale. Quest'approccio viene definito greedy.
Per evitare l'overfitting, oltre al criterio di stop, si può svolgere il pruning (o potatura) del modello. Per questo fine si introduce il concetto di complessità e di profondità di un albero. La complessità di un albero decisionale è definita come il numero di suddivisioni dell'albero, mentre la profondità massima è il numero massimo di cammini che vanno dal nodo radice alle sue foglie. Gli alberi più semplici o meno profondi sono infatti da preferire. Sono facili da interpretare e hanno meno probabilità di adattarsi eccessivamente ai dati, ovvero di andare in overfitting.
Metodi di pruning semplici consistono nel limitare la complessità (o la profondità

massima) dell'albero, oppure, si può esaminare ogni nodo foglia dell'albero e valutare l'effetto che si avrebbe sulla sua rimozione utilizzando un set di test di attesa.
I nodi foglia vengono rimossi solo se viene ottenuta una riduzione della funzione di costo complessiva sull'intero set di test. Si smette di rimuovere i nodi quando non è possibile apportare ulteriori miglioramenti.
È possibile utilizzare metodi più avanzati, come il pruning della complessità dei costi (detta anche potatura del legame più debole), in cui viene utilizzato un parametro di apprendimento per valutare se i nodi possano essere rimossi in base alla dimensione dei sottoalberi.

3.4.2 Bagging e Random Forest

Il modello Random Forest è una tecnica di apprendimento supervisionato basata sull'aggregazione di un insieme di alberi di decisione. Il suo obiettivo è quello di migliorare la generalizzazione del modello tramite la riduzione della varianza delle previsioni e la limitazione del fenomeno dell'overfitting, controbilanciata solo in parte dal processo di pruning dell'albero.
Il Bagging (o aggregazione Bootstrap), una tecnica utilizzata dal Random Forest, prevede la creazione di B campioni di dati dal dataset di partenza, ciascuno estratto in modo aleatorio e senza ripetizione con la stessa dimensione n del dataset originale. Su ogni campione viene addestrato un albero di decisione e le previsioni ottenute da tutti gli alberi vengono aggregate attraverso una media, ottenendo così una stima più stabile.
Nel Random Forest, in aggiunta al Bagging, viene utilizzata la tecnica di Random Feature Selection. Ad ogni passo della costruzione dell'albero di decisione, un sottoinsieme di m predittori viene selezionato in modo casuale dal set completo dei predittori. Questa tecnica aiuta a ridurre la correlazione tra gli alberi, limitando il rischio di sovrapposizione tra di essi.
Il parametro B viene solitamente calcolato attraverso il metodo di cross-validation o tramite l'out-of-bag error (OOB). Il parametro m è scelto generalmente come la radice quadrata del numero totale di predittori del dataset di partenza.
Nonostante il modello Random Forest ha una minore interpretabilità rispetto all'albero decisionale, le sue capacità predittive sono decisamente maggiori, grazie alla riduzione della varianza delle previsioni e alla limitazione dell'overfitting. In generale, il Random Forest si dimostra particolarmente efficace in contesti di grandi dataset con molte variabili predittive.

3.4.3 Boosting

Il Boosting prende ispirazione dal Bagging, ma vengono apportate alcune modifiche. Anche in questa tecnica si seleziona B volte un campione aleatorio con ripetizione di osservazioni della stessa dimensione n del dataset di partenza. A ciascun campione si applica un albero di regressione. La grande differenza rispetto al Bagging, è che il Boosting applica gli alberi in maniera sequenziale in modo tale che ciascuno di essi si basi sui risultati di quello precedente. In questo senso i pesi campionari delle osservazioni sono aggiornati ad ogni passaggio al fine di migliorare le stime e di evitare l'indipendenza tra gli alberi.

AdaBoost

Il più semplice modello ad alberi basato sul Boosting è l'AdaBoost (o Adaptive Boosting), il quale prende il nome dal corrispettivo algoritmo.
L'AdaBoost applica sequenzialmente degli alberi con un solo nodo radice e due soli nodi foglia, detti stump e indicati con $s_b(x)$, $b = 1, ..., B$. Di conseguenza, ad ogni stump corrisponderà una sola variabile. Inizialmente il primo stump si basa sul dataset iniziale e i pesi campionari iniziali $w_i^{(b)}$ sono tutti pari a $\frac{1}{n}$, in altri termini $w_i^{(1)} = \frac{1}{n}$. Ogni stump successivo prende come input il campione di osservazioni con ripetizione proveniente dallo stump precedente. Il campione viene realizzato aggiornando i pesi in modo da favorire la presenza delle osservazioni peggio previste. Per fare questo si calcola prima l'errore di previsione assoluto per ogni osservazione come segue:

$$l_i^{(b)} = \mid y_i - \hat{y_i}^{(b)} \mid , \;\; i = 1, ..., n \tag{3.13}$$

con y_i valore osservato della risposta per l'i-esima osservazione e $\hat{y_i}^{(b)}$ valore previsto della variabile risposta per l'osservazione i-esima da parte del b-esimo stump.

Gli errori di previsione vengono poi normalizzati applicando una funzione di costo $L_i^{(b)}$ che vincola i valori nell'insieme del dominio $[0, 1]$. La funzione di costo può essere di tipo lineare, quadratico o esponenziale. Viene inoltre indicata con $\bar{L}^{(b)}$ la media delle funzioni di costo per il b-esimo stump.
L'errore totale dello stump è:

$$\epsilon_b = \sum_{i=1}^{n} w_i^{(b)} L_i^{(b)} \;\; , \;\; i = 1, ..., n \tag{3.14}$$

La variabile scelta per lo stump sarà quella che minimizzerà ϵ_b.
I pesi campionari successivi avranno la seguente forma:

$$w_i^{(b+1)} = w_i^b \beta_b^{1-L_i^{(b)}} \;\; , \;\; i = 1, ..., n \tag{3.15}$$

con $\beta_b = \frac{\bar{L}^{(b)}}{1-\bar{L}^{(b)}}$ coefficienti di confidenza dei predittori, i quali vengono associati ai pesi campionari b-esimi.

Le previsioni per la variabile risposta corrisponderanno alle corrispettive mediane pesate dei valori previsti da ciascuno stump. Le previsioni avranno quindi la forma:

$$\hat{y}_i = inf\left\{ y_i \in Y : \sum_{b: y_i^{(b)} \leq y_i} log\frac{1}{\beta_b} \geq \frac{1}{2} \sum_b log\frac{1}{\beta_b} \right\} \tag{3.16}$$

Utilizzando questo metodo solitamente si osserva un miglioramento consistente rispetto al Random Forest in termini di capacità previsiva.

Gradient Boosting

Un altro celebre modello che può essere implementato con gli alberi di regressione è il Gradient Boosting. Il nome di questo modello è preso dal metodo di discesa del

gradiente su cui è basato.
Per presentare più facilmente questo modello si prende in considerazione il caso di una sola variabile risposta e di una sola esplicativa. I risultati possono essere facilmente generalizzati al caso di variabili esplicative multiple.
Innanzitutto, per costruire il modello, si imposta una particolare funzione di costo:

$$L = L(y, F(x)) = \frac{1}{2}(y - F(x))^2 = \frac{1}{2}(y - \hat{y})^2 \tag{3.17}$$

E si inizializza il modello con un valore costante:

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma) \tag{3.18}$$

dove γ è il valore che minimizza la somma delle funzioni di costo per ogni osservazione.

Si può dimostrare che γ corrisponde alla media della variabile risposta. Di conseguenza, l'albero iniziale sarà costituito da una sola foglia.
Successivamente si applicano in maniera sequenziale B alberi con un numero di foglie fissato. Solitamente il numero di foglie massimo viene impostato pari ad un numero compreso tra 8 e 32.
Questi alberi sono costruiti sui cosiddetti pseudo-residui del modello, i quali, per il b-esimo albero, possono essere espressi in termini di funzione di costo come segue:

$$r_{ib} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{b-1}}, \quad i = 1, ..., n \tag{3.19}$$

Data la particolare funzione di costo, gli r_{ib} non sono altro che le differenze tra i valori osservati e quelli previsti. Il termine pseudo-residuo viene indicato per distinguere gli r_{ib} dai residui della regressione lineare. Se venisse, inoltre, utilizzata una funzione di costo differente da quella sopra descritta, gli pseudo-residui non corrisponderebbero più ai residui classici.
La forma con cui sono stati scritti gli pseudo-residui è basata su una sola variabile esplicativa. Si potrebbe generalizzare questo risultato al caso di una moltitudine di variabili esplicative sostituendo la derivata con il più generico gradiente da cui prende il nome l'algoritmo.
Ad ogni iterazione, gli alberi sono realizzati a partire dai nuovi dati del tipo $\{(x_i, r_{ib})\}_{i=1}^{n}$. Nel costruire gli alberi sugli pseudo-residui r_{ib}, si vanno a creare delle regioni terminali R_{jb}, ciascuna corrispondente alla j-esima foglia del b-esimo albero. Il numero totale di regioni terminali per il b-esimo albero sarà J_b.
Ad ogni regione terminale si avranno dei nuovi valori previsti per la variabile risposta:

$$\gamma_{jb} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{b-1}(x_i) + \gamma), \quad j = 1, ..., J_b \tag{3.20}$$

Infine, al termine di ogni iterazione b-esima dell'algoritmo, si aggiorna il modello corrente come segue:

$$F_b(x) = F_{b-1}(x) + \eta \sum_{j=1}^{J_b} \gamma_{jb} I(x \in R_{jb}) \tag{3.21}$$

con η che corrisponde al tasso di apprendimento (o learning rate) del modello che viene fissato come valore nell'intervallo (0,1].

Una versione alternativa dell'algoritmo consiste nella scelta di un valore previsto ottimale γ_b a livello dell'intero albero invece che a livello di regione terminale. Per questa configurazione dell'algoritmo i valori previsti per la risposta sono:

$$\gamma_b = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, F_{b-1}(x_i) + \gamma \sum_{j=1}^{J_b} \gamma_{jb} I(x \in R_{jb})) \,, \quad j = 1, ..., J_b \tag{3.22}$$

e il modello si aggiorna ad ogni b-esima iterazione come segue:

$$F_b(x) = F_{b-1}(x) + \eta\gamma_b \sum_{j=1}^{J_b} \gamma_{jb} I(x \in R_{jb}) \tag{3.23}$$

Per entrambe le configurazioni, le previsioni risultanti del Gradient Boosting corrisponderanno ai rispettivi valori di $F_B(x)$.
Questo modello può essere soggetto ad overfitting, per questo risulta fondamentale applicare delle tecniche di regolarizzazione. In linea di massima il Gradient Boosting risulta avere delle capacità previsive superiori all'AdaBoost ed è un modello assai diffuso per trattare problemi previsivi complessi.

XG-Boost

L'XG-Boost, o Extreme Gradient Boosting, è un algoritmo alla base di un modello che prende ispirazione dal Gradient Boosting e lo estende ad una forma più efficiente. La configurazione dell'XG-Boost che andiamo a presentare è quella più elementare e viene spesso definita come versione greedy.
Per questo modello la funzione di costo L e l'inizializzazione con $F_0(x)$ sono solitamente le stesse del Gradient Boosting classico. Ad ogni passo b-esimo, con $b = 1, ..., B$, si calcolano due elementi fondamentali alla base dell'algoritmo:

$$\hat{g}_b(x_i) = \left[\frac{\partial L(y_i, F(x_i)}{\partial F(x_i)}\right]_{F(x)=F_{b-1}} \,, \quad i = 1, ..., n \tag{3.24}$$

e

$$\hat{h}_b(x_i) = \left[\frac{\partial^2 L(y_i, F(x_i)}{\partial F(x_i)^2}\right]_{F(x)=F_{b-1}} \,, \quad i = 1, ..., n \tag{3.25}$$

dove $\hat{g}_b(x_i)$ è il gradiente della funzione di costo L calcolata al passo precedente, mentre $\hat{h}_b(x_i)$ è l'hessiana della funzione di costo calcolata anch'essa al passo precedente.

Per la funzione di costo nella forma dell'equazione (3.17), si ha che il gradiente corrisponde alla somma dei residui, mentre l'hessiana rappresenta il numero dei residui presenti.
Gli alberi di regressione sono così realizzati a partire dai nuovi dati $\{(x_i, -\hat{g}_b(x_i)/\hat{h}_b(x_i))\}$.
I valori previsti per la variabile risposta sono:

$$\hat{\phi}_b = \frac{1}{2} \underset{\phi \in \Phi}{\operatorname{argmin}} \sum_{i=1}^{n} \hat{h}_b(x_i) \left(-\frac{\hat{g}_b(x_i)}{\hat{h}_b(x_i)} - \phi(x_i) \right) \tag{3.26}$$

Il modello viene conseguentemente aggiornato come segue:

$$F_b(x) = F_{b-1}(x) + \eta\hat{\phi}_b \tag{3.27}$$

Le previsioni risultanti dell'algoritmo corrisponderanno a $F_B(x)$.
Come il Gradient Boosting classico, l'XG-Boost viene spesso regolarizzato tramite una metrica di tipo $L1$ o $L2$. Esistono anche delle tecniche di regolarizzazione più generiche basate ad esempio sul Moltiplicatore Lagrangiano.
Una problematica che si presenta con l'utilizzo dell'XG-Boost è la modalità con cui ciascun albero sceglie la divisione dei suoi nodi. Normalmente si andrebbe infatti a testare ogni possibile soglia di suddivisione (split) dell'albero in base ai valori osservati, ma questo richiederebbe dei costi computazionali estremamente elevati per grandi dataset. Le soglie di suddivisione sono allora spesso approssimate con i quantili pesati delle variabili esplicative. Ulteriori vantaggi dell'XG-Boost sono relativi a motivi di efficienza legati all'ottimo utilizzo delle risorse computazionali.

3.5 Selezione dei Modelli

La selezione del modello è una procedura che ha l'obiettivo di eleggere il miglior modello di analisi tra quelli disponibili secondo un determinato criterio.
Questo procedimento si articola essenzialmente in due parti:

- Selezione delle Variabili Esplicative: si selezionano le variabili esplicative che vengono utilizzate nel modello prescelto al fine di mantenere un buon bilanciamento tra adattamento ai dati osservati e capacità previsive.

- Tuning degli Iperparametri: si cercano gli iperparametri, tra un range di valori possibili, che permettono di ottenere un modello efficiente.

Spesso nella selezione di un particolare modello è presente soltanto uno dei due procedimenti appenza descritti. Ad esempio, per i modelli di regressione parametrica non sono presenti iperparametri, di conseguenza, non è necessario svolgere il tuning.
Per la selezione delle variabili nei modelli parametrici il criterio che prenderemo come riferimento è il criterio di informazione Bayesiana (BIC).
In tutti gli altri casi, si prenderà in considerazione l'errore quadratico medio (RMSE) che si presenta nella seguente forma:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{3.28}$$

dove y_i sono i valori osservati della variabile risposta e $\hat{y}_i$ sono i valori previsti dal modello per la risposta, entrambe per l'i-esima unità.

Per confrontare tutti i modelli selezionati, oltre all'RMSE si valuteranno anche altre due metriche fondamentali in modo tale da non favorire determinati modelli per il loro metodo di costruzione.
Le metriche a cui ci riferiremo sono l'errore medio assoluto (MAE) e il coefficiente

di determinazione R^2.
L'errore medio assoluto è:

$$MAE = \frac{\sum_{i=1}^{n} \mid y_i - \hat{y}_i \mid}{n} \tag{3.29}$$

Mentre il coefficiente di determinazione è:

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{3.30}$$

dove RSS è la devianza residua e TSS è la devianza totale.

Per stimare accuratamente queste metriche, si consiglia di utilizzare una procedura di convalida incrociata (cross-validation).
In particolare, nel nostro caso si applicherà la k-fold cross validation. Con questa tecnica il campione originale viene suddiviso aleatoriamente in k sottocampioni (fold) di uguali dimensioni. Dei k sottocampioni, un solo sottocampione viene mantenuto come set di validazione per testare il modello, mentre i restanti $k-1$ sottocampioni vengono utilizzati come dati di training.
Il processo viene quindi ripetuto k volte, dove ognuno dei k sottocampioni viene utilizzato esattamente una volta come set di validazione. I k risultati possono poi essere raggruppati (ad esempio calcolando la loro media) per produrre una singola stima. L'indicatore che ne risulterà sarà uno stimatore di cross-validation.
Per mantenere un buon rapporto tra distorsione e variabilità del modello si deciderà di impostare il numero k di sottocampioni pari a 10. Questo numero di sottocampioni risulta adeguato per dataset di modeste dimensioni (con più di 1000 osservazioni).
Il tuning degli iperparametri può essere realizzato seguendo varie possibili tecniche. La più tradizionale è la Ricerca a Griglia (Grid Search) che consiste nell'assegnazione di un insieme di valori per gli iperparametri che vogliamo ottimizzare provando tutte le possibili combinazioni e verificando quale mostra capacità previsive migliori.
Questa procedura può essere eseguita in parallelo alla cross-validation attraverso numerosi approcci. Il più semplice consiste nell'applicare la k-fold cross-validation con ciascun set di iperparametri valutando a sua volta, per ognuno di esso, una stima dell'errore di cross-validation (come la stima dell'RMSE cross-validation).
Gli iperparametri associati alla stima dell'errore di cross-validation saranno quelli poi impiegati nel modello finale.
Approcci simili si hanno per unire per unire il processo di selezione delle esplicative alla cross-validation. In generale, tutti i metodi che combinano la selezione delle variabili con il processo di costruzione del modello (compreso il tuning degli iperparametri) sono chiamati "*Metodi Wrapper*".

Capitolo 4

Modelli Previsivi Spaziali

In questo capitolo si vogliono presentare dei modelli previsivi spaziali volti ad analizzare in maniera diretta l'impatto degli effetti spaziali presenti nei dati.
A tal fine, per descrivere gli effetti spaziali nei dati esistono due diversi termini:

- Autocorrelazione Spaziale: spesso indicata come interazione spaziale o variazione spaziale su piccola scala, è la correlazione di una variabile con sé stessa per motivi spaziali. In altri termini, è una misura di similarità o dissimilarità tra le caratteristiche di unità spazialmente vicine. Valori positivi di autocorrelazione spaziale indicano che le unità aventi simili valori per determinate variabili tendono a raggrupparsi nello spazio, mentre in caso di autocorrelazione spaziale negativa, le unità tendono a essere circondate da unità vicine con valori diversi tra loro. A volte si fa riferimento all'autocorrelazione spaziale come dipendenza spaziale, mentre per alcune fonti l'autocorrelazione spaziale è una forma specifica di dipendenza spaziale, dove quest'ultima indica una relazione generica tra una variabile e lo spazio circostante.

- Eterogeneità Spaziale: anche conosciuta come struttura spaziale, non stazionarietà o variazione spaziale su larga scala, si riferisce a differenze in una regione spaziale nella media, nella varianza, nelle strutture di correlazione spaziale o di autocorrelazione spaziale.

Nel capitolo capitolo precedente abbiamo visto come gli effetti spaziali possono essere trattati dai modelli non spaziali con l'introduzione di variabili territoriali. In questo capitolo ci focalizzeremo sul trattamento dell'autocorrelazione spaziale, assumendo di trovarci in un contesto di omogeneità spaziale.
Infatti, solitamente l'eterogeneità spaziale può essere ridotta dalla presenza di numerose variabili esplicative come nel nostro contesto applicativo. Le stime dei parametri e le capacità previsive dei modelli spaziali che assumono omogeneità spaziale sono spesso comparabili a quelli che non la assumono.
Un ultimo motivo per giustificare questa scelta è legato al costo computazionale. Per gli immobili di una certa area, avendo a che fare spesso con un cospicuo numero di osservazioni, i modelli utilizzati devono essere quanto più semplici ed efficienti possibili. Si può infatti dimostrare che i modelli per l'eterogeneità spaziale risultano particolarmente onerosi.

L'analisi dei dati spaziali per trattare l'autocorrelazione spaziale può essere quindi condotta attraverso differenti tipologie di modelli (o processi). Un tipo di modello frequentemente utilizzato nella letteratura scientifica è il modello autoregressivo simultaneo (SAR). In un modello SAR, le relazioni tra i valori della variabile risposta in tutte le unità spaziali sono descritte simultaneamente e gli effetti spaziali sono considerati endogeni.
Un altro modello molto diffuso è il modello autoregressivo condizionale (CAR). Nei modelli CAR, la distribuzione di una variabile risposta (o dell'errore di regressione) per un'unità spaziale è specificata condizionando i valori dei suoi vicini e gli effetti spaziali dei vicini che vengono considerati esogeni.
Per ultimo, i dati spaziali possono essere analizzati utilizzando il modello della media mobile spaziale (SMA), che modella il processo di errore in posizioni vicine con una combinazione lineare di errori casuali, detti rumori bianchi, in modo simile a quello che avviene per i modelli a media mobile nelle serie temporali.
In genere le varie modalità di specificazione dei modelli offrono risultati simili. Data quindi la facilità di interpretazione e modellazione, in questo testo tratteremo tutti i modelli come SAR. Questi modelli sono anche i più utilizzati in ambito economico.

4.1 Analisi Esplorativa Spaziale

L'analisi esplorativa spaziale dei dati (ESDA) è un processo aggiuntivo fondamentale rispetto all'analisi esplorativa classica per l'analisi preliminare dei dati spaziali.
Un obiettivo specifico dell'ESDA è la visualizzazione e la sintesi dei dati dalla prospettiva spaziale, che aiuta a suggerire potenziali modelli da formulare e metodi statistici da applicare per l'inferenza.
La fase iniziale dell'ESDA spesso prevede la visualizzazione spaziale dei dati di interesse. Ciò consente di identificare dei pattern spaziali tra i dati e i potenziali modelli statistici spaziali da applicare. A seguito di questo risulta utile specificare una struttura di vicinanza tra i dati e una conseguente matrice dei pesi spaziali. Infine si possono applicare degli indicatori spaziali per quantificare gli effetti spaziali presenti nei dati.

4.1.1 Struttura di Vicinanza

Per condurre l'ESDA e applicare la regressione spaziale, è spesso necessario specificare una struttura di vicinanza per ogni unità spaziale che comprende le unità vicine su un'area di studio. Da qui in avanti con il termine unità spaziale intenderemo specificatamente un'unità statistica che può essere collocata in uno specifico punto dello spazio. Altri tipi di analisi spaziale fanno invece riferimento alle unità spaziali come unità areali o geostatistiche.
Una struttura di vicinanza può essere basata sulla contiguità spaziale o sulla distanza. La struttura basata sulla contiguità è costruita in base al fatto che due unità spaziali confinino o meno tra loro. Su una griglia regolare, la struttura di vicinanza è relativamente semplice da specificare. Ad esempio, la struttura di vicinanza di contiguità "a torre" specifica i vicini come unità spaziali con confini condivisi in orizzontale o verticale. La struttura di vicinanza di contiguità "a regina" specifica i vicini come unità spaziali con confini o vertici condivisi.

I dati spaziali si trovano però tipicamente su griglie irregolari. Quando si costruiscono strutture di vicinanza su griglie irregolari si utilizzano le stesse regole che valgono per le griglie regolari. In questi casi, con una struttura di vicinanza di contiguità a torre, i vicini di un'unità spaziale sono le unità con linee di confine condivise. In una struttura di vicinanza di contiguità a regina, i vicini di un'unità spaziale sono le unità con linee di confine o vertici condivisi. Queste definizioni non sono però del tutto pertinenti quando le unità sono di punto e non areali.
Per quanto riguarda le strutture di vicinanza basate sulla distanza, ne esistono essenzialmente di due tipi. Si può avere una struttura di vicinanza basata su un numero specifico di vicini che ha lo stesso principio del K-nearest neighbors.
In tal caso i vicini sono calcolati in base ad una specifica metrica. Altrimenti i vicini possono essere trovati in base ad una fissata distanza. A partire da un'unità spaziale, le unità che si trovano entro quella distanza saranno i vicini dell'unità spaziale di riferimento.

Dopo aver determinato la struttura di vicinanza, occorre quantificare la prossimità dei vicini trovati per ciascuna unità spaziale. Si introducono quindi delle matrici dei pesi spaziali. Una matrice dei pesi spaziali è composta dai valori dei pesi spaziali che mettono in relazione il valore di una determinata variabile per un'unità spaziale con quello osservato nelle unità spaziali vicine secondo una struttura di vicinanza prestabilita.
I pesi spaziali per ciascuna unità dipendono strettamente dal numero di vicini.
Il numero dei vicini non è infatti necessariamente lo stesso per tutte le unità, il che comporta pesi spaziali differenti. Un modo per determinare i pesi consiste nello standardizzare ogni riga di una matrice di pesi spaziali (corrispondente ai vicini di una determinata unità) dividendo ogni valore per la somma di quella riga. L'applicazione di una matrice di pesi spaziali standardizzata per riga dà come risultato l'attributo medio ponderato dei vicini, rendendo l'interpretazione più significativa e più facile da comprendere.
I vicini di un'unità spaziale possono essere ponderati diversamente. In primo luogo, si potrebbero ponderare i vicini in base alle distanze da un'unità spaziale e utilizzare l'inverso di questi valori per dare ai vicini più vicini un peso maggiore rispetto a quelli più lontani.
In questo caso i pesi spaziali sono:

$$w_{ik} = \begin{cases} \frac{1}{d_{ik}} & \text{se } d_{ik} < \delta \\ 0 & \text{altrimenti} \end{cases} \tag{4.1}$$

dove d_{ik} è la distanza tra l'unità spaziale i-esima e l'unità k-esima e δ è una distanza soglia prefissata.

Più in generale, si può utilizzare la potenza p-esima dell'inverso delle distanze:

$$w_{ik} = \begin{cases} \left(\frac{1}{d_{ik}}\right)^p & \text{se } d_{ik} < \delta \\ 0 & \text{altrimenti} \end{cases} \tag{4.2}$$

Esistono poi ulteriori metodi di ponderazione, ma sono specifiche di alcuni tipi di unità spaziali. Essendo noi interessati a unità spaziali di punto, non è necessario

presentare altri metodi.
Per selezionare la corretta matrice dei pesi spaziali, la teoria è molto limitata. Un metodo che si può utilizzare è il confronto delle matrici tramite una specifica metrica a seguito dell'applicazione di un modello spaziale.

4.1.2 Indicatori di Autocorrelazione Spaziale

In letteratura esistono numerosi indicatori volti a valutare la presenza di effetti spaziali nei dati. Come anticipato precedentemente, ci concentreremo sull'aspetto dell'autocorrelazione spaziale. Tali indicatori possono indicare una forma globale o locale di dipendenza spaziale. Gli indicatori globali provvedono a studiare pattern spaziali tra i dati nell'intera area di studio, invece gli indicatori locali si riferiscono a precise sottoregioni spaziali.

Indice di Moran

L'indice di Moran (o indice I di Moran) è una misura di intensità globale dell'autocorrelazione spaziale, ovvero una misura della somiglianza tra unità spaziali vicine. E' definito come segue:

$$I = \frac{n}{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(y_i - \bar{y})(y_j - \bar{y})}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.3}$$

dove w_{ij} sono i pesi spaziali tra l'unità i-esima e l'unità j-esima, y_i è il valore osservato dell'unità i-esima per la variabile di interesse e $\bar{y}$ è la media della variabile per tutte le unità.

Se si standardizzano i pesi spaziali per riga, l'indice di Moran si può esprimere nella seguente forma semplificata:

$$I = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(y_i - \bar{y})(y_j - \bar{y})}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.4}$$

Quando i valori della variabile sono più simili (o dissimili) tra unità spaziali vicine, l'indice di Moran tende ad avere un valore più estremo. Quando la relazione tra i vicini è più debole l'indice tende a 0.
Pertanto, questo indicatore è simile all'indice di correlazione di Pearson, il quale misura l'intensità di una relazione lineare tra due variabili. In particolare, corrisponde all'indice di Pearson nel caso in cui si voglia descrivere la relazione tra una variabile Y e la variabile esprimibile come prodotto matriciale WY, dove W è la matrice dei pesi spaziali.
Per analizzare graficamente l'autocorrelazione spaziale si può utilizzare un diagramma di Moran, il quale mette in relazione i valori assunti da Y con WY. Questo illustra la relazione tra i valori dell'attributo scelto in ogni località e il valore medio dello stesso attributo nelle località vicine. È istruttivo considerare ogni quadrante del grafico. Nel primo quadrante si trovano i casi di tipo "High-High" in cui sia il valore effettivo che il valore medio locale dell'attributo sono superiori al valore medio generale. Allo stesso modo, nel terzo quadrante si trovano i casi di tipo "Low-Low"

in cui sia il valore effettivo che il valore medio locale dell'attributo sono inferiori al valore medio generale. Questi casi confermano l'autocorrelazione spaziale positiva. I casi negli altri due quadranti (High-Low e Low-High) indicano un'autocorrelazione spaziale negativa. A seconda dei gruppi dominanti, ci sarà una tendenza generale all'autocorrelazione spaziale positiva, negativa o nulla. Con un'attenta analisi si può quindi identificare quali aree della mappa sono maggiormente responsabili dell'alta o bassa autocorrelazione spaziale osservata e quali, eventualmente, sono in contrasto con l'assunzione.
L'indice di Moran può essere quindi interpretato come la pendenza della retta di regressione nel diagramma di Moran della variabile sulla sua media ponderata con i rispettivi pesi spaziali.
In assenza di autocorrelazione spaziale, il valore atteso dell'indice di Moran è pari a:

$$E(I) = -\frac{1}{n-1} \tag{4.5}$$

Di conseguenza, maggiore è la dimensione del campione n, più il valore atteso dell'indice di Moran assumerà valori prossimi a 0.
Analogamente, la varianza dell'indice è:

$$Var(I) = \frac{nS_4 - S_3 S_5}{(n-1)(n-2)(n-3)S_0^2} - (E(I))^2 \tag{4.6}$$

con

- $S_0 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}$
- $S_1 = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (w_{ij} w_{ji})^2$
- $S_2 = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} w_{ij} + \sum_{j=1}^{n} w_{ji} \right)^2$
- $S_3 = \frac{n \sum_{i=1}^{n} (y_i - \bar{y})^4}{(\sum_{i=1}^{n} (y_i - \bar{y})^2)^2}$
- $S_4 = (n^2 - 3n + 3)S_1 - nS_2 + 3S_0^2$
- $S_5 = (n^2 - n)S_1 - 2nS_2 + 6S_0^2$

Per grandi campioni, i valori che può assumere questo indice ricadono nell'intervallo di valori reali $(-1, 1)$. Valori positivi dell'indice indicheranno un'autocorrelazione spaziale positiva, mentre valori negativi denoteranno un'autocorrelazione spaziale negativa. Per una migliore interpretazione, si preferisce studiare l'indice in base al valore assunto da $(I - E(I))$, ovvero in base alla differenza tra il valore osservato e il suo valore atteso.
Per valori elevati di n, l'indice di Moran si distribuisce normalmente. Per testare la significatività dell'autocorrelazione spaziale si può costruire una statistica test z basata sull'indice di Moran del tipo:

$$z = \frac{I - E(I)}{\sqrt{Var(I)}} \tag{4.7}$$

dove il valore atteso $E(I)$, sotto l'ipotesi nulla di assenza di autocorrelazione spaziale, può essere approssimato dall'equazione (4.5), mentre la rispettiva varianza dall'equazione (4.6).

Conseguentemente, sotto l'ipotesi nulla, la statistica test z si distribuisce come una normale standard. Il p-value per questa statistica può essere ottenuto tramite test di approssimazione normale basato sulla distribuzione asintotica dell'indice, test di permutazione o simulazione Monte Carlo.

Indice di Geary

L'indice di Geary (o indice C di Geary) è una misura globale di autocorrelazione spaziale alternativa all'indice di Moran e ha la seguente forma:

$$C = \frac{n-1}{2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(y_i - y_j)}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.8}$$

I valori che assume l'indice di Geary sono necessariamente positivi. Sotto l'ipotesi di assenza di autocorrelazione spaziale, il valore atteso dell'indice di Geary è pari a 1, ovvero:

$$E(C) = 1 \tag{4.9}$$

Valori prossimi a 0 indicano un'autocorrelazione spaziale positiva, mentre valori maggiori di 1 sono propri di dati spazialmente autocorrelati negativamente.
Utilizzando gli stessi termini presentati con l'indice di Moran, si può scrivere la varianza dell'indice di Geary come:

$$Var(C) = \frac{(n-1)(2S_1 + S_2) - S_0^2}{2(n+1)S_0^2} \tag{4.10}$$

Per campioni di dimensione elevata, si può ricavare la statistica test z e il corrispettivo p-value come avvenuto per l'indice di Moran.
Nel confrontare i due indici globali, l'indice di Geary è inversamente proporzionale all'indice di Moran, ma non in modo identico e non ha la stessa valenza. L'indice di Geary è infatti più sensibile all'autocorrelazione spaziale locale. In genere i due indici possono essere utilizzati per tutti i tipi di dati quantitativi. L'indice di Moran è più potente statisticamente dell'indice di Geary, ad esclusione dei dati binari.

Indicatore Locale di Associazione Spaziale

Come accennato precedentemente, l'Indicatore Locale di Associazione Spaziale (LISA) è un qualsiasi indicatore che indica un'associazione di tipo spaziale attorno ad una osservazione che è costruito a partire da un indicatore di associazione globale. L'utilizzo del LISA può portare al raggiungimento di molteplici obiettivi, tra cui:

- valutare le ipotesi di stazionarietà, come la varianza costante nello spazio;
- individuare regioni locali di non stazionarietà, come cluster spaziali o unità spaziali con valori eccessivamente estremi;
- identificare le distanze oltre le quali l'associazione spaziale è debole o nulla;

- identificare gli outliers o i diversi regimi spaziali.

Gli indicatori a cui faremo riferimento per l'analisi sono quelli che vanno a misurare l'autocorrelazione spaziale locale. Questi indicatori possono essere costruiti a partire dagli indici globali che abbiamo precedentemente presentato. Analizziamo più nel dettaglio questi casi.
L'indicatore LISA associato ad un indice di Moran locale con pesi spaziali standardizzati per riga si presenta nella forma seguente:

$$I_i = (y_i - \bar{y}) \sum_{j=1}^{n} w_{ij}(y_j - \bar{y}) \tag{4.11}$$

dove y_i è il valore osservato della variabile di interesse per l'unità spaziale i-esima, y_j è il valore osservato per la j-esima unità, w_{ij} è il peso spaziale in corrispondenza dell'unità i e j, e, infine, $\bar{y}$ è la media dei valori della variabile.

La versione del LISA come indice di Geary locale è invece:

$$C_i = \sum_{j=1}^{n} w_{ij}(y_i - y_j)^2 \tag{4.12}$$

E' da notare che, a differenza degli indicatori globali, gli indicatori LISA sono relativi a ciascuna unità spaziale. Le statistiche test dell'indicatore LISA in queste due varianti possono essere formulate in modo analogo a quanto visto per l'indice di Moran e l'indice di Geary globali.
Anche in questo caso la significatività della statistica LISA può essere basata sull'approssimazione analitica tramite distribuzione normale, ma non è molto affidabile nella pratica. Un approccio preferibile consiste nell'utilizzo di un test di permutazione o nell'utilizzo di una simulazione di tipo Monte Carlo.
L'interpretazione dei valori LISA sono sulla stessa linea di quelli degli indici globali e, come anticipato, i risultati possono essere clusterizzati. L'assegnazione ad uno dei 4 diversi cluster (High-High, Low-Low, High-Low o Low-High) avviene prefissando una determinata soglia di assegnazione in valore assoluto. In corrispondenza dei valori osservati più estremi si andranno ad assegnare le unità ai cluster che indicano un'autocorrelazione positiva. In caso contrario, si terrà conto del gruppo dominante.

4.2 Modello di Lag Spaziale

Il modello di lag spaziale (SLM) è il modello spaziale più celebre e utilizzato tra quelli presenti in letteratura. Questo modello mette in relazione la variabile risposta e un insieme di variabili esplicative attraverso la regressione come avviene nella regressione lineare standard. Inoltre, la variabile risposta è autoregressiva sulle variabili risposta ritardate spazialmente (ovvero con lag spaziale sulla risposta).
Il modello di lag spaziale è specificato come:

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij} + \rho \sum_{k=1}^{n} w_{ik} Y_k + \epsilon_i \,, \;\; i = 1, ..., n \tag{4.13}$$

con

- Y_i : variabile dipendente (o risposta) quantitativa continua per l' i-esima osservazione, con $i = 1, ..., n$;
- Y_k : variabile risposta per la k-esima osservazione;
- X_{ij} : variabile indipendente (o esplicativa) j-esima, con $j = 1, ..., p$, per l' i-esima osservazione, con $i = 1, ..., n$;
- β_0 : intercetta del modello, corrisponde al valore atteso di Y quando tutte le variabili esplicative sono nulle;
- β_j : coefficiente angolare (o coefficiente di regressione) per la variabile X_j, con $j = 1, ..., p$;
- ρ : parametro scalare di lag spaziale;
- w_{ik} : peso spaziale per l'i-esima osservazione in corrispondenza della k-esima osservazione vicina;
- ϵ_i : errore statistico per l' i-esima osservazione, $i = 1, ..., n$. Si assume che gli errori siano tra loro indipendenti e normalmente distribuiti.

In forma matriciale il modello si può esprimere, senza intercetta, come:

$$Y = X\beta + \rho WY + \epsilon \tag{4.14}$$

con

- Y : vettore n×1 della variabile risposta;
- X : matrice n×p delle variabili esplicative;
- β : vettore p×1 dei coefficienti di regressione;
- ρ : parametro scalare di errore spaziale;
- W : matrice n×n dei pesi spaziali;
- ϵ : vettore n×1 degli errori indipendenti e normalmente distribuiti.

Si denota che il termine autoregressivo WY indica una variabile con lag spaziale in quanto è una media ponderata delle variabili risposta vicine.

Si può osservare, inoltre, che se il parametro scalare di lag spaziale ρ è nullo, il modello di lag spaziale assume la stessa forma del modello di regressione lineare. I parametri possono essere stimati tramite stime di massima verosimiglianza a seguito della determinazione della matrice dei pesi spaziali W. Definire questa matrice significa identificare le osservazioni vicine imponendo una struttura di vicinanza ed assegnando dei pesi a ciascuna di queste osservazioni.
Le assunzioni di questo modello sono analoghe al modello di regressione lineare, ad esclusione dell'assunzione di uguale distribuzione degli errori. Per questo motivo si possono applicare i consueti metodi diagnostici della regressione lineare.
Ulteriori metodi diagnostici servono a rilevare l'eventuale dipendenza spaziale residua

dopo aver tenuto conto della dipendenza da lag spaziale attraverso il modello. L'eventuale dipendenza da lag spaziale residua può essere diagnosticata da un test LR, mentre l'eventuale dipendenza da errore spaziale residua può essere diagnosticata dai test LM e LM robusto. Tratteremo più approfonditamente questi aspetti nella sezione 4.6.

4.3 Modello di Lag Spaziale sulle X

Il modello di lag spaziale sulle X (SLX) possiede, come il modello di lag spaziale classico, un termine di lag che mette in relazione le osservazioni con quelle spazialmente vicine. Il lag è però presente sulle variabili esplicative e non sulla risposta. La sua specificazione è la seguente:

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij} + \sum_{j=1}^{p} \left(\theta_j \sum_{k=1}^{n} w_{ik} X_{kj} \right) + \epsilon_i \ , \ \ i = 1, ..., n \tag{4.15}$$

con

- Y_i : variabile risposta quantitativa continua per l' i-esima osservazione, con $i = 1, ..., n$;
- X_{ij} : variabile esplicativa j-esima, con $j = 1, ..., p$, per l' i-esima osservazione, con $i = 1, ..., n$;
- X_{kj} : variabile esplicativa k-esima, con $k = 1, ..., p$, per l' i-esima osservazione, con $i = 1, ..., n$;
- β_0 : intercetta del modello, corrisponde al valore atteso di Y quando tutte le variabili esplicative sono nulle;
- β_j : coefficiente di regressione per la variabile X_j, con $j = 1, ..., p$;
- θ_j : parametro di spillover spaziale per la j-esima variabile;
- w_{ik} : peso spaziale per l'i-esima osservazione in corrispondenza della k-esima osservazione vicina;
- ϵ_i : errore statistico per l' i-esima osservazione, $i = 1, ..., n$. Si assume che gli errori siano tra loro indipendenti e normalmente distribuiti.

Senza intercetta, si ha la seguente forma matriciale:

$$Y = X\beta + WX\theta + \epsilon \tag{4.16}$$

con

- Y : vettore n×1 della variabile risposta;
- X : matrice n×p delle variabili esplicative;
- β : vettore p×1 dei coefficienti di regressione;

- W : matrice n×n dei pesi spaziali;
- θ : vettore p×1 dei parametri di spillover spaziali;
- ϵ : vettore n×1 degli errori indipendenti e normalmente distribuiti.

La matrice WX rappresenta il termine autoregressivo e corrisponde alla matrice di disegno delle variabili esplicative con lag spaziale, ovvero sono medie ponderate delle rispettive variabili esplicative dei vicini.

Se il vettore dei parametri di spillover spaziali θ è un vettore nullo, il problema di regressione si riduce al caso della regressione lineare. Come nel caso del modello di lag spaziale, nel modello di lag spaziale sulle X i parametri possono essere stimati attraverso il metodo di massima verosimiglianza dopo la determinazione della matrice dei pesi spaziali W. Le assunzioni del modello e i metodi diagnostici sono analoghi a quelli del modello di lag spaziale.

4.4 Modello di Errore Spaziale

Assieme ai modelli di lag spaziale, il modello di errore spaziale (SEM) è il principale tipo di modello di regressione spaziale. L'assunzione fondamentale è che la correlazione spaziale tra le osservazioni sia dovuta a caratteristiche non osservate, che sono raggruppate spazialmente o seguono un modello spaziale, e che siano indipendenti dalle covariate incluse. Un modello di errore spaziale tiene conto della dipendenza spaziale attraverso un termine di errore normalmente distribuito e un termine di errore associato al lag spaziale.
Il modello di errore spaziale è specificato come:

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij} + \lambda \sum_{k=1}^{n} w_{ik} u_k + \epsilon_i \ , \ \ i = 1, ..., n \tag{4.17}$$

con

- Y_i : variabile risposta quantitativa continua per l' i-esima osservazione, con $i = 1, ..., n$;
- X_{ij} : variabile esplicativa j-esima, con $j = 1, ..., p$, per l' i-esima osservazione, con $i = 1, ..., n$;
- β_0 : intercetta del modello, corrisponde al valore atteso di Y quando tutte le variabili esplicative sono nulle;
- β_j : coefficiente di regressione per la variabile X_j, con $j = 1, ..., p$;
- λ : parametro scalare di errore spaziale;
- w_{ik} : peso spaziale per l'i-esima osservazione in corrispondenza della k-esima osservazione vicina;
- u_k : termine di errore per la k-esima osservazione;

- ϵ_i : errore statistico per l' i-esima osservazione, $i = 1, ..., n$. Si assume che gli errori siano tra loro indipendenti e normalmente distribuiti.

La corrispondente forma matriciale senza intercetta è:

$$Y = X\beta + \lambda W u + \epsilon \tag{4.18}$$

con

- Y : vettore n×1 della variabile risposta;
- X : matrice n×p delle variabili esplicative;
- β : vettore p×1 dei coefficienti di regressione;
- u : vettore n×1 degli errori;
- λ : parametro scalare di errore spaziale;
- W : matrice n×n dei pesi spaziali;
- ϵ : vettore n×1 degli errori indipendenti e normalmente distribuiti.

Il termine autoregressivo Wu risulta essere un termine di errore con lag spaziale perché è una media ponderata dei termini di errori dei vicini.

Se il parametro scalare di errore spaziale λ è nullo, il modello di lag spaziale assume la forma di un modello di regressione lineare. I metodi di stima dei parametri, le assunzioni del modello e i metodi diagnostici, sono analoghi ai precedenti modelli.

4.5 Modelli Spaziali Derivati

Dai tre modelli appena descritti è possibile derivarne degli altri più complessi.
I modelli restituiti risulteranno essere delle combinazioni dei precedenti. Ognuno di questi modelli sarà caratterizzato dalla presenza di almeno due parametri spaziali. Questi modelli vengono solitamente utilizzati per gestire situazioni di dipendenza spaziale non adeguatamente descrivibili dai modelli di base.
Il modello di Durbin spaziale (SDM) combina il termine autoregressivo del modello SLM con la specificazione di spillover spaziale delle esplicative e si può scrivere come:

$$Y = X\beta + \rho WY + WX\theta + \epsilon \tag{4.19}$$

Il modello autoregressivo spaziale combinato (SAC) comprende una variabile risposta autocorrelata e un termine di errore autocorrelato, il tutto esprimibile come:

$$Y = X\beta + \rho WY + \lambda W u + \epsilon \tag{4.20}$$

Il terzo modello aggregato è il modello di errore spaziale di Durbin (SDEM), il quale combina le caratteristiche del modello SEM con quelle del SLX:

$$Y = X\beta + WX\theta + \lambda W u + \epsilon \tag{4.21}$$

Infine, combinando tutti e tre i modelli di base, si può ottenere un modello ancora più generale e comprensivo che prende il nome di modello spaziale generale annidato (GNS). Talvolta questo modello è anche indicato come modello di Manski e si può formulare come segue:

$$Y = X\beta + \rho WY + WX\theta + \lambda Wu + \epsilon \tag{4.22}$$

Si nota che nei modelli trattati la matrice dei pesi spaziali W è uguale per tutti i termini dei modelli; una diversa specificazione di W è propria di modelli ancor più generali che non vengono qui presi in considerazione.
Riportiamo i modelli presentati nella tabella sottostante in base alla presenza o meno degli specifici parametri spaziali:

Parametri	$\rho = 0$	$\rho \neq 0$
$\theta = 0\,,\ \lambda = 0$	modello lineare	SLM
$\theta \neq 0\,,\ \lambda = 0$	SLX	SDM
$\theta = 0\,,\ \lambda \neq 0$	SEM	SAC
$\theta \neq 0\,,\ \lambda \neq 0$	SDEM	GNS

Tabella 4.1. Modelli di Regressione Spaziale

4.6 Selezione del Modello Spaziale

La procedura di selezione del modello spaziale classica differisce dalla selezione dei modelli che non prevedono effetti spaziali. Si potrebbe infatti selezionare il modello spaziale utilizzando i criteri di selezione del modello usuali che sono stati trattati nella sezione 3.5. Questo approccio ha però il difetto di dover necessariamente stimare tutti i modelli possibili, inclusi quelli che non verranno poi utilizzati. Si vuole invece applicare una procedura basata su un'analisi spaziale mirata che permetta di evitare la specificazione di modelli spaziali troppo onerosi che verrebbero scartati.
Si propongono così alcune statistiche test per i modelli spaziali e una procedura che applica questi test in modo situazionale in base ai dati osservati.

4.6.1 Statistiche Test

Per la selezione del modello spaziale esistono numerosi test che possono essere applicati. Questi test hanno l'obiettivo di indicare formalmente quale modello spaziale è più consono al trattamento dei dati.
Inizialmente queste verifiche si applicano ad un modello di base. Poiché il punto di partenza in comune tra tutti i modelli spaziali esaminati è il modello di regressione lineare standard con stime OLS, prenderemo tale modello come riferimento

nell'esposizione dei test. I risultati possono però essere generalizzati per confronti di modelli più complessi.
Volendo semplificare la forma delle statistiche test, utilizzeremo una notazione matriciale. I test che mostreremo sono volti a verificare l'ipotesi implicita di presenza di autocorrelazione spaziale o l'esplicita presenza di un determinato parametro spaziale. Per quest'ultimo caso sono disponibili tre diversi approcci: il Test di Wald basato sulla stima dei valori dei parametri, il Test del Rapporto di Verosimiglianza basato sulla differenza di adattamento del modello e il Test dei Moltiplicatori di Lagrange basato sulla pendenza della funzione di verosimiglianza.
Questi test sono asintoticamente equivalenti, mentre per campioni finiti si ha che $W \geq LR \geq LM$. In genere si tenderebbe quindi a preferire il Test del Rapporto di Verosimiglianza in quanto più equilibrato e potente statisticamente.
Il test LR richiede però la specificazione di entrambi i modelli che si vogliono verificare, per questo è più oneroso. Il Test di Wald e il Test dei Moltiplicatori di Lagrange richiedono invece la specificazione di un solo modello. In particolare il test W richiede di stimare il modello dell'ipotesi alternativa, ovvero quello più complesso, mentre il test LM necessità della sola stima del modello dell'ipotesi nulla, ovvero quello più semplice.
Essendo i modelli molto dispendiosi in termini di costo computazionale, si preferisce utilizzare il Test dei Moltiplicatori di Lagrange quando si devono eseguire test ripetuti e il Test del Rapporto di Verosimiglianza per un confronto tra modelli più rigoroso. Il Test di Wald non è generalmente preso in considerazione per il confronto tra modelli spaziali.

Test di Moran

Il test di Moran può essere utilizzato per verificare la presenza dell'autocorrelazione spaziale a partire dai residui di un modello.
L'indice di Moran è definito dall'equazione (4.3) o dall'equazione (4.4). Nel caso dell'applicazione del test di Moran sui residui del modello di regressione lineare standard con stime OLS, i momenti dell'indice di Moran sono diversi rispetto a quelli presentati nella sezione 4.1. Infatti, quelli esposti in quella sezione corrispondono al caso particolare in cui i momenti dell'indice di Moran sono calcolati sul modello nullo (con sola intercetta).
Si può dimostrare che generalmente il valore atteso dell'indice è:

$$E(I) = \frac{tr(MW)}{n-p} \tag{4.23}$$

e la sua varianza:

$$Var(I) = \frac{T}{(n-p)(n-p+2)} - (E(I))^2 \tag{4.24}$$

dove $tr(\cdot)$ indica la traccia di una matrice quadrata, $M = I - X(X'X)^{-1}X'$ è la matrice di proiezione idempotente, W è la matrice dei pesi spaziali, $T = tr(MWMW') + tr(MWMW) + (tr(MW))^2$ è una matrice composta dalle tracce di MW, n è il numero di osservazioni e p è il numero di parametri nel modello regressivo lineare.

Dai momenti dell'indice si può costruire una statistica test z che assume una distribuzione normale standard sotto l'ipotesi nulla di assenza di autocorrelazione spaziale. Nel caso di accettazione dell'ipotesi nulla, l'utilizzo di modelli spaziali a discapito di un più semplice modello regressivo lineare è sconsigliato. Il rifiuto dell'ipotesi nulla deve essere invece interpretato con cautela. Esso indica infatti la presenza di autocorrealazione spaziale residua, ma non indica quale approccio e/o modello sarebbe preferibile.
Il test di Moran serve quindi ad offrire un'indicazione generica sulla presenza o assenza di autocorrelazione spaziale nei dati a seguito dell'applicazione di un modello. E' necessario successivamente applicare ulteriori analisi al fine di comprendere la fonte di autocorrelazione spaziale residua.

Test dei Moltiplicatori di Lagrange

Il Test dei Moltiplicatori di Lagrange (o Test Score di Rao) verifica i parametri statistici in base al gradiente della funzione di verosimiglianza, noto come punteggio o score, valutato sul valore puntuale del parametro ipotizzato sotto l'ipotesi nulla. Se lo stimatore ristretto è sufficientemente vicino al massimo della funzione di verosimiglianza, il punteggio non dovrebbe differire da 0 più dell'errore di campionamento. Nella sua forma generica il test si presenta come:

$$LM = \frac{(U(\theta_0))^2}{I(\theta_0)} \tag{4.25}$$

dove $U(\theta_0) = \frac{\partial logL(\theta_0|x)}{\partial \theta_0}$ è il gradiente calcolato in θ_0 della funzione di log-verosimiglianza $logL(\theta_0|x)$ e $I(\theta_0) = -E\left(\frac{\partial^2}{\partial \theta_0} logf(X;\theta)|\theta\right)$ è l'informazione attesa di Fisher sul vettore dei parametri θ_0.

Volendo condurre dei test sui modelli spaziali di base denotiamo con LM_{lag} la statistica test dei Moltiplicatori di Lagrange per la determinazione del lag spaziale e LM_{err} quella per l'errore spaziale.
Si ha allora che:

$$LM_{lag} = \frac{(e'Wy/S^2)^2}{T_2} = \frac{d_\rho^2}{T_2} \tag{4.26}$$

e

$$LM_{err} = \frac{(e'We/S^2)^2}{T_1} = \frac{d_\lambda^2}{T_1} \tag{4.27}$$

con e vettore dei residui del modello di regressione lineare standard, W matrice dei pesi spaziali, y vettore dei valori osservati, $S^2 = \frac{e'e}{n}$ valor medio dei quadrati dei residui, $T_1 = tr((W + W')W)$, $T_2 = T_1 + (WX\beta)'M(WX\beta)/S^2$ e β vettore dei coefficienti di regressione del modello lineare standard.

Le statistiche test (4.26) e (4.27), sotto l'ipotesi nulla di assenza, rispettivamente,

di lag spaziale ($\rho = 0$) e di errore spaziale ($\lambda = 0$), si distribuiscono secondo una distribuzione χ^2 (chi-quadrato) con un solo grado di libertà. Si rifiuta l'ipotesi nulla se viene osservato un valore della statistica test più estremo di quello che si sarebbe osservato in assenza di lag (o errore) spaziale. Il p-value può essere trovato conseguentemente.
La limitazione principale del LM_{lag} e LM_{err} consiste nel fatto che sono entrambi potenti statisticamente l'uno rispetto all'altro influenzandosi quindi a vicenda.
Per questo tali test possono essere scritti nella loro versione robusta come:

$$LM^*_{lag} = \frac{(d_\rho - d_\lambda)^2}{T_2 - T_1} \tag{4.28}$$

e

$$LM^*_{err} = \frac{(d_\lambda - T_1 T_2^{-1} d_\rho)^2}{T_1(1 - T_1 T_2)} \tag{4.29}$$

Anche in questo caso, sotto ipotesi di assenza dei relativi parametri spaziali, le statistiche test si distribuiscono come una χ^2_1.

Si nota infine che in letteratura non è stata presentata una forma esplicita per svolgere il Test dei Moltiplicatori di Lagrange per il lag spaziale sulle X. In quel caso è preferibile applicare un apposito Test di Rapporto di Verosimiglianza.

Test del Rapporto di Verosimiglianza

Il Test del Rapporto di Verosimiglianza (o Test di Wilks) valuta la bontà di adattamento di due modelli statistici messi in relazione tra loro in base al rapporto delle loro verosimiglianze. In particolare, un valore di verosimiglianza è trovato attraverso la massimizzazione sull'intero spazio dei parametri e l'altro dopo aver impostato un determinato vincolo proveniente dall'ipotesi nulla.
Il test è il seguente:

$$LR = -2\left(\frac{L(\theta_0)}{L(\hat{\theta})}\right) \tag{4.30}$$

dove $L(\theta_0)$ è la verosomiglianza calcolata sul parametro di ipotesi nulla θ_0 e $L(\hat{\theta})$ è la verosimiglianza calcolata sul parametro stimato $\hat{\theta}$.

Se l'ipotesi nulla è supportata dai dati osservati, le due verosimiglianze non dovrebbero differire maggiormente rispetto all'errore di campionamento. Pertanto, il Test del Rapporto di Verosimiglianza verifica se questo rapporto è significativamente diverso da 1 o, equivalentemente, se il suo logaritmo naturale è significativamente diverso da 0. Sotto l'ipotesi nulla, il test si distribuisce come una χ^2 con un numero di gradi di libertà pari alla differenza tra il numero di parametri del modello alternativo e il numero di parametri del modello nullo.

4.6.2 Procedura di Selezione

Per selezionare il corretto modello spaziale sono stati proposti diversi approcci.
Le strategie principali si catalogano in Forward Stepwise e Backward Stepwise. Nella Forward Stepwise il processo di selezione va dal modello più semplice a quello più complesso. Al contrario, la Backward Stepwise procede dal modello più complesso a quello più semplice. Poiché i modelli spaziali sono spesso molto onerosi da stimare, si preferisce l'utilizzo di un approccio Forward.

La Forward Stepwise si può articolare in due fasi successive:

1. La prima fase ha lo scopo di selezionare il modello spaziale di partenza tra iquelli di base di tipo SAR. Si parte dalla stima del modello regressivo lineare con stime OLS e si procede progressivamente. Viene svolto un test di Moran sui residui del modello OLS per verificare la presenza o meno di autocorrelazione spaziale residua. In caso di accettazione dell'ipotesi di assenza di autocorrelazione si seleziona il modello regressivo lineare. In caso contrario, si svolgono i test dei Moltiplicatori di Lagrange sia per il lag spaziale che per l'errore spaziale.
Se viene rifiutata l'ipotesi nulla per solo uno dei test, si seleziona il rispettivo modello. Se in entrambi i test vengono accettate le ipotesi nulle, si stima un modello di tipo SLX e si svolge un test di Rapporto di Verosimiglianza per verificare l'utilità di tale modello. Nel caso in cui nei test LM si rifiutano entrambe le ipotesi nulle, si svolgono i test LM robusti per verificare se il modello SLM e SEM risultano entrambi plausibili. Se un solo modello tra i due non viene scartato, si seleziona quel determinato modello. Se entrambi non vengono scartati, si seleziona il modello più significativo secondo i test oppure si applica un modello SAC che tenga conto sia del lag spaziale che dell'errore spaziale. In ultimo, se vengono accettate entrambe le ipotesi nulle, il modello da preferire non è chiaro. In questo caso si valuta la possibilità di selezionare il modello più significativo con i test LM non robusti, cambiare procedura di selezione o di utilizzare dei modelli spaziali non SAR.
La procedura di prima fase è rappresentata nella figura 4.1.

2. La seconda fase è opzionale e ha l'obiettivo di valutare se un modello spaziale più complesso può risultare più adeguato rispetto a quello selezionato in prima fase. Questa fase può essere svolta soltanto se non è stato selezionato il modello di regressivo lineare in prima fase. In modo analogo a quanto visto in prima fase, si svolge un test di Moran sui residui del modello corrente per verificare la presenza di autocorrelazione spaziale residua. Se viene indicata l'assenza di autocorrelazione spaziale, il modello corrente viene selezionato e sarà quello che verrà impiegato nell'analisi. Altrimenti, si dovranno svolgere test addizionali.
Ad esempio, si possono svolgere i Test dei Moltiplicatori di Lagrange sui residui del modello corrente per valutare l'efficacia di un modello con più parametri spaziali. Il vantaggio consisterebbe nel non dover stimare ulteriori modelli. Nella maggior parte dei software statistici non sono presenti delle implementazioni dei test LM per tali modelli. Per questo motivo si preferisce spesso svolgere dei Test di Rapporto di Verosimiglianza tra il modello corrente e uno più complesso. Il processo di selezione termina quando non è più presente autocorrelazione spaziale residua.

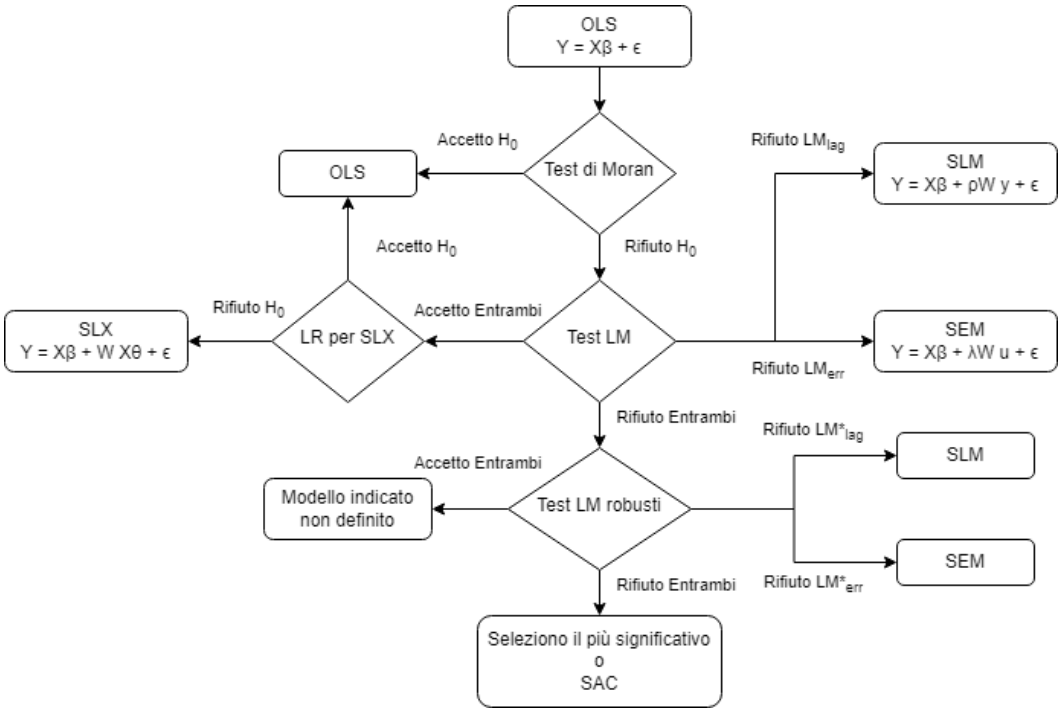


Figura 4.1. Forward Stepwise di Prima Fase

Capitolo 5

Applicazione

5.1 Dataset

Il dataset in questione è costituito da 21742 osservazioni di immobili di Madrid sulla base di 57 variabili descrittive originarie. I dati sono stati collezionati tramite la tecnica del Web Scraping esaminando gli annunci web provenienti dai più noti portali immobiliari spagnoli. Il periodo di riferimento degli annunci è il mese di marzo dell'anno 2020. E' possibile, inoltre, recuperare il dataset di partenza scaricando l'apposito file in formato CSV dalla piattaforma Kaggle al sito:

https://www.kaggle.com/datasets/mirbektoktogaraev/madrid-real-estate-market

L'obiettivo dell'analisi è quello di costruire dei modelli statistici capaci di prevedere accuratamente il prezzo di vendita degli immobili usufruendo dell'informazione contenuta nelle caratteristiche interne ed esterne alla casa. Al fine del raggiungimento di tale scopo, si proporranno dei modelli parametrici e non-parametrici per l'apprendimento supervisionato e si metteranno a confronto tra loro attraverso l'utilizzo di indicatori di accuratezza appositi. Distingueremo, inoltre, tra modelli spaziali e non seguendo le indicazioni presentate a livello teorico nei capitoli precedenti.
In questo contesto risulta di particolare importanza l'ottenimento di un dataset flessibile per l'applicazione dei diversi modelli e costituito da dati fedeli alla controparte reale. I dati raccolti risultano infatti spesso incompleti o ridondanti, per questo motivo diviene necessaria un'attenta analisi preliminare volta a rendere il dataset completo e facile da gestire.
Al fine di ottenere un'analisi che permetta di sfruttare al meglio l'utilizzo delle variabili spaziali e dei conseguenti modelli, prenderemo in considerazione soltanto i dati relativi ai 6287 immobili che possiedono un indirizzo di locazione preciso.

Descrizione delle Variabili

Dall'analisi escludiamo dapprima le variabili identificative generiche dell'immobile e le variabili che presentano soli valori mancanti. A seguito di questa eliminazione preliminare rimangono a disposizione 43 variabili.
Le variabili sono state catalogate con un sistema che prevede la differenziazione in base alla loro natura quantitativa o qualitativa e alle modalità che esse assumono.

Tenendo conto di questo, le descriviamo individualmente nel seguito:

VARIABILI QUANTITATIVE:

Continue:

- Prezzo di Vendita: prezzo con cui l'immobile è stato messo in vendita sul portale immobiliare. Essendo la variabile di interesse, avrà il ruolo di variabile dipendente nei modelli di regressione.
- Prezzo di Vendita per Area: prezzo di vendita dell'immobile al metro quadro.
- Prezzo di Affitto: canone di locazione consigliato dai proprietari dell'immobile o dall'agenzia immobiliare di riferimento.
- Prezzo del Parcheggio: prezzo del parcheggio se non incluso nel prezzo di vendita. L'inclusione del parcheggio nel prezzo di vendita è determinata dalla variabile binaria apposita.
- Superficie Costruita: numero di metri quadri della casa. Comprende lo spazio interno ed esterno all'abitazione, compresi i muri interni e condivisi. I muri condivisi, i balconi, i terrazzi, i parcheggi e, in generale, tutti gli spazi esterni ai muri perimetrali vengono conteggiati in maniera ridotta. La superficie costruita corrisponde nel sistema di misurazione italiano alla superficie commerciale.
- Superficie Utile: numero di metri quadri della casa. Comprende lo spazio interno al perimetro abitativo, esclusi i muri interni ed esterni. Gli spazi che fanno parte della superficie utile sono conteggiati rispettando le dimensioni reali. La superficie utile è sempre uguale o inferiore alla superficie costruita e corrisponde nel sistema di misurazione italiano alla superficie calpestabile.
- Superficie Coltivabile: numero di metri quadri della casa. Comprende lo spazio esterno al perimetro abitativo che può essere adibito alla coltivazione.

Discrete:

- Anno di Costruzione: anno in cui l'abitazione è stata costruita. Nel caso di appartamenti situati in condominii, si fa riferimento all'anno di costruzione dell'edificio. Non si tiene conto di eventuali ristrutturazioni.
- Numero di Bagni
- Numero di Piani: numero di piani di cui è composta l'abitazione. Nel caso in cui sia presente un piano terra, anch'esso viene conteggiato.
- Numero di Stanze: numero delle stanze abitabili che compongono l'immobile. Nel conteggio sono comprese le camere da letto, i soggiorni, le sale da pranzo e le cucine. D'altra parte, vengono esclusi tutti gli altri tipi di ambiente come i bagni, gli studi, le mansarde, i ripostigli, le lavanderie e tutti gli spazi esterni alle mura perimetrali. Si segnala che, con questo sistema di misura, per gli appartamenti studio e per gli attici ad un unico ambiente, il numero di stanze conteggiato risulta pari a 0.

VARIABILI QUALITATIVE:

Ordinali:

- Certificazione Energetica: punteggio da A a G per misurare l'efficienza energetica dell'immobile. Risulta possibile che la determinazione della certificazione energetica sia in corso al momento dell'inserimento dell'annuncio nel portale immobiliare o che l'immobile ne sia esente.
- Piano d'Ingresso: numero del piano d'accesso principale all'abitazione. L'entrata può essere collocata anche sul mezzanino o su piani interrati o semiinterrati.

Nominali Multicategoriche:

- Tipo di Casa: tipologia dell'abitazione. Si distingue in appartamento, casa indipendente/villa, casa semi-indipendente/bifamiliare e attico.
- Variabili di Posizione: segnalano la collocazione geografica della casa
  - Titolo: variabile composita contenente il tipo di abitazione e la locazione dell'immobile più specifica conosciuta; quest'ultima può comprendere il nome del quartiere, l'indirizzo o, alternativamente, una denominazione equivalente.
  - Sottotitolo: nome del quartiere e della città.
  - Indirizzo Completo: nome della via ed eventuale numero civico.
  - Indirizzo: nome della via senza numero civico.
  - Numero Civico
  - Identificativo della Zona: variabile composita contenente il nome del quartiere con il corrispettivo numero identificativo, il prezzo medio al metro quadro degli immobili nel quartiere di riferimento, il nome del distretto e il corrispettivo numero identificativo.

Nominali Binarie:

- Variabili Caratterizzanti dell'Immobile:
  possono essere definite come variabili del tipo:

$$X_{ij} = \begin{cases} 1 & \text{se l'immobile } i \text{ è definito dalla caratteristica } j \\ 0 & \text{altrimenti} \end{cases}$$

  - Accessibilità: indica se l'abitazione sia accessibile al momento dell'inserimento dell'annuncio nel portale immobiliare.
  - Esterno: indica se l'abitazione sia collocata in uno spazio esterno.
  - Indirizzo Nascosto: indica se l'indirizzo abitativo viene tenuto, anche solo parzialmente, nascosto. L'assenza della via o del numero civico nell'annuncio comporta a rendere l'indirizzo nascosto.
  - Necessaria Ristrutturazione: indica se è necessario eseguire una ristrutturazione al fine di rendere la casa agibile.

  - Nuova Costruzione: indica se l'edificio risulta essere una nuova costruzione. In tal caso, si può associare alla casa l'anno di costruzione risalente al periodo di riferimento dell'annuncio.
  - Parcheggio Incluso nel Prezzo: indica se il prezzo del parcheggio sia incluso nel prezzo di vendita della casa. Se il parcheggio non è incluso nel prezzo viene indicato il prezzo del parcheggio separatamente dal prezzo di vendita tramite l'apposita variabile.
  - Piano Interrato: indica se il piano d'ingresso all'abitazione è interrato o semi-interrato.
  - Variabili di Orientamento:
    segnalano l'orientamento delle facciate dell'immobile
    * Orientamento Nord
    * Orientamento Sud
    * Orientamento Est
    * Orientamento Ovest

- Variabili di Presenza/Assenza:
  sono definibili come:

$$X_{ij} = \begin{cases} 1 & \text{se l'immobile } i \text{ presenta l'attributo } j \\ 0 & \text{altrimenti} \end{cases}$$

  - Aria Condizionata
  - Armadio a Muro
  - Ascensore
  - Balcone
  - Giardino
  - Parcheggio
  - Piscina
  - Ripostiglio
  - Riscaldamento Autonomo
  - Riscaldamento Centralizzato
  - Terrazzo
  - Zona Verde

5.2 Pre-Processing

Il pre-processing dei dati è il processo preliminare dei dati costituito da una serie di operazioni che precedono l'analisi. E' costituita da fasi di manipolazione e pulizia dei dati al fine di garantire un elevato livello di prestazioni dei modelli. Durante questo processo si aggiustano gli errori nella struttura del dataset e nei dati stessi (dati anomali, contaminati, inconsistenti, invalidi e duplicati). Si vanno quindi a gestire adeguatamente le variabili per fare in modo che abbiano validità logica e che siano correttamente codificate. Infine, si svolge il trattamento dei dati mancanti.

5.2.1 Gestione delle Variabili

Poiché si è interessati a prevedere il prezzo di vendita degli immobili, non si tiene conto del prezzo di vendita per area e del prezzo di affitto. Avendo inoltre a disposizione l'eventuale prezzo del parcheggio, se presente, e il relativo indicatore di inclusione del prezzo del parcheggio in quello di vendita, si decide di includere nel prezzo di vendita anche il prezzo del parcheggio. A tal fine, se il parcheggio di un'abitazione non è incluso nel prezzo di vendita, verrà segnalato dall'apposita variabile indicatrice e si andranno a sommare i due prezzi che raffigureranno nel solo prezzo di vendita. La variabile sul prezzo del parcheggio e il corrispettivo indicatore vengono infine esclusi dal dataset in quanto ridondanti.
Si vogliono poi trasformare le variabili qualitative ordinali e nominali multicategoriche in modo da avere una struttura più semplice e più facile da trattare nei modelli. Per ricodificare queste variabili sono state applicate delle analisi esplorative specifiche che sono state inserite in appendice.
Abbiamo reso la variabile relativa alle certificazione energetica binaria. In particolare, le classi energetiche A, B e C sono considerate come classe alta, mentre le classi da D a G, le classi in fase di determinazione e l'esenzione dalla classe energetica sono state etichettate come classe bassa. In particolare, la variabile indicherà il valore 1 se la certificazione energetica è bassa e 0 altrimenti.
La variabile ordinale piano d'ingresso è stata trasformata in quantitativa discreta. A ciascuna categoria è stato associato il numero del piano corrispondente. Un preciso valore è stato associato anche ai piani d'ingresso speciali: il mezzanino corrisponde ad un valore pari a 0,5 in quanto è il piano collocato tra il primo piano e il piano terra, quest'ultimo associato al valore 0, il seminterrato è rappresentato invece dal valore -0,5 e il piano interrato da -1.
Per la variabile che indica il tipo di immobile, è stata integrata l'informazione contenuta nella variabile titolo. Si sono ottenute inizialmente 4 diverse categorie. Più specificatamente si è distinto tra appartamenti, case indipendenti, semi-indipendenti e attici. Appartamenti e case semi-indipendenti sono stati poi uniti in quanto non si sono osservate differenze significative sul prezzo di vendita.
Le variabili di posizione vengono impiegate per estrarre delle informazioni spaziali che saranno utilizzate successivamente. Dall'identificativo della zona viene estratto il numero identificativo del distretto che verrà utilizzato in un particolare tipo di analisi. Prendendo l'indirizzo abitativo si andranno invece a ricercare le corrispondenti coordinate geografiche. Vedremo più dettagliatamente tale procedimento. Eliminiamo infine dal dataset queste variabili.
Senza perdita di specificità, uniamo la variabile indicatrice per il balcone con quella per il terrazzo e la variabile per il giardino con quella per la zona verde. Negli annunci immobiliari i termini che si possono trovare per gli spazi verdi o esterni sono infatti spesso interscambiabili. La variabile binaria che fa riferimento alla presenza o assenza del riscaldamento autonomo ha significato esattamente opposto alla variabile che indica la presenza o l'assenza del riscaldamento centralizzato. Per evitare ridondanza si esclude quest'ultima dal dataset. Anche la variabile che indica se il piano d'ingresso è sotto il pianterreno risulta ridondante per la presenza della variabile piano d'ingresso.
In questa fase sono state così eliminate 15 variabili e ne rimangono a disposizione 28.

Variabili di Distanza

Le variabili di distanza indicano la lontananza degli immobili rispetto ai luoghi d'interesse della città. Queste informazioni non sono incluse nell'insieme di dati originario. Abbiamo introdotto queste variabili al fine di fornire delle informazioni aggiuntive ed esterne a ciascuna abitazione che siano caratteristiche del luogo in cui è collocato l'immobile. Questo approccio è in linea con la volontà di utilizzare dei metodi spaziali per prevedere il prezzo degli immobili. Queste informazioni esterne al dataset contribuiscono a rendere le previsioni più accurate.
Per calcolare le distanze è stato necessario utilizzare degli strumenti geografici. In particolare, si è utilizzato il software ArcGIS e OpenStreetMap. ArcGIS è un sistema informativo geografico prodotto dall'azienda Esri. Viene usato per la creazione di mappe, la loro analisi e, più in generale, l'uso statico e interattivo di tali mappe; altri utilizzi consistono nella compilazione di dati geografici, la condivisione di informazioni geospaziali e la loro gestione attraverso database. OpenStreetMap (OSM) è invece un progetto collaborativo finalizzato alla creazione di mappe del mondo a contenuto libero.

Il processo che prevede il calcolo delle distanze è articolato in tre fasi:

1. Geolocalizzazione degli Immobili: gli indirizzi delle abitazioni vengono geocodificati in punti sulla mappa. Ciascun punto sarà costituito dalle due coordinate geografiche, ovvero dalla longitudine e dalla latitudine. Per la geocodifica si utilizza l'ArcGIS World Geocoding Service offerto dall'omonima azienda.

2. Geolocalizzazione dei Luoghi d'Interesse: si estraggono le coordinate geografiche dei luoghi d'interesse accedendo al database aperto di OpenStreetMap. Le informazioni sono ricavate specificando degli appositi tag associati agli elementi di una città. Ogni tag è composto da una chiave che indica la categoria di appartenenza delle strutture e da un valore che specifica la sottocategoria.

3. Calcolo delle Distanze: si calcolano le distanze in linea d'aria tra tutti gli immobili e tutti i luoghi d'interesse. Successivamente si estraggono le distanze minime tra ciascun immobile e ogni tipologia di luogo d'interesse. La metrica di riferimento per il calcolo è la distanza di Haversine in metri che misura la lontananza tra due punti in una sfera ed è definita come:

$$d_{HAV} = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos\varphi_1 \cos\varphi_2 \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \tag{5.1}$$

dove r è il raggio della sfera, φ_1 e φ_2 sono rispettivamente la latitudine del primo e del secondo punto, λ_1 e λ_2 sono rispettivamente la longitudine del primo e del secondo punto.

Un metodo di misurazione più preciso si baserebbe sulla formula di Vincenty per il calcolo delle distanze tra punti di uno sferoide oblato. Si è deciso di non utilizzare la distanza di Vincenty in quanto le misurazioni risulterebbero leggermente più accurate a fronte di un costo computazionale decisamente maggiore. Le differenze

tra le distanze di Haversine e Vincenty per la Spagna sono generalmente nell'ordine dei centimetri (fino ad un massimo di 1 metro) per grandezze fino al chilometro. Per il nostro tipo di analisi queste differenze sono trascurabili.

I luoghi di interesse che sono stati scelti hanno lo scopo di descrivere i punti focali di una città, le comodità e le strutture che aiutano a rendere una zona ambita o da evitare. La vicinanza con i luoghi d'interesse comporterà a modificare il prezzo di un'immobile. Infatti, se un immobile è situato in una zona della città con molte comodità, al netto delle altre caratteristiche, il prezzo di vendita salirà, in caso contrario, se un immobile non è situato in una zona attrattiva o è vicino a edifici sgradevoli, il suo prezzo diminuirà.

Abbiamo raggruppato i luoghi di interesse in 6 macrocategorie:

- Prima Necessità / Sanità: sono le strutture di base per il sostentamento e la salute dell'individuo
  - Supermercato
  - Ospedale
  - Farmacia
- Finanza: sono gli edifici dove avvengono le transazioni economiche dell'individuo che comprendono in genere il ritiro e il deposito di denaro
  - Banca
  - Ufficio Postale
- Educazione: sono gli edifici legati all'istruzione scolastica
  - Università
  - Scuola dell'Obbligo
  - Scuola dell'Infanzia
- Trasporti: sono i luoghi dove avvengono i principali spostamenti con mezzi di trasporto di una città
  - Stazione dei Treni
  - Stazione dei Bus
  - Aeroporto
- Intrattenimento: sono i luoghi dove è possibile svolgere attività ludico ricreative
  - Palestra
  - Parco
  - Stadio
  - Discoteca
  - Cinema

  - Biblioteca

- Turismo: sono i posti che hanno valenza turistica e culturale
  - Edificio Storico
  - Attrazione Turistica

Una descrizione esaustiva dei luoghi d'interesse che sono stati presi in considerazione si può trovare sul portale web di OpenStreetMap che indicheremo in bibliografia.

5.2.2 Errori nei Dati

In fase di analisi esplorativa preliminare si è potuto osservare che alcune variabili binarie presentano una sola modalità osservata. Altrimenti, per quella variabile il dato risulta in alternativa mancante. Il motivo di questo avvenimento può essere legato ad un'errata codificazione delle variabili in questione.
Analizzando i valori osservati per quelle variabili e le relative percentuali di dati mancanti sul totale delle osservazioni, emerge che la mancanza dei dati possa essere dovuta a motivi deterministici. Infatti, quando i valori sono presenti, questi indicano sempre la presenza di un particolare attributo dell'immobile. E' ipotizzabile allora che la mancanza del dato corrisponda all'assenza di quello specifico attributo.
A supporto di questa tesi vi è la struttura stessa degli annunci immobiliari da cui sono state ricavate le informazioni sugli immobili. Non inserire una caratteristica importante della casa è infatti sinonimo solitamente di mancanza della stessa. Le percentuali di dati mancanti sul totale delle osservazioni sono oltretutto plausibili con l'ipotesi di assenza degli attributi.
Le variabili a cui abbiamo fatto riferimento sono:

- Aria Condizionata
- Armadio a Muro
- Balcone e Terrazzo
- Giardino e Zona Verde
- Piscina
- Ripostiglio

Decidiamo quindi di sostituire i valori mancanti con l'indicazione di assenza degli attributi. La sostituzione viene fatta a livello dell'intero dataset senza distinguere tra set di training e set di test. Il motivo di questo è supportato dal fatto che il procedimento impiegato per queste variabili è di tipo deterministico. Di conseguenza, l'imputazione o l'eliminazione di eventuali dati non causerebbe problemi di data leakage, ovvero di invalidità dei modelli a seguito dell'utilizzo di informazioni esterne al dataset di training.
Si segnala infine che la variabile relativa all'accessibilità dell'immobile non presenta invece una percentuale di dati mancanti plausibile. Nel solo $18,9\%$ dei casi l'immobile è segnalato come accessibile. La causa della mancanza di informazione non è quindi attribuibile all'impossibilità di accedere all'immobile. Per questo motivo si esclude tale variabile dal dataset.

5.2.3 Dataset Splitting

Dividere il dataset di partenza in training set e test set è una procedura fondamentale al fine di costruire dei modelli previsivi. Se mentre per l'applicazione di modelli statistici descrittivi non è di interesse verificare le capacità predittive, per i modelli statistici previsivi è necessario testare il modello su nuovi dati.
Per mantenere un buon bilanciamento tra capacità descrittive e previsive si decide quindi di dividere il dataset in training e test con una percentuale di osservazioni sul totale pari rispettivamente all'80% e al 20%. Questo rapporto di suddivisione (o split ratio) è adeguato alla dimensione campionaria osservata.
La teoria che regola il rapporto di suddivisione consiglia infatti di aumentare progressivamente la percentuale di osservazioni da includere nel training set al crescere del numero di osservazioni. La divisione del dataset è stata realizzata stratificando per la variabile risposta in modo da bilanciare i valori osservati dei prezzi nel training set con quelli del test set. Poiché la variabile risposta è quantitativa, la stratificazione è avvenuta in base ai decili della distribuzione.
Il training set servirà quindi per costruire ed allenare i modelli previsivi, mentre il test set servirà per stimare l'errore di previsione senza distorsione. Non viene invece introdotto un vero e proprio set di validazione. Come accennato nella parte teorica, utilizzeremo una processo di cross-validation per convalidare i modelli. Procedura che permette di ottenere dei risultati previsivi migliori rispetto alla consueta divisione del dataset in tre parti (training, validation e test).

5.2.4 Trattamento dei Dati Mancanti

Il trattamento dei dati mancanti è una delle parti principali e più delicate dell'intera analisi. Il numero di dati mancanti cambia notevolmente a seconda della variabile presa in esame. Si nota che solo le variabili esplicative presentano valori mancanti, in quanto la variabile risposta risulta completamente osservata.
Per evitare problemi di data leakage, si imputeranno separatamente i dati per il training e il test set. Prima i dati saranno infatti imputati per il training, poi, in un secondo momento, si andranno a riprodurre gli stessi metodi di imputazione utilizzati sul training set per imputare i dati anche sul test set.
Prima di procedere è importante verificare il tipo di dati mancanti. Non potendo verificare l'ipotesi di MAR contro l'alternativa di MNAR, è necessario fare riferimento alla conoscenza che è in nostro possesso delle variabili e del criterio con cui queste sono state misurate. Le variabili che andremo ad imputare sono relative a caratteristiche interne all'immobile. Per questo motivo non prenderemo in considerazione in questa fase le variabili di distanza. L'assenza di un dato è derivata dalla scelta più o meno intenzionale da parte del proprietario dell'immobile (o dell'agenzia immobiliare) di non includere nell'annuncio una specifica informazione. Il motivo della mancanza non è quindi totalmente aleatorio, ovvero non è ipotizzabile che i dati mancanti siano MCAR. Si pensa invece che i dati mancanti su una variabile siano determinabili attraverso le informazioni contenute nelle altre variabili.
Una caratteristica o un attributo di una abitazione dipende infatti strettamente dalle altre informazioni interne all'immobile stesso. Ad esempio, se un immobile è un appartamento collocato ad un piano elevato di un condominio di basso valore, allora

sarà maggiormente probabile che sia presente un ascensore e che la classe energetica non sia alta. Per questi motivi assumeremo che i dati mancanti siano di tipo MAR. Distinguendo quindi tra training e test set, riportiamo in tabella le variabili che possiedono dati mancanti con le rispettive percentuali di dati mancanti sul numero totale di osservazioni in ordine crescente:

Variabile	% Train	% Test	% Dataset
Numero di Bagni	0,1%	0,2%	0,1%
Superficie Costruita	0,1%	0,2%	0,1%
Nuova Costruzione	2,5%	1,5%	2,3%
Ascensore	5,3%	5,7%	5,3%
Esterno	8%	7,7%	8%
Piano d'Ingresso	8,7%	7,9%	8,5%
Certificazione Energetica	31,9%	32,4%	32%
Riscaldamento Autonomo	48,6%	48,2%	48,5%
Variabili di Orientamento	52,5%	48%	51,6%
Superficie Utile	58,6%	56,4%	58,2%
Anno di Costruzione	68,9%	70,7%	69,3%
Numero di Piani	96%	95,2%	95,8%
Superficie Coltivabile	97,4%	96,9%	97,3%

Tabella 5.1. Percentuali di Dati Mancanti

Si può osservare che alcune variabili hanno una percentuale di dati mancanti estremamente elevata. La loro inclusione nei modelli potrebbe portare a distorcere la validità dei risultati. Per questo motivo si è deciso di impostare delle soglie progressive di dati mancanti e verificare quale di queste restituisce dei risultati inferenziali migliori. Le soglie sono state impostate in base alle percentuali di valori mancanti osservati tra le variabili e sono pari al 5%, 25% e 50%.
Il metodo di imputazione su cui è stata svolta la verifica è il predictive mean matching, mentre i modelli tramite i quali si sono misurate le capacità predittive sono il modello di regressione lineare e il random forest con degli iperparametri comuni.
Le metriche valutate sono l'$RMSE$, l'MAE e l'R^2. Le misurazioni sono state fatte tramite una procedura di 10-fold cross-validation.

I risultati previsivi sono riportati nella tabella sottostante:

Soglia	Regressione Lineare			Random Forest		
	RMSE	MAE	R^2	RMSE	MAE	R^2
5	245647	147646	0,694	219478	127561	0,754
25	242669	144145	0,704	215632	121297	0,763
50	239525	144733	0,711	205800	118385	0,792

Tabella 5.2. Risultati per diverse Soglie di Dati Mancanti

Analizzando i risultati, si può osservare che i modelli basati su una soglia di imputazione del 50% restituiscono dei risultati sistematicamente migliori rispetto alle altre soglie sia per il modello di regressione lineare che per il random forest. Si è osservato inoltre che le variabili che sono state imputate sono sempre significative sotto il modello di regressione lineare e risultano determinanti secondo un criterio di importanza delle variabili per il random forest. Per ulteriori dettagli si rimanda all'analisi in appendice.

Andiamo ora ad applicare e confrontare i vari metodi di imputazione. Si utilizzano i metodi che sono stati presentati nella parte teorica. L'imputazione che verrà verificata è sia singola che multipla. In particolare, per l'imputazione multipla si è scelto un numero di dataset da imputare pari a 5. Il confronto avviene attraverso gli I-Scores nella forma di Density Ratio con diversi numeri di proiezioni aleatorie. Ad esclusione dell'imputazione con media, moda e mediana, nella totalità dei casi abbiamo utilizzato, sia nel caso dell'imputazione singola che multipla, la strategia di imputazione basata sulla specificazione completamente condizionale nella versione dell'algoritmo con equazioni concatenate (MICE).
La sequenza di imputazione delle variabili è in ordine crescente rispetto alla percentuale di dati mancanti sul totale delle osservazioni. Per la fase di imputazione è stata presa in considerazione anche la variabile risposta. Come segnalato infatti dallo studio di Evangelos Kontopantelis, la variabile risposta può essere utilizzata nel modello di imputazione senza andare a distorcere i risultati dei modelli.
Una problematica che sorge con l'imputazione multipla all'interno della nostra applicazione consiste nel fatto che i modelli di regressione che impiegheremo non sono tutti parametrici, per questo motivo non risulta chiaro come svolgere il pooling dei risultati. A tal fine si decide di utilizzare per l'analisi di regressione la sola imputazione singola prescelta, mentre l'imputazione multipla viene impiegata per valutare la consistenza dei metodi di imputazione confrontati tramite I-Scores.
Si riportano nella tabella 5.3 i valori degli I-Scores in termini relativi (il metodo di imputazione con score più elevato ha valore 0, mentre gli altri mostreranno un valore inferiore) sia per l'imputazione singola che per quella multipla:

Metodo	Imputazione Singola Numero di Proiezioni				Imputazione Multipla Numero di Proiezioni			
	5	10	20	50	5	10	20	50
MMM	-2,514	-2,291	-2,073	-2,163	Non Valutabile			
LM+LOG	-1,879	-2,149	-1,610	-1,756	-2,303	-1,905	-1,830	-1,723
SR+LOG	-1,295	-1,162	-0,750	-0,681	-1,663	-1,054	-1,074	-0,998
PMM+LOG	0	-0,098	-0,484	-0,533	0	-0,688	-0,757	-0,808
PMM	-0,325	-0,680	-0,605	-0,190	-0,733	-0,342	-0,579	-0,405
CART	-0,225	0	0	0	-0,562	-0,209	-0,256	-0,202
RF	-0,417	-0,255	-0,196	-0,286	-0,543	0	0	0

Legenda: MMM: Media, Moda e Mediana - LM+LOG: Regressione Lineare e Logistica SR+LOG: Regressione Stocastica e Logistica - PMM+LOG: Predictive Mean Matching e Regressione Logistica - PMM: Predictive Mean Matching - CART: Albero Decisionale RF: Random Forest

Tabella 5.3. I-Scores

Dai punteggi di imputazione il metodo basato su media, moda e mediana è, come ci si aspettava, quello che restituisce i risultati peggiori. Infatti, questo metodo può distorcere significativamente la distribuzione originaria dei dati.
I metodi basati sulla regressione lineare e stocastica (per le variabili quantitative) uniti alla regressione logistica mostrano anch'essi delle performance non desiderabili. In particolare, la regressione lineare mostra dei risultati che non si allontanano molto dall'imputazione tramite media, moda e mediana; la regressione stocastica si dimostra in tutti i casi migliore di quella lineare, ma sistematicamente peggiore dei metodi rimanenti. L'utilizzo combinato del predictive mean matching con la regressione logistica è preferibile se si considerano solo 5 proiezioni aleatorie sia per l'imputazione singola che multipla. Considerando un numero superiore di proiezioni tale metodo presenta dei risultati peggiori e non consistenti. Lo score per il metodo di imputazione basato sul solo predictive mean matching ha invece un andamento altalenante. I migliori risultati sono dati dai due modelli basati sugli alberi. L'albero decisionale è il preferibile per quanto riguarda l'imputazione singola, mentre il random forest per l'imputazione multipla. Il random forest è, di conseguenza, più consistente del singolo albero decisionale quando si imputano più dataset. I punteggi più elevati dell'albero decisionale per l'imputazione singola sono da associare in parte alla sua elevata variabilità per costruzione. Si decide quindi di imputare i dati attraverso il random forest. Lo stesso metodo di imputazione viene applicato sul test set con la differenza che non viene presa in considerazione la variabile risposta per la stima dei modelli.

5.3 Analisi Esplorativa

L'analisi esplorativa è un approccio all'analisi per avere una prima idea sui dati riassuemendo le caratteristiche principali del fenomeno d'interesse, spesso utilizzando grafici statistici e altri metodi di visualizzazione dei dati. Si tratta infatti di una fase preliminare alla modellazione, tramite la quale si individua la distribuzione empirica dei dati e i rapporti che caratterizzano le variabili.
Abbiamo diviso l'analisi esplorativa in esplorazione standard (EDA) ed esplorazione spaziale (ESDA). Le analisi che illustreremo fanno riferimento a sole alcune variabili statistiche, per un analisi completa si rimanda all'appendice.

5.3.1 Esplorazione Standard

Nell'esplorazione standard si analizzano principalmente le distribuzioni delle variabili. A tal fine impostiamo l'analisi dividendola tra univariata e bivariata.

Analisi Univariata

Si vanno ad analizzare le statistiche sommarie delle variabili e le loro distribuzioni. In questo senso riportiamo alcuni grafici relativi alla variabile risposta:

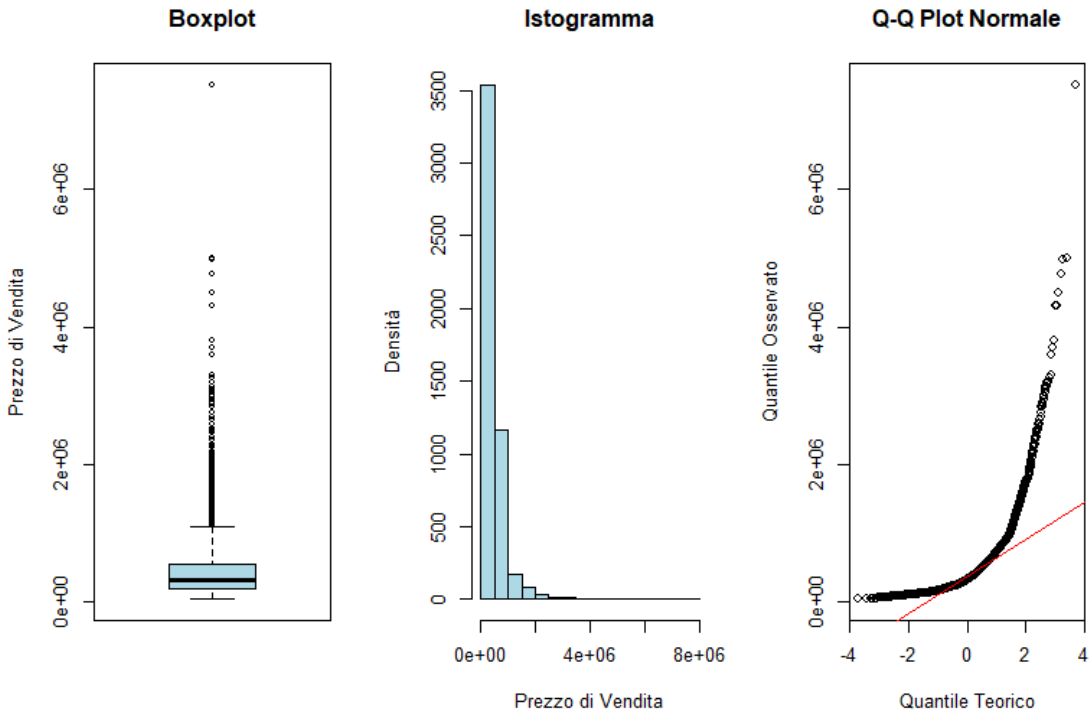


Figura 5.1. Boxplot, Istogramma e Q-Q plot per il Prezzo di Vendita

Il prezzo di vendita degli immobili varia da un minimo di 42.000 € ad un massimo di 7.525.000 €, con frequenze assolute molto elevate tra i 200.000 € e i 500.000 €. Il prezzo medio di vendita è di 460.105 € e quello mediano di 321.000 €. Dal boxplot e dall'istogramma si evince che sia presente una evidente asimmetria positiva,

confermata dal fatto che la media sia superiore alla mediana. Il q-q plot suggerisce che non sia possibile assumere la normalità per la distribuzione.

Analizziamo ora il prezzo di vendita con la sua trasformata logaritmica:

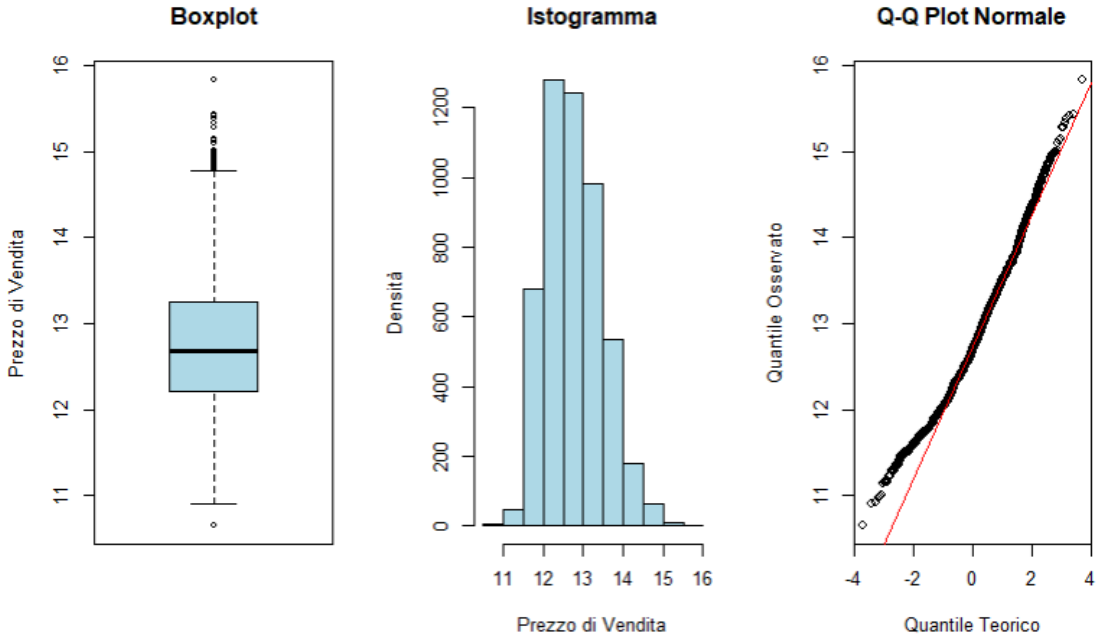


Figura 5.2. Boxplot, Istogramma e Q-Q plot per il logaritmo del Prezzo di Vendita

Si può notare dalla figura che la distribuzione osservata è più simile a quella di una normale, ma le code risultano però ancora pesanti.

Nella figura 5.3 si possono poi osservare le distribuzioni delle altre variabili quantitative. Si nota che nessuna delle variabili sembra seguire una distribuzione normale. In particolare, la variabile Piano d'Ingresso possiede una distribuzione simmetrica con code pesanti, mentre tutte le altre distribuzioni hanno un'asimmetria positiva.

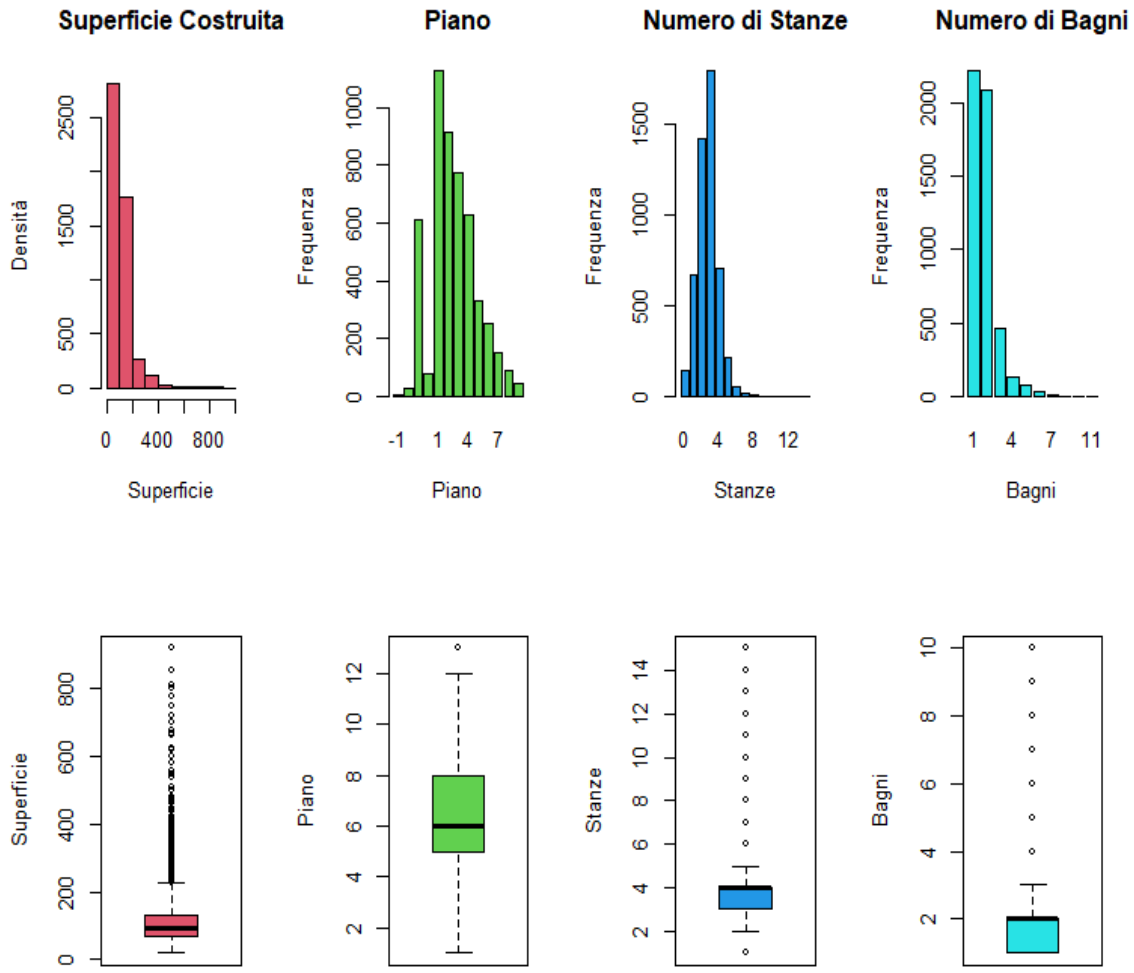


Figura 5.3. Istogrammi (o Diagrammi a Barre) e Boxplot delle Variabili Quantitative

Analisi Bivariata

Nell'analisi esplorativa bivariata si prende in considerazione due variabili per volta. Si mostrano allora nella figura 5.4 i diagrammi di dispersione delle variabili quantitative. Il colore dei grafici varia in base all'intensità della correlazione lineare tra le variabili prese a due a due. Il colore rosso indica una correlazione lineare forte ($> 0,7$), il giallo una correlazione lineare moderata ($> 0,3$) e in verde troviamo una correlazione debole ($< 0,3$).
Tra tutte le variabili è presente una correlazione lineare positiva. Il Prezzo di Vendita è correlato fortemente con la Superficie Costruita e il Numero di Bagni, più leggera invece è la correlazione con il Numero di Stanze e il Piano d'Ingresso. Anche tra la Superficie Costruita e il Numero di Bagni è presente una forte correlazione, mentre, tra le altre variabili, la correlazione risulta modesta o leggera.

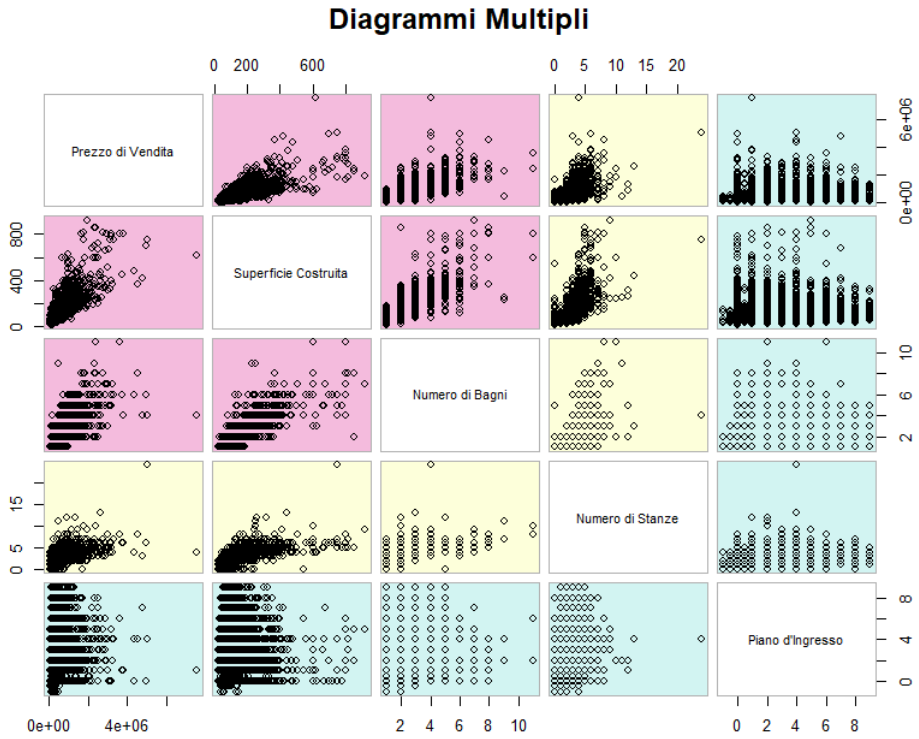


Figura 5.4. Diagrammi a Dispersione

Visualizziamo poi la distribuzione congiunta tra il Prezzo di Vendita e due variabili categoriche attraverso dei boxplot condizionati alle modalità assunte.

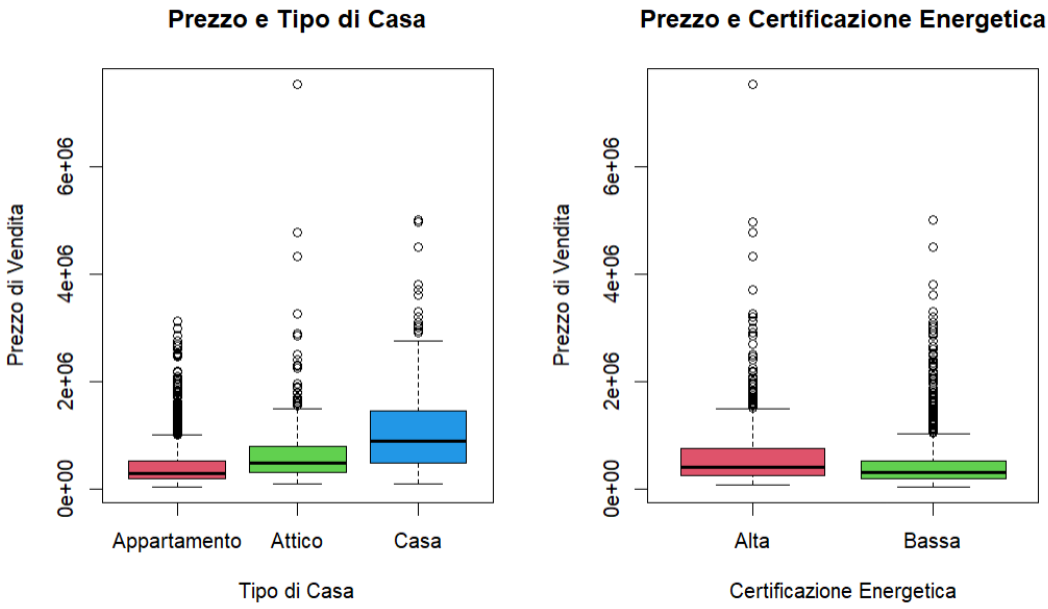


Figura 5.5. Boxplot di Variabili Categoriche in base al Prezzo di Vendita

Il Tipo di Casa influenza notevolmente il Prezzo di Vendita. Le case indipendenti sono in media gli immobili più costosi, seguno poi gli attici e infine gli appartamenti. Per la Certificazione Energetica le differenze di prezzo sono meno evidenti. Le abitazioni con classe elevata sono associate a prezzi più elevati, mentre una classe bassa comporta dei valori di prezzo leggermente più bassi.

5.3.2 Esplorazione Spaziale

Nell'analisi esplorativa spaziale andiamo ad osservare se intuitivamente è presente l'autocorrelazione spaziale tra i prezzi delle abitazioni. In primo luogo dividiamo la variabile prezzo di vendita in quartili e mostriamo sulla mappa stradale di Madrid la posizione delle abitazioni in base alla nuova variabile così trasformata:

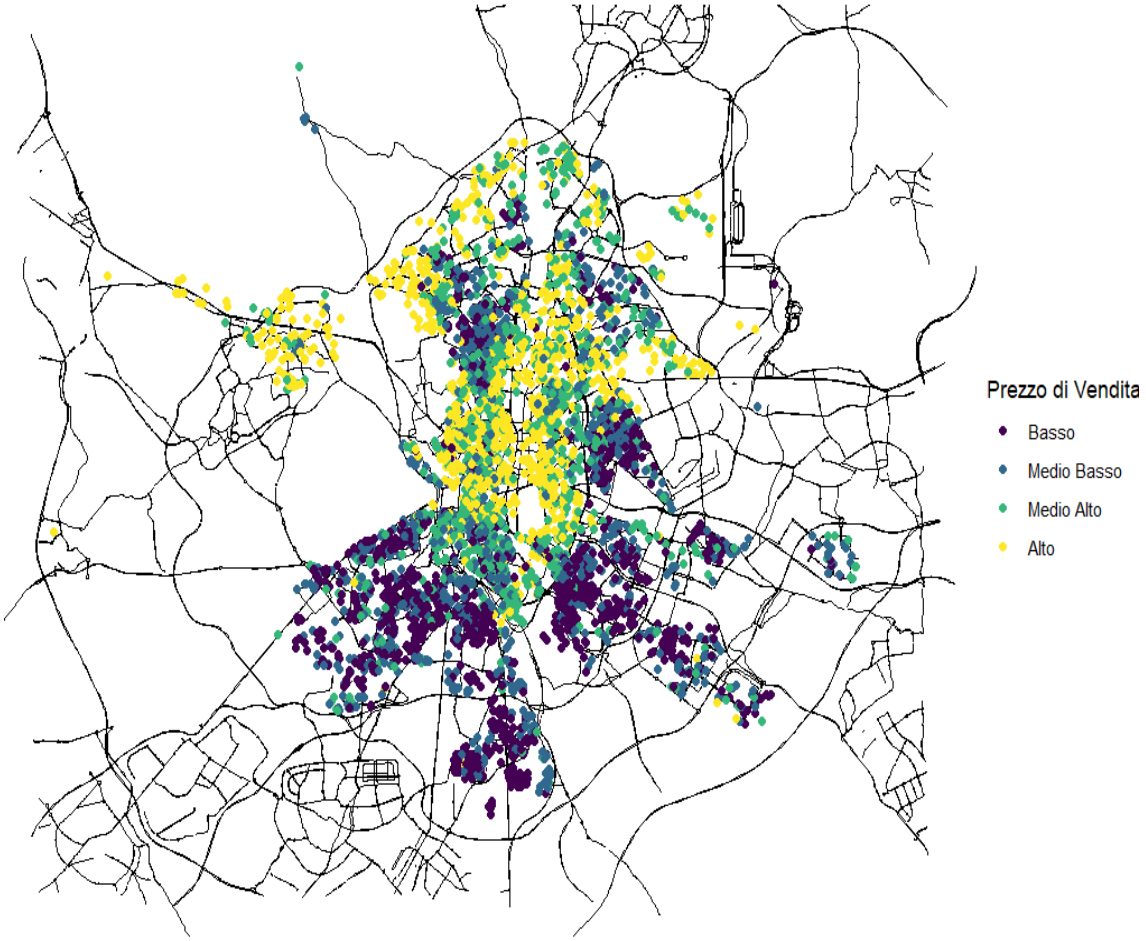


Figura 5.6. Mappa dei Prezzi di Vendita

Si può notare che i prezzi di vendita si presentano in modo eterogeneo sulla mappa; in particolare, gli immobili più cari sono situati nella zona centrale e a nord-ovest della città, mentre quelli più economici si trovano soprattutto nel sud di Madrid. Questo grafico suggerisce quindi che vi sia una evidente forma di autocorrelazione spaziale.

Successivamente, si deve cercare di determinare la struttura di vicinanza più idonea. Dovendo trattare con unità spaziali di punto non diviene naturale utilizzare delle strutture di vicinanza basate sulla contiguità spaziale. Risulta quindi preferibile una struttura basata sulla distanza. Tale struttura è specificata in base ad un numero fissato di vicini o ad una fissata distanza soglia.
Non esiste un criterio univoco per decidere il numero k di vicini o la distanza soglia d. Questi valori devono necessariamente dipendere dal numero di unità del dataset. A questo fine calcoliamo il coefficiente di correlazione di Pearson ρ per il prezzo di vendita a vari livelli di k. Un valore $\mid \rho \mid > 0,7$ indica una correlazione lineare forte, $0,3 < \mid \rho \mid < 0,7$ una correlazione lineare moderata e $\mid \rho \mid < 0,3$ una correlazione lineare debole. Mostriamo l'andamento della correlazione lineare in un diagramma:

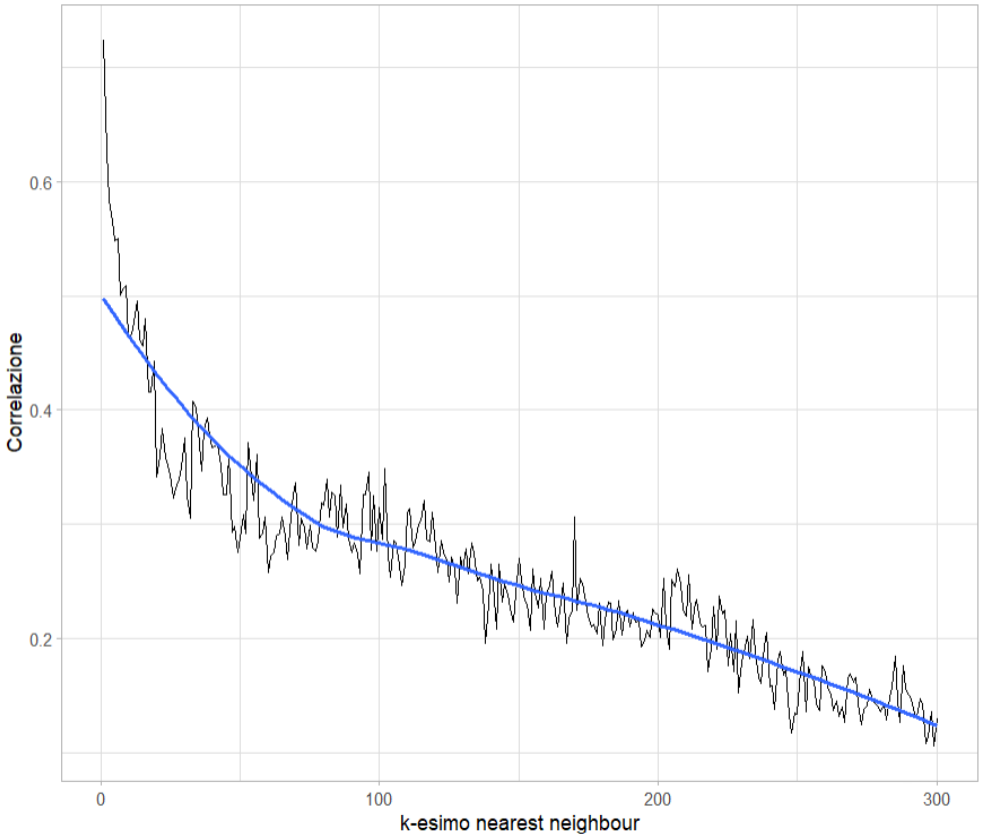


Figura 5.7. Andamento della Correlazione Lineare in base al Numero dei Vicini k

Dalla figura si osserva un valore dell'indice consistentemente superiore a 0,3 in corrispondenza di un numero di vicini fino a 75. Decidiamo quindi di impostare $k = 75$. In maniera analoga si è visto che è possibile ottenere un numero medio di vicini per abitazione approssimabile a 75 per una distanza soglia d di 800 metri.
Nel decidere quale tipologia di struttura di vicinanza utilizzare, si tiene conto di alcune caratteristiche determinanti. Sono presenti alcune abitazioni distanti dalle altre che vengono rappresentate dai dei punti isolati sulla mappa. Nel caso si utilizzi un numero di vicini fissato k, non saranno presenti unità senza vicini. Al contrario, fissata una distanza d, si è osservato che alcune unità non hanno alcun vicino. Per

fare in modo che ogni unità abbia almeno un vicino bisognerebbe fissare una distanza superiore ai 3000 metri, ma il numero medio di vicini aumenterebbe a dismisura. In ogni caso molte unità hanno un numero di vicini eccessivamente basso o elevato. D'altra parte, se viene fissato un numero di vicini, le stime dei prezzi degli immobili isolati saranno necessariamente influenzati anche da abitazioni molto distanti.
In ogni caso, fissare il numero di vicini permette sia di controllare il coefficiente di correlazione osservato di Pearson in maniera diretta, sia di poter impiegare una struttura di vicinanza analoga anche sul dataset di test. In definitiva è quindi preferibile utilizzare un numero fissato di vicini.
La matrice dei pesi spaziali corrispondente alla struttura di vicinanza è determinata dai valori inversi delle distanze di Haversine, aggiustate in modo tale da evitare valori invalidi. Specificatamente, prima di invertire le distanze, a queste è stato aggiunto un termine costante pari ad 1 metro. Ciascun peso spaziale viene poi successivamente standardizzato per riga.

Visualizziamo ora il Diagramma di Moran sul prezzo di vendita:

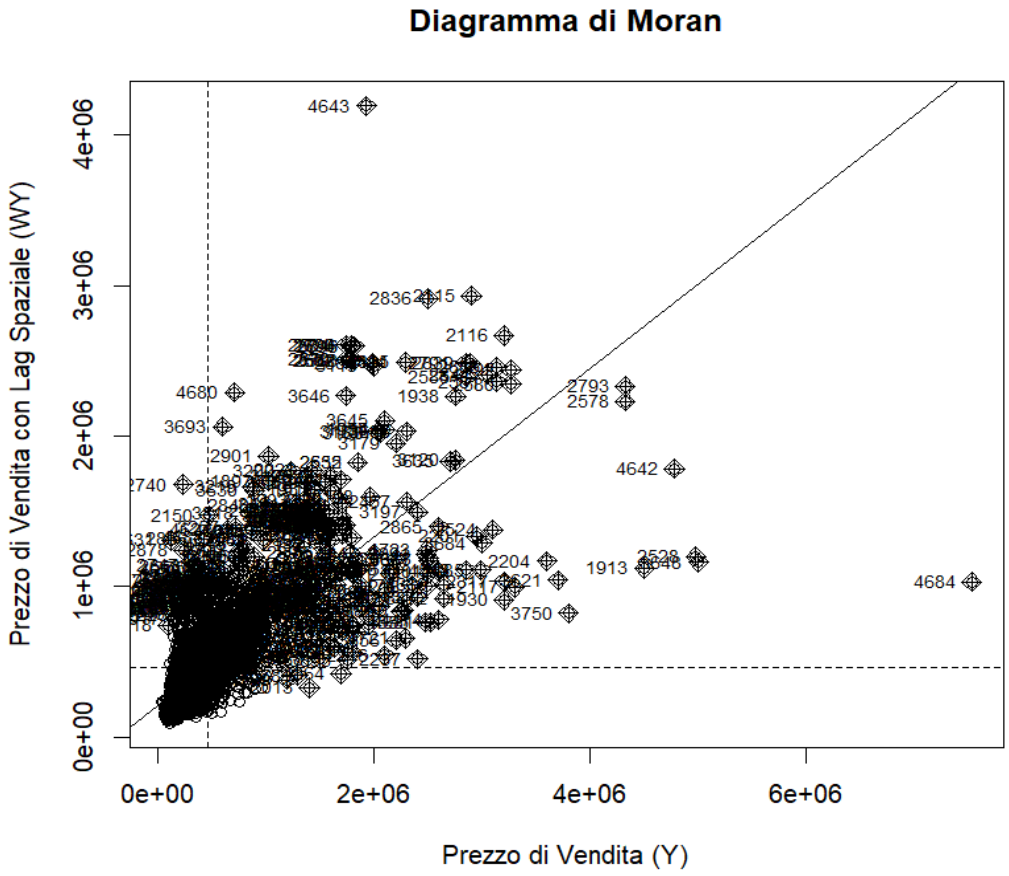


Figura 5.8. Diagramma di Moran sul Prezzo di Vendita

I prezzi degli immobili sono sull'asse orizzontale e le loro controparti spazialmente ritardate sono sull'asse verticale. Le osservazioni nel grafico sono sbilanciate e si concentrano soprattutto tra il primo e il terzo quadrante indicando una tendenziale

presenza di autocorrelazione positiva. Si può notare che la forma della nuvola di punti è influenzata dalla presenza di diversi outliers nel primo quadrante. Nel terzo quadrante i punti sono molto meno sparsi e finiscono per raggrupparsi.
In generale, casi "High-High" sono maggiormente presenti dei "Low-Low", di conseguenza, ci sono più abitazioni costose all'interno di zone altolocate rispetto ad abitazioni di prezzo basso in zone economiche.
L'indice di Moran osservato (corrispondente alla pendenza della retta di regressione nel diagramma di Moran) associato alla matrice dei pesi spaziali è pari a $I = 0,561$ e il p-value relativo alla statistica test ottenuto tramite simulazioni Monte Carlo è prossimo a 0. L'indice di Geary è invece pari a $C = 0,403$ e il p-value è di 0,02. Entrambi gli indici testimoniano quindi, come anticipato, una sostanziale presenza di autocorrelazione spaziale positiva all'interno del dataset che viene confermata dal livello di significatività dei rispettivi test statistici (fissando una soglia di significatività del 5%). A livello globale ciò significa che i valori dei prezzi degli immobili tenderanno a raggrupparsi spazialmente sulla mappa di Madrid con una certa consistenza.

5.4 Analisi di Regressione

5.4.1 Analisi Classica

Si vanno a confrontare le varie tecniche presentate nel paragrafo 3.1 per introdurre le variabili spaziali che hanno la funzione di sottomercati spaziali.
In tal senso si considerano dapprima le aree amministrative: la città di Madrid è divisa in 21 distretti i quali sono suddivisi a loro volta in 131 quartieri (o barrios). Si è verificato che l'utilizzo del quartiere come variabile spaziale comporta un aumento eccessivo del costo computazionale, nonché un aumento delle difficoltà interpretative dei modelli dovuto al numero elevato di variabili dummy da introdurre. Inoltre, il quartiere non risulta essere un'entità con un potere amministrativo locale adeguato. Di conseguenza, senza perdita di specificità, si prenderanno in considerazione i distretti. Questi hanno una sufficiente autorità all'interno della città e vanno a rappresentare il livello di suddivisione territoriale immediatamente successivo a quello delle province.
Per i cluster spaziali, come anticipatamente suggerito, si sono impiegati i modelli di clustering a mistura finita con componenti Gaussiane unite al criterio di informazione Bayesiana. In base a 7 possibili modelli selezionabili e ad un limite massimo di 25 cluster spaziali, il modello con le prestazioni preferibili è risultato essere quello composto da 15 cluster con distribuzione elissoidale, volume costante e forma equa (EEV). L'inizializzazione di riferimento è stata creata tramite agglomerazioni casuali per rendere i modelli più consistenti e le variabili sono state standardizzate tramite una decomposizione SVD.
Per ultimi, i cluster LISA di tipo Moran sono stati costruiti fissando il numero di vicini in modo analogo a quanto visto in analisi esplorativa, sulla base però dell'intero dataset. In questo caso il numero di vicini considerati è pari a $k = 100$, poiché si è osservato un indice di Pearson consistentemente superiore a 0,3 in corrispondenza di tale valore. Il numero di vicini fissati risulta leggermente superiore al caso in cui si consideri solo il dataset di training, dove veniva invece selezionato $k = 75$. Anche in

questo caso la matrice dei pesi spaziali è determinata dai valori inversi aggiustati delle distanze di Haversine e ognuno di essi viene successivamente standardizzato per riga. Si è quindi osservato che 1837 unità hanno autocorrelazione spaziale positiva di tipo "High-High" e 3218 di tipo "Low-Low". Mentre 296 unità hanno autocorrelazione spaziale negativa "High-Low" e 936 "Low-High".

Riportiamo nella tabella sottostante i confronti tra le capacità predittive del modello di regressione lineare e del random forest con iperparametri comuni per le varie scelte dei criteri di determinazione dei sottomercati spaziali:

Metodo	Regressione Lineare			Random Forest		
	RMSE	MAE	R^2	RMSE	MAE	R^2
Aree Amministrative	206898	118509	0,786	171237	76977	0,861
Cluster Spaziali	207322	118234	0,787	174633	78709	0,854
Cluster LISA	207425	115640	0,785	165349	71593	0,868

Tabella 5.4. Confronto tra Sottomercati Spaziali

Con il modello di regressione lineare i metodi hanno delle prestazioni previsive simili preferendo leggermente uno dei tre modelli a seconda dell'indicatore preso come riferimento. Nel random forest queste differenze sono più evidenti e il modello basato sui cluster LISA risulta sempre il preferibile. Generalmente il random forest con i cluster LISA risulta essere il miglior modello, dove viene raggiunto un $RMSE$ di 165349. Oltretutto il numero di predittori introdotti nei modelli con i cluster LISA è di 4, quindi decisamente inferiore agli altri casi. Per questi motivi si decide di impiegare i cluster LISA come sottomercati spaziali.

Modello Lineare

Applichiamo il modello di regressione lineare. Nella costruzione dei modelli le variabili selezionate si basano su delle soglie del p-value pari a 0,05. Si costruisce quindi inizialmente il modello regressivo basato sulle sole variabili interne all'immobile, ovvero quelle che non fanno riferimento alla collocazione spaziale.

Si indicano in tabella 5.5 i risultati del modello così trovato con i relativi coefficienti di regressione, errori standard ed estremi degli intervalli di confidenza al 95% di tipo Wald. Dalla tabella si può vedere che numerose variabili non sono state selezionate nel modello per insufficiente livello di significatività dei coefficienti di regressione associati. In particolare, sono state scartate le variabili relative alla certificazione energetica, al piano d'ingresso, alla necessaria ristrutturazione, alla nuova costruzione e gli indicatori di presenza/assenza dell'armadio a muro, del balcone, del parcheggio, della piscina e del ripostiglio.

Parametro	Stima	Errore Standard	IC_{inf}	IC_{sup}
Intercetta	-80902	17586	-115378	-46427
Superficie Costruita	3851	83	3689	4013
Numero di Bagni	102406	6397	89865	114946
Numero di Stanze	-17203	3852	-24754	-9652
Attico	137106	14306	109059	165152
Casa Indipendente	-367574	20714	-408184	-326965
Esterno	-64267	12310	-88400	-40133
Nuova Costruzione	110494	10233	90433	130555
Aria Condizionata	30692	7552	15886	45497
Ascensore	51886	9400	33458	70315
Giardino	-50734	8115	-66643	-34825
Riscaldamento Autonomo	-45219	9243	-63339	-27099

Tabella 5.5. Modello di Regressione Lineare Ridotto

Analizziamo ora le variabili selezionate. Il parametro legato alla superficie costruita indica che a parità delle altre variabili il prezzo di un immobile aumenta di 3851 € per un aumento unitario dei metri quadri costruiti, ovvero sotto le dovute ipotesi si ha un prezzo di 3851 €/m^2. Si nota che il modello con sola intercetta e superficie costruita mostrerebbe un prezzo di 4210 €/m^2. Questo risultato è in linea con il prezzo medio al metro quadro di marzo 2020 segnalato dalle principali analisi di mercato. Ad esempio, un'indagine del portale immobiliare spagnolo indomio.es ha indicato un prezzo medio di 3643 €/m^2 nel comune di Madrid a marzo 2020.
Al pari delle altre variabili un maggior numero di bagni comporta ad un netto aumento del prezzo di vendita dell'immobile (102406€ in più per ciascun bagno), mentre un maggior numero di stanze implica un leggero abbassamento del prezzo.
Con gli stessi presupposti, per quanto riguarda la tipologia di abitazione, un attico ha un maggior valore rispetto ad un appartamento (baseline del modello), invece una casa indipendente fa scendere drasticamente il prezzo di vendita.
Gli attributi che aggiungono valore all'immobile a parità delle altre variabili sono l'aria condizionata e l'ascensore, al contrario, la presenza dell'esterno, del giardino e del riscaldamento autonomo ne detraggono il valore.

Si ripete l'analisi di regressione lineare aggiungendo le variabili legate alle distanze dai luoghi d'interesse e le variabili spaziali che segnalano l'appartenenza ad un determinato cluster LISA:

Parametro	Stima	Errore Standard	IC_{inf}	IC_{sup}
Intercetta	143200	18020	107907	178562
Superficie Costruita	3308	70	3170	3445
Numero di Bagni	67510	5390	56947	78080
C. Energetica Bassa	-34730	8295	-50998	-18472
Piano d'Ingresso	6123	1501	3182	9065
Attico	117600	12610	92934	142362
Casa Indipendente	-234000	18780	-270814	-197172
Necessaria Ristrutturazione	-39620	9070	-57405	-21844
Nuova Costruzione	102400	9522	83706	121042
Piscina	36480	8520	19778	53183
Riscaldamento Autonomo	31420	8170	15408	47441
D. Supermercato	50	13	25	75
D. Farmacia	-91	28	-146	-36
D. Banca	-64	12	-86	-41
D. Università	-22	4	-30	-14
D. Scuola dell'Obbligo	-82	18	-116	-47
D. Scuola dell'Infanzia	160	18	124	195
D. Stazione dei Treni	-14	3	-20	-8
D. Parco	228	30	169	287
D. Stadio	-17	2	-21	-13
D. Discoteca	-13	3	-20	-6
D. Cinema	-19	4	-27	-11
D. Biblioteca	37	8	22	52
D. Attrazione Turistica	-14	3	-20	-7
Cluster High-Low	-167300	15030	-196756	-137814
Cluster Low-High	-128100	11170	-149969	-106174
Cluster Low-Low	-179700	10590	-200451	-158911

Tabella 5.6. Modello di Regressione Lineare Completo

Dalla tabella emerge che alcune variabili interne all'abitazione, prima significative, ora non lo sono più e viceversa. Il numero di stanze, l'esterno, l'aria condizionata, l'ascensore e il giardino sono tutte variabili che non sono state più selezionate.
Al contrario, la certificazione energetica, il piano d'ingresso, gli indicatori di necessaria ristrutturazione e nuova costruzione e la presenza/assenza della piscina diventano variabili da includere nel modello. Tra le variabili già presenti nel modello ridotto, si segnala che i segni dei coefficienti di regressione vengono mantenuti, ad esclusione del riscaldamento autonomo. Il valore positivo di quest'ultimo indica infatti che il riscaldamento autonomo aggiunge valore ad un immobile. E' interessante notare anche come il valore del parametro relativo alla superficie costruita si sia abbassato a 3308€. Per quanto riguarda le nuove variabili, la classe energetica di livello basso e la necessaria ristrutturazione tolgono di valore all'immobile, mentre, se l'abitazione fa parte di una nuova costruzione o possiede al suo interno una piscina, la valutazione dell'immobile sale.
Per quanto concerne le variabili basate sulla distanza, un valore di stima positivo indica che all'allontanamento di un'unità di distanza (il metro) dell'abitazione da un determinato luogo d'interesse, segue un prezzo dell'immobile maggiore che crescerà di una quantità di denaro pari al coefficiente di regressione di riferimento. Se il valore di stima è invece negativo, il prezzo dell'immobile diminuirà linearmente all'allontanamento del luogo d'interesse.
Infine, i valori dei parametri dei cluster LISA negativi indicano che il prezzo dell'immobile è inferiore al prezzo di base se non si trova in una zona considerata di tipo "High-High" (che viene presa come baseline).

Eseguiamo delle analisi dignostiche del modello basate sull'andamento dei residui del modello:

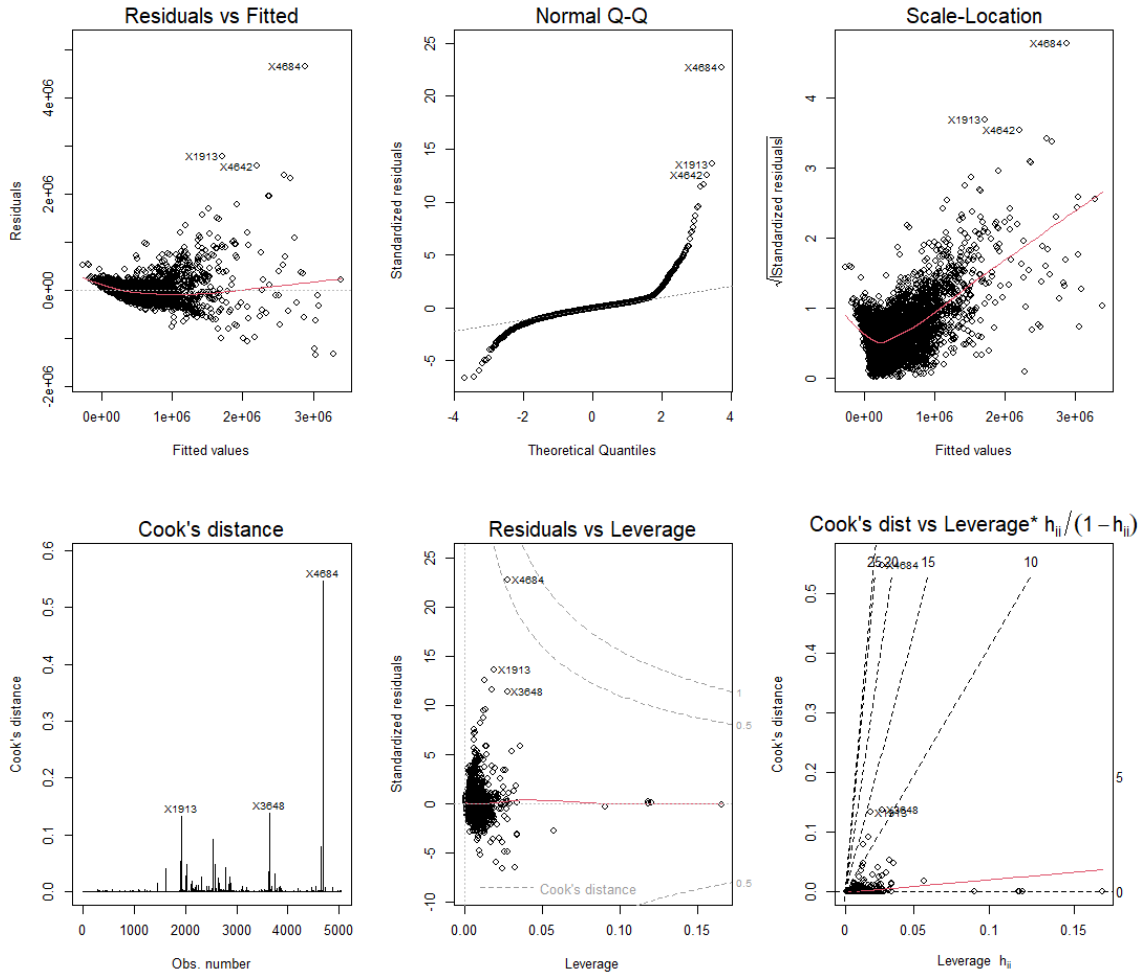


Figura 5.9. Diagnostiche del Modello
1) Residuals vs Fitted 2) Normal Q-Q Plot 3) Scale-Location 4) Cook's Distance 5) Standardized Residuals vs Leverage 6) Cook's Distance vs Standardized Leverage

Dal primo grafico appare che l'assunzione di linearità sia valida, infatti i residui sono distribuiti uniformemente attorno allo 0 al variare dei valori previsti della Y. L'assunzione di normalità non sembra invece valida. Il Q-Q plot mostra infatti che i quantili dei residui standardizzati non seguono una distribuzione normale standard e che quindi sono difformi dai quantili teorici. Quest'assunzione non è tuttavia essenziale al fine degli scopi della nostra indagine, in quanto siamo interessati maggiormente all'aspetto previsionale dei modelli.
Come si può vedere dal primo e dal terzo grafico l'ipotesi di omoschedasticità è evidentemente violata. All'aumentare dei valori assunti della variabile risposta, la variabilità delle stime aumenta nettamente. Questo riflette sia nella relazione tra i residui e i valori previsti, che nella relazione tra residui standardizzati e valori previsti. In caso di omoschedasticità, il terzo grafico mostrerebbe infatti dei residui

standardizzati distribuiti attorno al valore 1 al variare dei valori previsti della Y. Un metodo per risolvere il problema della mancata omoschedasticità consiste nel trasformare la variabile risposta attraverso una funzione logaritmica. Si decide quindi di passare da un modello lineare ad uno log-lineare.
Abbiamo inoltre osservato che l'introduzione di alcune variabili esplicative che sono state ricavate come trasformazioni quadratiche delle originali portano ad un modello che descrive meglio il rapporto tra la variabile risposta e i regressori.
Infine, si verifica la presenza di multicollinearità. La verifica avviene generalmente a livello quantitativo attraverso i fattori di inflazione della varianza (VIF) associati ciascuno ai coefficienti di regressione. Il VIF indica quanto una determinata variabile esplicativa sia dipendente dalle altre. Nel nostro caso, fissando una soglia limite del VIF pari a 5 (valore solitamente consigliato), si ha che le variabili che si presentano come trasformata quadratica sono collineari con la loro forma semplice. Questo risultato è dovuto al metodo di costruzione stesso della forma polinomiale del modello e non rappresenta un problema a livello inferenziale. Si segnala oltretutto che i grafici dal 4 al 6 non indicano la presenza di outliers o punti leva.

Riportiamo nella tabella sottostante il confronto a livello previsivo tra tutti e 4 i modelli che sono stati presi in considerazione:

Modello	Capacità Previsive		
	RMSE	MAE	R^2
Modello Lineare Ridotto	243064	144218	0,703
Modello Lineare Completo	207425	115640	0,785
Modello Log-Lineare Ridotto	248837	135127	0,692
Modello Log-Lineare Completo	189767	87296	0,814

Tabella 5.7. Confronto tra Modelli di Regressione Lineare

Il modello log-lineare completo è quello che prevede più accuratamente i prezzi degli immobili secondo tutti e tre gli indici di previsione. Una grande differenza è stata generalmente osservata tra i modelli di regressione ridotti e quelli completi, a favore di questi ultimi. Simili prestazioni si hanno invece tra il modello di regressione lineare ridotto e quello log-lineare.
Per apportare ulteriori migliorie potrebbero essere introdotti degli effetti di interazione tra le variabili a discapito di una maggiore difficoltà interpretativa.

Modelli Lineari Generalizzati

Considerando il prezzo di vendita dell'immobile come una variabile quantitativa continua, i modelli lineari generalizzati che si possono utilizzare devono avere un supporto per la variabile risposta che coincida con l'insieme dei numeri reali (o in questo caso anche solo reali positivi). I modelli che soddisfano questo requisito si

basano sulla distribuzione normale, esponenziale o gamma.
Altrimenti, se il prezzo di vendita viene trattato come variabile discreta, si può impiegare una distribuzione di Poisson.
Il modello normale viene stimato sulla variabile risposta non trasformata e la funzione legame utilizzata è logaritmica. Tra il modello esponenziale e gamma non ci sono per costruzione stime dei coefficienti di regressione differenti. Prendendo quindi in considerazione il più generale modello gamma, si utilizza come risposta il logaritmo del prezzo di vendita e la funzione legame è quella canonica (per il modello glm gamma la funzione legame canonica è l'inversa). Infine, per il modello di Poisson non viene applicata alcuna trasformazione sulla risposta e anche in questo caso si applica la sua funzione legame canonica (la funzione logaritmica).

Si mettono a confronto i modelli appena citati:

Distribuzione	Capacità Previsive		
	RMSE	MAE	R^2
Normale	166762	90323	0,858
Gamma	188031	88223	0,814
Poisson	172424	85709	0,849

Tabella 5.8. Confronto tra GLM

Tenendo conto del solo indicatore $RMSE$, tutti e tre i modelli glm hanno capacità previsive maggiori rispetto al modello regressivo lineare e log-lineare. In particolare, il modello glm normale con funzione legame logaritmica risulta essere il miglior modello per quanto riguarda l'$RMSE$ e l'R^2. Il modello di Poisson è preferibile secondo l'MAE e risulta nel complesso molto ben bilanciato per tutti gli indicatori.

Modelli con Regolarizzazione

I modelli con regolarizzazione sono stati costruiti a partire dal modello regressivo lineare completo in modo tale da poter mettere a paragone i risultati trovati in modo immediato e avere un'idea generale dell'impatto che avrebbero le tecniche di regolarizzazione sulle capacità previsive di un modello.
In successione abbiamo apportato una regolarizzazione di tipo Ridge, seguita dalla Lasso e infine dall'Elastic Net. Per selezionare un'adeguata costante di restringimento λ (l'iperparametro dei modelli) nel caso Ridge e Lasso, sono stati messi a confronti dei valori in un range tra 0,1 e 10000 intervallati su una scala logaritmica. L'insieme di valori da assegnare a λ è stato ottenuto osservando graficamente che un valore fuori range non apporta alcun miglioramento a livello previsivo.
Per l'Elastic Net si è presa in considerazione la forma semplicficata della funzione di costo (formula 3.9) ed è stata eseguita una Grid Search dell'iperparametro λ nello stesso range di valori di cui sopra, unita alla ricerca della costante regolatrice α nell'intervallo $[0, 1]$.

Si visualizza graficamente il processo di selezione degli iperparametri:

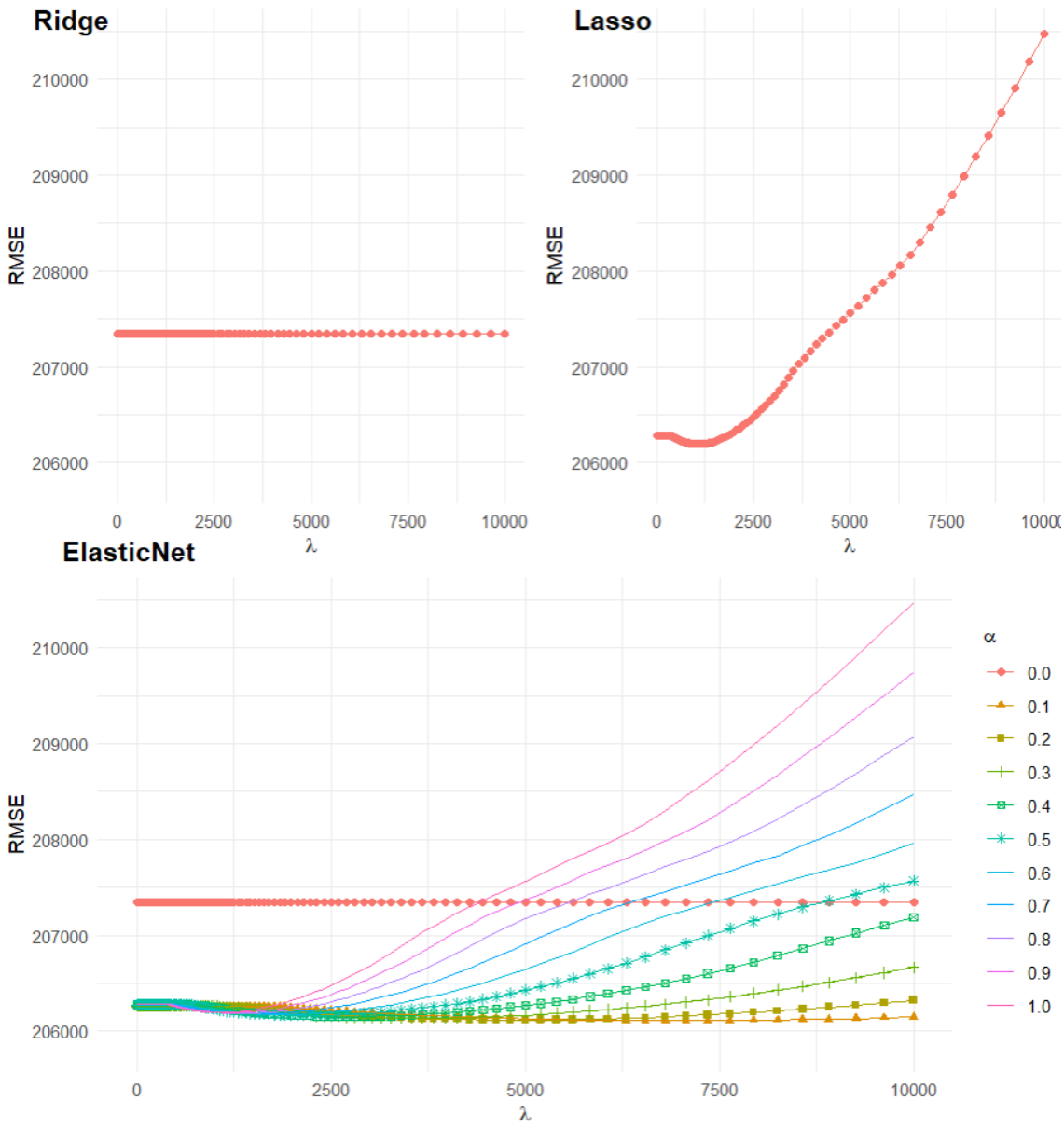


Figura 5.10. Processo di Regolarizzazione

La regressione Ridge non apporta alcun miglioramento di tipo previsionale rispetto al modello regressivo lineare per valori di λ fino a 10000. Più in generale questo avviene per $\lambda \leq 33685$, mentre per valori maggiori di questa soglia l'$RMSE$ di cross-validation aumenta in maniera costante. La regressione Lasso porta invece ad un miglioramento previsivo rispetto al modello regressivo lineare (e quindi alla regressione Ridge) abbassando l'$RMSE$ da 207425 a 206193 con $\lambda = 1031$.
Infine, risultati migliori si hanno applicando una regolarizzazione di tipo Elastic Net con $\lambda = 7071$ e $\alpha = 0,1$ dove l'$RMSE$ è pari a 206111.

K-Nearest Neighbours

Il KNN è il primo modello non-parametrico che si va ad applicare. Poiché le variabili presenti nel dataset sono di diversa natura, è necessario riuscire a trovare una codifica comune. A tal fine, si suppone che le variabili nel dataset siano di natura quantitativa e le si standardizzano centrando i valori attorno allo 0 e apportando opportuni cambiamenti di scala.
Per misurare la dissimilarità tra le osservazioni si va ad utilizzare la distanza euclidea sui dati standardizzati. Dopo l'applicazione del modello i risultati previsivi saranno riportati sulla scala originale.
Il numero ideale di vicini k è ricercato all'interno del range di valori compresi tra 1 e 50. Si è visto che per $k = 6$ viene ottenuto l'errore previsivo minimo. Infatti, come visualizzabile dalla figura 5.11 l'$RMSE$ di cross-validation diminuisce fino in corrispondenza di $k = 6$ per poi aumentare costantemente per scelte di valori di k più elevati. L'$RMSE$ del modello finale è pari a 200389.

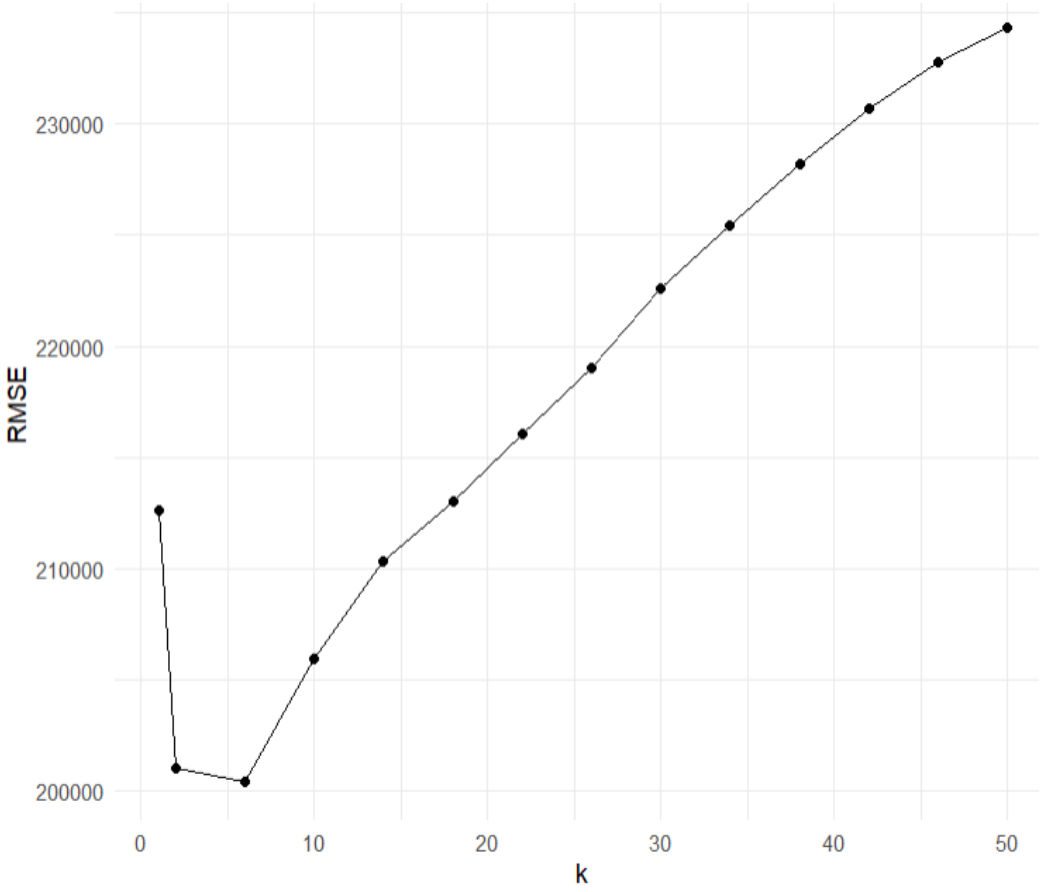


Figura 5.11. Tuning del Modello KNN

MARS

Il modello MARS si può presentare sotto diverse formulazioni. Nelle varie versioni di questo modello possono essere presenti numerosi termini che regolano sia il Forward Step che il Backward Step.
In questa sede si prende in considerazione la versione più semplice del MARS, ovvero quella esposta nella sezione 3.3.2. Sono presenti due iperparametri: il grado del modello e il livello di pruning, ovvero il numero massimo di termini conservati dopo il processo di Backward Step. Si deve quindi eseguire una Grid Search misurando l'errore di previsione risultante dalla combinazione di questi due parametri.
Il grado del modello viene fatto variare tra 1 e 3, mentre il livello di pruning tra 2 e 50. Si è infatti osservato che i valori dei parametri esterni a questi intervalli non portano a risultati predittivi migliori.
Il processo di tuning è rappresentato dal grafico 5.12. Il modello selezionato si trova in corrispondenza di un grado pari a 2 e di un livello di pruning di 30. L'RMSE di cross-validation è pari a 194751.

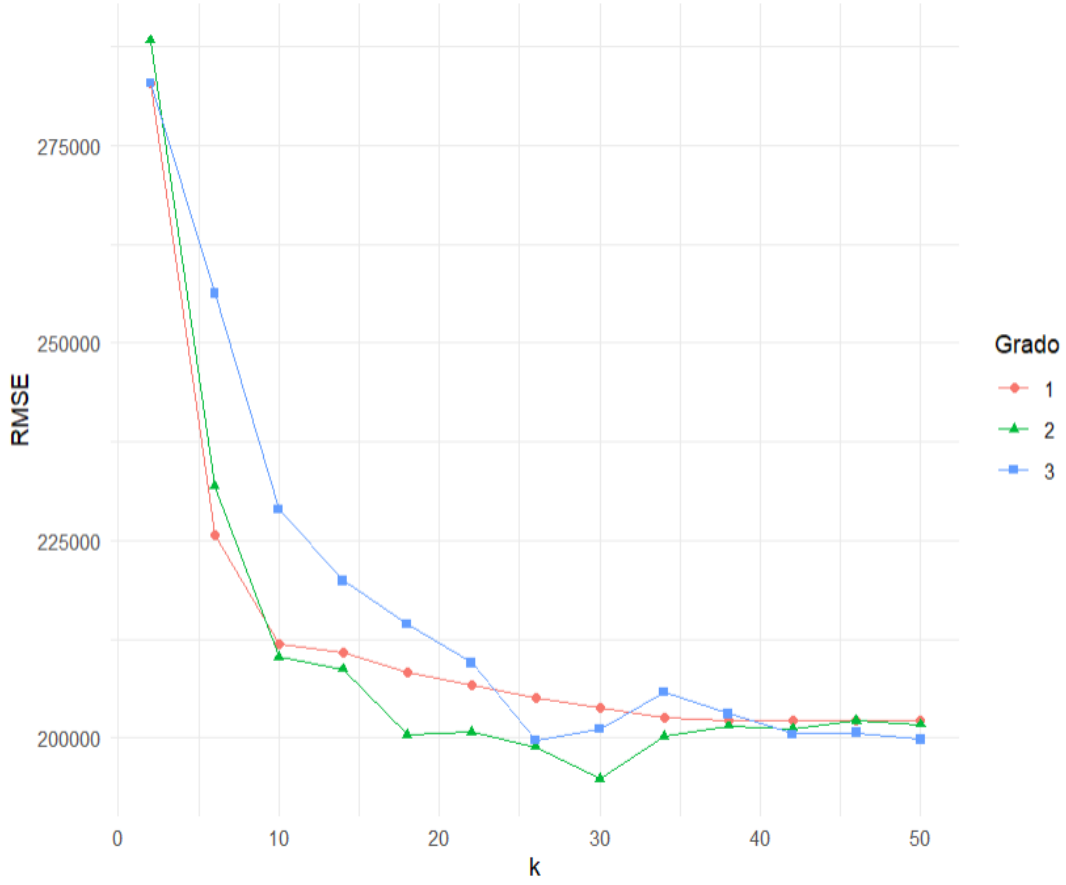


Figura 5.12. Tuning del Modello MARS

Albero Decisionale

Si vuole ora applicare una forma semplice di albero decisionale. Alla versione greedy dell'albero si svolge così il tuning del modello sull'iperparametro di pruning. Questo esprime la profondità massima dell'albero. Il valore dell'iperparametro che restituisce l'$RMSE$ minimo nell'intervallo discreto $[1, 20]$ è pari a 10. Una profondità massima superiore a 10 non induce a capacità predittive migliori.
L'$RMSE$ assume il valore 243439.

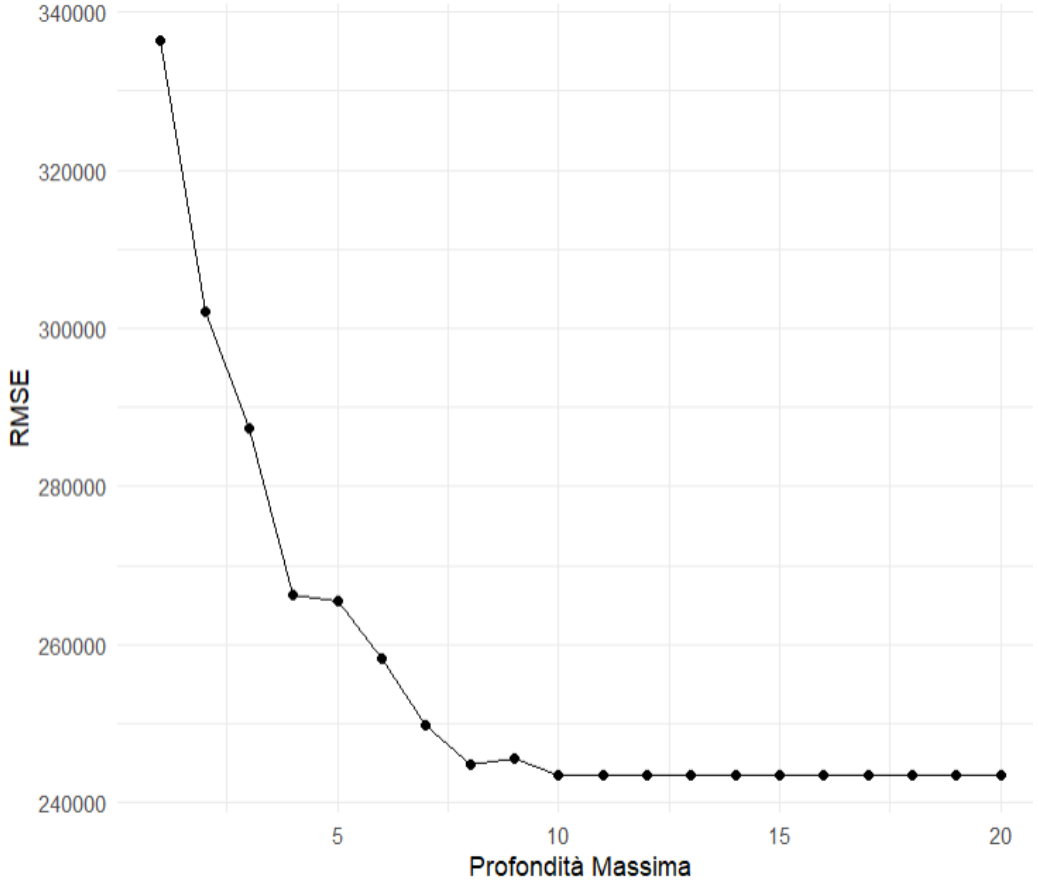


Figura 5.13. Tuning dell'Albero Decisionale

Random Forest

Il modello Random Forest può essere formulato in una moltitudine di modalità. Per semplicità abbiamo utilizzato una versione del modello standard. Si decide di non regolare la profondità massima dell'albero in quanto il Random Forest è poco influenzato dall'overfitting dovuto alla mancanza di tuning di questo iperparametro quando il numero di alberi è sufficientemente elevato.
Il tuning degli iperparametri è stato svolto sequenzialmente.
In primo luogo, fissando gli altri parametri, si stabilisce la regola di split scegliendo tra quella basata sulla minimizzazione della varianza e l'Extra Tree:

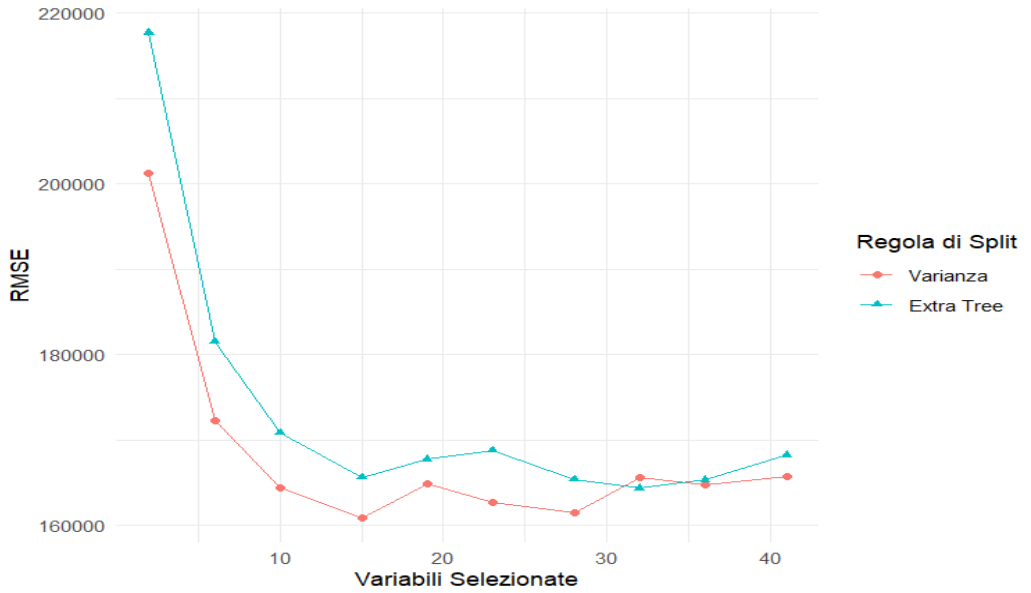


Figura 5.14. Selezione della Regola di Split

Appare evidente che la regola di split basata sulla varianza risulta più efficace. Successivamente, il numero di alberi è stato fissato a 150. Questo valore è la soglia dopo la quale non si ottengono risultati previsivi consistentemente migliori e, oltretutto, è coerente con la numerosità campionaria del dataset.
Per ultimo, si svolge il tuning congiunto degli iperparametri relativi al numero di variabili di split per ciascun nodo e alla dimensione minima di ciascun nodo.

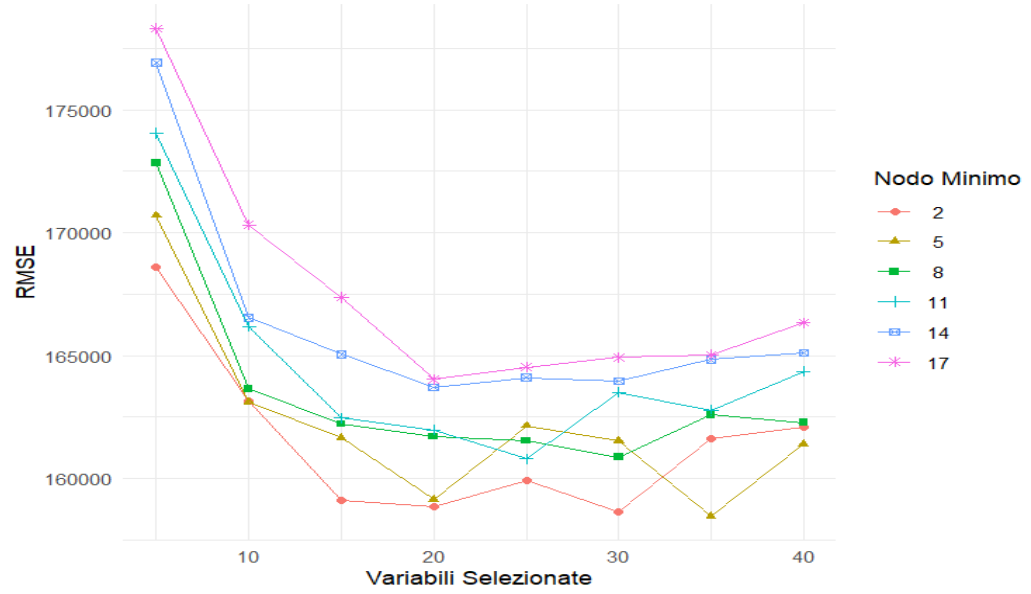


Figura 5.15. Tuning del Modello Random Forest

Il modello finale selezionato ha un numero fissato di variabili di split di 35 e una dimensione minima del nodo di 5. L'$RMSE$ corrispondente è pari a 158453.

XG-Boost

L'XG-Boost è un modello che può presentare numerosi iperparametri. Utilizzare un'attenta procedura di selezione diviene quindi particolarmente importante.

A tal fine si articola il processo di selezione in 5 fasi successive:

1. Profondità Massima: numero massimo di cammini che vanno dal nodo radice alle foglie di ciascun albero. Corrsiponde all'iperparametro di pruning dell'albero decisionale. Rispetto al Random Forest, nell'XG-Boost questo parametro è di maggior rilevanza e controlla l'overfitting del modello.
Poiché la profondità massima è strettamente legata al numero di iterazioni di boosting e al learning rate, si mostrano i risultati facendo variare questi altri due iperparametri. Visualizziamo i risultati graficamente:

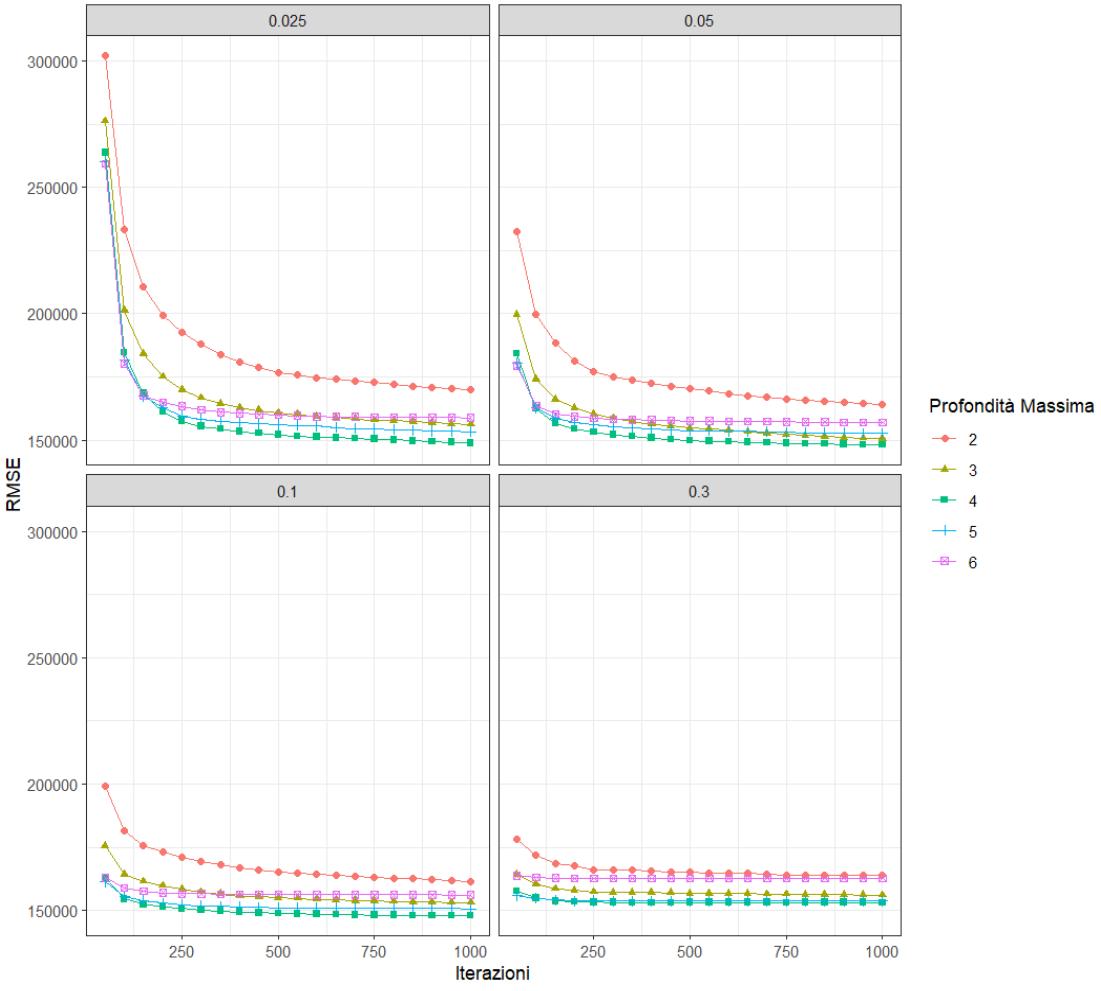


Figura 5.16. Tuning del Modello Random Forest

Ciascun sottografico è associato ad un determinato learning rate. Di tale parametro sono stati presi 4 valori tipicamente utilizzati. Il numero di iterazioni massimo è 1000. Ogni curva di ciascun sottografico è associata ad una profondità massima. Si

può notare che in generale l'$RMSE$ decresce all'aumentare del numero di iterazioni. Prendendo il range di valori discreti $[2, 6]$ si nota che la profondità massima ideale è pari a 4 per ogni valore di learning rate fissato.

2. Peso Minimo del Nodo Figlio: nella regressione si riferisce al numero minimo di istanze (ovvero di osservazioni) richieste per formare un nodo figlio. Si prende in considerazione il range di valori discreti $[2, 6]$:

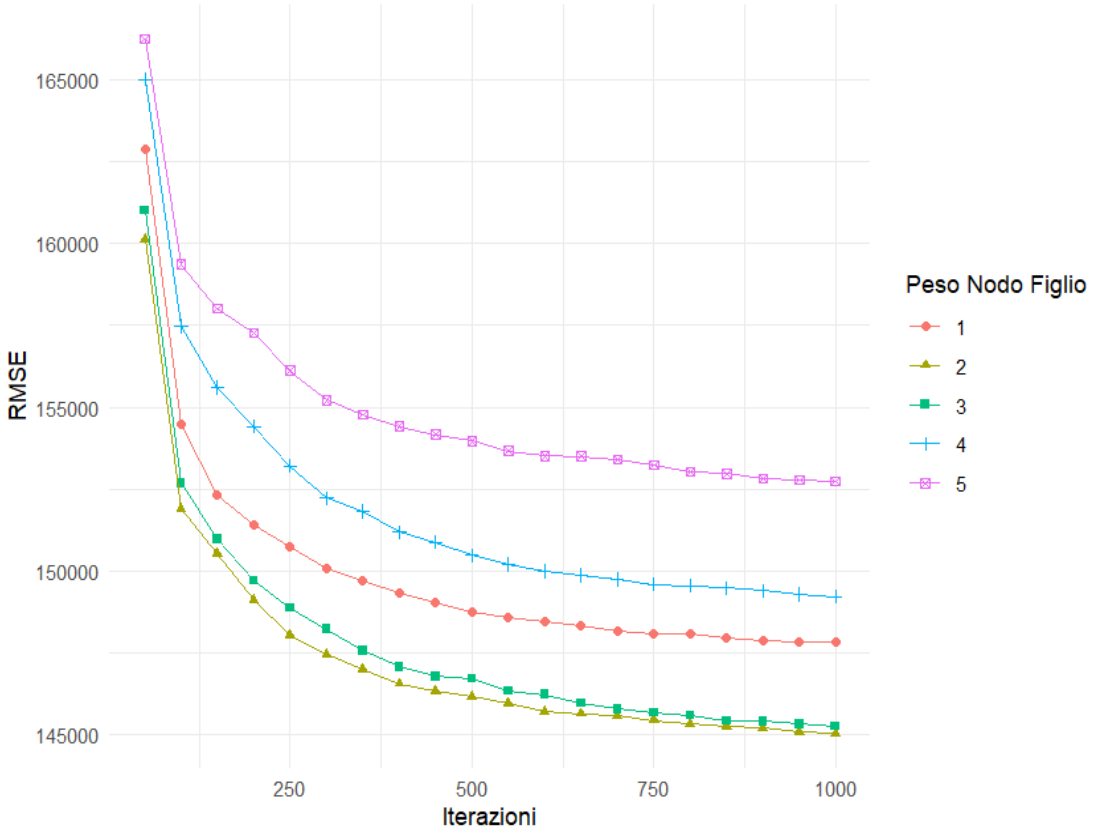


Figura 5.17. Tuning del Modello Random Forest

Si seleziona un peso minimo del nodo figlio pari 2, in quanto l'$RMSE$ associato è consistentemente inferiore agli altri valori dell'iperparametro al variare del numero di iterazioni.

3. Percentuale di Variabili e di Osservazioni di Split: rapporto tra il numero di variabili (o osservazioni) utilizzabili per ciasun split dell'albero e il corrispondente totale. Si prendono in esame tre diversi valori della percentuale di osservazioni di split (50%, 75%, 100%) e si mostrano le curve associate a varie percentuali di variabili di split tra il 40% e il 100%:

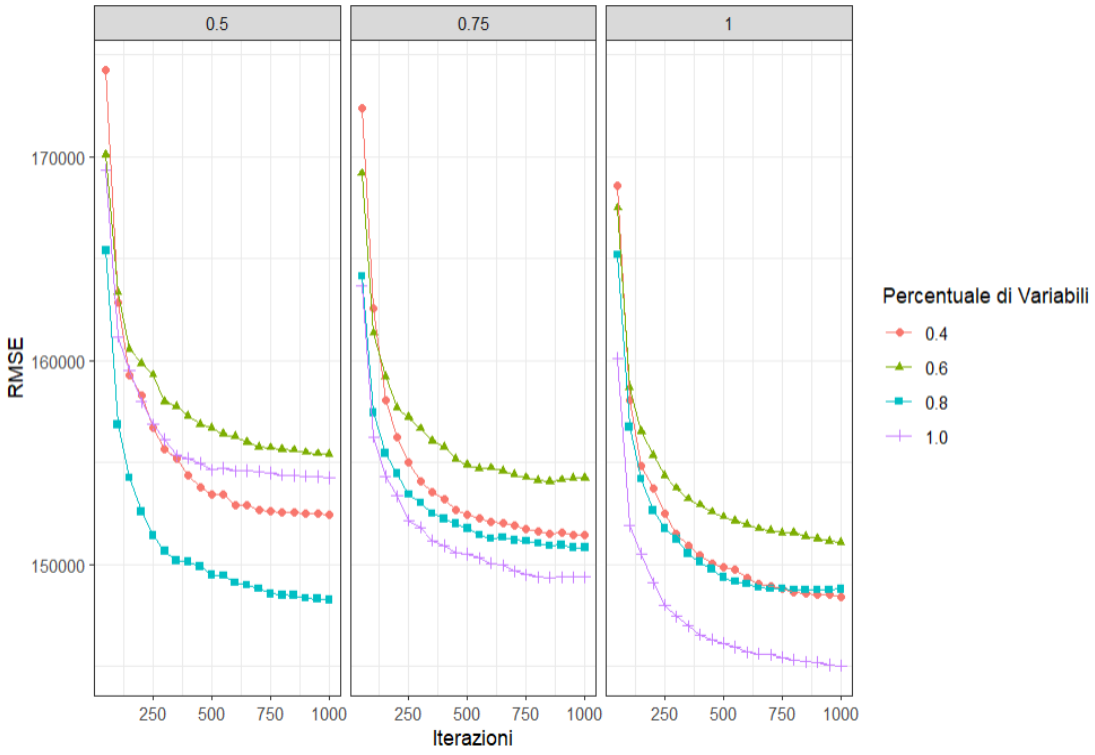


Figura 5.18. Tuning del Modello Random Forest

Non si segnalano miglioramenti diminuendo le due percentuali di split. Per questo motivo si prosegue l'analisi con il modello corrente.

4. Moltiplicatore Lagrangiano: detto anche semplicemente γ, è un parametro di pseudo-regolarizzazione. In contrasto con la profondità massima e il peso minimo del nodo foglia che regolarizzano il modello all'interno dell'albero, γ regolarizza il modello tra gli alberi. In generale, penalizza i coefficienti elevati che non migliorano le performance previsive. Per il dataset in esame, introdurre il Moltiplicatore Lagrangiano non induce a risultati predittivi migliori.

5. Learning Rate: indicato con η, rappresenta la velocità con cui il modello "impara" dai dati. Come esposto nella formula (3.19) è l'iperparametro tipico dei modelli basati sull'algoritmo della discesa del gradiente. Nonostante η sia stato precedentemente fissato al passo 1, la modifica degli altri iperparametri potrebbe implicare una necessaria ricalibrazione del learning rate. Si svolge quindi il tuning di η facendolo variare tra 0,01 e 0,1. In questo passo finale si fa variare il numero di iterazioni fino ad un massimo di 10000.

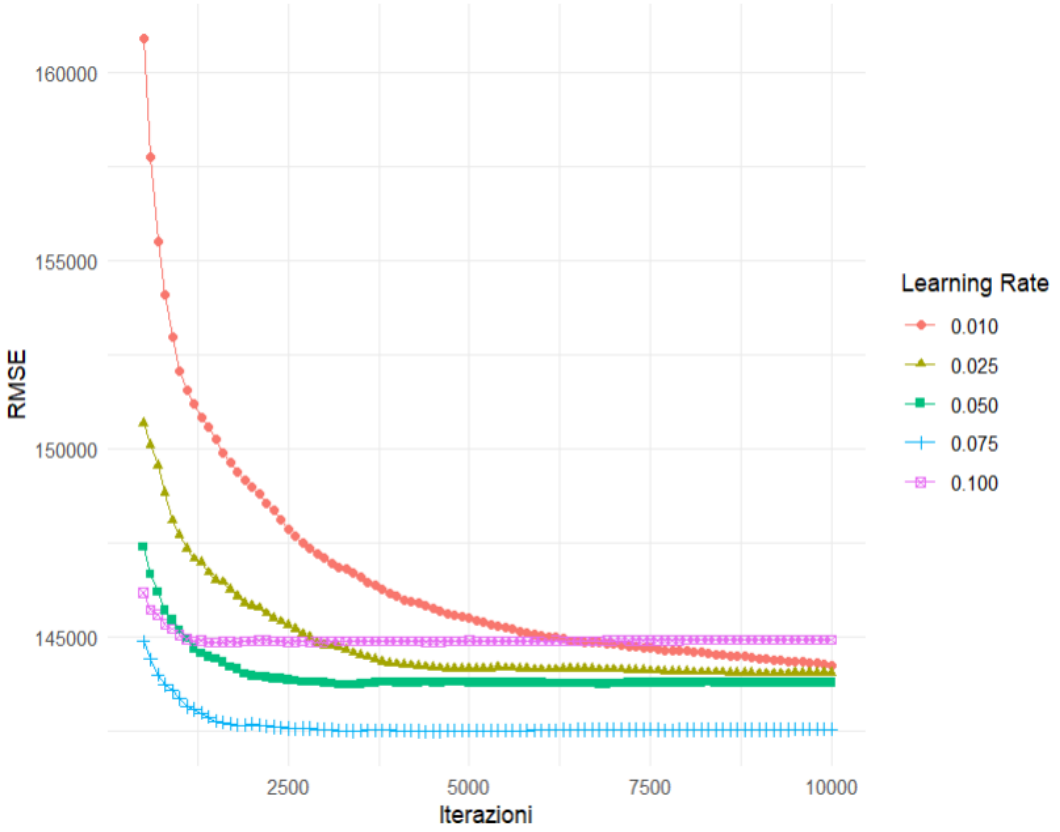


Figura 5.19. Tuning del Modello Random Forest

Il modello finale è quello trovato con 4400 iterazioni boostrap e un learning rate di 0,075. L'$RMSE$ al termine del processo di selezione è pari a 142482.

5.4.2 Analisi Spaziale

A partire dalla matrice dei pesi spaziali ricavata durante la fase di analisi esplorativa, si procede applicando i passaggi descritti a livello teorico nella sezione 4.6.2.
La presenza di autocorrelazione spaziale nei dati è già stata verificata nell'ESDA. Si prende allora come punto di inizializzaze il modello regressivo lineare ridotto i cui valori dei coefficienti di regressione sono nella tabella 5.5. Viene quindi mantenuto il processo di selezione delle variabili. Sui residui di tale modello viene misurato l'indice di Moran. Il valore dell'indice osservato è di 0,422 e il p-value è estremamente significativo. Rispetto al modello con sola intercetta, il quale mostrava un indice pari a 0,561, l'autocorrelazione spaziale non descritta dal modello è più leggera, ma comunque fortemente presente.
Conseguentemente, si svolgono i Test dei Moltiplicatori di Lagrange per verificare la presenza di lag spaziale e di errore spaziale. In entrambi i casi le statistiche test indicano la presenza del rispettivo effetto spaziale. Risultati analoghi si osservano con le controparti robuste degli stessi test. Poiché la statistica test assume un valore maggiore (e quindi più significativo) in corrispondenza dell'errore spaziale rispetto al lag spaziale (1021 contro 263), si opta per l'utilizzo del modello SEM.

Andando ad applicare nuovamente un test di Moran sui residui del modello corrente (in questo caso il modello di errore spaziale), questo presenta un valore del p-value pari a 0,337. Il test non risulta quindi significativo ad un livello soglia del 5% e si decide di non proseguire oltre con la ricerca di modelli più complessi.
L'analisi diagnostica standard del modello è riportata in appendice. I risultati sono molto simili a quanto già analizzato per il modello regressivo lineare. Le diagnostiche spaziali coincidono invece con il processo di selezione del modello che è stato applicato.

I valori dei coefficienti del modello sono riportati nella tabella seguente:

Parametro	Stima	Errore Standard
Intercetta	-137001	17468
Superficie Costruita	3083	68
Numero di Bagni	70111	5196
Numero di Stanze	13076	3097
Attico	120567	10528
Casa Indipendente	-143711	19390
Esterno	13595	9952
Nuova Costruzione	137328	13253
Aria Condizionata	22935	6119
Ascensore	22681	8254
Giardino	4414	7676
Riscaldamento Autonomo	16279	7521

Tabella 5.9. Modello di Errore Spaziale SEM

Il parametro di errore spaziale associato al modello è pari a $\lambda = 0,74$.
I valori dei coefficienti di regressione sono distinti da quelli prodotti dal modello regressivo lineare. Molti di essi cambiano infatti di segno con l'introduzione del parametro di errore spaziale. In questo modello, il parametro λ è una variabile impiegata come esplicativa aggiunta al modello, in modo da poter tenere conto in modo appropriato del clustering spaziale rilevato dal test I di Moran. Il coefficiente stimato per questo termine è positivo e statisticamente significativo secondo il Test del Rapporto di Verosimiglianza. In altre parole, fissando le altre variabili, in media il prezzo degli immobili varia in modo direttamente proporzionale in aree circoscritte. L'errore previsivo espresso in termini di $RMSE$ di cross-validation è pari a 242303. Le trasformazioni della risposta, l'introduzione di ulteriori esplicative e il cambiamento della struttura del modello non portano a risultati predittivi preferibili.

5.5 Confronto tra Modelli

Avendo utilizzato la stessa metrica di costruzione, si possono mettere a confronto tutti i modelli applicati. I risultati sono riportati nella tabella sottostante:

Modello	Train Set		Cross-Validation		Test Set	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Modello Lineare	202158	109172	208429	115426	181631	104805
Modello Log-Lineare	178699	85775	189767	87296	166736	85520
GLM (Gaussiano)	154540	86978	165623	90193	155223	84706
KNN	165629	74384	200389	90493	177983	90260
MARS	151783	86082	194751	98120	156170	88701
Albero Decisionale	218283	124475	243439	129806	220336	127196
Random Forest	66186	27364	160546	70152	136943	64999
XG-Boost	7368	4730	142482	63478	121206	58188
Modello Spaziale (SEM)	182558	90826	242303	143714	232234	147465

Tabella 5.10. Confronto tra Modelli Previsivi

Questa tabella compara i valori dell'$RMSE$ e dell'MAE dei vari modelli per le tre tipologie di set di dati (training set, cross-validation, test set).
Si può da subito notare che, ad esclusione del modello spaziale, si ottengono risultati migliori col test set rispetto alla cross-validation e, talvolta, anche rispetto al training set. Si possono avere fondamentalmente due spiegazioni. La prima possibilità è che il test set sia particolarmente favorevole e in linea con i modelli applicati. Alternativamente, al test set non è stato assegnato un numero sufficiente di osservazioni per avere delle stime degli errori di previsione accurate. Poiché l'obiettivo principale non è quello di generalizzare l'entità dell'errore di previsione, non si procede con l'applicazione di un diverso dataset splitting.
I modelli che offrono le migliori prestazioni sono, in ordine, l'XG-Boost, il Random Forest e il GLM gaussiano con funzione legame logaritmica. I primi due modelli sembrano andare in overfitting. Apportando delle dovute regolarizzazioni è possibile diminuire l'eccessivo adattamento al modello a discapito di un leggero peggioramento in termini di errore previsivo di cross-validation.
Si nota che il modello spaziale selezionato mostra degli scarsi risultati previsivi. Discreto è invece il suo adattamento ai dati osservati. A parità di costruzione (trasformazioni e numero di variabili) il modello SEM si adatta maggiormente ai dati rispetto al corrispettivo modello regressivo. E' possibile migliorare notevolmente l'adattamento del modello spaziale, ma le previsioni non risultano più promettenti.

Capitolo 6

Conclusione

In questa tesi si sono utilizzate diverse tecniche di imputazione dei dati mancanti e si è messo a confronto un approccio non spaziale con uno spaziale.
I nostri risultati hanno dimostrato che il metodo di imputazione basato sul Random Forest è il preferibile sul dataset di Madrid secondo gli I-Scores. Per quanto riguarda la regressione, sebbene tutti i modelli fossero in grado di svolgere le previsioni con una certa accuratezza, il modello XG-Boost ha mostrato la migliore performance tra tutti i modelli esaminati.
In particolare, l'XG-Boost è stato in grado di prevedere i prezzi immobiliari sbagliando in media di 63.478 €, corrispondente ad un $RMSE$ di 142482, ovvero la metrica con la quale sono stati sviluppati i modelli. L'errore previsivo deve essere messo in relazione con il prezzo medio degli immobili pari a 460.105 €. In termini di accuratezza percentuale (R^2), il modello spiega l'89,9% della variazione nella variabile dipendente. Questo dimostra la sua capacità di apprendere a fondo le informazioni contenute nei dati e di effettuare previsioni precise.
Tuttavia, va notato che ogni modello ha i suoi vantaggi e limitazioni, e che la scelta del modello migliore dipende dal dataset, dall'obiettivo dell'analisi e, nel caso delle previsioni, dall'indicatore di errore di riferimento.
L'approccio spaziale non ha restituito dei buoni risultati e si è dimostrato affidabile soltanto nel caso in cui si voglia ottenere un modello descrittivo piuttosto che uno previsivo. In questa casistica, infatti, il modello di errore spaziale SEM si è adattato maggiormente ai dati osservati rispetto ai metodi basati sulla regressione lineare.
In definitiva, è quindi in genere preferibile optare per un modello tradizionale basato sugli alberi, o, altrimenti, su un modello GLM per bilanciare interpretabilità a capacità previsive.

6.1 Critiche e Suggerimenti

Questa analisi presenta alcuni aspetti critici che potrebbero essere rifiniti con ulteriori ricerche e approfondimenti. In primo luogo, nella parte applicativa si potrebbero impiegare tutti i metodi di imputazione presentati nella parte teorica e confrontarli con i dovuti criteri. Inoltre, si potrebbe approfondire l'utilizzo dell'imputazione multipla nei modelli di regressione parametrica.
In secondo luogo, è possibile migliorare i risultati previsivi dei modelli utilizzando dei

metodi di selezione basati sulla k-fold cross-validation ripetuta o stratificata al fine di ridurre la varianza delle stime. In aggiunta, si può decidere di cambiare lo stimatore per misurare l'errore di previsione sostituendo l'$RMSE$ con altri indicatori.
Un altro aspetto che può essere trattato in maniera differente è la componente spaziale. Nei modelli tradizionali sono introducibili numerose altre variabili di distanza oltre a quelle indicate ed è per di più possibile trasformarle in dummy in modo tale che possano essere più facilmente interpretabili e gestibili. D'altra parte, i modelli spaziali utilizzabili sono molteplici e in questa sede è stata esposta soltanto una delle tante categorie possibili.
In conclusione, si auspica che questa ricerca possa essere una risorsa utile e di riferimento per tutti coloro che vogliano cimentarsi nel campo delle previsioni immobiliari in futuro.

6.2 Appendice

Riportiamo in appendice il codice sorgente scritto nel linguaggio R.

Abbiamo diviso l'analisi in 5 diverse fasi:
1. Pre-Processing (parte prima): gestione delle variabili e geocodifica degli indirizzi
2. Pre-Processing (parte seconda): gestione delle variabili di distanza
3. Pre-Processing (parte terza): gestione dei dati mancanti
4. Analisi Esplorativa
5. Regressione

Pre-Processing 1

```
# Caricamento delle Librerie

library(readr)
library(stringr)
library(Hmisc)
library(tidygeocoder)
library(writexl)
```

```
data=read_csv('houses_Madrid.csv')
data=as.data.frame(data)
data=subset(data,!is.na(data$street_number))
#View(data)
```

```
data=subset(data,select=c(buy_price,buy_price_by_area,rent_price,parking_price,sq_mt_built,
                          sq_mt_useful,sq_mt_allotment,built_year,n_bathrooms,n_floors,n_rooms,
                          energy_certificate,floor,house_type_id,title,subtitle,raw_address,
                          street_name,street_number,neighborhood_id,is_accessible,is_exterior,
                          is_exact_address_hidden,is_renewal_needed,is_new_development,
                          is_parking_included_in_price,is_floor_under,is_orientation_north,
                          is_orientation_south,is_orientation_east,is_orientation_west,
                          has_ac,has_fitted_wardrobes,has_lift,has_balcony,has_garden,
                          has_parking,has_pool,has_storage_room,has_individual_heating,
                          has_central_heating,has_terrace,has_green_zones))

data=unique(data)  # eliminazione dei duplicati
data=data[-c(5511,5989,6180),]  # eliminazione delle osservazioni esterne all'area di Madrid
rownames(data)=1:nrow(data)

dim(data)
```

```
## [1] 6287   43
```

```
summary(data)  # statistiche descrittive di sintesi delle variabili
```

```
##    buy_price       buy_price_by_area   rent_price        parking_price
##  Min.   :  42000   Min.   :  688     Min.   :-17691896   Min.   :     0
##  1st Qu.: 200000   1st Qu.: 2550     1st Qu.:      868   1st Qu.:     0
##  Median : 320000   Median : 3600     Median :     1182   Median :     0
##  Mean   : 457566   Mean   : 3881     Mean   :    -6016   Mean   :  2477
##  3rd Qu.: 560000   3rd Qu.: 4785     3rd Qu.:     1664   3rd Qu.:     0
##  Max.   :7525000   Max.   :18462     Max.   :     2517   Max.   :380000
##                                                         NA's   :3897
##   sq_mt_built     sq_mt_useful    sq_mt_allotment   built_year    n_bathrooms
##  Min.   : 20.0   Min.   :  1.00   Min.   :  1.0   Min.   :1850   Min.   : 1.0
```

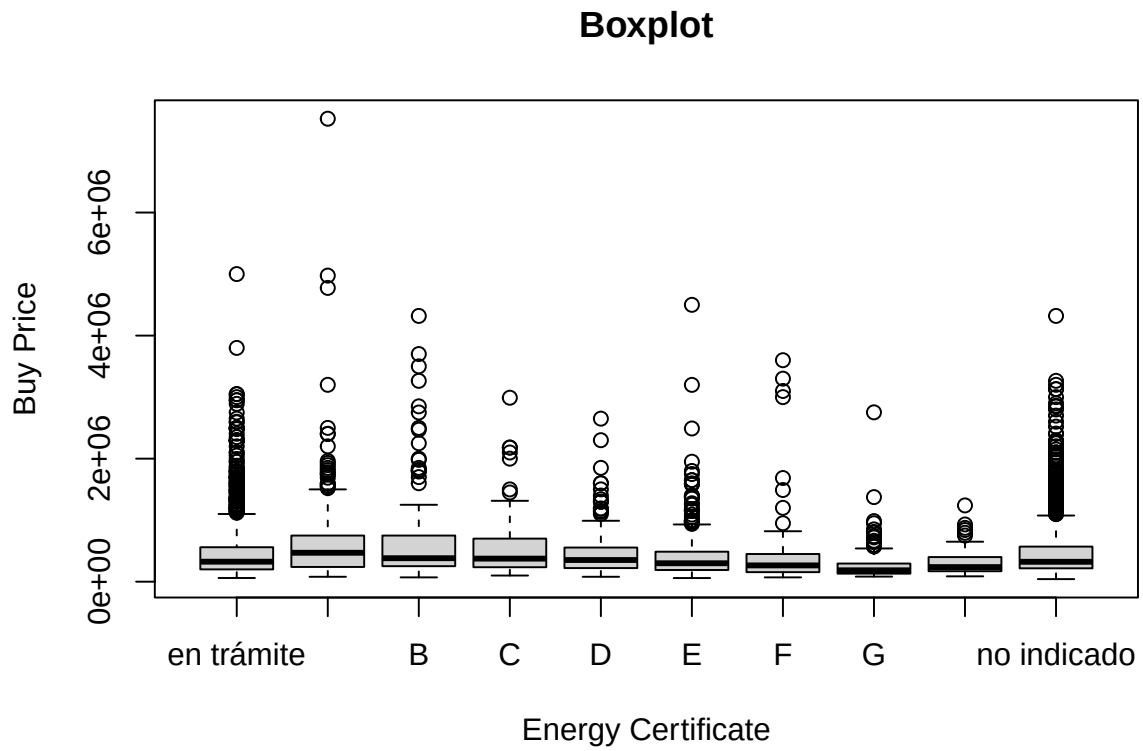
```
##  1st Qu.: 69.0   1st Qu.: 59.00   1st Qu.: 97.5   1st Qu.:1960   1st Qu.: 1.0
##  Median : 93.0   Median : 78.00   Median :250.0   Median :1975   Median : 2.0
##  Mean   :114.6   Mean   : 91.91   Mean   :277.2   Mean   :1976   Mean   : 1.8
##  3rd Qu.:132.0   3rd Qu.:106.00   3rd Qu.:360.0   3rd Qu.:2003   3rd Qu.: 2.0
##  Max.   :920.0   Max.   :750.00   Max.   :994.0   Max.   :2022   Max.   :11.0
##  NA's   :9       NA's   :3658     NA's   :6115    NA's   :4356   NA's   :5
##     n_floors        n_rooms       energy_certificate    floor
##  Min.   :1.000   Min.   : 0.000   Length:6287        Length:6287
##  1st Qu.:2.000   1st Qu.: 2.000   Class :character   Class :character
##  Median :3.000   Median : 3.000   Mode  :character   Mode  :character
##  Mean   :3.095   Mean   : 2.657
##  3rd Qu.:4.000   3rd Qu.: 3.000
##  Max.   :5.000   Max.   :24.000
##  NA's   :6025
##  house_type_id         title             subtitle         raw_address
##  Length:6287        Length:6287        Length:6287        Length:6287
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  street_name        street_number      neighborhood_id    is_accessible
##  Length:6287        Length:6287        Length:6287        Mode:logical
##  Class :character   Class :character   Class :character   TRUE:1186
##  Mode  :character   Mode  :character   Mode  :character   NA's:5101
##
##
##
##
##  is_exterior     is_exact_address_hidden is_renewal_needed is_new_development
##  Mode :logical   Mode :logical           Mode :logical     Mode :logical
##  FALSE:577       FALSE:6287              FALSE:5458        FALSE:4999
##  TRUE :5210                              TRUE :829         TRUE :1142
##  NA's :500                                                 NA's :146
##
##
##
##  is_parking_included_in_price is_floor_under  is_orientation_north
##  Mode :logical                Mode :logical   Mode :logical
##  FALSE:208                    FALSE:5175      FALSE:2283
##  TRUE :2182                   TRUE :838       TRUE :758
##  NA's :3897                   NA's :274       NA's :3246
##
##
##
##  is_orientation_south is_orientation_east is_orientation_west  has_ac
##  Mode :logical        Mode :logical       Mode :logical       Mode:logical
##  FALSE:1572           FALSE:1810          FALSE:2028          TRUE:3033
##  TRUE :1469           TRUE :1231          TRUE :1013          NA's:3254
##  NA's :3246           NA's :3246          NA's :3246
##
##
##
```

```
##  has_fitted_wardrobes  has_lift       has_balcony    has_garden
##  Mode:logical         Mode :logical   Mode:logical   Mode:logical
##  TRUE:3810            FALSE:1311      TRUE:1000      TRUE:215
##  NA's:2477            TRUE :4640      NA's:5287      NA's:6072
##                       NA's :336
##
##
##
##  has_parking     has_pool       has_storage_room has_individual_heating
##  Mode :logical   Mode:logical   Mode:logical     Mode :logical
##  FALSE:3897      TRUE:1877      TRUE:2431        FALSE:857
##  TRUE :2390      NA's:4410      NA's:3856        TRUE :2381
##                                                  NA's :3049
##
##
##
##  has_central_heating has_terrace    has_green_zones
##  Mode :logical       Mode:logical   Mode:logical
##  FALSE:2381          TRUE:2905      TRUE:1692
##  TRUE :857           NA's:3382      NA's:4595
##  NA's :3049
##
##
##
```

```
# Gestione delle Variabili

# 1) gestione delle variabili di prezzo
# 2) trasformazione delle variabili categoriali
# 3) gestione delle variabili di posizione
# 4) eliminazione delle variabili superflue


## 1

for(i in 1:nrow(data)){  # aggiunta del prezzo del parcheggio a quello di vendita
  if(!is.na(data$is_parking_included_in_price[i])&data$is_parking_included_in_price[i]==FALSE){
    data$buy_price[i]=data$buy_price[i]+data$parking_price[i]
  }
}


## 2

### Certificazione Energetica

energy_certificate_factor=factor(data$energy_certificate)
energy_certificate_factor=relevel(energy_certificate_factor,ref="en trámite")
summary(energy_certificate_factor)
```

```
##      en trámite               A               B               C               D
##            2266             347             224             177             268
##               E               F               G inmueble exento     no indicado
##             640             134             161              56            2014
```

```
boxplot(data$buy_price~energy_certificate_factor,main='Boxplot',
        xlab='Energy Certificate',ylab='Buy Price')
```




```
summary(lm(data$buy_price~energy_certificate_factor))  # non significatività delle dummy
```

```
##
## Call:
## lm(formula = data$buy_price ~ energy_certificate_factor)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -552447 -245328 -125228  104772 6892553
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                                445228       9064  49.119  < 2e-16
## energy_certificate_factorA                 187219      24874   7.527 5.93e-14
## energy_certificate_factorB                 157938      30221   5.226 1.79e-07
## energy_certificate_factorC                  81642      33675   2.424   0.0154
## energy_certificate_factorD                   8585      27872   0.308   0.7581
## energy_certificate_factorE                 -44653      19315  -2.312   0.0208
## energy_certificate_factorF                 -35968      38361  -0.938   0.3485
## energy_certificate_factorG                -171207      35193  -4.865 1.17e-06
## energy_certificate_factorinmueble exento  -107133      58368  -1.835   0.0665
## energy_certificate_factorno indicado        16563      13214   1.253   0.2101
```

```
##
## (Intercept)                                 ***
## energy_certificate_factorA                  ***
## energy_certificate_factorB                  ***
## energy_certificate_factorC                  *
## energy_certificate_factorD
## energy_certificate_factorE                  *
## energy_certificate_factorF
## energy_certificate_factorG                  ***
## energy_certificate_factorinmueble exento .
## energy_certificate_factorno indicado
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 431500 on 6277 degrees of freedom
## Multiple R-squared:  0.02111,    Adjusted R-squared:  0.01971
## F-statistic: 15.04 on 9 and 6277 DF,  p-value: < 2.2e-16
```

```
summary(lm(data$buy_price~energy_certificate_factor+data$sq_mt_built))
```

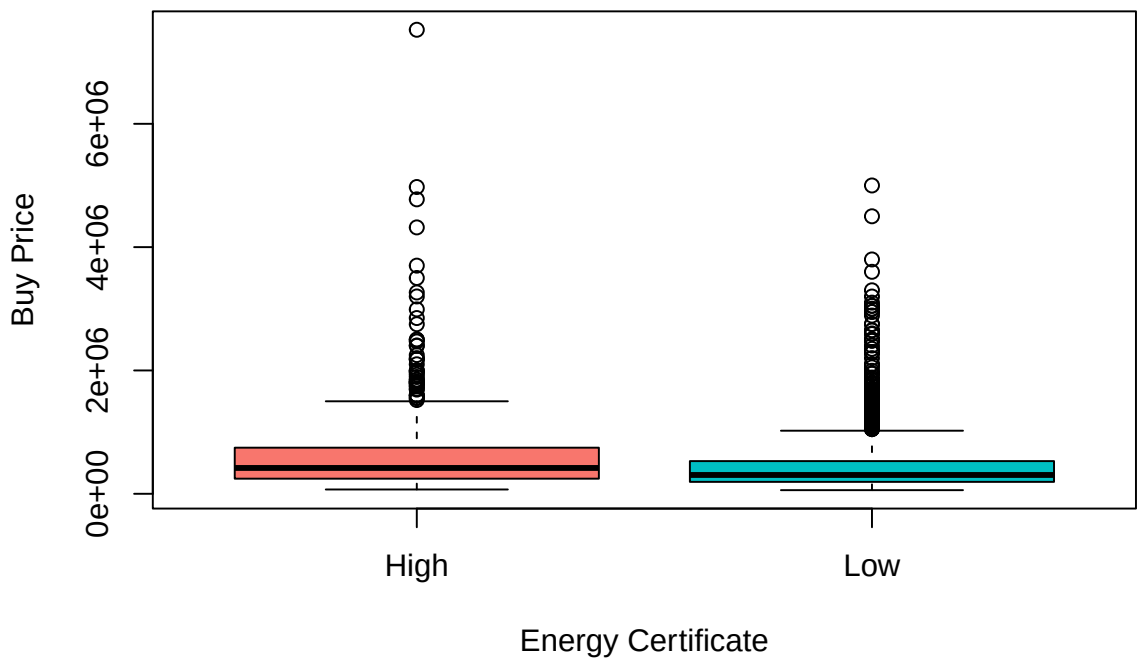
```
##
## Call:
## lm(formula = data$buy_price ~ energy_certificate_factor + data$sq_mt_built)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1848864  -119024   -24185    86108  4922877
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              -22181.06    7080.55  -3.133  0.00174
## energy_certificate_factorA               109079.06   14870.78   7.335 2.50e-13
## energy_certificate_factorB               112280.21   18032.07   6.227 5.07e-10
## energy_certificate_factorC                54312.01   20087.74   2.704  0.00687
## energy_certificate_factorD                 1724.33   16625.05   0.104  0.91740
## energy_certificate_factorE               -19716.15   11532.07  -1.710  0.08737
## energy_certificate_factorF               -25287.65   22962.65  -1.101  0.27083
## energy_certificate_factorG               -28122.42   21107.03  -1.332  0.18279
## energy_certificate_factorinmueble exento -23751.61   34820.52  -0.682  0.49519
## energy_certificate_factorno indicado       7775.11    7885.92   0.986  0.32420
## data$sq_mt_built                           4069.94      40.07 101.561  < 2e-16
##
## (Intercept)                              **
## energy_certificate_factorA               ***
## energy_certificate_factorB               ***
## energy_certificate_factorC               **
## energy_certificate_factorD
## energy_certificate_factorE               .
## energy_certificate_factorF
## energy_certificate_factorG
## energy_certificate_factorinmueble exento
## energy_certificate_factorno indicado
## data$sq_mt_built                         ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 257300 on 6267 degrees of freedom
##   (9 osservazioni eliminate a causa di valori mancanti)
## Multiple R-squared:  0.6307, Adjusted R-squared:  0.6301
## F-statistic:  1070 on 10 and 6267 DF,  p-value: < 2.2e-16
```

```
# -> trasformazione della variabile

data$energy_certificate[data$energy_certificate=='A'|data$energy_certificate=='B'|
                        data$energy_certificate=='C']='High'
data$energy_certificate[data$energy_certificate=='D'|data$energy_certificate=='E'|
                        data$energy_certificate=='F'|data$energy_certificate=='G'|
                        data$energy_certificate=='inmueble exento'|
                        data$energy_certificate=='en trámite']='Low'
data$energy_certificate[data$energy_certificate=='no indicado']=NA
summary(as.factor(data$energy_certificate))
```

```
## High  Low NA's
##  748 3525 2014
```

```
boxplot(data$buy_price~as.factor(data$energy_certificate),
        xlab='Energy Certificate',ylab='Buy Price',col=c('#F8766D','#00BFC4'))
```



```
summary(lm(buy_price~as.factor(energy_certificate)+sq_mt_built,data=data))  # variabile significativa
```

```
##
## Call:
## lm(formula = buy_price ~ as.factor(energy_certificate) + sq_mt_built,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1838381  -114862   -19448    85041  4933302
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         74467.94   11089.17   6.715 2.12e-11 ***
## as.factor(energy_certificate)Low -103070.93   10295.93 -10.011  < 2e-16 ***
## sq_mt_built                         4073.19      47.17  86.348  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 254800 on 4262 degrees of freedom
##   (2022 osservazioni eliminate a causa di valori mancanti)
## Multiple R-squared:  0.6447, Adjusted R-squared:  0.6445
## F-statistic:  3867 on 2 and 4262 DF,  p-value: < 2.2e-16
```

```
### Piano

summary(as.factor(data$floor))
```

```
##                   1                   2                   3
##                1282                1017                 914
##                   4                   5                   6
##                 699                 405                 269
##                   7                   8                   9
##                 171                 106                  50
##               Bajo Entreplanta exterior Entreplanta interior
##                 693                  86                  13
## Semi-sótano exterior Semi-sótano interior              Sótano
##                   19                  17                   2
##      Sótano exterior      Sótano interior                NA's
##                    2                   6                 536
```

```
data$floor[data$floor=='Entreplanta'|data$floor=='Entreplanta exterior'|
             data$floor=='Entreplanta interior']='0.5'
data$floor[data$floor=='Bajo']='0'
data$floor[data$floor=='Semi-sótano'|data$floor=='Semi-sótano exterior'|
             data$floor=='Semi-sótano interior']='-0.5'
data$floor[data$floor=='Sótano'|data$floor=='Sótano exterior'|
             data$floor=='Sótano interior']='-1'
data$floor=as.numeric(data$floor)
summary(data$floor)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   -1.00    1.00    2.00    2.61    4.00    9.00     536
```

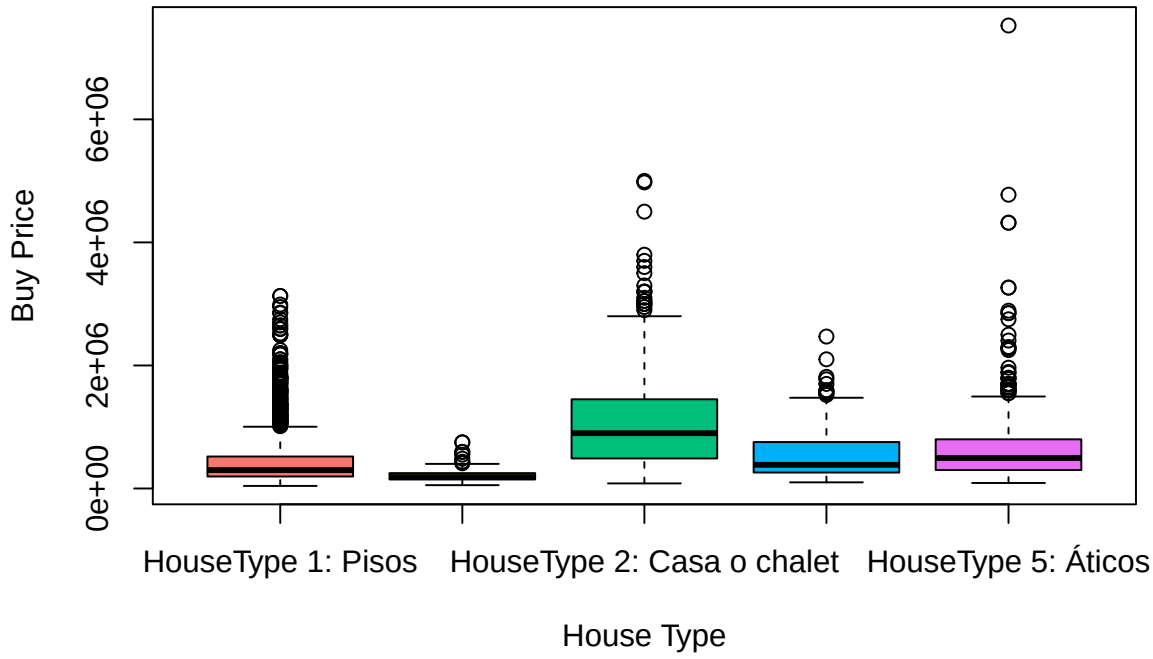
```
### Tipo di Casa

data_house_type=str_split_fixed(data$title,' ',2)[,1]
summary(as.factor(data_house_type))
```

```
##   Ático    Casa  Chalet  Dúplex Estudio    Piso
##     391     114     191     235     166    5190
```

```
for(i in 1:nrow(data)){
  if(is.na(data$house_type_id[i])){
    data$house_type_id[i]=data_house_type[i]
  }
}

house_type_factor=factor(data$house_type_id)
house_type_factor=relevel(house_type_factor,ref='HouseType 1: Pisos')
summary(house_type_factor)
```

```
##         HouseType 1: Pisos                    Estudio
##                       5190                        166
## HouseType 2: Casa o chalet       HouseType 4: Dúplex
##                        305                        235
##        HouseType 5: Áticos
##                        391
```

```
boxplot(data$buy_price~house_type_factor,xlab='House Type',ylab='Buy Price',
        col=c('#F8766D','#A3A500','#00BF7D','#00B0F6','#E76BF3'))
```




```
summary(lm(data$buy_price~house_type_factor))  # dummy significative
```

```
##
## Call:
## lm(formula = data$buy_price ~ house_type_factor)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1049505  -224818  -105818   115182  6829860
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                                404818       5557  72.851
## house_type_factorEstudio                  -189765      31564  -6.012
## house_type_factorHouseType 2: Casa o chalet  727686      23586  30.852
## house_type_factorHouseType 4: Dúplex         142927      26699   5.353
## house_type_factorHouseType 5: Áticos         290321      20994  13.829
##                                          Pr(>|t|)
## (Intercept)                               < 2e-16 ***
## house_type_factorEstudio                 1.93e-09 ***
## house_type_factorHouseType 2: Casa o chalet  < 2e-16 ***
## house_type_factorHouseType 4: Dúplex        8.94e-08 ***
## house_type_factorHouseType 5: Áticos         < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 400300 on 6282 degrees of freedom
## Multiple R-squared:  0.1567, Adjusted R-squared:  0.1562
## F-statistic: 291.9 on 4 and 6282 DF,  p-value: < 2.2e-16
```

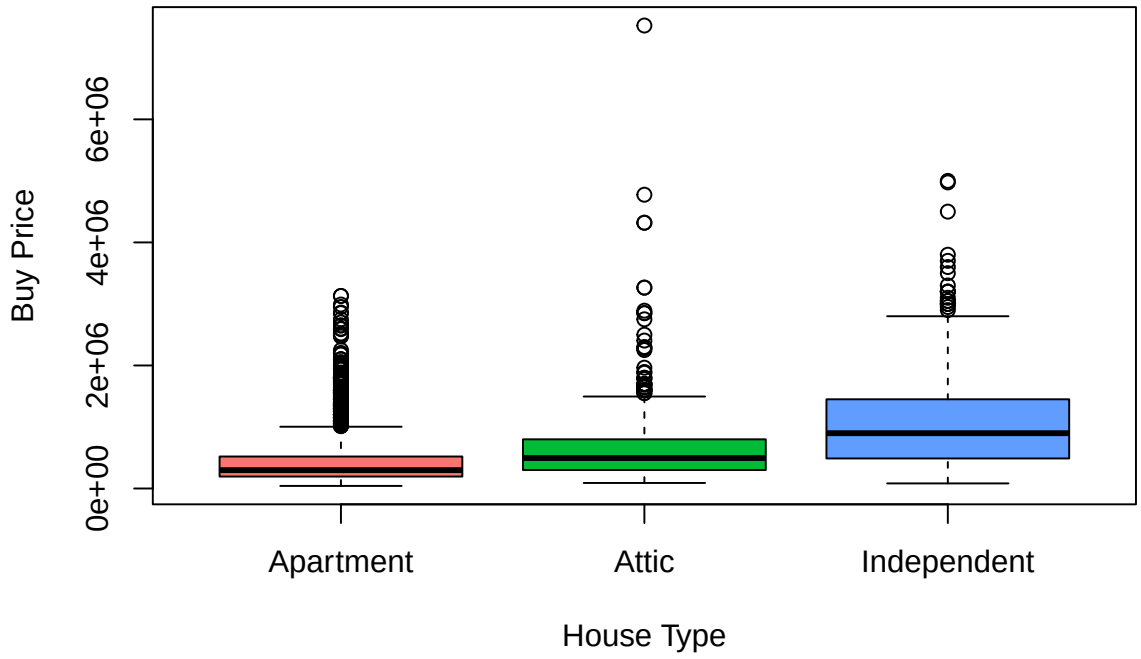
```
summary(lm(data$buy_price~house_type_factor+data$sq_mt_built))  # dummy non più significative
```

```
##
## Call:
## lm(formula = data$buy_price ~ house_type_factor + data$sq_mt_built)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1921939  -116494   -27158    84891  4545762
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                -74467.38    5971.51 -12.470
## house_type_factorEstudio                    21458.70   19551.14   1.098
## house_type_factorHouseType 2: Casa o chalet -409824.00   18389.18 -22.286
## house_type_factorHouseType 4: Dúplex        -20294.27   16520.92  -1.228
## house_type_factorHouseType 5: Áticos        141187.85   13013.93  10.849
## data$sq_mt_built                              4712.81      48.13  97.924
##                                            Pr(>|t|)
## (Intercept)                                  <2e-16 ***
## house_type_factorEstudio                      0.272
## house_type_factorHouseType 2: Casa o chalet  <2e-16 ***
## house_type_factorHouseType 4: Dúplex          0.219
## house_type_factorHouseType 5: Áticos         <2e-16 ***
## data$sq_mt_built                             <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 246500 on 6272 degrees of freedom
##   (9 osservazioni eliminate a causa di valori mancanti)
## Multiple R-squared:  0.661,  Adjusted R-squared:  0.6607
## F-statistic:  2446 on 5 and 6272 DF,  p-value: < 2.2e-16
```

```
# -> trasformazione della variabile

data$house_type_id[data$house_type_id=='HouseType 1: Pisos'|data$house_type_id=='Estudio'|
                     data$house_type_id=='HouseType 4: Dúplex']='Apartment'
data$house_type_id[data$house_type_id=='HouseType 2: Casa o chalet'|data$house_type_id=='Casa'|
                     data$house_type_id=='Finca']='Independent'
data$house_type_id[data$house_type_id=='HouseType 5: Áticos']='Attic'
summary(as.factor(data$house_type_id))
```

```
##   Apartment       Attic Independent
##        5591         391         305
```

```
boxplot(data$buy_price~as.factor(data$house_type_id),xlab='House Type',ylab='Buy Price',
        col=c('#F8766D','#00BA38','#619CFF'))
```




```
summary(lm(buy_price~as.factor(house_type_id)+sq_mt_built,data=data))
```

```
##
## Call:
## lm(formula = buy_price ~ as.factor(house_type_id) + sq_mt_built,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1916906  -116588   -27306    85493  4551625
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        -73451.44    5859.37  -12.54   <2e-16 ***
## as.factor(house_type_id)Attic      141785.04   12979.72   10.92   <2e-16 ***
## as.factor(house_type_id)Independent -406841.59   18288.11  -22.25   <2e-16 ***
## sq_mt_built                          4700.71      47.58   98.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 246500 on 6274 degrees of freedom
##   (9 osservazioni eliminate a causa di valori mancanti)
## Multiple R-squared:  0.6608, Adjusted R-squared:  0.6607
## F-statistic:  4075 on 3 and 6274 DF,  p-value: < 2.2e-16
```

```
### Balcone

summary(as.factor(data$has_balcony))
```

```
## TRUE NA's
## 1000 5287
```

```
summary(as.factor(data$has_terrace))
```

```
## TRUE NA's
## 2905 3382
```

```
data$has_balcony[data$has_balcony==TRUE|data$has_terrace==TRUE]='TRUE'
summary(as.factor(data$has_balcony))
```

```
## TRUE NA's
## 3425 2862
```

```
### Giardino

summary(as.factor(data$has_garden))
```

```
## TRUE NA's
##  215 6072
```

```
summary(as.factor(data$has_green_zones))
```

```
## TRUE NA's
## 1692 4595
```

```
data$has_garden[data$has_garden==TRUE|data$has_green_zones==TRUE]='TRUE'
summary(as.factor(data$has_garden))
```

```
## TRUE NA's
## 1907 4380
```

```
### Riscaldamento

summary(as.factor(data$has_central_heating))
```

```
## FALSE  TRUE  NA's
##  2381   857  3049
```

```
summary(as.factor(data$has_individual_heating))
```

```
## FALSE  TRUE  NA's
##   857  2381  3049
```

```
## 3

data_ng=as.data.frame(str_split_fixed(data$neighborhood_id,': ',3))
data_neighborhood=str_split_fixed(data_ng[,1],' ',2)[,2]
data_district=str_split_fixed(data_ng[,2],'District ',2)[,2]
summary(as.factor(data_neighborhood))
```

```
##      30     126      62     129      35      39     131     134      53     113
##     138     133     122     118     114     112     110     110     110     105
##      19     115      73      89     111      34      22      42      72      90
##     104     100     100      99      98      93      88      88      88      87
##      17      23      37      18      75      63      31      70      95      59
##      82      79      79      77      76      75      74      74      69      67
##      69      87       4     112       2      20      12      15      91     100
##      67      67      66      65      65      63      62      62      62      61
##     114       3      33      13      32      51      67      93       6       5
##      61      61      60      59      58      58      57      57      56      54
##      61      16     133      86     118      77     117      58      92      24
##      54      52      51      51      49      49      48      48      48      47
##      45      99     128      29      27      28      40      55      26      56
##      47      47      46      46      45      45      44      43      42      42
##      36      54      78     132       1      44      38      46     125      68
##      41      41      41      40      39      39      37      37      36      36
##     120      41      52     121     119      50      21      64      66      71
##      35      35      33      32      30      30      29      29      29      29
##     135      14     101     116      74      82     124      25      57 (Other)
##      27      26      25      25      25      25      24      24      24     330
```

```
summary(as.factor(data_district))
```

```
##   1  10  11  12  13  14  15  17  18  19   2  20  21   3   4   5   6   7   8   9
## 341 293 395  77 346 272 183 454 221 216   3 179 338 524 330 390 445 452 385 443
```

```
data_district=as.data.frame(data_district)

data_address=matrix(NA,nrow=nrow(data))
for(i in 1:nrow(data)){
  if(!is.na(data$raw_address[i])){
    data_address[i,1]=paste(data$raw_address[i],data$subtitle[i],sep=', ')
  }
}
data_address=as.data.frame(data_address)
colnames(data_address)='address'
head(data_address)
```

```
##                                        address
## 1     Calle de Godella, 64, San Cristóbal, Madrid
## 2         Calle del Talco, 68, San Andrés, Madrid
## 3 Calle de la Unanimidad, 67, Los Rosales, Madrid
## 4        Calle de Anoeta, 63, Los Ángeles, Madrid
## 5    Concepción de la Oliva, 21, Butarque, Madrid
## 6 Calle Arroyo de la Bulera, 31, Butarque, Madrid
```

```
## 4

data=subset(data,select=-c(buy_price_by_area,rent_price,parking_price,
                           is_parking_included_in_price,is_floor_under,has_central_heating,
                           title,subtitle,raw_address,street_name,street_number,
                           neighborhood_id,is_exact_address_hidden,has_terrace,has_green_zones))
dim(data)
```

```
## [1] 6287   28
```

```
# Errori nei Dati

round(colSums(is.na(data))/nrow(data),3)*100  # percentuali di dati mancanti
```

```
##              buy_price            sq_mt_built           sq_mt_useful
##                    0.0                    0.1                   58.2
##        sq_mt_allotment             built_year           n_bathrooms
##                   97.3                   69.3                    0.1
##               n_floors                n_rooms     energy_certificate
##                   95.8                    0.0                   32.0
##                  floor          house_type_id          is_accessible
##                    8.5                    0.0                   81.1
##            is_exterior      is_renewal_needed     is_new_development
##                    8.0                    0.0                    2.3
##   is_orientation_north   is_orientation_south    is_orientation_east
##                   51.6                   51.6                   51.6
##    is_orientation_west                 has_ac   has_fitted_wardrobes
##                   51.6                   51.8                   39.4
##               has_lift            has_balcony             has_garden
##                    5.3                   45.5                   69.7
##            has_parking               has_pool       has_storage_room
##                    0.0                   70.1                   61.3
## has_individual_heating
##                   48.5
```

```
data[sapply(data,is.logical)]=lapply(data[sapply(data,is.logical)],as.factor)
data[sapply(data,is.character)]=lapply(data[sapply(data,is.character)],as.factor)
summary(data)
```

```
##    buy_price         sq_mt_built     sq_mt_useful    sq_mt_allotment
##  Min.   :  42000   Min.   : 20.0   Min.   :  1.00   Min.   :  1.0
##  1st Qu.: 200000   1st Qu.: 69.0   1st Qu.: 59.00   1st Qu.: 97.5
##  Median : 324000   Median : 93.0   Median : 78.00   Median :250.0
##  Mean   : 458508   Mean   :114.6   Mean   : 91.91   Mean   :277.2
##  3rd Qu.: 564500   3rd Qu.:132.0   3rd Qu.:106.00   3rd Qu.:360.0
##  Max.   :7525000   Max.   :920.0   Max.   :750.00   Max.   :994.0
##                    NA's   :9       NA's   :3658     NA's   :6115
##    built_year    n_bathrooms      n_floors        n_rooms
##  Min.   :1850   Min.   : 1.0   Min.   :1.000   Min.   : 0.000
##  1st Qu.:1960   1st Qu.: 1.0   1st Qu.:2.000   1st Qu.: 2.000
##  Median :1975   Median : 2.0   Median :3.000   Median : 3.000
##  Mean   :1976   Mean   : 1.8   Mean   :3.095   Mean   : 2.657
```

```
##  3rd Qu.:2003   3rd Qu.: 2.0   3rd Qu.:4.000   3rd Qu.: 3.000
##  Max.   :2022   Max.   :11.0   Max.   :5.000   Max.   :24.000
##  NA's   :4356   NA's   :5      NA's   :6025
##  energy_certificate     floor          house_type_id  is_accessible
##  High: 748          Min.   :-1.00   Apartment  :5591   TRUE:1186
##  Low :3525          1st Qu.: 1.00   Attic      : 391   NA's:5101
##  NA's:2014          Median : 2.00   Independent: 305
##                     Mean   : 2.61
##                     3rd Qu.: 4.00
##                     Max.   : 9.00
##                     NA's   :536
##  is_exterior  is_renewal_needed is_new_development is_orientation_north
##  FALSE: 577   FALSE:5458        FALSE:4999         FALSE:2283
##  TRUE :5210   TRUE : 829        TRUE :1142         TRUE : 758
##  NA's : 500                     NA's : 146         NA's :3246
##
##
##
##
##  is_orientation_south is_orientation_east is_orientation_west  has_ac
##  FALSE:1572           FALSE:1810          FALSE:2028          TRUE:3033
##  TRUE :1469           TRUE :1231          TRUE :1013          NA's:3254
##  NA's :3246           NA's :3246          NA's :3246
##
##
##
##
##  has_fitted_wardrobes  has_lift    has_balcony has_garden  has_parking
##  TRUE:3810            FALSE:1311   TRUE:3425   TRUE:1907   FALSE:3897
##  NA's:2477            TRUE :4640   NA's:2862   NA's:4380   TRUE :2390
##                       NA's : 336
##
##
##
##
##  has_pool    has_storage_room has_individual_heating
##  TRUE:1877   TRUE:2431        FALSE: 857
##  NA's:4410   NA's:3856        TRUE :2381
##                               NA's :3049
##
##
##
##
```

```
data$has_ac=impute(data$has_ac,FALSE)
data$has_fitted_wardrobes=impute(data$has_fitted_wardrobes,FALSE)
data$has_balcony=impute(data$has_balcony,FALSE)
data$has_garden=impute(data$has_garden,FALSE)
data$has_storage_room=impute(data$has_storage_room,FALSE)
data$has_pool=impute(data$has_pool,FALSE)

data=subset(data,select=-is_accessible)

round(colSums(is.na(data))/nrow(data),3)*100
```

```
##              buy_price            sq_mt_built           sq_mt_useful
##                    0.0                    0.1                   58.2
##        sq_mt_allotment             built_year            n_bathrooms
##                   97.3                   69.3                    0.1
##               n_floors                n_rooms     energy_certificate
##                   95.8                    0.0                   32.0
##                  floor          house_type_id            is_exterior
##                    8.5                    0.0                    8.0
##      is_renewal_needed     is_new_development   is_orientation_north
##                    0.0                    2.3                   51.6
##   is_orientation_south    is_orientation_east    is_orientation_west
##                   51.6                   51.6                   51.6
##                 has_ac   has_fitted_wardrobes               has_lift
##                    0.0                    0.0                    5.3
##            has_balcony             has_garden            has_parking
##                    0.0                    0.0                    0.0
##               has_pool       has_storage_room has_individual_heating
##                    0.0                    0.0                   48.5
```

```
# Geocodifica degli Indirizzi

data_geo=geo(address=data_address$address,method='arcgis',lat=latitude,long=longitude)
data_geo=as.data.frame(data_geo)
head(data_geo)
```

```
##                                       address latitude longitude
## 1     Calle de Godella, 64, San Cristóbal, Madrid 40.34286  -3.68896
## 2         Calle del Talco, 68, San Andrés, Madrid 40.34462  -3.71521
## 3 Calle de la Unanimidad, 67, Los Rosales, Madrid 40.35811  -3.68510
## 4        Calle de Anoeta, 63, Los Ángeles, Madrid 40.35110  -3.70180
## 5    Concepción de la Oliva, 21, Butarque, Madrid 40.35411  -3.68135
## 6 Calle Arroyo de la Bulera, 31, Butarque, Madrid 40.33846  -3.67963
```

```
data_coord=data_geo[,c(3,2)]
colnames(data_coord)=c('x','y')
summary(data_coord)
```

```
##        x                y
##  Min.   :-3.884   Min.   :40.33
##  1st Qu.:-3.712   1st Qu.:40.39
##  Median :-3.695   Median :40.42
##  Mean   :-3.688   Mean   :40.42
##  3rd Qu.:-3.664   3rd Qu.:40.45
##  Max.   :-3.546   Max.   :40.53
```

```
# Salvataggio dei Dataset

#write_xlsx(data,'data_incomplete.xlsx')
#write_xlsx(data_coord,'data_coord.xlsx')
#write_xlsx(data_district,'data_district.xlsx')
```

Pre-Processing 2

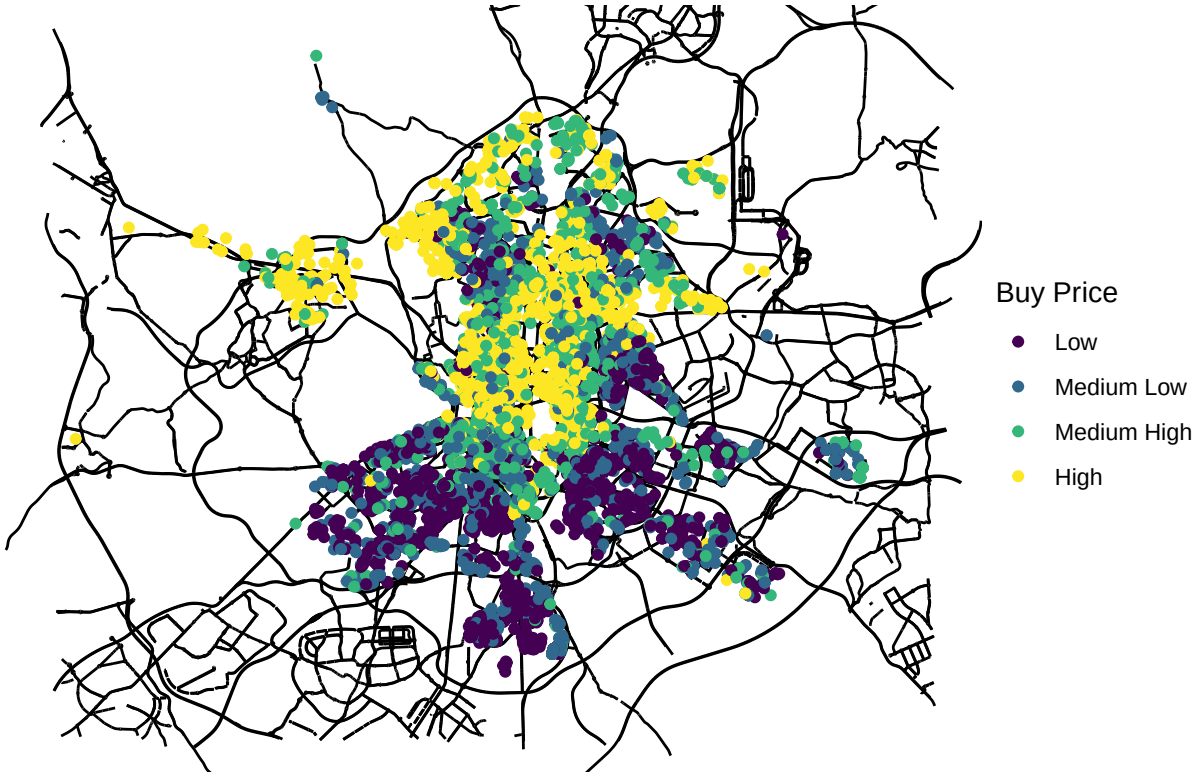
```
library(readxl)
library(osmdata)
library(ggplot2)
library(dvmisc)
library(viridis)
library(sf)
library(ggpubr)
library(geosphere)
library(writexl)
```

```
data_coord=read_excel('data_coord.xlsx')
data_coord=as.data.frame(data_coord)
data=read_excel('data_incomplete.xlsx')
data=as.data.frame(data)
```

```
# Mappe di Madrid

## Mappa Stradale

street_major=getbb(place_name='Madrid') %>%  # strade principali
  opq(timeout=100) %>%
  add_osm_feature(key='highway',value=c('motorway','trunk','primary')) %>%
  osmdata_sf()
street_major

street_minor=getbb(place_name='Madrid') %>%  # strade secondarie
  opq(timeout=100) %>%
  add_osm_feature(key='highway',value=c('secondary','tertiary')) %>%
  osmdata_sf()
street_minor

street_map=ggplot()+
  geom_sf(data=street_major$osm_lines,inherit.aes=FALSE,color='black',size=0.2) +
  geom_sf(data=street_minor$osm_lines,inherit.aes=FALSE,color='black',size=0.1) +
  theme_void()
street_map
```

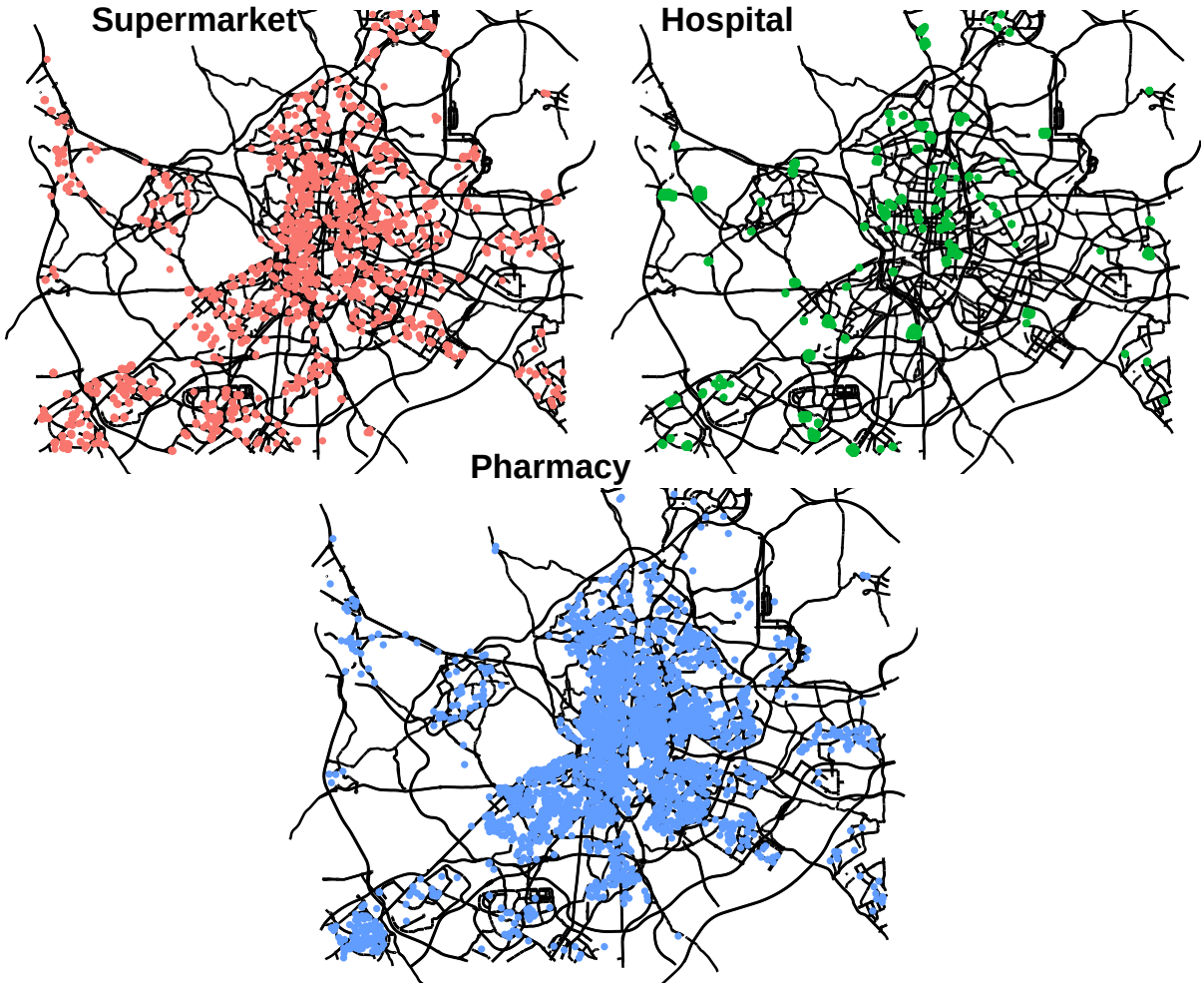
```
buy_price_q=quant_groups(data$buy_price,groups=4)  # raggruppamento per quartili
buy_price_q=as.factor(buy_price_q)
levels(buy_price_q)=c('Low','Medium Low','Medium High','High')
```

```
## Mappa dei Prezzi di Vendita

y_limit=c(40.3,40.55)

street_map +
  geom_point(data=data_coord,aes(x=x, y=y,colour=buy_price_q),size=1.5) +
  scale_color_viridis(discrete=TRUE)+
  coord_sf(ylim=y_limit,expand=FALSE) +
  labs(color='Buy Price')
```



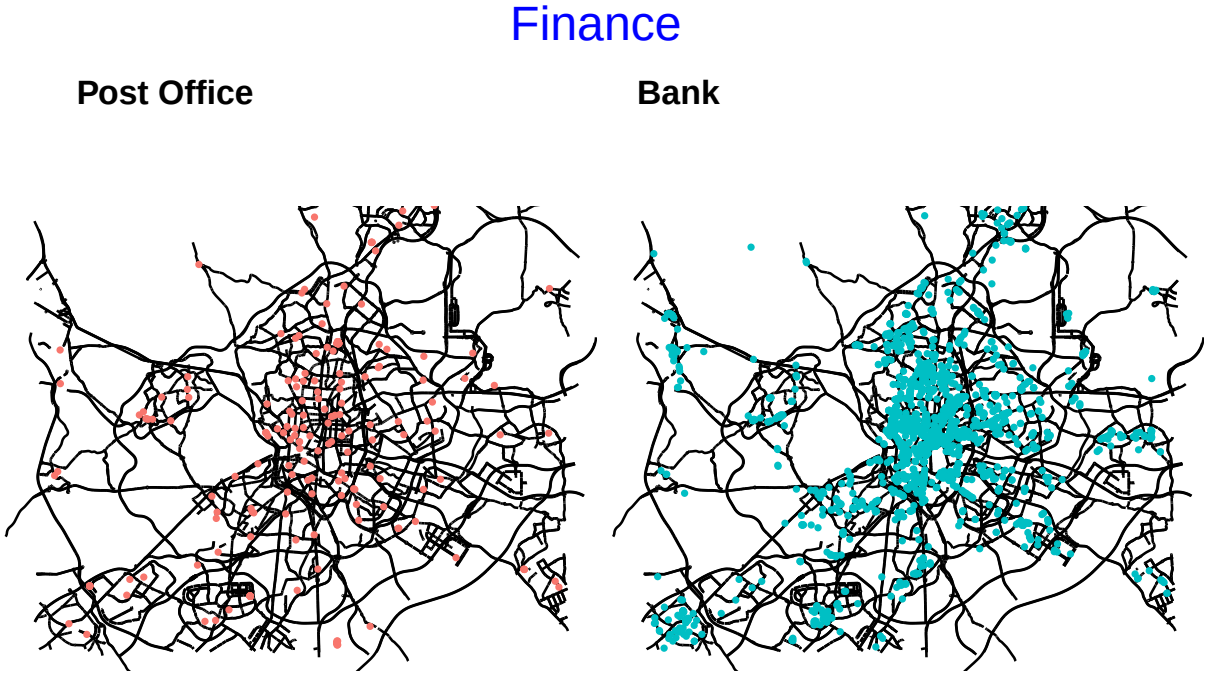

```
## Prima Necessita' / Salute

supermarket_madrid=getbb(place_name='Madrid') %>%  # supermercato
  opq(timeout=100) %>%
  add_osm_feature(key='shop',value='supermarket') %>%
  osmdata_sf()
supermarket_madrid

coord_supermarket=supermarket_madrid['osm_points']
coord_supermarket=do.call(rbind,coord_supermarket) %>% dplyr::select('osm_id','geometry')
coord_supermarket=st_coordinates(coord_supermarket)


hospital_madrid=getbb(place_name='Madrid') %>%  # ospedale
  opq(timeout=100) %>%
  add_osm_feature(key='amenity',value='hospital') %>%
  osmdata_sf()
hospital_madrid

coord_hospital=hospital_madrid['osm_points']
coord_hospital=do.call(rbind,coord_hospital) %>% dplyr::select('osm_id','geometry')
coord_hospital=st_coordinates(coord_hospital)


pharmacy_madrid=getbb(place_name='Madrid') %>%  # farmacia
  opq(timeout=100) %>%
```

```
  add_osm_feature(key='amenity',value='pharmacy') %>%
  osmdata_sf()
pharmacy_madrid

coord_pharmacy=pharmacy_madrid['osm_points']
coord_pharmacy=do.call(rbind,coord_pharmacy) %>% dplyr::select('osm_id','geometry')
coord_pharmacy=st_coordinates(coord_pharmacy)


supermarket_map=street_map +
  geom_sf(data=supermarket_madrid$osm_points,color='#F8766D',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

hospital_map=street_map +
  geom_sf(data=hospital_madrid$osm_points,color='#00BA38',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

pharmacy_map=street_map +
  geom_sf(data=pharmacy_madrid$osm_points,color='#619CFF',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

health_map=ggarrange(ggarrange(supermarket_map,hospital_map,ncol=2,
                               labels=c('Supermarket','Hospital')),
                     pharmacy_map,labels='Pharmacy',nrow=2,vjust=20,hjust=-3)
annotate_figure(health_map,top=text_grob('Healthcare',color='blue',size=20))
```



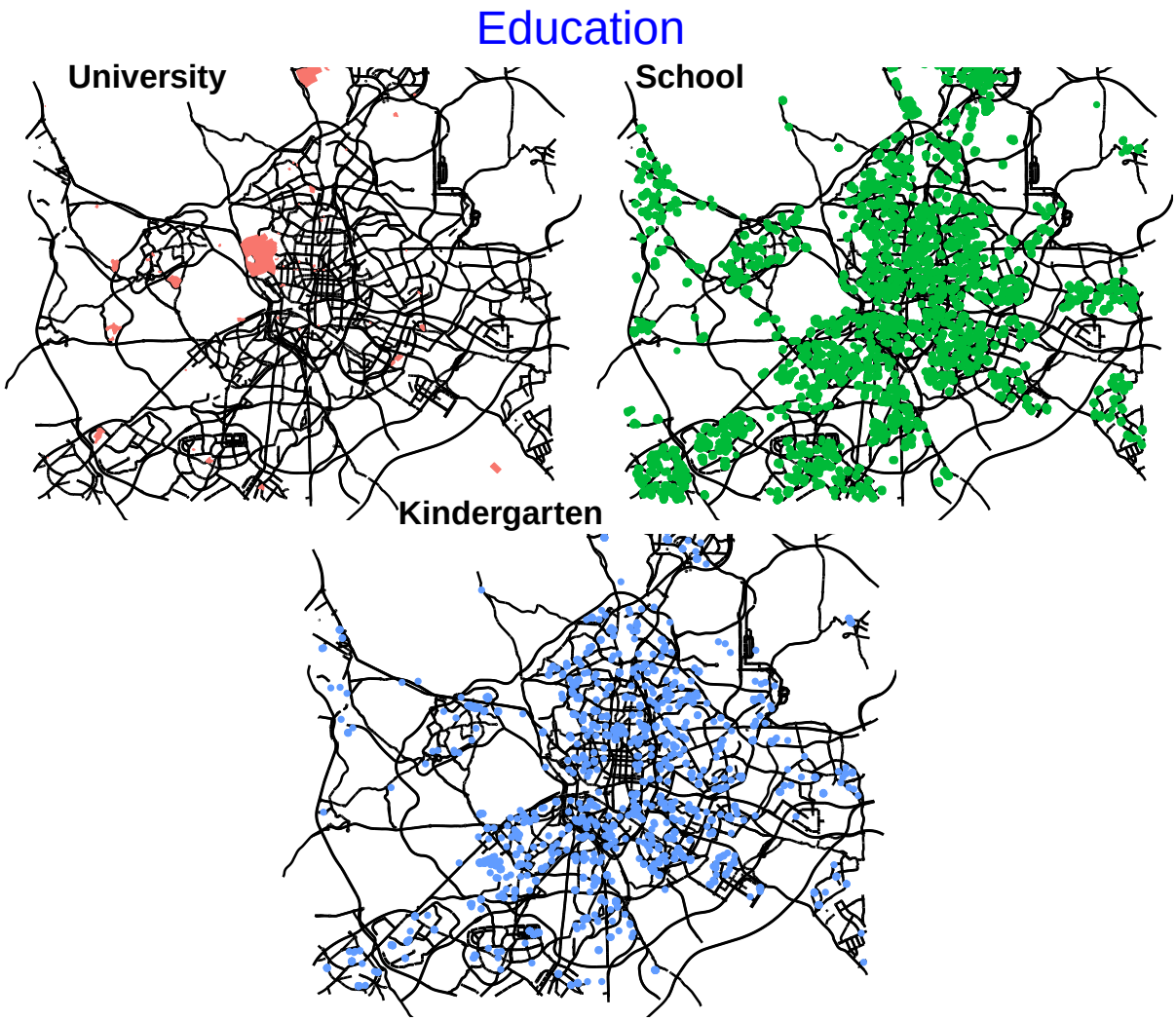

```
## Finanza

post_madrid=getbb(place_name='Madrid') %>%  # ufficio postale
  opq(timeout=100) %>%
  add_osm_feature(key='amenity',value='post_office') %>%
  osmdata_sf()
post_madrid

coord_post=post_madrid['osm_points']
coord_post=do.call(rbind,coord_post) %>% dplyr::select('osm_id','geometry')
coord_post=st_coordinates(coord_post)


bank_madrid=getbb(place_name='Madrid') %>%  # banca
  opq(timeout=100) %>%
  add_osm_feature(key='amenity',value='bank') %>%
  osmdata_sf()
bank_madrid
```

```
coord_bank=bank_madrid['osm_points']
coord_bank=do.call(rbind,coord_bank) %>% dplyr::select('osm_id','geometry')
coord_bank=st_coordinates(coord_bank)


post_map=street_map +
  geom_sf(data=post_madrid$osm_points,color='#F8766D',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

bank_map=street_map +
  geom_sf(data=bank_madrid$osm_points,color='#00BFC4',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

finance_map=ggarrange(post_map,bank_map,labels=c('Post Office','Bank'),vjust=3)
annotate_figure(finance_map,top=text_grob('Finance',color='blue',size=20,vjust=1))
```



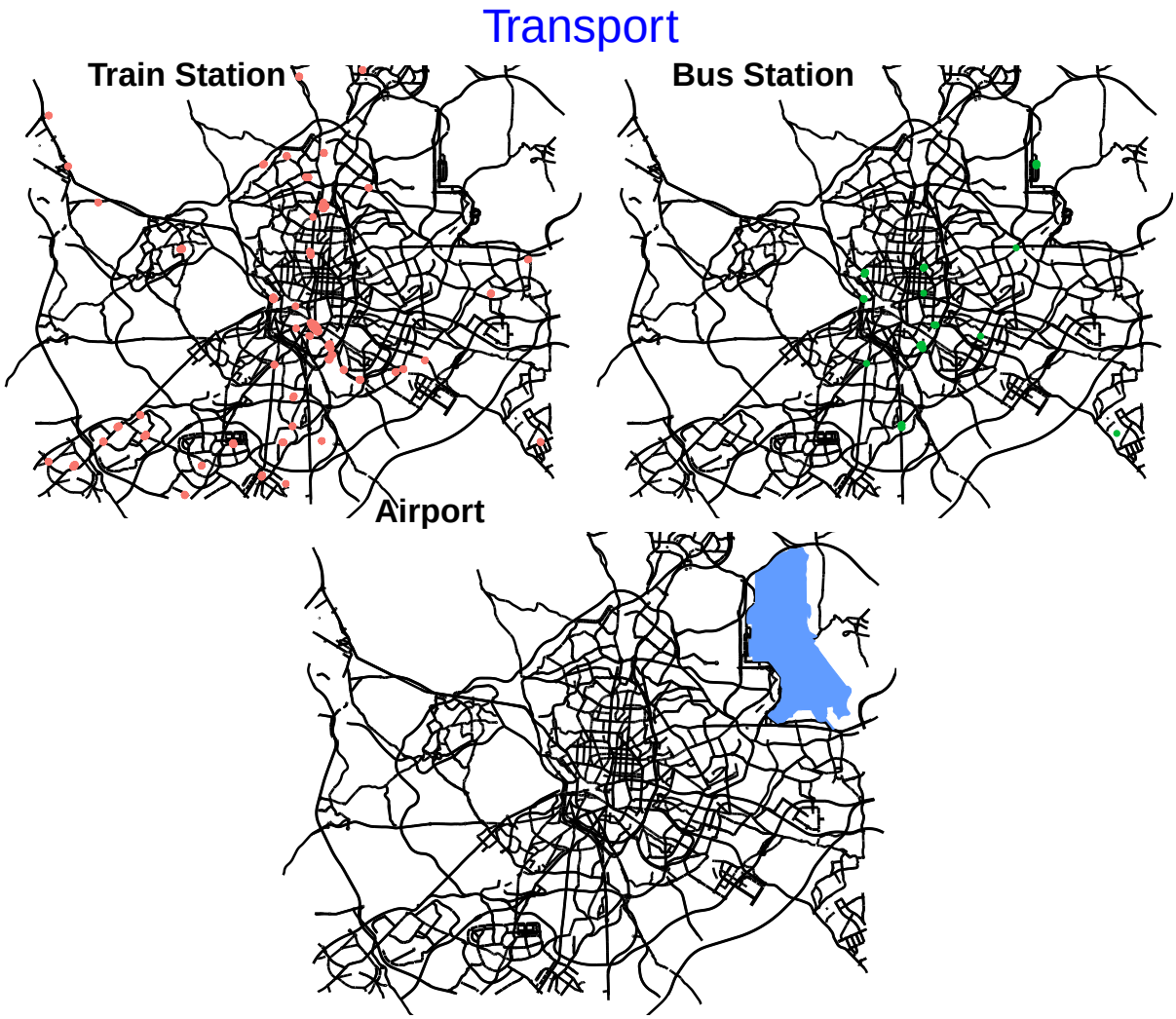

```
## Educazione

university_madrid=getbb(place_name='Madrid') %>%  # universita'
  opq(timeout=100) %>%
  add_osm_feature(key='amenity',value='university') %>%
  osmdata_sf()
university_madrid
```

```
coord_university=university_madrid['osm_points']
coord_university=do.call(rbind,coord_university) %>% dplyr::select('osm_id','geometry')
coord_university=st_coordinates(coord_university)


school_madrid=getbb(place_name='Madrid') %>%  # scuola dell'obbligo
  opq(timeout=100) %>%
  add_osm_feature(key='amenity',value='school') %>%
  osmdata_sf()
school_madrid

coord_school=school_madrid['osm_points']
coord_school=do.call(rbind,coord_school) %>% dplyr::select('osm_id','geometry')
coord_school=st_coordinates(coord_school)


kindergarten_madrid=getbb(place_name='Madrid') %>%  # scuola dell'infanzia
  opq(timeout=100) %>%
  add_osm_feature(key='amenity',value='kindergarten') %>%
  osmdata_sf()
kindergarten_madrid

coord_kindergarten=kindergarten_madrid['osm_points']
coord_kindergarten=do.call(rbind,coord_kindergarten) %>% dplyr::select('osm_id','geometry')
coord_kindergarten=st_coordinates(coord_kindergarten)


university_map=street_map +
  geom_sf(data=university_madrid$osm_polygons,color='#F8766D',fill='#F8766D') +
  coord_sf(ylim=y_limit,expand=FALSE)

school_map=street_map +
  geom_sf(data=school_madrid$osm_points,color='#00BA38',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

kindergarten_map=street_map +
  geom_sf(data=kindergarten_madrid$osm_points,color='#619CFF',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

education_map=ggarrange(ggarrange(university_map,school_map,ncol=2,
                                  labels=c('University','School')),
                        kindergarten_map,labels='Kindergarten',nrow=2,vjust=20,hjust=-2)
annotate_figure(education_map,top=text_grob('Education',color='blue',size=20))
```




```
## Trasporti

train_madrid=getbb(place_name='Madrid') %>%  # stazione dei treni
  opq(timeout=100) %>%
  add_osm_feature(key='building',value='train_station') %>%
  osmdata_sf()
train_madrid

coord_train=train_madrid['osm_points']
coord_train=do.call(rbind,coord_train) %>% dplyr::select('osm_id','geometry')
coord_train=st_coordinates(coord_train)


bus_madrid=getbb(place_name='Madrid') %>%  # stazione dei bus
  opq(timeout=100) %>%
  add_osm_feature(key='amenity',value='bus_station') %>%
  osmdata_sf()
bus_madrid
```

```
coord_bus=bus_madrid['osm_points']
coord_bus=do.call(rbind,coord_bus) %>% dplyr::select('osm_id','geometry')
coord_bus=st_coordinates(coord_bus)


airport_madrid=getbb(place_name='Madrid') %>%  # aeroporto
  opq(timeout=100) %>%
  add_osm_feature(key='aeroway',value='aerodrome') %>%
  osmdata_sf()
airport_madrid

coord_airport=airport_madrid['osm_points']
coord_airport=do.call(rbind,coord_airport) %>% dplyr::select('osm_id','geometry')
coord_airport=st_coordinates(coord_airport)


train_map=street_map +
  geom_sf(data=train_madrid$osm_points,color='#F8766D',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

bus_map=street_map +
  geom_sf(data=bus_madrid$osm_points,color='#00BA38',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

airport_map=street_map +
  geom_sf(data=airport_madrid$osm_polygons,color='#619CFF',fill='#619CFF') +
  coord_sf(ylim=y_limit,expand=FALSE)

transport_map=ggarrange(ggarrange(train_map,bus_map,ncol=2,
                                  labels=c('Train Station','Bus Station')),
                        airport_map,labels='Airport',nrow=2,vjust=20,hjust=-3.5)
annotate_figure(transport_map,top=text_grob('Transport',color='blue',size=20))
```

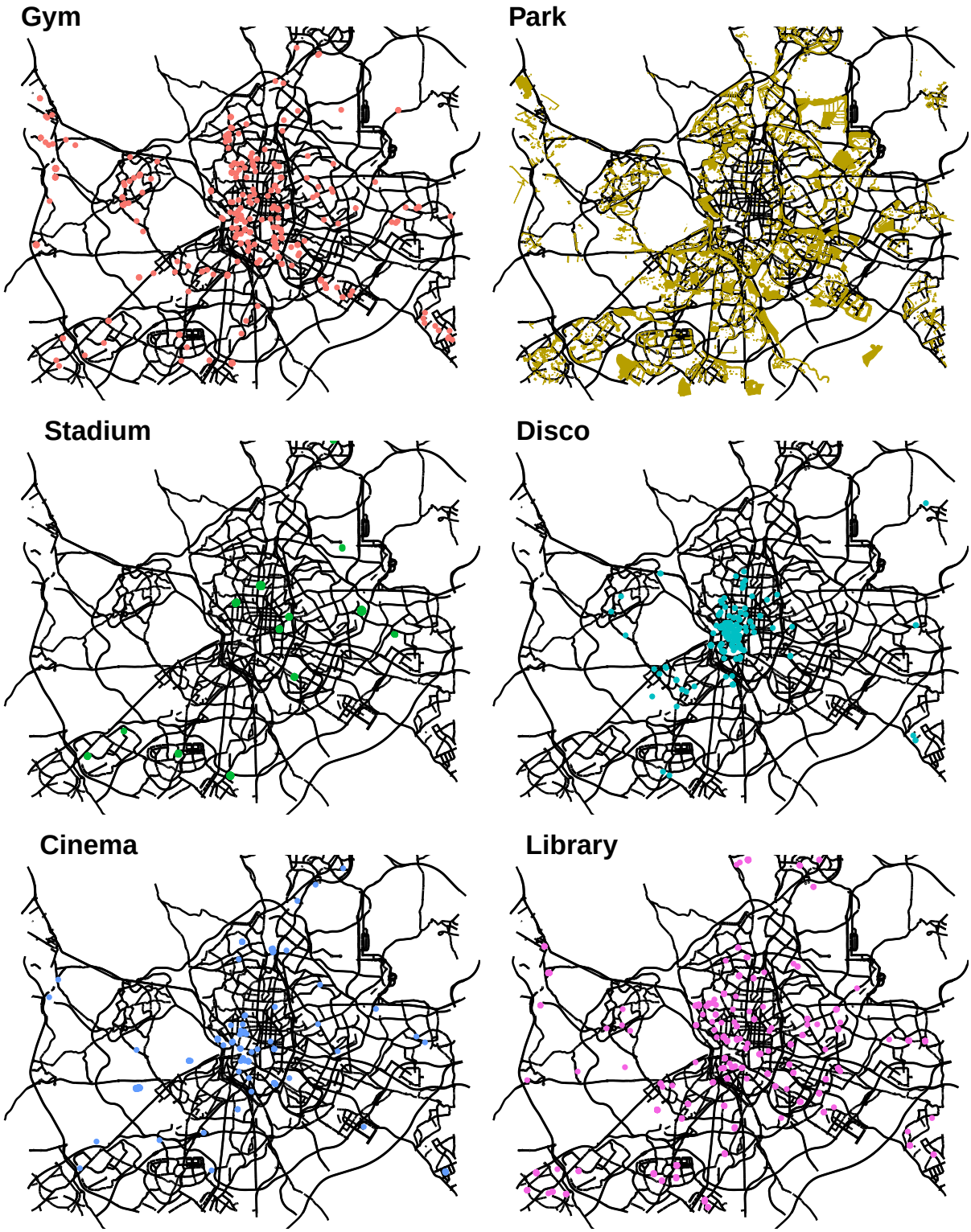



```
## Intrattenimento

gym_madrid=getbb(place_name='Madrid') %>%  # palestra
  opq(timeout=100) %>%
  add_osm_feature(key='leisure',value='fitness_centre') %>%
  osmdata_sf()
gym_madrid

coord_gym=gym_madrid['osm_points']
coord_gym=do.call(rbind,coord_gym) %>% dplyr::select('osm_id','geometry')
coord_gym=st_coordinates(coord_gym)


park_madrid=getbb(place_name='Madrid') %>%  # parco
  opq(timeout=100) %>%
  add_osm_feature(key='leisure',value='park') %>%
  osmdata_sf()
park_madrid
```

```
coord_park=park_madrid['osm_points']
coord_park=do.call(rbind,coord_park) %>% dplyr::select('osm_id','geometry')
coord_park=st_coordinates(coord_park)


stadium_madrid=getbb(place_name='Madrid') %>%  # stadio
  opq(timeout=100) %>%
  add_osm_feature(key='building',value='stadium') %>%
  osmdata_sf()
stadium_madrid

coord_stadium=stadium_madrid['osm_points']
coord_stadium=do.call(rbind,coord_stadium) %>% dplyr::select('osm_id','geometry')
coord_stadium=st_coordinates(coord_stadium)


disco_madrid=getbb(place_name='Madrid') %>%  # discoteca
  opq(timeout=100) %>%
  add_osm_feature(key='amenity',value='nightclub') %>%
  osmdata_sf()
disco_madrid

coord_disco=disco_madrid['osm_points']
coord_disco=do.call(rbind,coord_disco) %>% dplyr::select('osm_id','geometry')
coord_disco=st_coordinates(coord_disco)


cinema_madrid=getbb(place_name='Madrid') %>%  # cinema
  opq(timeout=100) %>%
  add_osm_feature(key='amenity',value='cinema') %>%
  osmdata_sf()
cinema_madrid

coord_cinema=cinema_madrid['osm_points']
coord_cinema=do.call(rbind,coord_cinema) %>% dplyr::select('osm_id','geometry')
coord_cinema=st_coordinates(coord_cinema)


library_madrid=getbb(place_name='Madrid') %>%  # biblioteca
  opq(timeout=100) %>%
  add_osm_feature(key='amenity',value='library') %>%
  osmdata_sf()
library_madrid

coord_library=library_madrid['osm_points']
coord_library=do.call(rbind,coord_library) %>% dplyr::select('osm_id','geometry')
coord_library=st_coordinates(coord_library)


gym_map=street_map +
  geom_sf(data=gym_madrid$osm_points,color='#F8766D',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)
```

```
park_map=street_map +
  geom_sf(data=park_madrid$osm_polygons,color='#B79F00',fill='#B79F00') +
  coord_sf(ylim=y_limit,expand=FALSE)

stadium_map=street_map +
  geom_sf(data=stadium_madrid$osm_points,color='#00BA38',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

disco_map=street_map +
  geom_sf(data=disco_madrid$osm_points,color='#00BFC4',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

cinema_map=street_map +
  geom_sf(data=cinema_madrid$osm_points,color='#619CFF',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

library_map=street_map +
  geom_sf(data=library_madrid$osm_points,color='#F564E3',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)


entertainment_map=ggarrange(gym_map,park_map,stadium_map,disco_map,cinema_map,library_map,
                            labels=c('Gym','Park','Stadium','Disco','Cinema','Library'),
                            vjust=1,nrow=3,ncol=2)
annotate_figure(entertainment_map,top=text_grob('Entertainment',color='blue',size=20,vjust=0.3))
```

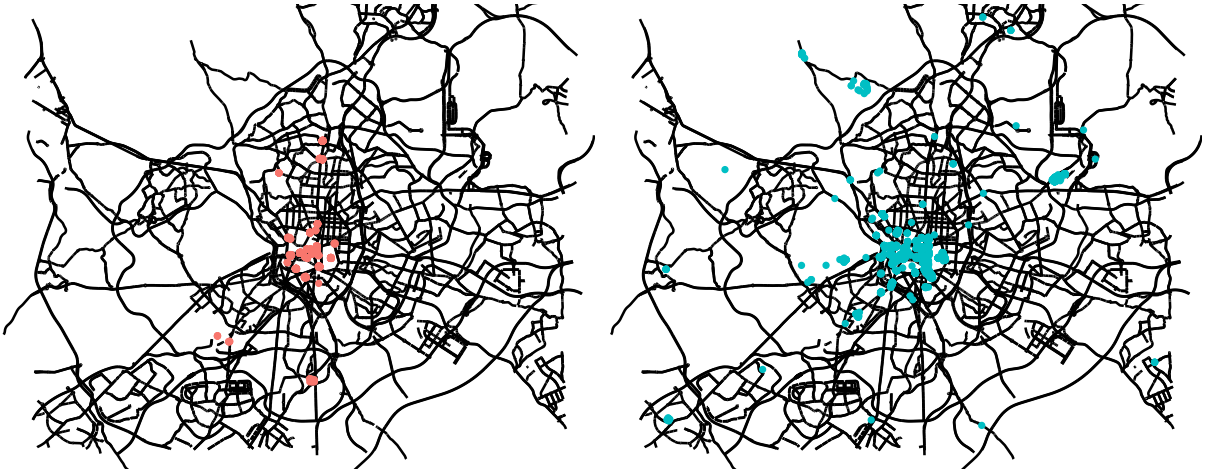
Entertainment



```
## Turismo

historic_madrid=getbb(place_name='Madrid') %>%  # cinema
  opq(timeout=100) %>%
  add_osm_feature(key='historic',value='building') %>%
  osmdata_sf()
historic_madrid

coord_historic=historic_madrid['osm_points']
coord_historic=do.call(rbind,coord_historic) %>% dplyr::select('osm_id','geometry')
coord_historic=st_coordinates(coord_historic)


attraction_madrid=getbb(place_name='Madrid') %>%  # cinema
  opq(timeout=100) %>%
  add_osm_feature(key='tourism',value='attraction') %>%
  osmdata_sf()
attraction_madrid

coord_attraction=attraction_madrid['osm_points']
coord_attraction=do.call(rbind,coord_attraction) %>% dplyr::select('osm_id','geometry')
coord_attraction=st_coordinates(coord_attraction)


historic_map=street_map +
  geom_sf(data=historic_madrid$osm_points,color='#F8766D',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

attraction_map=street_map +
  geom_sf(data=attraction_madrid$osm_points,color='#00BFC4',size=0.75) +
  coord_sf(ylim=y_limit,expand=FALSE)

tourism_map=ggarrange(historic_map,attraction_map,labels=c('Historic','Attraction'),vjust=3)
annotate_figure(tourism_map,top=text_grob('Tourism',color='blue',size=20,vjust=1))
```

Tourism

Historic **Attraction**



```
# Calcolo delle Distanze Minime

house_coord=as.data.frame(cbind(1:nrow(data_coord),data_coord))
names(house_coord)=c('id','long','lat')

min_dist=function(loc){
  from=house_coord[house_coord$id==loc,]
  distance=distHaversine(from[,2:3],to)
  min=data.frame(id=loc,dist=min(distance))
  return(min)
}

to=coord_supermarket
d_supermarket=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_supermarket)='d_supermarket'

to=coord_hospital
d_hospital=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_hospital)='d_hospital'

to=coord_pharmacy
d_pharmacy=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_pharmacy)='d_pharmacy'

to=coord_post
```

```
d_post=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_post)='d_post'

to=coord_bank
d_bank=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_bank)='d_bank'

to=coord_university
d_university=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_university)='d_university'

to=coord_school
d_school=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_school)='d_school'

to=coord_kindergarten
d_kindergarten=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_kindergarten)='d_kindergarten'

to=coord_train
d_train=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_train)='d_train'

to=coord_bus
d_bus=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_bus)='d_bus'

to=coord_airport
d_airport=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_airport)='d_airport'

to=coord_gym
d_gym=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_gym)='d_gym'

to=coord_park
d_park=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_park)='d_park'

to=coord_stadium
d_stadium=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_stadium)='d_stadium'

to=coord_library
d_library=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_library)='d_library'

to=coord_disco
d_disco=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_disco)='d_disco'

to=coord_cinema
d_cinema=bind_rows(lapply(house_coord$id,min_dist))[2]
```

```
colnames(d_cinema)='d_cinema'

to=coord_historic
d_historic=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_historic)='d_historic'

to=coord_attraction
d_attraction=bind_rows(lapply(house_coord$id,min_dist))[2]
colnames(d_attraction)='d_attraction'

data_distance=as.data.frame(cbind(d_supermarket,d_hospital,d_pharmacy,d_post,d_bank,
                                  d_university,d_school,d_kindergarten,d_train,d_bus,
                                  d_airport,d_gym,d_park,d_stadium,d_disco,d_cinema,
                                  d_library,d_historic,d_attraction))
data_distance=round(data_distance,0)
#View(data_distance)

#write_xlsx(data_distance,'data_distance.xlsx')
```

Pre-Processing 3

```
library(readxl)
library(caret)
library(mice)
library(Hmisc)
library(Iscores)
library(data.table)
library(mltools)
library(writexl)
```

```
data=read_excel('data_incomplete.xlsx')
data_coord=read_excel('data_coord.xlsx')
data_distance=read_excel('data_distance.xlsx')
data_district=read_excel('data_district.xlsx')
data=as.data.frame(data)
data_coord=as.data.frame(data_coord)
data_distance=as.data.frame(data_distance)
data_district=as.data.frame(data_district)
```

```
data[sapply(data,is.logical)]=lapply(data[sapply(data,is.logical)],as.factor)
data[sapply(data,is.character)]=lapply(data[sapply(data,is.character)],as.factor)

data$energy_certificate=factor(data$energy_certificate)
data$energy_certificate=relevel(data$energy_certificate,ref='Low')

data$house_type_id=factor(data$house_type_id)
data$house_type_id=relevel(data$house_type_id,ref='Apartment')

str(data)
```

```
## 'data.frame':    6287 obs. of  27 variables:
##  $ buy_price             : num  85000 144247 195000 205000 100000 ...
##  $ sq_mt_built           : num  64 94 123 109 61 97 93 74 125 158 ...
##  $ sq_mt_useful          : num  60 54 104 90 56 73 70 NA NA NA ...
##  $ sq_mt_allotment       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ built_year            : num  1960 NA 1992 1983 1966 ...
##  $ n_bathrooms           : num  1 2 2 2 1 2 2 1 2 2 ...
##  $ n_floors              : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ n_rooms               : num  2 2 3 3 3 2 2 3 3 3 ...
##  $ energy_certificate    : Factor w/ 2 levels "Low","High": 1 NA 1 1 1 NA NA NA 1 1 ...
##  $ floor                 : num  3 1 4 3 5 0 7 1 7 4 ...
##  $ house_type_id         : Factor w/ 3 levels "Apartment","Attic",..: 1 1 1 1 1 1 1 1 1 2 ...
##  $ is_exterior           : Factor w/ 2 levels "FALSE","TRUE": 2 2 2 2 2 2 2 2 2 2 ...
##  $ is_renewal_needed     : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
##  $ is_new_development    : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 2 2 1 1 1 ...
##  $ is_orientation_north  : Factor w/ 2 levels "FALSE","TRUE": 1 NA 1 1 2 NA NA 1 NA NA ...
```

```
##  $ is_orientation_south  : Factor w/ 2 levels "FALSE","TRUE": 1 NA 1 2 2 NA NA 2 NA NA ...
##  $ is_orientation_east   : Factor w/ 2 levels "FALSE","TRUE": 1 NA 2 1 1 NA NA 1 NA NA ...
##  $ is_orientation_west   : Factor w/ 2 levels "FALSE","TRUE": 2 NA 1 1 1 NA NA 1 NA NA ...
##  $ has_ac                : Factor w/ 2 levels "FALSE","TRUE": 2 1 2 1 1 1 1 1 1 2 ...
##  $ has_fitted_wardrobes  : Factor w/ 2 levels "FALSE","TRUE": 1 2 2 2 1 1 1 2 2 2 ...
##  $ has_lift              : Factor w/ 2 levels "FALSE","TRUE": 1 2 2 2 1 2 2 1 2 2 ...
##  $ has_balcony           : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 2 2 2 2 1 2 2 ...
##  $ has_garden            : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 2 2 1 2 2 ...
##  $ has_parking           : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 2 ...
##  $ has_pool              : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 2 2 1 2 2 ...
##  $ has_storage_room      : Factor w/ 2 levels "FALSE","TRUE": 1 2 1 2 1 1 1 2 2 2 ...
##  $ has_individual_heating: Factor w/ 2 levels "FALSE","TRUE": NA 2 NA NA 2 NA NA 2 2 2 ...
```

```
# Data Splitting

set.seed(123)

sample=createDataPartition(data$buy_price,p=0.8,groups=10,list=FALSE)  # splitting stratificato

ind_train=rep(FALSE,dim(data)[1])
ind_train[sample]=TRUE

train=data[sample,]
test=data[-sample,]

train_coord=data_coord[sample,]
test_coord=data_coord[-sample,]

train_distance=data_distance[sample,]
test_distance=data_distance[-sample,]

train_district=as.data.frame(data_district[sample,])
test_district=as.data.frame(data_district[-sample,])

c(dim(train)[1],dim(test)[1])
```

```
## [1] 5033 1254
```

```
# Dati Mancanti

round(colSums(is.na(train))/nrow(train),3)*100
```

```
##              buy_price           sq_mt_built           sq_mt_useful
##                    0.0                   0.1                   58.6
##        sq_mt_allotment            built_year           n_bathrooms
##                   97.4                  68.9                   0.1
##               n_floors               n_rooms     energy_certificate
##                   96.0                   0.0                   31.9
##                  floor         house_type_id           is_exterior
##                    8.7                   0.0                   8.0
##      is_renewal_needed     is_new_development   is_orientation_north
##                    0.0                   2.5                   52.5
##   is_orientation_south    is_orientation_east    is_orientation_west
```

```
##                   52.5                   52.5                   52.5
##                 has_ac   has_fitted_wardrobes               has_lift
##                    0.0                    0.0                    5.3
##            has_balcony            has_garden            has_parking
##                    0.0                    0.0                    0.0
##               has_pool      has_storage_room has_individual_heating
##                    0.0                    0.0                   48.6
```

```
round(colSums(is.na(test))/nrow(test),3)*100
```

```
##              buy_price           sq_mt_built           sq_mt_useful
##                    0.0                   0.2                   56.4
##        sq_mt_allotment            built_year           n_bathrooms
##                   96.9                  70.7                    0.0
##               n_floors               n_rooms     energy_certificate
##                   95.2                   0.0                   32.4
##                  floor         house_type_id            is_exterior
##                    7.9                   0.0                    7.7
##      is_renewal_needed     is_new_development   is_orientation_north
##                    0.0                   1.5                   48.0
##   is_orientation_south    is_orientation_east    is_orientation_west
##                   48.0                  48.0                   48.0
##                 has_ac   has_fitted_wardrobes               has_lift
##                    0.0                   0.0                    5.7
##            has_balcony            has_garden            has_parking
##                    0.0                   0.0                    0.0
##               has_pool      has_storage_room has_individual_heating
##                    0.0                   0.0                   48.2
```

```
## Verifica della Soglia

### 50%
train_50=subset(train,select=-c(sq_mt_useful,sq_mt_allotment,n_floors,built_year,is_orientation_north,
                                is_orientation_south,is_orientation_west,is_orientation_east))
imp_50=mice(train_50,m=1,method='pmm',visitSequence='monotone',seed=123,printFlag=FALSE)
train_50_complete=complete(imp_50,1)

### 25%
train_25=subset(train_50,select=-c(energy_certificate,has_individual_heating))
imp_25=mice(train_25,m=1,method='pmm',visitSequence='monotone',seed=123,printFlag=FALSE)
train_25_complete=complete(imp_25,1)

### 5%
train_5=subset(train_25,select=-c(floor,is_exterior,has_lift))
imp_5=mice(train_5,m=1,method='pmm',visitSequence='monotone',seed=123,printFlag=FALSE)
train_5_complete=complete(imp_5,1)


set.seed(123)
control=trainControl(method='cv',number=10)
tune_rf=expand.grid(mtry=10,splitrule='variance',min.node.size=10)

fit_50=train(buy_price~.,data=train_50_complete,method='lmStepAIC',direction='both',
```

```
            k=log(nrow(train_50_complete)),trControl=control,trace=FALSE)
fit_50$results[which.min(fit_50$results$RMSE),]
```

```
##   parameter     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      none 239524.9 0.7117134 144733.1 33484.33 0.02480867 8648.584
```

```
fit_rf_50=train(buy_price~.,data=train_50_complete,method='ranger',importance='impurity',
                num.trees=150,tuneGrid=tune_rf,trControl=control,trace=FALSE)
fit_rf_50$results[which.min(fit_rf_50$results$RMSE),]
```

```
##   mtry splitrule min.node.size     RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 1   10  variance            10 205800.2 0.7916392 118385.1 36379.83 0.04016012
##      MAESD
## 1 6557.227
```

```
fit_25=train(buy_price~.,data=train_25_complete,method='lmStepAIC',direction='both',
             k=log(nrow(train_25_complete)),trControl=control,trace=FALSE)
fit_25$results[which.min(fit_25$results$RMSE),]
```

```
##   parameter     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      none 242668.9 0.7044997 144144.5 19216.48 0.03629738 6340.727
```

```
fit_rf_25=train(buy_price~.,data=train_25_complete,method='ranger',importance='impurity',
                num.trees=150,tuneGrid=tune_rf,trControl=control)
fit_rf_25$results[which.min(fit_rf_25$results$RMSE),]
```

```
##   mtry splitrule min.node.size     RMSE  Rsquared      MAE  RMSESD RsquaredSD
## 1   10  variance            10 215631.7 0.7634431 121296.7 22544.7 0.03233627
##      MAESD
## 1 3904.114
```

```
fit_5=train(buy_price~.,data=train_5_complete,method='lmStepAIC',direction='both',
            k=log(nrow(train_5_complete)),trControl=control,trace=FALSE)
fit_5$results[which.min(fit_5$results$RMSE),]
```

```
##   parameter     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      none 245647.4 0.6938566 147645.7 26442.07 0.04325232 7822.453
```

```
fit_rf_5=train(buy_price~.,data=train_5_complete,method='ranger',importance='impurity',
               num.trees=150,tuneGrid=tune_rf,trControl=control)
fit_rf_5$results[which.min(fit_rf_5$results$RMSE),]
```

```
##   mtry splitrule min.node.size     RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 1   10  variance            10 219477.6 0.7541298 127560.7 37976.66 0.04411452
##      MAESD
## 1 8639.789
```

```
summary(fit_50$finalModel)
```

```
##
## Call:
## lm(formula = .outcome ~ sq_mt_built + n_bathrooms + n_rooms +
##     energy_certificateHigh + floor + house_type_idAttic + house_type_idIndependent +
##     is_exteriorTRUE + is_new_developmentTRUE + has_acTRUE + has_liftTRUE +
##     has_gardenTRUE + has_individual_heatingTRUE, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1635973  -109130    -6965    82591  4549880
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -59230.90   17497.42  -3.385 0.000717 ***
## sq_mt_built                    4040.98      84.69  47.713  < 2e-16 ***
## n_bathrooms                   86195.78    6489.67  13.282  < 2e-16 ***
## n_rooms                      -22207.90    3831.04  -5.797 7.17e-09 ***
## energy_certificateHigh        51127.04    9471.19   5.398 7.04e-08 ***
## floor                         10817.87    1771.93   6.105 1.10e-09 ***
## house_type_idAttic           112101.11   14653.01   7.650 2.39e-14 ***
## house_type_idIndependent    -352311.19   20506.51 -17.180  < 2e-16 ***
## is_exteriorTRUE              -68900.15   12015.34  -5.734 1.04e-08 ***
## is_new_developmentTRUE        99636.94   11388.64   8.749  < 2e-16 ***
## has_acTRUE                    26011.16    7514.67   3.461 0.000542 ***
## has_liftTRUE                  33021.02    9375.02   3.522 0.000432 ***
## has_gardenTRUE               -46146.69    8000.68  -5.768 8.51e-09 ***
## has_individual_heatingTRUE   -67668.71    9280.28  -7.292 3.54e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 239000 on 5019 degrees of freedom
## Multiple R-squared:   0.71,  Adjusted R-squared:  0.7093
## F-statistic: 945.4 on 13 and 5019 DF,  p-value: < 2.2e-16
```

```
varImp(fit_50)  # importanza delle variabili nel modello regressivo lineare
```

```
## loess r-squared variable importance
##
##                          Overall
## sq_mt_built             100.0000
## n_bathrooms              84.1615
## n_rooms                  40.7187
## house_type_id            22.2370
## has_lift                 14.4727
## has_parking              12.6594
## has_pool                 10.8137
## has_storage_room          6.9495
## is_new_development        6.8684
## has_individual_heating    6.1837
## has_garden                5.2336
```

```
## energy_certificate       4.9279
## has_balcony              4.0931
## is_exterior              1.9452
## floor                    1.4476
## has_fitted_wardrobes     0.5273
## has_ac                   0.3816
## is_renewal_needed        0.0000
```

```
varImp(fit_rf_50)  # importanza delle variabili nel Random Forest
```

```
## ranger variable importance
##
##                             Overall
## sq_mt_built                100.0000
## n_bathrooms                 39.9546
## n_rooms                     10.9108
## floor                        7.5507
## house_type_idIndependent     3.7786
## energy_certificateHigh       3.2569
## has_liftTRUE                 3.0410
## house_type_idAttic           2.9774
## is_new_developmentTRUE       2.7030
## has_individual_heatingTRUE   2.0715
## has_gardenTRUE               1.0370
## has_poolTRUE                 0.9613
## has_fitted_wardrobesTRUE     0.9541
## has_balconyTRUE              0.8891
## has_storage_roomTRUE         0.8738
## has_acTRUE                   0.8076
## has_parkingTRUE              0.6000
## is_renewal_neededTRUE        0.4228
## is_exteriorTRUE              0.0000
```

```
train=train_50
test=subset(test,select=-c(sq_mt_useful,sq_mt_allotment,n_floors,built_year,is_orientation_north,
                           is_orientation_south,is_orientation_west,is_orientation_east))
```

```
## Missing at Random (MAR)

### Imputazione Singola

train_mmm=train
train_mmm$sq_mt_built=as.numeric(impute(train_mmm$sq_mt_built,mean))
train_mmm$n_bathrooms=as.numeric(impute(train_mmm$n_bathrooms,mean))
train_mmm$floor=as.numeric(impute(train_mmm$floor,mean))

mode=function(x) {
  x=na.omit(x)
  uniq=unique(x)
  uniq[which.max(tabulate(match(x,uniq)))]
}

train_mmm$is_floor_under[is.na(train_mmm$is_floor_under)]=mode(train_mmm$is_floor_under)
```

```
train_mmm$has_lift[is.na(train_mmm$has_lift)]=mode(train_mmm$has_lift)
train_mmm$has_individual_heating[is.na(train_mmm$has_individual_heating)]=
  mode(train_mmm$has_individual_heating)
train_mmm$is_new_development[is.na(train_mmm$is_new_development)]=
  mode(train_mmm$is_new_development)
train_mmm$is_exterior[is.na(train_mmm$is_exterior)]=mode(train_mmm$is_exterior)
train_mmm$energy_certificate[is.na(train_mmm$energy_certificate)]=
  mode(train_mmm$energy_certificate)


imp_lm=mice(train,m=1,defaultMethod=c('norm.predict','logreg','polyreg','polr'),
            visitSequence='monotone',seed=123,printFlag=FALSE)
train_lm=complete(imp_lm,1)

imp_sr=mice(train,m=1,defaultMethod=c('norm.nob','logreg','polyreg','polr'),
            visitSequence='monotone',seed=123,printFlag=FALSE)
train_sr=complete(imp_sr,1)

imp_default=mice(train,visitSequence='monotone',m=1,seed=123,printFlag=FALSE)
train_default=complete(imp_default,1)

imp_pmm=mice(train,m=1,method='pmm',visitSequence='monotone',seed=123,printFlag=FALSE)
train_pmm=complete(imp_pmm,1)

imp_cart=mice(train,m=1,method='cart',visitSequence='monotone',seed=123,printFlag=FALSE)
train_cart=complete(imp_cart,1)

imp_rf=mice(train,m=1,method='rf',visitSequence='monotone',seed=123,printFlag=FALSE)
train_rf=complete(imp_rf,1)

tr_imp=list()
tr_imp[[1]]=train_mmm
tr_imp[[2]]=train_lm
tr_imp[[3]]=train_sr
tr_imp[[4]]=train_default
tr_imp[[5]]=train_pmm
tr_imp[[6]]=train_cart
tr_imp[[7]]=train_rf

imputations=list()

for(i in 1:7){
  imputations[[i]]=lapply(1,function(j){
    newdata=one_hot(as.data.table(tr_imp[i]))
    return(newdata)
  })
}
train_hot=one_hot(as.data.table(train))
```

```
#### I-Scores 1

set.seed(123)
methods=c('MMM','LM+LOG','SR+LOG','PMM+LOG','PMM','CART','RF')
```

```
IS=Iscores(imputations,methods,train_hot,num.proj=5)
#Iscores(imputations,methods,train_hot,num.proj=10)
#Iscores(imputations,methods,train_hot,num.proj=20)
#Iscores(imputations,methods,train_hot,num.proj=50)
```

```
IS
```

```
##             MMM    LM+LOG   SR+LOG PMM+LOG       PMM      CART        RF
## [1,] -2.513805 -1.878854 -1.29501       0 -0.325211 -0.224616 -0.417388
```

```
### Imputazione Multipla

imp_lm_m=mice(train,m=5,defaultMethod=c('norm.predict','logreg','polyreg','polr'),
              visitSequence='monotone',seed=123,printFlag=FALSE)
train_lm_m=list()
for(i in 1:5){
  train_lm_m[[i]]=complete(imp_lm_m,i)
}

imp_sr_m=mice(train,m=5,defaultMethod=c('norm.nob','logreg','polyreg','polr'),
              visitSequence='monotone',seed=123,printFlag=FALSE)
train_sr_m=list()
for(i in 1:5){
  train_sr_m[[i]]=complete(imp_sr_m,i)
}

imp_default_m=mice(train,m=5,visitSequence='monotone',seed=123,printFlag=FALSE)
train_default_m=list()
for(i in 1:5){
  train_default_m[[i]]=complete(imp_default_m,i)
}

imp_pmm_m=mice(train,m=5,method='pmm',visitSequence='monotone',seed=123,printFlag=FALSE)
train_pmm_m=list()
for(i in 1:5){
  train_pmm_m[[i]]=complete(imp_pmm_m,i)
}

imp_cart_m=mice(train,m=5,method='cart',visitSequence='monotone',seed=123,printFlag=FALSE)
train_cart_m=list()
for(i in 1:5){
  train_cart_m[[i]]=complete(imp_cart_m,i)
}

imp_rf_m=mice(train,m=5,method='rf',visitSequence='monotone',seed=123,printFlag=FALSE)
train_rf_m=list()
for(i in 1:5){
  train_rf_m[[i]]=complete(imp_rf_m,i)
}


imputations=list()
```

```
imputations[[1]]=lapply(1:5,function(i){
  newdata=one_hot(as.data.table(train_lm_m[[i]]))
  return(newdata)
})

imputations[[2]]=lapply(1:5,function(i){
  newdata=one_hot(as.data.table(train_sr_m[[i]]))
  return(newdata)
})

imputations[[3]]=lapply(1:5,function(i){
  newdata=one_hot(as.data.table(train_default_m[[i]]))
  return(newdata)
})

imputations[[4]]=lapply(1:5,function(i){
  newdata=one_hot(as.data.table(train_pmm_m[[i]]))
  return(newdata)
})

imputations[[5]]=lapply(1:5,function(i){
  newdata=one_hot(as.data.table(train_cart_m[[i]]))
  return(newdata)
})

imputations[[6]]=lapply(1:5, function(i) {
  newdata=one_hot(as.data.table(train_rf_m[[i]]))
  return(newdata)
})
```

```
#### I-Scores 2

#set.seed(123)
#methods=c('LM+LOG','SR+LOG','PMM+LOG','PMM','CART','RF')
#Iscores(imputations,methods,train_hot,num.proj=5,m=5)
#Iscores(imputations,methods,train_hot,num.proj=10,m=5)
#Iscores(imputations,methods,train_hot,num.proj=20,m=5)
#Iscores(imputations,methods,train_hot,num.proj=50,m=5)
```

```
## Selezione del Metodo di Imputazione

train=train_rf
summary(train)
```

```
##    buy_price        sq_mt_built     n_bathrooms        n_rooms
##  Min.   :  42000   Min.   : 20.0   Min.   : 1.000   Min.   : 0.000
##  1st Qu.: 202000   1st Qu.: 69.0   1st Qu.: 1.000   1st Qu.: 2.000
##  Median : 321000   Median : 94.0   Median : 2.000   Median : 3.000
##  Mean   : 460105   Mean   :115.3   Mean   : 1.806   Mean   : 2.666
##  3rd Qu.: 565000   3rd Qu.:132.0   3rd Qu.: 2.000   3rd Qu.: 3.000
##  Max.   :7525000   Max.   :920.0   Max.   :11.000   Max.   :24.000
##  energy_certificate     floor           house_type_id  is_exterior
##  Low :3953          Min.   :-1.000   Apartment  :4474   FALSE: 458
```

```
##  High:1080          1st Qu.: 1.000   Attic      : 321   TRUE :4575
##                     Median : 2.000   Independent: 238
##                     Mean   : 2.608
##                     3rd Qu.: 4.000
##                     Max.   : 9.000
##  is_renewal_needed is_new_development   has_ac     has_fitted_wardrobes
##  FALSE:4374        FALSE:4095         FALSE:2596   FALSE:1990
##  TRUE : 659        TRUE : 938         TRUE :2437   TRUE :3043
##
##
##
##
##   has_lift    has_balcony  has_garden   has_parking   has_pool
##  FALSE:1087   FALSE:2290   FALSE:3502   FALSE:3082   FALSE:3514
##  TRUE :3946   TRUE :2743   TRUE :1531   TRUE :1951   TRUE :1519
##
##
##
##
##  has_storage_room has_individual_heating
##  FALSE:3080       FALSE:1049
##  TRUE :1953       TRUE :3984
##
##
##
##
```

```
set.seed(123)  # riproduzione del metodo di imputazione per il test set
pred_m=matrix(1,nrow=length(test),ncol=length(test))
diag(pred_m)=0
pred_m[1,]=0
pred_m[,1]=0
imp_rf_test=mice(test,m=1,method='rf',visitSequence='monotone',
                 predictorMatrix=pred_m,seed=123,printFlag=FALSE)
test_rf=complete(imp_rf_test,1)
test=test_rf

data=subset(data,select=-c(sq_mt_useful,sq_mt_allotment,n_floors,built_year,is_orientation_north,
                           is_orientation_south,is_orientation_west,is_orientation_east))
data[sample,]=train
data[-sample,]=test
data=as.data.frame(cbind(data,ind_train))
```

```
#write_xlsx(data,'data_complete1.xlsx')
#write_xlsx(train,'train1.xlsx')
#write_xlsx(test,'test1.xlsx')
#write_xlsx(train_coord,'train_coord.xlsx')
#write_xlsx(test_coord,'test_coord.xlsx')
#write_xlsx(train_distance,'train_distance.xlsx')
#write_xlsx(test_distance,'test_distance.xlsx')
#write_xlsx(train_district,'train_district.xlsx')
#write_xlsx(test_district,'test_district.xlsx')
```

Analisi Esplorativa

```
library(readxl)
library(gclus)
library(mclust)
library(sf)
library(spdep)
library(ggplot2)
library(caret)
library(spatialreg)
library(doParallel)
library(writexl)
```

```
data=as.data.frame(read_excel('data_complete.xlsx'))
train=as.data.frame(read_excel('train.xlsx'))
test=as.data.frame(read_excel('test.xlsx'))

data_distance=as.data.frame(read_excel('data_distance.xlsx'))
train_distance=as.data.frame(read_excel('train_distance.xlsx'))
test_distance=as.data.frame(read_excel('test_distance.xlsx'))

train_district=as.data.frame(read_excel('train_district.xlsx'))
test_district=as.data.frame(read_excel('test_district.xlsx'))
data_coord=as.data.frame(read_excel('data_coord.xlsx'))

train_coord=as.data.frame(read_excel('train_coord.xlsx'))
test_coord=as.data.frame(read_excel('test_coord.xlsx'))


train[sapply(train,is.logical)]=lapply(train[sapply(train,is.logical)],as.factor)
train[sapply(train,is.character)]=lapply(train[sapply(train,is.character)],as.factor)
test[sapply(test,is.logical)]=lapply(test[sapply(test,is.logical)],as.factor)
test[sapply(test,is.character)]=lapply(test[sapply(test,is.character)],as.factor)

attach(train)
```

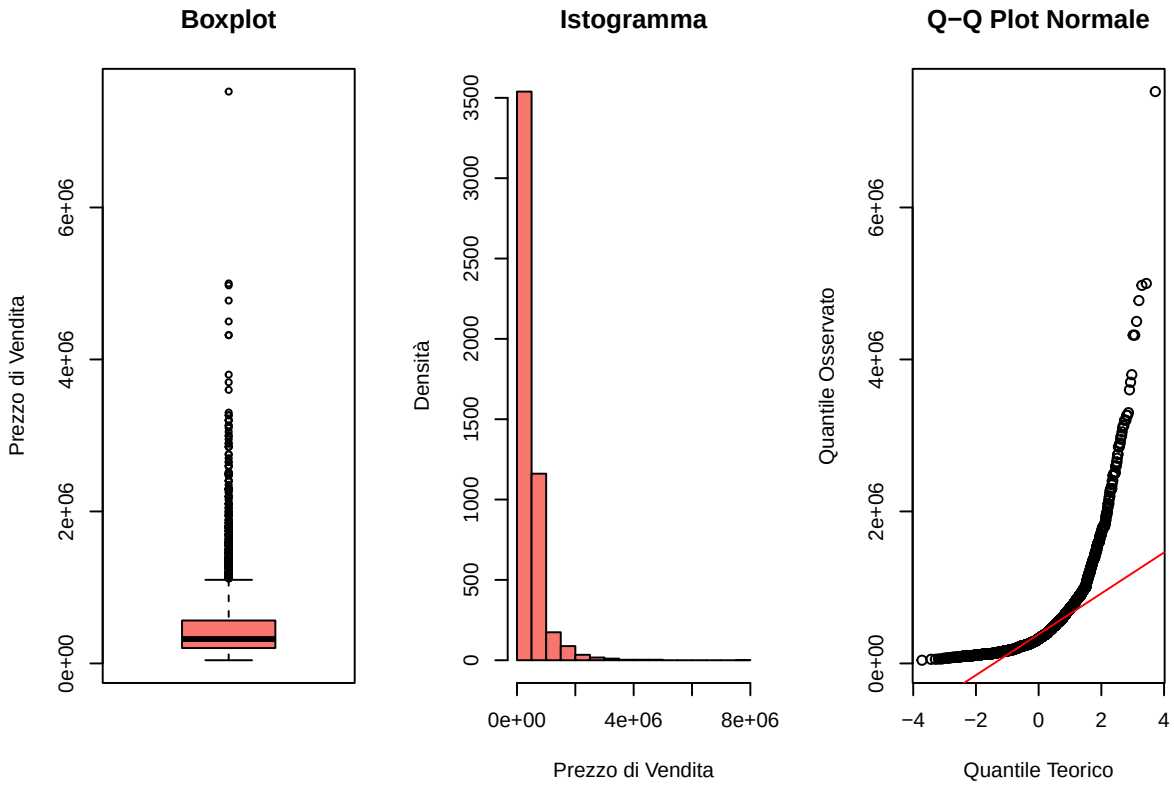
```
# Analisi Univariata

## Variabile Risposta Y (Prezzo di Vendita)

summary(buy_price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   42000  202000  321000  460105  565000 7525000
```

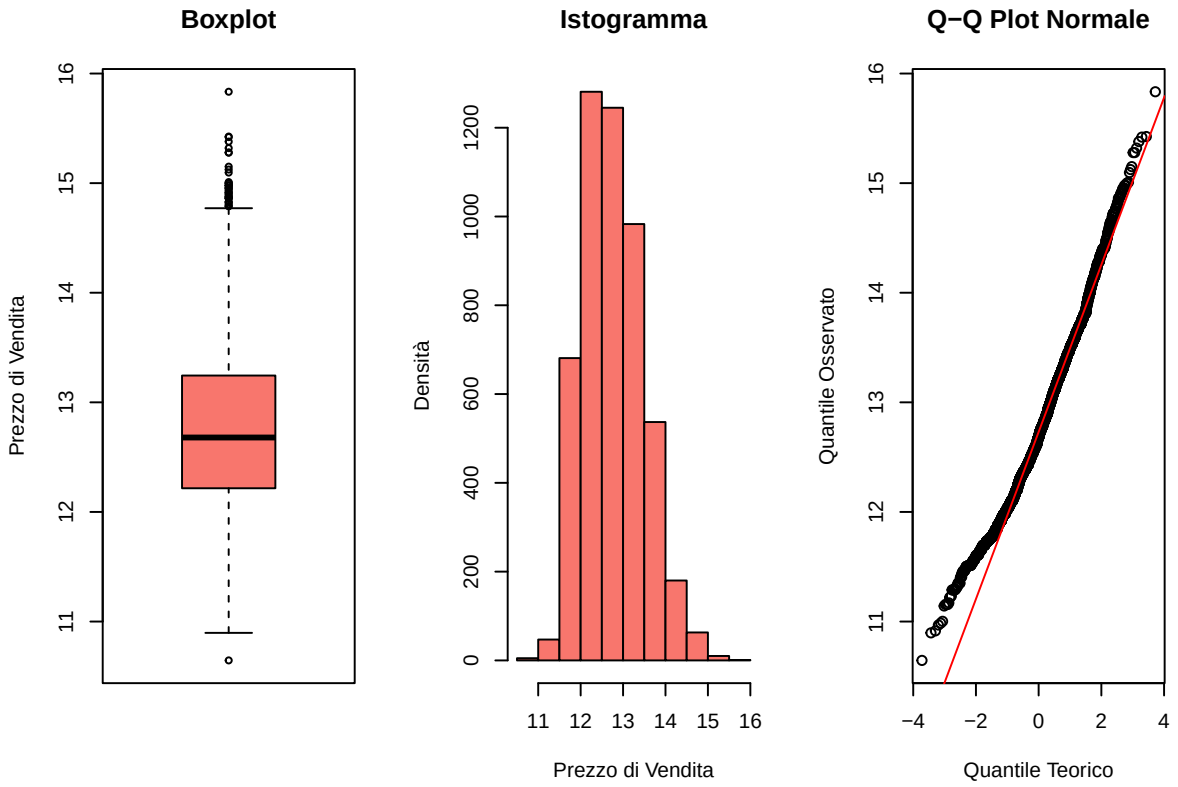
```
var(buy_price)
```

```
## [1] 196444909456
```

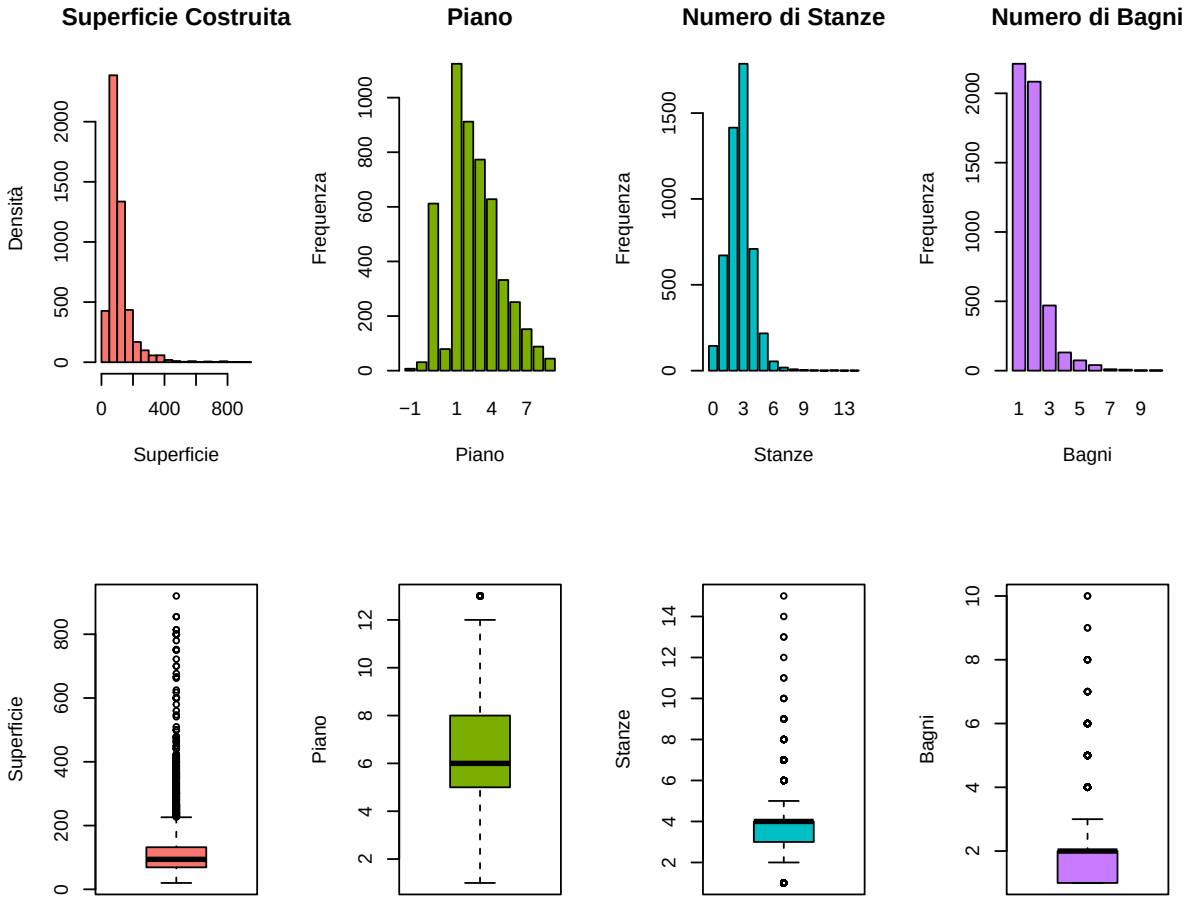
```
par(mfrow=c(1,3))
boxplot(buy_price,main='Boxplot',ylab='Prezzo di Vendita',col='#F8766D')
hist(buy_price,nclass=15,main='Istogramma',xlab='Prezzo di Vendita',ylab='Densità',col='#F8766D')
qqnorm(buy_price,main='Q-Q Plot Normale',xlab='Quantile Teorico',ylab='Quantile Osservato')
qqline(buy_price,col='red')
```



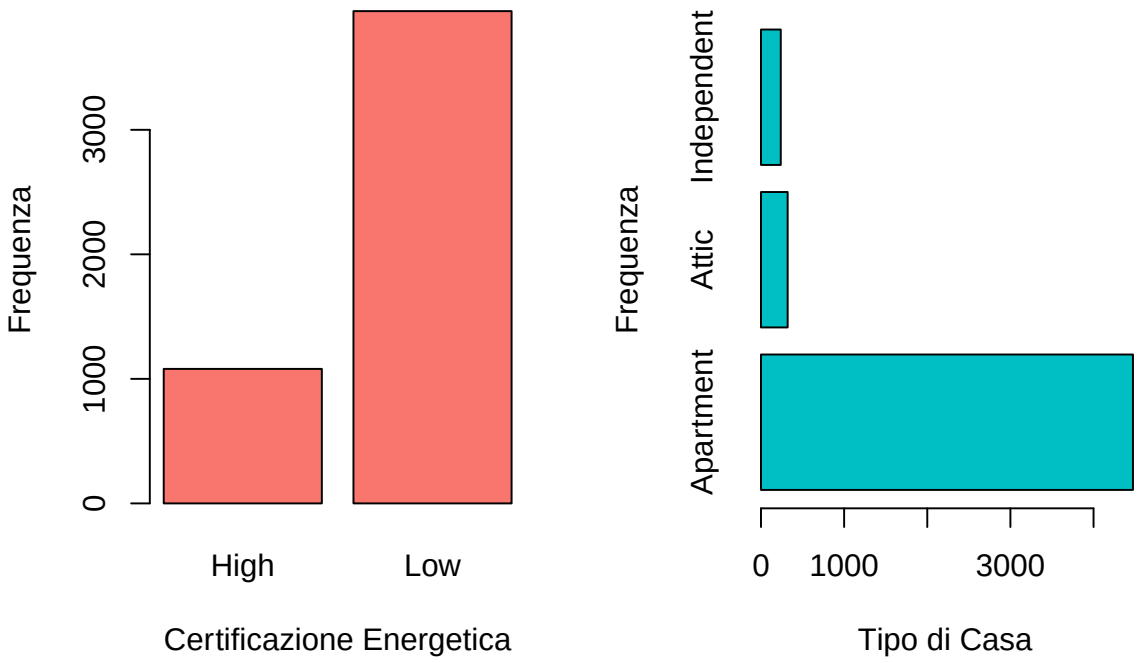

```
boxplot(log(buy_price),main='Boxplot',ylab='Prezzo di Vendita',col='#F8766D')
hist(log(buy_price),nclass=15,main='Istogramma',xlab='Prezzo di Vendita',ylab='Densità',col='#F8766D')
qqnorm(log(buy_price),main='Q-Q Plot Normale',xlab='Quantile Teorico',ylab='Quantile Osservato')
qqline(log(buy_price),col='red')
```



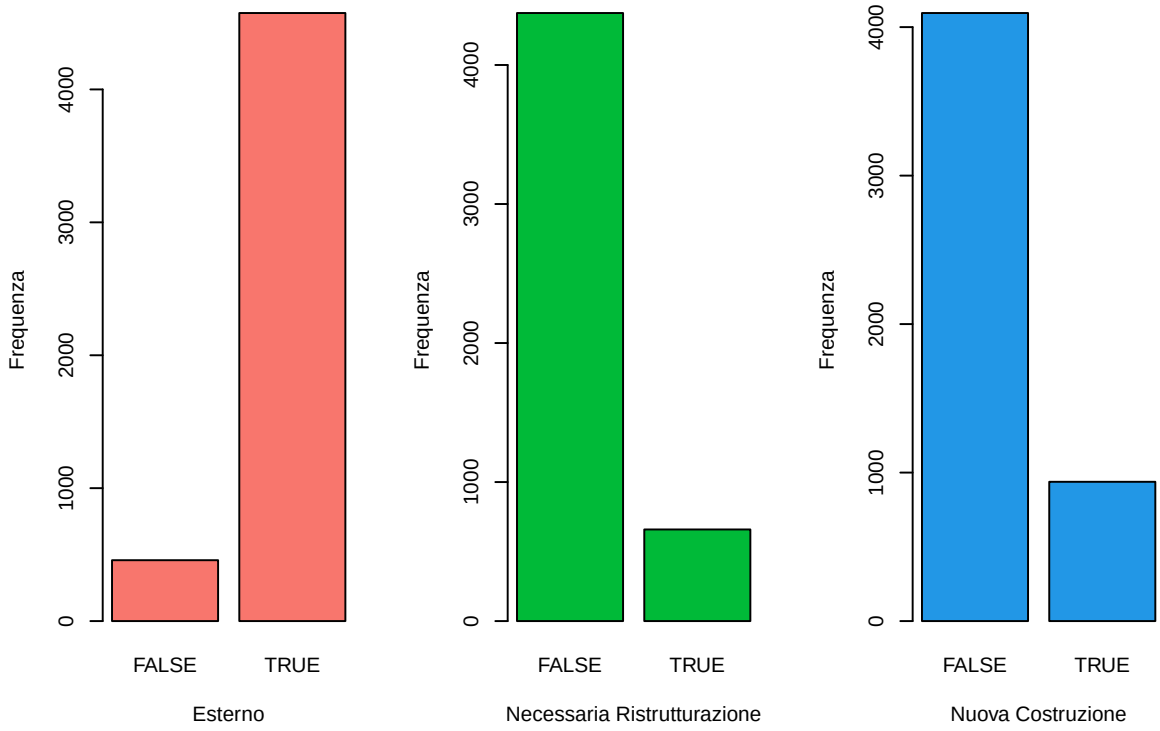

```
## Variabili Numeriche

par(mfrow=c(2,4))
hist(sq_mt_built,nclass=15,main='Superficie Costruita',
     xlab='Superficie',ylab='Densità',col='#F8766D')
plot(as.factor(floor),main='Piano',xlab='Piano',ylab='Frequenza',col='#7CAE00')
plot(as.factor(n_rooms),main='Numero di Stanze',xlab='Stanze',ylab='Frequenza',col='#00BFC4')
plot(as.factor(n_bathrooms),main='Numero di Bagni',xlab='Bagni',ylab='Frequenza',col='#C77CFF')

boxplot(sq_mt_built,ylab='Superficie',col='#F8766D')
boxplot(as.factor(floor),ylab='Piano',col='#7CAE00')
boxplot(as.factor(n_rooms),ylab='Stanze',col='#00BFC4')
boxplot(as.factor(n_bathrooms),ylab='Bagni',col='#C77CFF')
```



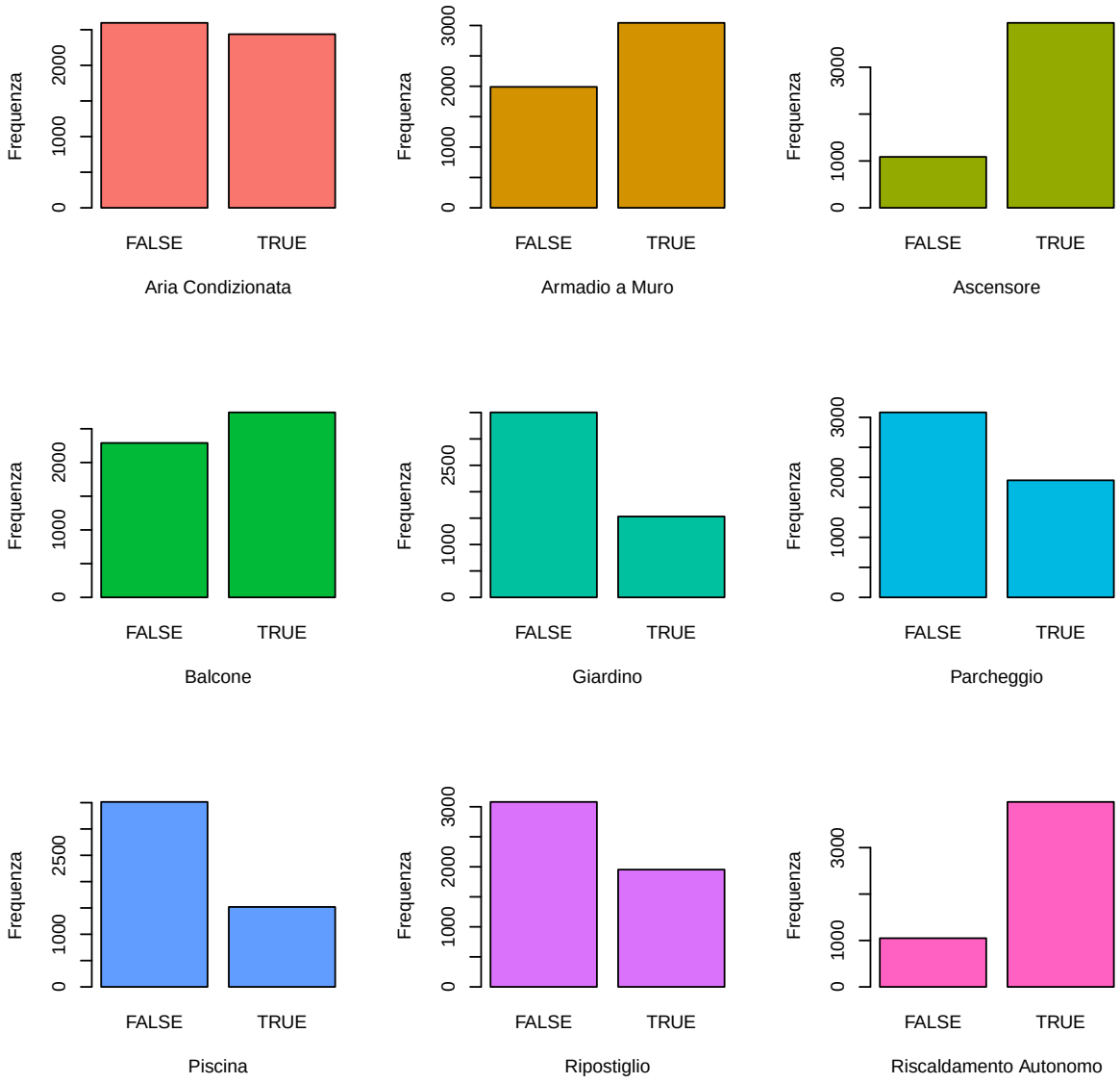

```
## Variabili Categoriche

par(mfrow=c(1,2))
plot(energy_certificate,xlab='Certificazione Energetica',ylab='Frequenza',col='#F8766D')
plot(house_type_id,xlab='Tipo di Casa',ylab='Frequenza',col='#00BFC4',horiz=TRUE)
```



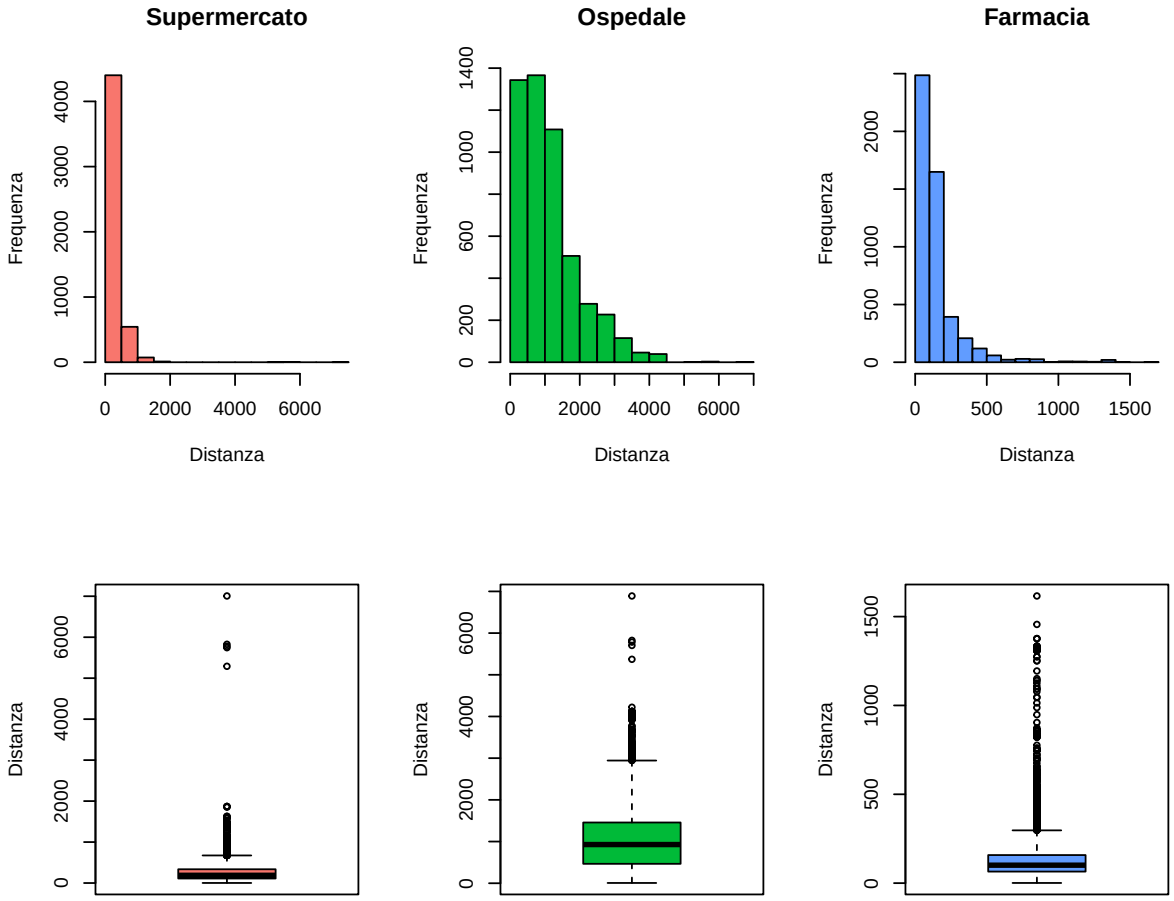

```
par(mfrow=c(1,3))
plot(is_exterior,xlab='Esterno',ylab='Frequenza',col='#F8766D')
plot(is_renewal_needed,xlab='Necessaria Ristrutturazione',ylab='Frequenza',col='#00BA38')
plot(is_new_development,xlab='Nuova Costruzione',ylab='Frequenza',col=4)
```



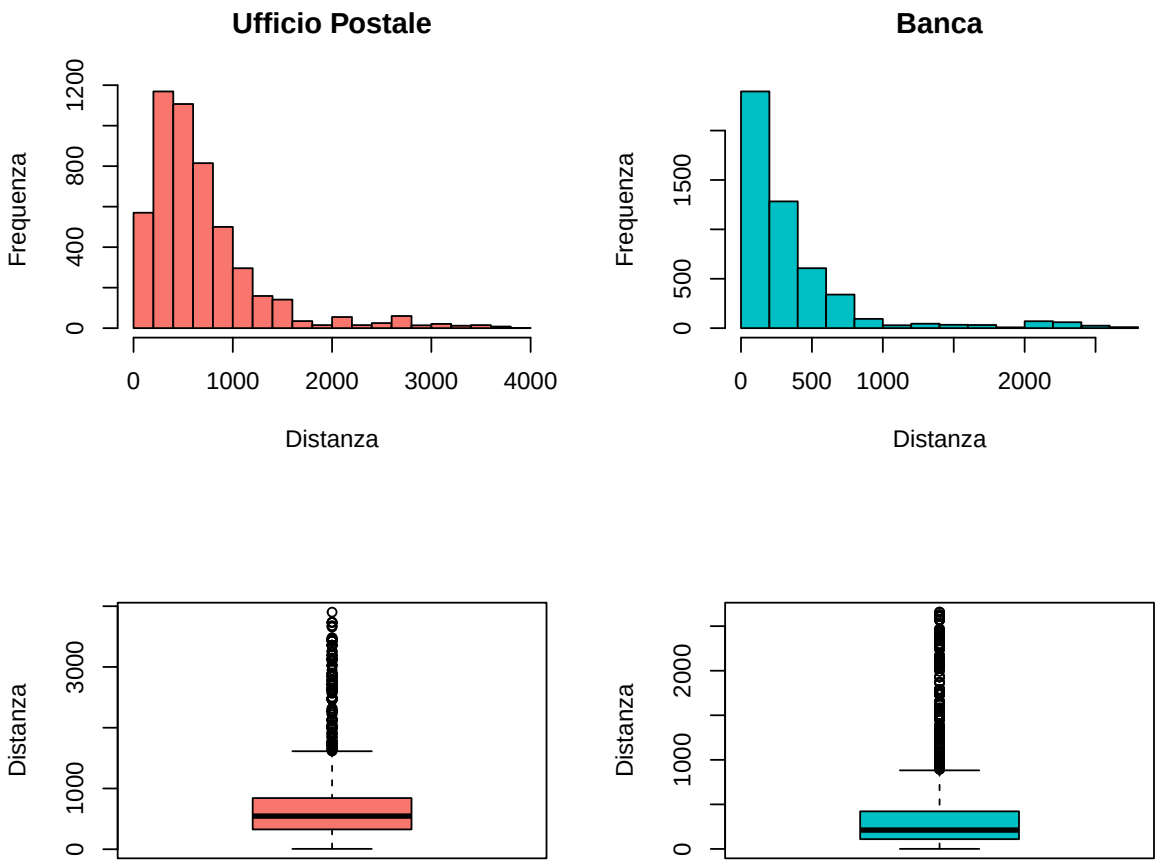

```
## Variabili di Presenza / Assenza

par(mfrow=c(3,3))
plot(has_ac,xlab='Aria Condizionata',ylab='Frequenza',col='#F8766D')
plot(has_fitted_wardrobes,xlab='Armadio a Muro',ylab='Frequenza',col='#D39200')
plot(has_lift,xlab='Ascensore',ylab='Frequenza',col='#93AA00')
plot(has_balcony,xlab='Balcone',ylab='Frequenza',col='#00BA38')
plot(has_garden,xlab='Giardino',ylab='Frequenza',col='#00C19F')
plot(has_parking,xlab='Parcheggio',ylab='Frequenza',col='#00B9E3')
plot(has_pool,xlab='Piscina',ylab='Frequenza',col='#619CFF')
plot(has_storage_room,xlab='Ripostiglio',ylab='Frequenza',col='#DB72FB')
plot(has_individual_heating,xlab='Riscaldamento Autonomo',ylab='Frequenza',col='#FF61C3')
```



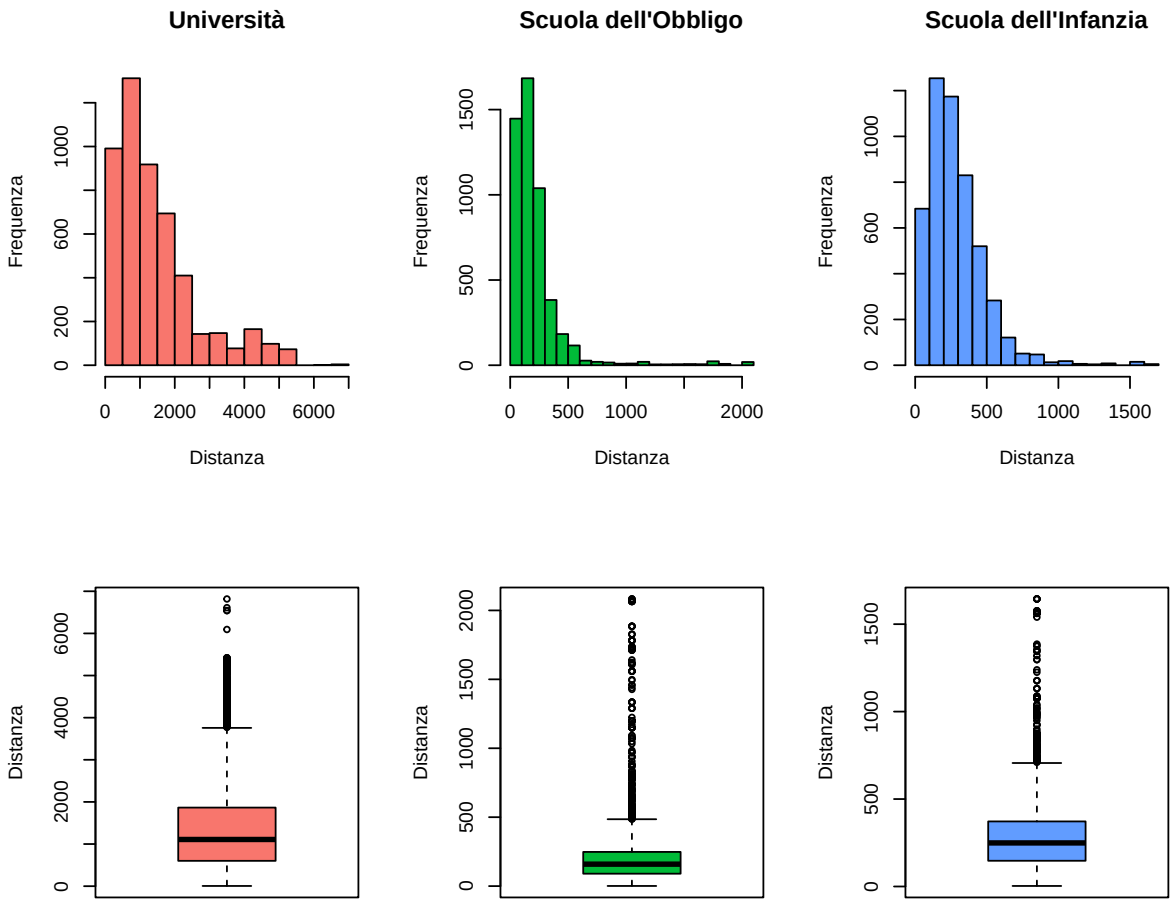

```
## Variabili di Distanza

par(mfrow=c(2,3))
hist(train_distance$d_supermarket,nclass=15,main='Supermercato',xlab='Distanza',ylab='Frequenza',col='#
hist(train_distance$d_hospital,nclass=15,main='Ospedale',xlab='Distanza',ylab='Frequenza',col='#00BA38'
hist(train_distance$d_pharmacy,nclass=15,main='Farmacia',xlab='Distanza',ylab='Frequenza',col='#619CFF'
boxplot(train_distance$d_supermarket,ylab='Distanza',col='#F8766D')
boxplot(train_distance$d_hospital,ylab='Distanza',col='#00BA38')
boxplot(train_distance$d_pharmacy,ylab='Distanza',col='#619CFF')
```



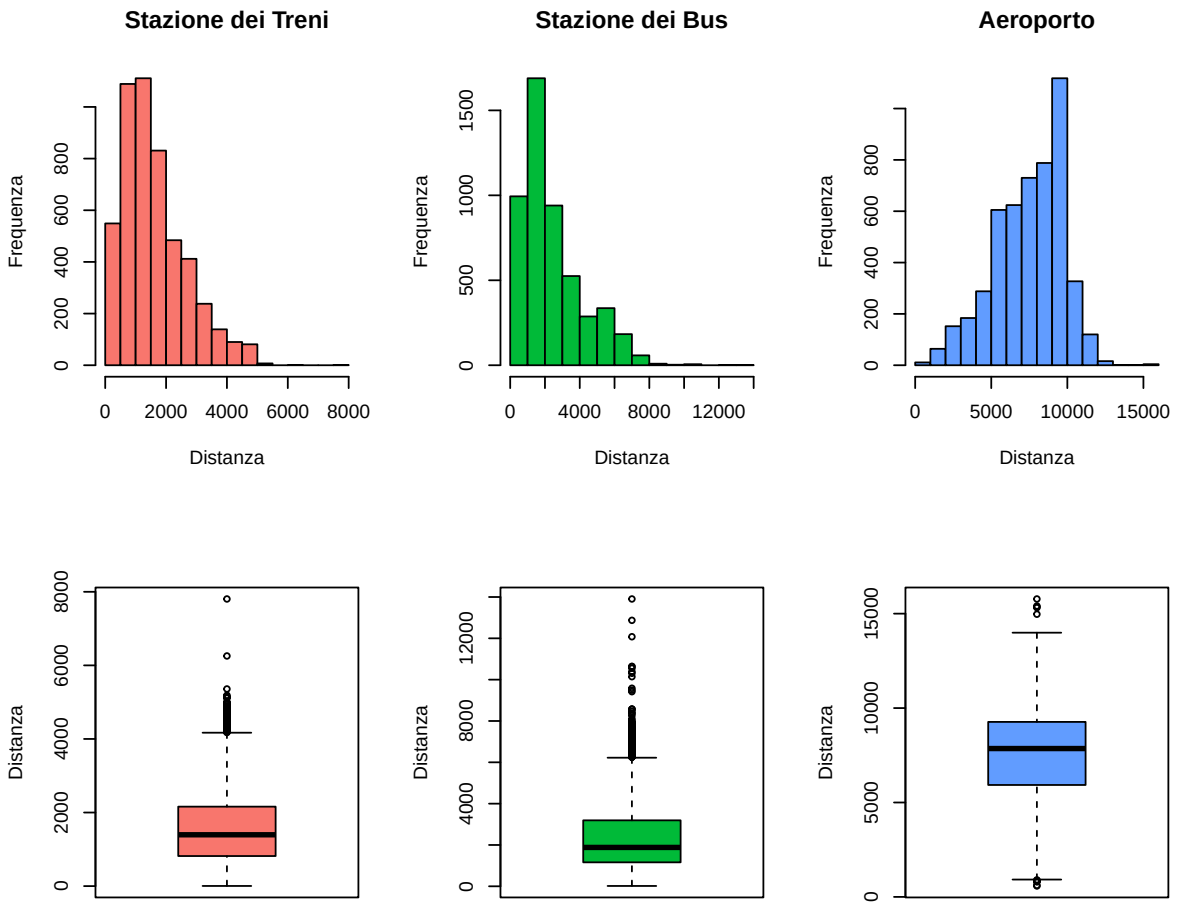

```
par(mfrow=c(2,2))
hist(train_distance$d_post,nclass=15,main='Ufficio Postale',xlab='Distanza',ylab='Frequenza',col='#F876
hist(train_distance$d_bank,nclass=15,main='Banca',xlab='Distanza',ylab='Frequenza',col='#00BFC4')
boxplot(train_distance$d_post,ylab='Distanza',col='#F8766D')
boxplot(train_distance$d_bank,ylab='Distanza',col='#00BFC4')
```



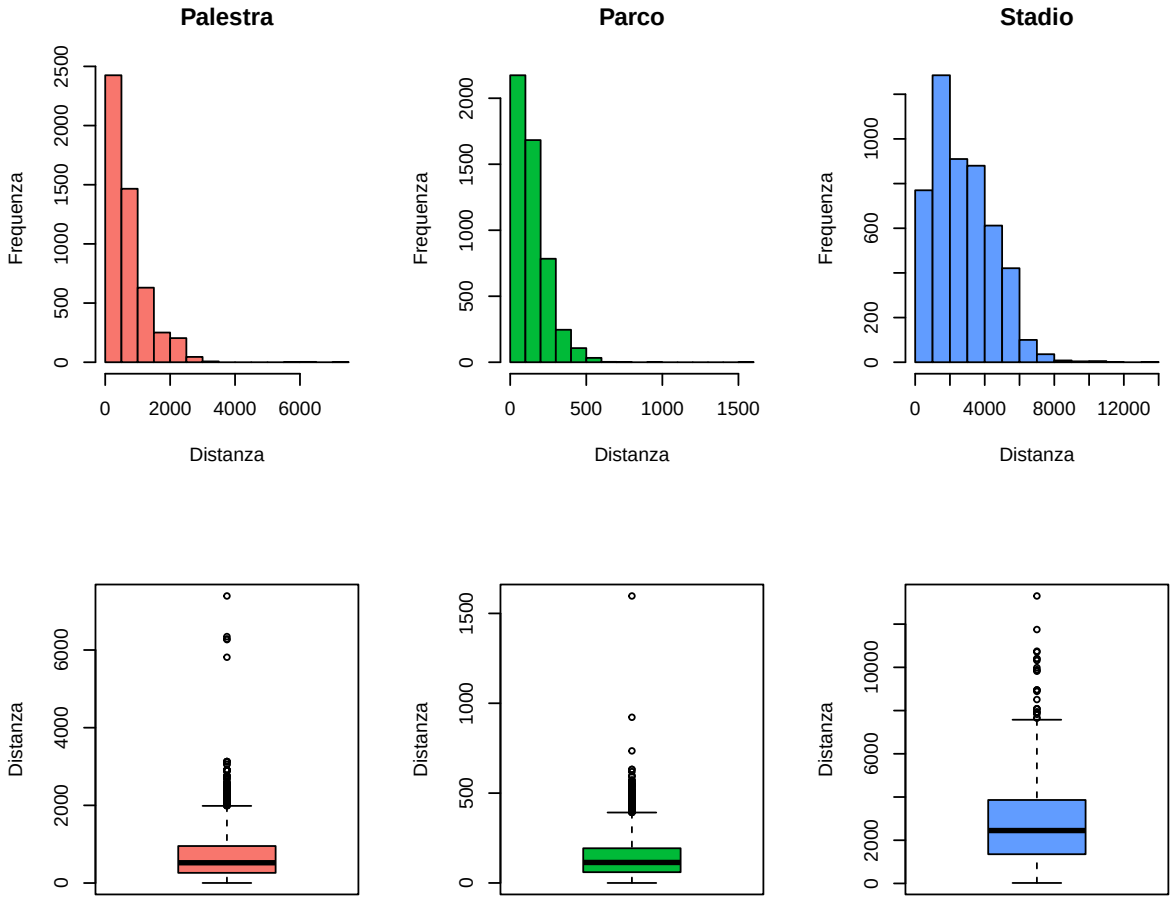

```
par(mfrow=c(2,3))
hist(train_distance$d_university,nclass=15,main='Università',
     xlab='Distanza',ylab='Frequenza',col='#F8766D')
hist(train_distance$d_school,nclass=15,main="Scuola dell'Obbligo",
     xlab='Distanza',ylab='Frequenza',col='#00BA38')
hist(train_distance$d_kindergarten,nclass=15,main="Scuola dell'Infanzia",
     xlab='Distanza',ylab='Frequenza',col='#619CFF')
boxplot(train_distance$d_university,ylab='Distanza',col='#F8766D')
boxplot(train_distance$d_school,ylab='Distanza',col='#00BA38')
boxplot(train_distance$d_kindergarten,ylab='Distanza',col='#619CFF')
```



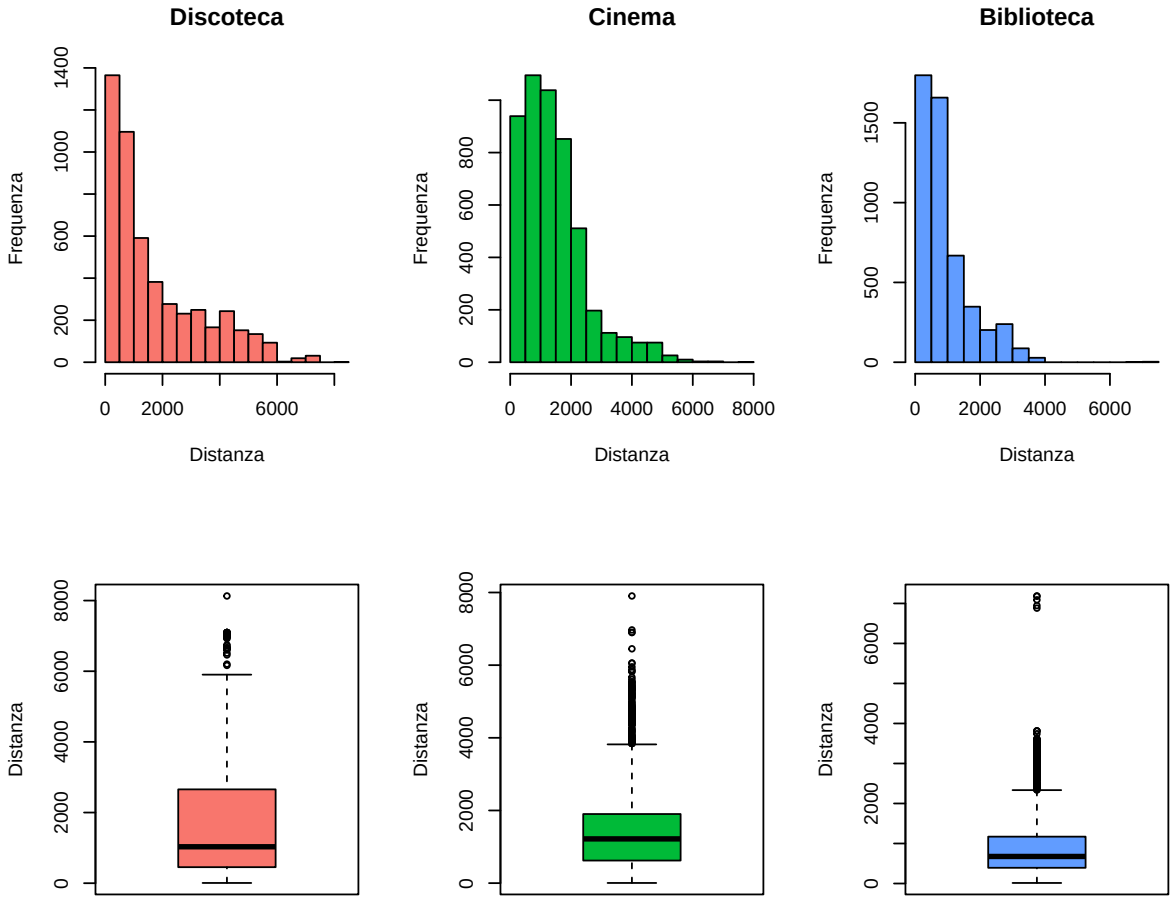

```
par(mfrow=c(2,3))
hist(train_distance$d_train,nclass=15,main='Stazione dei Treni',
     xlab='Distanza',ylab='Frequenza',col='#F8766D')
hist(train_distance$d_bus,nclass=15,main='Stazione dei Bus',
     xlab='Distanza',ylab='Frequenza',col='#00BA38')
hist(train_distance$d_airport,nclass=15,main='Aeroporto',
     xlab='Distanza',ylab='Frequenza',col='#619CFF')
boxplot(train_distance$d_train,ylab='Distanza',col='#F8766D')
boxplot(train_distance$d_bus,ylab='Distanza',col='#00BA38')
boxplot(train_distance$d_airport,ylab='Distanza',col='#619CFF')
```



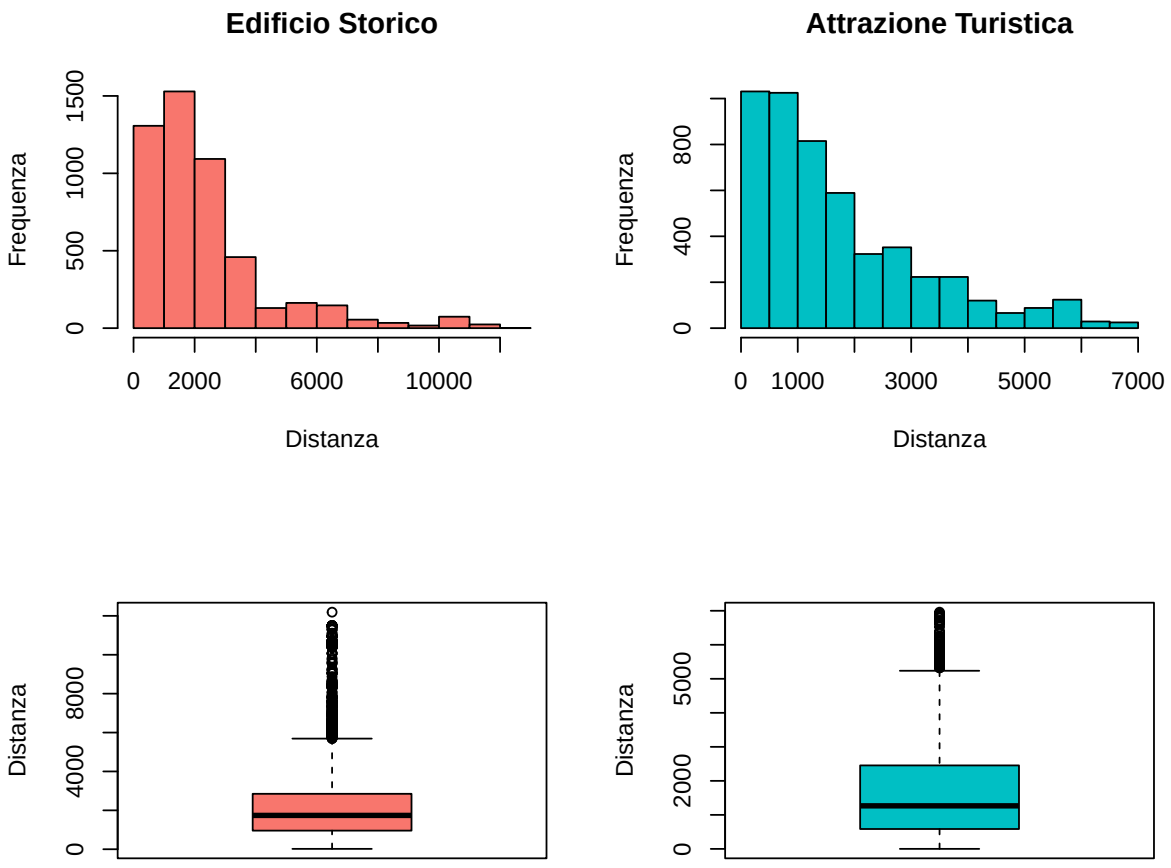

```
par(mfrow=c(2,3))
hist(train_distance$d_gym,nclass=15,main='Palestra',xlab='Distanza',ylab='Frequenza',col='#F8766D')
hist(train_distance$d_park,nclass=15,main='Parco',xlab='Distanza',ylab='Frequenza',col='#00BA38')
hist(train_distance$d_stadium,nclass=15,main='Stadio',xlab='Distanza',ylab='Frequenza',col='#619CFF')
boxplot(train_distance$d_gym,ylab='Distanza',col='#F8766D')
boxplot(train_distance$d_park,ylab='Distanza',col='#00BA38')
boxplot(train_distance$d_stadium,ylab='Distanza',col='#619CFF')
```



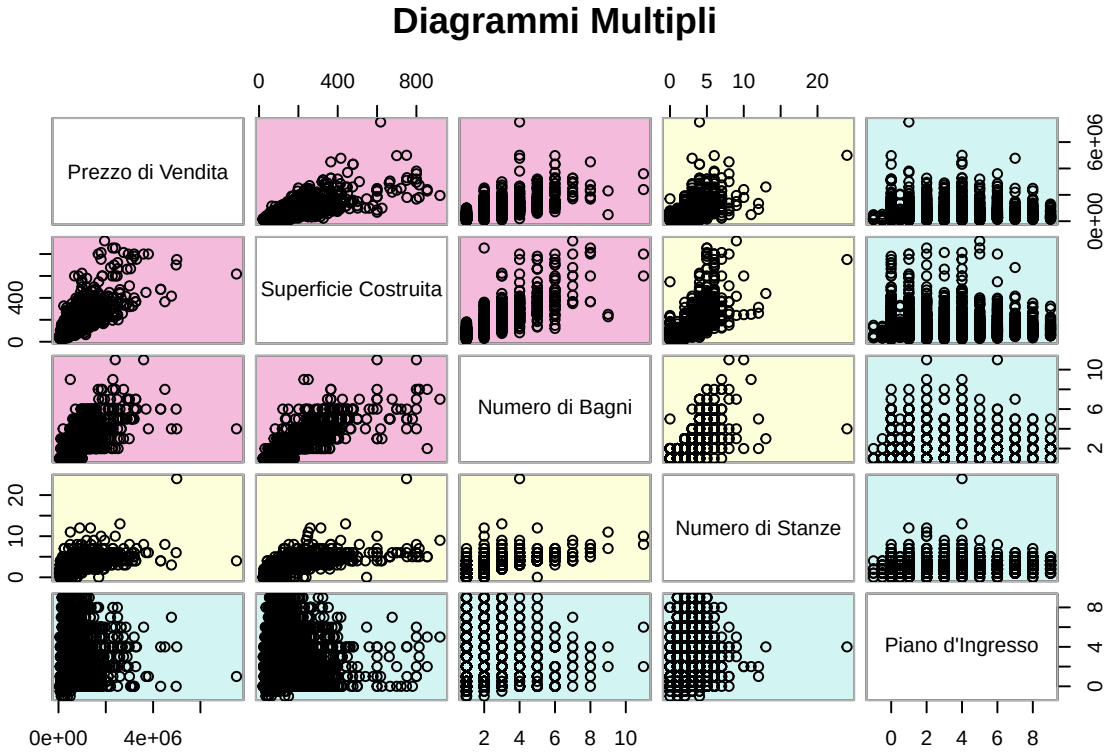

```
par(mfrow=c(2,3))
hist(train_distance$d_disco,nclass=15,main='Discoteca',xlab='Distanza',ylab='Frequenza',col='#F8766D')
hist(train_distance$d_cinema,nclass=15,main='Cinema',xlab='Distanza',ylab='Frequenza',col='#00BA38')
hist(train_distance$d_library,nclass=15,main='Biblioteca',xlab='Distanza',ylab='Frequenza',col='#619CFF
boxplot(train_distance$d_disco,ylab='Distanza',col='#F8766D')
boxplot(train_distance$d_cinema,ylab='Distanza',col='#00BA38')
boxplot(train_distance$d_library,ylab='Distanza',col='#619CFF')
```




```
par(mfrow=c(2,2))
hist(train_distance$d_historic,nclass=15,main='Edificio Storico',
     xlab='Distanza',ylab='Frequenza',col='#F8766D')
hist(train_distance$d_attraction,nclass=15,main='Attrazione Turistica',
     xlab='Distanza',ylab='Frequenza',col='#00BFC4')
boxplot(train_distance$d_historic,ylab='Distanza',col='#F8766D')
boxplot(train_distance$d_attraction,ylab='Distanza',col='#00BFC4')
```




```
# Analisi Bivariata

## Numeriche - Numeriche

train_num=train[,sapply(train,is.numeric)]
corr=cor(train_num)
colors=dmat.color(abs(corr),breaks=c(0,0.3,0.7,1),col=c('#D2F4F2','#FDFFDA','#F4BBDD'))
colnames(train_num)=c('Prezzo di Vendita','Superficie Costruita','Numero di Bagni',
                      'Numero di Stanze',"Piano d'Ingresso")
cpairs(train_num,panel.colors=colors,gap=0.5,main='Diagrammi Multipli')  # scatterplots
```




```
round(corr,3)  # correlazione lineare di Pearson
```

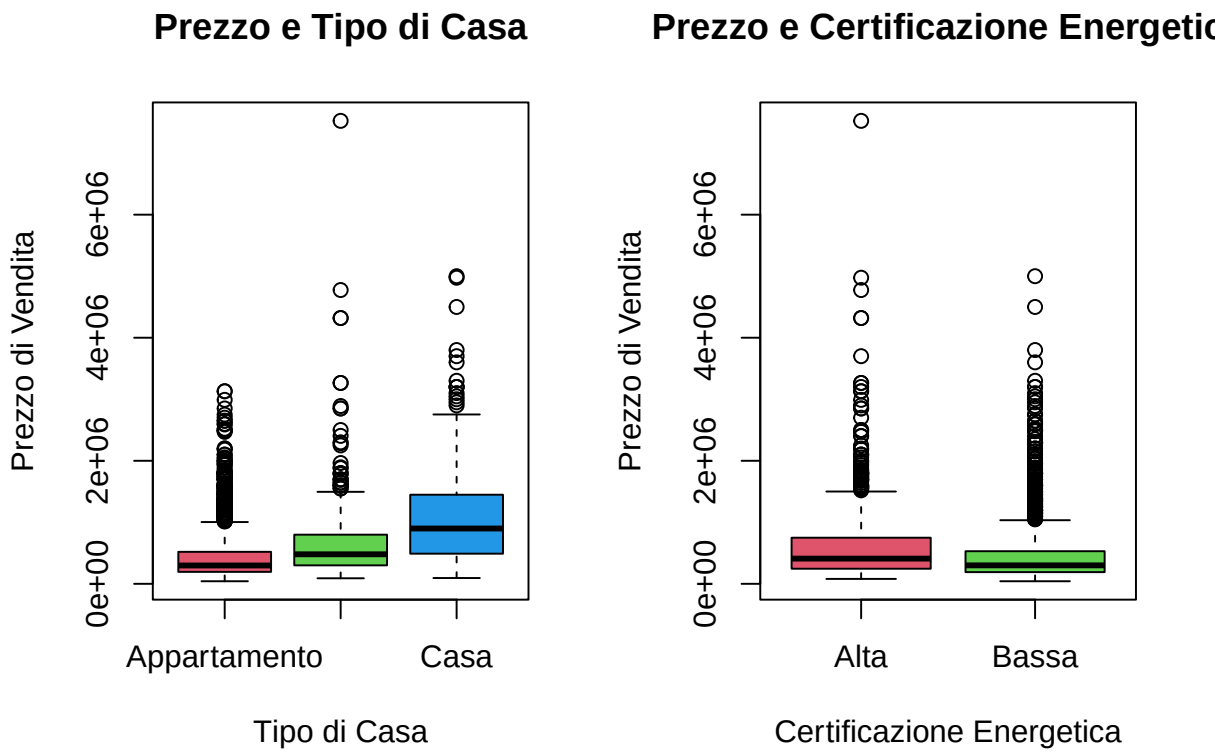
```
##             buy_price sq_mt_built n_bathrooms n_rooms floor
## buy_price       1.000       0.793       0.732   0.509 0.105
## sq_mt_built     0.793       1.000       0.820   0.660 0.065
## n_bathrooms     0.732       0.820       1.000   0.635 0.074
## n_rooms         0.509       0.660       0.635   1.000 0.119
## floor           0.105       0.065       0.074   0.119 1.000
```

```
## Numeriche - Categoriche

par(mfrow=c(1,2))
boxplot(buy_price~house_type_id,main='Prezzo e Tipo di Casa',xlab='Tipo di Casa',
        ylab='Prezzo di Vendita',names=c('Appartamento','Attico','Casa'),col=c(2,3,4))
summary(lm(buy_price~house_type_id))
```

```
##
## Call:
## lm(formula = buy_price ~ house_type_id)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1045116  -226672  -116672   115328  6823609
##
## Coefficients:
```

```
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 406672       6136   66.28   <2e-16 ***
## house_type_idAttic          294719      23716   12.43   <2e-16 ***
## house_type_idIndependent    732444      27303   26.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 410400 on 5030 degrees of freedom
## Multiple R-squared:  0.1428, Adjusted R-squared:  0.1425
## F-statistic: 419.1 on 2 and 5030 DF,  p-value: < 2.2e-16
```

```
boxplot(buy_price~energy_certificate,main='Prezzo e Certificazione Energetica',
        xlab='Certificazione Energetica',ylab='Prezzo di Vendita',names=c('Alta','Bassa'),col=c(2,3))
```

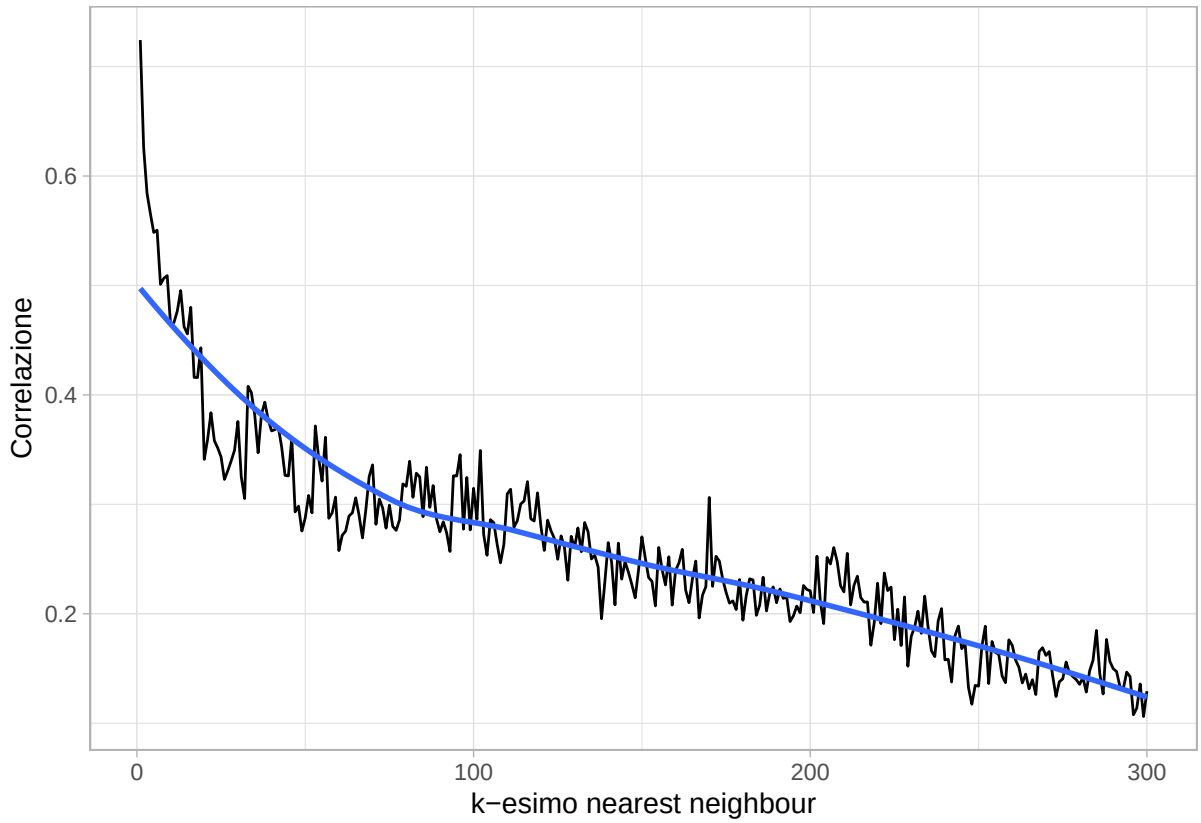



```
summary(lm(buy_price~energy_certificate))
```

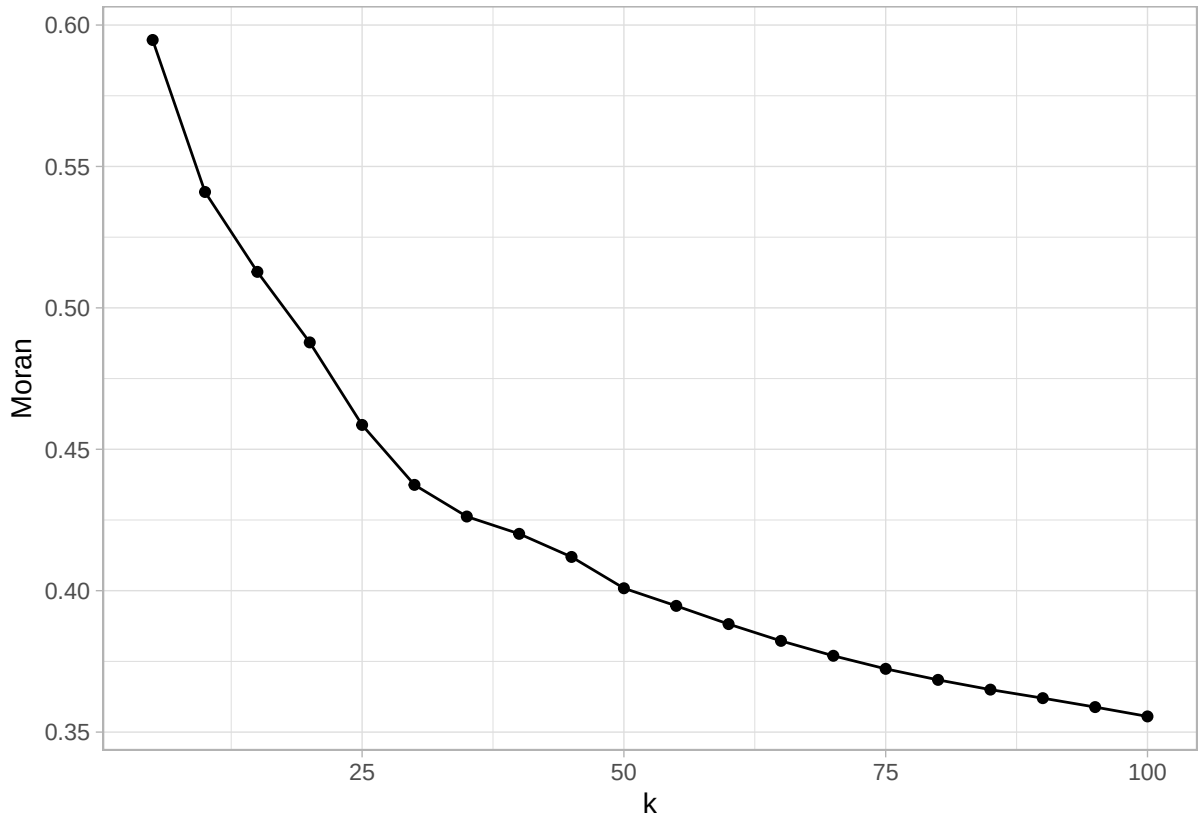
```
##
## Call:
## lm(formula = buy_price ~ energy_certificate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -516764 -247768 -127768  112232 6928236
##
## Coefficients:
```

```
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               596764      13312   44.83   <2e-16 ***
## energy_certificateLow    -173996      15021  -11.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 437500 on 5031 degrees of freedom
## Multiple R-squared:  0.02598,    Adjusted R-squared:  0.02579
## F-statistic: 134.2 on 1 and 5031 DF,  p-value: < 2.2e-16
```

```r
# Esplorazione Spaziale

## knearneigh
knear=knearneigh(train_coord,k=300,longlat=TRUE)  # numero di vicini k fissato
x=train$buy_price
r=sapply(1:300,function(i){
  cor(x,x[knear$nn[,i]])
})

data.frame(k=1:300,r=r) %>%  # correlazione al variare di k -> correlazione 0,3 per k=75
  ggplot(aes(x=k,y=r)) +
  geom_line() +
  geom_smooth(se=FALSE) +
  xlab('k-esimo nearest neighbour') +
  ylab('Correlazione') +
  theme_light()
```

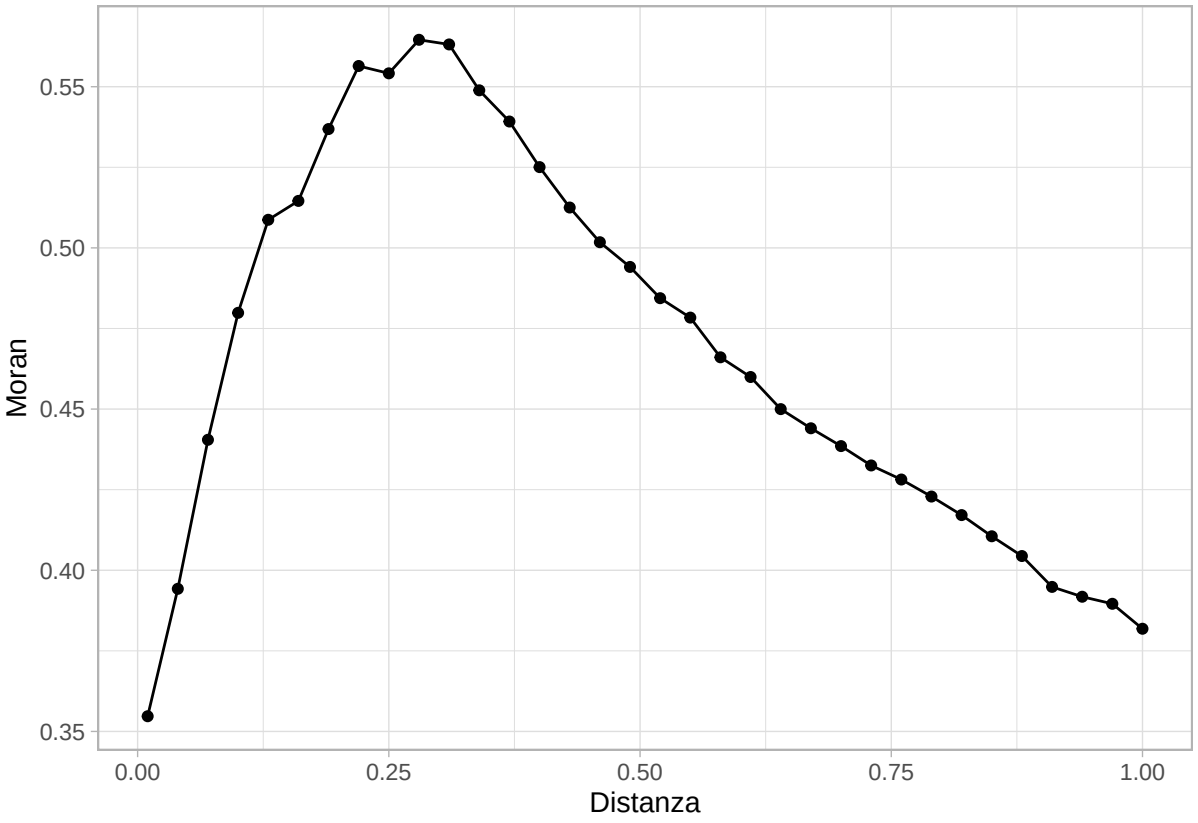


```
k_moran_I=c()  # indice di Moran al variare di k
for (i in seq(5,100,5)){
  k_neigh=knn2nb(knearneigh(train_coord,k=i,longlat=TRUE))
  k_distance=nbdists(k_neigh,train_coord,longlat=TRUE)
  inv_k_distance=lapply(k_distance,function(x)(1/(x+1)))
  k_weight=nb2listw(k_neigh,glist=inv_k_distance,style='W')
  k_moran=moran.mc(train$buy_price,k_weight,nsim=100,alternative='two.sided')
  k_moran_I=c(k_moran_I,k_moran$statistic)
}
moran_I_k=data.frame(moran=k_moran_I,k=seq(5,100,5))

ggplot(moran_I_k,aes(x=k,y=moran)) +
  geom_point() +
  geom_line() +
  xlab('k') +
  ylab('Moran') +
  theme_light()
```

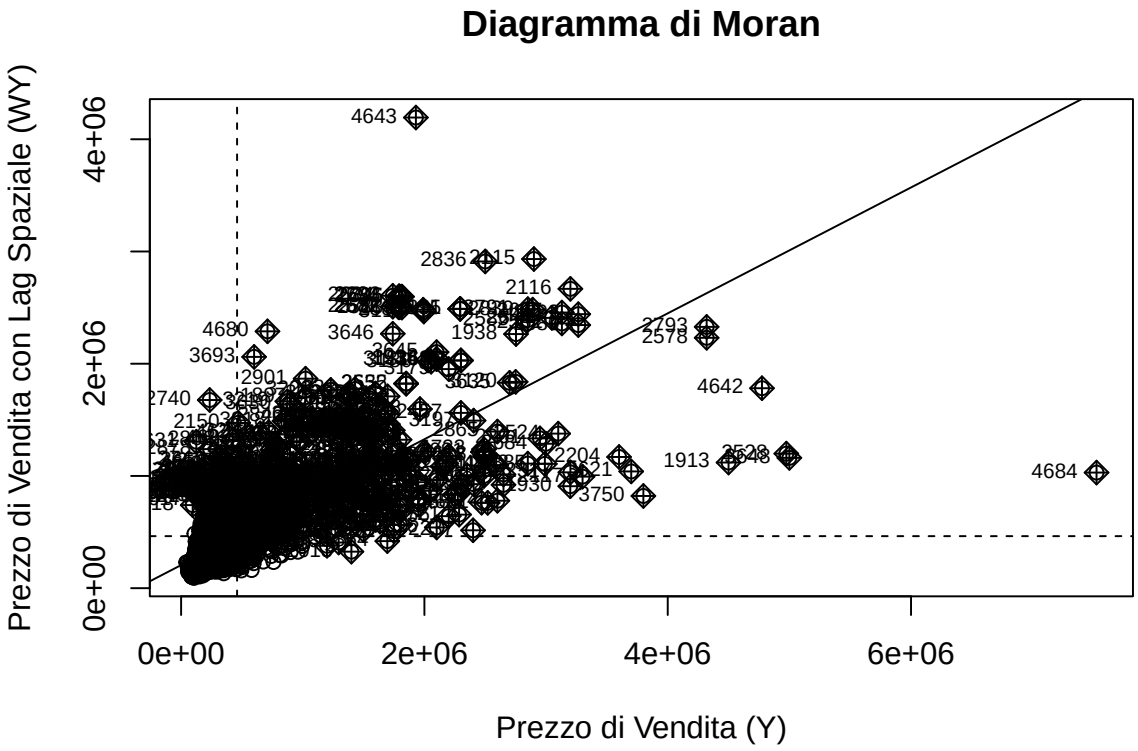



```
## dnearneigh
d_moran_I=c()  # indice di Moran al variare di una distanza soglia d
for (d in seq(0.01,1,0.03)){
  d_neigh=dnearneigh(train_coord,0,d,longlat=TRUE)
  d_distance=nbdists(d_neigh,train_coord,longlat=TRUE)
  inv_d_distance=lapply(d_distance, function(x)(1/(x+1)))
  d_weight=nb2listw(d_neigh,glist=inv_d_distance,style='W',zero.policy=TRUE)
```

```
  d_moran=moran.mc(train$buy_price,d_weight,nsim=100,alternative='two.sided',zero.policy=TRUE)
  d_moran_I=c(d_moran_I,d_moran$statistic)
}
moran_I_d=data.frame(moran=d_moran_I,distance=seq(0.01,1,0.03))

ggplot(moran_I_d,aes(x=distance,y=moran)) +
  geom_point() +
  geom_line() +
  xlab('Distanza') +
  ylab('Moran') +
  theme_light()
```




```
k_neigh_75=knn2nb(knearneigh(train_coord,k=75,longlat=TRUE))  # k=75
k_distance_75=nbdists(k_neigh_75,train_coord,longlat=TRUE)
inv_k_distance_75=lapply(k_distance_75, function(x) (1/(x+0.001)))
k_weight_75=nb2listw(k_neigh_75,glist=inv_k_distance_75,style='W')

moran.plot(train$buy_price,k_weight_75,xlab='Prezzo di Vendita (Y)',
           ylab='Prezzo di Vendita con Lag Spaziale (WY)',main='Diagramma di Moran')
```




```
moran.mc(train$buy_price,k_weight_75,nsim=100,alternative='two.sided')
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  train$buy_price
## weights: k_weight_75
## number of simulations + 1: 101
##
## statistic = 0.5608, observed rank = 101, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

```
geary.mc(train$buy_price,k_weight_75,nsim=100,alternative='two.sided')
```

```
##
##  Monte-Carlo simulation of Geary C
##
## data:  train$buy_price
## weights: k_weight_75
## number of simulations + 1: 101
##
## statistic = 0.40296, observed rank = 1, p-value = 0.0198
## alternative hypothesis: two.sided
```

```
d_neigh_75=dnearneigh(train_coord,0,0.8,longlat=TRUE)  # d=0,8 km -> n.medio di vicini=75
d_distance_75=nbdists(d_neigh_75,train_coord,longlat=TRUE)
inv_d_distance_75=lapply(d_distance_75, function(x) (1/(x+0.001)))
d_weight_75=nb2listw(d_neigh_75,glist=inv_d_distance_75,style='W',zero.policy=TRUE)
d_weight_75$neighbours
```

```
## Neighbour list object:
## Number of regions: 5033
## Number of nonzero links: 391178
## Percentage nonzero weights: 1.54426
## Average number of links: 77.72263
## 5 regions with no links:
## 331 2223 2462 2648 4994
```

```
moran.mc(train$buy_price,d_weight_75,nsim=100,alternative='two.sided',zero.policy=TRUE)
```

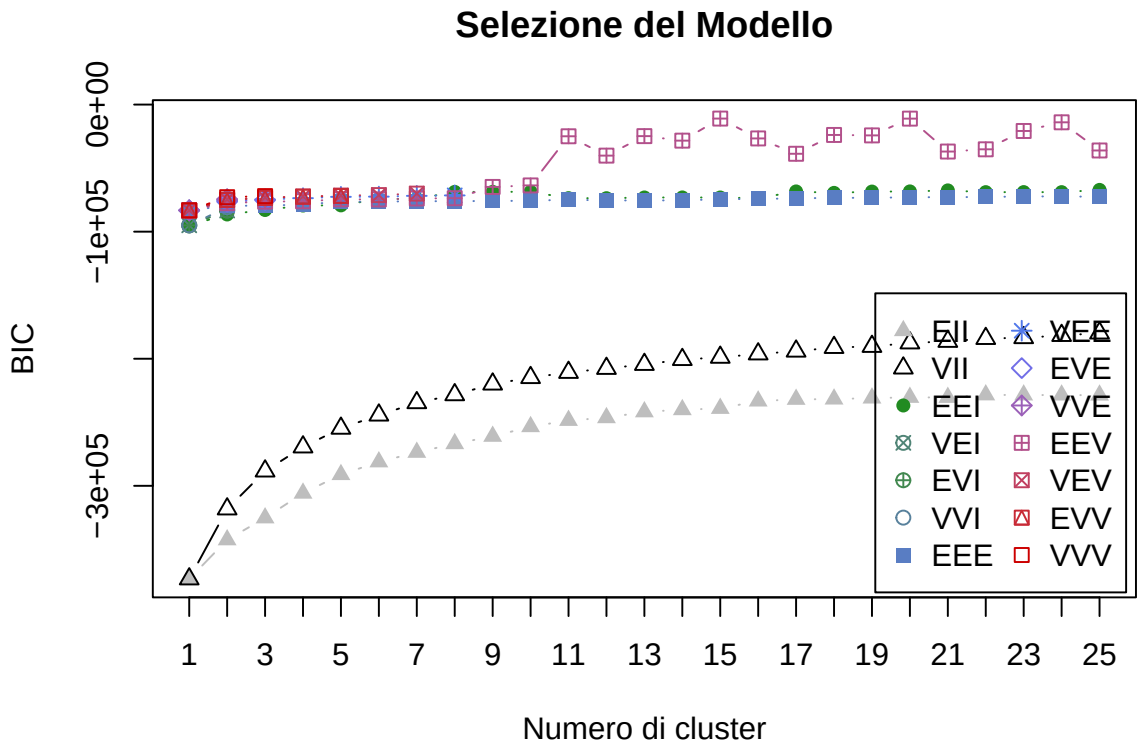
```
##
##  Monte-Carlo simulation of Moran I
##
## data:  train$buy_price
## weights: d_weight_75
## number of simulations + 1: 101
##
## statistic = 0.57371, observed rank = 101, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

```
geary.mc(train$buy_price,d_weight_75,nsim=100,alternative='two.sided',zero.policy=TRUE)
```

```
##
##  Monte-Carlo simulation of Geary C
##
## data:  train$buy_price
## weights: d_weight_75
## number of simulations + 1: 101
##
## statistic = 0.39189, observed rank = 1, p-value = 0.0198
## alternative hypothesis: two.sided
```

```
# Variabili Spaziali

## Distretti

train_c_district=as.data.frame(cbind(train,train_distance,train_district))
test_c_district=as.data.frame(cbind(test,test_distance,test_district))
colnames(train_c_district)[39]='district'
colnames(test_c_district)[39]='district'
train_c_district$district=as.factor(train_c_district$district)
test_c_district$district=as.factor(test_c_district$district)


## Cluster
```

```
data_cluster=as.data.frame(cbind(data[,c(2,3,4,6)],data_coord))

set.seed(123)
fit_mm=mclustBIC(data_cluster,verbose=FALSE,G=1:25,  # scelta del numero di cluster
                 initialization=list(hcRandomPairs(data_cluster)))
plot(fit_mm,xlab="Numero di cluster")
title(main="Selezione del Modello")
```




```
fit_mm
```

```
## Bayesian Information Criterion (BIC):
##          EII       VII       EEI       VEI       EVI       VVI       EEE
## 1  -373297.2 -373297.2 -95024.96 -95024.96 -95024.96 -95024.96 -83257.53
## 2  -342529.6 -318249.7 -86465.47 -83368.57 -83083.37 -80824.90 -79643.90
## 3  -325272.5 -288402.6 -82936.95        NA        NA        NA -79540.82
## 4  -305833.8 -269470.3 -79425.45        NA        NA        NA -78668.82
## 5  -291205.8 -254745.1 -79297.28        NA        NA        NA -76435.14
## 6  -281193.8 -244373.6 -74297.52        NA        NA        NA -76458.41
## 7  -273679.4 -234641.9 -74183.43        NA        NA        NA -76004.81
## 8  -266868.5 -228380.9 -68729.72        NA        NA        NA -76060.03
## 9  -261078.1 -219978.7 -68698.39        NA        NA        NA -75800.68
## 10 -253493.0 -214875.9 -68612.95        NA        NA        NA -75735.09
## 11 -248557.4 -210786.7 -73650.87        NA        NA        NA -74761.87
## 12 -246446.7 -207652.6 -73694.25        NA        NA        NA -75496.37
```

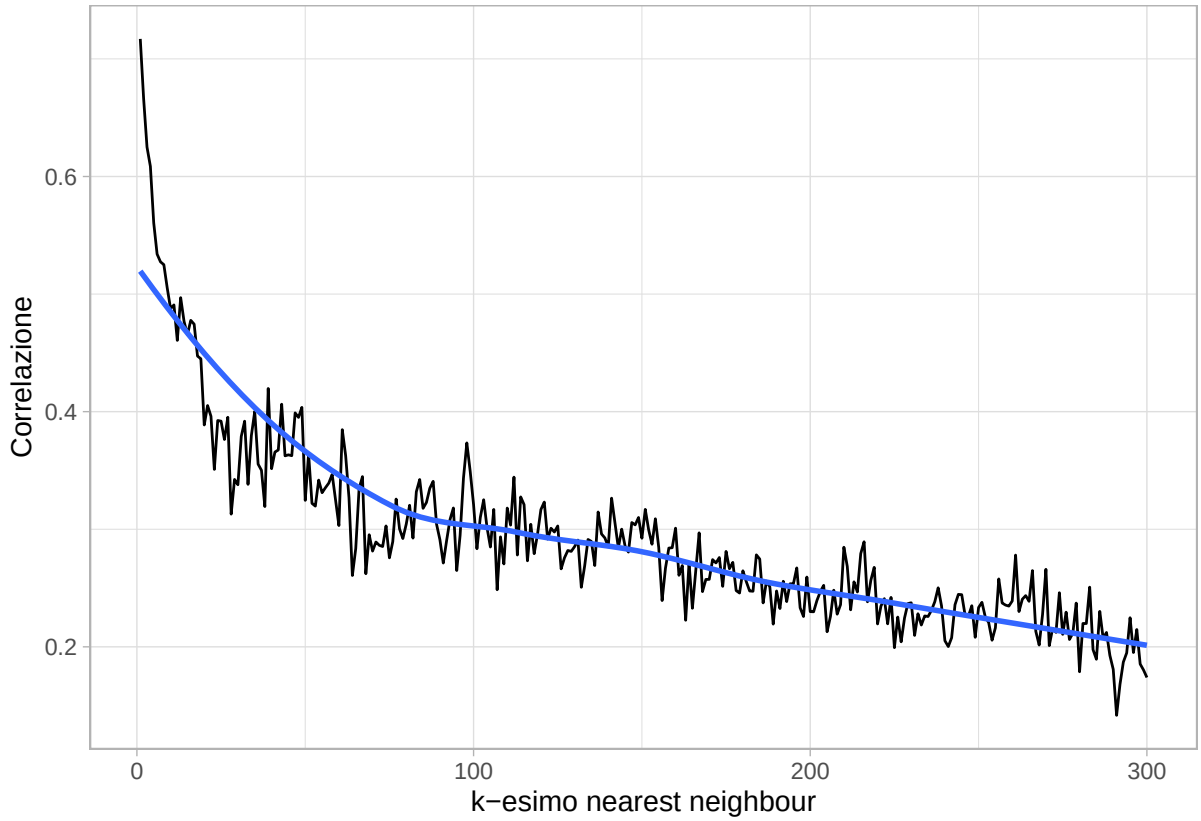
```
## 13 -241795.0 -204527.4 -73116.09        NA        NA        NA -75376.87
## 14 -240118.4 -200684.9 -73108.97        NA        NA        NA -75433.86
## 15 -239106.4 -198895.3 -73131.53        NA        NA        NA -74420.88
## 16 -233302.7 -196258.3 -73713.30        NA        NA        NA -74272.89
## 17 -231981.4 -194059.3 -68382.24        NA        NA        NA -73811.46
## 18 -231754.5 -191172.8 -69512.63        NA        NA        NA -73364.86
## 19 -230961.6 -190332.9 -68492.05        NA        NA        NA -73421.95
## 20 -230428.8 -187604.6 -68354.56        NA        NA        NA -72896.63
## 21 -230467.8 -186369.7 -67557.01        NA        NA        NA -72887.08
## 22 -228483.1 -184105.6 -68933.08        NA        NA        NA -72710.02
## 23 -228535.5 -183457.2 -68881.16        NA        NA        NA -72305.29
## 24 -228596.7 -181747.5 -68940.76        NA        NA        NA -72256.01
## 25 -228643.2 -179795.9 -67155.88        NA        NA        NA -72563.92
##          VEE       EVE       VVE       EEV       VEV       EVV       VVV
## 1  -83257.53 -83257.53 -83257.53 -83257.53 -83257.53 -83257.53 -83257.53
## 2  -75561.42 -76379.61 -74508.06 -77604.96 -74709.31 -75130.47 -72873.15
## 3  -74657.52 -75186.39        NA -76361.50 -73174.35 -73423.82 -72008.07
## 4  -73652.34        NA        NA -76106.80 -72360.21 -72416.13        NA
## 5  -72616.76        NA        NA -74560.38 -71595.75 -71983.75        NA
## 6  -72451.05        NA        NA -74633.86 -71212.58        NA        NA
## 7  -71825.18        NA        NA -73895.71 -70079.01        NA        NA
## 8  -71488.00        NA        NA -73693.32        NA        NA        NA
## 9         NA        NA        NA -64806.23        NA        NA        NA
## 10        NA        NA        NA -63614.70        NA        NA        NA
## 11        NA        NA        NA -24944.11        NA        NA        NA
## 12        NA        NA        NA -40163.93        NA        NA        NA
## 13        NA        NA        NA -24824.20        NA        NA        NA
## 14        NA        NA        NA -28376.08        NA        NA        NA
## 15        NA        NA        NA -11022.71        NA        NA        NA
## 16        NA        NA        NA -26650.42        NA        NA        NA
## 17        NA        NA        NA -38713.17        NA        NA        NA
## 18        NA        NA        NA -23899.45        NA        NA        NA
## 19        NA        NA        NA -24244.33        NA        NA        NA
## 20        NA        NA        NA -11047.47        NA        NA        NA
## 21        NA        NA        NA -36965.18        NA        NA        NA
## 22        NA        NA        NA -35203.03        NA        NA        NA
## 23        NA        NA        NA -20775.03        NA        NA        NA
## 24        NA        NA        NA -13967.82        NA        NA        NA
## 25        NA        NA        NA -36115.53        NA        NA        NA
##
## Top 3 models based on the BIC criterion:
##    EEV,15    EEV,20    EEV,24
## -11022.71 -11047.47 -13967.82
```

```
fit=Mclust(data_cluster,verbose=FALSE,G=15,modelNames='EEV',
           initialization=list(hcRandomPairs(data_cluster)))
summary(fit)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EEV (ellipsoidal, equal volume and shape) model with 15 components:
##
```

```
##  log-likelihood    n  df       BIC       ICL
##       -18774.12 6287 335 -40478.24 -43358.07
##
## Clustering table:
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 2228  108   70   46 2561  165   16  463   96   93  181   17  105  107   31
```

```
cl=fit$classification
cl_train=cl[data$ind_train]
cl_test=cl[data$ind_train==FALSE]

train_c_cluster=as.data.frame(cbind(train,train_distance,cl_train))
test_c_cluster=as.data.frame(cbind(test,test_distance,cl_test))
colnames(train_c_cluster)[39]='cluster'
colnames(test_c_cluster)[39]='cluster'
train_c_cluster$cluster=as.factor(train_c_cluster$cluster)
test_c_cluster$cluster=as.factor(test_c_cluster$cluster)


## LISA

data_lisa=as.data.frame(cbind(data$buy_price,data_coord))
colnames(data_lisa)[1]='buy_price'

x_lisa=data$buy_price
knear_lisa=knearneigh(data_coord,k=300,longlat=TRUE)

r_lisa=sapply(1:300,function(i){
  cor(x_lisa,x_lisa[knear_lisa$nn[,i]])
})

data.frame(k=1:300,r=r_lisa) %>%
  ggplot(aes(x=k,y=r_lisa)) +
  geom_line() +
  geom_smooth(se=FALSE) +
  xlab('k-esimo nearest neighbour') +
  ylab('Correlazione') +
  theme_light()
```




```
neigh_lisa=knn2nb(knearneigh(data_coord,100,longlat=TRUE))
k_distance_lisa=nbdists(neigh_lisa,data_coord,longlat=TRUE)
inv_k_distance_lisa=lapply(k_distance_lisa, function(x) (1/(x+1)))
k_weight_lisa=nb2listw(neigh_lisa,glist=inv_k_distance_lisa,style='W')

lisa=localmoran(data_lisa$buy_price,k_weight_lisa)
lisa=attributes(lisa)$quadr[,1]
summary(lisa)
```

```
##   Low-Low  High-Low  Low-High High-High
##      3218       296       936      1837
```

```
lisa_data=as.data.frame(cbind(data,data_distance,lisa))
lisa_train=lisa[data$ind_train]
lisa_test=lisa[data$ind_train==FALSE]

train_c_lisa=as.data.frame(cbind(train,train_distance,lisa_train))
test_c_lisa=as.data.frame(cbind(test,test_distance,lisa_test))
colnames(train_c_lisa)[39]='lisa'
colnames(test_c_lisa)[39]='lisa'
train_c_lisa$lisa=factor(train_c_lisa$lisa)
train_c_lisa$lisa=relevel(train_c_lisa$lisa,ref='High-High')
test_c_lisa$lisa=as.factor(test_c_lisa$lisa)
```

```r
## Verifica delle Migliori Variabili Spaziali

cl=makePSOCKcluster(11)  # calcolo parallelo con 11 cores
registerDoParallel(cl)

control=trainControl(method='cv',number=10,allowParallel=TRUE)
tune_rf=expand.grid(mtry=10,splitrule='variance',min.node.size=10)

set.seed(123)
fit_lm_district=train(buy_price~.,data=train_c_district,method='lmStepAIC',direction='both',
                      k=log(nrow(train_c_district)),trControl=control,trace=FALSE)
summary(fit_lm_district$finalModel)
```

```
##
## Call:
## lm(formula = .outcome ~ sq_mt_built + n_bathrooms + energy_certificateLow +
##     floor + house_type_idAttic + house_type_idIndependent + is_renewal_neededTRUE +
##     is_new_developmentTRUE + has_parkingTRUE + has_poolTRUE +
##     d_supermarket + d_hospital + d_pharmacy + d_bank + d_university +
##     d_school + d_kindergarten + d_train + d_park + d_disco +
##     d_cinema + d_library + d_attraction + district14 + district15 +
##     district18 + district19 + district20 + district21 + district3 +
##     district4 + district5 + district6 + district9, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1452615   -87085    -7018    68576  4443529
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                4.543e+03  1.607e+04   0.283 0.777368
## sq_mt_built                3.447e+03  6.797e+01  50.720  < 2e-16 ***
## n_bathrooms                7.755e+04  5.299e+03  14.634  < 2e-16 ***
## energy_certificateLow     -3.737e+04  8.299e+03  -4.502 6.87e-06 ***
## floor                      5.153e+03  1.483e+03   3.476 0.000514 ***
## house_type_idAttic         1.206e+05  1.251e+04   9.639  < 2e-16 ***
## house_type_idIndependent -2.391e+05  1.830e+04 -13.060  < 2e-16 ***
## is_renewal_neededTRUE     -4.416e+04  8.997e+03  -4.908 9.48e-07 ***
## is_new_developmentTRUE     1.082e+05  9.500e+03  11.385  < 2e-16 ***
## has_parkingTRUE            2.430e+04  7.554e+03   3.217 0.001305 **
## has_poolTRUE               4.967e+04  8.958e+03   5.545 3.09e-08 ***
## d_supermarket              6.702e+01  1.361e+01   4.924 8.73e-07 ***
## d_hospital                -2.102e+01  5.620e+00  -3.741 0.000185 ***
## d_pharmacy                -1.212e+02  2.861e+01  -4.236 2.31e-05 ***
## d_bank                    -7.448e+01  1.164e+01  -6.398 1.72e-10 ***
## d_university              -3.131e+01  5.384e+00  -5.816 6.41e-09 ***
## d_school                  -8.720e+01  1.995e+01  -4.370 1.27e-05 ***
## d_kindergarten             1.658e+02  1.825e+01   9.085  < 2e-16 ***
## d_train                   -2.416e+01  3.274e+00  -7.380 1.84e-13 ***
## d_park                     1.867e+02  3.094e+01   6.036 1.70e-09 ***
## d_disco                   -2.214e+01  4.498e+00  -4.921 8.87e-07 ***
## d_cinema                  -1.508e+01  4.628e+00  -3.258 0.001129 **
## d_library                  5.620e+01  7.171e+00   7.836 5.63e-15 ***
```

```
## d_attraction             -2.983e+01  4.543e+00  -6.565 5.72e-11 ***
## district14                5.847e+04  1.557e+04   3.755 0.000175 ***
## district15                2.033e+05  1.901e+04  10.694  < 2e-16 ***
## district18               -8.149e+04  1.913e+04  -4.260 2.09e-05 ***
## district19                1.090e+05  2.685e+04   4.062 4.94e-05 ***
## district20                9.893e+04  2.457e+04   4.026 5.76e-05 ***
## district21                1.090e+05  2.400e+04   4.541 5.72e-06 ***
## district3                -9.317e+04  1.210e+04  -7.700 1.63e-14 ***
## district4                 5.428e+04  1.570e+04   3.458 0.000548 ***
## district5                 1.297e+05  1.315e+04   9.862  < 2e-16 ***
## district6                 1.038e+05  1.407e+04   7.377 1.88e-13 ***
## district9                 1.272e+05  1.609e+04   7.908 3.21e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 206100 on 4998 degrees of freedom
## Multiple R-squared:  0.7852, Adjusted R-squared:  0.7837
## F-statistic: 537.3 on 34 and 4998 DF,  p-value: < 2.2e-16
```

```
fit_lm_district$results[which.min(fit_lm_district$results$RMSE),]
```

```
##   parameter     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      none 206897.7 0.7863771 118508.8 38455.04 0.03388563 7730.648
```

```
fit_rf_district=train(buy_price~.,data=train_c_district,method='ranger',
                      importance='impurity',num.trees=150,tuneGrid=tune_rf,trControl=control)
fit_rf_district$results[which.min(fit_rf_district$results$RMSE),]
```

```
##   mtry splitrule min.node.size     RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 1   10  variance            10 171236.6 0.8613662 76976.59 56037.58 0.06083446
##      MAESD
## 1 6577.168
```

```
set.seed(123)
fit_lm_cluster=train(buy_price~.,data=train_c_cluster,method='lmStepAIC',direction='both',
                     k=log(nrow(train_c_cluster)),trControl=control,trace=FALSE)
summary(fit_lm_cluster$finalModel)
```

```
##
## Call:
## lm(formula = .outcome ~ sq_mt_built + n_bathrooms + n_rooms +
##     energy_certificateLow + floor + house_type_idAttic + house_type_idIndependent +
##     is_renewal_neededTRUE + is_new_developmentTRUE + has_parkingTRUE +
##     has_poolTRUE + d_supermarket + d_hospital + d_bank + d_university +
##     d_school + d_kindergarten + d_train + d_bus + d_park + d_stadium +
##     d_cinema + d_library + d_attraction + cluster4 + cluster5 +
##     cluster9 + cluster11 + cluster13 + cluster14, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1521772   -82025    -8526    67599  4493465
##
```

```
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.561e+04  1.515e+04   1.691 0.090907 .
## sq_mt_built                 3.442e+03  7.468e+01  46.092  < 2e-16 ***
## n_bathrooms                 1.017e+05  6.097e+03  16.687  < 2e-16 ***
## n_rooms                     1.405e+04  3.382e+03   4.155 3.30e-05 ***
## energy_certificateLow      -2.731e+04  8.245e+03  -3.312 0.000933 ***
## floor                       7.184e+03  1.480e+03   4.854 1.25e-06 ***
## house_type_idAttic          1.115e+05  1.252e+04   8.908  < 2e-16 ***
## house_type_idIndependent -2.331e+05  1.933e+04 -12.059  < 2e-16 ***
## is_renewal_neededTRUE      -4.477e+04  9.121e+03  -4.909 9.47e-07 ***
## is_new_developmentTRUE      1.440e+05  9.487e+03  15.181  < 2e-16 ***
## has_parkingTRUE             3.257e+04  7.537e+03   4.321 1.58e-05 ***
## has_poolTRUE                4.932e+04  8.887e+03   5.550 3.00e-08 ***
## d_supermarket               4.636e+01  1.244e+01   3.727 0.000196 ***
## d_hospital                 -4.294e+01  4.904e+00  -8.756  < 2e-16 ***
## d_bank                     -8.578e+01  1.115e+01  -7.692 1.73e-14 ***
## d_university               -3.188e+01  4.041e+00  -7.888 3.74e-15 ***
## d_school                   -6.698e+01  1.711e+01  -3.914 9.20e-05 ***
## d_kindergarten              1.815e+02  1.741e+01  10.425  < 2e-16 ***
## d_train                    -2.752e+01  2.976e+00  -9.248  < 2e-16 ***
## d_bus                      -8.913e+00  2.591e+00  -3.440 0.000586 ***
## d_park                      2.549e+02  2.936e+01   8.685  < 2e-16 ***
## d_stadium                  -2.408e+01  2.349e+00 -10.254  < 2e-16 ***
## d_cinema                   -2.663e+01  4.135e+00  -6.442 1.29e-10 ***
## d_library                   8.053e+01  6.685e+00  12.046  < 2e-16 ***
## d_attraction               -1.846e+01  2.852e+00  -6.475 1.04e-10 ***
## cluster4                   -3.440e+05  4.194e+04  -8.201 2.99e-16 ***
## cluster5                   -1.031e+05  7.285e+03 -14.147  < 2e-16 ***
## cluster9                   -2.407e+05  3.189e+04  -7.548 5.23e-14 ***
## cluster11                  -6.531e+04  2.238e+04  -2.918 0.003543 **
## cluster13                  -2.831e+05  2.509e+04 -11.284  < 2e-16 ***
## cluster14                  -2.423e+05  2.435e+04  -9.951  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 205200 on 5002 degrees of freedom
## Multiple R-squared:  0.7869, Adjusted R-squared:  0.7856
## F-statistic: 615.8 on 30 and 5002 DF,  p-value: < 2.2e-16
```

```
fit_lm_cluster$results[which.min(fit_lm_cluster$results$RMSE),]
```

```
##   parameter     RMSE Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      none 207321.7 0.786619 118234.4 38814.93 0.03386087 6779.463
```

```
fit_rf_cluster=train(buy_price~.,data=train_c_cluster,method='ranger',
                     importance='impurity',num.trees=150,tuneGrid=tune_rf,trControl=control)
fit_rf_cluster$results[which.min(fit_rf_cluster$results$RMSE),]
```

```
##   mtry splitrule min.node.size     RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 1   10  variance            10 174633.4 0.8542185 78709.42 53502.09 0.05870588
##      MAESD
## 1 6177.151
```

```
set.seed(123)
fit_lm_lisa=train(buy_price~.,data=train_c_lisa,method='lmStepAIC',direction='both',
                  k=log(nrow(train_c_lisa)),trControl=control,trace=FALSE)
summary(fit_lm_lisa$finalModel)
```

```
##
## Call:
## lm(formula = .outcome ~ sq_mt_built + n_bathrooms + energy_certificateLow +
##     floor + house_type_idAttic + house_type_idIndependent + is_renewal_neededTRUE +
##     is_new_developmentTRUE + has_poolTRUE + has_individual_heatingTRUE +
##     d_supermarket + d_pharmacy + d_bank + d_university + d_school +
##     d_kindergarten + d_train + d_park + d_stadium + d_disco +
##     d_cinema + d_library + d_attraction + `lisaLow-Low` + `lisaHigh-Low` +
##     `lisaLow-High`, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1353832   -81024    -5317    66483  4643807
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.432e+05  1.802e+04   7.949 2.31e-15 ***
## sq_mt_built                 3.308e+03  7.011e+01  47.181  < 2e-16 ***
## n_bathrooms                 6.751e+04  5.390e+03  12.526  < 2e-16 ***
## energy_certificateLow      -3.473e+04  8.295e+03  -4.187 2.87e-05 ***
## floor                       6.123e+03  1.501e+03   4.081 4.56e-05 ***
## house_type_idAttic          1.176e+05  1.261e+04   9.332  < 2e-16 ***
## house_type_idIndependent   -2.340e+05  1.878e+04 -12.458  < 2e-16 ***
## is_renewal_neededTRUE      -3.962e+04  9.070e+03  -4.369 1.27e-05 ***
## is_new_developmentTRUE      1.024e+05  9.522e+03  10.751  < 2e-16 ***
## has_poolTRUE                3.648e+04  8.520e+03   4.282 1.89e-05 ***
## has_individual_heatingTRUE  3.142e+04  8.170e+03   3.846 0.000121 ***
## d_supermarket               4.995e+01  1.295e+01   3.857 0.000116 ***
## d_pharmacy                 -9.124e+01  2.815e+01  -3.241 0.001200 **
## d_bank                     -6.388e+01  1.168e+01  -5.467 4.80e-08 ***
## d_university               -2.190e+01  3.937e+00  -5.564 2.78e-08 ***
## d_school                   -8.154e+01  1.756e+01  -4.645 3.49e-06 ***
## d_kindergarten              1.597e+02  1.796e+01   8.887  < 2e-16 ***
## d_train                    -1.382e+01  2.976e+00  -4.643 3.52e-06 ***
## d_park                      2.279e+02  2.996e+01   7.607 3.33e-14 ***
## d_stadium                  -1.714e+01  2.218e+00  -7.727 1.32e-14 ***
## d_disco                    -1.298e+01  3.482e+00  -3.728 0.000195 ***
## d_cinema                   -1.912e+01  4.109e+00  -4.653 3.36e-06 ***
## d_library                   3.709e+01  7.612e+00   4.872 1.14e-06 ***
## d_attraction               -1.360e+01  3.370e+00  -4.036 5.53e-05 ***
## `lisaLow-Low`              -1.797e+05  1.059e+04 -16.960  < 2e-16 ***
## `lisaHigh-Low`             -1.673e+05  1.503e+04 -11.128  < 2e-16 ***
## `lisaLow-High`             -1.281e+05  1.117e+04 -11.466  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 207000 on 5006 degrees of freedom
## Multiple R-squared:  0.7831, Adjusted R-squared:  0.782
```

```
## F-statistic: 695.1 on 26 and 5006 DF,  p-value: < 2.2e-16
```

```
fit_lm_lisa$results[which.min(fit_lm_lisa$results$RMSE),]
```

```
##   parameter     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      none 207425.3 0.7851833 115640.2 41250.98 0.03614779 8064.372
```

```
fit_rf_lisa=train(buy_price~.,data=train_c_lisa,method='ranger',
                  importance='impurity',num.trees=150,tuneGrid=tune_rf,trControl=control)
fit_rf_lisa$results[which.min(fit_rf_lisa$results$RMSE),]
```

```
##   mtry splitrule min.node.size     RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 1   10  variance            10 165348.7 0.8675168 71593.76 53492.58 0.05859579
##      MAESD
## 1 6999.482
```

```
#write_xlsx(lisa_data,'data_reg.xlsx')
#write_xlsx(train_c_lisa,'train_reg.xlsx')
#write_xlsx(test_c_lisa,'test_reg.xlsx')
```

Regressione

```
library(readxl)
library(car)
library(Metrics)
library(nortest)
library(doParallel)
library(ggpubr)
library(sf)
library(spdep)
library(caret)
library(spatialreg)
```

```
train=as.data.frame(read_excel('train.xlsx'))
test=as.data.frame(read_excel('test.xlsx'))
train_reg=as.data.frame(read_excel('train_reg.xlsx'))
test_reg=as.data.frame(read_excel('test_reg.xlsx'))
train_coord=as.data.frame(read_excel('train_coord.xlsx'))
test_coord=as.data.frame(read_excel('test_coord.xlsx'))

x_train=train_reg[,-1]
y_train=train_reg[,1]
x_test=test_reg[,-1]
y_test=test_reg[,1]
```

```
set.seed(123)
control=trainControl(method='cv',number=10,allowParallel=TRUE)
cl=makePSOCKcluster(11)
registerDoParallel(cl)


# Modello Lineare

fit_lm=train(buy_price~.,data=train,method='lmStepAIC',direction='both',
             k=log(nrow(train)),trControl=control,trace=FALSE)
fit_lm$results[which.min(fit_lm$results$RMSE),]
```

```
##   parameter     RMSE Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      none 243064.3 0.703151 144217.7 37284.54 0.03728231 9711.555
```

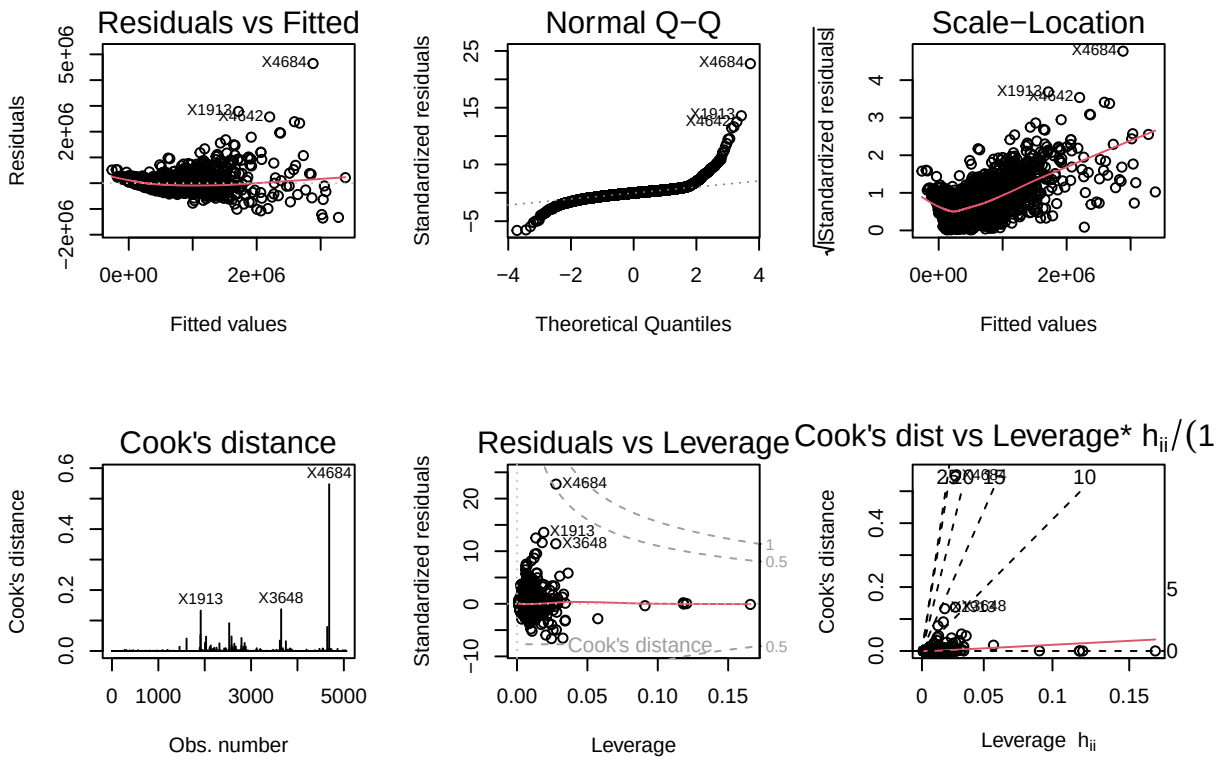
```
summary(fit_lm$finalModel)
```

```
##
## Call:
## lm(formula = .outcome ~ sq_mt_built + n_bathrooms + n_rooms +
##     house_type_idAttic + house_type_idIndependent + is_exteriorTRUE +
```

```
##     is_new_developmentTRUE + has_acTRUE + has_liftTRUE + has_gardenTRUE +
##     has_individual_heatingTRUE, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1628677  -107980    -5402    78575  4695266
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -80902.4    17585.5  -4.601 4.32e-06 ***
## sq_mt_built                    3850.8       82.7  46.563  < 2e-16 ***
## n_bathrooms                  102405.5     6396.9  16.009  < 2e-16 ***
## n_rooms                      -17202.8     3851.8  -4.466 8.14e-06 ***
## house_type_idAttic           137105.5    14306.4   9.584  < 2e-16 ***
## house_type_idIndependent    -367574.4    20714.4 -17.745  < 2e-16 ***
## is_exteriorTRUE              -64266.5    12310.3  -5.221 1.86e-07 ***
## is_new_developmentTRUE       110493.8    10233.0  10.798  < 2e-16 ***
## has_acTRUE                    30691.5     7552.2   4.064 4.90e-05 ***
## has_liftTRUE                  51886.3     9400.3   5.520 3.57e-08 ***
## has_gardenTRUE               -50734.1     8114.9  -6.252 4.39e-10 ***
## has_individual_heatingTRUE   -45219.0     9242.6  -4.892 1.03e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 242400 on 5021 degrees of freedom
## Multiple R-squared:  0.7015, Adjusted R-squared:  0.7008
## F-statistic:  1073 on 11 and 5021 DF,  p-value: < 2.2e-16
```

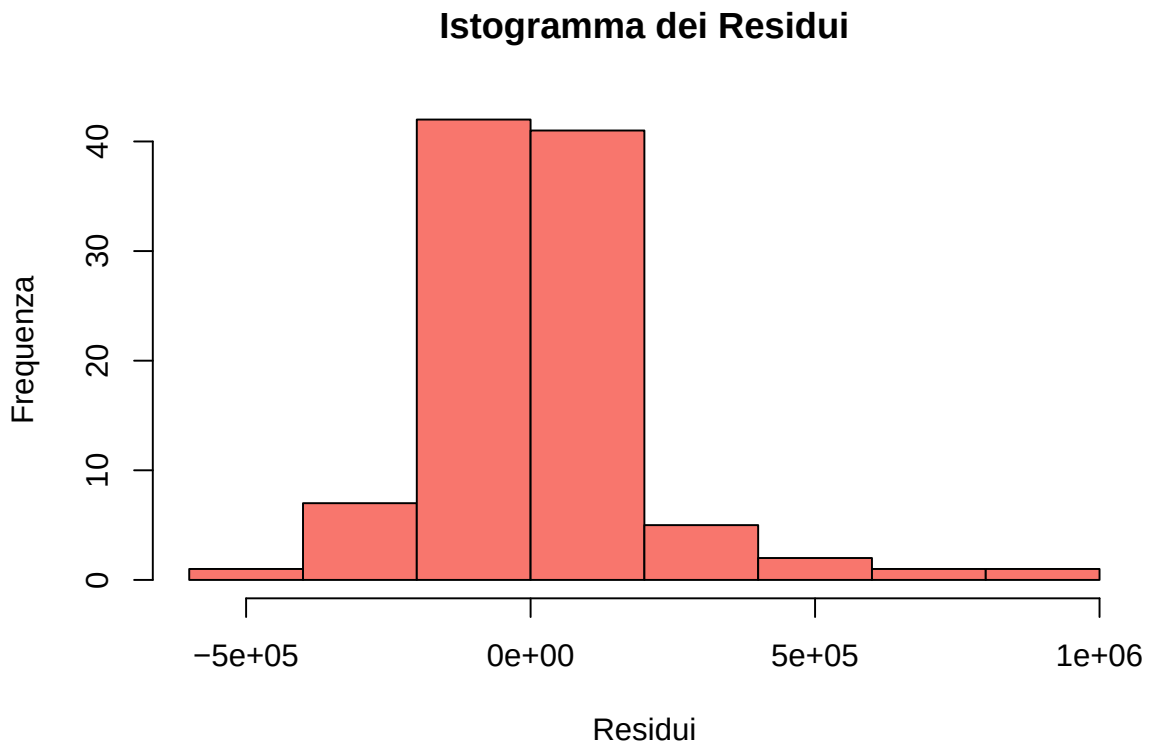
```
#confint(fit_lm$finalModel)


set.seed(123)
fit_lm_complete=train(buy_price~.,data=train_reg,method='lmStepAIC',direction='both',
                      k=log(nrow(train_reg)),trControl=control,trace=FALSE)
#fit_lm_complete$results[which.min(fit_lm_complete$results$RMSE),]
#summary(fit_lm_complete$finalModel)
#confint(fit_lm_complete$finalModel)
```

```
par(mfrow=c(2,3))
plot(fit_lm_complete$finalModel,which=1:6)  # diagnostiche
```




```
par(mfrow=c(1,1))
```

```
set.seed(123)
sub_res=sample(fit_lm_complete$finalModel$residuals,100)
hist(sub_res,main='Istogramma dei Residui',xlab='Residui',ylab='Frequenza',col='#F8766D')
```



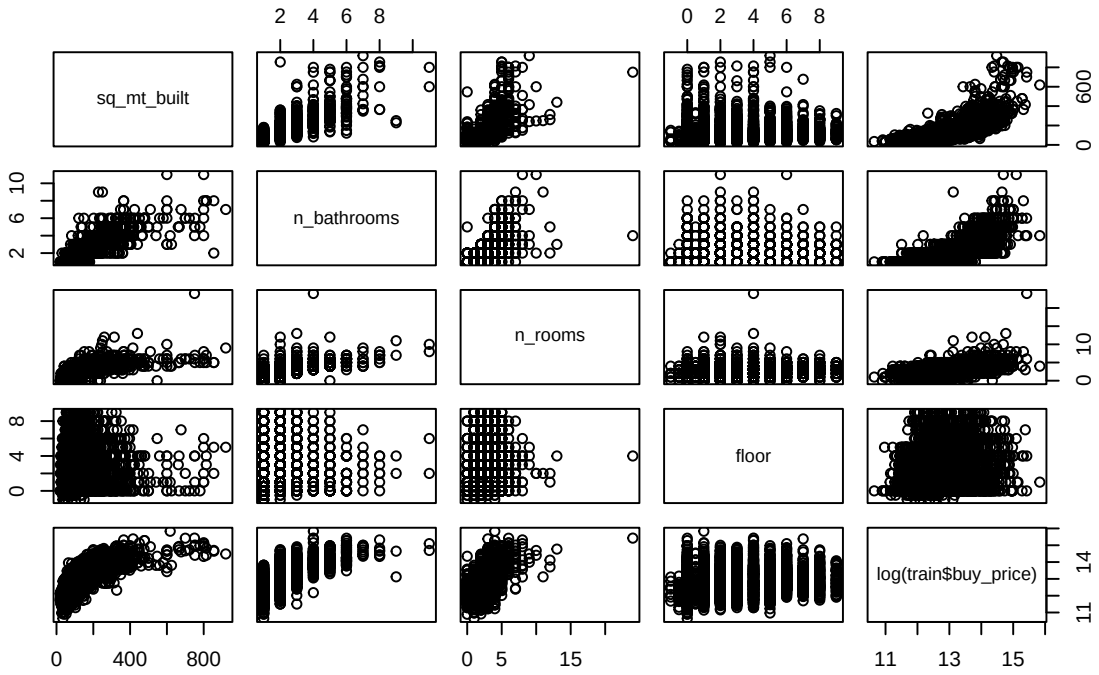

```
shapiro.test(sub_res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sub_res
## W = 0.84076, p-value = 5.482e-09
```

```
lillie.test(fit_lm_complete$finalModel$residuals)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  fit_lm_complete$finalModel$residuals
## D = 0.15641, p-value < 2.2e-16
```

```
data_num_log=train[,sapply(train,is.numeric)]
data_num_log=as.data.frame(cbind(data_num_log[,-1],log(train$buy_price)))
pairs(~.,data=data_num_log,main="Diagrammi Multipli")  # scatterplots con log(y)
```




```
## Modello Log-Lineare

set.seed(123)
fit_loglm=train(log(buy_price)~.+I(sq_mt_built^2)+I(n_rooms^2)+I(n_bathrooms^2),
                data=train,method='lmStepAIC',direction='both',k=log(nrow(train)),
                trControl=control,trace=FALSE)
fit_loglm$results[which.min(fit_loglm$results$RMSE),]
```

```
##   parameter      RMSE  Rsquared       MAE     RMSESD RsquaredSD       MAESD
## 1      none 0.3604108 0.7441636 0.2889728 0.01181533 0.01861354 0.009412958
```

```
summary(fit_loglm$finalModel)
```

```
##
## Call:
## lm(formula = .outcome ~ sq_mt_built + n_bathrooms + n_rooms +
##     floor + house_type_idAttic + house_type_idIndependent + is_exteriorTRUE +
##     is_renewal_neededTRUE + is_new_developmentTRUE + has_acTRUE +
##     has_fitted_wardrobesTRUE + has_liftTRUE + has_gardenTRUE +
##     has_individual_heatingTRUE + `I(sq_mt_built^2)` + `I(n_rooms^2)` +
##     `I(n_bathrooms^2)`, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.79431 -0.24686 -0.00811  0.24539  1.45678
```

```
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.157e+01  2.986e-02 387.634  < 2e-16 ***
## sq_mt_built                   9.189e-03  2.550e-04  36.041  < 2e-16 ***
## n_bathrooms                   1.786e-01  1.952e-02   9.153  < 2e-16 ***
## n_rooms                      -5.679e-02  8.962e-03  -6.336 2.56e-10 ***
## floor                         1.133e-02  2.625e-03   4.314 1.63e-05 ***
## house_type_idAttic            1.459e-01  2.199e-02   6.634 3.62e-11 ***
## house_type_idIndependent     -4.501e-01  3.092e-02 -14.559  < 2e-16 ***
## is_exteriorTRUE              -1.500e-01  1.834e-02  -8.181 3.55e-16 ***
## is_renewal_neededTRUE        -4.914e-02  1.646e-02  -2.985 0.002853 **
## is_new_developmentTRUE        2.100e-01  1.695e-02  12.388  < 2e-16 ***
## has_acTRUE                    8.583e-02  1.201e-02   7.144 1.03e-12 ***
## has_fitted_wardrobesTRUE      4.433e-02  1.275e-02   3.476 0.000513 ***
## has_liftTRUE                  3.477e-01  1.448e-02  24.005  < 2e-16 ***
## has_gardenTRUE               -5.983e-02  1.217e-02  -4.916 9.09e-07 ***
## has_individual_heatingTRUE -1.648e-01  1.399e-02 -11.780  < 2e-16 ***
## `I(sq_mt_built^2)`           -7.359e-06  3.305e-07 -22.269  < 2e-16 ***
## `I(n_rooms^2)`                3.518e-03  7.763e-04   4.531 6.00e-06 ***
## `I(n_bathrooms^2)`           -8.163e-03  2.574e-03  -3.172 0.001525 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.358 on 5015 degrees of freedom
## Multiple R-squared:  0.7478, Adjusted R-squared:  0.747
## F-statistic: 874.7 on 17 and 5015 DF,  p-value: < 2.2e-16
```

```
set.seed(123)
fit_loglm_complete=train(log(buy_price)~.+I(sq_mt_built^2)+I(n_rooms^2)+I(n_bathrooms^2),
                         data=train_reg,method='lmStepAIC',direction='both',k=log(nrow(train_reg)),
                         trControl=control,trace=FALSE)
fit_loglm_complete$results[which.min(fit_loglm_complete$results$RMSE),]
```

```
##   parameter      RMSE  Rsquared       MAE     RMSESD RsquaredSD       MAESD
## 1      none 0.2276164 0.8978986 0.1745979 0.01763031  0.0134047 0.005423899
```

```
summary(fit_loglm_complete$finalModel)
```

```
##
## Call:
## lm(formula = .outcome ~ sq_mt_built + n_bathrooms + n_rooms +
##     floor + house_type_idAttic + house_type_idIndependent + is_renewal_neededTRUE +
##     is_new_developmentTRUE + has_acTRUE + has_liftTRUE + has_balconyTRUE +
##     has_parkingTRUE + has_poolTRUE + has_storage_roomTRUE + d_hospital +
##     d_bank + d_university + d_kindergarten + d_train + d_airport +
##     d_park + d_stadium + d_disco + d_cinema + d_library + d_historic +
##     d_attraction + `lisaHigh-Low` + `lisaLow-High` + `lisaLow-Low` +
##     `I(sq_mt_built^2)` + `I(n_bathrooms^2)`, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.02986 -0.14021  0.00264  0.14243  1.02426
```

```
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.215e+01  2.738e-02 443.756  < 2e-16 ***
## sq_mt_built                 5.173e-03  1.662e-04  31.122  < 2e-16 ***
## n_bathrooms                 1.018e-01  1.227e-02   8.301  < 2e-16 ***
## n_rooms                     2.939e-02  3.866e-03   7.601 3.50e-14 ***
## floor                       1.469e-02  1.629e-03   9.019  < 2e-16 ***
## house_type_idAttic          1.137e-01  1.380e-02   8.239  < 2e-16 ***
## house_type_idIndependent -1.870e-01  1.982e-02  -9.436  < 2e-16 ***
## is_renewal_neededTRUE    -8.969e-02  1.013e-02  -8.855  < 2e-16 ***
## is_new_developmentTRUE    1.951e-01  1.032e-02  18.898  < 2e-16 ***
## has_acTRUE                5.925e-02  7.266e-03   8.153 4.43e-16 ***
## has_liftTRUE              2.292e-01  9.064e-03  25.286  < 2e-16 ***
## has_balconyTRUE           2.089e-02  6.719e-03   3.110 0.001884 **
## has_parkingTRUE           5.687e-02  8.684e-03   6.549 6.36e-11 ***
## has_poolTRUE              9.056e-02  9.759e-03   9.280  < 2e-16 ***
## has_storage_roomTRUE      2.628e-02  7.626e-03   3.447 0.000572 ***
## d_hospital               -4.841e-05  5.919e-06  -8.179 3.60e-16 ***
## d_bank                   -9.585e-05  1.199e-05  -7.991 1.65e-15 ***
## d_university             -4.989e-05  4.341e-06 -11.493  < 2e-16 ***
## d_kindergarten            1.060e-04  1.770e-05   5.987 2.29e-09 ***
## d_train                  -3.125e-05  4.158e-06  -7.515 6.70e-14 ***
## d_airport                 1.184e-05  1.817e-06   6.516 7.93e-11 ***
## d_park                    2.252e-04  3.225e-05   6.985 3.22e-12 ***
## d_stadium                -2.872e-05  2.540e-06 -11.306  < 2e-16 ***
## d_disco                  -1.623e-05  3.787e-06  -4.285 1.86e-05 ***
## d_cinema                 -2.348e-05  4.467e-06  -5.257 1.52e-07 ***
## d_library                 8.161e-05  8.134e-06  10.033  < 2e-16 ***
## d_historic                1.017e-05  2.857e-06   3.560 0.000374 ***
## d_attraction             -4.963e-05  4.629e-06 -10.723  < 2e-16 ***
## `lisaHigh-Low`           -1.392e-01  1.657e-02  -8.404  < 2e-16 ***
## `lisaLow-High`           -2.845e-01  1.262e-02 -22.536  < 2e-16 ***
## `lisaLow-Low`            -4.974e-01  1.288e-02 -38.622  < 2e-16 ***
## `I(sq_mt_built^2)`       -3.663e-06  2.074e-07 -17.664  < 2e-16 ***
## `I(n_bathrooms^2)`       -6.520e-03  1.617e-03  -4.033 5.58e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2216 on 5000 degrees of freedom
## Multiple R-squared:  0.9037, Adjusted R-squared:  0.9031
## F-statistic:  1466 on 32 and 5000 DF,  p-value: < 2.2e-16
```

```
cv_log_error=function(data,k,seed=123){  # funzione per cross-validation con log(y)
  set.seed(seed)
  yourdata=data[sample(nrow(data)),]
  fold=cut(seq(1,nrow(yourdata)),breaks=k,labels=FALSE)
  rmse_val=NULL
  mae_val=NULL
  r2_val=NULL
  for(i in 1:k){
    test_index=which(fold==i,arr.ind=TRUE)
    test_data=yourdata[test_index,]
    train_data=yourdata[-test_index,]
```

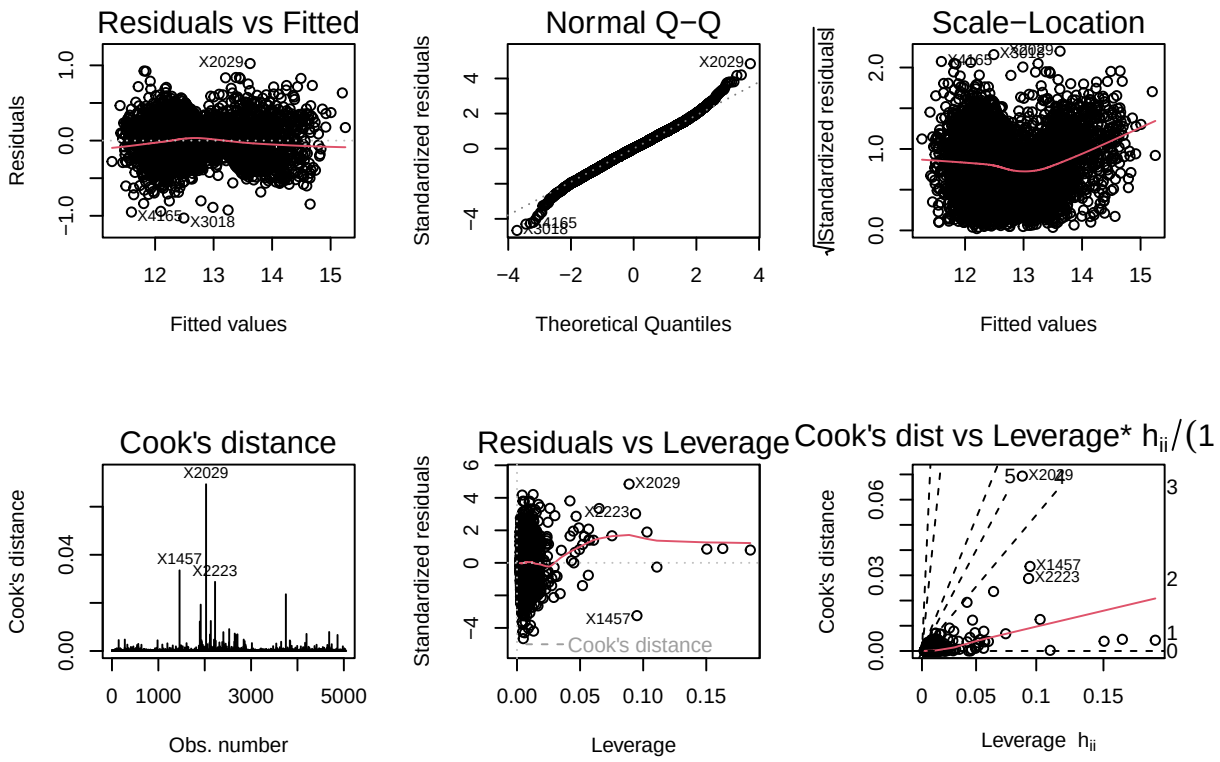
```
    null_m=lm(log(buy_price)~1,data=train_data)
    all_m=lm(log(buy_price)~.+I(sq_mt_built^2)+I(n_rooms^2)+I(n_bathrooms^2),data=train_data)
    model=step(null_m,direction='both',scope=formula(all_m),k=log(nrow(train_data)),trace=FALSE)
    pred=predict(model,newdata=test_data[,-1])
    pred_exp=exp(pred)
    rmse_val[i]=rmse(test_data[,1],pred_exp)
    r2_val[i]=cor(test_data[,1],pred_exp)^2
    mae_val[i]=mae(test_data[,1],pred_exp)
  }
  rmse_mean=mean(rmse_val)
  r2_mean=mean(r2_val)
  mae_mean=mean(mae_val)
  cv_error=as.data.frame(cbind(rmse_mean,r2_mean,mae_mean))
  colnames(cv_error)=c('RMSE','Rsquared','MAE')
  return(cv_error)
}

cv_log_error(train,10)
```

```
##       RMSE  Rsquared      MAE
## 1 248836.9 0.6917868 135127.1
```

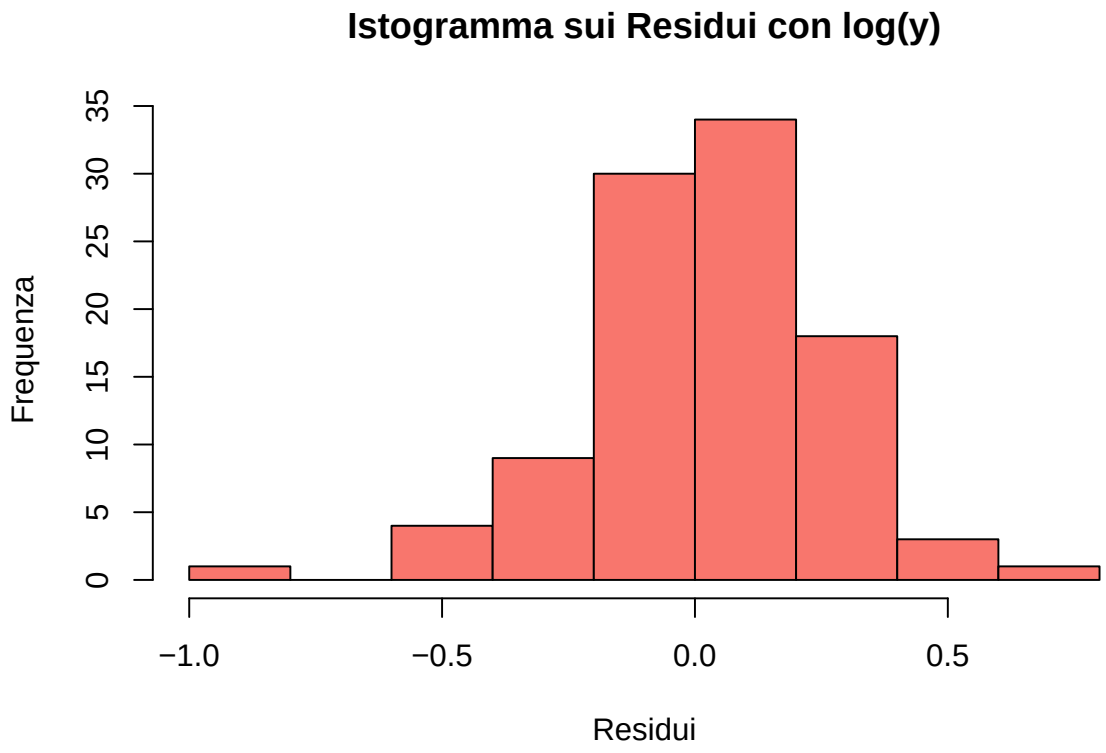
```
cv_log_error(train_reg,10)
```

```
##       RMSE  Rsquared      MAE
## 1 189766.5 0.8143128 87295.54
```

```
par(mfrow=c(2,3))
plot(fit_loglm_complete$finalModel,which=1:6)
```




```
par(mfrow=c(1,1))

set.seed(123)
sub_res_log=sample(fit_loglm_complete$finalModel$residuals,100)
hist(sub_res_log,main='Istogramma sui Residui con log(y)',xlab='Residui',ylab='Frequenza',col='#F8766D'
```




```
shapiro.test(sub_res_log)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sub_res_log
## W = 0.97318, p-value = 0.03891
```

```
lillie.test(fit_loglm_complete$finalModel$residuals)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  fit_loglm_complete$finalModel$residuals
## D = 0.015261, p-value = 0.009488
```

```
vif(fit_lm$finalModel)
```

```
##                sq_mt_built               n_bathrooms
##                   4.085910                  3.541150
##                   n_rooms         house_type_idAttic
##                   2.109430                  1.046565
##   house_type_idIndependent            is_exteriorTRUE
##                   1.655409                  1.073466
```

```
##     is_new_developmentTRUE                 has_acTRUE
##                   1.359739                   1.219848
##                has_liftTRUE              has_gardenTRUE
##                   1.281344                   1.193599
## has_individual_heatingTRUE
##                   1.206934
```

```
vif(fit_loglm$finalModel)
```

```
##                sq_mt_built                n_bathrooms
##                  17.805482                  15.114543
##                    n_rooms                      floor
##                   5.236344                   1.159193
##         house_type_idAttic   house_type_idIndependent
##                   1.133570                   1.690916
##            is_exteriorTRUE      is_renewal_neededTRUE
##                   1.092404                   1.211158
##     is_new_developmentTRUE                 has_acTRUE
##                   1.711310                   1.415265
##   has_fitted_wardrobesTRUE               has_liftTRUE
##                   1.526414                   1.394575
##             has_gardenTRUE has_individual_heatingTRUE
##                   1.230654                   1.267923
##         `I(sq_mt_built^2)`             `I(n_rooms^2)`
##                  10.495963                   3.293684
##         `I(n_bathrooms^2)`
##                  10.773136
```

```
vif(fit_lm_complete$finalModel)
```

```
##                sq_mt_built                n_bathrooms
##                   4.029489                   3.449794
##      energy_certificateLow                      floor
##                   1.362870                   1.133360
##         house_type_idAttic   house_type_idIndependent
##                   1.115098                   1.867547
##      is_renewal_neededTRUE     is_new_developmentTRUE
##                   1.099973                   1.615697
##               has_poolTRUE has_individual_heatingTRUE
##                   1.797343                   1.294026
##              d_supermarket                 d_pharmacy
##                   1.742188                   2.459565
##                    d_bank               d_university
##                   3.289510                   2.582222
##                   d_school             d_kindergarten
##                   2.224209                   1.537405
##                    d_train                    d_park
##                   1.148420                   1.192924
##                  d_stadium                   d_disco
##                   1.625153                   3.757847
##                   d_cinema                  d_library
##                   2.217061                   4.256526
##               d_attraction             `lisaHigh-Low`
```

```
##                   2.908031                   1.210703
##             `lisaLow-High`              `lisaLow-Low`
##                   1.871342                   3.296066
```

```
vif(fit_loglm_complete$finalModel)
```

```
##              sq_mt_built             n_bathrooms                   n_rooms
##                19.761768               15.596109                 2.544391
##                    floor      house_type_idAttic house_type_idIndependent
##                 1.164620                1.165132                 1.814742
##    is_renewal_neededTRUE  is_new_developmentTRUE               has_acTRUE
##                 1.196736                1.656428                 1.351864
##             has_liftTRUE         has_balconyTRUE          has_parkingTRUE
##                 1.426179                1.147791                 1.835050
##             has_poolTRUE    has_storage_roomTRUE               d_hospital
##                 2.057447                1.415630                 2.599854
##                   d_bank             d_university           d_kindergarten
##                 3.024621                2.738599                 1.302233
##                  d_train               d_airport                   d_park
##                 1.955122                1.741197                 1.205903
##                d_stadium                  d_disco                 d_cinema
##                 1.859295                3.878708                 2.285380
##                d_library               d_historic             d_attraction
##                 4.240352                3.609735                 4.785577
##           `lisaHigh-Low`           `lisaLow-High`            `lisaLow-Low`
##                 1.282622                2.085574                 4.249629
##       `I(sq_mt_built^2)`       `I(n_bathrooms^2)`
##                10.789318               11.096210
```

```
c(AIC(fit_lm$finalModel),BIC(fit_lm$finalModel))
```

```
## [1] 139100.1 139184.9
```

```
c(AIC(fit_lm_complete$finalModel),BIC(fit_lm_complete$finalModel))
```

```
## [1] 137522.4 137705.0
```

```
c(AIC(fit_loglm$finalModel)+2*sum(log(train_reg$buy_price)),
  BIC(fit_loglm$finalModel)+2*sum(log(train_reg$buy_price)))
```

```
## [1] 132384.3 132508.2
```

```
c(AIC(fit_loglm_complete$finalModel)+2*sum(log(train_reg$buy_price)),
  BIC(fit_loglm_complete$finalModel)+2*sum(log(train_reg$buy_price)))
```

```
## [1] 127569.0 127790.8
```

```
## Errore di Previsione

### modello lineare
obs_lm=predict(fit_lm_complete,newdata=x_train)  # train error
c(rmse(y_train,obs_lm),mae(y_train,obs_lm))
```

```
## [1] 206398.7 113972.3
```

```
pred_lm=predict(fit_lm_complete,newdata=x_test)  # test error
c(rmse(y_test,pred_lm),mae(y_test,pred_lm))
```

```
## [1] 183844.4 110095.6
```

```
### modello log-lineare
obs_loglm=predict(fit_loglm_complete,newdata=x_train)
obs_loglm=exp(obs_loglm)
c(rmse(y_train,obs_loglm),mae(y_train,obs_loglm))
```

```
## [1] 178699.36  85774.89
```

```
pred_loglm=predict(fit_loglm_complete,newdata=x_test)
pred_loglm=exp(pred_loglm)
c(rmse(y_test,pred_loglm),mae(y_test,pred_loglm))
```

```
## [1] 166736.23  85520.44
```

```
# Modelli Lineari Generalizzati (GLM)

set.seed(123)
fit_glm=train(buy_price~.+I(sq_mt_built^2)+I(n_rooms^2)+I(n_bathrooms^2),  ## Normale -> log
              data=train_reg,method='glmStepAIC',family=gaussian(link='log'),
              direction='both',k=log(nrow(train_reg)),trControl=control,trace=FALSE)
summary(fit_glm)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1340057    -51590      -811     53685   2705265
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.254e+01  4.199e-02 298.636  < 2e-16 ***
## sq_mt_built                  4.944e-03  1.212e-04  40.809  < 2e-16 ***
## n_bathrooms                  1.031e-01  1.059e-02   9.732  < 2e-16 ***
## n_rooms                      2.303e-02  1.895e-03  12.148  < 2e-16 ***
## energy_certificateLow       -7.987e-02  9.446e-03  -8.456  < 2e-16 ***
## floor                        6.523e-03  1.794e-03   3.636 0.000279 ***
## house_type_idAttic           1.621e-01  1.235e-02  13.130  < 2e-16 ***
## house_type_idIndependent    -2.343e-01  1.577e-02 -14.860  < 2e-16 ***
## is_exteriorTRUE              7.792e-02  2.150e-02   3.625 0.000292 ***
## is_renewal_neededTRUE       -1.085e-01  1.260e-02  -8.613  < 2e-16 ***
## is_new_developmentTRUE       1.777e-01  1.281e-02  13.878  < 2e-16 ***
## has_acTRUE                   4.447e-02  9.400e-03   4.731 2.29e-06 ***
## has_fitted_wardrobesTRUE    -6.550e-02  9.850e-03  -6.649 3.26e-11 ***
```

```
## has_liftTRUE                1.783e-01  2.138e-02   8.338  < 2e-16 ***
## has_parkingTRUE             7.057e-02  9.781e-03   7.215 6.19e-13 ***
## has_poolTRUE                6.125e-02  1.059e-02   5.785 7.69e-09 ***
## has_individual_heatingTRUE  5.151e-02  9.970e-03   5.166 2.48e-07 ***
## d_supermarket               8.205e-05  2.173e-05   3.777 0.000161 ***
## d_hospital                 -4.775e-05  8.929e-06  -5.348 9.28e-08 ***
## d_bank                     -1.106e-04  1.937e-05  -5.708 1.21e-08 ***
## d_university               -5.915e-05  7.469e-06  -7.919 2.93e-15 ***
## d_school                   -8.802e-05  2.261e-05  -3.893 0.000100 ***
## d_kindergarten              2.945e-04  1.796e-05  16.399  < 2e-16 ***
## d_train                    -6.005e-05  6.057e-06  -9.915  < 2e-16 ***
## d_airport                  -9.792e-06  2.315e-06  -4.231 2.37e-05 ***
## d_gym                      -4.118e-05  1.227e-05  -3.355 0.000799 ***
## d_park                      4.225e-04  3.125e-05  13.522  < 2e-16 ***
## d_stadium                  -4.227e-05  3.803e-06 -11.114  < 2e-16 ***
## d_disco                    -4.388e-05  4.846e-06  -9.055  < 2e-16 ***
## d_cinema                   -6.600e-05  5.810e-06 -11.360  < 2e-16 ***
## d_library                   1.103e-04  1.040e-05  10.604  < 2e-16 ***
## d_historic                  3.366e-05  4.221e-06   7.974 1.89e-15 ***
## d_attraction               -6.619e-05  6.499e-06 -10.185  < 2e-16 ***
## `lisaHigh-Low`             -1.862e-01  1.895e-02  -9.825  < 2e-16 ***
## `lisaLow-High`             -3.240e-01  2.031e-02 -15.953  < 2e-16 ***
## `lisaLow-Low`              -4.949e-01  1.972e-02 -25.093  < 2e-16 ***
## `I(sq_mt_built^2)`         -2.835e-06  1.190e-07 -23.817  < 2e-16 ***
## `I(n_bathrooms^2)`         -6.469e-03  9.447e-04  -6.848 8.39e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 24064538248)
##
##     Null deviance: 9.8851e+14  on 5032  degrees of freedom
## Residual deviance: 1.2020e+14  on 4995  degrees of freedom
## AIC: 134632
##
## Number of Fisher Scoring iterations: 7
```

```
fit_glm
```

```
## Generalized Linear Model with Stepwise Feature Selection
##
## 5033 samples
##   38 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4530, 4529, 4530, 4530, 4529, 4531, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   165623.1  0.8589126  90192.79
```

```
set.seed(123)
fit_glm_gamma=train(buy_price~.+I(sq_mt_built^2)+I(n_rooms^2)+I(n_bathrooms^2),  ## Gamma -> inverse
```

```
                    data=train_reg,method='glmStepAIC',family=Gamma(link='log'),
                    direction='both',k=log(nrow(train_reg)),trControl=control,trace=FALSE)
fit_glm_gamma
```

```
## Generalized Linear Model with Stepwise Feature Selection
##
## 5033 samples
##   38 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4530, 4529, 4530, 4530, 4529, 4531, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   188328.3  0.8211155  88588.41
```

```
summary(fit_glm_gamma)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.88750  -0.15972  -0.01767   0.12265   1.05003
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.218e+01  2.779e-02 438.195  < 2e-16 ***
## sq_mt_built                 5.404e-03  1.679e-04  32.183  < 2e-16 ***
## n_bathrooms                 9.249e-02  1.246e-02   7.420 1.37e-13 ***
## n_rooms                     2.659e-02  3.917e-03   6.789 1.26e-11 ***
## floor                       1.463e-02  1.653e-03   8.849  < 2e-16 ***
## house_type_idAttic          1.208e-01  1.384e-02   8.733  < 2e-16 ***
## house_type_idIndependent -1.557e-01  2.015e-02  -7.729 1.31e-14 ***
## is_renewal_neededTRUE    -8.565e-02  1.029e-02  -8.320  < 2e-16 ***
## is_new_developmentTRUE    1.905e-01  1.048e-02  18.168  < 2e-16 ***
## has_acTRUE                5.792e-02  7.383e-03   7.846 5.22e-15 ***
## has_liftTRUE              2.245e-01  9.206e-03  24.382  < 2e-16 ***
## has_parkingTRUE           5.682e-02  8.825e-03   6.438 1.32e-10 ***
## has_poolTRUE              8.746e-02  9.914e-03   8.822  < 2e-16 ***
## has_storage_roomTRUE      2.646e-02  7.751e-03   3.414 0.000646 ***
## d_hospital               -4.890e-05  6.016e-06  -8.128 5.45e-16 ***
## d_bank                   -9.782e-05  1.218e-05  -8.030 1.21e-15 ***
## d_university             -4.940e-05  4.411e-06 -11.199  < 2e-16 ***
## d_kindergarten            1.086e-04  1.799e-05   6.039 1.66e-09 ***
## d_train                  -3.314e-05  4.226e-06  -7.841 5.42e-15 ***
## d_airport                 1.272e-05  1.846e-06   6.887 6.38e-12 ***
## d_park                    2.328e-04  3.277e-05   7.102 1.40e-12 ***
## d_stadium                -2.923e-05  2.581e-06 -11.325  < 2e-16 ***
## d_disco                  -1.505e-05  3.849e-06  -3.909 9.38e-05 ***
```

```
## d_cinema                 -2.333e-05  4.540e-06  -5.138 2.88e-07 ***
## d_library                 8.208e-05  8.267e-06   9.929  < 2e-16 ***
## d_historic                1.103e-05  2.903e-06   3.798 0.000147 ***
## d_attraction             -5.197e-05  4.705e-06 -11.047  < 2e-16 ***
## `lisaHigh-Low`           -1.404e-01  1.683e-02  -8.342  < 2e-16 ***
## `lisaLow-High`           -2.890e-01  1.283e-02 -22.524  < 2e-16 ***
## `lisaLow-Low`            -4.897e-01  1.309e-02 -37.411  < 2e-16 ***
## `I(sq_mt_built^2)`       -3.912e-06  2.097e-07 -18.658  < 2e-16 ***
## `I(n_bathrooms^2)`       -4.872e-03  1.643e-03  -2.965 0.003039 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.05071781)
##
##     Null deviance: 2831.99  on 5032  degrees of freedom
## Residual deviance:  245.93  on 5001  degrees of freedom
## AIC: 127617
##
## Number of Fisher Scoring iterations: 6
```

```
set.seed(123)
fit_glm_poisson=train(buy_price~.+I(sq_mt_built^2)+I(n_rooms^2)+I(n_bathrooms^2),  ## Poisson -> log
                      data=train_reg,method='glmStepAIC',family=poisson,
                      direction='both',k=log(nrow(train_reg)),trControl=control,trace=FALSE)
summary(fit_glm_poisson)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -847.71   -92.53   -11.41    73.11  1571.89
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 1.234e+01  2.158e-04 57190.7   <2e-16 ***
## sq_mt_built                 5.027e-03  8.543e-07  5884.4   <2e-16 ***
## n_bathrooms                 9.837e-02  6.928e-05  1420.0   <2e-16 ***
## n_rooms                     1.689e-02  3.371e-05   501.0   <2e-16 ***
## energy_certificateLow      -4.460e-02  5.656e-05  -788.6   <2e-16 ***
## floor                       1.135e-02  1.052e-05  1079.5   <2e-16 ***
## house_type_idAttic          1.338e-01  8.012e-05  1669.7   <2e-16 ***
## house_type_idIndependent   -2.323e-01  1.064e-04 -2183.0   <2e-16 ***
## is_exteriorTRUE             3.979e-02  9.275e-05   429.0   <2e-16 ***
## is_renewal_neededTRUE      -9.553e-02  7.100e-05 -1345.4   <2e-16 ***
## is_new_developmentTRUE      1.929e-01  7.796e-05  2473.7   <2e-16 ***
## has_acTRUE                  4.598e-02  5.221e-05   880.6   <2e-16 ***
## has_fitted_wardrobesTRUE   -1.933e-02  5.700e-05  -339.1   <2e-16 ***
## has_liftTRUE                2.148e-01  8.196e-05  2620.4   <2e-16 ***
## has_balconyTRUE             6.445e-03  4.703e-05   137.0   <2e-16 ***
## has_gardenTRUE              2.035e-02  5.922e-05   343.7   <2e-16 ***
## has_parkingTRUE             4.981e-02  5.676e-05   877.6   <2e-16 ***
## has_poolTRUE                6.205e-02  6.613e-05   938.3   <2e-16 ***
```

```
## has_storage_roomTRUE        1.494e-02  4.913e-05   304.0   <2e-16 ***
## has_individual_heatingTRUE  1.871e-02  5.677e-05   329.5   <2e-16 ***
## d_supermarket               6.960e-05  1.093e-07   636.5   <2e-16 ***
## d_hospital                 -4.430e-05  4.598e-08  -963.5   <2e-16 ***
## d_pharmacy                 -3.699e-05  1.890e-07  -195.7   <2e-16 ***
## d_post                     -2.770e-05  6.865e-08  -403.4   <2e-16 ***
## d_bank                     -7.473e-05  1.003e-07  -744.9   <2e-16 ***
## d_university               -5.073e-05  3.561e-08 -1424.5   <2e-16 ***
## d_school                   -3.960e-05  1.363e-07  -290.6   <2e-16 ***
## d_kindergarten              2.112e-04  1.152e-07  1832.4   <2e-16 ***
## d_train                    -4.343e-05  3.212e-08 -1352.2   <2e-16 ***
## d_bus                      -5.464e-06  2.106e-08  -259.5   <2e-16 ***
## d_airport                   3.713e-06  1.249e-08   297.3   <2e-16 ***
## d_gym                      -2.003e-05  5.813e-08  -344.5   <2e-16 ***
## d_park                      3.311e-04  1.963e-07  1686.7   <2e-16 ***
## d_stadium                  -3.213e-05  2.195e-08 -1463.9   <2e-16 ***
## d_disco                    -2.644e-05  2.656e-08  -995.4   <2e-16 ***
## d_cinema                   -3.210e-05  3.087e-08 -1039.8   <2e-16 ***
## d_library                   9.222e-05  5.621e-08  1640.6   <2e-16 ***
## d_historic                  1.822e-05  2.186e-08   833.3   <2e-16 ***
## d_attraction               -5.272e-05  3.492e-08 -1509.7   <2e-16 ***
## `lisaHigh-Low`             -1.584e-01  9.720e-05 -1629.6   <2e-16 ***
## `lisaLow-High`             -3.096e-01  8.754e-05 -3536.4   <2e-16 ***
## `lisaLow-Low`              -5.007e-01  8.945e-05 -5596.9   <2e-16 ***
## `I(sq_mt_built^2)`         -3.134e-06  9.166e-10 -3419.7   <2e-16 ***
## `I(n_rooms^2)`              3.763e-04  1.631e-06   230.8   <2e-16 ***
## `I(n_bathrooms^2)`         -4.967e-03  7.003e-06  -709.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1427524899  on 5032  degrees of freedom
## Residual deviance:  126596723  on 4988  degrees of freedom
## AIC: 126670273
##
## Number of Fisher Scoring iterations: 4
```

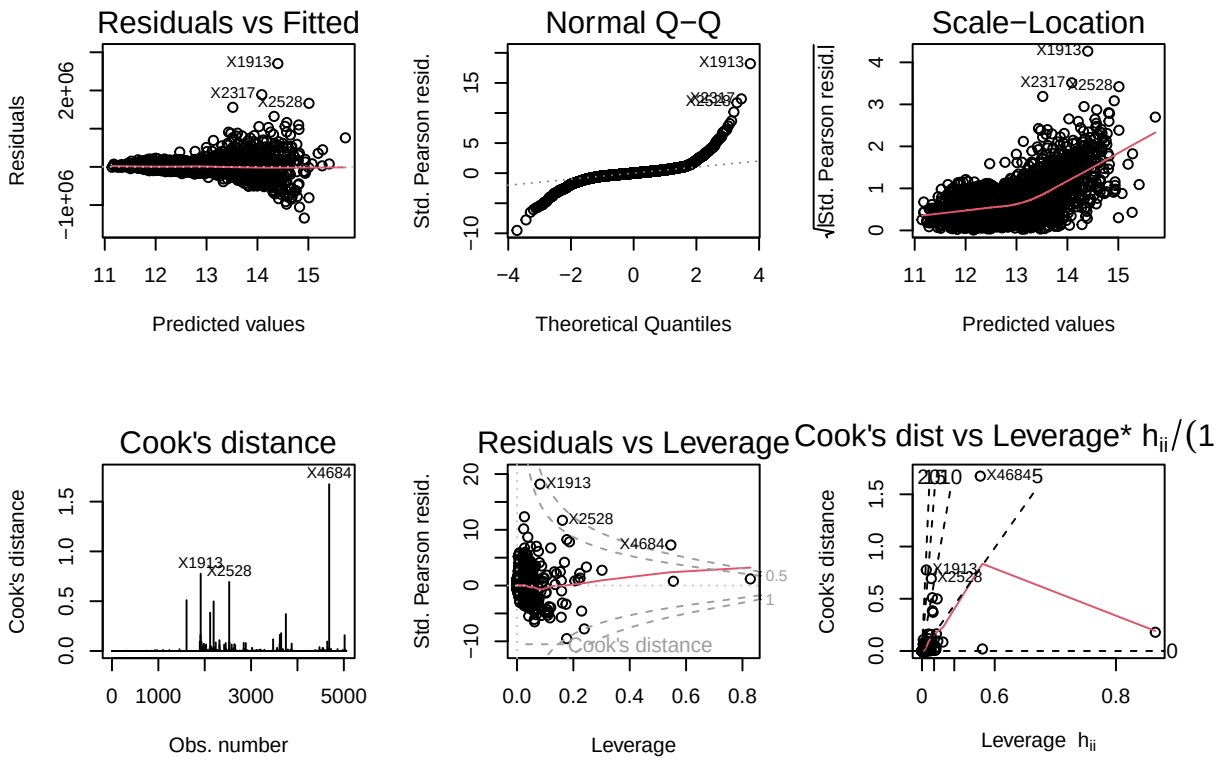
```
fit_glm_poisson
```

```
## Generalized Linear Model with Stepwise Feature Selection
##
## 5033 samples
##   38 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4530, 4529, 4530, 4530, 4529, 4531, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   172423.5  0.8487033  85709
```

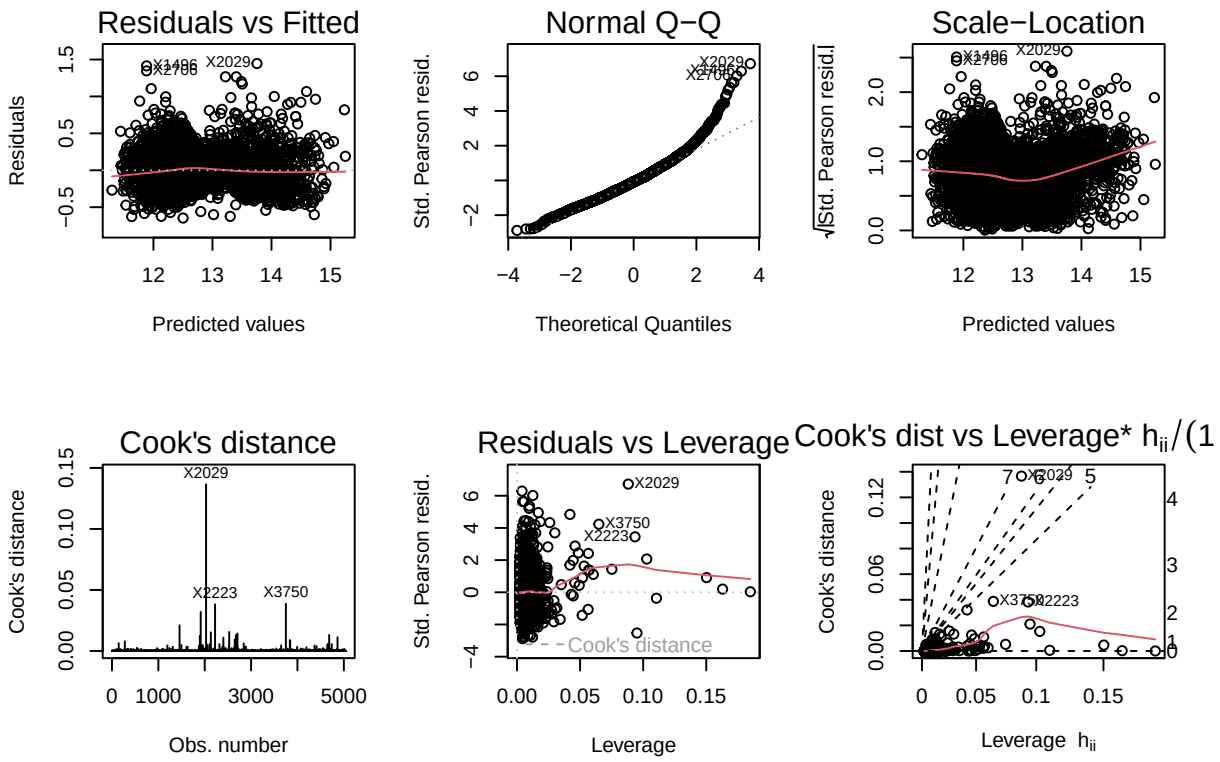
```
cv_log_gamma_error=function(data,k,seed=123){
  set.seed(seed)
  yourdata=data[sample(nrow(data)),]
  fold=cut(seq(1,nrow(yourdata)),breaks=k,labels=FALSE)
  rmse_val=NULL
  mae_val=NULL
  r2_val=NULL
  cc=trainControl('none',allowParallel=TRUE)
  for(i in 1:k){
    test_index=which(fold==i,arr.ind=TRUE)
    test_data=yourdata[test_index,]
    train_data=yourdata[-test_index,]
    model=train(log(buy_price)~.+I(sq_mt_built^2)+I(n_rooms^2)+I(n_bathrooms^2),
                data=train_data,method='glmStepAIC',family=Gamma,direction='both',
                k=log(nrow(train_data)),trControl=cc,trace=FALSE)
    pred=predict(model,newdata=test_data[,-1])
    pred=exp(pred)
    rmse_val[i]=rmse(test_data[,1],pred)
    mae_val[i]=mae(test_data[,1],pred)
    r2_val[i]=cor(test_data[,1],pred)^2
  }
  rmse_mean=mean(rmse_val)
  r2_mean=mean(r2_val)
  mae_mean=mean(mae_val)
  cv_error=as.data.frame(cbind(rmse_mean,r2_mean,mae_mean))
  colnames(cv_error)=c('RMSE','Rsquared','MAE')
  return(cv_error)
}
cv_log_gamma_error(train_reg,10)
```

```
##       RMSE  Rsquared      MAE
## 1 188030.7 0.8136976 88222.95
```

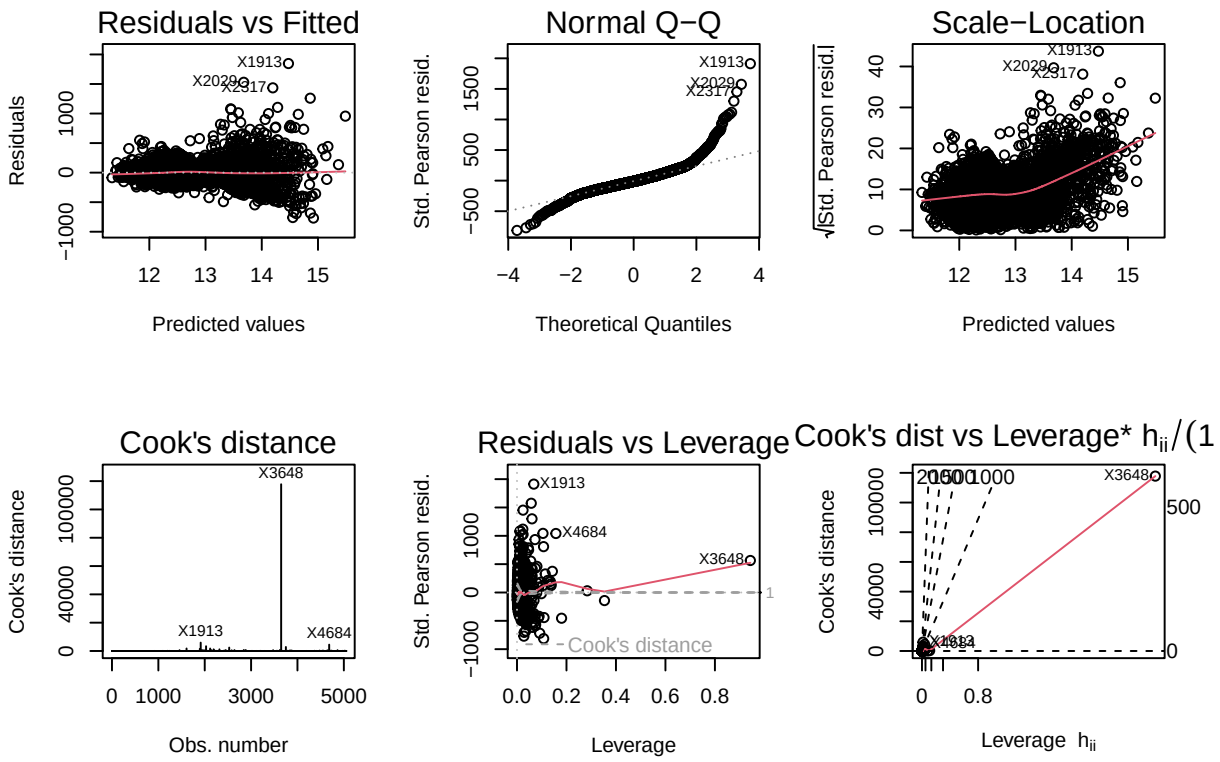
```
par(mfrow=c(2,3))
plot(fit_glm$finalModel,which=1:6)
```




```
plot(fit_glm_gamma$finalModel,which=1:6)
```




```
plot(fit_glm_poisson$finalModel,which=1:6)
```




```
par(mfrow=c(1,1))
```

```
obs_glm=predict(fit_glm,newdata=x_train)
c(rmse(obs_glm,y_train),mae(obs_glm,y_train))
```

```
## [1] 154540  86978
```

```
pred_glm=predict(fit_glm,newdata=x_test)
c(rmse(pred_glm,y_test),mae(pred_glm,y_test))
```

```
## [1] 155223.29  84706.39
```

```
obs_glm_gamma=predict(fit_glm_gamma,newdata=x_train)
obs_glm_gamma=exp(obs_glm_gamma)
c(rmse(obs_glm_gamma,y_train),mae(obs_glm_gamma,y_train))
```

```
## [1] Inf Inf
```

```
pred_glm_gamma=predict(fit_glm,newdata=x_test)
pred_glm=exp(pred_glm_gamma)
c(rmse(pred_glm_gamma,y_test),mae(pred_glm_gamma,y_test))
```

```
## [1] 155223.29  84706.39
```

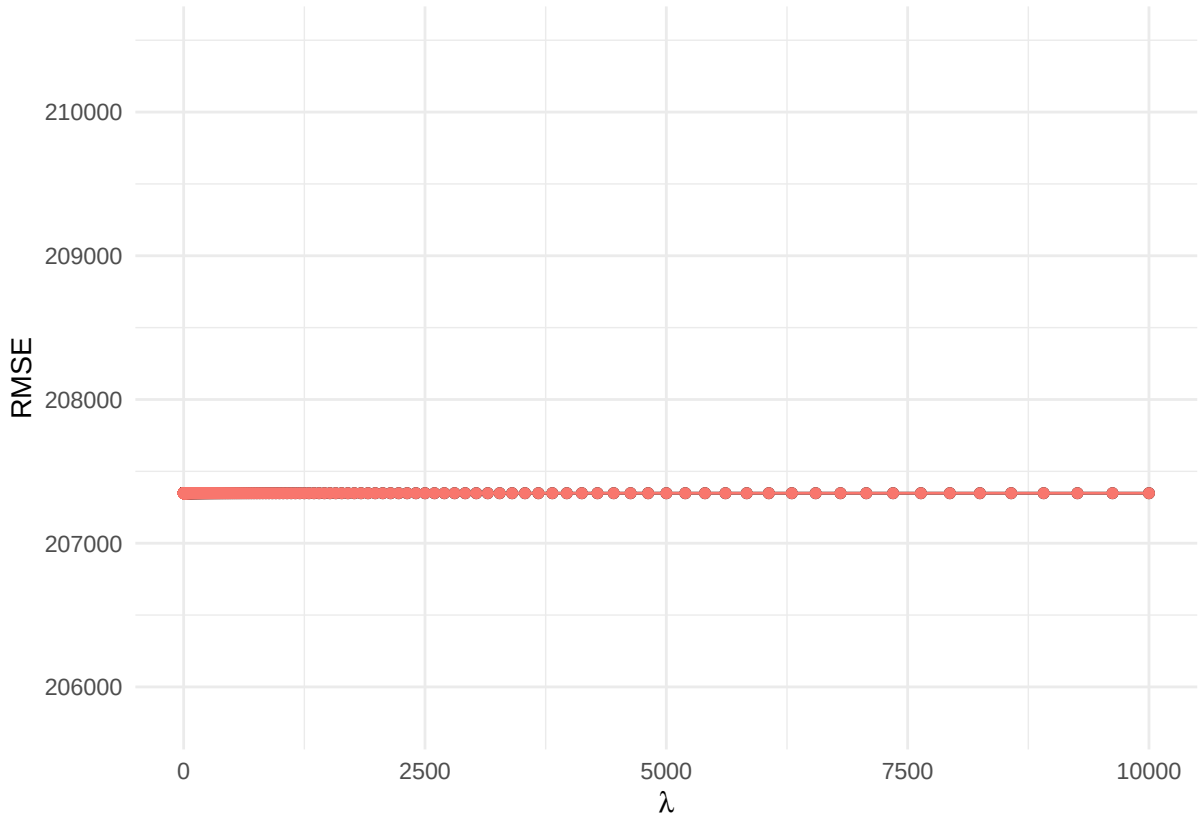
```
obs_glm_poisson=predict(fit_glm_poisson,newdata=x_train)
c(rmse(obs_glm_poisson,y_train),mae(obs_glm_poisson,y_train))
```

```
## [1] 160640.19  83376.35
```

```
pred_glm_poisson=predict(fit_glm_poisson,newdata=x_test)
c(rmse(pred_glm_poisson,y_test),mae(pred_glm_poisson,y_test))
```

```
## [1] 154233.23  81849.38
```

```
# Regolarizzazione (Ridge, Lasso, ElasticNet)

## Ridge

set.seed(123)
tune_ridge=expand.grid(alpha=0,lambda=c(0,10^seq(log10(10000),log10(0.1),length.out=300)))
#tune_ridge=expand.grid(alpha=0,lambda=c(0,10^seq(log10(50000),log10(0.1),length.out=300)))
fit_ridge=train(buy_price~.,data=train_reg,method='glmnet',tuneGrid=tune_ridge,trControl=control)
plot_ridge=ggplot(fit_ridge) + geom_point(color='#F8766D') + geom_line(color='#F8766D') +
  ylim(205800,210500) + theme_minimal() + labs(x=expression(lambda),y='RMSE')
plot_ridge
```



```
#fit_ridge$bestTune
fit_ridge$results[which.min(fit_ridge$results$RMSE),]
```

```
##   alpha lambda     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1     0      0 207348.1 0.7851804 112575.2 45347.47 0.03981968 7943.062
```

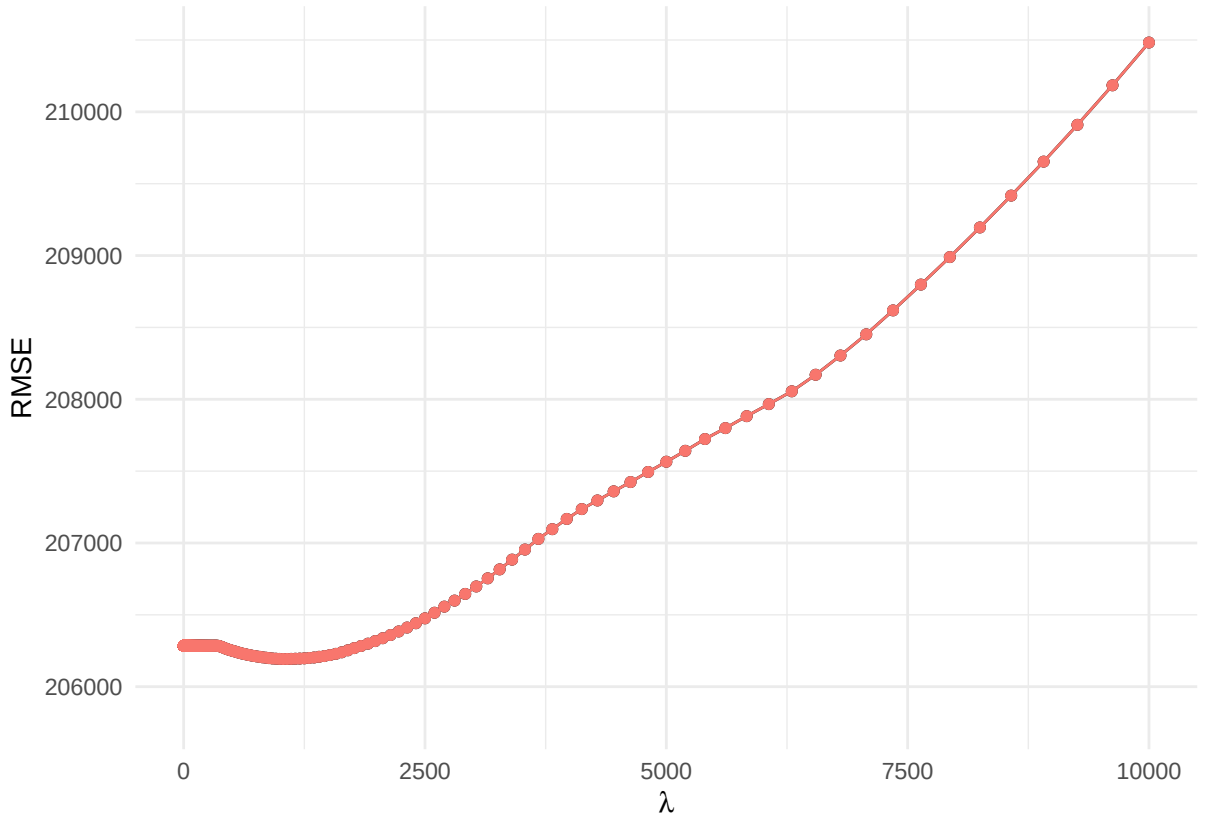
```
obs_ridge=predict(fit_ridge,newdata=x_train)
c(rmse(y_train,obs_ridge),mae(y_train,obs_ridge))
```

```
## [1] 208140.9 111354.2
```

```
pred_ridge=predict(fit_ridge,newdata=x_test)
c(rmse(y_test,pred_ridge),mae(y_test,pred_ridge))
```

```
## [1] 183072.7 108046.0
```

```
## Lasso

set.seed(123)
tune_lasso=expand.grid(alpha=1,lambda=c(0,10^seq(log10(10000),log10(0.1),length.out=300)))
fit_lasso=train(buy_price~.,data=train_reg,method='glmnet',tuneGrid=tune_lasso,trControl=control)
plot_lasso=ggplot(fit_lasso) + geom_point(color='#F8766D') + geom_line(color='#F8766D') +
  ylim(205800,210500) + theme_minimal() + labs(x=expression(lambda),y='RMSE')
plot_lasso
```



```
fit_lasso$results[which.min(fit_lasso$results$RMSE),]
```

```
##     alpha   lambda     RMSE  Rsquared      MAE  RMSESD RsquaredSD    MAESD
## 242     1 1031.283 206192.6 0.7877747 113463.6 41804.5  0.0363941 8080.349
```

```
coef(fit_lasso$finalModel,fit_lasso$bestTune$lambda)
```

```
## 42 x 1 sparse Matrix of class "dgCMatrix"
##                                       s1
## (Intercept)                 1.205040e+05
## sq_mt_built                 3.282558e+03
## n_bathrooms                 6.647855e+04
## n_rooms                     .
## energy_certificateLow      -3.306635e+04
## floor                       5.544308e+03
## house_type_idAttic          1.164952e+05
## house_type_idIndependent   -2.245795e+05
## is_exteriorTRUE             .
## is_renewal_neededTRUE      -3.422818e+04
## is_new_developmentTRUE      9.886891e+04
## has_acTRUE                  1.072991e+04
## has_fitted_wardrobesTRUE   -1.322061e+04
## has_liftTRUE                .
## has_balconyTRUE            -5.387472e+03
## has_gardenTRUE              .
## has_parkingTRUE             1.567042e+04
## has_poolTRUE                2.931537e+04
## has_storage_roomTRUE        7.475514e+02
## has_individual_heatingTRUE  2.370838e+04
## d_supermarket               4.192907e+01
## d_hospital                 -1.468600e+01
## d_pharmacy                 -6.375466e+01
## d_post                     -1.318164e+01
## d_bank                     -4.619588e+01
## d_university               -2.158565e+01
## d_school                   -4.916368e+01
## d_kindergarten              1.540051e+02
## d_train                    -7.477438e+00
## d_bus                      -8.168432e+00
## d_airport                   3.602992e+00
## d_gym                       .
## d_park                      2.275924e+02
## d_stadium                  -1.118545e+01
## d_disco                    -6.487132e+00
## d_cinema                   -1.786537e+01
## d_library                   3.094211e+01
## d_historic                 -1.428683e+00
## d_attraction               -1.248754e+01
## lisaHigh-Low               -1.569210e+05
## lisaLow-High               -1.209663e+05
## lisaLow-Low                -1.696735e+05
```

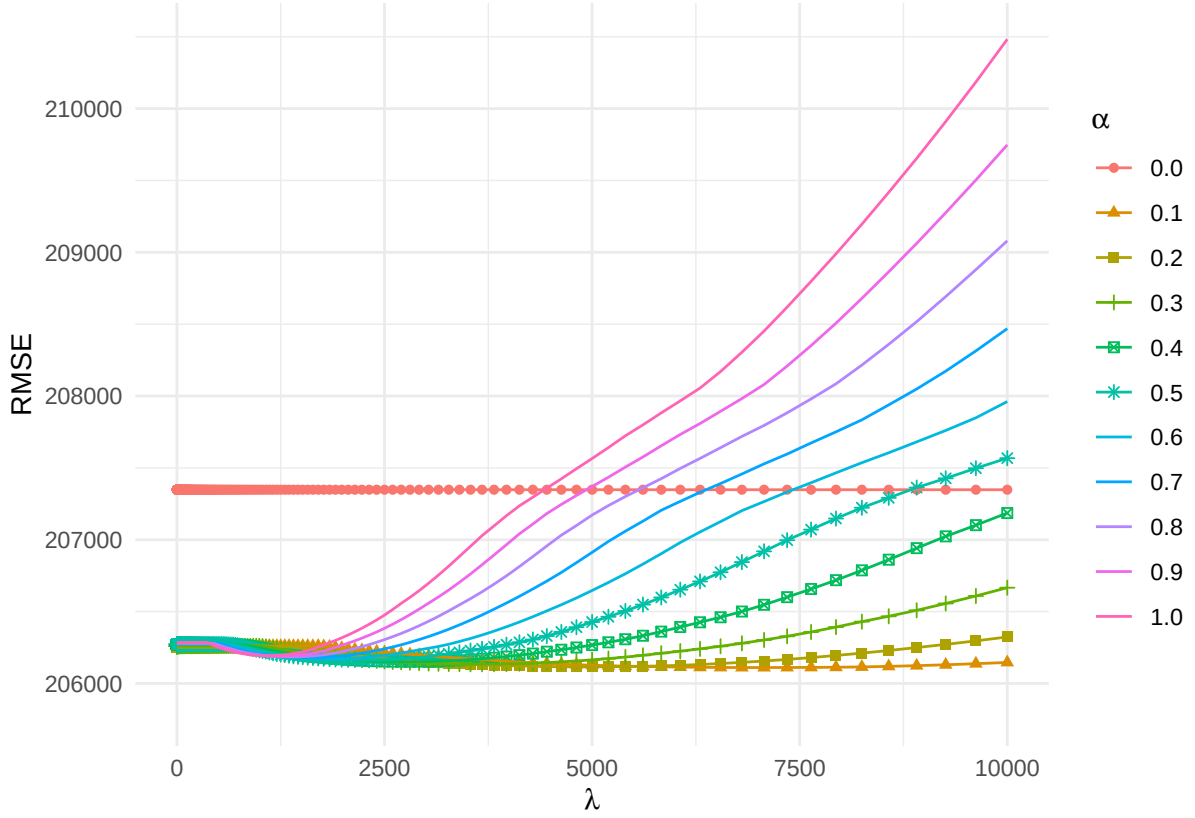
```
obs_lasso=predict(fit_lasso,newdata=x_train)
c(rmse(y_train,obs_lasso),mae(y_train,obs_lasso))
```

```
## [1] 205819.2 112077.4
```

```
pred_lasso=predict(fit_lasso,newdata=x_test)
c(rmse(y_test,pred_lasso),mae(y_test,pred_lasso))
```

```
## [1] 182830.7 108314.5
```

```
## ElasticNet

set.seed(123)
tune_elnet=expand.grid(alpha=seq(0,1,by=0.1),lambda=c(0,10^seq(log10(10000),log10(0.1),length.out=300)))
fit_elnet=train(buy_price~.,data=train_reg,method='glmnet',tuneGrid=tune_elnet,trControl=control)
plot_elnet=ggplot(fit_elnet) + ylim(205800,210500) + theme_minimal() + labs(x=expression(lambda),y='RMSE
  scale_color_discrete(name=expression(alpha)) + scale_shape_discrete(name=expression(alpha))
plot_elnet
```




```
fit_elnet$results[which.min(fit_elnet$results$RMSE),]
```

```
##     alpha   lambda     RMSE  Rsquared      MAE   RMSESD RsquaredSD   MAESD
## 593   0.1 7071.285 206110.6 0.7877472 113263.5 42632.52  0.0371397 8067.05
```

```
coef(fit_elnet$finalModel,fit_elnet$bestTune$lambda)
```

```
## 42 x 1 sparse Matrix of class "dgCMatrix"
##                                       s1
## (Intercept)                 1.195005e+05
## sq_mt_built                 3.126918e+03
## n_bathrooms                 7.127770e+04
## n_rooms                     2.377634e+03
## energy_certificateLow      -3.355637e+04
## floor                       5.533643e+03
## house_type_idAttic          1.188371e+05
## house_type_idIndependent   -2.034214e+05
## is_exteriorTRUE             .
## is_renewal_neededTRUE      -3.384917e+04
## is_new_developmentTRUE      9.735734e+04
## has_acTRUE                  1.126492e+04
## has_fitted_wardrobesTRUE   -1.445115e+04
## has_liftTRUE                5.561355e+02
## has_balconyTRUE            -4.960684e+03
## has_gardenTRUE              1.159656e+02
## has_parkingTRUE             1.693653e+04
## has_poolTRUE                3.235810e+04
## has_storage_roomTRUE        2.412862e+03
## has_individual_heatingTRUE  2.253978e+04
## d_supermarket               4.327984e+01
## d_hospital                 -1.664401e+01
## d_pharmacy                 -6.339132e+01
## d_post                     -1.318377e+01
## d_bank                     -4.690102e+01
## d_university               -2.206577e+01
## d_school                   -4.974161e+01
## d_kindergarten              1.550361e+02
## d_train                    -7.780459e+00
## d_bus                      -8.488299e+00
## d_airport                   4.037425e+00
## d_gym                       .
## d_park                      2.317813e+02
## d_stadium                  -1.142166e+01
## d_disco                    -6.323801e+00
## d_cinema                   -1.756156e+01
## d_library                   3.261735e+01
## d_historic                 -1.459673e+00
## d_attraction               -1.324225e+01
## lisaHigh-Low               -1.547282e+05
## lisaLow-High               -1.209102e+05
## lisaLow-Low                -1.663796e+05
```

```
obs_elnet=predict(fit_elnet,newdata=x_train)
c(rmse(y_train,obs_elnet),mae(y_train,obs_elnet))
```

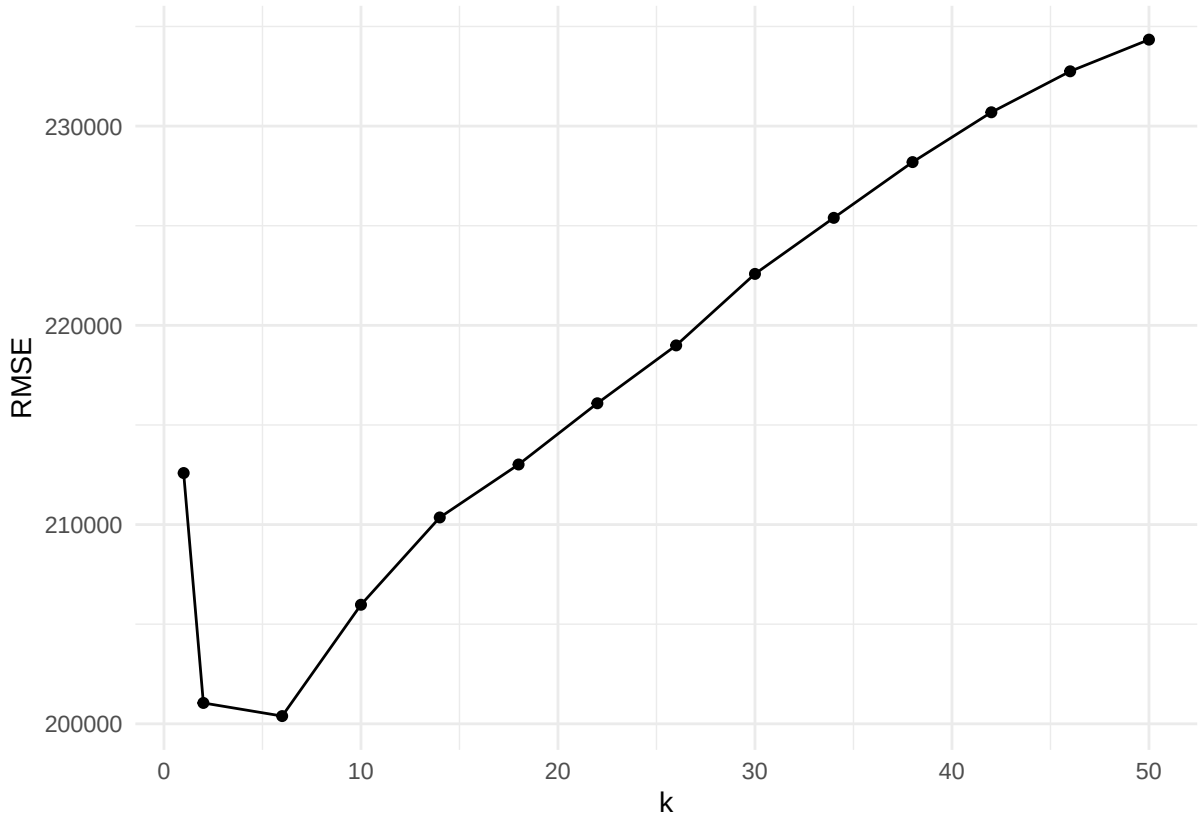
```
## [1] 205940.6 111880.3
```

```
pred_elnet=predict(fit_elnet,x_test)
c(rmse(y_test,pred_elnet),mae(y_test,pred_elnet))
```

```
## [1] 182403.0 108199.9
```

```
#reg_plot=ggarrange(ggarrange(plot_ridge,plot_lasso,ncol=2,labels=c('Ridge','Lasso')),
#                   plot_elnet,labels='ElasticNet',vjust=27,nrow=2)
#annotate_figure(reg_plot)
```

```
# Regressione Non-Parametrica

## K-nearest neighbors (KNN)

set.seed(123)
fit_knn=train(buy_price~.,data=train_reg,preProcess=c('center','scale'),method='knn',
              tuneGrid=expand.grid(k=c(1,seq(2,50,4))),trControl=control)
ggplot(fit_knn) + theme_minimal() + labs(x='k',y='RMSE')
```




```
fit_knn$results[which.min(fit_knn$results$RMSE),]
```

```
##   k     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 3 6 200388.9 0.8078803 90492.86 40940.98 0.03459265 4166.386
```

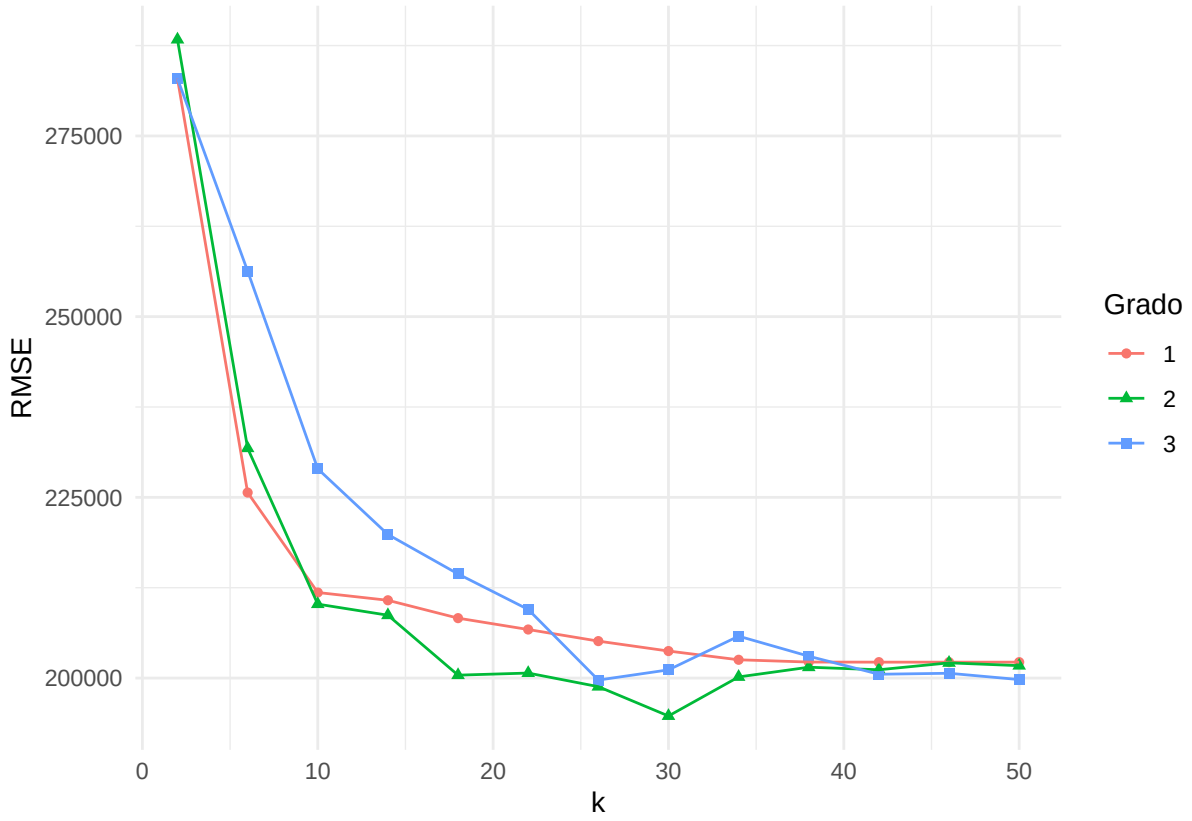
```
obs_knn=predict(fit_knn,newdata=x_train)
c(rmse(y_train,obs_knn),mae(y_train,obs_knn))
```

```
## [1] 165629.16  74384.41
```

```
pred_knn=predict(fit_knn,newdata=x_test)
c(rmse(y_test,pred_knn),mae(y_test,pred_knn))
```

```
## [1] 177983.17  90260.29
```

```
## Multivariate Adaptive Regression Spline (MARS)

set.seed(123)
fit_mars=train(buy_price~.,data=train_reg,method='earth',
               tuneGrid=expand.grid(degree=1:3,nprune=seq(2,50,by=4)),trControl=control)
ggplot(fit_mars) + theme_minimal() + labs(x='k',y='RMSE') +
  scale_color_discrete(name='Grado') + scale_shape_discrete(name='Grado')
```




```
fit_mars$results[which.min(fit_mars$results$RMSE),]
```

```
##    degree nprune     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 21      2     30 194750.6 0.8111468 98120.46 37704.02 0.04162673 7001.987
```

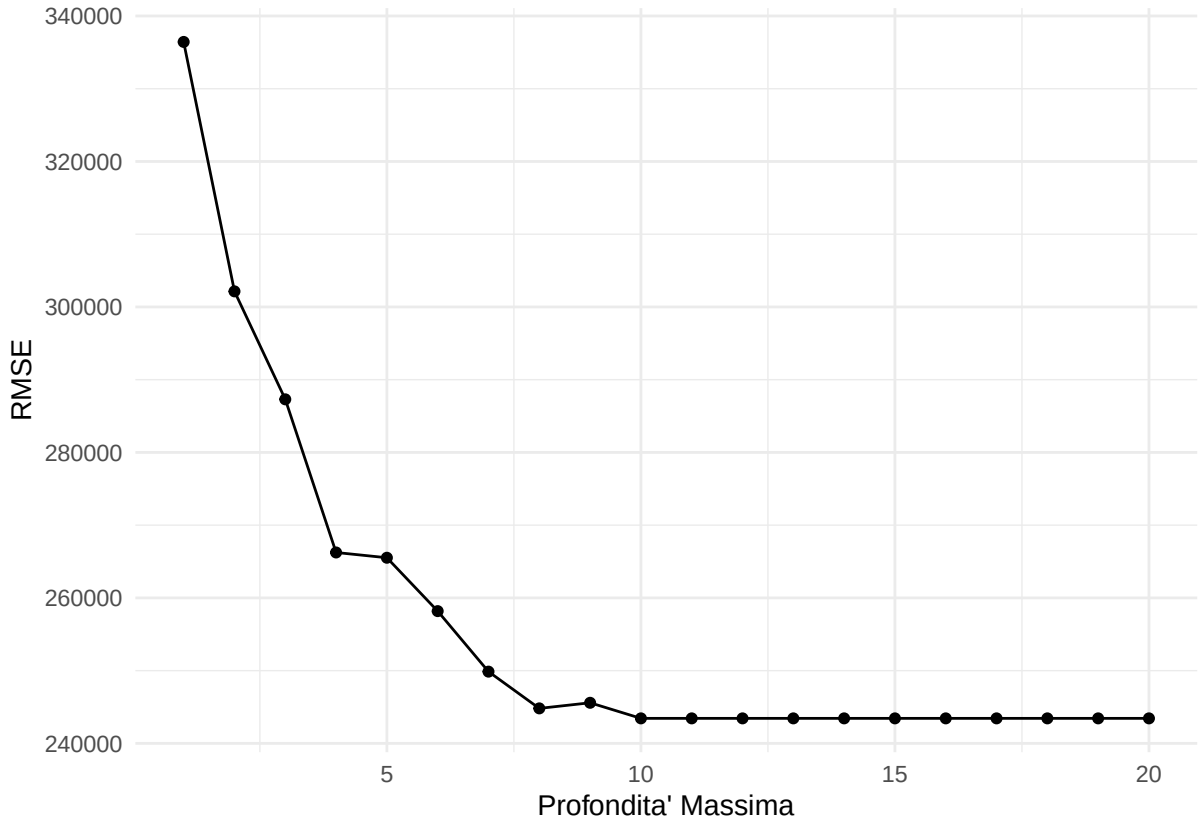
```
obs_mars=predict(fit_mars,newdata=x_train)
c(rmse(y_train,obs_mars),mae(y_train,obs_mars))
```

```
## [1] 151783.22  86081.75
```

```
pred_mars=predict(fit_mars,newdata=x_test)
c(rmse(y_test,pred_mars),mae(y_test,pred_mars))
```

```
## [1] 156169.67  88701.33
```

```
# Modelli basati sugli Alberi

## Albero Decisionale

set.seed(123)
tune_tree=expand.grid(maxdepth=1:20)
fit_tree=train(buy_price~.,data=train_reg,method='rpart2',tuneGrid=tune_tree,trControl=control)
ggplot(fit_tree) + theme_minimal() + labs(x="Profondita' Massima",y='RMSE')
```




```
fit_tree$results[which.min(fit_tree$results$RMSE),]
```

```
##    maxdepth     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 10       10 243439.1 0.7024124 129806.3 39838.93 0.04491422 5981.967
```

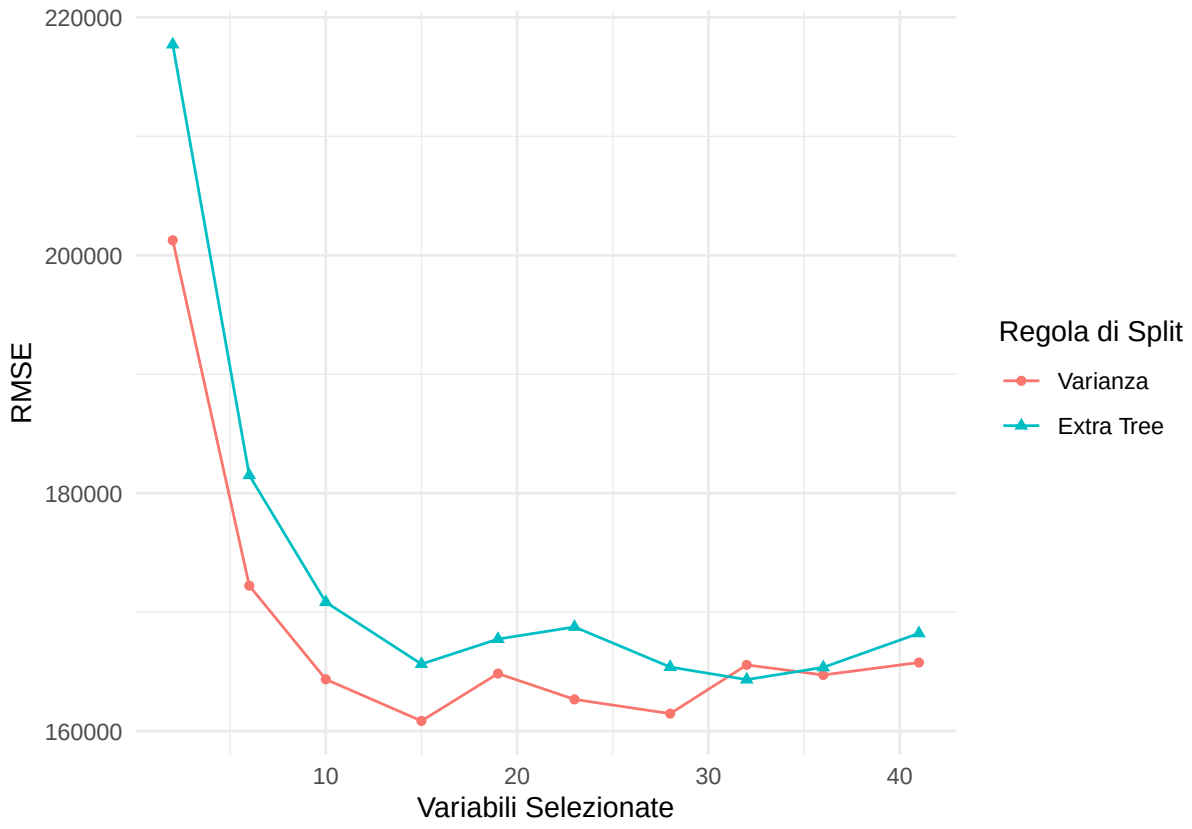
```
obs_tree=predict(fit_tree,newdata=x_train)
c(rmse(y_train,obs_tree),mae(y_train,obs_tree))
```

```
## [1] 218283.4 124474.8
```

```
pred_tree=predict(fit_tree,newdata=x_test)
c(rmse(y_test,pred_tree),mae(y_test,pred_tree))
```

```
## [1] 220336.1 127196.0
```

```
## Random Forest

set.seed(123)
fit_rf_split=train(buy_price~.,data=train_reg,method='ranger',importance='impurity',
                   num.trees=50,tuneLength=10,trControl=control)
ggplot(fit_rf_split) + theme_minimal() + labs(x='Variabili Selezionate',y='RMSE') +
  scale_color_discrete(name='Regola di Split',labels=c('Varianza','Extra Tree')) +
  scale_shape_discrete(name='Regola di Split',labels=c('Varianza','Extra Tree'))
```




```
fit_rf_split$results[which.min(fit_rf_split$results$RMSE),]
```

```
##   mtry min.node.size splitrule     RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 7   15             5  variance 160860.4 0.8721192 70700.93 44285.59  0.0360551
##      MAESD
## 7 5155.132
```

```
tune_rf=expand.grid(mtry=15,splitrule='variance',min.node.size=5)
fit_rf_n25=train(buy_price~.,data=train_reg,method='ranger',importance='impurity',
                 num.trees=25,tuneGrid=tune_rf,trControl=control)
fit_rf_n50=train(buy_price~.,data=train_reg,method='ranger',importance='impurity',
                 num.trees=50,tuneGrid=tune_rf,trControl=control)
fit_rf_n100=train(buy_price~.,data=train_reg,method='ranger',importance='impurity',
                  num.trees=100,tuneGrid=tune_rf,trControl=control)
fit_rf_n150=train(buy_price~.,data=train_reg,method='ranger',importance='impurity',
                  num.trees=150,tuneGrid=tune_rf,trControl=control)
fit_rf_n200=train(buy_price~.,data=train_reg,method='ranger',importance='impurity',
                  num.trees=200,tuneGrid=tune_rf,trControl=control)
fit_rf_n500=train(buy_price~.,data=train_reg,method='ranger',importance='impurity',
                  num.trees=500,tuneGrid=tune_rf,trControl=control)

fit_rf_n25$results[which.min(fit_rf_n25$results$RMSE),]
```

```
##   mtry splitrule min.node.size     RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 1   15  variance             5 164486.7 0.8647952 71518.04 45820.89 0.04640623
##      MAESD
## 1 5203.423
```

```
fit_rf_n50$results[which.min(fit_rf_n50$results$RMSE),]
```

```
##   mtry splitrule min.node.size   RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 1   15  variance             5 161119 0.8734158 71159.73 40986.86 0.03136113
##     MAESD
## 1 4866.74
```

```
fit_rf_n100$results[which.min(fit_rf_n100$results$RMSE),]
```

```
##   mtry splitrule min.node.size     RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 1   15  variance             5 164551.9 0.8643183 70798.22 41322.66 0.04438159
##      MAESD
## 1 5775.854
```

```
fit_rf_n150$results[which.min(fit_rf_n150$results$RMSE),]
```

```
##   mtry splitrule min.node.size   RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 1   15  variance             5 160430 0.8760274 70364.54 52502.72  0.0305007
##     MAESD
## 1 8168.97
```

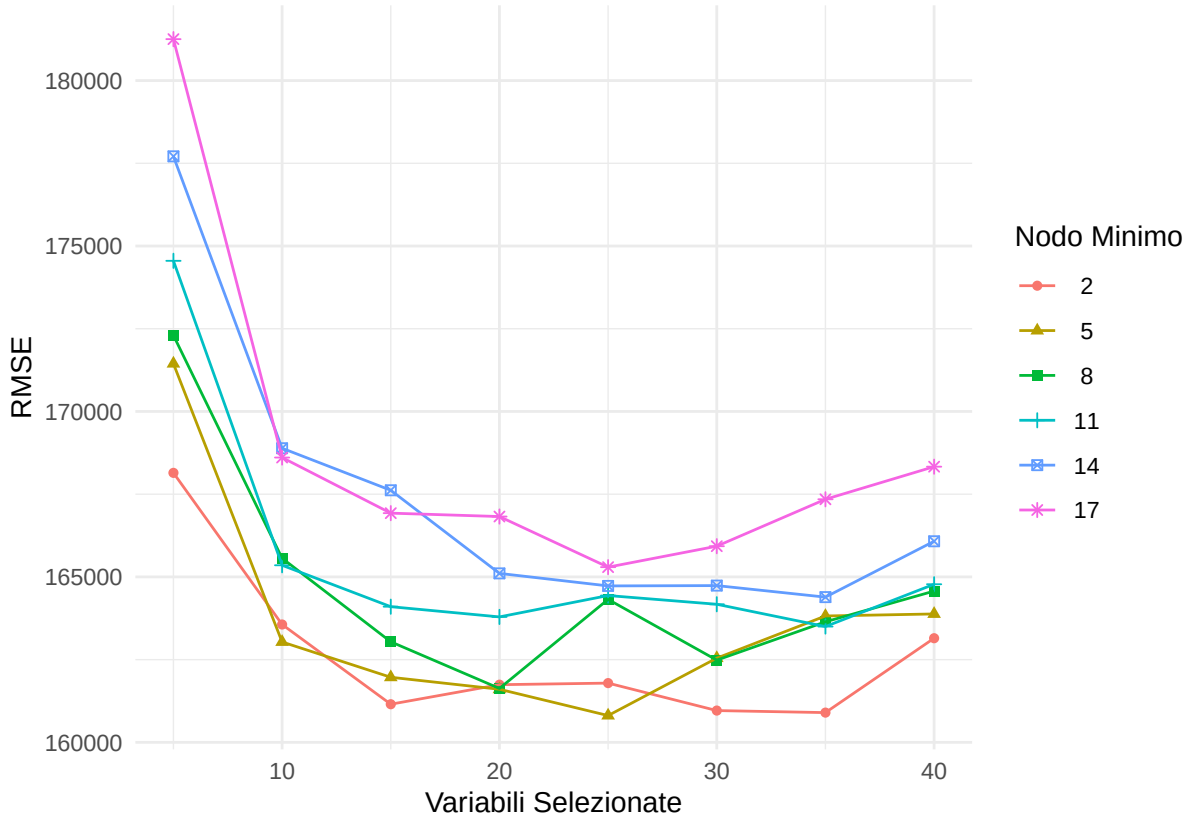
```
fit_rf_n200$results[which.min(fit_rf_n200$results$RMSE),]
```

```
##   mtry splitrule min.node.size     RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 1   15  variance             5 161557.8 0.8699279 69782.77 40863.41 0.04530318
##      MAESD
## 1 2649.401
```

```
fit_rf_n500$results[which.min(fit_rf_n500$results$RMSE),]
```

```
##   mtry splitrule min.node.size     RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 1   15  variance             5 163927.4 0.8683842 70071.99 36506.62 0.03898208
##      MAESD
## 1 3761.338
```

```
set.seed(123)
tune_rf=expand.grid(mtry=seq(5,40,5),splitrule='variance',min.node.size=seq(2,17,3))
fit_rf=train(buy_price~.,data=train_reg,method='ranger',importance='impurity',
             num.trees=150,tuneGrid=tune_rf,trControl=control)
ggplot(fit_rf) + theme_minimal() + labs(x='Variabili Selezionate',y='RMSE') +
  scale_color_discrete(name='Nodo Minimo') + scale_shape_discrete(name='Nodo Minimo')
```




```
fit_rf$results[which.min(fit_rf$results$RMSE),]
```

```
##    mtry splitrule min.node.size     RMSE  Rsquared      MAE   RMSESD RsquaredSD
## 26   25  variance             5 160809.6 0.8720488 70682.79 42636.75 0.03478304
##       MAESD
## 26 6360.411
```

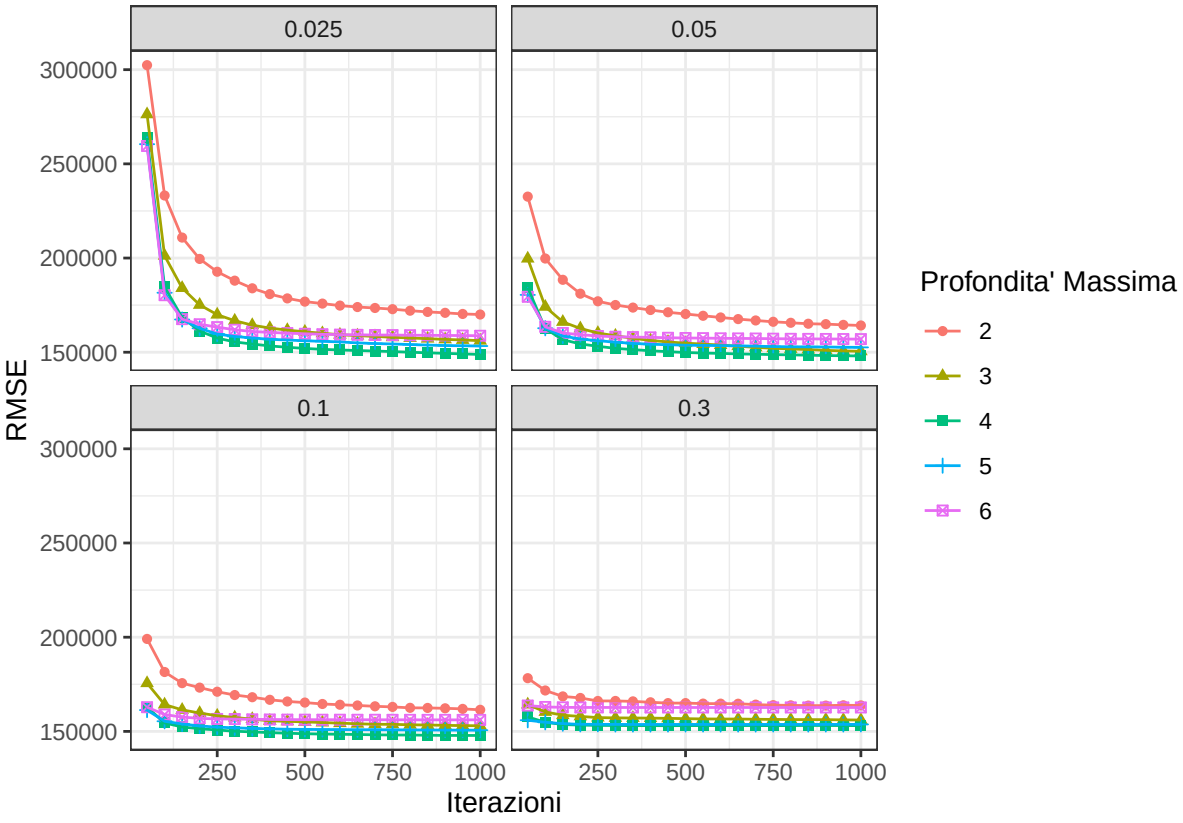
```
obs_rf=predict(fit_rf,newdata=x_train)
c(rmse(y_train,obs_rf),mae(y_train,obs_rf))
```

```
## [1] 68387.77 28319.37
```

```
pred_rf=predict(fit_rf,newdata=x_test)
c(rmse(y_test,pred_rf),mae(y_test,pred_rf))
```

```
## [1] 132151.6  64077.1
```

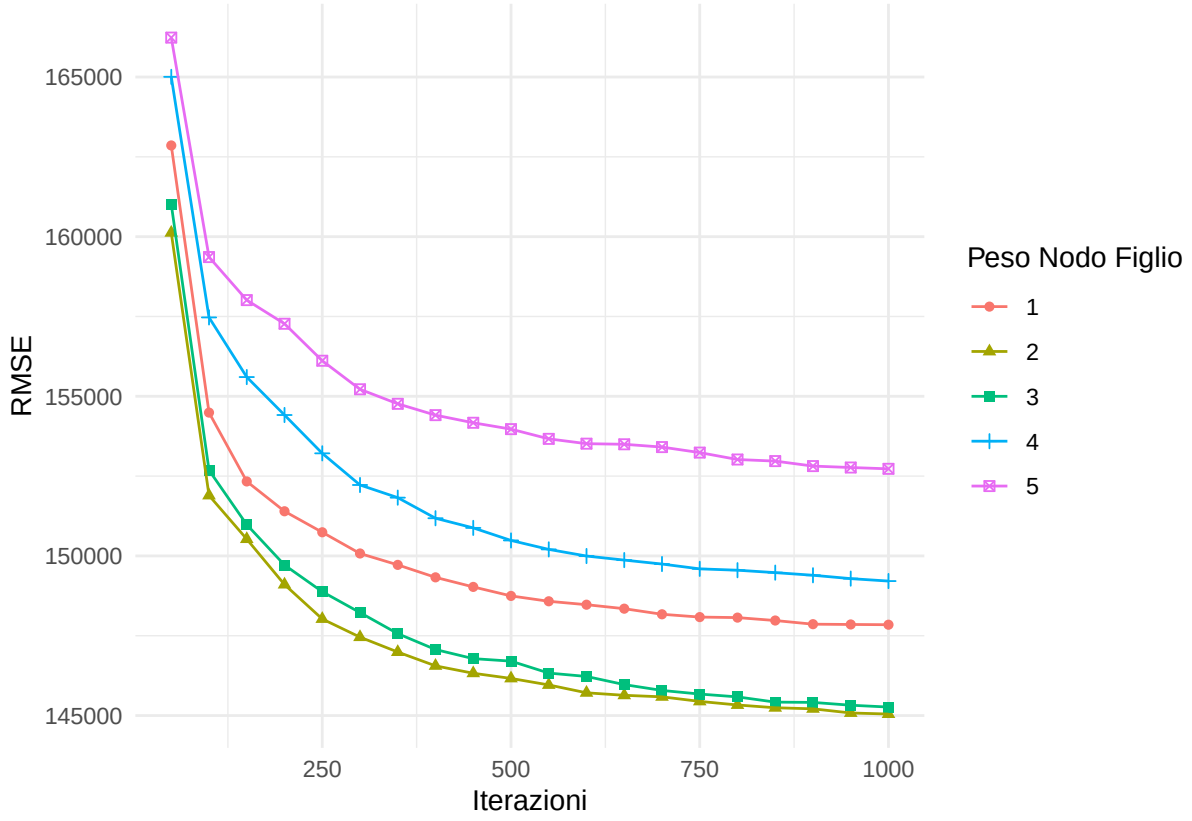
```
## XG-Boost

set.seed(123)
tune_xgb1=expand.grid(nrounds=seq(50,1000,50),max_depth=2:6,eta=c(0.025,0.05,0.1,0.3),gamma=0,
                      min_child_weight=1,subsample=1,colsample_bytree=1)
fit_xgb1=train(buy_price~.,data=train_reg,method='xgbTree',tuneGrid=tune_xgb1,trControl=control)
ggplot(fit_xgb1) + theme_bw() + labs(x='Iterazioni',y='RMSE') +
  scale_color_discrete(name="Profondita' Massima") + scale_shape_discrete(name="Profondita' Massima")
```




```
fit_xgb1$results[which.min(fit_xgb1$results$RMSE),]
```

```
##     eta max_depth gamma colsample_bytree min_child_weight subsample nrounds
## 260 0.1         4     0                1                1         1    1000
##         RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 260 147844.8 0.8912704 65020.97 33390.83 0.02786055 4810.507
```

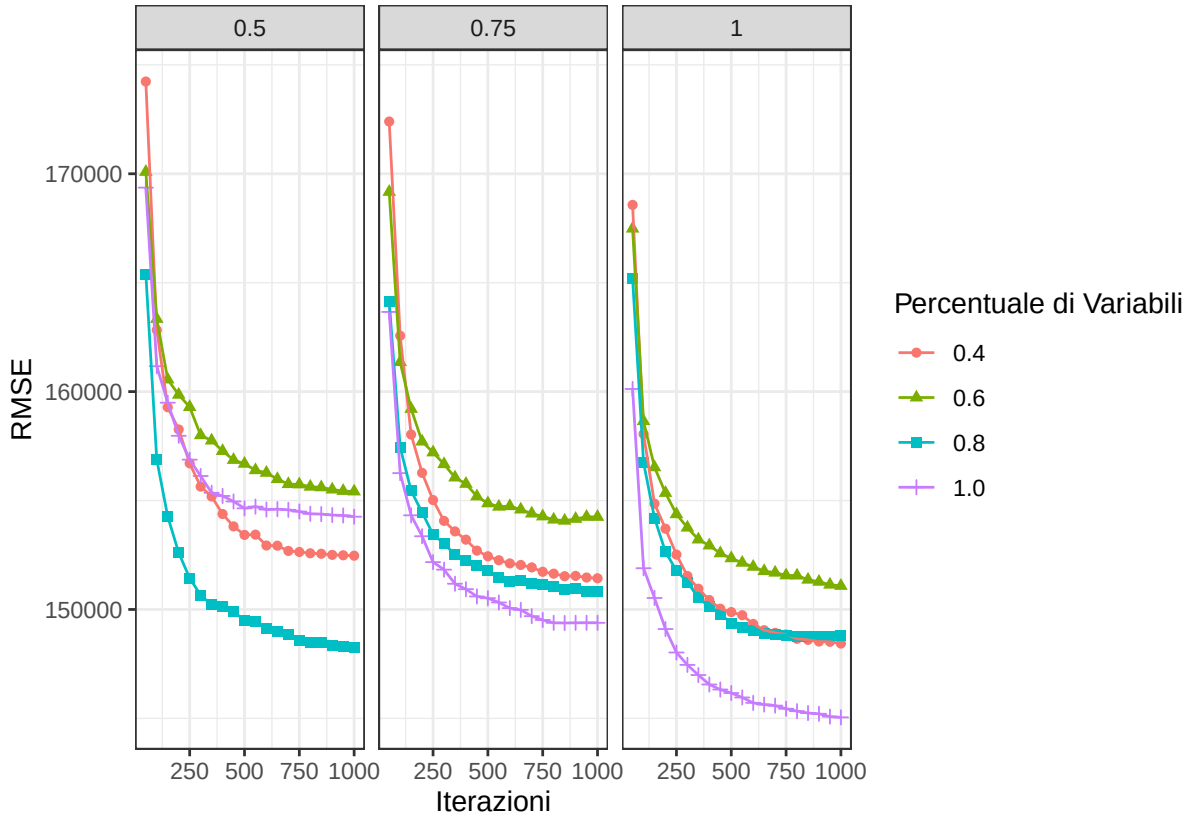
```
set.seed(123)
tune_xgb2=expand.grid(nrounds=seq(50,1000,50),max_depth=4,eta=0.1,gamma=0,
                      min_child_weight=1:5,subsample=1,colsample_bytree=1)
fit_xgb2=train(buy_price~.,data=train_reg,method='xgbTree',tuneGrid=tune_xgb2,trControl=control)
ggplot(fit_xgb2) + theme_minimal() + labs(x='Iterazioni',y='RMSE') +
  scale_color_discrete(name='Peso Nodo Figlio') + scale_shape_discrete(name='Peso Nodo Figlio')
```




```
fit_xgb2$results[which.min(fit_xgb2$results$RMSE),]
```

```
##    eta max_depth gamma colsample_bytree min_child_weight subsample nrounds
## 40 0.1         4     0                1                2         1    1000
##        RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 40 145046.9 0.8952544 64472.44 34468.85 0.02973147 5829.137
```

```
set.seed(123)
tune_xgb3=expand.grid(nrounds=seq(50,1000,50),max_depth=4,eta=0.1,gamma=0,
                      min_child_weight=2,subsample=c(0.5,0.75,1),colsample_bytree=c(0.4,0.6,0.8,1))
fit_xgb3=train(buy_price~.,data=train_reg,method='xgbTree',tuneGrid=tune_xgb3,trControl=control)
ggplot(fit_xgb3) + theme_bw() + labs(x='Iterazioni',y='RMSE') +
  scale_color_discrete(name='Percentuale di Variabili') +
  scale_shape_discrete(name='Percentuale di Variabili')
```



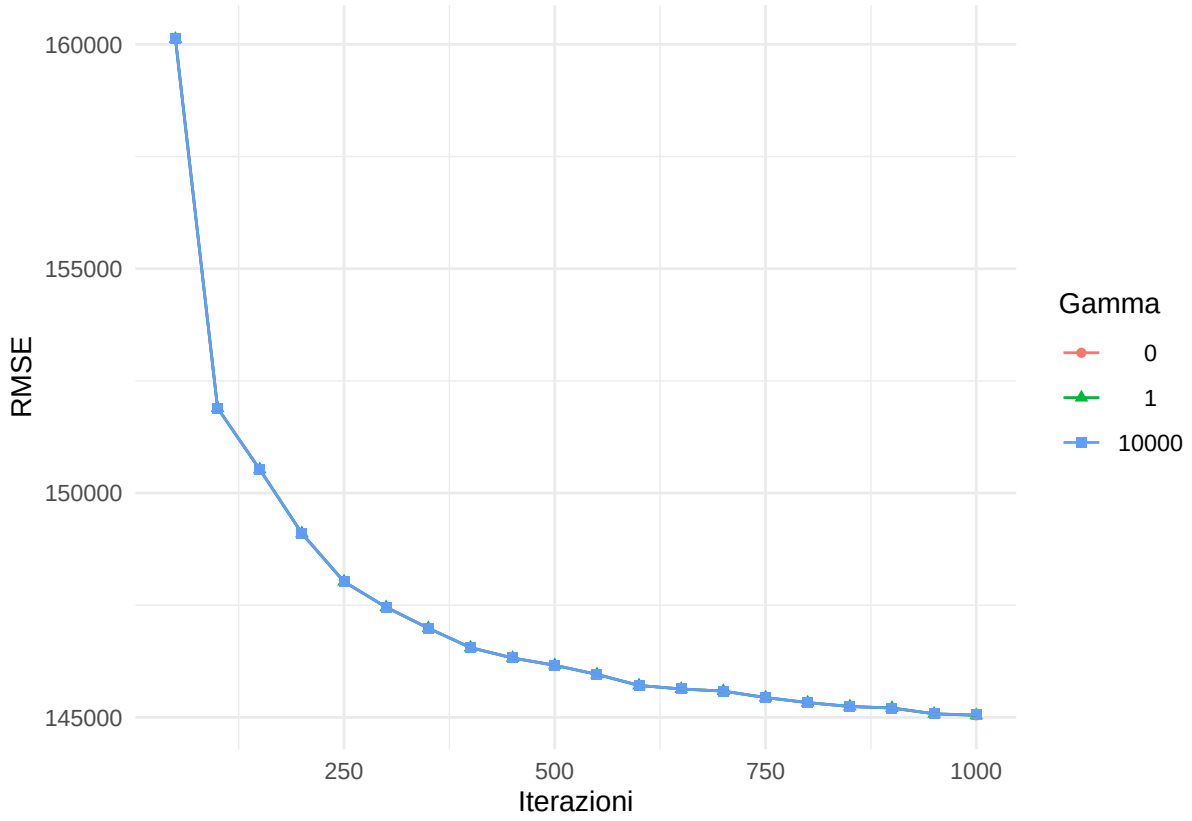

```
fit_xgb3$results[which.min(fit_xgb3$results$RMSE),]
```

```
##     eta max_depth gamma colsample_bytree min_child_weight subsample nrounds
## 240 0.1         4     0                1                2         1    1000
##         RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 240 145046.9 0.8952544 64472.44 34468.85 0.02973147 5829.137
```

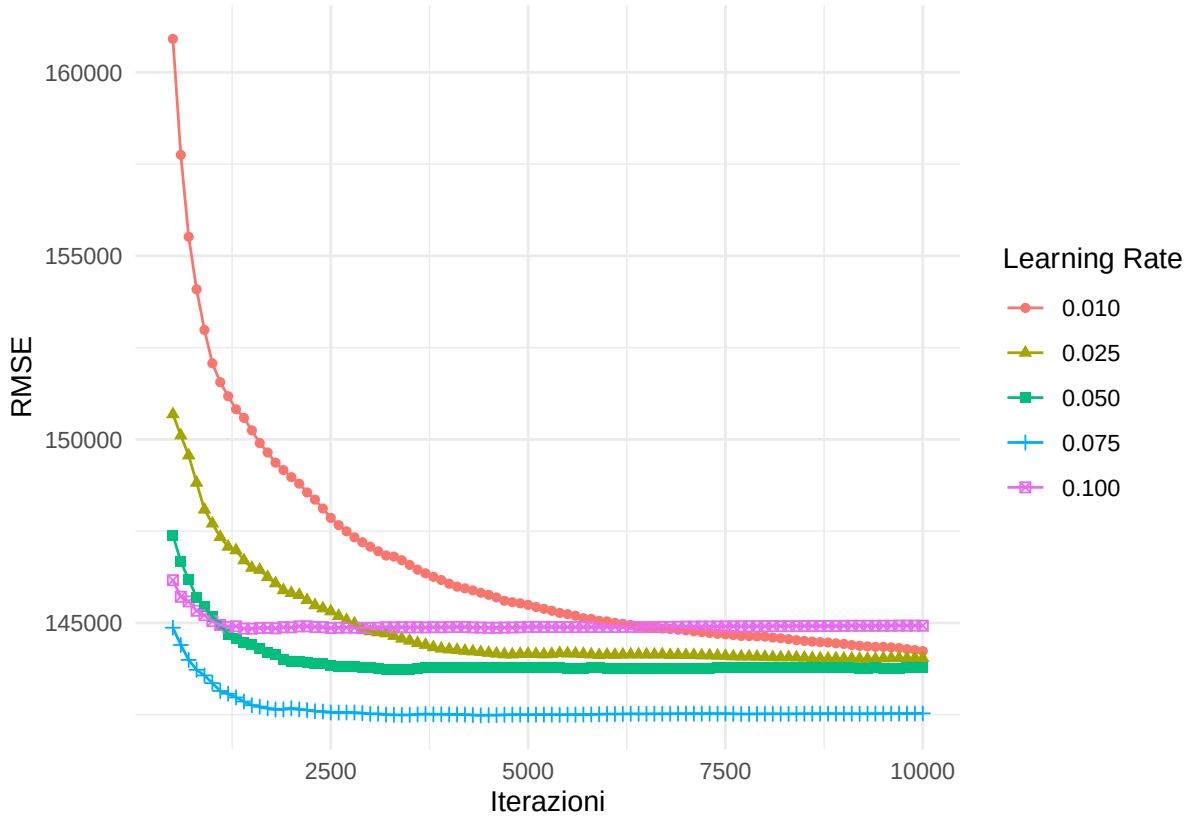
```
set.seed(123)
tune_xgb4=expand.grid(nrounds=seq(50,1000,50),max_depth=4,eta=0.1,gamma=c(0,1,10000),
                      min_child_weight=2,subsample=1,colsample_bytree=1)
fit_xgb4=train(buy_price~.,data=train_reg,method='xgbTree',tuneGrid=tune_xgb4,trControl=control)
fit_xgb4$results[which.min(fit_xgb4$results$RMSE),]
```

```
##    eta max_depth gamma colsample_bytree min_child_weight subsample nrounds
## 20 0.1         4     0                1                2         1    1000
##        RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 20 145046.9 0.8952544 64472.44 34468.85 0.02973147 5829.137
```

```
ggplot(fit_xgb4) + theme_minimal() + labs(x='Iterazioni',y='RMSE') +
  scale_color_discrete(name='Gamma') + scale_shape_discrete(name='Gamma')
```




```
set.seed(123)
tune_xgb5=expand.grid(nrounds=seq(500,10000,100),max_depth=4,eta=c(0.01,0.025,0.05,0.075,0.1),gamma=0,
                      min_child_weight=2,subsample=1,colsample_bytree=1)
fit_xgb5=train(buy_price~.,data=train_reg,method='xgbTree',tuneGrid=tune_xgb5,trControl=control)
ggplot(fit_xgb5) + theme_minimal() + labs(x='Iterazioni',y='RMSE') +
  scale_color_discrete(name='Learning Rate') + scale_shape_discrete(name='Learning Rate')
```




```
fit_xgb5$results[which.min(fit_xgb5$results$RMSE),]
```

```
##       eta max_depth gamma colsample_bytree min_child_weight subsample nrounds
## 328 0.075         4     0                1                2         1    4400
##         RMSE  Rsquared      MAE  RMSESD RsquaredSD    MAESD
## 328 142481.5 0.8989825 63477.85 32346.8 0.02672123 4964.076
```

```
obs_xgb=predict(fit_xgb5,newdata=x_train)
c(rmse(y_train,obs_xgb),mae(y_train,obs_xgb))
```

```
## [1] 7367.810 4729.638
```

```
pred_xgb=predict(fit_xgb5,newdata=x_test)
c(rmse(y_test,pred_xgb),mae(y_test,pred_xgb))
```

```
## [1] 122237.11  58936.02
```

```
# Modello Spaziale

k_neigh_75=knn2nb(knearneigh(train_coord,k=75,longlat=TRUE))
k_distance_75=nbdists(k_neigh_75,train_coord,longlat=TRUE)
inv_k_distance_75=lapply(k_distance_75, function(x) (1/(x+0.001)))
k_weight_75=nb2listw(k_neigh_75,glist=inv_k_distance_75,style='W')
```

```
set.seed(123)
moran.mc(fit_lm$finalModel$residuals,k_weight_75,nsim=100,alternative='two.sided')
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  fit_lm$finalModel$residuals
## weights: k_weight_75
## number of simulations + 1: 101
##
## statistic = 0.42235, observed rank = 101, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

```
#lm.morantest(fit_lm$finalModel,k_weight_75,alternative='two.sided',resfun=residuals)

lm.LMtests(fit_lm$finalModel,k_weight_75,test=c('LMlag','LMerr'))
```

```
##
##  Lagrange multiplier diagnostics for spatial dependence
##
## data:
## model: lm(formula = .outcome ~ sq_mt_built + n_bathrooms + n_rooms +
## house_type_idAttic + house_type_idIndependent + is_exteriorTRUE +
## is_new_developmentTRUE + has_acTRUE + has_liftTRUE + has_gardenTRUE +
## has_individual_heatingTRUE, data = dat)
## weights: k_weight_75
##
## LMlag = 2112.4, df = 1, p-value < 2.2e-16
##
##
##  Lagrange multiplier diagnostics for spatial dependence
##
## data:
## model: lm(formula = .outcome ~ sq_mt_built + n_bathrooms + n_rooms +
## house_type_idAttic + house_type_idIndependent + is_exteriorTRUE +
## is_new_developmentTRUE + has_acTRUE + has_liftTRUE + has_gardenTRUE +
## has_individual_heatingTRUE, data = dat)
## weights: k_weight_75
##
## LMerr = 2871, df = 1, p-value < 2.2e-16
```

```
lm.LMtests(fit_lm$finalModel,k_weight_75,test=c('RLMlag','RLMerr'))
```

```
##
##  Lagrange multiplier diagnostics for spatial dependence
##
## data:
## model: lm(formula = .outcome ~ sq_mt_built + n_bathrooms + n_rooms +
## house_type_idAttic + house_type_idIndependent + is_exteriorTRUE +
## is_new_developmentTRUE + has_acTRUE + has_liftTRUE + has_gardenTRUE +
## has_individual_heatingTRUE, data = dat)
```

```
## weights: k_weight_75
##
## RLMlag = 262.54, df = 1, p-value < 2.2e-16
##
##
##  Lagrange multiplier diagnostics for spatial dependence
##
## data:
## model: lm(formula = .outcome ~ sq_mt_built + n_bathrooms + n_rooms +
## house_type_idAttic + house_type_idIndependent + is_exteriorTRUE +
## is_new_developmentTRUE + has_acTRUE + has_liftTRUE + has_gardenTRUE +
## has_individual_heatingTRUE, data = dat)
## weights: k_weight_75
##
## RLMerr = 1021.1, df = 1, p-value < 2.2e-16
```

```
## Modello di Errore Spaziale (SEM)

fit_err=errorsarlm(buy_price~sq_mt_built+n_bathrooms+n_rooms+house_type_id+is_exterior+
               is_new_development+has_ac+has_lift+has_garden+has_individual_heating,
               data=train,listw=k_weight_75,method='MC',tol.solve=1e-30)

moran.mc(fit_err$residuals,k_weight_75,nsim=100,alternative='two.sided')
```

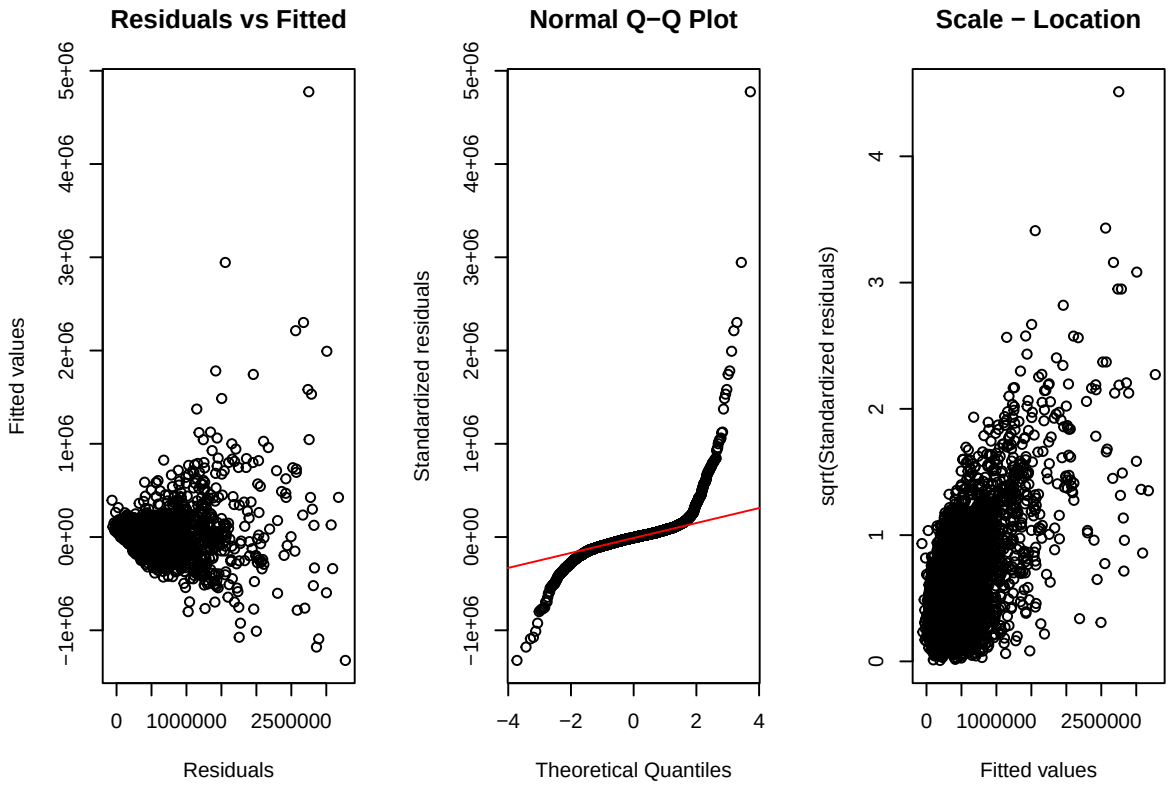
```
##
##  Monte-Carlo simulation of Moran I
##
## data:  fit_err$residuals
## weights: k_weight_75
## number of simulations + 1: 101
##
## statistic = -0.0066871, observed rank = 17, p-value = 0.3366
## alternative hypothesis: two.sided
```

```
summary(fit_err)
```

```
##
## Call:errorsarlm(formula = buy_price ~ sq_mt_built + n_bathrooms +
##     n_rooms + house_type_id + is_exterior + is_new_development +
##     has_ac + has_lift + has_garden + has_individual_heating,
##     data = train, listw = k_weight_75, method = "MC", tol.solve = 1e-30)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1322783.5   -63949.1    -6091.6    44166.1  4775112.7
##
## Type: error
## Coefficients: (asymptotic standard errors)
##                               Estimate  Std. Error z value  Pr(>|z|)
## (Intercept)                -137000.839   17423.695 -7.8629 3.775e-15
## sq_mt_built                   3082.982      67.901 45.4040 < 2.2e-16
## n_bathrooms                  70110.614    5198.703 13.4862 < 2.2e-16
## n_rooms                      13075.771    3098.484  4.2201 2.442e-05
```

```
## house_type_idAttic          120566.823   10536.107 11.4432 < 2.2e-16
## house_type_idIndependent   -143710.836   19392.205 -7.4108 1.257e-13
## is_exteriorTRUE              13594.701    9957.061  1.3653 0.1721485
## is_new_developmentTRUE      137328.310   13239.326 10.3728 < 2.2e-16
## has_acTRUE                   22934.658    6122.293  3.7461 0.0001796
## has_liftTRUE                 22680.855    8256.705  2.7470 0.0060150
## has_gardenTRUE                4272.675    7677.514  0.5565 0.5778568
## has_individual_heatingTRUE   16144.476    7525.405  2.1453 0.0319265
##
## Lambda: 0.73698, LR test value: 2228.7, p-value: < 2.22e-16
## Approximate (numerical Hessian) standard error: 0.010657
##     z-value: 69.152, p-value: < 2.22e-16
## Wald statistic: 4782, p-value: < 2.22e-16
##
## Log likelihood: -68422.68 for error model
## ML residual variance (sigma squared): 3.3327e+10, (sigma: 182560)
## Number of observations: 5033
## Number of parameters estimated: 14
## AIC: 136870, (AIC for lm: 139100)
```

```
### Diagnostiche

y=fit_err$y
X=fit_err$X

fitted=X %*% fit_err$coefficients
residuals=y-fitted
H=X %*% solve(t(X) %*% X) %*% t(X)
variance=sum(residuals^2)/(nrow(X)-ncol(X))
stdres=sqrt(abs(residuals/sqrt(variance*(1-diag(H)))))

par(mfrow=c(1,3))
plot(fit_err$fitted.values,fit_err$residuals,xlab='Residuals',ylab='Fitted values',
     main='Residuals vs Fitted')
qqnorm(fit_err$residuals,ylab='Standardized residuals')
qqline(fit_err$residuals,col='red')
plot(fit_err$fitted.values,stdres,xlab='Fitted values',ylab='sqrt(Standardized residuals)',
     main='Scale - Location')
```




```
### Errore Previsivo Spaziale

cv_error_sem=function(data,k,coord,v,seed=123){
  set.seed(seed)
  yourdata=data[sample(nrow(data)),]
  fold=cut(seq(1,nrow(yourdata)),breaks=k,labels=FALSE)
  rmse_val=NULL
  mae_val=NULL
  r2_val=NULL
  for(i in 1:k){
    test_index=which(fold==i,arr.ind=TRUE)
    test_data=yourdata[test_index,]
    train_data=yourdata[-test_index,]
    te_coord=coord[test_index,]
    tr_coord=coord[-test_index,]

    k_neigh_75=knn2nb(knearneigh(tr_coord,k=v,longlat=TRUE))
    k_distance_75=nbdists(k_neigh_75,tr_coord,longlat=TRUE)
    inv_k_distance_75=lapply(k_distance_75, function(x) (1/(x+0.001)))
    k_weight_75=nb2listw(k_neigh_75,glist=inv_k_distance_75,style='W')

    model=errorsarlm(buy_price~sq_mt_built+n_bathrooms+n_rooms+house_type_id+
                       is_exterior+is_new_development+has_ac+has_lift+has_garden+
                       has_individual_heating,data=train_data,
                     listw=k_weight_75,method="MC",tol.solve=1e-30)
```

```
    cc=as.data.frame(rbind(tr_coord,te_coord))

    k_neigh_75_test=knn2nb(knearneigh(cc,k=v,longlat=TRUE))
    k_distance_75_test=nbdists(k_neigh_75_test,cc,longlat=TRUE)
    inv_k_distance_75_t=lapply(k_distance_75_test, function(x) (1/(x+0.001)))
    k_weight_75_test=nb2listw(k_neigh_75_test,glist=inv_k_distance_75_t,style='W')

    pred=predict(model,newdata=test_data[,-1],listw=k_weight_75_test,pred.type='trend')

    rmse_val[i]=rmse(test_data[,1],pred)
    r2_val[i]=cor(test_data[,1],pred)^2
    mae_val[i]=mae(test_data[,1],pred)
  }
  rmse_mean=mean(rmse_val)
  r2_mean=mean(r2_val)
  mae_mean=mean(mae_val)
  cv_error=as.data.frame(cbind(rmse_mean,r2_mean,mae_mean))
  colnames(cv_error)=c('RMSE','Rsquared','MAE')
  return(cv_error)
}

cv_error_sem(train,10,train_coord,75)  # errore di cross-validation
```

```
##       RMSE  Rsquared      MAE
## 1 242303.4 0.7007407 143713.9
```

```
rmse(y_train,fit_err$fitted.values)
```

```
## [1] 182558.1
```

```
mae(y_train,fit_err$fitted.values)
```

```
## [1] 90825.75
```

```
cc=as.data.frame(rbind(train_coord,test_coord))

k_neigh_75_test=knn2nb(knearneigh(cc,k=75,longlat=TRUE))
k_distance_75_test=nbdists(k_neigh_75_test,cc,longlat=TRUE)
inv_k_distance_75_t=lapply(k_distance_75_test, function(x) (1/(x+0.001)))
k_weight_75_test=nb2listw(k_neigh_75_test,glist=inv_k_distance_75_t,style='W')

pred_err=predict(fit_err,newdata=x_test,listw=k_weight_75_test,pred.type='trend')

rmse(y_test,pred_err)
```

```
## [1] 232233.9
```

```
mae(y_test,pred_err)
```

```
## [1] 147464.5
```

Bibliografia

[1] European Medicines Agency. *Guideline on Missing Data in Confirmatory Clinical Trials*. 2009.

[2] Jesùs Mur, Ana Angulo. *Model Selection Strategies in a Spatial Setting: Some Additional Results*. 2009.

[3] Luc Anselin. *Local Indicators of Spatial Association - LISA*. 1995.

[4] Evangelos Kontopantelis, Ian R. White, Matthew Sperrin, Iain Buchan. *Outcome-sensitive multiple imputation: a simulation study*. 2017.

[5] Harris Drucker. *Improving Regressors using Boosting Techniques*. 1997.

[6] Craig K. Enders. *Applied Missing Data Analysis*. 1^{st} edition, 2010.

[7] Nizar Bouguila, Wentao Fao. *Mixture Models and Applications*. 2020.

[8] Jerome H. Friedman. *Multivariate Adaptive Regression Splines*. 1990.

[9] Jerome H. Friedman. *Stochastic Gradient Boosting*. 1999.

[10] Bor-Ming Hsieh. *Analisi della dipendenza spaziale dei prezzi delle abitazioni e dei sottomercati abitativi nella Tainan Metropolis, Taiwan*. 2012.

[11] Rebecca R. Andridge, Roderick J. A. Little. *A Review of Hot Deck Imputation for Survey Non-response*. 2011.

[12] Greg Ridgeway, David Madigan, and Thomas Richardson. *Boosting Methodology for Regression Problems*. 1999.

[13] Jeffrey Naf, Meta-Lina Spohn, Loris Michel, Nicolai Meinshausen. *Imputation Scores*. 2021.

[14] Mike Nguyen. *A Guide on Data Analysis*. 2022.

[15] Alessandra Salvan, Nicola Sartori, Luigi Pace. *Modelli Statistici 2*. 2019.

[16] David Randahl. *Raoul: An R-Package for Handling Missing Data*. 2016.

[17] Roderick J. A. Little, Donald B. Rubin. *Statistical Analysis with Missing Data*. 3^{rd} edition, 2020.

[18] Tobias Ruttenauer. *Spatial Regression Models: A Systematic Comparison of Different Model Specifications Using Monte Carlo Experiments*. 2019.

[19] Yoav Freund, Robert E. Schapire. *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. 1997.

[20] Nathaniel Schenker, Jeremy M. G. Taylor. *Partially parametric techniques for multiple imputation*. 1996.

[21] Stef van Buuren. *Flexible Imputation of Missing Data*. 2^{nd} edition, 2018.

[22] Matteo Grigoletto, Francesco Pauli, Laura Ventura. *Modello Lineare*. 2017.

[23] Geert Molenberghs, Garrett Fitzmaurice, Michael G. Kenward, Anastasios Tsiatis, Geert Verbeke. *Handbook of Missing Data Methodology*. 2015.

[24] Luc Anselin, Anil K. Bera, Raymond Florax, Mann J. Yoon. *Simple diagnostic tests for spatial dependence*. 1996.

[25] Guangqing Chi, Jun Zhu. *Spatial Regression Models for the Social Sciences*. 2021.