UNIVERSITÀ DEGLI STUDI
DI SALERNO

Corso di Laurea in Informatica

PriceMyHouse

Autori:
Luigi Potestà
Giulia Buonafine

Docenti:
Prof. Polese Giuseppe
Prof.ssa Caruccio Loredana

Indice

1. Introduzione

Il mercato immobiliare rappresenta uno dei settori più complessi e dinamici dell'economia moderna. La valutazione accurata del prezzo di un immobile è un processo che richiede competenze multidisciplinari, conoscenza del territorio, analisi di variabili eterogenee e una forte componente di esperienza soggettiva. Negli ultimi anni, tuttavia, l'avanzamento delle tecniche di Machine Learning ha reso possibile affrontare questo problema in modo più sistematico, oggettivo e scalabile.

L'obiettivo del progetto *PriceMyHouse* è sviluppare un modello predittivo in grado di stimare il valore di un immobile sulla base delle sue caratteristiche strutturali, qualitative e contestuali. Il progetto si inserisce nel più ampio contesto dell'applicazione del Machine Learning al settore Real Estate, un ambito in cui la disponibilità di dati strutturati e la maturità degli algoritmi di regressione consentono di ottenere stime sempre più affidabili.

La sfida principale consiste nel trattare un dataset ricco ma complesso, caratterizzato da variabili eterogenee (numeriche, categoriche, ordinali), valori mancanti, differenze qualitative difficili da codificare e relazioni non lineari tra le feature. Per affrontare queste problematiche, il progetto adotta un approccio modulare e professionale, basato su una pipeline di preprocessing rigorosa e sull'utilizzo di modelli avanzati come XGBoost, particolarmente adatti alla gestione di dati tabellari.

Il progetto è stato sviluppato seguendo una struttura chiara e riproducibile, con l'obiettivo di garantire trasparenza, manutenibilità e possibilità di estensione futura. Ogni fase, dalla comprensione dei dati alla preparazione del dataset, dalla selezione del modello alla valutazione finale, è stata implementata in moduli Python dedicati.

2. Data Understanding

2.1 Origine del Dataset

Il progetto utilizza *Ames Housing Dataset*, un dataset ampiamente adottato nella comunità del Machine Learning per la predizione dei prezzi immobiliari. Il dataset è stato originariamente raccolto dal City Assessor's Office della città di Ames (Iowa, USA) e contiene informazioni dettagliate su oltre 2.900 abitazioni vendute tra il 2006 e il 2010.

La ricchezza delle feature, la qualità dei dati e la presenza di variabili sia numeriche che categoriche rendono questo dataset particolarmente adatto allo sviluppo di modelli predittivi avanzati.

2.2 Struttura del Dataset Originale

Il dataset Ames Housing è composto da numerose variabili che descrivono in modo dettagliato le caratteristiche strutturali, qualitative e contestuali di ciascun immobile.

Di seguito vengono riportate le principali categorie di feature, accompagnate dal nome originale utilizzato nel dataset.

Caratteristiche generali dell'immobile:

Feature	Descrizione
MSSubClass	Classe edilizia dell'abitazione
MSZoning	Zona residenziale/commerciale
YearBuilt	Anno di costruzione
YearRemodAdd	Anno dell'ultima ristrutturazione
HouseStyle	Stile architettonico (1Story, 2Story, ecc.)
OverallQual	Qualità complessiva dei materiali
OverallCond	Condizione generale dell'immobile

Caratteristiche del lotto:

Feature	Descrizione
LotArea	Superficie totale del lotto
LotShape	Forma del lotto (regolare/irregolare)
LandContour	Conformazione del terreno
LotConfig	Configurazione del lotto (interno, angolare, ecc.)
Neighborhood	Quartiere di appartenenza
LandSlope	Pendenza del terreno

Caratteristiche esterne:

Feature	Descrizione
Exterior1st	Materiale esterno principale
Exterior2nd	Materiale esterno secondario
ExterQual	Qualità dei materiali esterni
ExterCond	Condizione delle superfici esterne
RoofStyle	Tipologia del tetto
RoofMatl	Materiale del tetto

Caratteristiche interne:

Feature	Descrizione
GrLivArea	Superficie abitabile sopra il livello del suolo
BedroomAbvGr	Numero di camere da letto
TotRmsAbvGrd	Numero totale di stanze
KitchenQual	Qualità della cucina
HeatingQC	Qualità del sistema di riscaldamento
FullBath	Numero di bagni completi
HalfBath	Numero di mezzi bagni

Seminterrato:

Feature	Descrizione
BsmtQual	Qualità del seminterrato
BsmtCond	Condizione del seminterrato
TotalBsmtSF	Superficie totale del seminterrato
BsmtFinType1	Tipo di finitura principale
BsmtFinSF1	Superficie finita principale
BsmtFinType2	Tipo di finitura secondaria
BsmtFinSF2	Superficie finita secondaria

Garage:

Feature	Descrizione
GarageType	Tipologia di garage
GarageYrBlt	Anno di costruzione del garage
GarageFinish	Finitura interna del garage
GarageCars	Numero di posti auto
GarageArea	Superficie del garage
GarageQual	Qualità del garage
GarageCond	Condizione del garage

Caratteristiche Aggiuntive:

Feature	Descrizione
Fireplaces	Numero di camini
FireplaceQu	Qualità del camino
PoolArea	Superficie della piscina
PoolQC	Qualità della piscina
Fence	Tipo di recinzione
ScreenPorch	Superficie del portico schermato
OpenPorchSF	Superficie del portico aperto
EnclosedPorch	Superficie del portico chiuso

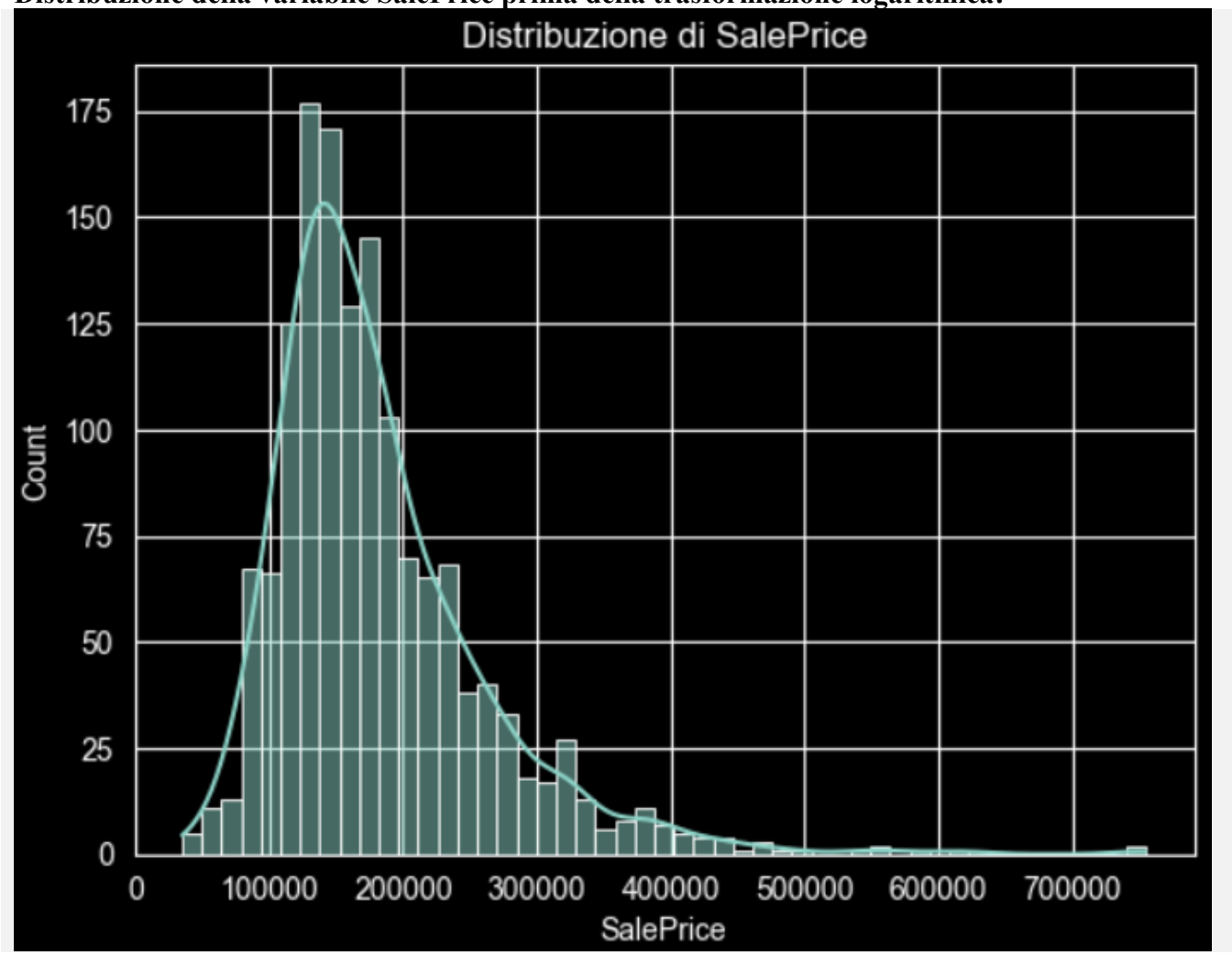
Variabile Target:

Feature	Descrizione
SalePrice	Prezzo finale di vendita dell'immobile

2.3 Analisi esplorativa iniziale

L'analisi esplorativa del dataset ha rappresentato un passaggio fondamentale per comprendere la natura dei dati e le dinamiche che legano le diverse variabili al prezzo di vendita degli immobili.
Fin dalle prime osservazioni è emerso che la variabile target SalePrice presenta una distribuzione fortemente asimmetrica: i valori più elevati sono meno frequenti e generano una coda lunga verso destra. Questa caratteristica è chiaramente visibile nell'istogramma riportato in Figura X e suggerisce l'opportunità di applicare una trasformazione logaritmica, utile per stabilizzare la varianza e rendere più lineare la relazione tra le feature e il prezzo finale.
Esaminando le variabili numeriche, alcune di esse hanno mostrato una relazione particolarmente forte con il prezzo di vendita. OverallQual, che sintetizza la qualità complessiva dell'immobile, e GrLivArea, che rappresenta la superficie abitabile, risultano tra i predittori più influenti. Anche variabili come GarageCars e TotalBsmtSF mostrano una correlazione significativa con SalePrice, confermando l'importanza degli spazi accessori e delle superfici complessive.
L'analisi ha inoltre evidenziato la presenza di valori anomali, soprattutto nelle superfici abitative e nei prezzi. Questi outlier, se non trattati correttamente, possono influenzare in modo significativo il processo di apprendimento del modello. Infine, le variabili categoriche si sono rivelate numerose e spesso caratterizzate da molte modalità, rendendo necessario un lavoro accurato di codifica per evitare un'eccessiva espansione dello spazio delle feature e garantire una rappresentazione coerente dei dati.

Distribuzione della variabile SalePrice prima della trasformazione logaritmica:



2.4 Identificazione dei problemi nei dati

L'esplorazione preliminare ha permesso di individuare diverse criticità che richiedono un intervento mirato nella fase di preparazione dei dati.

Una delle problematiche più evidenti riguarda la presenza di valori mancanti. Molte variabili relative al seminterrato, al garage, alla piscina o alla recinzione presentano numerosi valori nulli. In molti casi, tuttavia, l'assenza di un valore non rappresenta un errore, ma indica semplicemente che l'abitazione non dispone di quella specifica struttura. È quindi necessario distinguere tra valori mancanti informativi e valori mancanti da imputare.

Un'altra criticità riguarda la presenza di variabili ridondanti o fortemente correlate tra loro. Alcuni feature descrivono lo stesso concetto da prospettive diverse, come nel caso di GarageCars e GarageArea, oppure di TotRmsAbvGrd e GrLivArea. Una valutazione attenta di queste relazioni è essenziale per evitare problemi di multicollinearità che potrebbero compromettere la stabilità del modello.

La presenza di outlier rappresenta un ulteriore elemento di complessità. Valori estremamente elevati nelle superfici o nei prezzi possono distorcere il comportamento del modello, soprattutto nelle fasi iniziali del training.
Infine, molte variabili qualitative presentano un ordine intrinseco; tuttavia, nel presente progetto si è scelto di adottare un encoding uniforme basato su one-hot encoding, così da garantire semplicità, coerenza e compatibilità con l'intera pipeline.

2.5 Considerazioni preliminari

Dall'analisi svolta emerge un quadro chiaro: il dataset Ames Housing è ricco, dettagliato e potenzialmente molto informativo, ma richiede una fase di preparazione accurata per poter essere utilizzato in modo efficace. La varietà delle variabili, la presenza di valori mancanti, la distinzione tra categorie nominali e variabili con ordine intrinseco e la gestione degli outlier rendono indispensabile un approccio metodico e ben strutturato. Nel presente progetto, tuttavia, si è scelto di adottare un encoding uniforme basato su one-hot encoding, così da garantire semplicità, coerenza e compatibilità con l'intera pipeline.
La natura non lineare delle relazioni tra le feature e il prezzo di vendita suggerisce l'utilizzo di modelli avanzati, come XGBoost, in grado di catturare pattern complessi senza richiedere trasformazioni eccessivamente invasive. Allo stesso tempo, la costruzione di una pipeline modulare permette di applicare trasformazioni specifiche a ciascun tipo di variabile, garantendo coerenza, riproducibilità e facilità di manutenzione.
Queste considerazioni costituiscono la base per la fase successiva, dedicata alla preparazione dei dati, in cui il dataset verrà trasformato in una forma adeguata al training del modello e ottimizzata per ottenere prestazioni affidabili.

3. Data Preparation

La fase di preparazione dei dati costituisce uno dei momenti più importanti dell'intero progetto, poiché determina la qualità del dataset che verrà utilizzato per l'addestramento dei modelli di regressione. Sebbene Ames Housing sia un dataset ampiamente studiato e generalmente ben strutturato, la sua complessità richiede un lavoro approfondito di pulizia, trasformazione e codifica. Le variabili presenti sono numerose e molto eterogenee: comprendono misure numeriche continue, variabili discrete, categorie nominali e categorie ordinali, ciascuna delle quali necessita di un trattamento specifico.
L'obiettivo di questa fase è ottenere una rappresentazione dei dati coerente, priva di ambiguità e adeguata all'addestramento di diversi modelli di regressione, tra cui Linear Regression, Random Forest e XGBoost, che verranno successivamente confrontati per individuare la soluzione più efficace.

3.1 Pulizia dei dati

La prima operazione ha riguardato la verifica dell'integrità del dataset. È stato necessario controllare la presenza di duplicati, valori incoerenti o variabili prive di utilità predittiva. Il dataset Ames Housing non presenta duplicati, ma include alcune colonne che, pur essendo formalmente corrette, risultano ridondanti o scarsamente informative.
Una revisione preliminare ha permesso di individuare variabili che descrivono lo stesso concetto da prospettive diverse o che presentano una variabilità troppo limitata per essere utili al modello. La rimozione di queste feature ha contribuito a semplificare la struttura del dataset, riducendo il rischio di rumore e migliorando la leggibilità complessiva dei dati.
Questa fase ha rappresentato un primo passo verso la costruzione di un dataset più pulito e coerente, pronto per essere trasformato nelle fasi successive.

3.2 Gestione dei valori mancanti

Una parte significativa del lavoro è stata dedicata alla gestione dei valori mancanti, particolarmente numerosi in alcune categorie del dataset. La presenza di valori nulli non è uniforme: in molti casi, l'assenza di un valore non rappresenta un errore, ma un'informazione vera e propria. Ad esempio, se un'abitazione non dispone di un garage o di un seminterrato, le variabili corrispondenti risultano naturalmente vuote.
È stato quindi necessario distinguere tra valori mancanti informativi e valori mancanti dovuti a incompletezza del dato. Nel primo caso, la mancanza è stata trattata come una categoria specifica, così da preservare l'informazione implicita. Nel secondo caso, si è proceduto con tecniche di imputazione appropriate: per le variabili numeriche è stata utilizzata la mediana, mentre per le variabili categoriche è stata adottata la modalità più frequente.
Questo approccio ha permesso di mantenere la coerenza interna del dataset, evitando di introdurre distorsioni che avrebbero potuto influenzare negativamente le prestazioni dei modelli.

3.3 Encoding delle variabili categoriche

La presenza di numerose variabili categoriche ha richiesto un processo di codifica volto a trasformare le categorie in valori numerici interpretabili dai modelli di regressione.
Nel presente progetto è stato adottato un approccio uniforme basato sul one-hot encoding, applicato tramite la funzione di Pandas.
Questo metodo converte ciascuna modalità di una variabile categorica in una colonna binaria distinta, evitando di introdurre relazioni ordinali artificiali tra categorie che non possiedono un ordine naturale.
Sebbene alcune variabili presentino un ordine intrinseco (come i livelli qualitativi), si è scelto di utilizzare un encoding non ordinale per garantire semplicità, coerenza e compatibilità con tutti i modelli considerati. Il risultato è un dataset numerico completo, in cui tutte le variabili categoriche sono state trasformate in modo coerente e direttamente utilizzabile dai modelli di regressione.

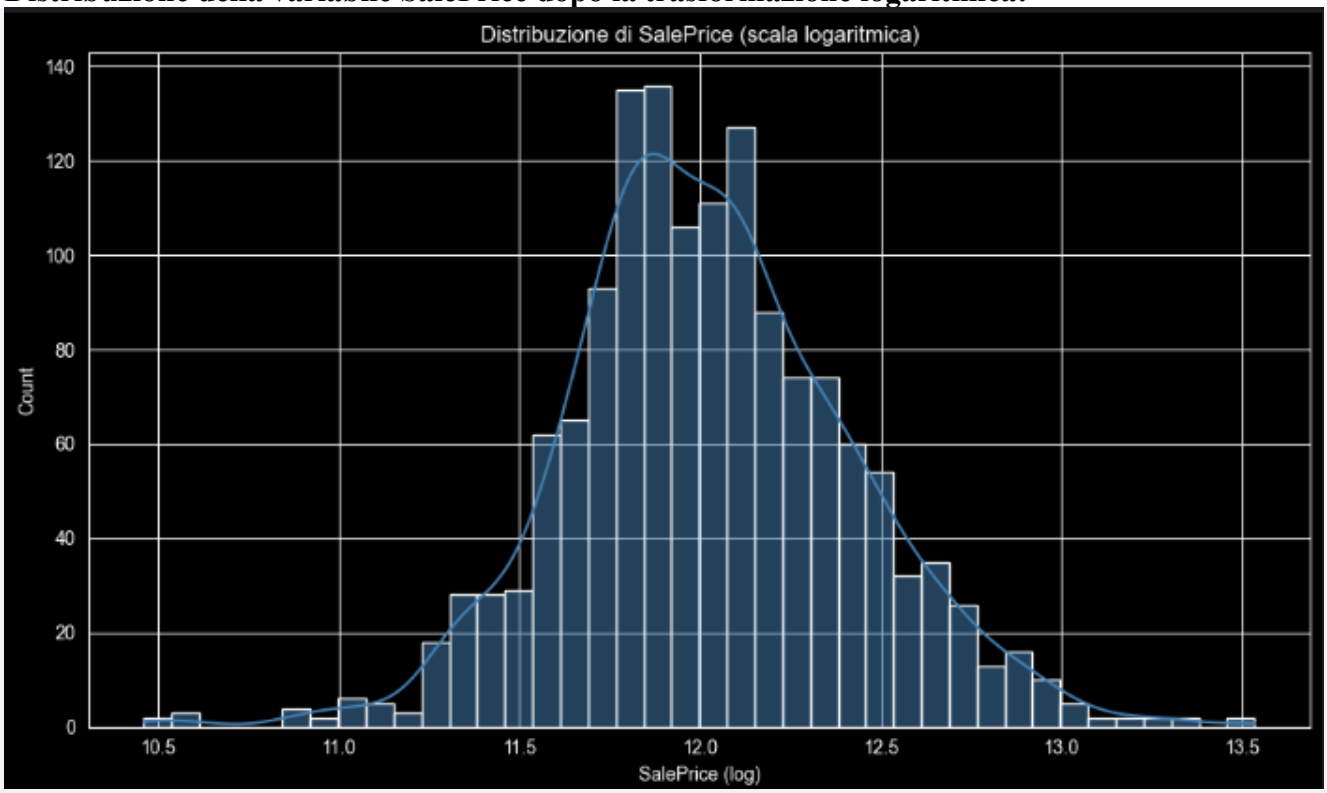
3.4 Feature Engineering

Parallelamente alle operazioni di pulizia e codifica, è stato svolto un lavoro di feature engineering finalizzato a migliorare la capacità predittiva dei modelli.
Alcune variabili sono state combinate per ottenere nuove informazioni più significative. Ad esempio, la superficie totale dell'immobile è stata calcolata combinando diverse aree interne ed esterne, mentre l'età effettiva dell'abitazione è stata ricavata confrontando l'anno di costruzione con l'anno dell'ultima ristrutturazione.
Queste trasformazioni hanno permesso di catturare relazioni che non erano immediatamente evidenti nel dataset originale, fornendo ai modelli informazioni più ricche e strutturate. Il feature engineering ha quindi contribuito a migliorare la qualità del dataset, rendendolo più rappresentativo delle caratteristiche reali degli immobili.

3.5 Normalizzazione e scaling

Sebbene alcuni modelli, come XGBoost e Random Forest, non richiedano una normalizzazione esplicita delle variabili numeriche, è stato comunque utile analizzare la distribuzione delle feature più influenti. In particolare, la variabile target SalePrice mostrava una forte asimmetria, con una coda lunga verso destra che evidenziava la presenza di valori molto elevati e potenzialmente influenti sul processo di apprendimento.
Per questo motivo è stata applicata una trasformazione logaritmica al prezzo di vendita, con l'obiettivo di stabilizzare la varianza e rendere più lineare la relazione tra le feature e il valore finale dell'immobile. La trasformazione ha prodotto una distribuzione più simmetrica e regolare, come mostrato in figura, migliorando la stabilità dei modelli e riducendo l'impatto degli outlier più estremi.
La normalizzazione delle altre variabili numeriche non è stata applicata in modo generalizzato, ma valutata caso per caso in base alle esigenze dei modelli utilizzati. Poiché gli algoritmi basati su alberi non sono sensibili alla scala delle feature, si è preferito evitare trasformazioni non necessarie, mantenendo il dataset il più fedele possibile alla sua struttura originale.

Distribuzione della variabile SalePrice dopo la trasformazione logaritmica:



3.6 Gestione degli outlier

Durante l'analisi esplorativa sono stati individuati alcuni valori estremi nelle variabili relative alle superfici e al prezzo di vendita. Questi outlier possono potenzialmente influenzare il comportamento dei modelli di regressione, soprattutto nelle fasi iniziali del training.

Nel presente progetto, tuttavia, non è stata effettuata una rimozione esplicita degli outlier tramite tecniche statistiche come IQR, z-score o filtri basati su soglie predefinite.

La gestione dei valori estremi è stata affrontata in modo implicito, attraverso due strategie principali:

- Trasformazione logaritmica della variabile target (**SalePrice**), che riduce l'impatto dei valori molto elevati e stabilizza la distribuzione.
- Utilizzo di modelli basati su alberi (Random Forest e **XGBoost**), noti per la loro robustezza nei confronti degli outlier e per la capacità di ridurre l'influenza dei casi rari durante la fase di apprendimento.

Questo approccio consente di preservare l'informazione contenuta nei valori estremi senza alterare artificialmente la distribuzione del dataset, mantenendo al tempo stesso stabilità e affidabilità nelle predizioni.

3.7 Risultato finale del preprocessing

Al termine della fase di preparazione, il dataset presenta una gestione coerente dei principali valori mancanti, ottenuta tramite imputazione o assegnazione di categorie specifiche nei casi in cui l'assenza rappresenta un'informazione strutturale (ad esempio, l'assenza di garage o seminterrato).

Tutte le variabili categoriche sono state convertite in forma numerica tramite one-hot encoding, mentre le variabili numeriche sono state armonizzate attraverso trasformazioni mirate, come la log-trasformazione della variabile target SalePrice. Il dataset risultante è quindi pulito, coerente e completamente numerico, con una struttura ottimizzata per l'addestramento dei modelli di regressione. Questa preparazione accurata costituisce la base per ottenere prestazioni affidabili e garantire la riproducibilità dell'intero progetto.

3.8 Pipeline automatizzata

Per garantire coerenza, riproducibilità e manutenibilità, tutte le operazioni di preprocessing sono state implementate all'interno di una pipeline modulare in Python.
Ogni trasformazione (dalla gestione dei valori mancanti all'encoding, dalla creazione di nuove feature alla gestione degli outlier) è stata incapsulata in funzioni dedicate, organizzate in modo da poter essere applicate in sequenza.
Questo approccio permette di mantenere il codice ordinato e facilmente estendibile, oltre a garantire che le stesse trasformazioni vengano applicate sia ai dati di training sia ai nuovi dati in fase di predizione. La pipeline rappresenta quindi un elemento fondamentale per assicurare coerenza tra le diverse fasi del progetto e per facilitare eventuali estensioni future.

4. Sviluppo del Modello

La fase di sviluppo del modello rappresenta il cuore del progetto PriceMyHouse, poiché consente di trasformare il dataset preparato nelle fasi precedenti in uno strumento predittivo capace di stimare il prezzo di un immobile con un buon livello di accuratezza. L'approccio adottato non si è limitato all'utilizzo di un singolo algoritmo, ma ha previsto la sperimentazione di diversi modelli di regressione, ciascuno con caratteristiche e punti di forza differenti.

Questa scelta ha permesso di confrontare le prestazioni dei vari metodi e di individuare la soluzione più adatta alla natura dei dati. In particolare, sono stati considerati tre modelli: Linear Regression, Random Forest Regressor e XGBoost Regressor. Ognuno di essi è stato addestrato sul dataset preprocessato e valutato attraverso metriche standard, con l'obiettivo di comprendere quale modello fosse in grado di catturare meglio le relazioni complesse tra le feature e il prezzo di vendita.

4.1 Linear Regression

La regressione lineare ha rappresentato il punto di partenza dell'analisi modellistica. Si tratta di un modello semplice, interpretabile e ampiamente utilizzato come baseline nei problemi di regressione. La sua forza risiede nella capacità di fornire una prima stima delle relazioni tra le variabili indipendenti e la variabile target, permettendo di valutare rapidamente la qualità del dataset e l'eventuale presenza di pattern lineari. Tuttavia, la regressione lineare presenta alcune limitazioni quando applicata a dataset complessi come Ames Housing.
Le relazioni tra le feature e il prezzo di vendita non sono sempre lineari, e la presenza di variabili categoriche codificate tramite one-hot encoding può generare un numero elevato di coefficienti difficili da interpretare. Inoltre, il modello è sensibile alla multicollinearità e agli outlier, che possono influenzare significativamente i risultati. Nonostante ciò, la regressione lineare ha svolto un ruolo fondamentale come modello di riferimento, fornendo un primo benchmark utile per confrontare le prestazioni dei modelli più avanzati.

4.2 Random Forest Regressor

Il secondo modello considerato è stato il Random Forest Regressor, un algoritmo basato su un insieme di alberi decisionali addestrati su campioni differenti del dataset. Questo approccio consente di ridurre la varianza tipica dei singoli alberi e di ottenere un modello più stabile e robusto. Random Forest è particolarmente adatto ai dataset tabellari e si comporta bene in presenza di variabili eterogenee, senza richiedere trasformazioni complesse o scaling delle feature. Inoltre, è in grado di catturare relazioni non lineari e interazioni tra variabili che la regressione lineare non riesce a modellare.
Nel contesto del progetto, Random Forest ha fornito risultati migliori rispetto alla regressione lineare, mostrando una maggiore capacità di adattarsi alla complessità del dataset. Tuttavia, pur essendo un modello potente, presenta alcune limitazioni: tende a essere meno efficiente rispetto a modelli più avanzati e può risultare meno preciso quando le relazioni tra le feature sono particolarmente intricate.

4.3 XGBoost Regressor

Il modello che ha fornito le migliori prestazioni è stato XGBoost, un algoritmo basato sul boosting di alberi decisionali. A differenza del Random Forest, che costruisce gli alberi in parallelo, XGBoost li costruisce in sequenza, correggendo progressivamente gli errori commessi dagli alberi precedenti.
Questa strategia permette al modello di catturare pattern complessi e di adattarsi in modo molto efficace ai dati tabellari, soprattutto quando sono presenti relazioni non lineari e interazioni tra variabili. XGBoost offre inoltre numerosi parametri di ottimizzazione che consentono di controllare la complessità del modello, prevenire l'overfitting e migliorare la generalizzazione.
Nel progetto PriceMyHouse, XGBoost si è dimostrato il modello più performante, ottenendo un errore di predizione inferiore rispetto agli altri algoritmi testati. La sua capacità di gestire in modo efficiente sia variabili numeriche sia categoriche codificate, unita alla robustezza nei confronti degli outlier, lo ha reso la scelta ideale per il modello finale.

4.4 Confronto delle performance

Per valutare le prestazioni dei modelli è stata utilizzata la metrica Root Mean Squared Error (RMSE), particolarmente adatta ai problemi di regressione in cui si desidera penalizzare maggiormente gli errori più grandi. La regressione lineare ha fornito un risultato accettabile, ma inferiore rispetto ai modelli basati su alberi. Random Forest ha mostrato un miglioramento significativo, grazie alla sua capacità di catturare relazioni non lineari e interazioni tra variabili.
È stato inoltre testato un modello Gradient Boosting Regressor, che ha ottenuto prestazioni intermedie rispetto a Random Forest e XGBoost, confermando la maggiore efficacia degli approcci basati su boosting sequenziale.
Tra tutti i modelli considerati, XGBoost è quello che ha raggiunto l'RMSE più basso, dimostrando la migliore capacità di adattarsi alla complessità del dataset e di fornire predizioni più stabili e accurate. Il confronto complessivo conferma quindi XGBoost come la soluzione più efficace per il problema di regressione affrontato.

4.5 Scelta del modello finale

Sulla base dei risultati ottenuti, XGBoost è stato selezionato come modello finale del progetto. La sua capacità di adattarsi alla complessità del dataset, unita alle prestazioni superiori rispetto agli altri algoritmi testati, lo rende la scelta più adeguata al problema affrontato.
Il modello finale è stato addestrato utilizzando i parametri ottimizzati e successivamente salvato in formato serializzato, così da poter essere riutilizzato per effettuare predizioni su nuovi dati senza dover ripetere l'intero processo di addestramento.

4.6 Implementazione modulare

Tutte le operazioni relative allo sviluppo dei modelli sono state implementate in modo modulare, seguendo le migliori pratiche di ingegneria del software. Ogni modello è stato gestito all'interno di

file Python dedicati, con funzioni specifiche per l'addestramento, la valutazione e il salvataggio. Questa struttura ha permesso di mantenere il codice ordinato, facilmente leggibile e semplice da estendere. Inoltre, la modularità facilita l'integrazione di nuovi modelli o l'aggiornamento di quelli esistenti, rendendo il progetto flessibile e adatto a sviluppi futuri.

5. Training e Valutazione

La fase di training e valutazione rappresenta il momento in cui il lavoro svolto nella preparazione dei dati e nello sviluppo dei modelli viene messo alla prova. L'obiettivo è verificare la capacità dei modelli di apprendere le relazioni tra le feature e il prezzo di vendita degli immobili e di generalizzare correttamente su dati mai visti prima. Per garantire una valutazione affidabile, il dataset preprocessato è stato suddiviso in due parti: una porzione destinata all'addestramento (80%) e una riservata al test finale (20%). Questa separazione consente di misurare le prestazioni dei modelli su dati indipendenti, evitando che la valutazione venga influenzata da fenomeni di overfitting.

5.1 Metrica di valutazione

Per confrontare i modelli è stata scelta la metrica Root Mean Squared Error (RMSE), una delle più utilizzate nei problemi di regressione.
L'RMSE misura la differenza media tra i valori predetti e quelli reali, penalizzando maggiormente gli errori più grandi. Questa caratteristica la rende particolarmente adatta al contesto immobiliare, dove errori elevati possono avere un impatto significativo sulla qualità della stima. In aggiunta, è stato calcolato anche il coefficiente di determinazione R^2, utile per valutare la proporzione di variabilità del target spiegata dal modello.

5.2 Addestramento dei modelli

L'addestramento dei modelli è stato condotto applicando a ciascun algoritmo la stessa suddivisione dei dati e le stesse trasformazioni ottenute nella fase di preprocessing, inclusa la trasformazione logaritmica della variabile target.
Sono stati considerati tre modelli:

- Linear Regression, utilizzata come baseline semplice e interpretabile
- Random Forest Regressor, modello basato su bagging e alberi decisionali
- XGBoost Regressor, modello basato su boosting, noto per le sue ottime prestazioni su dati tabellari

Ogni modello è stato addestrato sui dati di training e valutato esclusivamente sul test set, così da ottenere una stima realistica della capacità di generalizzazione.

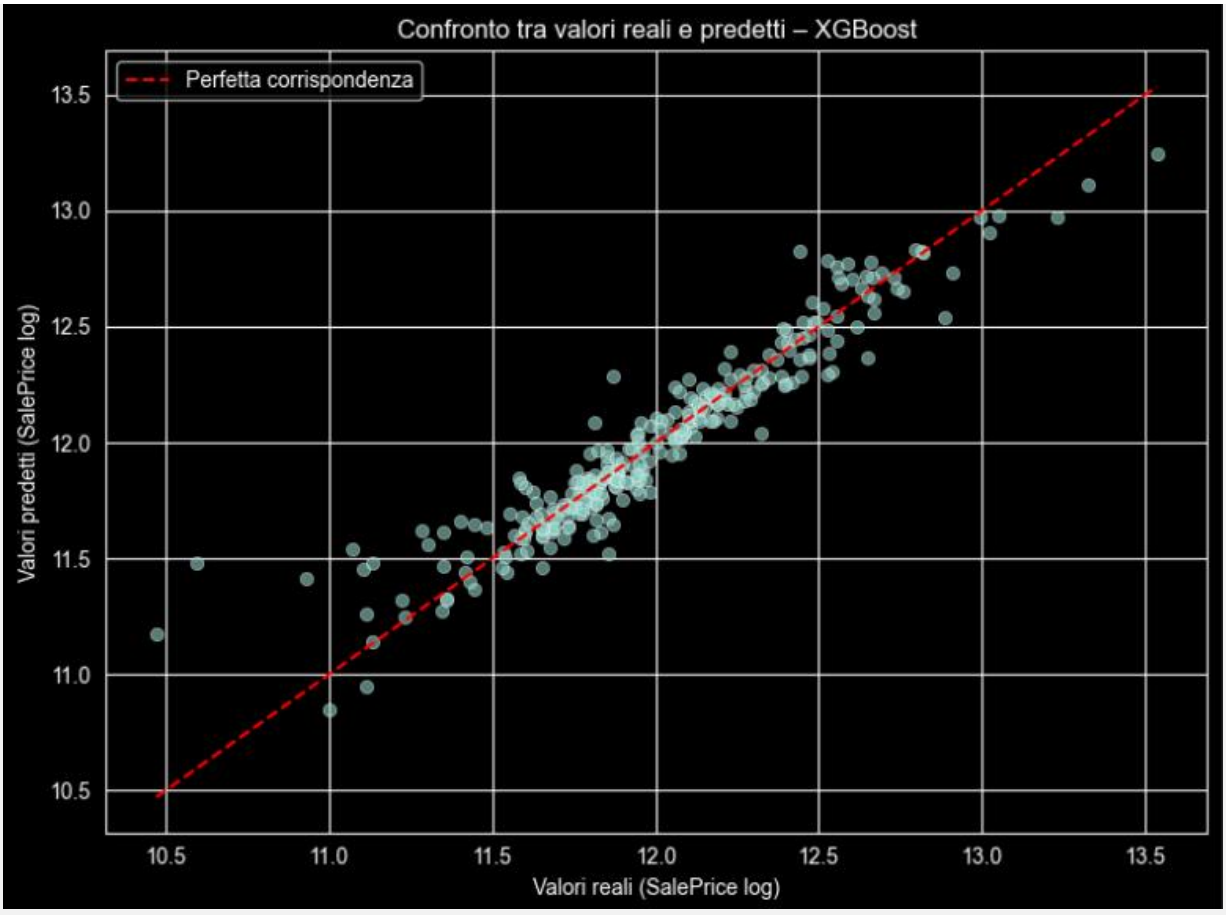
5.3 Risultati dei modelli

I risultati ottenuti evidenziano differenze significative tra i modelli considerati. La regressione lineare, pur essendo semplice e veloce da addestrare, mostra una capacità limitata nel catturare la complessità del dataset e le relazioni non lineari tra le variabili. Random Forest migliora sensibilmente le prestazioni, grazie alla sua capacità di modellare interazioni più articolate e di gestire in modo robusto la variabilità del dataset. È tuttavia XGBoost a ottenere i risultati migliori, confermando la sua efficacia nei problemi di regressione complessi e la sua capacità di adattarsi a pattern non lineari. Le metriche calcolate sul test set mostrano chiaramente il vantaggio del boosting rispetto agli altri algoritmi. Di seguito sono riportate le metriche ottenute sul test set:

Modello	RMSE	R^2
XGBoost	0.1364	0.9004
Random Forest	0.1496	0.8801
Linear Regression	0.2105	0.7626

XGBoost risulta quindi il modello più accurato, con un RMSE inferiore e un R^2 superiore rispetto agli altri algoritmi. L'analisi numerica è stata ulteriormente supportata da un grafico di confronto tra valori reali e predetti (in figura), che mostra una forte concentrazione dei punti attorno alla diagonale ideale, confermando la buona capacità del modello di approssimare correttamente la variabile target.

Confronto tra valori reali e predetti dal modello XGBoost:



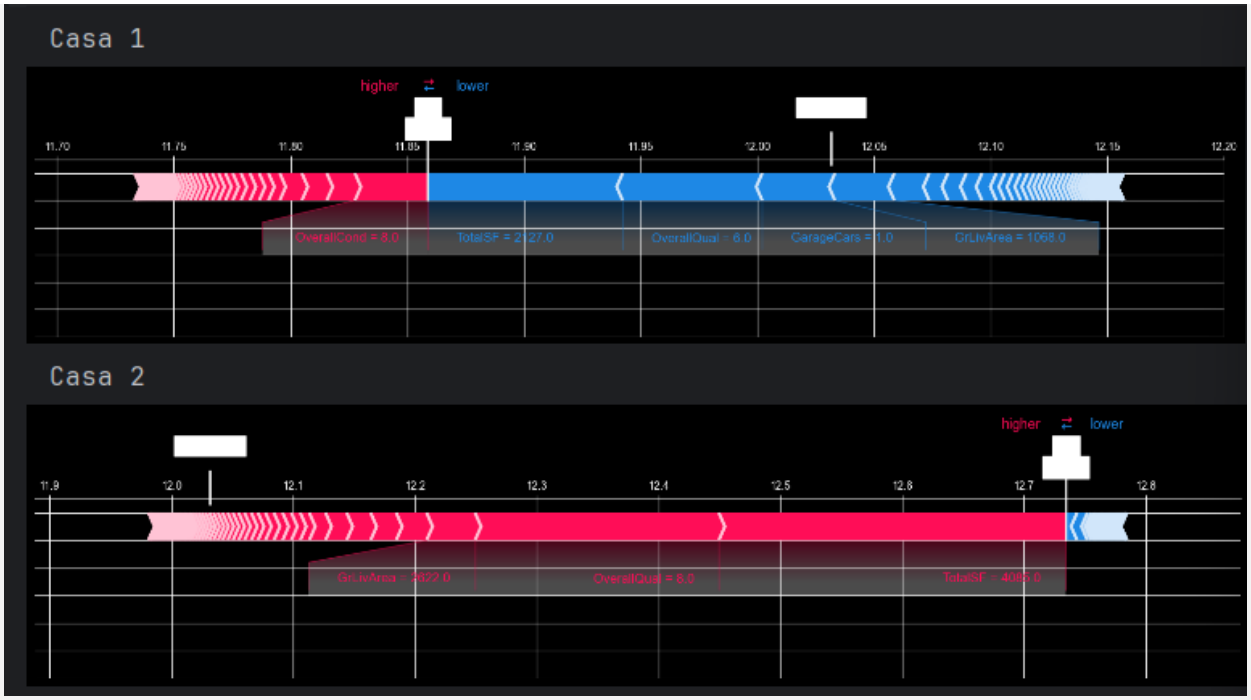
5.4 Analisi degli errori

L'analisi degli errori ha permesso di comprendere meglio i punti di forza e le limitazioni dei modelli.
Linear Regression tende a sottostimare gli immobili di valore elevato, a causa della sua natura lineare che non riesce a catturare adeguatamente le variazioni più marcate.
Random Forest mostra una leggera tendenza a sovrastimare alcune tipologie di abitazioni, probabilmente a causa della presenza di outlier che influenzano la costruzione degli alberi.
XGBoost presenta errori distribuiti in modo più uniforme e una maggiore capacità di adattarsi alle caratteristiche specifiche degli immobili, confermando la sua robustezza.

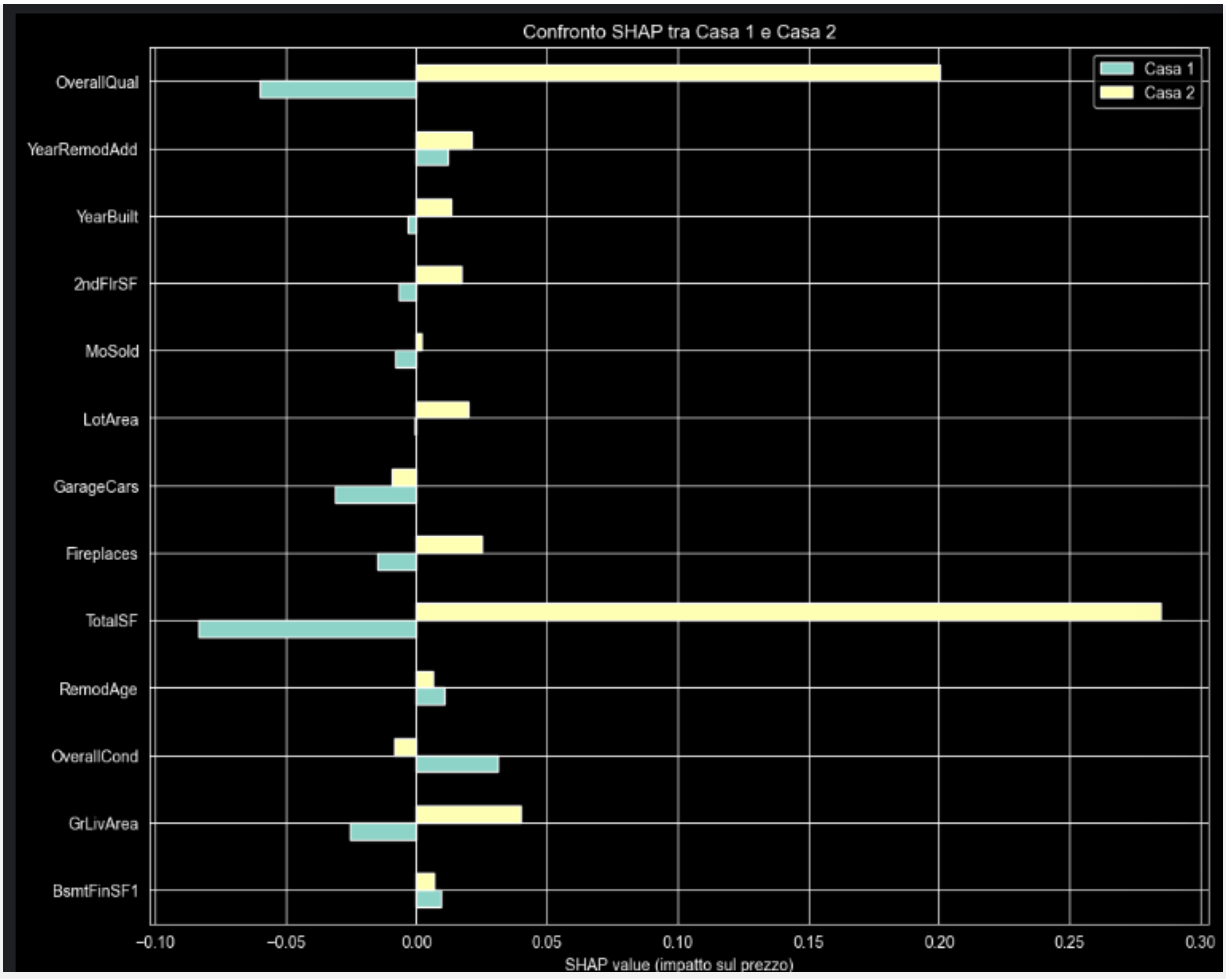
5.5 Interpretazione dei risultati

L'interpretazione dei risultati ha permesso di comprendere in modo più approfondito quali variabili influenzino maggiormente il prezzo di vendita degli immobili e in che modo i modelli, in particolare XGBoost, utilizzino tali informazioni per generare le predizioni.
A livello globale, la Feature Importance ha evidenziato il ruolo centrale di alcune variabili chiave, tra cui la qualità complessiva dell'immobile (OverallQual), la superficie abitabile (GrLivArea) e il numero di posti auto nel garage (GarageCars). Queste feature mostrano un impatto significativo sul prezzo finale, confermando le osservazioni emerse durante l'analisi esplorativa.
La capacità dei modelli basati su alberi di cogliere interazioni non lineari consente inoltre di valorizzare variabili come TotalSF e YearBuilt, che contribuiscono in modo rilevante alla predizione.
Per approfondire ulteriormente il comportamento del modello, sono stati analizzati i valori SHAP, che permettono di spiegare il contributo di ciascuna feature alla predizione per singoli esempi. L'analisi locale (Figura 1) mostra come alcune variabili possano spingere la stima del prezzo verso l'alto o verso il basso, offrendo una lettura trasparente e interpretabile delle decisioni del modello.
Infine, il confronto tra due abitazioni differenti (Figura 2) evidenzia come lo stesso insieme di feature possa avere impatti diversi a seconda delle caratteristiche specifiche dell'immobile.
Ad esempio, superfici molto ampie o una qualità costruttiva superiore possono determinare incrementi significativi nella predizione, mentre valori più bassi in variabili strutturali tendono a ridurre il prezzo stimato. Questo tipo di analisi comparativa consente di comprendere non solo quali feature siano importanti, ma anche come e perché influenzino la predizione in casi concreti.

1. **Valori SHAP per una singola abitazione:**



2. **Confronto SHAP tra due abitazioni:**



5.6 Considerazioni sulle performance

Le prestazioni ottenute dimostrano che XGBoost rappresenta la soluzione più efficace per il problema affrontato. La sua capacità di gestire dati eterogenei, catturare relazioni non lineari e adattarsi alla complessità del dataset lo rende particolarmente adatto al contesto immobiliare.
Il confronto con gli altri modelli ha evidenziato:

- un miglioramento di circa il 35% rispetto alla regressione lineare
- un miglioramento di circa il 9% rispetto a Random Forest
- un R^2 superiore al 90%, indice di un'elevata capacità esplicativa

La pipeline di preprocessing ha contribuito in modo significativo al miglioramento delle prestazioni, garantendo un dataset pulito, coerente e ben strutturato. Nel complesso, i risultati ottenuti confermano la validità dell'approccio adottato e forniscono una base solida per gli sviluppi futuri del progetto.

6. Conclusioni e Sviluppi Futuri

6.1 Conclusioni

Il progetto PriceMyHouse ha dimostrato come l'applicazione di tecniche di Machine Learning al settore immobiliare possa fornire strumenti predittivi accurati, affidabili e potenzialmente utilizzabili in contesti reali. A partire dall'Ames Housing Dataset, è stata sviluppata una pipeline completa che copre tutte le fasi fondamentali di un progetto di Data Science:

- comprensione e analisi esplorativa dei dati
- pulizia, trasformazione e codifica delle variabili
- feature engineering mirato
- gestione degli outlier e dei valori mancanti
- sviluppo, addestramento e valutazione di diversi modelli di regressione

L'approccio adottato ha permesso di ottenere un dataset coerente, con una gestione accurata dei principali valori mancanti, e strutturato in modo ottimale per l'addestramento dei modelli.
Il confronto tra Linear Regression, Random Forest e XGBoost ha evidenziato differenze significative nelle capacità predittive dei tre algoritmi. I risultati ottenuti sul test set mostrano chiaramente la superiorità di XGBoost, che ha raggiunto un RMSE pari a 0.1364 e un R^2 pari a 0.9004, dimostrando un'elevata capacità di catturare le relazioni non lineari e le interazioni complesse tra le feature.
Questi risultati confermano la validità dell'approccio basato sul boosting e mostrano come modelli avanzati possano fornire stime molto accurate del prezzo di vendita degli immobili, anche in presenza di dati eterogenei e strutturalmente complessi.
Nel complesso, PriceMyHouse rappresenta un progetto completo, riproducibile e facilmente estendibile, che integra buone pratiche di ingegneria del software con tecniche moderne di Machine Learning. La pipeline modulare sviluppata consente inoltre di applicare lo stesso flusso a nuovi dataset, rendendo il progetto un'ottima base per applicazioni reali o per ulteriori approfondimenti accademici.

6.2 Sviluppi Futuri

Nonostante i risultati ottenuti siano molto soddisfacenti, il progetto offre numerose possibilità di estensione e miglioramento. Tra i principali sviluppi futuri possibili:

1. Ottimizzazione avanzata degli iperparametri: l'utilizzo di tecniche come Grid Search, Random Search o Bayesian Optimization potrebbe migliorare ulteriormente le prestazioni dei modelli, in particolare di XGBoost.
2. Introduzione di modelli alternativi: modelli come LightGBM e CatBoost, particolarmente efficaci sui dati tabellari, potrebbero essere testati e confrontati con XGBoost.
3. Gestione avanzata degli outlier: tecniche più sofisticate, come Isolation Forest o Local Outlier Factor, potrebbero migliorare ulteriormente la robustezza del dataset.
4. Estensione del dataset: l'integrazione di nuove fonti informative (ad esempio dati geografici, socio-economici, servizi nelle vicinanze) potrebbe aumentare la capacità predittiva del modello.
5. Deployment del modello: la creazione di un'API o di una web app permetterebbe di utilizzare il modello in un contesto reale, rendendo PriceMyHouse uno strumento accessibile anche a utenti non tecnici.

6.3 Considerazioni finali

Il progetto ha raggiunto pienamente gli obiettivi prefissati, dimostrando come un approccio metodico, supportato da una pipeline ben strutturata e da modelli avanzati, possa portare a risultati concreti e di elevata qualità. PriceMyHouse rappresenta un esempio efficace di applicazione del Machine Learning al settore immobiliare e costituisce una base solida per ulteriori sviluppi.