Relazione
Progetto di Data Web Mining 2022/23

Membri: Vettori Massimo, Masiero Stefano, Vego Scocco Thomas

Indice

1. Feature Analysis Engineering

Prima di partire con l'utilizzo dei modelli, abbiamo analizzato le features del dataset.
L'analisi delle features è stata fatta partendo da quelle numeriche, vedendo la loro distribuzione e i valori che assumevano.

1.1. Trasformazione e Rescaling Variabili Numeriche:

È stato visto che alcune features numeriche non erano continue ma bensì ordinali, inoltre, osservando il grafico, alcune sembravano seguire una distribuzione normale, ma asimmetrica: per sistemarle è stata eseguita una trasformazione logaritmica (questo anche per la variabile target "Sale_Price").
Dopo la trasformazione, abbiamo verificato il valore teorico dei quantili campionando alcune variabili trasformate ed è stato notato un miglioramento per la maggior parte di tali variabili.

Siccome molti algoritmi di M.L. operano molto bene se i valori delle features sono riscalati, è stato deciso di riscalare le features numeriche continue che abbiamo ritenuto ne avessero bisogno (è stato fatto un rescaling con StandardScaler per standardizzare i valori).

1.2. Outliers

L'analisi per gli outliers è stata fatta guardando principalmente i grafici di relazione tra la variabile target e tutte le altre features.
È stato preso principalmente in considerazione la relazione tra il prezzo di vendita e lo spazio della casa, in particolare la variabile "Gr_Liv_Area" che indica l'area vivibile della casa.
Da questa relazione sono stati individuati tre punti che ci sembravano dei punti di rumore e dunque sono stati eliminati; sono stati individuati altri due punti che potevano essere considerati potenziali punti di rumore, ma sono stati comunque tenuti perché rientravano nel cono di variabilità dei dati al crescere del valore del prezzo rispetto alla variabile in questione.

1.3. Trasformazione Variabili Categoriche

Dai grafici delle variabili categoriche e dai valori che assumevano, abbiamo notato che alcune features avevano una relazione d'ordine e quindi potevano essere trasformate in variabili numeriche ordinali.
Per le restanti features categoriche è stato optato l'utilizzo del classico approccio di One-Hot Encoding.

1.4. Clustering vs Neighborhood

La nostra idea iniziale era quella di scartare i reali quartieri della città di Ames e costruirli noi mediante algoritmi di clustering, per vedere se possa migliorare o meno la predizione dei modelli (il modello campione utilizzato è lo stesso per la selezione delle features, quindi anche tale modello è cambiato nel tempo).
Tra i vari algoritmi di clustering utilizzati, i due che performano meglio sono K-Means e Agglomerative Clustering. È stato visto che, in ogni caso, la precisione utilizzando il risultato del clustering è leggermente inferiore a quella con i quartieri originali, quindi è stato deciso di non tenere tale variabile.

1.5. Aggiunta di Nuove Variabili

Per aiutare i modelli nella predizione e dare a loro un po' più di informazioni, sono state aggiunte delle nuove variabili, tra cui "Total_SF" (Square Feet Totali della casa), "House_Age" (Età della casa) o "Total_Baths" (N° di bagni totali della casa).

Un'altra feature che è stata aggiunta è "Neighborhood_Median_Sale_Price": molto semplicemente, rappresenta la mediana dei prezzi, per ogni quartiere, di tutte le case vendute precedentemente rispetto a una certa casa.

1.6. Analisi della Correlazione

Per vedere come tutte le features, comprese quelle aggiunte, si relazionano con la variabile d'interesse "Sale_Price", è stata calcolata e visualizzata la matrice di correlazione con il metodo di Spearman perché tiene conto della relazione di monotonia (a differenza di Pearson che tiene conto della relazione lineare).
Per restringere il campo tra tutte le features, sono state filtrate le features in modo da tenere conto solo quelle più impattanti, ovvero quelle con una correlazione sopra la soglia (0.5).
Tra tutte le variabili, "Total_SF" e "Overall_Qual" sono quelle che si relazionano meglio con la variabile target.

1.7. Features a Bassissima Varianza

Siccome è stato notato un notevole numero di features quasi-costanti, ovvero le features che mostrano lo stesso valore per la maggior parte delle osservazioni nel set di dati, abbiamo proceduto con la selezione delle features con una varianza molto bassa per poi poterle eliminare.
La selezione è avvenuta verificando che, per ogni feature, la varianza superasse una certa soglia: tutte quelle che non hanno superato il threshold (0.1) sono state selezionate per poter essere successivamente rimosse.
La soglia è stata decisa analizzando la varianza delle features del dataset (dopo il One-Hot Encoding, l'aggiunta delle features e dopo la loro trasformazione e rescaling) e scegliendo il valore che sembrava andare bene per il dataset in questione.

1.8. Selezione delle Features

La selezione delle features è stata eseguita utilizzando un algoritmo di apprendimento automatico.
All'inizio ci siamo basati sul parere della Random Forest, ma dopo una successiva valutazione dei modelli è stato visto che, tra quelli che performano meglio, il Ridge Regressor era un ottimo candidato per il suo costo di computazione e la qualità della performance. Dunque è stato scelto tale modello per la selezione delle features, guadagnando principalmente sul tempo d'esecuzione e su una (successiva) leggera precisione generale dei vari modelli.
Siccome il numero di features scelte erano comunque troppe, sono state scelte le migliori 100 valutate da tale modello. Sono state valutate le performance dei modelli anche su 50 e 70 features, ma abbiamo ottenuto una precisione (in termini di R^2 e MSE) peggiore.

1.9. Test su un Dummy Model

Per concludere, abbiamo valutato l'operato su un modello "fantoccio": tale modello è stato costruito in modo che ritornasse la mediana dei prezzi del quartiere in cui si trova la casa di cui vogliamo predire il prezzo.
Ovviamente le aspettative non potevano essere alte, ma è stato notato che spiega poco più della metà della variabilità, il che, per un modello così semplice, è abbastanza buono.

2. Analisi e Valutazione dei Modelli

Dopo aver sistemato le features del dataset non rimane altro che utilizzare i modelli e vedere come performano e quali sono i migliori.

2.1. Splitting del Dataset e Valutazione dei Modelli

Durante la prima fase dell'analisi di un modello, solitamente viene suddiviso il dataset in test-set e train-set, la nostra scelta è stata quella di non utilizzare i metodi forniti da SkLearn a favore di uno split più "realistico".
È stata, pertanto, implementata una funzione che ci permette di suddividere le istanze basandosi sull'effettiva data di vendita di una casa, simulando una situazione reale, nella quale le case da predire sono quelle ancora non messe in vendita.
Questo ci permette di calcolare le nuove feature sull'intero dataset, senza rischiare di intaccare il test-set, in quanto ogni casa viene presa come elemento indipendente da quelle future, facendo sì che vengano calcolate solamente su quelle passate.

Seguendo la stessa linea di ragionamento abbiamo costruito un metodo di cross validation che usa le Time Series, in modo da splittare in train e validation set facendo sì che siano completamente indipendenti fra loro e che il train set abbia solamente dati passati, rispetto al validation set che contiene dati futuri.

Per la valutazione dei modelli abbiamo scelto alcune metriche che possono descrivere la loro accuratezza, fra le quali si possono vedere R^2, Explained Variance, RMSE e MAE (Mean Absolute Error). Per confrontare i modelli fra loro, sfruttiamo specialmente però l' R^2.

2.2. Modus Operandi per l'Analisi dei Modelli

È stato seguito uno schema molto semplice per l'analisi di ogni modello:

1. Viene fatta una ricerca dei migliori iperparametri da passare al modello utilizzando una GridSearch Cross-Validation e vengono salvati in un file per poterli riutilizzare in futuro;
2. Trovati i parametri migliori, viene creato il modello definitivo e validato utilizzando il Cross-Validation temporale spiegato nel paragrafo precedente;
3. Viene successivamente testato il modello sul test-set e vengono valutate le metriche, anche confrontandole con quelle ottenute dal CV temporale;
4. Come ulteriore analisi, vengono controllati i residui e la loro distribuzione per avere una visione grafica delle performance del modello.

2.3. Analisi dei Modelli

Sono stati utilizzati e analizzati molti modelli, tra cui alcuni di SkLearn (come la Random Forest, Regressione Lineare o Support Vector Machine) e altri esterni (come XGBoost, CatBoost, LightGBM o una ANN con Tensorflow).
Tra tutti i modelli, i tre migliori, ovvero quelli che operano molto bene nel test-set, sono SVM (89.5% R^2), CatBoost (89.8% R^2) e i modelli di Regressione Lineare (Ridge: 89.9% R^2, Lasso: 89.8% R^2, Elastic-Net: 89.9% R^2).

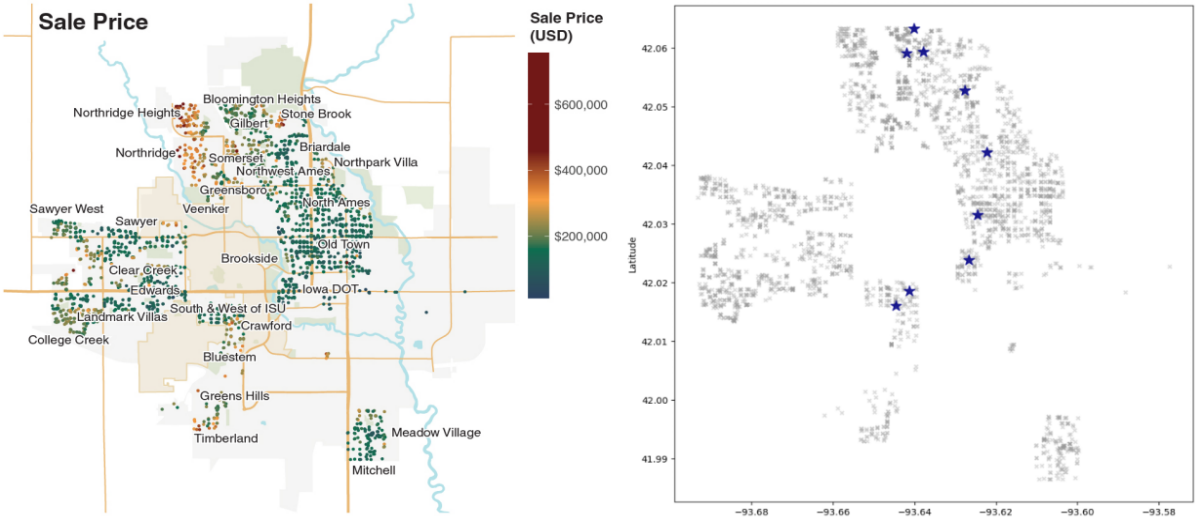
Mentre i tre peggiori sono DecisionTree (79.9% R^2), AdaBoost (83.6% R^2) e KNN (84.5% R^2).

2.4. Predizioni

Per le predizioni sono state prese le peggiori 10 predizioni per modello e 1 predizione migliore per ogni modello.

2.4.1 Migliori

Da quello che abbiamo potuto osservare dalle nostre analisi, abbiamo intuito che i punti dove sono state fatte le migliori previsioni sui prezzi delle case sono le zone nella mappa dove le case hanno bene o male prezzi simili.



2.4.2 Peggiori

Invece le previsioni peggiori sono state fatte sempre guardando la mappa dei prezzi nelle zone dove le case hanno bene o male tutte gli stessi prezzi ma ce ne sono alcune che invece si distinguono per il loro prezzo elevato.