UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO | DIPARTIMENTO DI INFORMATICA

CASO DI STUDIO

INGEGNERIA DELLA CONOSCENZA

AA 23/24

A Machine Learning Approach To AirBnB

Autore:

Nicola Visci

Matricola: 700797

E-mail: n.visci2@studenti.uniba.it

Docente:

Prof. Nicola Fanizzi

GitHub Repository

https://github.com/NicoVisci/ProgettoIcon/tree/master

Introduzione

Negli ultimi anni, il settore degli affitti a breve termine ha conosciuto una crescita significativa, trainata soprattutto da piattaforme come Airbnb, che consentono ai proprietari di affittare i propri spazi a turisti e viaggiatori. L'espansione di questo mercato, già di per sé dinamico, ha reso sempre più complessa la gestione delle proprietà e delle strategie di prezzo, poiché una varietà di fattori – tra cui posizione, caratteristiche dell'alloggio, stagionalità e feedback degli ospiti – possono influenzare la competitività di un annuncio.

In questo contesto, un approccio avanzato basato sul machine learning può offrire un vantaggio importante. Il presente progetto mira a sviluppare modelli di machine learning per esplorare le strutture sottostanti ai dati in possesso con l'obiettivo di apprendere le dinamiche e i pattern presenti. Sarà quindi possibile utilizzare i modelli ricavati per produrre previsioni, essenziali per massimizzare il profitto e soddisfare i clienti, e per derivare insight più approfonditi sul comportamento del mercato e sulle variabili chiave.

Attraverso l'utilizzo di algoritmi di apprendimento supervisionato e non supervisionato, il progetto si propone di investigare le variabili che maggiormente influenzano le scelte dei consumatori e le strategie di mercato dei proprietari. Questo approccio analitico è supportato da Python e dalle sue librerie per la gestione dei dati e la modellazione (Pandas, scikit-learn, ecc.), che rendono possibile la trasformazione, pulizia e analisi dei dati in modo accurato e flessibile.

In sintesi, il progetto mira a fornire strumenti informativi affinati che possono guidare proprietari e analisti nella comprensione del mercato delle locazioni a breve termine, contribuendo alla definizione di strategie informate e all'ottimizzazione delle offerte.

Obiettivi del Progetto

Il progetto si articola in una serie di obiettivi distinti, ciascuno mirato ad esplorare e apprendere diverse caratteristiche degli annunci Airbnb. Gli obiettivi principali includono:

1. **Analisi delle Caratteristiche per la Predizione del Prezzo:** Identificare le variabili chiave che influenzano il prezzo degli annunci Airbnb e sviluppare un modello di machine learning che possa prevedere accuratamente il costo di una proprietà. Questo obiettivo permette di comprendere meglio il valore economico di ciascuna soluzione abitativa in relazione alle sue specificità, permettendo a potenziali clienti di scernere tra annunci sovra prezzati ed occasioni, mentre i proprietari possono individuare chiaramente i propri margini di guadagno.
2. **Analisi delle Caratteristiche per la Predizione della Disponibilità:** Esaminare i fattori che contribuiscono alla disponibilità di un annuncio, con l'obiettivo di predire la frequenza con cui una proprietà è accessibile per nuove prenotazioni. La comprensione delle dinamiche di disponibilità è utile sia per i proprietari, che possono ottimizzare la gestione dell'immobile, sia per i clienti che cercano opzioni facilmente prenotabili.
3. **Analisi Non Supervisionata per il Raggruppamento degli Annunci:** Applicare tecniche di clustering per raggruppare gli annunci con caratteristiche simili, al fine di identificare pattern comuni e segmenti di mercato. Questa analisi non supervisionata permette di creare gruppi di proprietà con tratti omogenei, fornendo insight utili per individuare le preferenze degli utenti, formulare strategie di marketing mirate e per la personalizzazione delle offerte.
4. **Analisi Probabilistica per la Validazione del Raggruppamento:** Validare i gruppi formati attraverso un'analisi probabilistica che confermi la coerenza dei cluster identificati. Questo passaggio assicura che i gruppi siano statisticamente significativi e rappresentativi, supportando una segmentazione affidabile del mercato.

Strumenti Utilizzati

Per realizzare questo progetto, il linguaggio di programmazione principale scelto è Python, noto per la sua facilità d'uso e la vasta gamma di librerie disponibili per l'analisi dei dati e il machine learning, rendendolo una scelta ideale per questo studio. Tra le librerie di Python, sono state scelte le seguenti per via delle loro funzionalità:

1. **Pandas**
   La libreria Pandas è stata utilizzata per la gestione e la manipolazione dei dati. Con Pandas, è stato possibile caricare dataset di grandi dimensioni, eseguire operazioni di pulizia e trasformazione delle variabili, e ottenere statistiche descrittive.

2. **Scikit-Learn (Sklearn)**
   Scikit-Learn è una libreria di machine learning fondamentale in Python, utilizzata in questo progetto per implementare diversi algoritmi di apprendimento supervisionato e non supervisionato. Tra le tecniche applicate vi sono la regressione (ad esempio, per predire i prezzi), la classificazione (per predire la disponibilità), il clustering (per il raggruppamento degli annunci) e le tecniche di cross validation (per valutare la performance dei modelli). Scikit-Learn offre inoltre strumenti di pre-processing, come la normalizzazione delle feature e l'encoding delle variabili categoriche.

3. **Matplotlib e Seaborn**
   Per visualizzare i dati e i risultati dell'analisi, sono state utilizzate le librerie Matplotlib e Seaborn. Questi strumenti permettono di creare grafici e visualizzazioni dettagliate che aiutano a comprendere la distribuzione delle variabili, le correlazioni e l'efficacia dei modelli.

4. **NumPy**
   NumPy è stato utilizzato per operazioni di calcolo numerico, che risultano essenziali per il trattamento dei dati all'interno delle pipeline di machine learning. Con NumPy è possibile effettuare calcoli efficienti su grandi array e matrici, migliorando le prestazioni complessive del progetto.

1. Dataset

Il dataset di riferimento per questo progetto è stato scaricato dal repository di open data Kaggle ed è disponibile al seguente link:

https://www.kaggle.com/datasets/arianazmoudeh/airbnbopendata/data

La composizione del dataset è stata rappresentata nel seguente schema, sottolineando, per ogni caratteristica (feature), il numero di elementi non nulli e il tipo di rappresentazione

```
RangeIndex: 102599 entries, 0 to 102598
Data columns (total 26 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   id                              102599 non-null  int64
 1   NAME                            102349 non-null  object
 2   host id                         102599 non-null  int64
 3   host_identity_verified          102310 non-null  object
 4   host name                       102193 non-null  object
 5   neighbourhood group             102570 non-null  object
 6   neighbourhood                   102583 non-null  object
 7   lat                             102591 non-null  float64
 8   long                            102591 non-null  float64
 9   country                         102067 non-null  object
 10  country code                    102468 non-null  object
 11  instant_bookable                102494 non-null  object
 12  cancellation_policy             102523 non-null  object
 13  room type                       102599 non-null  object
 14  Construction year               102385 non-null  float64
 15  price                           102352 non-null  object
 16  service fee                     102326 non-null  object
 17  minimum nights                  102190 non-null  float64
 18  number of reviews               102416 non-null  float64
 19  last review                     86706 non-null   object
 20  reviews per month               86720 non-null   float64
 21  review rate number              102273 non-null  float64
 22  calculated host listings count  102280 non-null  float64
 23  availability 365                102151 non-null  float64
 24  house_rules                     50468 non-null   object
 25  license                         2 non-null       object
dtypes: float64(9), int64(2), object(15)
```

1.1 Descrizione del Dataset

Le diverse variabili del dataset descrivono le proprietà, i profili degli host, le recensioni, e i dati relativi alla disponibilità e ai prezzi. Le variabili principali includono:

- **Attributi dell'Host**: come "host_identity_verified" e "host_response_rate".
- **Attributi della Proprietà**: tipo di soluzione, numero di posti letto, bagni, camere, ecc.
- **Attributi della Posizione**: quartiere, latitudine, e longitudine.
- **Attributi delle Recensioni**: valutazione complessiva, numero di recensioni.
- **Attributi di Prenotazione**: disponibilità, notti minime e politica di cancellazione.

1.2 Esplorazione Iniziale dei Dati

Sono state condotte analisi statistiche di base per esaminare le caratteristiche di ciascuna variabile. Sono stati calcolati indicatori come media, mediana, deviazione standard e percentili, per comprendere la distribuzione dei dati. Un'attenzione particolare è stata posta nel rilevare la presenza di valori nulli o duplicati.

	id	host id	lat	long	Construction year	minimum nights	number of reviews	reviews per month	review rate number	calculated host listings count	availability 365
count	1.025990e+05	1.025990e+05	102591.000000	102591.000000	102385.000000	102190.000000	102416.000000	86720.000000	102273.000000	102280.000000	102151.000000
mean	2.914623e+07	4.925411e+10	40.728094	-73.949644	2012.487464	8.135845	27.483743	1.374022	3.279106	7.936605	141.133254
std	1.625751e+07	2.853900e+10	0.055857	0.049521	5.765556	30.553781	49.508954	1.746621	1.284657	32.218780	135.435024
min	1.001254e+06	1.236005e+08	40.499790	-74.249840	2003.000000	-1223.000000	0.000000	0.010000	1.000000	1.000000	-10.000000
25%	1.508581e+07	2.458333e+10	40.688740	-73.982580	2007.000000	2.000000	1.000000	0.220000	2.000000	1.000000	3.000000
50%	2.913660e+07	4.911774e+10	40.722290	-73.954440	2012.000000	3.000000	7.000000	0.740000	3.000000	1.000000	96.000000
75%	4.320120e+07	7.399650e+10	40.762760	-73.932350	2017.000000	5.000000	30.000000	2.000000	4.000000	2.000000	269.000000
max	5.736742e+07	9.876313e+10	40.916970	-73.705220	2022.000000	5645.000000	1024.000000	90.000000	5.000000	332.000000	3677.000000

Durante l'analisi, è stato notato che:

- **Valori Nulli**: Molte variabili contenevano valori nulli, specialmente nelle informazioni sull'host e in alcuni attributi di prenotazione.

- **Distribuzione delle Variabili**: Alcune variabili, come la “neighbourhood”, hanno mostrato una distribuzione relativamente uniforme, mentre altre, come “room_type”, erano dominate da poche categorie.
- **Errori di Formattazione:** Diverse variabili di tipo testuale erano affette da errori grammaticali o di formato, che avrebbero inficiato sulla consistenza delle analisi.

1.3 Pulizia e Pre-processing dei Dati

Per garantire l'integrità dei dati e migliorare l'efficacia dei modelli, è stato effettuato un processo di pulizia articolato in più fasi:

1. **Gestione dei Valori Nulli**:
    - Per variabili con un numero ridotto di valori mancanti, sono stati utilizzati valori predefiniti o medie, a seconda del contesto.
    - Alcuni valori sono stati interpolati in base ad attributi correlati (ad esempio, valori di “price” o “service fee” mancanti sono stati imputati con la media per annunci simili).
    - Dove non era possibile effettuare una stima attendibile e il valore era critico per l'analisi, l’intera riga è stata rimossa.
2. **Correzione degli Errori di Formattazione**:
    - Alcune colonne, come “price” e “service fee”, erano memorizzate come stringhe e sono state convertite in numeri per facilitare le analisi.
    - Errori grammaticali o formattazioni non uniformi sono stati corretti per garantire consistenza.
    - Sono stati anche gestiti i valori non logicamente attendibili, come “availability_365” che non può avere un valore superiore a 365 giorni, attraverso l’applicazione di regole logiche.

3. **Eliminazione dei Duplicati**:
    - È stato verificato e rimosso ogni dato duplicato per assicurare che ogni annuncio nel dataset rappresentasse una proprietà unica.
4. **Eliminazione delle Feature Ridondanti o Poco Rilevanti**:
    - Durante il processo di feature selection, sono stati scartati gli attributi considerati irrilevanti, come "country", "country_code" e "neighbourhood_group", che non fornivano informazioni aggiuntive utili per i modelli.
    - Altre feature, come "host_name" e "latitude/longitude" sono state eliminate perché non rilevanti ai fini della predizione e per evitare ridondanze.
5. **Encoding delle Variabili Categoriche**:
    - Le variabili categoriche rilevanti sono state trasformate in formato numerico per poter essere utilizzate nei modelli di machine learning. È stato applicato il one-hot encoding o la trasformazione attraverso variabili indicatrici, a seconda del modello, per garantire che i modelli stessi potessero interpretare correttamente queste variabili.

2. Price Prediction Task (Regressione)

L'obiettivo è sviluppare un modello di machine learning in grado di stimare accuratamente il costo dell'affitto di una proprietà. La feature target per questo task è '**price'**, il prezzo dell'annuncio. Questa variabile è continua e rappresenta un valore economico, rendendo questo task un problema di **regressione**.

Sono state selezionate diverse variabili di input, raggruppabili in cinque categorie principali:

- **Caratteristiche dell'Host** (ad es. "host_response_rate")
- **Caratteristiche della Proprietà** (ad es. "room_type", "number_of_bedrooms")
- **Caratteristiche di Posizione** (ad es. "neighbourhood")
- **Feedback e Recensioni** (ad es. "number_of_reviews", "review_scores_rating")
- **Attributi di Prenotazione** (ad es. "availability_365", "minimum nights")

Alcune variabili non rilevanti o ridondanti sono state eliminate, mentre altre, in formato categorico, sono state convertite tramite l'utilizzo di variabili indicatrici per essere incluse nel modello. Le variabili numeriche sono state invece scalate con un Min-Max Scaler.

2.1 Algoritmi di Regressione Utilizzati

Per il task di predizione del prezzo sono stati sperimentati e confrontati diversi algoritmi di machine learning, tutti basati sul paradigma dell'apprendimento supervisionato:

- **Random Forest Regressor**: Algoritmo di apprendimento basato su alberi decisionali, ideale per la gestione di dataset con molte feature. Grazie alla sua capacità di evitare l'overfitting, può rivelarsi efficace per la predizione di prezzi in un contesto complesso come quello degli annunci Airbnb.

- **Extreme Gradient Boosting (XGBoost)**: Un'implementazione avanzata del gradient boosting, ottimizzata per alte prestazioni e scalabilità. Questo modello è particolarmente adatto per migliorare l'accuratezza della predizione in presenza di variabili interdipendenti.

- **AdaBoost (Adaptive Boosting):** Un algoritmo di ensemble di machine learning che crea un modello forte combinando molti modelli deboli, come piccoli alberi decisionali. L'idea principale dietro AdaBoost è migliorare iterativamente la precisione del modello concentrandosi sugli errori commessi nelle previsioni del modello precedente.

2.2 Valutazione e Ottimizzazione dei Modelli

Il dataset è stato suddiviso in **training set** (80%) e **test set** (20%) e i modelli sono stati addestrati e valutati sul training set. La performance di ciascun modello è stata misurata utilizzando metriche di regressione, tra cui:

- **R-squared (R^2)**: Metrica che indica la percentuale di varianza della feature target spiegata dalle variabili di input. Un R^2 vicino a 1 indica un modello ben adattato.
- **Mean Squared Error (MSE)**: Metrica che misura la media dei quadrati degli errori tra i valori predetti e quelli osservati, con valori più bassi che indicano una predizione più accurata.

Inoltre, è stata applicata la tecnica della **K-Fold Cross Validation** per ottenere una valutazione robusta, suddividendo il training set in 10 fold e valutando sulla media dei risultati per ogni iterazione.

Tramite iterate applicazioni della Cross Validation sono stati individuati gli Iperparametri migliori per ogni modello in esame.

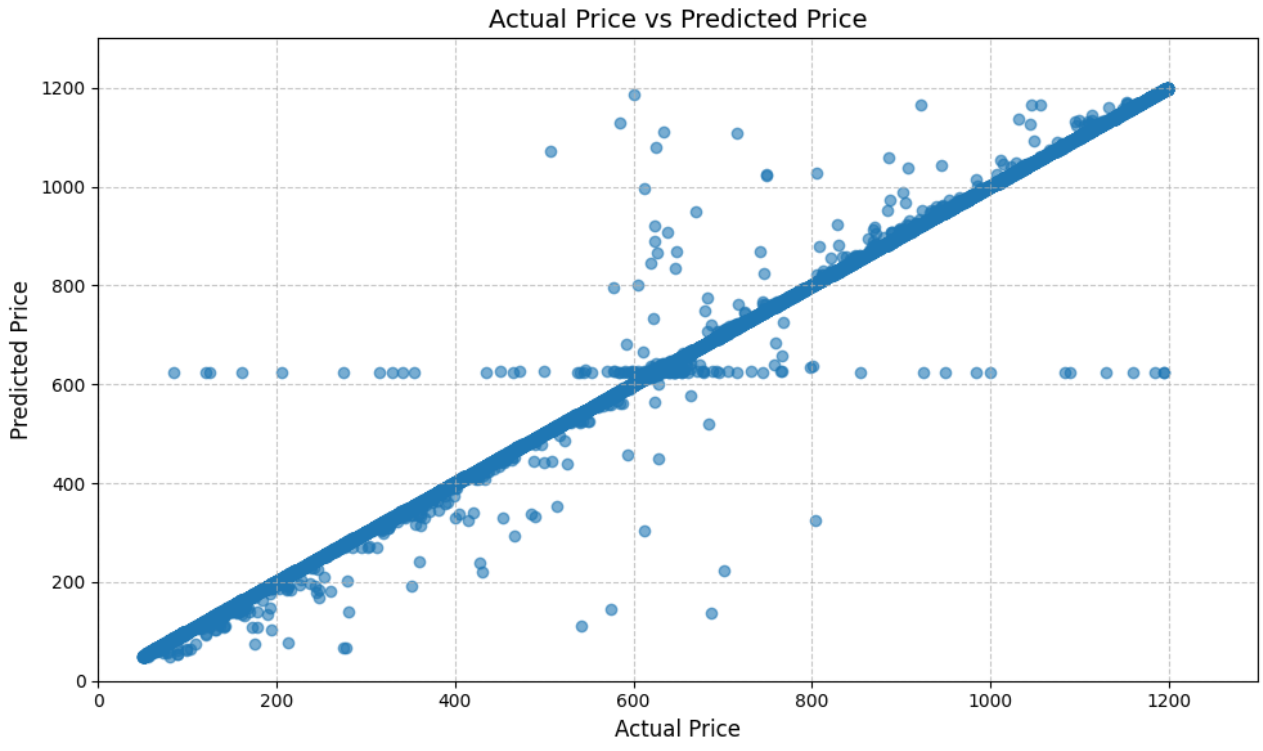
2.3 Risultati

Ogni modello è stato quindi addestrato su tutto il training set con la miglior configurazione di Iperparametri individuata. Come per la validazione delle configurazioni intermedie, anche per i risultati verranno valutati attraverso le metriche MSE ed R^2.

- Il RandomForest ha ottenuto questi risultati:

```
Final Model Performance (trained on entire dataset):
Mean Squared Error: 585.5587
R-squared Score: 0.9947
Standard Deviation of Predicted Values: 331.0689
Standard Deviation of Actual Values: 331.6778
```

Questo modello ha mostrato prestazioni eccellenti, con un alto valore di R^2 e un basso MSE. Essendo un metodo ensemble che combina più alberi decisionali, risulta particolarmente robusto contro l'overfitting grazie alla randomizzazione. Il plotting risultante tra i valori predetti e i valori effettivi è il seguente:



- Il XGBoost Regressor ha ottenuto questi risultati:

```
Final Model Performance (trained on entire dataset):
Mean Squared Error: 572.4861
R-squared Score: 0.9948
Standard Deviation of Predicted Values: 331.3416
Standard Deviation of Actual Values: 331.6778
```

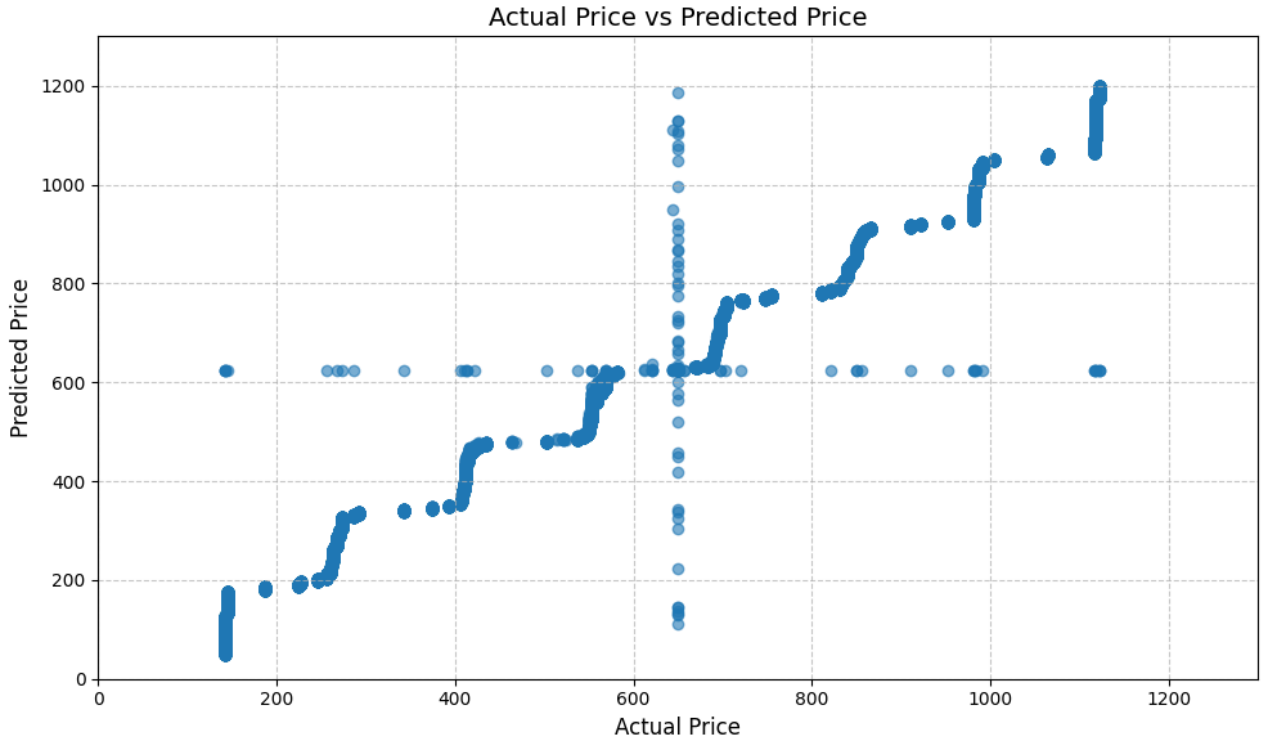
XGBoost ha prodotto risultati identici a quelli di Random Forest, con un MSE e un R^2 perfettamente comparabili, rendendolo una scelta valida per una predizione precisa. Il plotting dei risultati è il seguente:

La visualizzazione dei risultati di XGBoost lascia intendere che il modello produce predizioni un po' meno accurate del RandomForest, evidenziando una varianza dei risultati più accentuata e meno resistenza all'overfitting, seppur di bassa entità.

- AdaBoost ha ottenuto questi risultati:

```
Final Model Performance (trained on entire dataset):
Mean Squared Error: 2068.6654
R-squared Score: 0.9812
Standard Deviation of Predicted Values: 309.4457
Standard Deviation of Actual Values: 331.6778
```

AdaBoost tende a performare leggermente peggio di Random Forest e XGBoost su questo dataset, mostrando un MSE sensibilmente più elevato e un R^2 inferiore. Anche la deviazione standard ne risulta affetta, con AdaBoost che tende ad essere più conservativo nelle predizioni rispetto alla reale distribuzione. Il plotting dei risultati è:



Il riscontro visivo conferma i risultati, mostrando un modello molto conservativo e tendente all'overfitting su questo set di dati.

2.4 Considerazioni finali

Tra gli algoritmi analizzati, Random Forest e XGBoost offrono i migliori risultati complessivi, con Random Forest che si distingue per la sua robustezza all'overfitting, fornendo il modello che ha ottenuto i risultati migliori.

Gli iperparametri per Random Forest con cui è stato costruito il modello sono:

***n_estimators:* 100** Per controllare il numero di alberi predittori nella foresta

min_samples_split: 2 Numero minimo di esempi per una divisione

min_samples_leaf: 1 Numero minimo di esempi per una foglia

max_features: 1.0 Feature da considerare per un best_split (1.0 = tutte)

max_leaf_nodes: None Limite massimo di nodi foglia in un albero

min_impurity_decrease: 0 Minima soglia di riduzione dell'errore per split

ccp_alpha: 0.0 Fattore di pruning (0.0 = no pruning)

random_state: 42 Per stabilire il fattore di randomicità nel bootstrap sampling e nel best split sampling

L'obbiettivo prefissato è stato raggiunto, poiché il modello permette ai proprietari per ottimizzare le proprie tariffe mentre i clienti possono individuare delle occasioni. In particolare il plot dei risultati precedente permette di visualizzare in maniera immediata gli annunci sovrapprezzati e le occasioni.

- Annunci sovraprezzati: Punti che si trovano al di sotto della linea diagonale (dove il prezzo reale supera quello predetto).
- Annunci sottoprezzati: Punti che si trovano al di sopra della linea diagonale (dove il prezzo predetto supera quello reale).

3. Availability Prediction Task (Classificazione)

L’obbiettivo di questa task è sviluppare un modello di machine learning capace di predire la disponibilità di un annuncio Airbnb, cioè se una soluzione sarà prenotabile o meno in un determinato periodo. La variabile target, "bookable", è binaria, rendendo questo task un problema di classificazione.

Sono state selezionate alcune variabili rappresentative, tra cui:

- Caratteristiche della Proprietà (es. "room_type", "number_of_beds")
- Attributi di Prenotazione (es. "minimum nights", "availability_365")
- Caratteristiche dell'Host (es. "host_response_rate")
- Posizione (es. "neighbourhood")

Le variabili categoriche sono state convertite tramite one-hot encoding, mentre quelle numeriche sono state normalizzate con un Min-Max Scaler per garantire che i modelli interpretino correttamente queste variabili.

3.1 Algoritmi di Classificazione Utilizzati

Come per il task precedente, abbiamo testato i 3 algoritmi principali per l’apprendimento supervisionato, questa volta nella loro versione per la classificazione:

- Random Forest Classifier: per sfruttare la sua robustezza all’overfitting
- XGBoost Classifier
- AdaBoost Classifier

3.2 Valutazione e Ottimizzazione dei Modelli

Il dataset è stato suddiviso in training set (80%) e test set (20%) per addestrare e valutare i modelli.

Le metriche utilizzate per valutare i modelli di classificazione includono:

Accuracy: Percentuale di previsioni corrette sul totale.

F1 Score: Metrica che considera sia precision che recall, utile per problemi con classi sbilanciate.

AUC-ROC: Area sotto la curva ROC, per misurare la qualità delle previsioni in termini di sensibilità e specificità.

Per garantire una valutazione affidabile, abbiamo applicato la K-Fold Cross Validation con 10 fold durante il tuning degli iperparamentri, calcolando la media delle performance per ogni iterazione.

3.3 Risultati

Dopo aver addestrato i modelli con la miglior configurazione di iperparametri, i risultati ottenuti sono i seguenti.

- Random Forest Classifier:

```
RandomForestClassifier model scores:
              precision    recall  f1-score   support

           0       0.51      0.51      0.51     10345
           1       0.50      0.50      0.50     10036

    accuracy                           0.50     20381
   macro avg       0.50      0.50      0.50     20381
weighted avg       0.50      0.50      0.50     20381



Mean Squared Error: 0.4964
R-squared Score: -0.9860
Standard Deviation of Predicted Values: 0.5000
Standard Deviation of Actual Values: 0.4999
AUC-ROC Score: 0.5035
```

RandomForest non è riuscito a costruire un modello che sia in grado di predire accuratamente la feature target, di fatto regredendo ad un random guessing della stessa, mostrandosi non adatto allo scopo.

- XGBoost Classifier:

```
XGBClassifier model scores:
              precision    recall  f1-score   support

           0       0.51      0.50      0.51     10345
           1       0.50      0.50      0.50     10036

    accuracy                           0.50     20381
   macro avg       0.50      0.50      0.50     20381
weighted avg       0.50      0.50      0.50     20381



Mean Squared Error: 0.4971
R-squared Score: -0.9888
Standard Deviation of Predicted Values: 0.5000
Standard Deviation of Actual Values: 0.4999
AUC-ROC Score: 0.5029
```

Sembra che neanche XGBoost sia riuscito nell'intento, costruendo un modello che ha ottenuto gli stessi risultati del modello precedente. Probabilmente è segnale sulla natura della feature target, poco incline alla predizione.

- AdaBoost Classifier:

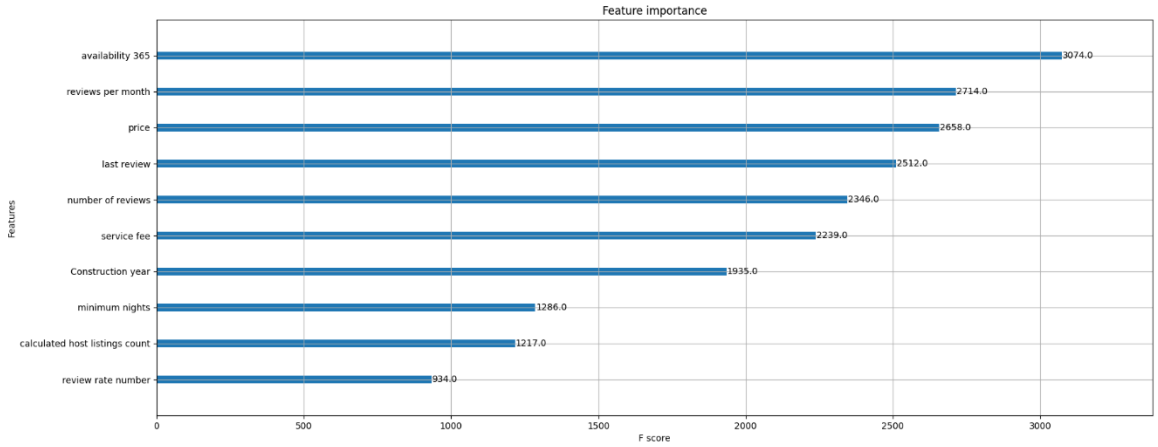
```
AdaBoostClassifier model scores:
              precision    recall  f1-score   support

           0       0.51      0.62      0.56     10345
           1       0.49      0.38      0.43     10036

    accuracy                           0.50     20381
   macro avg       0.50      0.50      0.49     20381
weighted avg       0.50      0.50      0.49     20381



Mean Squared Error: 0.4999
R-squared Score: -1.0000
Standard Deviation of Predicted Values: 0.4862
Standard Deviation of Actual Values: 0.4999
AUC-ROC Score: 0.4984
```

Neanche AdaBoost ha fornito risultati ottimali: il modello generato predice leggermente meglio i True False ma in generale le perfomance non raggiungono soglie ottimali.

3.4 Considerazioni sui Risultati

Tra gli algoritmi testati, nessuno è riuscito a soddisfare efficacemente le richieste del task in esame. Sembra che la feature target non sia deducibile dai dati in nostro possesso, oppure che sia completamente indipendente. Potrebbe anche significare che l'approccio scelto non è corretto ed andrebbero esplorate altre soluzioni.

E' stata condotta un'analisi della Features correlation tra la feature target e le altre feature del dataset: di seguito ne vengono presentate le prime 10 per importanza.



Ciò dimostra che la feature target abbia buone correlazioni con le altre feature, confermando le considerazioni iniziali sul task; se ne deduce che probabilmente l'approccio al problema è sbagliato ed andrebbero esplorate altre soluzioni.

4. Guest Preference Segmentation Task (Clustering)

L'obiettivo di questa task è applicare tecniche di hard clustering per segmentare le soluzioni Airbnb in gruppi omogenei, identificando pattern comuni nelle preferenze. Questa analisi può aiutare a sviluppare strategie di marketing più mirate e a personalizzare le offerte per differenti tipologie di utenti.

Dopo analisi a posteriori, sono state selezionate le migliori variabili che descrivono le caratteristiche dell'annuncio e le preferenze degli ospiti, tra cui:

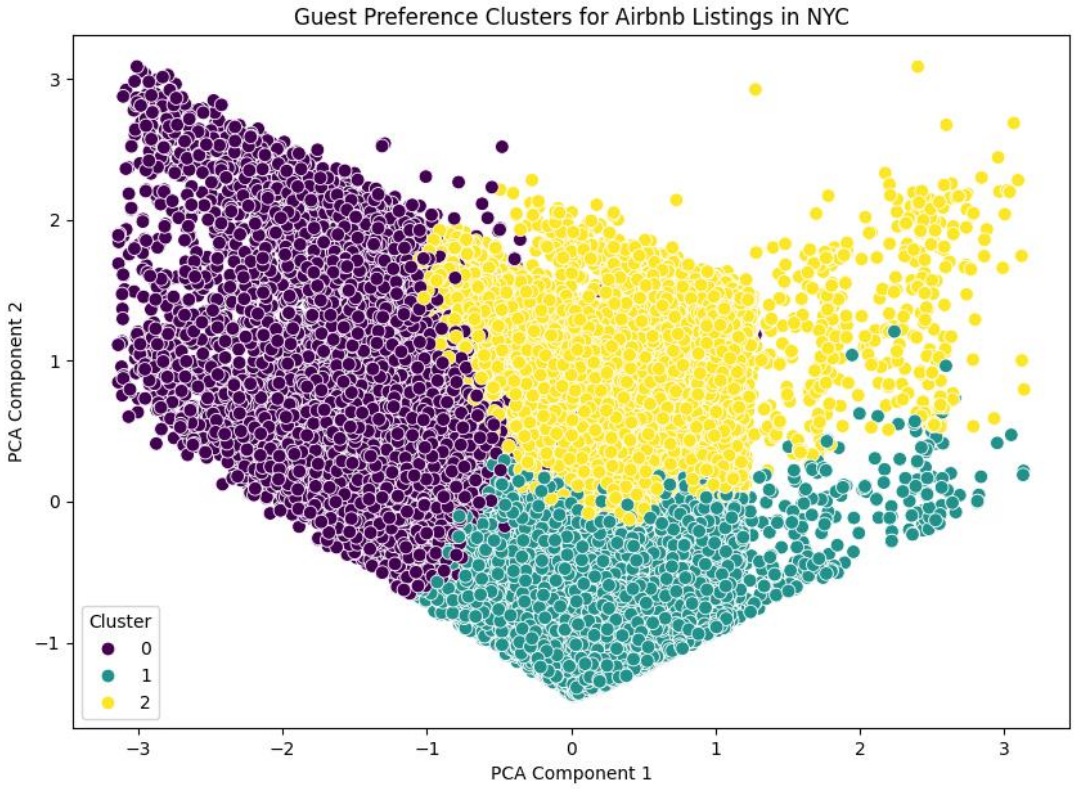
- **neighbourhood group**: il gruppo o la zona in cui si trova la proprietà
- **host_identity_verified**: indica se l'identità dell'host è verificata
- **room type**: tipologia di stanza o spazio offerto (es. intera proprietà, stanza privata)
- **price**: prezzo dell'annuncio
- **minimum nights**: numero minimo di notti richieste per la prenotazione

- **instant_bookable**: specifica se la proprietà è prenotabile immediatamente
- **cancellation_policy**: politica di cancellazione applicata (es. flessibile, moderata, rigida)
- **availability 365**: numero di giorni in cui la proprietà è disponibile all'anno
- **reviews per month**: numero medio di recensioni ricevute mensilmente

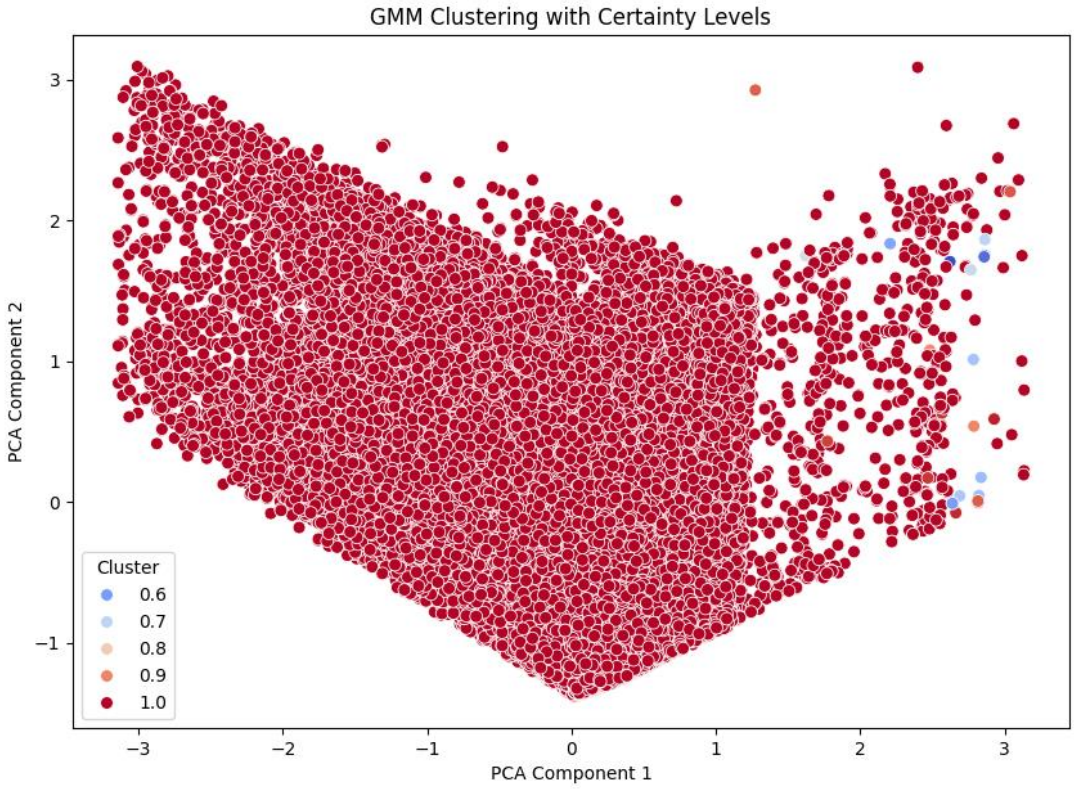
Per preparare i dati per il clustering, le variabili categoriche sono state codificate tramite **one-hot encoding**, mentre le variabili numeriche sono state normalizzate utilizzando uno **Standard Scaler** per evitare che le scale di misura influenzino i risultati del clustering.

4.1 L'algoritmo

L'algoritmo scelto per l'apprendimento è il **K-Means Clustering:** Un metodo popolare che richiede di definire a priori il numero di cluster, ideale per identificare gruppi distinti e ben separati. Diversi round di validazione hanno portato all'individuazione di 3 cluster completamente separabili nel dataset. Di seguito se ne fornisce una forma visiva dopo aver eliminato gli outliers che avevano una performance di Z-score maggiore di 3 deviazioni standard. La visualizzazione è favorita attraverso il metodo del PCA (Principal Component Analysis) a 2 dimensioni.



E’ stata condotta una validazione statistica della suddivisione attraverso l’applicazione di un algoritmo di soft clustering probabilistico, l’Expectation-Minimization (EM) algorithm con un Gaussian Mixture Model. I risultati ottenuti vanno a validare perfettamente la suddivisione.



4.2 Considerazioni sui risultati

I risultati permettono di visualizzare le preferenze degli utenti nel mercato degli AirBnB newyorkesi: per esempio si può notare, attraverso un analisi delle caratteristiche dei centroidi dei vari cluster, che un cliente preferirebbe un intero appartamento per soste brevi nel quartiere di Manhattan e preferirebbe pagare una cifra attorno ai 500 dollari a notte; Ma la stessa cifra è disposto a spenderla a Brooklyn solo se il soggiorno è della durata di un mese. Altre considerazioni possono essere inferenziate dai risultati.

6. Conclusioni

Il progetto ha esplorato l'utilizzo del machine learning per analizzare e comprendere meglio le dinamiche degli annunci Airbnb, fornendo strumenti predittivi e informativi utili per ottimizzare le strategie di prezzo, prevedere la disponibilità e segmentare le preferenze degli ospiti. Attraverso tecniche di regressione, classificazione e clustering, sono stati

sviluppati modelli che offrono insight dettagliati sulle variabili chiave che influenzano il mercato degli affitti a breve termine. Sebbene alcuni modelli abbiano raggiunto prestazioni soddisfacenti, come quelli per la previsione dei prezzi e la segmentazione delle preferenze, altri, come il modello di predizione della disponibilità, hanno evidenziato la necessità di approcci alternativi per una maggiore precisione.

Sviluppi Futuri

Per migliorare ulteriormente i risultati ottenuti e superare le attuali limitazioni, è possibile sviluppare nuove strategie, tra cui:

- **Feature Engineering Avanzato**: L'aggiunta di nuove caratteristiche, come la stagionalità o eventi locali, potrebbe rendere i modelli più robusti e accurati.
- **Modelli Alternativi**: L'utilizzo di reti neurali o tecniche di apprendimento profondo potrebbe offrire miglioramenti nelle task di classificazione, soprattutto per feature più complesse.
- **Integrazione di Dati Esterni**: Incorporare dati provenienti da fonti esterne, come la domanda turistica o il calendario degli eventi locali, potrebbe contribuire a rendere le previsioni più accurate.

In conclusione, il progetto ha rappresentato un primo passo importante nella costruzione di modelli di machine learning applicati al mercato Airbnb. Gli sviluppi futuri, orientati a una maggiore personalizzazione e all’uso di fonti di dati esterne, potrebbero ulteriormente ampliare il potenziale di questi modelli, rendendoli strumenti sempre più efficaci per la gestione e l’ottimizzazione delle proprietà su piattaforme di affitto a breve termine.