Airbnb Price Prediction

Austin, TX
Group 3 – Gokul Govindasamy Sutharsan, Aashini Bala, Nashra Ali

Overview

This project analyzes Airbnb listing and review data for the city of Austin to gain insights into pricing patterns and guest behavior. The primary goals are to develop a machine learning model that accurately predicts listing prices based on key features and to create an AI chatbot that can interactively answer questions using the dataset. The workflow includes data cleaning, exploratory data analysis (EDA), predictive modeling, anomaly detection for fake reviews, and chatbot integration using modern AI tools and libraries such as Pandas, Scikit-learn, LangChain, and OpenAI's GPT-4o-mini.

Libraries Used

- pandas: Used for data manipulation and analysis.
- numpy: Utilized for numerical operations.
- Scikitlearn: Utilized for building ML model
- Langchain: Utilized for building AI chatbot pipeline
- OpenAI: OpenAI's gpt-4o-mini model has been used to build the chatbot.

Data Import

The script loads two CSV files:

- listings-2.csv: Contains information about Airbnb listings.
- reviews-2.csv: Contains user reviews related to the listings.

```
listing = pd.read_csv('listings-2.csv')
review = pd.read_csv('reviews-2.csv')
```

Data Cleaning and Transformation

1. Initial Data Exploration

The script first inspects the column names of both datasets:

```
listing.columns
review.columns
```

2. Dropping Irrelevant Columns

To simplify the dataset and retain only useful features, several columns are removed from the listing DataFrame:

```
listing.drop(columns = ['listing_url', 'scrape_id', 'last_scraped', 'source', 'host_thumbnail_url', 'host_picture_url',
      'host_neighbourhood', 'host_has_profile_pic', 'bathrooms_text', 'calendar_last_scraped', 'first_review', 'last_review', 'license',
      'host_verifications', 'neighbourhood_group_cleansed', 'calendar_updated', 'neighbourhood', 'has_availability'], inplace = True)
```

Similarly, the review dataset is cleaned by removing rows where the comments field is missing:

```
review.dropna(subset='comments', inplace= True)
```

3. Handling Missing Values

The script ensures essential columns do not contain missing values:

```
listing.dropna(subset = ['review_scores_rating'], inplace = True)
```

4. Converting Price Column to Numeric Format

Since price data includes a currency symbol ($), it is converted to a numeric format:

```
listing['price'].replace({'\\$':''}, regex=True, inplace=True)
listing['price'] = pd.to_numeric(listing['price'], errors='coerce')
listing.dropna(subset=['price'], inplace=True)
```

5. Encoding Categorical Variables

Some categorical columns are converted into binary format for easier analysis:

```
listing['instant_bookable'].replace({'t':1, 'f':0}, inplace=True)
listing['host_identity_verified'].replace({'t':1, 'f':0}, inplace=True)
```

Conclusion

The dataset is now cleaned and prepared for further analysis. The following key transformations were applied:

1. Removal of unnecessary columns.
2. Handling of missing values.
3. Conversion of the price column into a numeric format.
4. Encoding of categorical variables.

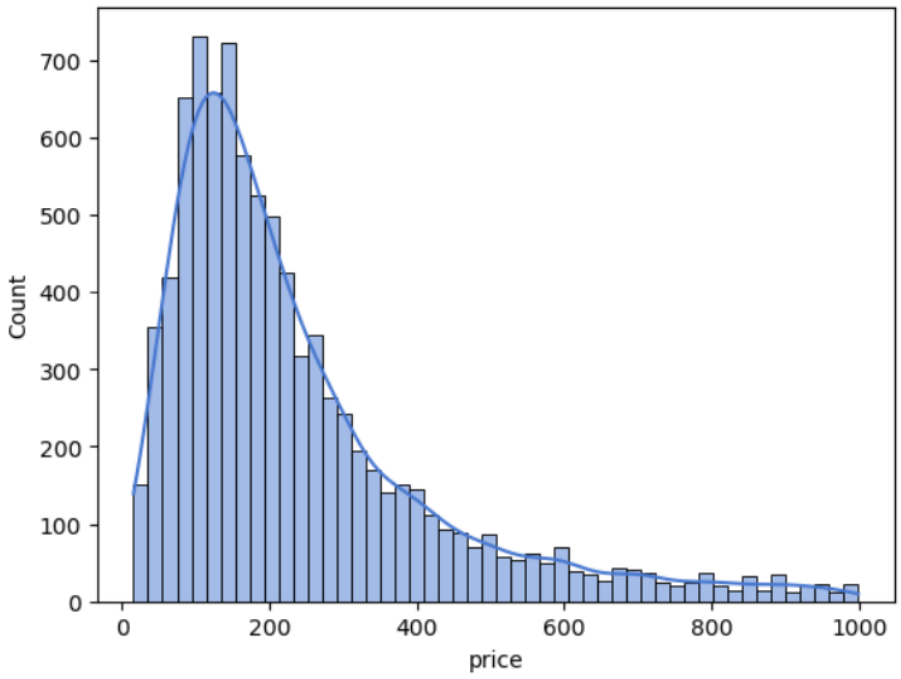
Exploratory Data Analysis (EDA) Report

The EDA focuses on understanding the relationships between various features in the Airbnb listings dataset. The analysis uses Seaborn and Matplotlib for visualization.

1. Price Distribution

- **Observation:** The histogram of `listing['price']` is right-skewed. This indicates that a majority of the listings have lower prices, while a smaller number of listings have significantly higher prices.

- **Interpretation:** This is common in many markets, including real estate and rentals, where you have a larger supply of standard options and a smaller supply of luxury or high-end options.

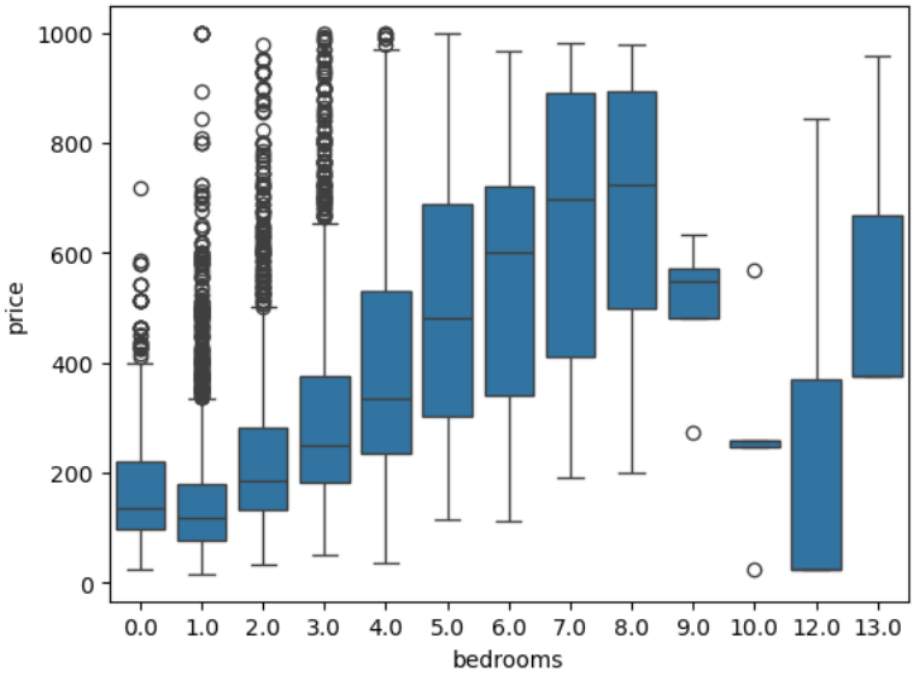


2. Room Type vs. Price

- **Observation:** The boxplot compares the price distribution across different numbers of bedrooms. Generally, the median price increases as the number of bedrooms increases, which is expected. However, there's significant variability in prices for each bedroom category, and there are many outliers, especially for higher bedroom counts.

- **Interpretation:**

- The increasing median price confirms the general expectation that larger properties (more bedrooms) cost more.
- The wide spread and the presence of outliers within each bedroom category suggest that factors other than the number of bedrooms influence price (e.g., location, amenities, condition).



3. Superhost Status vs. Estimated Occupancy

- **Observation:** The bar plot compares the estimated occupancy for superhosts (presumably 1) and non-superhosts (presumably 0). There appears to be a difference in estimated occupancy between the two groups.

- **Interpretation:**

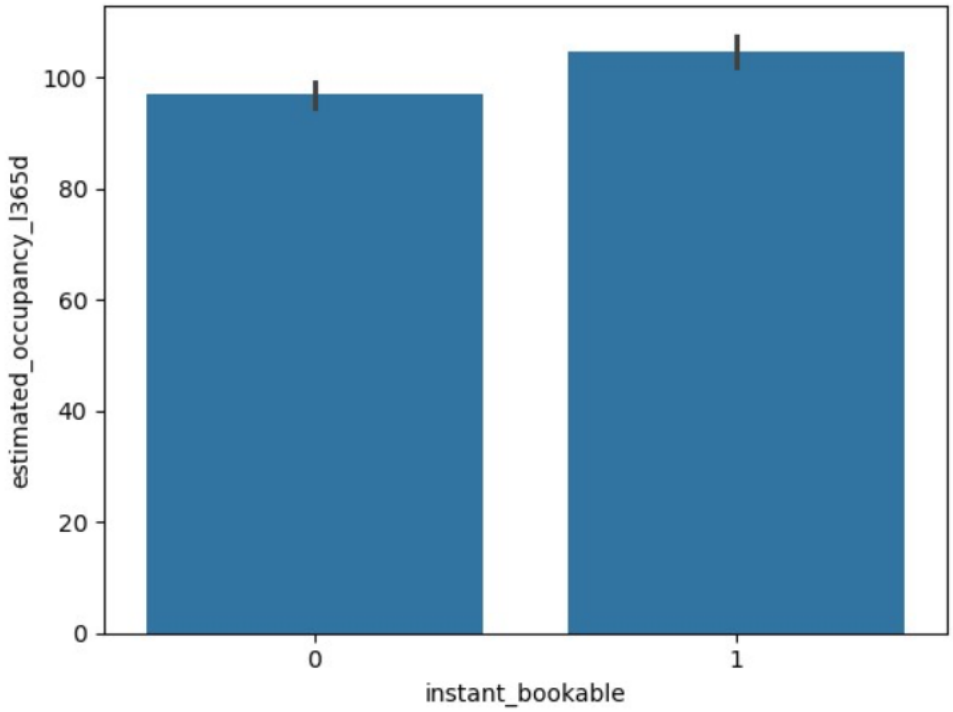
The plot suggests that superhosts may have a different occupancy rate than non-superhosts. This could indicate that superhost status is associated with higher or lower occupancy.

4. Instant Bookable vs. Estimated Occupancy

- **Observation:** This bar plot compares the estimated occupancy for instantly bookable listings (presumably 1) and non-instantly bookable listings (presumably 0). A difference in occupancy between the two groups is visible.

- **Interpretation:**

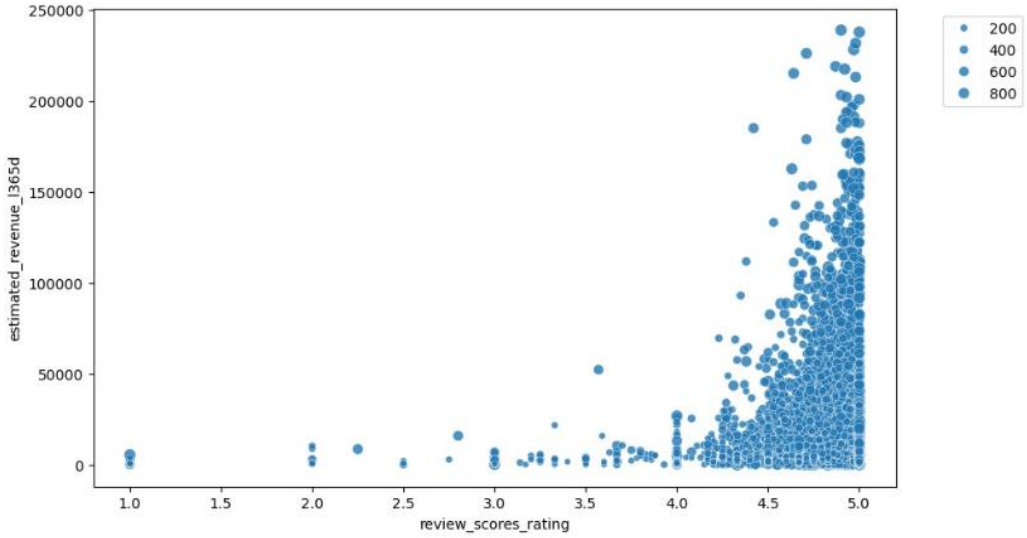
The plot suggests that instant bookability might affect occupancy rates.



5. Revenue vs. Reviews

- **Observation:** This scatter plot visualizes the relationship between `review_scores_rating` and `estimated_revenue_1365d`, with the size of the points representing `price`. There's a general trend of increasing revenue with higher review scores, but there's also a wide spread. The size of the points adds another dimension, showing how price relates to both revenue and reviews.
- **Interpretation:**

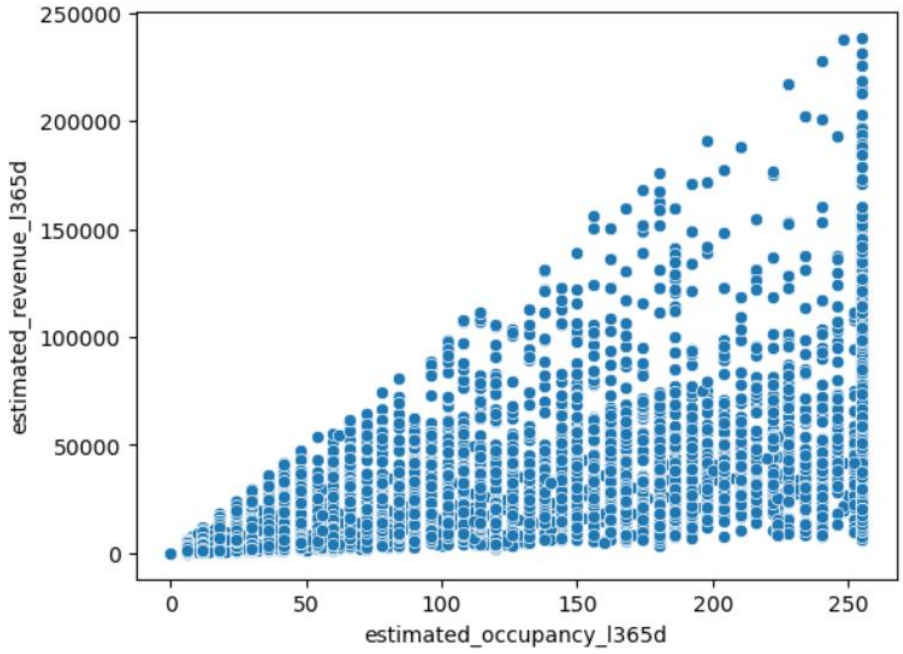
    - Higher review scores tend to be associated with higher estimated revenue, suggesting that guest satisfaction plays a role in financial performance.
    - The variability in revenue at each review score indicates that other factors (e.g., pricing strategy, location) are also important.
    - The size of the points (`price`) reveals that higher-priced listings can achieve high revenue, but this isn't solely dependent on review scores.



6. Occupancy vs. Revenue

- **Observation:** The scatter plot shows the relationship between `estimated_occupancy_1365d` and `estimated_revenue_1365d`. There's a clear positive correlation, indicating that higher occupancy generally leads to higher revenue.
- **Interpretation:**

This strong positive relationship is intuitive: more bookings directly translate to more revenue.



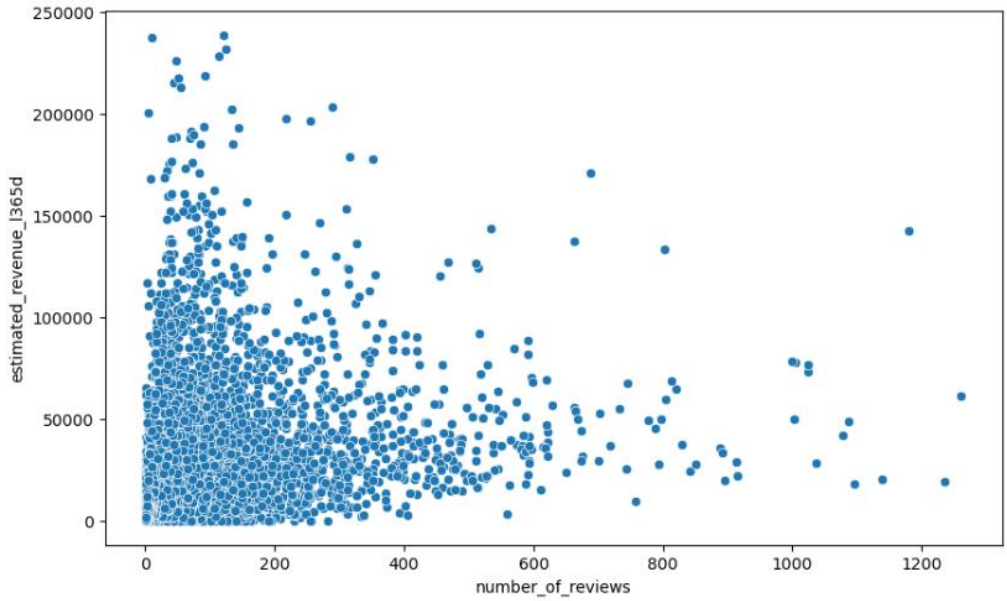
7. Number of Reviews vs. Revenue

Interpretation:

Dense clustering at lower review counts – Most data points are concentrated in the range of 0-200 reviews. Revenue varies significantly – Estimated revenue ranges from near zero to over $200,000, with most values concentrated in the lower range. Sparse data at high review counts – Fewer products have more than 600 reviews, and even among them, revenue varies widely. Outliers exist – Some points represent exceptionally high revenues despite relatively low review counts, while others have many reviews but moderate revenue.

Interpretation:

- **No clear linear correlation** – While higher reviews generally indicate popularity, revenue does not always increase proportionally. Some products generate high revenue despite having fewer reviews.
- **Possible niche products** – High-revenue, low-review items may indicate premium or high-value products that sell well with fewer purchases.
- **Long-tail effect** – Many products with low reviews also have low revenue, suggesting they are not bestsellers but contribute to overall sales.
- **Diminishing returns on reviews?** – Beyond a certain point (600+ reviews), additional reviews do not appear to significantly impact revenue growth.



8. Review Score Rating vs. Number of Reviews

Observation:

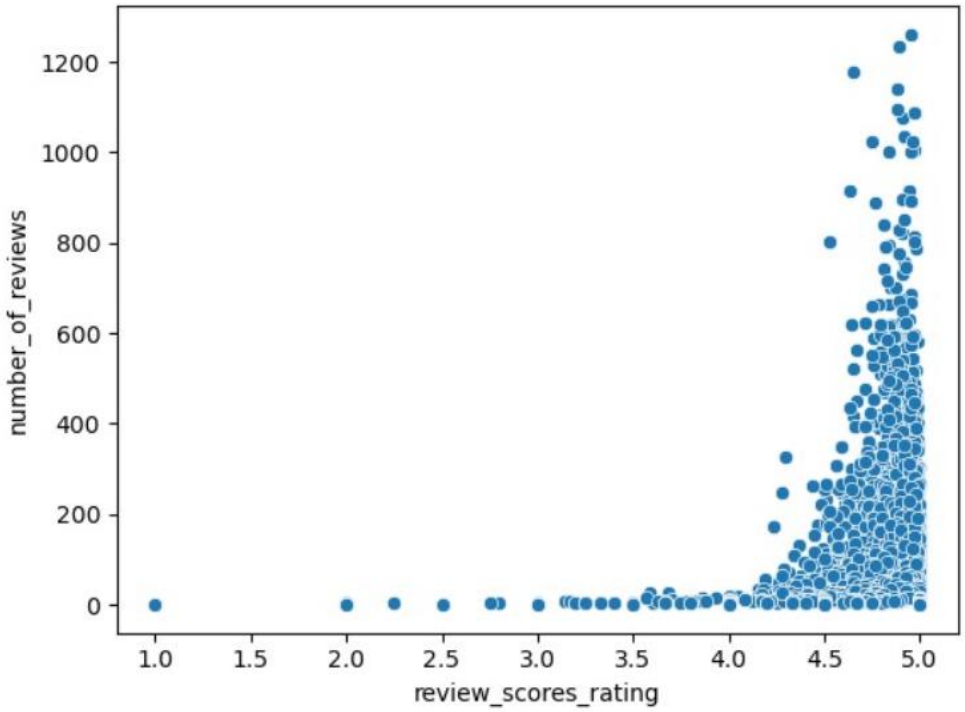
The scatter plot illustrates the relationship between the **review scores rating** and the **number of reviews** for Airbnb listings.

Key Insights:

- **Spread of Data:** The data points are spread across a wide range of review scores and numbers of reviews, indicating a diverse range of listings.
- **No Strong Correlation:** There doesn't appear to be a strong linear correlation between review scores and the number of reviews. This suggests that a higher rating doesn't necessarily translate to a larger number of reviews, and vice versa.
- **Outliers:** There are some outliers, particularly listings with very high numbers of reviews but relatively low review scores, and vice versa. These outliers might warrant further investigation to understand their unique characteristics.

Interpretation:

The scatter plot suggests that there's no simple, direct relationship between review scores and the number of reviews for Airbnb listings. While some listings with high ratings might attract more reviews, and vice versa, there are many other factors that influence both.



Key Observations from Exploratory Data Analysis (EDA) Informing Model Selection:

- ## High Correlation with Price:

  Variables such as accommodates, bedrooms, bathrooms, beds, and review_scores_rating displayed strong relationships with listing prices, suggesting they are critical predictors.

- ## Categorical Influence:

  Features like room_type, property_type, and host_is_superhost showed distinct price distributions across categories, making them valuable for one-hot encoding during model training.

- ## Review Metrics as Demand Signals:

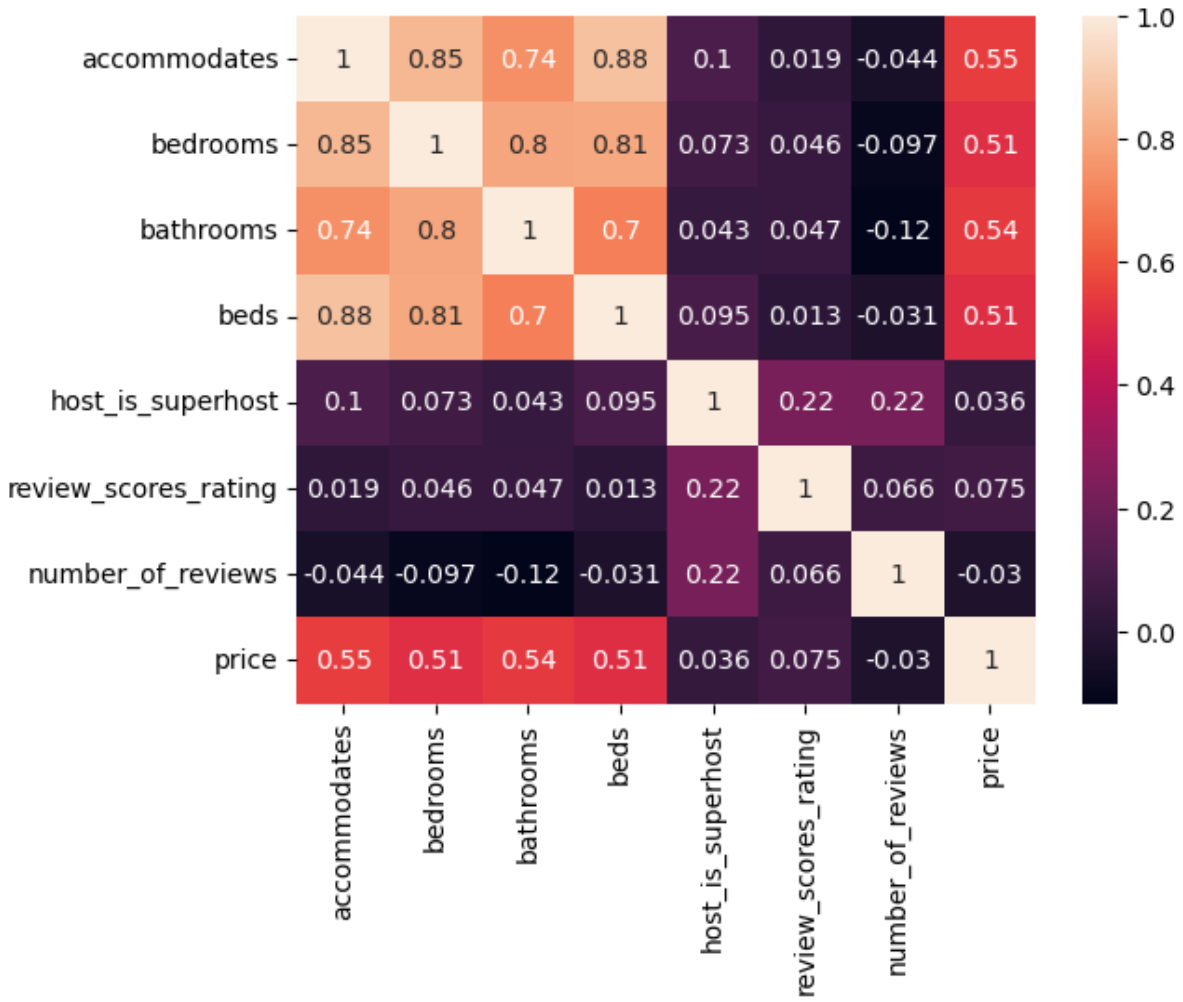
  Metrics like number_of_reviews and review_scores_rating served as proxies for guest satisfaction and listing popularity, which correlate with both revenue and pricing.

- ## Occupancy and Revenue Linkage:

  Listings with higher occupancy rates and superhost status tended to generate greater revenue—highlighting the combined influence of quality, trust, and visibility.

- ## Outlier Awareness:

  The EDA visualizations uncovered several high-priced listings with fewer reviews or lower ratings. Including robust features helps the model handle such variance better.



Predictive Modelling:

Feature Preparation

- **Selection:** We identify the key listing attributes that most influence price—size metrics (number of guests accommodated, bedrooms, bathrooms, beds), listing type (entire home vs. private room), host quality signals (superhost status, review scores, review counts).
- **Transformation:**
  - Categorical features (e.g. room type) are converted into separate indicator variables so the model can learn distinct price patterns for each category.
  - Numerical features remain in their original scale (though in some projects you might standardize or log-transform skewed values).

Train/Test Split

- The cleaned dataset is divided into two parts: a larger “training” set to teach the model, and a held-out “test” set to evaluate its performance. This prevents overfitting and gives an honest estimate of how the model will behave on new listings.

Model Training

- **Random Forest Regressor:**
  - Builds many decision trees, each trained on a random subset of listings and features.
  - Each tree "votes" a price prediction, and the forest's output is the average of these votes.
  - Key advantages: automatically accounts for feature interactions, less prone to overfitting than single trees, and provides internal measures of feature importance.
  - Built a Random Forest Regressor with 100 estimators and a fixed random seed for reproducibility.

Evaluation Metrics

- **Mean Absolute Error (MAE):** The average absolute difference between predicted and actual prices—easy to interpret in dollars.
- **Root Mean Squared Error (RMSE):** Like MAE but penalizes larger errors more heavily, highlighting models that occasionally make big mistakes.
- **R^2 (Coefficient of Determination):** The proportion of variance in listing prices explained by the model (with 1.0 being perfect, 0 meaning no better than simply using the average price).
- On average, the model's nightly-rate predictions differ from actual prices by about $97, with most errors falling under around $143. It captures roughly 37% of the observed variation in listing prices, demonstrating that core listing features like size, room type and review signals carry substantial pricing information. These results offer a solid foundation for guiding hosts on general pricing trends, with room to refine precision through additional data and tuning.

Anomaly Detection Using Isolation Forest

This stage uses an unsupervised machine learning approach to detect potentially fake reviews. The model selected for this task is the Isolation Forest, which is well-suited for identifying anomalies in high-dimensional data without requiring labeled examples.

Model Overview

Isolation Forest works by isolating data points through random splits. Anomalous points — such as fake reviews — tend to be isolated faster because they differ significantly from the majority. This characteristic makes the model particularly effective in flagging outliers in review data.

Assumed Contamination Rate

The model is configured with an assumed contamination rate of 5%. This means it expects about 5% of the reviews to be potentially fake, helping it set a threshold for what constitutes an anomaly.

Input Features

The model relies on a set of engineered features that capture both textual characteristics and behavioral patterns of reviews:

- **Review Length**: Unusually short or excessively long reviews may be suspicious.
- **Punctuation Ratio**: Overuse of punctuation might indicate unnatural or exaggerated language.
- **Stopword Ratio**: Abnormal patterns in common word usage can signal automated or manipulated content.
- **Average Word Length**: Unusual word lengths might suggest synthetic text generation.
- **Repetition Score**: A high number of repeated words or phrases often occurs in templated or spammy content.
- **Time Gap Between Reviews**: Irregular or highly clustered timing of reviews on the same listing can reflect coordinated activity.

Output Interpretation

After processing the data, the model assigns a score to each review, indicating whether it is considered an anomaly. Reviews flagged as anomalous are labeled as suspected fakes, while others are treated as genuine. This results in a binary indicator for each review, allowing further analysis or filtering based on suspected authenticity.

AI Chatbot

The chatbot is delivered as a simple web interface where users type questions or select options and immediately see data-driven answers. Under the hood, it relies on three main libraries:

LangChain

- Provides the “Pandas DataFrame agent,” which connects the language model to your in-memory DataFrames and translates natural-language questions into the exact filtering, grouping, and aggregation operations needed.

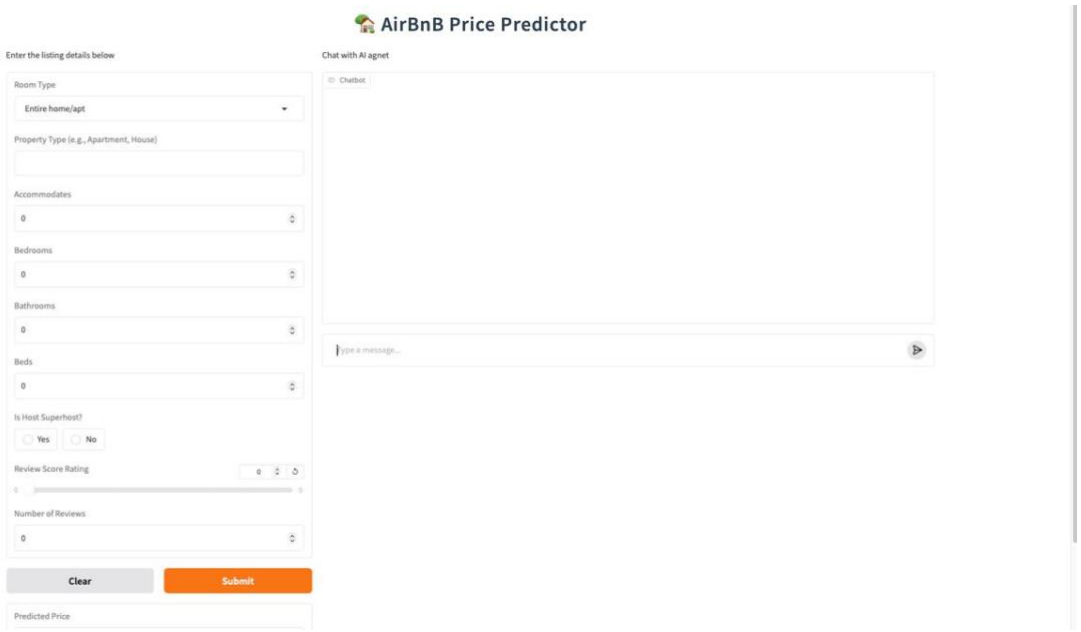
OpenAI (GPT-4o-mini)

- Powers the agent's understanding of intent and language—identifying what the user is asking for (e.g. "average price," "top-rated listings," "revenue trends") and extracting relevant entities (like room type or location).

Gradio

- Wraps everything in a lightweight web UI:
  - A chat panel for free-form questions handled by the DataFrame agent,
  - A form-based interface for structured price predictions.
- No heavy frontend code is needed—Gradio components automatically render inputs, outputs, and the chat box.

When someone opens the Gradio app, their message is sent to the LangChain agent, which uses GPT-4o-mini to parse it, runs the corresponding pandas queries on the listings/reviews data, and returns a clear, human-readable response—all in real time.



Recommendations for Stakeholders

Based on the data analysis and modeling of Airbnb listings in Austin, several actionable insights can help stakeholders—such as hosts, platform managers, and data teams—make informed decisions:

1. **Optimize Listings Based on High-Impact Features**
   Hosts should prioritize enhancing features that significantly influence price and revenue, such as number of bedrooms, guest capacity, and review scores. Investing in cleanliness and guest experience to improve review ratings can lead to higher occupancy and pricing.
2. **Encourage Hosts to Become Superhosts**
   Superhost status correlates with higher occupancy and revenue. Airbnb could incentivize hosts to maintain high standards to achieve or retain this status, improving overall platform performance.
3. **Promote Instant Bookable Listings**
   Listings that are instantly bookable tend to have higher occupancy rates. Encouraging more hosts to enable this feature may enhance booking volume and user experience.
4. **Use Price Prediction Tools for Dynamic Pricing**
   The developed machine learning model provides reasonably accurate price predictions. Airbnb could integrate such models into their platform to guide hosts on optimal pricing, helping them stay competitive while maximizing revenue.
5. **Detect and Moderate Anomalous Reviews**
   The anomaly detection model using Isolation Forest helps identify potentially fake or manipulated reviews. Implementing this system can improve trust and authenticity on the platform, enhancing guest confidence.
6. **Leverage AI Chatbots for Data-Driven Support**
   The AI chatbot developed in this project can be deployed to assist both hosts and guests in real time by answering data-related queries and offering personalized recommendations, reducing support overhead.
7. **Monitor and Manage Outliers**
   Listings with unusually high prices and low review counts should be further analyzed, as they may distort market perception or reflect mispriced units. Tools can be developed to alert hosts or admins to such anomalies.

Group Member Contribution:

Aashini Bala – Data Collection, Validation and Data Cleaning

Nashra Ali – Documentation

Gokul Govindasamy Sutharsan – Exploratory Data Analysis, Building ML model for price prediction and Anomaly Detection and Building Chatbot and UI.

References

Langchain create_pandas_dataframe_agent -
https://python.langchain.com/docs/integrations/tools/pandas/

Book – ***Hands on Large Language Models –*** *Jay Alammar & Maarten Grootendorst*