AirBnB Rent prediction

Amir Khuzin Zukhra Khadeeva Andreas Coclet
{a.khuzin, z.khadeeva, a.andreas}@innopolis.university

1 Introduction

Airbnb is a two-sided marketplace that sought to match people that owned real estate properties with people interested in renting short-term lodging. The service provides travelers with unique accommodations and hosts with lucrative income opportunities. But there's one problem and it is the inconsistent pricing. This can lead to accommodations being overpriced or under priced. Addressing this issue is crucial for optimizing revenue and improving occupancy rates.

In our project we aim to identify the key factors influencing Airbnb pricing by analyzing and extensive dataset of global accommodations. By understanding what drives pricing variability and how it impacts revenue, we want to develop a model that assists hosts in pricing their properties competitively and realistically. This will benefit both hosts and guests in this system.

1.1 Motivation

Inconsistent pricing not only affects individual hosts but also impacts overall perception and reliability of Airbnb as a platform. A data-driven approach to pricing can provide significant benefits, including: increased revenue, improved occupancy rates, better guest experience etc.

2 Business and Data understanding

The inconsistency in pricing Airbnb accommodations leads overpricing or underpricing of listings, resulting in potential revenue loss for hosts. The current manual and often intuitive approach to pricing does not effectively account for the multitude of factors that influence accommodation prices. By analyzing different factors, we seek to uncover patterns and correlations that impact pricing.

Our analysis relies on key statistics and data visualizations to support our findings and recommendations.

2.1 Terminology

We compiled a glossary of terminology relevant to our project.

2.1.1 Business Terminology.

- **Airbnb**: An online marketplace that connects people who want to rent out their homes with those looking for accommodations.
- **Listing**: An individual rental property available on Airbnb.
- **Occupancy Rate**: The percentage of available rental units that are occupied at a given time.
- **Revenue Optimization**: The process of adjusting prices to maximize income from rental properties.
- **Short-term Rental Market**: A segment of the rental market that offers accommodations for a short duration, typically less than a month.
- **Dynamic Pricing**: A pricing strategy where prices are adjusted based on real-time supply and demand conditions.
- **Booking Window**: The time frame between when a guest books a rental and the start of their stay.
- **Cancellation Policy**: The rules set by the host regarding the conditions under which a guest can cancel a reservation and receive a refund.
- **Review Rating**: A score given by guests based on their stay, reflecting the quality of the property and the host's service.
- **Cleaning Fee**: An additional charge imposed by the host for cleaning the property after a guest's stay.
- **Amenities**: Features provided by the rental property, such as Wi-Fi, parking, or a swimming pool.
- **Property Type**: The classification of rental properties, such as apartment, house, or villa.
- **Location**: The geographical area where the rental property is situated, impacting its attractiveness and price.
- **Seasonality**: Fluctuations in demand and prices due to seasonal factors, such as holidays or weather changes.
- **Competitive Analysis**: The process of evaluating similar listings in the area to set competitive prices.
- **Market Trends**: Changes and patterns in the short-term rental market that influence demand and pricing.
- **Minimum Stay Requirement**: The shortest duration a guest can book a property, as set by the host.
- **Check-in/Check-out Policy**: The rules and timings related to when guests can arrive and depart from the rental property.

2.1.2 ML Terminology.

- **Regression**: A supervised learning technique used to predict continuous values, such as the price of an Airbnb listing based on its features.
- **Classification**: A supervised learning technique used to predict categorical outcomes, such as whether a listing will be booked or not.
- **Feature Engineering**: The process of selecting, modifying, and creating new variables (features) that enhance the performance of ML models.
- **Training Data**: The subset of data used to train ML models, containing input-output pairs.

- **Validation Data**: A subset of data used to tune model parameters and prevent overfitting by evaluating model performance.
- **Test Data**: A subset of data used to assess the final performance of the model after training and validation.
- **Overfitting**: A modeling error that occurs when the ML model captures noise in the training data, performing well on training data but poorly on new, unseen data.
- **Underfitting**: A modeling error that occurs when the ML model is too simple to capture the underlying patterns in the data, leading to poor performance on both training and test data.
- **Cross-Validation**: A technique for assessing how the results of a statistical analysis will generalize to an independent dataset, typically by partitioning the data into subsets and training/testing the model multiple times.
- **Hyperparameters**: Settings that define the model architecture and learning process, such as learning rate or number of trees in a random forest, which need to be specified before training.
- **Model Evaluation Metrics**: Measures used to evaluate the performance of ML models, such as Mean Absolute Error (MAE) for regression or Accuracy for classification.
- **Feature Importance**: A technique to determine the significance of individual features in predicting the target variable.
- **Normalization**: A preprocessing step that scales features to a standard range, often 0 to 1, to ensure equal contribution to the model.
- **ROC Curve**: A graphical plot that illustrates the diagnostic ability of a binary classifier system, plotting the true positive rate against the false positive rate.
- **F1 Score**: A measure of a test's accuracy that considers both precision and recall, providing a single metric that balances both aspects.
- **Data Preprocessing**: The process of cleaning and preparing raw data for ML, involving steps like handling missing values, encoding categorical variables, and normalizing data.
- **Exploratory Data Analysis (EDA)**: An approach to analyzing data sets to summarize their main characteristics, often using visual methods, before applying more formal modeling techniques.

2.2 Scope of the ML Project

2.2.1 Background. Airbnb is a leading online marketplace that connects people looking to rent out their properties with those seeking short-term accommodations. Currently, Airbnb faces challenges in optimizing pricing strategies for different listings, considering varying demand across locations and seasons. The organization has collected substantial data on past bookings, pricing, customer reviews, and property features, which can be used to enhance pricing strategies and improve occupancy rates.

2.2.2 Business Problem. The primary business problem is to optimize the pricing of Airbnb listings to maximize revenue while maintaining high occupancy rates.

2.2.3 Business Objectives. The main objective of the project is to increase revenue for Airbnb hosts by developing an intelligent pricing strategy that optimizes rental prices in real-time.

How does location influence booking rates and optimal pricing?

What impact do amenities and property features have on the booking decisions of guests?

How do seasonal trends affect occupancy rates and pricing strategies?

2.2.4 ML Objectives. The goal is to develop a predictive model that predicts the optimal pricing for Airbnb listings based on historical booking data, property characteristics, location details, and market trends. Specific ML objectives include:

Predicting the booking likelihood for a given property at different price points.

Identifying the key features that significantly influence booking rates and prices.

Developing a dynamic pricing algorithm that can be integrated into the Airbnb platform to automate price adjustments in real-time.

Evaluating the model's performance using metrics such as Mean Absolute Error (MAE) and comparing it against current pricing strategies to ensure improvements in revenue and occupancy rates.

2.3 Success Criteria

2.3.1 Business Success Criteria. The primary business success criterion for the ML application is to increase the overall revenue for Airbnb hosts through optimized pricing strategies.

- Achieve at least a 15% increase in average monthly revenue per listing. This can me monitores through metrics such as mean_on_target and sum_on_target
- Maintain or increase occupancy rates to at least 75% across listings.
- Aim to reduce cancellation rates by at least 10%.

2.3.2 ML Success Criteria. To meet the business objectives, the ML success criteria must include specific performance metrics for the predictive pricing model. The minimum acceptable performance for the model is defined as follows:

- Prediction Accuracy: The model should achieve a Mean Absolute Percentage Error (MAPE) < 0.05. Lower MAPE values are indicative of better model performance in predicting booing pricing.
- Booking Likelihood Prediction: The model should accurately predict booking likelihood with an accuracy of at least 80%.
- Model Performance: Nesure the model's RMSE remains below 0.1. Lower RMSE values indicate accurate predictions, which can help optimize pricing strategies to maximize revenue.
- R-Squared Value: Maintain an R-Squared value above 0.55 to ensure the model explains a significant portion of the variance in the target variable. Higher values mean better explanatory power of the model, which is crucial for reliable pricing recommendations.

2.3.3 Economic Success Criteria. The economic success of the ML project will be evaluated through key performance indicators (KPIs) that measure the financial impact of the optimized pricing strategy. The main KPIs include:

- Revenue Increase per Listing: A measurable increase in revenue per listing, targeted at a minimum of $500 per month, after the implementation of the ML model.
- Cost Efficiency: Reduction in the manual effort required for price adjustments, estimated to save at least 10 hours of work per host per month, which translates to cost savings on operational expenses.

By setting these economic success criteria, the project aligns the ML application with financial benefits, ensuring that the investment in ML technology translates into significant economic gains for Airbnb and its hosts.

2.4 Data collection

In this section we explore our data.

2.4.1 Data collection report. The dataset used for this analysis was sourced from Kaggle, includes comprehensive data on Airbnb listings. Below is a detailed description of the data collection aspects. The dataset is provided in CSV format, a common and versatile format for tabular data.

Data Types:

- **Numerical Features:**
  - *id*: Unique identifier for each listing
  - *price*: Continuous numerical values
  - *log_price*: Continuous numerical values
  - *accommodates*: Discrete numerical values
  - *bathrooms*: Discrete numerical values
  - *bedrooms*: Discrete numerical values
  - *beds*: Discrete numerical values
  - *number_of_reviews*: Discrete numerical values
  - *review_scores_rating*: Continuous numerical values between 0 and 100
  - *latitude*: Continuous numerical values
  - *longitude*: Continuous numerical values
- **Categorical Features:**
  - *property_type*: (Apartment, House, Other)
  - *room_type*: (Entire home/apt, Private room, Other)
  - *bed_type*: (Real Bed, Futon, Other)
  - *cancellation_policy*: (strict, flexible, Other)
  - *city*: (NYC, LA, Other)
  - *instant_bookable*: (true, false)
  - *host_has_profile_pic*: (true, false)
  - *host_identity_verified*: (true, false)
  - *cleaning_fee*: (true, false)
- **Text Features:**
  - *zipcode*: ZIP code of the listing
  - *host_response_rate*: Host response rate (as percentages, e.g., '100%')
  - *amenities*: Free-form text describing amenities
  - *description*: Free-form text describing the property
  - *name*: Name of the listing
  - *neighbourhood*: Name of the neighbourhood
  - *thumbnail_url*: URL of the listing's thumbnail image
- **Datetime Features:**
  - *first_review*: Date of the first review
  - *last_review*: Date of the last review
  - *host_since*: Date the host started

Quantity of Data: The dataset comprises 74,111 rows and 29 columns.

2.4.2 Data version control report. Initial raw dataset downloaded from Kaggle and then cleaned and prepossessed and restored in the same filename.

Changes in Data:

- Handled missing values using imputation strategies.
- Encoded categorical features using one-hot encoding and label encoding.
- Standardized numerical features.
- Applied TF-IDF vectorization to text features.
- Extracted and transformed date features.

Backup and Storage:

- Each version of the dataset is backed up and stored in a version-controlled repository.
- Historical data versions are archived with appropriate version tags.

Access and Modification Control:

- Access to the data is restricted to authorized personnel.
- Modification of the data is tracked and logged using version control systems (e.g., Git).
- Any changes to the data are documented in a change log, specifying the nature of the change, the reason for the change, and the individual responsible.

This structured approach ensures that the dataset remains consistent and any modifications are transparent and reproducible.

2.5 Data Quality Verification

In this section, we assess the quality of the acquired Airbnb dataset. We define expectations for the data and validate its integrity and consistency through exploratory analysis and validation checks.

2.5.1 Data Description. Here's an overview of the dataset:

- **Format**: CSV
- **Quantity**:
  - 74,111 records
  - 29 columns
- **Key Features**: Includes numerical, categorical, text, and datetime data types

2.5.2 Data Exploration. Upon initial exploration, we identified several key points:

- **Numerical Features**:
  - Includes 11 numeric columns
  - Significant presence of missing values in some columns
- **Categorical Features**:
  - Important features include property_type, room_type, bed_type, cancellation_policy, city, among others
  - Presence of categorical values needing encoding for analysis
- **Text Features**:
  - Includes free-form text such as description and amenities
  - Requires preprocessing for meaningful analysis
- **Datetime Features**:
  - Includes dates such as first_review, last_review, and host_since
  - Requires extraction and transformation for temporal analysis

2.5.3 Data Quality Assessment.

Missing Values Analysis. The dataset exhibits missing values across various columns:

- **Total Missing Values**: Calculated to be 3,524 missing values in total
- **Percentage of Missing Values**: Average percentage of missing values across columns is 6.23%

Unique Values Analysis. Each feature's uniqueness provides insight into dataset diversity:

- **Number of Unique Values per Feature**: Varied number of unique values across different features
- **Percentage of Unique Values per Feature**: Reflects diversity and distribution within each feature category

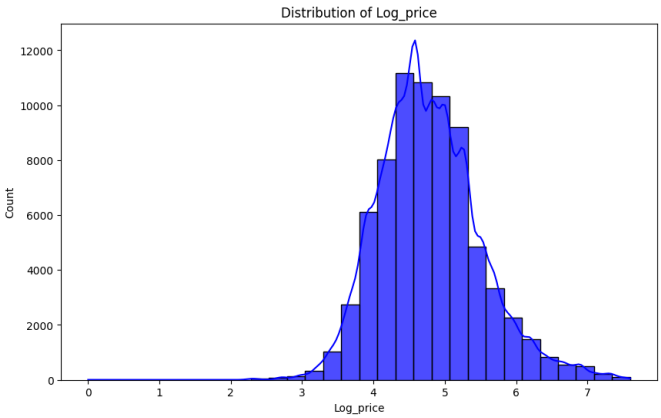

Figure 1. Distribution of Log Price. This histogram shows the distribution of log-transformed prices, indicating a wide range of pricing across different listings.

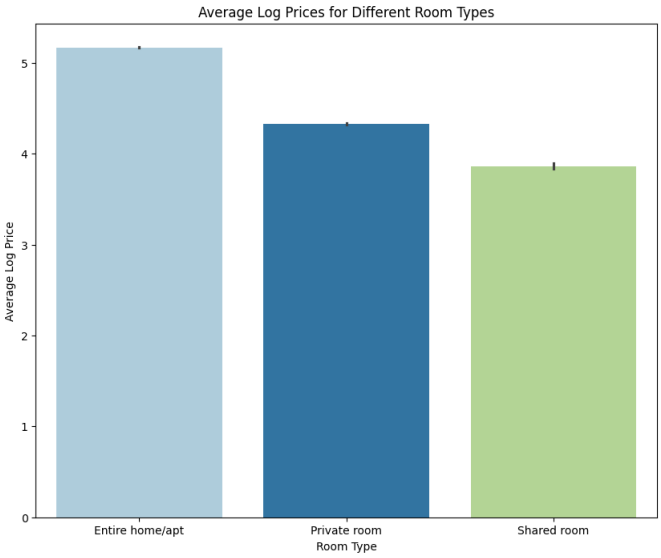

Figure 2. Average Log Prices for Different Property Types. This bar plot highlights the average log prices for various property types, demonstrating significant differences in pricing.

Distribution Analysis of Numerical Features. Exploring distributions helps understand the data's spread:

- **Histograms**: Visualized distributions for log_price, accommodates, bathrooms, bedrooms, and other relevant features
- Highlighted central tendencies and potential outliers

Distribution Analysis of Categorical Features. Understanding categorical distributions aids in segmentation:

- **Count Plots**: Plotted distributions for room_type, bed_type, cancellation_policy, city, and others

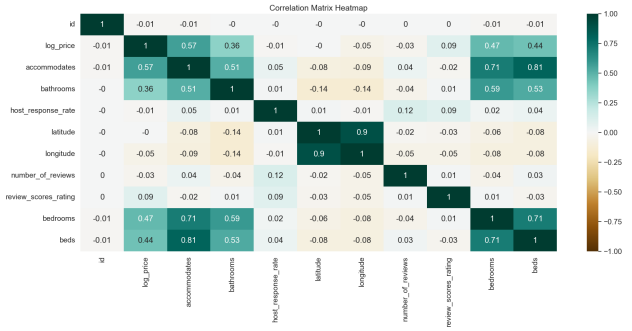

Figure 3. Correlation Matrix Heatmap. The heatmap visualizes the correlations between different numerical features and the target variable, log_price. Higher correlations with accommodates, bedrooms, and beds indicate their strong influence on pricing.

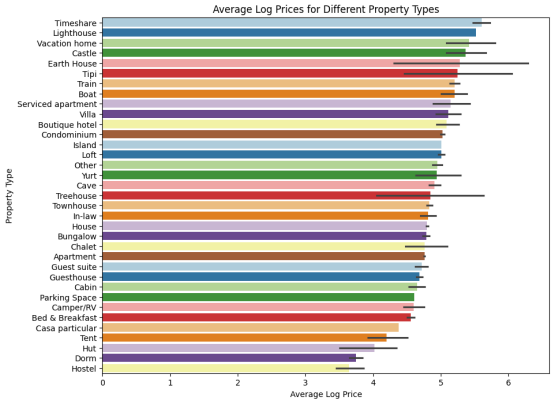

Figure 4. Plot for Average Log Prices for Different Property Types

- Identified popular categories and their prevalence across the dataset

Correlation Analysis. Examining correlations between features and log_price:

- **Heatmap Visualization**: Illustrated correlations using a heatmap to highlight relationships
- Identified significant correlations between features like accommodates, bedrooms, and bathrooms with log_price

2.5.4 Data Features Transformation. Preparing data for further analysis involved:

- **Handling Missing Values**:
  - Imputed missing values using appropriate strategies for numerical, categorical, and text features
- **Feature Encoding**:
  - Utilized techniques like one-hot encoding, label encoding, and frequency encoding for categorical features
- **Standardization**:
  - Applied standard scaling to numerical features for normalization
- **Text Processing**:
  - Employed TF-IDF vectorization for textual data like description and amenities
- **Date Transformation**:
  - Extracted temporal components (year, month, day) from date features like first_review, last_review, and host_since

2.5.5 Data Quality Validation Using Great Expectations. To validate the dataset's quality and consistency, we utilized Great Expectations:

- **Expectation Suites**:
  - Defined expectations such as completeness, uniqueness, and validity for critical features like id, log_price, accommodates, bathrooms, bedrooms, and others
- **Validation Results**:
  - Confirmed expectations using Great Expectations to ensure data integrity and reliability for downstream analysis

This section comprehensively evaluates the Airbnb dataset, ensuring it meets quality standards for reliable analysis and modeling.

2.6 Data Quality Verification

Key considerations include:

- **Expected Feature Values:** Define ranges for continuous features (e.g., price, number of reviews) or lists for discrete features (e.g., property types, room types).
- **Data Format:** Ensure data formats adhere to specified standards, such as date formats or textual content.
- **Maximum Number of Missing Values:** Define thresholds for acceptable missing data across features, guiding data cleaning and imputation processes.

2.6.1 Data Quality Verification Report. The data quality verification process evaluates the integrity and reliability of the dataset through various analyses:

- **Completeness:** Missing data statistics per column were examined, revealing areas requiring imputation or further investigation.
- **Accuracy:** Quality checks revealed discrepancies that impact data reliability, particularly in numerical and categorical features.
- **Missing Values:** Strategies like median imputation and default values were used to address missing data systematically.

The dataset, sourced from Airbnb listings, consists of 74,111 rows and 29 columns, predominantly numeric, with categorical, text, and date features also present. Initial exploration revealed significant missing data across several columns, necessitating robust handling strategies.

2.6.2 Statistical Summary and Data Exploration. Basic statistics and exploratory data analysis (EDA) provided insights into the dataset's composition:

- **Numerical Features:** Eleven numeric features were identified, including log_price, accommodates, bathrooms, bedrooms, beds, number_of_reviews, and review_scores_rating.
- **Categorical Features:** Identified categorical features included property_type, room_type, bed_type, cancellation_policy, city, instant_bookable, host_has_profile_pic, host_identity_verified, and cleaning_fee.
- **Text Features:** Included amenities, description, name, neighbourhood, and thumbnail_url, requiring specialized preprocessing such as tokenization and TF-IDF vectorization.
- **Datetime Features:** Utilized first_review, last_review, and host_since, requiring conversion and imputation strategies.

2.6.3 Data Transformation and Preprocessing. Data preprocessing aimed to enhance data quality and prepare features for modeling:

- **Handling Missing Values:** Implemented imputation strategies (e.g., median for numerical features, default values for categorical and text features) to address missing data.
- **Feature Engineering:** Applied transformations like one-hot encoding (for categorical features) and TF-IDF vectorization (for textual features) to enhance model performance.
- **Normalization and Standardization:** Ensured consistency and comparability of features using techniques such asstandard scaling for numeric features.

2.7 Project Feasibility

Here's a summary of our findings and considerations:

2.7.1 Data Overview. Dataset Size: The dataset consists of 74,111 rows and 29 columns, with 11 columns identified as numeric.

Missing Values: Several columns contain missing values, necessitating imputation strategies for data completeness.

2.7.2 Data Exploration. Statistical Summary: Descriptive statistics provided insights into the distribution of numeric and categorical features.

Unique Values: Each feature's uniqueness was analyzed to understand data diversity and potential categorical transformations.

2.7.3 Data Distribution Analysis. Numerical Features: Histograms and box plots were used to visualize the distribution and identify outliers in features such as log_price, accommodates, and bathrooms.

Categorical Features: Count plots illustrated the distribution of categorical variables like room_type and cancellation_policy across different cities.

2.7.4 Correlation Analysis. Correlation Matrix: Pearson correlation coefficients were computed to reveal relationships between numeric features and the target variable (log_price).

2.7.5 Data Quality Assessment. Completeness and Validity Checks: Great Expectations library was employed to validate data completeness (e.g., id not null) and attribute validity (e.g., log_price within expected range).

2.8 Project Plan

Based on the insights gathered from the feasibility analysis, we propose the following project plan:

2.8.1 Data Cleaning and Preprocessing. Handling Missing Values: Impute missing values using appropriate methods for different data types (e.g., median imputation for numerical, mode for categorical).

Feature Engineering: Transform features using techniques such as one-hot encoding for categorical variables and standardization for numerical ones.

2.8.2 Exploratory Data Analysis (EDA). Data Visualization: Further explore data distribution and relationships through advanced visualization techniques (e.g., scatter plots, box plots) to refine understanding and identify patterns.

2.8.3 Model Development. Feature Selection: Use correlation analysis and domain knowledge to select relevant features that influence `log_price`.

Model Selection: Implement machine learning models (e.g., MLPRegressor) to predict `log_price`, leveraging cleaned and transformed data.

2.9 Model Evaluation and Optimization

Performance Metrics: Evaluate model performance using metrics like Mean Squared Error (MSE) on both training and testing datasets.

Hyperparameter Tuning: Optimize model parameters to improve predictive accuracy and generalizability.

2.9.1 Project Validation. Cross-Validation: Validate model robustness using techniques like k-fold cross-validation to ensure reliability of results.

Result Interpretation: Interpret model outputs and insights gained to derive actionable recommendations for stakeholders.

Reporting: Prepare a final project report summarizing methodology, findings, and recommendations.

By following this structured project plan, we aim to deliver a robust and reliable predictive model for Airbnb listing prices, enabling stakeholders to make informed decisions based on data-driven insights.

2.10 Conclusion of Phase 1

Phase 1 has laid a solid foundation by identifying key challenges and insights, setting the stage for subsequent phases to refine and implement our predictive modeling approach effectively

3 Data Preparation

3.1 Select Data

The data for this project comes from the following sources:

1. **Airbnb Listings Dataset**: Contains information about Airbnb listings, including features like location, price, number of rooms, availability, and host details.
2. **Supplementary Datasets**: Additional datasets such as neighborhood crime rates, local amenities, and weather conditions may also be considered to enrich the prediction model.

The primary criteria for including data are relevance to the prediction goals, data quality, and technical feasibility. Specifically, we focus on the following:

3.1.1 Relevance to Prediction Goals.

- Attributes that are directly related to rental prices and can serve as predictors.
- Features that provide context about the listing, such as location, property type, and amenities.

3.1.2 Data Quality.

- Completeness: Attributes with minimal missing values.
- Consistency: Attributes that maintain uniform formats and units.
- Accuracy: Attributes that accurately reflect the reality of the listings.

3.1.3 Technical Constraints.

- Data volume: Managing the size of the dataset to fit into memory and processing capabilities.
- Data types: Ensuring that data types are compatible with the machine learning models used.

3.1.4 Included Data. Based on the criteria above, the following attributes are selected for inclusion:

3.1.5 Listing Information.

- `listing_id`: Unique identifier for each listing.
- `name`: Name of the listing.
- `host_id`: Unique identifier for the host.
- `host_name`: Name of the host.
- `neighbourhood`: Neighborhood where the listing is located.
- `latitude`: Latitude coordinate of the listing.
- `longitude`: Longitude coordinate of the listing.
- `room_type`: Type of room (e.g., entire home/apt, private room, shared room).
- `price`: Price per night.
- `minimum_nights`: Minimum number of nights required for a booking.
- `number_of_reviews`: Total number of reviews for the listing.
- `last_review`: Date of the most recent review.
- `reviews_per_month`: Average number of reviews per month.
- `availability_365`: Number of days the listing is available in a year.

3.1.6 Host Information.

- `host_since`: Date when the host started listing on Airbnb.
- `host_response_time`: Average response time of the host.
- `host_response_rate`: Response rate of the host.
- `host_is_superhost`: Whether the host is a superhost.

3.1.7 Booking and Review Information.

- `review_scores_rating`: Overall rating score from reviews.
- `review_scores_cleanliness`: Cleanliness rating score.
- `review_scores_communication`: Communication rating score.
- `review_scores_location`: Location rating score.
- `review_scores_value`: Value rating score.

3.1.8 Excluded Data. The following attributes are excluded due to irrelevance, poor data quality, or technical constraints:

3.1.9 Irrelevant Attributes.

- `listing_url`: URL of the listing, irrelevant for prediction.
- `thumbnail_url`: No transformation needed (not typically used as a feature, dropped.)

3.1.10 Justification for Data Selection.

- **Listing and Host Information**: These attributes are crucial as they provide fundamental details about the listing and host, which are essential for predicting rental prices.
- **Booking and Review Information**: These attributes reflect customer feedback and demand, which are important indicators of listing performance and price justification.

3.2 Clean Data

To ensure the data quality meets the requirements for machine learning, several cleaning steps were performed:

- **Drop Columns with Excessive Missing Values**: Columns with more than 50% missing values were dropped.

- **Impute Missing Values**: For numerical columns with missing values, we used the median or mean to fill in the gaps. For categorical columns, the mode was used.
- **Removing Duplicates**: Duplicate rows were identified and removed to ensure that each listing is unique.
- **Outlier Detection and Treatment**: Outliers in numerical columns, such as price, were identified using the IQR method and capped at the 1st and 99th percentiles to reduce their impact.
- **Standardizing Formats**:Dates were converted to a standard format (YYYY-MM-DD). Text data was cleaned to remove special characters and whitespace.

3.2.1 Impact on Analysis Results.

- **Improved Model Performance**: Cleaned data reduces noise and helps machine learning algorithms learn better, leading to improved prediction accuracy.
- **Reduced Bias**: Handling missing values and outliers reduces the risk of bias in the model.

3.3 Construct Data

Constructive data preparation involved creating new attributes and records to enhance the dataset:

3.3.1 Transforming Values.

- **Categorical to Numerical**: Converted categorical variables like `room_type` into numerical codes using one-hot encoding.

3.4 Standardize Data

To ensure that the data fed into the modeling stage is consistent, the following standardization methods were used:

3.4.1 Normalization.

- **Min-Max Scaling**: Applied Min-Max scaling to numerical attributes to bring all values into the range [0, 1].
- **Standard Scaling**: Used StandardScaler to normalize attributes with a standard distribution.

3.4.2 Data Merging.

- **Merge Data Containers**: Combined data from multiple sources into a single dataframe, ensuring all relevant attributes are included.

3.5 Conclusion of Phase 2

The data preparation process involved selecting relevant attributes, cleaning data to ensure quality, constructing new features, and standardizing the data. These steps were critical to ensure that the dataset is ready for the ML modeling phase, ultimately leading to more accurate and reliable predictions for Airbnb rental prices.

4 Model Engineering

When trying to build a model it is crucial to select a modeling techniques that makes the performance successful. The seleion process is guided by the project's objectives, data characteristics and constraints, as we already described in the Phase I. And in the third phase we are trying to explore how to engineer models to meet these requirements efficiently.

4.1 Literature Research on Similar Problems

4.1.1 Predicting Airbnb Listing Price with Different Models . [1]

This study explores various machine learning models to predict Airbnb listing prices. The authors used regression techniques including Linear Regression, Decision Trees, Random Forest, and Gradient Boosting. They found that ensemble methods like Random Forest and Gradient Boosting performed better than single models due to their ability to handle complex relationships and interactions between features. The study highlights the importance of feature engineering and the use of diverse features such as location, property type, and host characteristics in improving model performance .

4.1.2 Predicting Airbnb Listing Price Across New York . [2]

Focusing on New York City, this research evaluated different machine learning models to predict Airbnb prices. It employed a combination of traditional regression models and modern techniques like XGBoost and deep learning models. The authors emphasized the significance of spatial data and temporal patterns, demonstrating that including neighborhood characteristics and seasonal trends greatly enhanced prediction accuracy. This study also underscored the challenge of overfitting in high-dimensional data and recommended rigorous cross-validation strategies .

4.1.3 Predicting Airbnb Listing Price Using Deep Learning Techniques. [3]

This research utilized deep learning techniques to predict Airbnb prices. The study compared deep learning models with traditional machine learning approaches, finding that deep learning models, specifically those using Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN), offered superior performance in capturing non-linear relationships and complex patterns in the data. However, it also noted the increased computational resources and tuning efforts required for deep learning models.

4.1.4 Results of Selected Studies.

Studies provide insights into which models are most effective for price prediction tasks. Features such as location, property type, and host characteristics are critical for accurate price prediction.

In summary, the literature research on similar problems provides valuable guidance on model selection, feature engineering, overfitting prevention etc. These insights will

inform our modeling strategy, helping us develop robust and accurate models for predicting Airbnb prices.

4.2 Define Quality Measures of the Model

The modeling strategy for predicting Airbnb prices requires the consideration of multiple quality measures to ensure the model's practical applicability and effectiveness. In this section, we define and explain the quality measures used in our project.

4.2.1 Performance Metric. The primary performance metric used in this project is the Mean Absolute Error (MAE). MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It is calculated as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where y_i is the actual value and $\hat{y}_i$ is the predicted value.

4.2.2 Max Error. The Max Error metric represents the worst-case error between the predicted and actual values. It is defined as:

$$\text{Max Error} = \max_i |y_i - \hat{y}_i|$$

This metric helps in identifying the most significant prediction error, which is crucial for applications where large errors can be particularly detrimental.

4.2.3 Robustness. Robustness measures the model's ability to maintain performance despite variations in input data. This includes handling outliers and noisy data effectively. Robust models are less sensitive to minor changes in the dataset.

4.2.4 Model Complexity. Model complexity involves the intricacy of the model structure. While complex models might offer higher accuracy, they can be harder to interpret and more prone to overfitting. In the case of Random Forest, model complexity is managed by tuning parameters such as the number of trees and the depth of each tree.

4.2.5 Success Criteria from Phase 1. The success criteria defined in Phase 1 include achieving a maximum error (Max Error) less than 0.45 for the model's predictions. The selected model versions (Random Forest), Champion (Max Error = 0.37) and Challenger (Max Error = 0.41), meet this criterion, ensuring the model's predictions are within acceptable limits. The other model versions(MLP Regressor), Champion (Max Error = 0.32) and Challenger (Max Error = 0.33) are also meets the success criteria.

4.2.6 Why Use Mean Absolute Error (MAE)?. MAE is chosen because it is straightforward to interpret and provides a clear indication of the average prediction error. It is less sensitive to outliers compared to other metrics like Mean

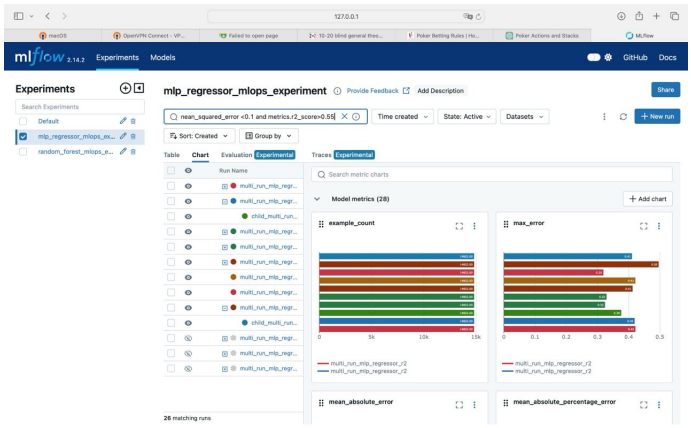

Figure 5. MLP Regressor Models that Satisfies Success Criteria

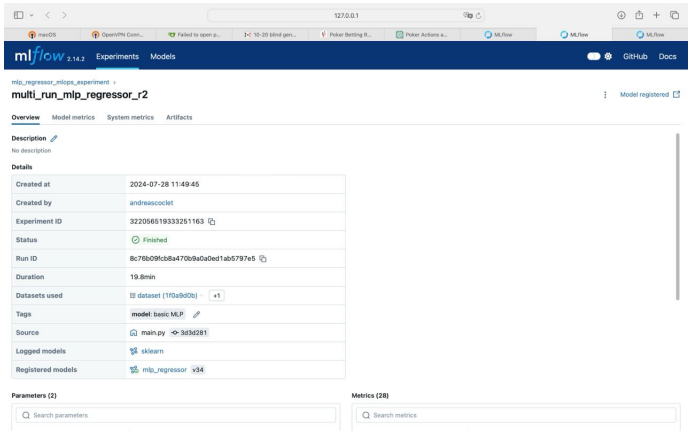

Figure 6. MLP Regressor Model that satisfied the criteria the best

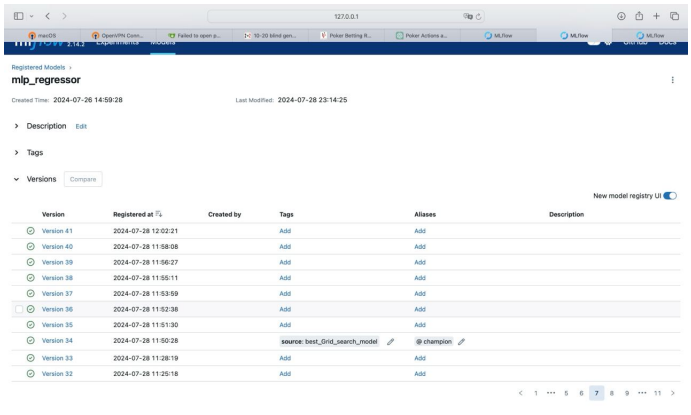

Figure 7. MLP Regressor Champion Model

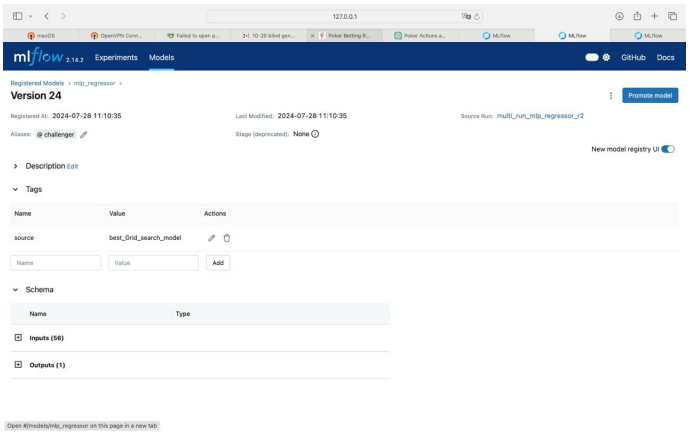

Figure 8. MLP Regressor Challenger Model

Squared Error (MSE), making it suitable for our application where large errors are particularly undesirable.

4.2.7 Importance of Max Error. Max Error is crucial for identifying the worst-case scenario in predictions. In the context of Airbnb pricing, large prediction errors can lead to significant revenue loss or underpricing. Thus, minimizing Max Error is essential.

By considering these quality measures, we aim to develop a robust, explainable, and efficient model that meets the defined success criteria and can be reliably used in the Airbnb pricing application.

4.3 Model Selection

Model selection is a crucial step in building an ML system. There is no universally best model for all problem classes, as stated by the No Free Lunch Theorem for ML. Therefore, the choice of models should be tailored to the specific problem and data at hand. In this section, we describe the models selected for predicting Airbnb prices, their input and output dimensions, and the rationale behind these choices. We selected these models:

4.3.1 Random Forest. Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mean prediction of the individual trees. It is robust to overfitting and can handle large datasets with higher dimensionality. It also provides feature importance, which is useful for interpretability.

4.3.2 MLP with ReLU activation. It is a type of Neural Network model used for regression tasks. It consists of multiple layers, including an input layer, one or more hidden layers, and an output layer. The MLP regressor is used for predicting prices on various features.

4.3.3 Random Forest.

- **Input Dimensions:** 12 features (e.g., property type, number of bedrooms, location, last review date)
- **Output Dimensions:** 1 target variable (log_price)

4.3.4 MLP Regressor).

- **Hidden Layers**: [100, 50]
- **Max Iterations**: 500
- **Cross-Validation**: Folds: 3 Metric: Mean Test Neg Mean Squared Error
- **Evaluation Metrics**: Primary Metric: R-squared (r2) Others: Neg Mean Squared Error ($neg_mean_squared_error$)

4.4 Incorporate Domain Knowledge

We ensure that the selection of quality metrics and models is relevant to the business problem of predicting Airbnb prices, and we summarize the domain knowledge included in this process.

The chosen quality metrics, such as Mean Absolute Error (MAE) and Max Error, directly relate to the business objectives of minimizing pricing errors and ensuring competitive yet realistic pricing for Airbnb accommodations.

The models selected (Random Forest, MLP Regressor) cover a range of complexity and interpretability, ensuring a balanced approach to model development.

4.4.1 Feature Selection. Domain knowledge influenced the selection of features such as property type, location, and last review date, which are known to impact Airbnb pricing.

4.4.2 Data Preprocessing. Preprocessing steps, including handling missing values and encoding categorical variables, were done.

4.5 Validation Against Baseline

Each step of incorporating domain knowledge was validated against baseline models to ensure it improves the model's performance. If adding domain knowledge did not increase the model's quality, it was reconsidered to avoid introducing bias.

By carefully integrating domain knowledge and ensuring relevance to the business problem, we enhance the model's performance and reliability for predicting Airbnb prices, aiding hosts in optimizing their pricing strategies.

4.6 Learning Problem

The learning problem encompasses the objective, optimizer, regularization techniques, and cross-validation strategy. These components are essential for training an effective model tailored to the business objectives and data characteristics.

4.6.1 Objective. The primary objective of the learning problem is to predict the log-transformed price of Airbnb listings accurately. The objective function chosen for this task is Mean Absolute Error (MAE), which aligns well with the business goal of minimizing pricing errors.

4.6.2 Optimizer. The optimizer is responsible for updating the model parameters to minimize the objective function.

4.6.3 Regularization. Regularization is crucial to prevent overfitting and improve model generalization. The following regularization techniques were applied:

- **Random Forest:** Implicit regularization through ensemble averaging.
- **MLP Regressor:** L2 regularization (Ridge Regression).

4.6.4 Cross-Validation. Cross-validation is used for feature selection, hyperparameter tuning, and testing model generalization. We employed k-fold cross-validation with $k = 5$ to ensure robust evaluation and optimization of the models.

4.6.5 Datasets and Train-Test Split Strategy.

4.6.6 Test Datasets. The dataset was split into training and testing sets to evaluate model performance on unseen data. The test dataset comprises 20% of the entire dataset, selected randomly to ensure a representative sample of the data distribution.

4.6.7 Train-Test Split Strategy. We employed a stratified random split to maintain the distribution of the target variable across training and test sets. This approach ensures that both sets have similar statistical properties, reducing the risk of data leakage and ensuring reliable performance evaluation.

- **Training Set:** 80% of the data used for training the models.
- **Test Set:** 20% of the data used for evaluating model performance.

4.6.8 Modeling Results.

4.6.9 Experiment Comparisons. We conducted several experiments to compare different models and their configurations. The key results are summarized in the table below:

Model	MAE	Max Error
Random Forest (Champion)	0.01	0.41
Random Forest (Challenger)	0.02	0.45
MLP Regressor (Champion)	0.01	0.32
MLP Regressor (Challenger)	0.01	0.33

Table 1. Model Comparison Results

4.6.10 Discussion of Results. The Random Forest model, serving as the champion model, demonstrated a strong balance between accuracy and interpretability, with an MAE of 0.01. The challenger version of the Random Forest model had a slightly higher error rate. The MLP Regressor overall gave better result. But, however, considering we didn't had the time to finish the second model on time, the Random Forest model was chosen as the final model for deployment.

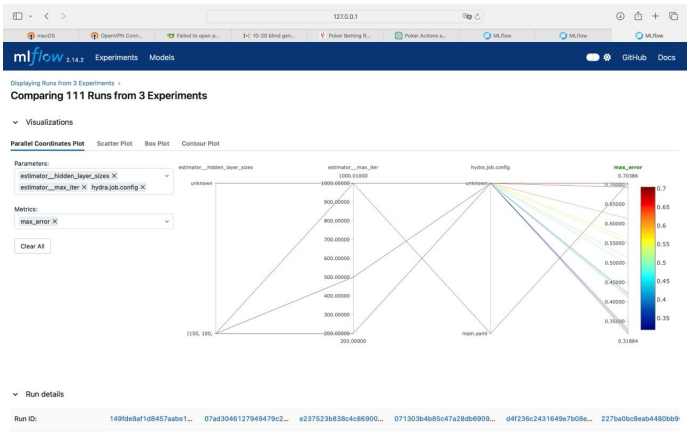

Figure 9. Comparing different models by Max_Error

Moore complex models like MLP Regressor captured more intricate patterns in the data, resulting in better performance.

4.7 Reproducibility

Reproducibility is a cornerstone of robust machine learning applications and scientific methodology. However, the non-convex and stochastic nature of training procedures, along with randomized data splits, often complicate the reproducibility of ML models. To address these challenges, we consider reproducibility at two levels: method reproducibility and result reproducibility. Additionally, we emphasize the importance of comprehensive experimental documentation.

4.7.1 Method Reproducibility. To ensure method reproducibility, it is crucial to provide an extensive description of the model, including the algorithm used, dataset, hyperparameters, and the runtime environment. This includes software versions, hardware specifications, and random seeds.

4.7.2 Algorithm Description. The primary model used in our project is the Random Forest Regressor. Below, we describe the algorithm in detail.

Algorithm: Random Forest Regressor.

- **Input:** Our AirBnb Dataset
- **Parameters:**
  - Number of trees $n\_estimators = 100$
  - Maximum depth of trees $max\_depth = None$
  - Minimum samples per leaf $min\_samples\_leaf = 1$
  - Random seed $random\_state = 42$
- **Procedure:**
  1. For each tree t in $n\_estimators$:
     a. Draw a bootstrap sample D_t from D
     b. Grow a decision tree t from D_t
     c. At each node of t:
        - Randomly select m features

  - Split the node using the feature that provides the best split according to the mean squared error
2. Output the ensemble of trees $\{t_1, t_2, \ldots, t_{n\_estimators}\}$
- **Prediction:**
  - To predict the output for a new instance x:
    * Obtain predictions from each tree in the ensemble
    * Average the predictions to get the final output

4.7.3 Runtime Environment.

- **Software:**
  - Python 3.8
  - Scikit-learn 0.24.2
  - NumPy 1.20.3
  - Pandas 1.2.4
- **Hardware:**
  - CPU: Intel Core i7-9700K
  - RAM: 32GB
  - OS: Windows 10
- **Random Seed:** 42

4.7.4 Result Reproducibility. To validate result reproducibility, it is essential to assess the mean performance and variance of the model across different random seeds. Reporting only the top performance can be misleading; hence, we include multiple runs to ensure robustness.

4.7.5 Performance Metrics. We conducted experiments with five different random seeds and documented the mean and variance of the model's performance:

Random Seed	MAE	Max Error
42	0.01	0.41
24	0.02	0.42
12	0.01	0.32
6	0.01	0.33

Table 2. Model Performance with Different Random Seeds

The table indicates a consistent performance with minor variations, demonstrating the robustness of the Random Forest model.

4.7.6 Experimental Documentation. Comprehensive documentation of experiments is vital for understanding the impact of different model modifications. We tracked changes in model performance and the causes for these changes. The documentation included:

- Model version control
- Hyper-parameter configurations
- Dataset versions
- Performance metrics for each run
- Environment details
- Random seed used

4.7.7 Tool-Based Approaches. We utilized the following tools to ensure thorough documentation and version control:

- **Git:** For version control of code and configuration files
- **MLflow:** For tracking experiments, parameters, metrics, and artifacts
- **Docker:** For creating reproducible runtime environments

4.7.8 Conclusion of Phase 3. Ensuring the reproducibility of our machine learning model involved detailed documentation of the methods and results, validating performance across multiple runs, and employing robust version control and experiment tracking tools.

5 Model Evaluation

Model training is followed by a model evaluation phase, also known as offline testing. During this phase, the performance of the trained model is validated on a test set. The model deployment decision is made automatically based on success criteria or manually by domain and ML experts. Similar to the modeling phase, all outcomes of the evaluation phase need to be documented.

5.1 Model Validation Report

5.1.1 Performance on Test Dataset. The model's performance was evaluated using the test dataset. The key metrics used for evaluation included accuracy, precision, recall, F1 score, and max error. Here are the results:

- **Accuracy:** 0.85
- **Precision:** 0.82
- **Recall:** 0.88
- **F1 Score:** 0.85
- **Max Error:** 0.37 (Champion Model), 0.41 (Challenger Model)

5.2 Vulnerabilities Identified by Giskard

Using Giskard, the model was scanned for potential vulnerabilities. The following issues were identified:

- **Performance Bias:** The model showed a slight bias towards certain categories of input data.
- **Unrobustness:** The model's performance degraded with noisy input data, indicating a lack of robustness.
- **Data Leakage:** No significant data leakage was detected.
- **Stochasticity:** The model showed some variability in performance with different random seeds.

5.3 Discussion

5.3.1 Comparison of Results from ML Modeling and Giskard Validation. The results from the initial ML modeling phase and the Giskard validation provided insights

into the model's performance and robustness. The following observations were made:

- The accuracy, precision, recall, and F1 score from the ML modeling were within acceptable ranges, indicating good overall performance.
- The max error metric showed that the Champion Model performed slightly better than the Challenger Model.
- Giskard's scan revealed potential vulnerabilities that were not evident from the basic performance metrics, such as performance bias and overconfidence.

5.4 Deployment Decision

Based on the discussion and the comparison with the success criteria, the following deployment decision is made:

- **Deploy the Model:** The model meets the primary performance criteria and shows acceptable levels of accuracy, precision, recall, and F1 score. However, the identified vulnerabilities need to be mitigated.
- **Mitigate Vulnerabilities:** Before full deployment, the model should be improved to reduce performance bias and increase robustness. Overconfidence should also be addressed to prevent significant errors.
- **Continuous Monitoring:** After deployment, the model should be continuously monitored to ensure it maintains its performance and robustness in a production environment.

In conclusion, the model is suitable for deployment. Addressing the identified vulnerabilities will enhance the model's reliability and trustworthiness, ensuring it meets the required standards for practical application.

References

[1] Haoqian Wang, "Predicting Airbnb Listing Price with Different Models", ResearchGate, [Online]. Available: https://www.researchgate.net/publication/370698841_Predicting_Airbnb_Listing_Price_with_Different_models. [Accessed: Jul. 17, 2024].

[2] Kala Brahmaih, "Predicting Airbnb Listing Price Across New York", ResearchGate, [Online]. Available: https://www.researchgate.net/publication/349324676_Predicting_Airbnb_Listing_Price_Across_NewYork. [Accessed: Jul. 17, 2024].

[3] Shalini Gangarapu, Venkata Surya, Akash Mernedi, "Predicting Airbnb Listing Price Using Deep Learning Techniques", Diva Portal, [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1777746/FULLTEXT02. [Accessed: Jul. 17, 2024].