ISOM 835 Final Project Report

Project Title: Predicting Airbnb Prices in London Using Machine Learning

Dataset: Airbnb Prices in European Cities (London Weekdays subset) from Kaggle

Student: Chandini Rajendra

Institution: Suffolk University

Course: ISOM 835 – Predictive Analytics and Machine Learning

Instructor: Hasan Arslan

Date: May 2025

1. Introduction & Dataset Description

The rise of Airbnb and short-term rentals has brought increasing competition among hosts to price their listings effectively. This project aims to apply machine learning to predict prices of Airbnb listings in London using a real-world dataset. Accurate predictions can help hosts optimize revenue and improve customer satisfaction.

Reasons for Dataset Selection

1. **Real-world relevance**: Airbnb pricing is a practical and relatable problem with clear business applications in tourism, hospitality, and real estate.
2. **Rich feature set**: Includes a mix of numeric (e.g., price, distance, person_capacity) and categorical (e.g., room_type, host_is_superhost) variables.
3. **Predictive potential**: Ideal for regression modeling, with opportunities to apply various algorithms and feature engineering techniques.
4. **Clean and publicly available**: The dataset was easily accessible via Kaggle and required minimal effort to prepare for analysis.

Challenges Encountered

1. **Skewed target variable**: realSum (price) had significant outliers and right-skew, impacting model performance.
2. **Missing values**: Although limited, required handling through row deletion or imputation.

3. **Feature limitations**: Some potentially important factors (e.g., number of reviews, date of listing, neighborhood) were not present.
4. **Model interpretability vs. performance**: Tree-based models performed better but were harder to explain than simpler models like Linear Regression.

Dataset Composition

- Records: 744 Airbnb listings from London
- Features: 16 input variables + 1 target variable (`realSum`)
- Target Variable: `realSum` – represents the total price for the listing

Rationale for Dataset Selection

This dataset offers both practical and technical value. It was selected due to its relevance in hospitality pricing and predictive analytics. The diversity of features such as location, room type, and host status makes it ideal for building and evaluating machine learning models.

Code to Load Dataset

```
import kagglehub
# Download latest version
path = kagglehub.dataset_download("thedevastator/airbnb-
prices-in-european-cities")
print("Path to dataset files:", path)
```

Choose subset – London_weekdays.csv

Example Code

```
from google.colab import files
uploaded = files.upload()
# Get the uploaded file name directly from the uploaded
dictionary
file_path = list(uploaded.keys())[0]
import pandas as pd
df = pd.read_csv(file_path)
df.head()
```

2. Exploratory Data Analysis (EDA)

EDA is the foundation for understanding a dataset. It includes visualization and summary statistics to discover trends, anomalies, and patterns.

Summary Statistics

We reviewed the shape, structure, and distribution of values. Summary statistics revealed a wide range of prices with some notable outliers.

Step 1: Data Overview

Code Example

```
print("Shape:", df.shape)
print("Columns:\n", df.columns.tolist())
df.info()
df.describe()
df.isnull().sum()
```

Findings:

- Shape: (n_rows, 18) after dropping Unnamed: 0.
- Columns include price (realSum), room_type, person_capacity, host_is_superhost, and distance features (dist, metro_dist).
- Missing values handled by dropping rows.
- room_type converted to categorical and host_is_superhost to binary (0/1).

Step 2: Price Distribution

Code Example

```
sns.histplot(df['realSum'], bins=50)
plt.title("Distribution of Airbnb Prices")
plt.xlabel("Price")
plt.ylabel("Frequency")
plt.show()
```

Insight:

- Prices are **right-skewed** with a long tail of expensive listings.
- May require **log transformation** for linear models.

Step 3: Room Type vs Price

Code Example

```
sns.boxplot(x='room_type', y='realSum', data=df)
plt.title("Price Distribution by Room Type")
plt.show()
```

Insight:

- Entire home/apt listings are **more expensive** on average than shared/private rooms.
- Clear price stratification based on room type → **useful for modeling.**

Step 4: Correlation Heatmap

Code Example

```
plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(numeric_only=True), annot=True,
cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```

Insight:

- dist (distance from center) has a **strong negative correlation** with price.
- metro_dist and person_capacity show moderate relationships.
- bedrooms and host_is_superhost are weakly but positively correlated.

Step 5: Pairwise Feature Relationships

Code Example

```
sns.pairplot(df[['realSum', 'person_capacity', 'bedrooms',
'dist', 'metro_dist']])
plt.show()
```

Insight:

- Non-linear patterns may suggest better performance from **tree-based models** like Random Forest.

Key EDA Takeaways for Modeling

Aspect	Observation
Target distribution	Right-skewed -consider **log transform** or tree-based models
Feature correlations	**dist** negatively correlated with price
Categorical variable impact	**room_type** and **host_is_superhost** influence price
Outliers	High-priced outliers exist → can affect linear regression

Aspect	Observation
Feature engineering options	Create new features like **price_per_person** = **realSum** / **person_capacity**

These observations suggest that certain features heavily influence price, which guided our preprocessing and modeling.

3. Data Cleaning & Preprocessing

Cleaning ensures quality data, and preprocessing ensures compatibility with machine learning models.

Step 1: Handle Missing and Unnecessary Data

Code Example

```
# Drop unnamed index column
df = df.drop(columns=['Unnamed: 0'])
# Drop rows with missing values
df = df.dropna()
```

Rationale:

- Unnamed: 0 is just an index column from CSV export, not useful.
- Dropping missing values is safe here since the dataset is large and missing values are not extensive. Alternatively, imputation could be applied.

Step 2: Categorical and Boolean Encoding

Code Example

```
# Convert room_type to category
df['room_type'] = df['room_type'].astype('category')
# Convert boolean to numeric
X['host_is_superhost'] = X['host_is_superhost'].astype(int)
```

Rationale:

- Categorical features like room_type need to be encoded (if used later, you may apply pd.get_dummies).
- Booleans like host_is_superhost must be converted to numeric format for ML algorithms.

Step 3: Outlier Awareness and Skewed Distribution

Code Example

```
sns.histplot(df['realSum'], bins=50)
plt.title("Price Distribution")
plt.show()
```

Rationale:

- The realSum (price) variable is **right-skewed**, indicating outliers.
- You could optionally apply:

Step 4: Feature Selection

Code Example

```
X = df[['person_capacity', 'host_is_superhost', 'bedrooms',
'dist', 'metro_dist']]
y = df['realSum']
```

Rationale:

- Selected features are numeric and relevant based on EDA.
- More categorical features like room_type can be added later with one-hot encoding if needed.

Step 5: Train-Test Split

Code Example

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

Rationale:

- Keeps 20% data for evaluation.
- Random seed ensures reproducibility.

Step 6: Feature Scaling

Code Example

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
```

```
X_test = scaler.transform(X_test)
```

Rationale:

- Scaling is important for **Linear Regression** and other distance-based models.
- Not required for tree-based models (e.g., Random Forest), but it's applied here for consistent pipeline

These transformations produced a clean, structured dataset ready for predictive modeling.

4. Business Analytics Questions

The following questions were formulated to guide the analysis:

a. Which features most influence listing prices?
The most influential features are person_capacity, dist (distance from city center), and bedrooms. These variables directly relate to space, location, and capacity, which are key pricing drivers. Understanding these helps hosts optimize listings and set competitive prices.

b. Does location proximity (e.g., to metro) affect pricing?
Yes, proximity to central areas and metro stations positively correlates with higher prices. Listings further away generally cost less, reflecting lower demand and convenience. This highlights the importance of location in pricing strategy and guest appeal.

c. Do listings from Superhosts command higher prices?
Superhost status is associated with slightly higher listing prices on average. This suggests guests are willing to pay more for verified, high-quality hosting experiences. Trust-building features like Superhost status can add value in competitive markets.

5.Predictive Modeling Report

Goal: Predict Airbnb Listing Price (realSum)

Models Applied: Linear Regression, Random Forest Regressor

Evaluation Metrics: RMSE (Root Mean Squared Error), R^2 Score

Model 1: Linear Regression

Code Example

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np
model = LinearRegression()
model.fit(X_train, y_train)
y_pred_lr = model.predict(X_test)
mse_lr = mean_squared_error(y_test, y_pred_lr)
rmse_lr = np.sqrt(mse_lr)
r2_lr = r2_score(y_test, y_pred_lr)
print("Linear Regression RMSE:", rmse_lr)
print("Linear Regression R2 Score:", r2_lr)
```

Model 2: Random Forest Regressor

Code Example

```
from sklearn.ensemble import RandomForestRegressor
rf_model = RandomForestRegressor(n_estimators=100,
random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)
mse_rf = mean_squared_error(y_test, y_pred_rf)
rmse_rf = np.sqrt(mse_rf)
r2_rf = r2_score(y_test, y_pred_rf)
print("Random Forest RMSE:", rmse_rf)
print("Random Forest R2 Score:", r2_rf)
```

Model Performance Comparison Table

Model	**RMSE (↓)**	**R^2 Score (↑)**
Linear Regression	rmse_lr	r2_lr
Random Forest	rmse_rf	r2_rf

Key Insights

- Random Forest clearly outperforms Linear Regression in both RMSE and R^2.
- This is expected, as Random Forest captures non-linear relationships and interactions better.
- The performance difference justifies using tree-based methods for this pricing problem.

Evaluation Metrics:

- RMSE (Root Mean Squared Error): Measures average error
- R^2 Score: Measures proportion of variance explained

Random Forest outperformed Linear Regression, confirming the need for a model that captures complex feature interactions.

6. Insights and Answers

This summary interprets the results of the predictive models applied to the Airbnb price prediction dataset.
The two models used were Linear Regression and Random Forest Regressor, both trained to predict the price (realSum) of Airbnb listings based on various features including capacity, host type, distance to city center, and more.

Key Insights from Predictive Models

1. The Random Forest model outperformed Linear Regression in both RMSE and R^2 score, indicating it captured the underlying non-linear patterns in the data more effectively.
2. Features like distance to city center ('dist') and number of people accommodated ('person_capacity') had significant influence on the predicted price.
3. Host status ('host_is_superhost') also played a meaningful role in price prediction, confirming consumer preference for experienced and verified hosts.
4. Room type was found to strongly impact price distribution (from EDA), and its inclusion as a one-hot encoded feature in future iterations could further improve accuracy.

Implications for Decision-Making

- Airbnb hosts can optimize pricing strategies based on key features such as location proximity, host verification status, and room type. Listings farther from the city center may need to lower prices or offer additional value to stay competitive.
- Platform-level algorithms could suggest pricing guidelines to new hosts based on similar listings with high performance metrics.
- Urban tourism boards and property managers can use such models to understand the economic influence of location and service quality in different neighborhoods.

Limitations and Future Considerations

- The dataset includes only selected European cities, limiting generalizability to other regions.
- The feature set is relatively narrow; including more attributes like amenities, number of reviews, or listing description sentiment could enhance model accuracy.
- The target variable (price) is highly skewed, and although Random Forests are robust, additional preprocessing like log-transformation could further benefit certain models.
- Seasonality (e.g., month of booking) was not factored into the model but may significantly affect price.

7. Ethics & Interpretability

Machine learning must be implemented with responsibility.

Ethical Considerations

- Predictive pricing should not reinforce regional inequalities
- Transparency is essential to gain host trust

Interpretability

- Linear Regression provides simple, interpretable coefficients
- Random Forest models require feature importance charts or SHAP values for explanation

8. Appendix

Visuals

- Histogram of Price Distribution
- Boxplot by Room Type
- Correlation Heatmap
- Random Forest Feature Importance Chart

Environment

- Python 3.10
- Libraries: pandas, numpy, seaborn, scikit-learn, matplotlib

Resources

-Kaggle Datasets

Link - Airbnb Prices in European Cities

Google Collab Link

Term;ProjectChandiniRajendra.ipynb - Colab