Project Report: Airbnb Dynamic Pricing Recommendation Engine

Author: Mangesh Pawar | **Date:** September 8, 2025

Introduction

Setting the right price for an Airbnb listing is a significant challenge for hosts. A price that is too high can lead to low occupancy, while a price that is too low results in lost revenue. This project addresses this challenge by creating a machine learning-powered engine to recommend optimal daily prices for Airbnb listings. The goal is to analyze historical data, listing features, and seasonality to provide hosts with dynamic, data-driven price suggestions, empowering them to maximize revenue while maintaining competitive and fair market rates.

Abstract

This report details the end-to-end development of a dynamic pricing recommendation engine for Airbnb listings. The project utilized a comprehensive dataset of listings, which underwent extensive cleaning, preprocessing, and feature engineering to prepare it for modeling. Key features such as seasonality, host tenure, and specific amenities were engineered to enhance predictive power. Several regression models were trained and evaluated, including Linear Regression, Decision Tree, Random Forest, and Gradient Boosting. The **Gradient Boosting Regressor** was identified as the best-performing model, achieving an **R-squared of 0.97**. The final phase involved creating a recommendation function that adjusts the model's base prediction with dynamic factors and integrating it into an interactive user interface built with ipywidgets for real-world application.

Tools Used

The project was developed in a Python environment, leveraging the following key libraries and tools:

- **Data Manipulation & Analysis:** Pandas, NumPy
- **Data Visualization:** Matplotlib, Seaborn
- **Machine Learning:** Scikit-learn
- **Interactive Widgets:** Ipywidgets

Steps Involved in Building the Project

The project was executed through a structured, multi-stage process from data ingestion to final recommendation.

1. Data Cleaning and Preprocessing
The initial dataset contained 79 columns with significant missing values across more than 35

of them. The cleaning strategy included:

- **Dropping Columns:** Two columns (neighbourhood_group_cleansed, calendar_updated) were dropped as they were 100% empty.
- **Imputation:** Numerical columns like price, bedrooms, and bathrooms were imputed with their median values to handle skewness. Categorical columns were imputed with an 'Unknown' category, and review-related columns were imputed with 0 or 'No Reviews' to signify the absence of reviews.
- **Type Conversion:** The price column was converted from an object to a numeric type, and date columns were converted to datetime objects.

2. Exploratory Data Analysis (EDA)

EDA was performed to uncover key relationships between features and price. Visualizations revealed that:

- **Price is heavily influenced by room_type**, with 'Entire home/apt' and 'Hotel room' commanding higher prices.
- **Location (neighbourhood_cleansed)** is a major price driver, with certain neighborhoods showing significantly higher price distributions.
- A season feature was created from the last_scraped date, and analysis showed potential variations in price based on the season.

3. Feature Engineering

To prepare the data for modeling, several new features were created:

- Irrelevant columns like IDs and URLs were dropped.
- Dummy variables were created from the amenities column to represent the presence of popular amenities like 'Wifi' and 'Kitchen'.
- Time-based features such as host_tenure_days and time_since_last_review_days were engineered from date columns.
- Categorical features were one-hot encoded to be used in the machine learning models.

4. Model Building and Evaluation

Five different regression models were trained to predict the price: Linear Regression, Decision Tree, Random Forest, Gradient Boosting, and a Voting Regressor (Ensemble). Models were evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2) metrics.

- **Result:** The **Gradient Boosting Regressor** proved to be the most effective model, explaining 97% of the variance in price ($R^2 = 0.97$) and demonstrating the lowest RMSE (323.39).

5. Recommendation Engine and UI

The final step was to build a practical recommendation function. This function uses the trained Gradient Boosting model to generate a base_price prediction. It then applies adjustments based on dynamic factors like seasonality, local events, and demand to produce a final recommended price. An interactive dashboard was created using ipywidgets, allowing a user to input these dynamic factors and receive an instant price recommendation.

Conclusion

This project successfully developed a robust machine learning model for Airbnb dynamic pricing and translated it into a functional recommendation engine. The **Gradient Boosting** model demonstrated high accuracy, and the final tool provides an intuitive interface for hosts to receive data-driven pricing suggestions. This engine has the potential to significantly enhance host revenue and occupancy rates by adapting to complex market dynamics in real-time. Future work could involve integrating live data streams for local events and competitor pricing to further enhance the model's responsiveness.