Predicting Airbnb Prices with XGBoost

Paul BULLIER

December 9, 2025

Abstract

This project aims to predict the logarithm of Airbnb listing prices using data preprocessing, feature engineering, and the XGBoost regression model. The report details the steps of exploratory data analysis, feature preparation, model training, evaluation, and test set prediction. Possible future improvements are also discussed.

1 Introduction

The goal of this project is to predict the price of Airbnb listings based on various features such as location, room type, amenities, and historical reviews. Machine learning models, specifically XGBoost, are used to capture complex relationships in the data.

2 Exploratory Data Analysis

2.1 Data Overview

The training dataset contains `train_df.shape[0]` listings with features such as location, room type, number of reviews, last review date, and amenities.

2.2 Price Distribution

Figure 1 shows the distribution of log-transformed prices, highlighting the skewness in the data.

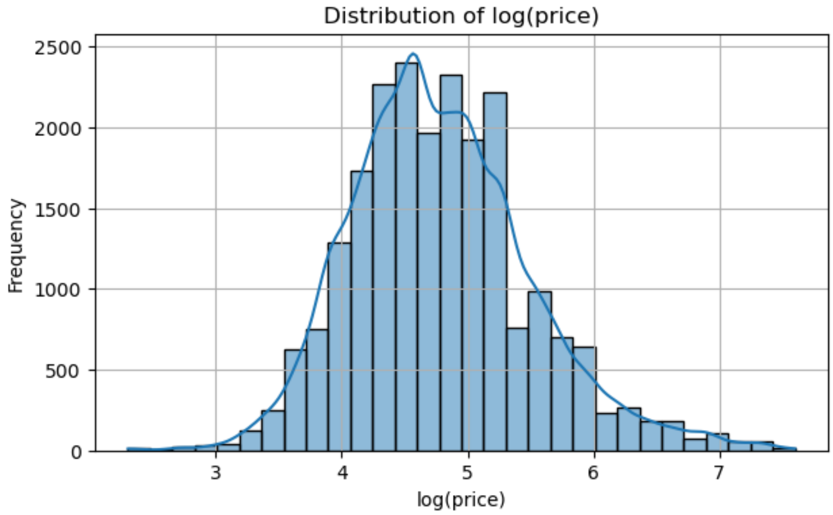

Figure 1: Distribution of log(price)

2.3 Room Type vs Price

Boxplots (Figure 2) show that room type significantly affects prices.

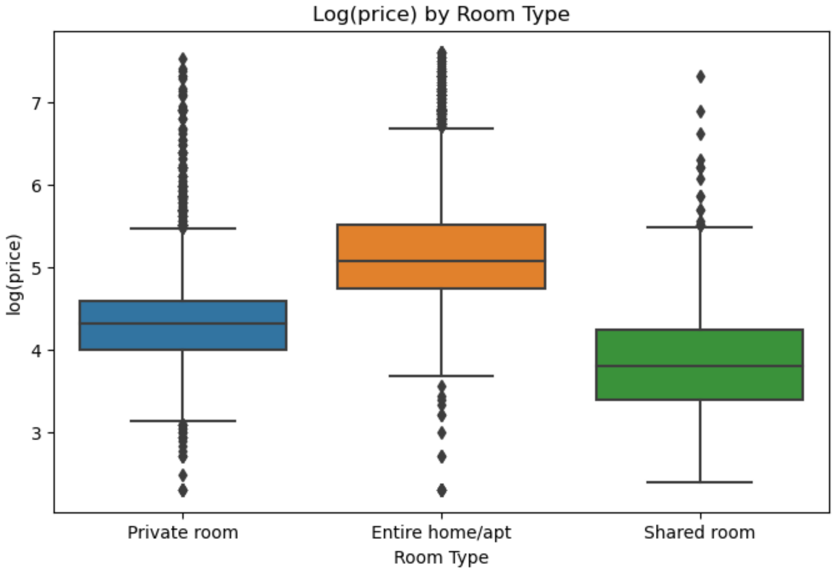

Figure 2: Log(price) by room type

2.4 Geographical Distribution

Figure 3 shows the geographic distribution of listings colored by log(price), indicating the impact of location on pricing.

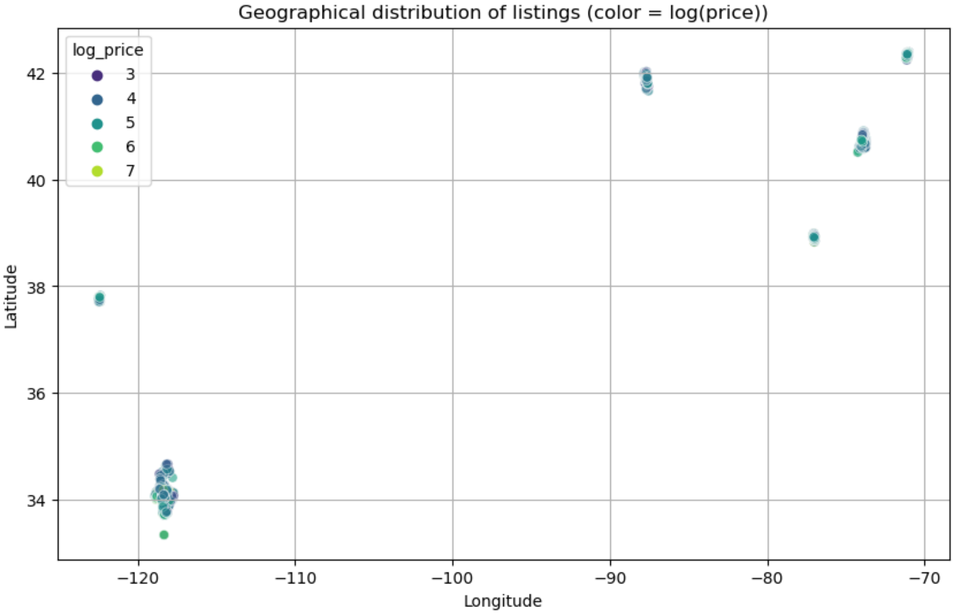

Figure 3: Geographical distribution of listings

2.5 Amenities Effect

Binary features such as `has_wifi`, `has_pool`, etc., were derived from the `amenities` column to quantify the effect of specific amenities on price.

3 Data Preparation

Data preparation included:

- Extracting binary features from the `amenities` column.
- Computing `name_length` from the listing title.
- Converting `last_review` to `last_review_year` and `last_review_month`.
- Encoding rare neighborhoods as "Other".

Numerical features were standardized, and categorical features were one-hot encoded. Skewed features like `number_of_reviews` were log-transformed.

4 Modeling

4.1 XGBoost Regressor

An XGBoost regressor was trained using the preprocessed features. Key hyperparameters included:

- `n_estimators = 500`
- `max_depth = 7`

- `learning_rate = 0.1`

- `subsample = 0.8`

- `colsample_bytree = 0.8`

4.2 Training and Validation

The training set was split into 95% for training and 5% for validation. Model performance was evaluated using RMSE and R^2.

Metric	Value
RMSE	0.39
R^2	0.69

Table 1: Validation performance of XGBoost model

Figure 4 shows predicted vs actual log(price) on the validation set.

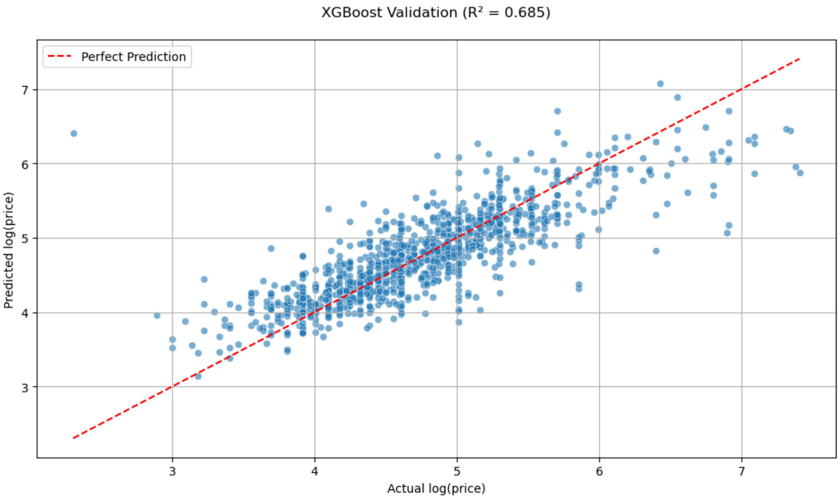

Figure 4: XGBoost Validation: predicted vs actual log(price)

5 Test Set Predictions

The test dataset was processed with the same pipeline. Predictions were exported to `test_predictions.csv` for submission.

6 Conclusion

The project demonstrates that even simple feature engineering and XGBoost can produce reasonable predictions. Lessons learned include:

- Importance of location and room type

- Feature engineering on textual and categorical variables

- Preprocessing and handling skewed distributions

Future improvements:

- Explore alternative models such as LightGBM and CatBoost
- Add temporal features (seasonality, time since last booking)
- Perform advanced hyperparameter tuning with Bayesian optimization or GridSearchCV