Airbnb Pricing and Value Assessment in New York City Using Machine Learning

Karol Marszałek - S31952

January 14, 2026

Abstract

In this project, I analyze Airbnb pricing in New York City using machine learning methods. While browsing Airbnb listings, I noticed that prices often vary significantly even for listings that appear similar. The goal of this work was to better understand what drives these price differences and to build models that can both estimate a fair market price and assess whether a given listing is worth its price. I followed a complete data science workflow, focusing not only on predictive performance, but also on interpretability, evaluation, and understanding model behavior.

Contents

1 Problem Description

When I started working on this project, I approached it from a very practical perspective. As an Airbnb user, I often wondered whether a given listing was fairly priced or simply expensive because of market noise. At the same time, hosts face the opposite problem: deciding how much to charge to stay competitive.

The goal of this project was therefore to answer two concrete questions:

- What is a reasonable market price for a given Airbnb listing?
- Given an advertised price, is the listing worth it compared to similar listings?

2 Data

2.1 Data Sources

I used two publicly available datasets:

- **NYC Airbnb Open Data (2019)**
  https://www.kaggle.com/datasets/dgomonov/new-york-city-airbnb-open-data
- **MTA Subway Stations Dataset**
  https://catalog.data.gov/dataset/mta-subway-stations

The Airbnb dataset provides detailed information about listings, while the subway dataset allowed me to incorporate transportation accessibility, which I considered an important real-world factor.

2.2 Temporal Scope and Inflation Considerations

The Airbnb dataset reflects market conditions from 2019. Because housing prices and short-term rental markets evolve over time, the absolute price levels learned by the models cannot be interpreted as present-day prices. To make it more realistic, it would be necessary to adjust prices for inflation and market trends, such as using external housing market indices.

2.3 Data Structure

The Airbnb dataset contains 48,895 observations and 16 original variables. I grouped the variables as follows:

- **Numerical variables**: price, availability_365, minimum_nights, number_of_reviews, reviews_per_month, calculated_host_listings_count
- **Categorical variables**: room_type, neighbourhood_group, neighbourhood
- **Geographic variables**: latitude, longitude
- **Identifier variables**: id, host_id, name, host_name

3 Data Cleaning and Preparation

Before any modeling, I carefully inspected the dataset. I verified data types, checked missing values, and tried to understand what those missing values actually meant. For example, missing values in *reviews_per_month* and *last_review* usually indicate listings without recent activity rather than data errors.

Instead of aggressively removing rows or outliers, I chose to preserve most of the data. Airbnb prices are naturally noisy, and removing extreme values too early could distort the structure. I addressed this variability later through model choice and evaluation metrics.

4 Technology Stack

The project was implemented using the standard Python data science stack. All analysis and modeling steps were carried out using:

- **Python** as the main programming language,
- **pandas** for data loading, cleaning, and preparation,
- **NumPy** for numerical computations,
- **scikit-learn** for machine learning models, evaluation metrics, and cross-validation,
- **matplotlib** and **seaborn** for data visualization,
- **Jupyter Notebook** for exploratory analysis and iterative development.

5 Feature Engineering

5.1 Motivation

Latitude and longitude are difficult to interpret directly, both for humans and for many models. From a user perspective, location is usually understood in terms of proximity to important places rather than raw coordinates. For this reason, I engineered additional spatial features.

5.2 Distance-Based Features

Using the Haversine formula, I computed:

- Distance to the nearest subway station (meters) as there are many of them,
- distance to the center of Manhattan (kilometers).

The center of Manhattan was approximated using the coordinates of Times Square:

$$\text{Times Square} = (40.7580^\circ N, 73.9855^\circ W)$$

5.3 Interpretation of Distance and Availability

Distance to Manhattan acts as a proxy for proximity to business districts, tourist attractions, and economic activity. Listings closer to Manhattan tend to be more expensive and attract higher demand.

The Distance to the nearest subway station captures transportation accessibility, which is a key factor in New York.

I also observed that *availability_365* does not measure quality. In practice, high availability often corresponds to low demand or overpricing. Listings that are far from Manhattan and available most of the year are often overpriced relative to comparable listings.

6 Exploratory Data Analysis

6.1 Price Distribution

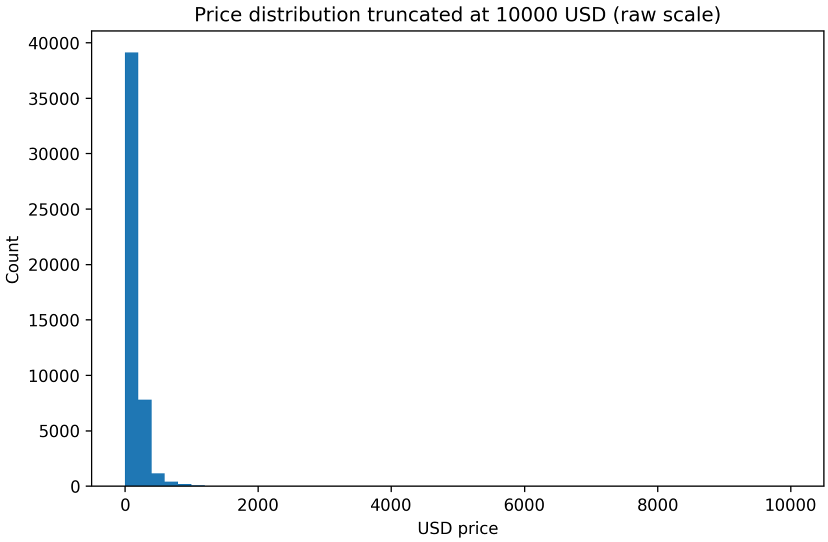


Figure 1: Raw price distribution.

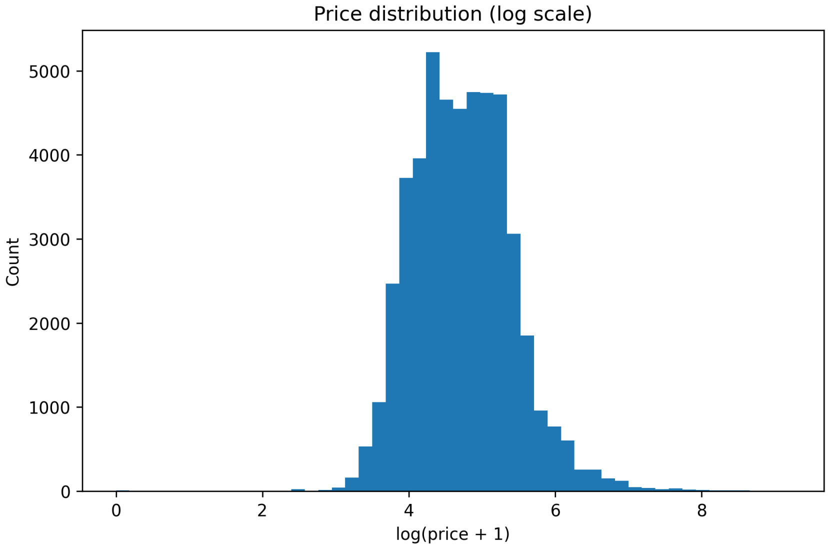


Figure 2: Log-transformed price distribution.

Prices are strongly right-skewed, motivating the use of non-linear models.

6.2 Room Type and Neighborhood

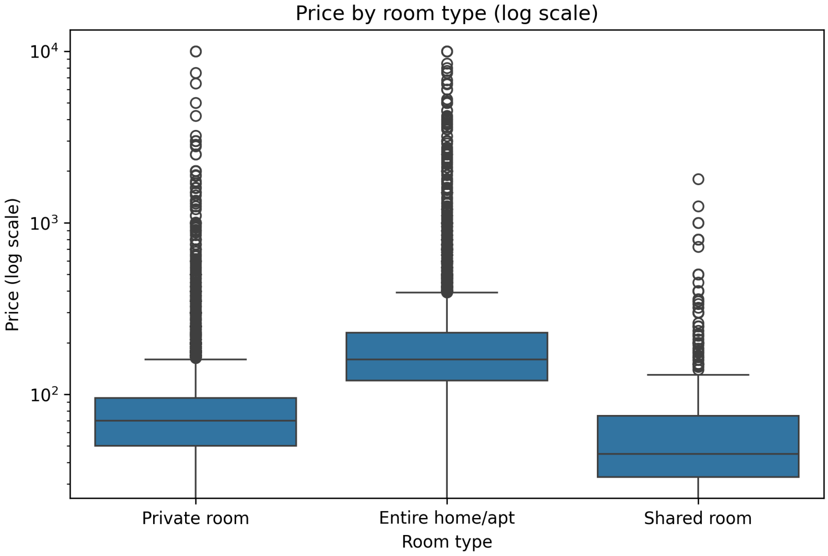


Figure 3: Price distribution by room type.

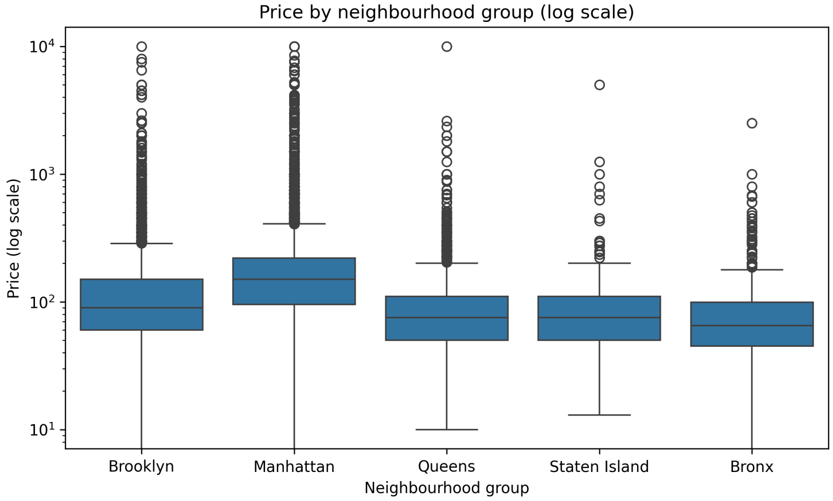


Figure 4: Price distribution by neighborhood group.

6.3 Availability and Spatial Patterns

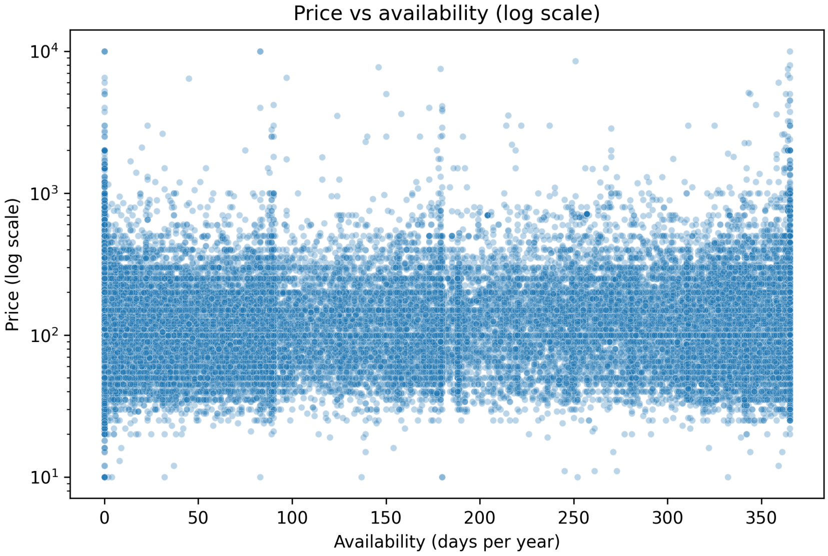


Figure 5: Price vs availability.

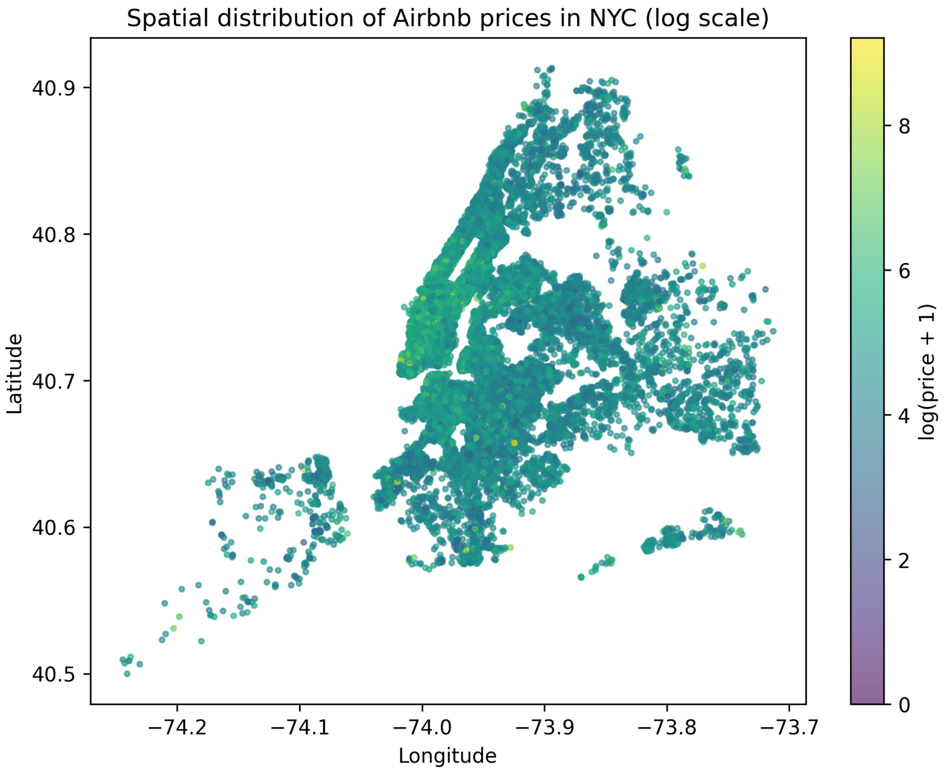
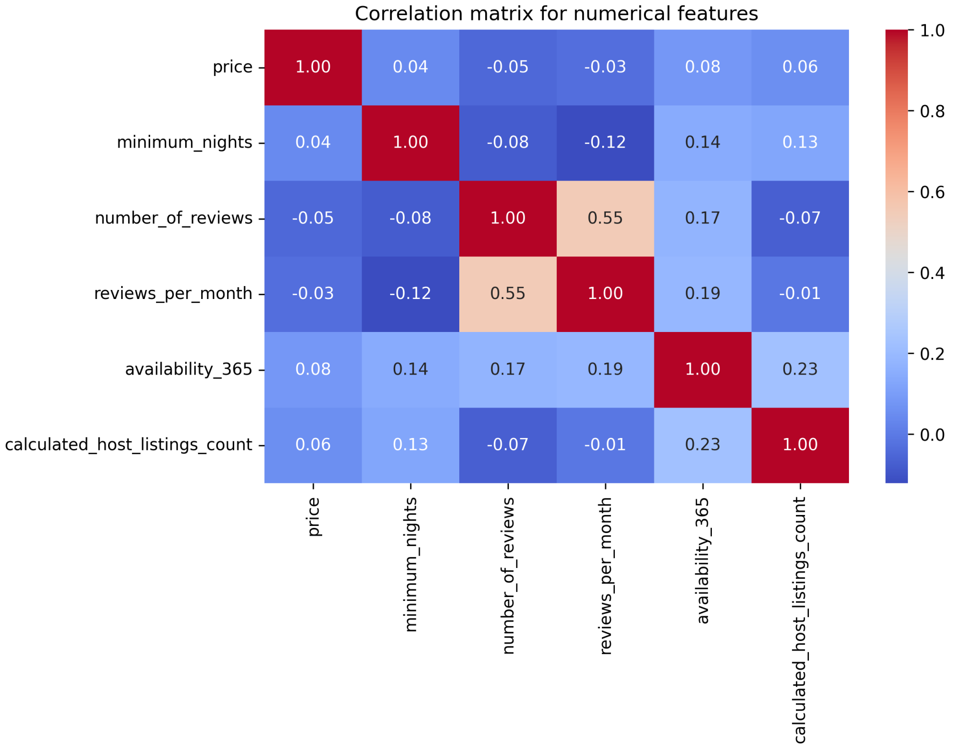


Figure 6: Spatial distribution of prices.

6.4 Correlation Analysis



	price	minimum_nights	number_of_reviews	reviews_per_month	availability_365	calculated_host_listings_count
price	1.00	0.04	-0.05	-0.03	0.08	0.06
minimum_nights	0.04	1.00	-0.08	-0.12	0.14	0.13
number_of_reviews	-0.05	-0.08	1.00	0.55	0.17	-0.07
reviews_per_month	-0.03	-0.12	0.55	1.00	0.19	-0.01
availability_365	0.08	0.14	0.17	0.19	1.00	0.23
calculated_host_listings_count	0.06	0.13	-0.07	-0.01	0.23	1.00

Figure 7: Correlation matrix of numerical features.

Linear correlations with price are generally weak, supporting the use of tree-based models.

7 Regression Modeling

The first modeling task was regression, where the goal was to predict nightly price:

$$y = \text{price}$$

I used the same 80/20 train–test split for all regression models to ensure a fair comparison. I started with a simple baseline model and gradually increased model complexity.

7.1 Models and Evaluation

$$MAE = \frac{1}{n}\sum |y - \hat{y}|, \quad RMSE = \sqrt{\frac{1}{n}\sum (y - \hat{y})^2}$$

Table 1: Regression model performance.

Model	MAE	RMSE	CV RMSE Mean
Linear Regression	49.78	79.66	79.85
Decision Tree	46.97	78.40	80.59
Random Forest	**44.26**	**72.49**	**74.32**

7.2 Cross-Validation Strategy

In addition to a single train-test split, I applied k-fold cross-validation with $k = 5$ to evaluate model stability. This helped identify overfitting in decision trees and confirmed that random forests achieve a better balance between bias and variance.

7.3 Regression Feature Importance

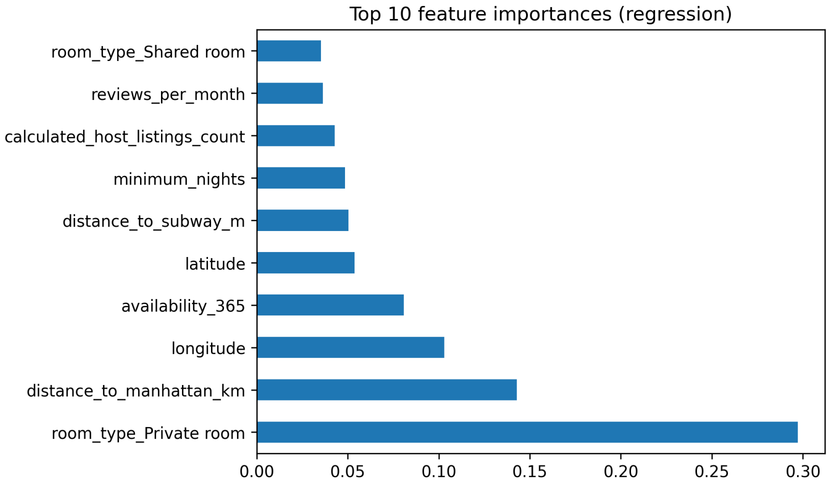


Figure 8: Feature importance from Random Forest regression.

Distance to Manhattan and room type emerged as the most influential predictors.

8 Classification Modeling

After regression, I formulated a classification task to answer a different question: whether a listing is worth its advertised price compared to similar listings.

8.1 Evaluation Metrics

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}, \quad F1 = 2\frac{PR}{P + R}$$

Table 2: Classification model performance.

Model	Precision	Recall	F1	ROC-AUC
Logistic Regression	0.77	0.02	0.03	0.56
Decision Tree	0.70	0.73	0.71	0.77
Random Forest	**0.70**	**0.74**	**0.72**	**0.80**

8.2 Interpretation of Classification Confidence

During testing and while building the CLI application, I observed that predicted confidence values often remain close to 40–50%. This reflects genuine market ambiguity rather than a modeling error. The classifier learns probabilistic patterns from historical data and does not see the user provided price, which explains moderate confidence levels.

8.3 ROC Curve and Feature Importance

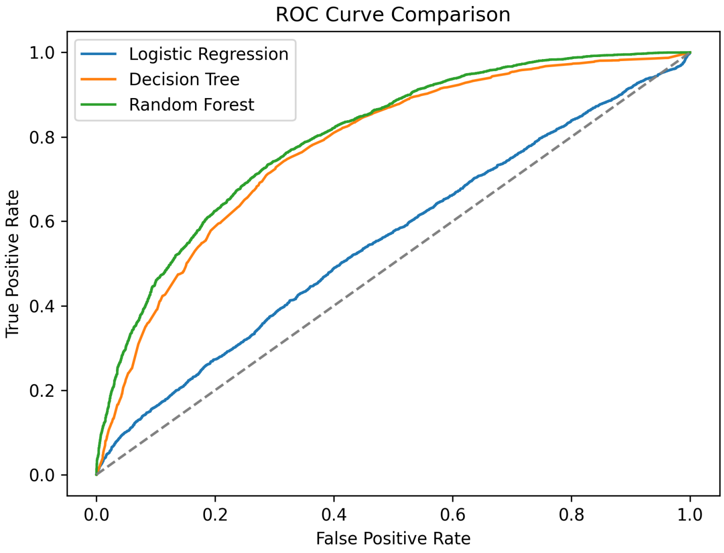


Figure 9: ROC curve comparison.

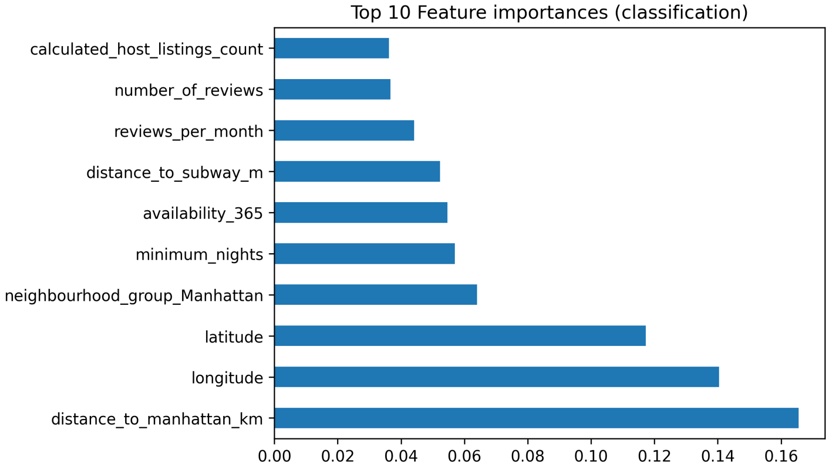


Figure 10: Feature importance from Random Forest classification.

9 Summary and Conclusions

In this project, I applied machine learning models to analyze Airbnb pricing in New York City. The results show that location, accessibility, room type, and availability are key drivers of price, while pricing behavior remains noisy and ambiguous.

Tree-based models performed better than linear models, indicating the importance of non-linear relationships. At the same time, moderate classification confidence highlights natural market uncertainty.