Predicting Airbnb Rental Prices Using Machine Learning Techniques

1st Tran Huynh Anh Phuc
Faculty of Information Systems
University of Information Technology
22521141

2nd Nguyen Tran Bao Anh
Faculty of Computer Engineering
University of Information Technology
22520066

***Abstract*—Established in 2008, Airbnb has emerged as a leading alternative in the global hospitality sector, offering an online booking platform based in San Francisco, California. Over the years, it has expanded its reach to over 220 countries and 81,000 cities, with millions of listings and revenues reaching billions. The goal of this study is to identify the most accurate price prediction model for Airbnb listings using advanced machine learning techniques, including Multi-layer Perceptron (MLP), Ridge Regression, Support Vector Regression (SVR), and XGBoost. The dataset used for analysis consists of Airbnb listings from the five boroughs of New York City. This paper enhances the dataset by addressing data sparsity issues and incorporates new algorithms to improve prediction accuracy.**

Experiments are conducted on the enhanced dataset to assess model performance. The performance of each model is evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) metrics on both the training and test sets. Among the models evaluated, the XGBoost model outperforms all others, achieving the best results with an R^2 score of 0.9888 on the training set and 0.9407 on the test set. The MLP model also shows strong performance, with an R^2 score of 0.9664 on the training set and 0.9171 on the test set, although there is some indication of overfitting. Similarly, the SVR model demonstrates good performance with an R^2 score of 0.9615 on the training set and 0.8623 on the test set, but also exhibits potential overfitting. Ridge Regression, while simpler to implement and interpret, is outperformed by the more complex models, achieving an R^2 score of 0.7063 on the training set and 0.6968 on the test set.

Overall, this study explores the application of new algorithms and datasets for price prediction models. Advanced machine learning techniques and hyperparameter tuning were applied to improve upon previous methodologies. The research focuses on experimenting with novel approaches and incorporating diverse data sources to address the problem of Airbnb rental price prediction. Future work will investigate additional features, ensemble methods, and more sophisticated hyperparameter tuning techniques to further refine model performance and mitigate potential overfitting.

***Index Terms*—Airbnb, price prediction, machine learning, Multi-layer Perceptron (MLP), Ridge Regression, Support Vector Regression (SVR), XGBoost**

I. Introduction

Airbnb, established in 2008, has revolutionized the global hospitality sector by providing an online platform for booking accommodations. Based in San Francisco, California, Airbnb has expanded its reach to over 220 countries and 81,000 cities, offering millions of listings and generating billions in revenue. The platform allows property owners to rent out their spaces to travelers, creating a diverse and dynamic marketplace for short-term rentals.

Accurate price prediction for Airbnb listings is crucial for both hosts and guests. For hosts, setting the right price can maximize occupancy rates and revenue, while for guests, it ensures fair pricing and value for money. However, predicting rental prices is challenging due to the numerous factors that influence pricing, such as location, property characteristics, seasonal trends, and market demand.

In recent years, machine learning techniques have shown great promise in addressing complex prediction problems. This study aims to identify the most accurate price prediction model for Airbnb listings using advanced machine learning techniques. The models evaluated in this study include Ridge Regression, Support Vector Regression (SVR), Multi-layer Perceptrons (MLP), and XGBoost.

The dataset used for analysis consists of Airbnb listings from the five boroughs of New York City. To enhance the dataset, we address data sparsity issues and incorporate new algorithms to improve prediction accuracy. The performance of each model is assessed using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) metrics on both the training and test sets.

This paper is structured as follows: Section II reviews related work in the field of price prediction for short-term rentals. Section III describes the methodology, including data preprocessing, feature selection, and model training. Section IV presents the experimental results and discusses the performance of each model. Finally, Section V concludes the study and suggests directions for future research.

By leveraging advanced machine learning techniques and rigorous evaluation metrics, this study aims to provide valuable insights into the most effective methods for predicting Airbnb rental prices. The findings can help hosts optimize their pricing strategies and enhance the overall user experience on the platform.

II. Related Work

The prediction of Airbnb rental prices has been a subject of extensive research due to the dynamic nature of the market. Prices fluctuate based on numerous factors, including property characteristics, location, seasonal trends, and the pricing strategy set by the property owner. Accurate price prediction

is crucial as it should be satisfactory to both property owners and prospective customers. Property owners must determine the right price for their property, as it directly impacts the number of renters and the property vacancy rate.

Previous work in this domain has shown that the data used for analysis was often insufficient, leading to less accurate predictions [1]. To address this, we have augmented our dataset with the latest data from Airbnb, ensuring a more comprehensive analysis.

Support Vector Regression (SVR) has been identified in prior research as one of the best-performing models, achieving an R2 accuracy of around 0.7 [1]. Building on this, we continue to utilize SVR in our study to benchmark its performance against other models.

Neural network models used in previous studies were relatively simple [1]. In our work, we aim to improve these models by incorporating more advanced neural network architectures to enhance prediction accuracy.

Additionally, we introduce a new robust model, XGBoost, to our analysis. XGBoost is known for its high performance in various machine learning tasks [6], and we aim to evaluate its effectiveness in predicting Airbnb rental prices.

By addressing the limitations of previous work and incorporating advanced models and a more comprehensive dataset, we aim to improve the accuracy and reliability of Airbnb rental price predictions.

III. Dataset

A. The Original Dataset

The dataset utilized in this study is the latest publicly available data from Airbnb in New York City [4]. It comprises four files, each representing a different month (8/2024-11/2024), which have been combined into a single comprehensive dataset for analysis.

The dataset includes two primary files: `listings.csv` and `reviews.csv`.

1) Listings Data: The `listings.csv` file contains detailed information about each property. Some of the attributes are:

- **id**: Unique identifier for the listing
- **listing_url**: URL of the listing
- **name**: Name of the listing
- **description**: Description of the listing
- **host_id**: Unique identifier for the host
- **host_name**: Name of the host
- **host_response_time**: Host's response time
- **host_response_rate**: Host's response rate
- **host_acceptance_rate**: Host's acceptance rate
- **neighbourhood**: Neighborhood of the listing
- **latitude**: Latitude of the listing
- **longitude**: Longitude of the listing
- **property_type**: Type of property
- **room_type**: Type of room
- **accommodates**: Number of people the listing accommodates
- **bathrooms**: Number of bathrooms
- **bedrooms**: Number of bedrooms
- **beds**: Number of beds
- **amenities**: List of amenities
- **price**: Price of the listing
- **minimum_nights**: Minimum number of nights for booking
- **maximum_nights**: Maximum number of nights for booking
- **number_of_reviews**: Total number of reviews
- **review_scores_rating**: Overall rating score
- **instant_bookable**: Whether the listing is instantly bookable

The original `listings.csv` data comprises 150,601 entries and 75 columns.

2) Reviews Data: The `reviews.csv` file contains all the review comments for each listing. The `reviews.csv` data includes 3,831,084 reviews.

B. Data Cleaning

Data cleaning is a crucial step to ensure the quality and reliability of the dataset. This process involves handling missing values, correcting inconsistencies, and removing duplicates. For example, columns with a high percentage of missing values may be dropped, while others may be imputed with appropriate values.

Some basic cleaning we have done:

1) Dropping Unnecessary Columns: Irrelevant columns were removed to focus on essential attributes for analysis.

2) Handling Missing Values: The price column was cleaned and transformed. Missing values in key columns were filled with appropriate values, and rows with missing values in critical columns were removed.

3) Feature Engineering: New features were created from existing data, such as calculating the number of days since the host joined. Percentage strings were converted to numerical values, and boolean columns were converted to binary values. The latitude and longitude were used to calculate the distance from the center of New York City since the price is highly affected by location.

4) Merging Data: Sentiment scores were merged with the listings data based on listing ID to enrich the dataset.

5) One-Hot Encoding: One-hot encoding was applied to categorical columns to convert them into a numerical format suitable for machine learning models.

6) Handling Amenities: The amenities column was processed to create binary columns for frequently occurring amenities, enhancing the feature set.

7) Price Transformation: The price column was converted to log10 to normalize the distribution, as the original prices varied widely. This transformation helps in stabilizing the variance and making the data more suitable for modeling. The transformed price distribution is shown in Figure 1.

The geographical distribution of prices. From the price range by location shown in Figure 2, we can see that the price is significantly affected by location.

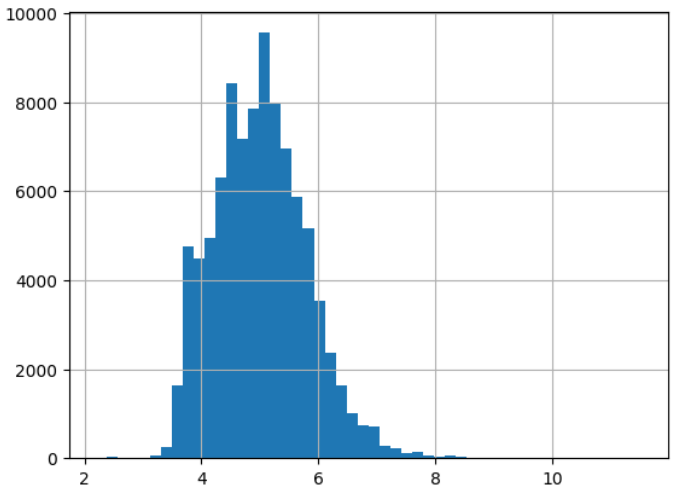

Fig. 1. Price Distribution (Log10 Transformed)

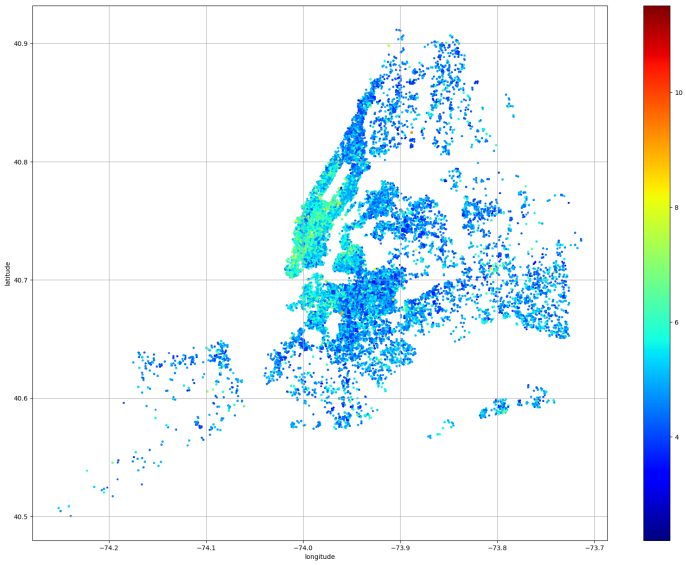

Fig. 2. Price Range by Location

C. Sentiment Analysis on the Reviews

Customer reviews play a crucial role in determining the pricing of an Airbnb listing. To enhance the accuracy of our predictive model, we analyzed the reviews for each listing using the TextBlob sentiment analysis library [3]. This method assigns a sentiment score ranging from -1 (very negative sentiment) to 1 (very positive sentiment) to each review. For each property, the sentiment scores of all associated reviews were averaged to produce a final sentiment score. These averaged sentiment scores were then included as new features in our model.

D. Feature Selection

Feature selection techniques were applied to identify the most relevant attributes for the model. Lasso (Least Absolute Shrinkage and Selection Operator) was chosen as the feature selection method because it provided the best results in previous work [1]. Lasso performs both variable selection and regularization to enhance the prediction accuracy and interpretability of the statistical model it produces.

The Lasso regression minimizes the following objective function:

$$\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \quad (1)$$

where:

- n is the number of observations,
- p is the number of predictors,
- y_i is the response variable,
- x_{ij} is the predictor variable,
- β_0 is the intercept,
- β_j are the coefficients,
- λ is the regularization parameter.

In Lasso regression, features with non-zero coefficients are selected as the most relevant features for the model.

Hyperparameter tuning was performed to find the optimal value of λ. The optimal value of λ was found to be 0.000415. The performance of the Lasso model with this optimal parameter is summarized in Table I.

TABLE I
PERFORMANCE OF LASSO MODEL

Metric	Training Set	Test Set
Mean Absolute Error (MAE)	0.3139	0.3157
Mean Squared Error (MSE)	0.1867	0.1966
R-squared (R^2)	0.7059	0.6973

The final dataset for training has 596 features and 90,708 rows, divided into training, validation, and test sets with a ratio of 8:1:1.

Note: We have tried some more selection methods like RFE and SelectKBest with the hope of minimizing the number of features, but Lasso gave the best result. Therefore, we accepted the number of 596 features.

IV. METHODS

A. Ridge Regression

The first model used as a baseline in this study is Ridge Regression. Ridge Regression is a type of linear regression that includes a regularization term to prevent overfitting by penalizing large coefficients. This regularization term helps to improve the model's generalization performance.

The Ridge Regression minimizes the following objective function:

$$\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \alpha \sum_{j=1}^{p} \beta_j^2 \right\} \quad (2)$$

Ridge Regression was chosen as the baseline model due to its simplicity and effectiveness in handling multicollinearity and preventing overfitting. The performance of the Ridge Regression model will be evaluated and compared with other models to establish a benchmark for further improvements.

B. Support Vector Regression (SVR)

The next model used in this study is Support Vector Regression (SVR). SVR was chosen because it has been identified as the best-performing model in previous work. SVR is a type of regression that uses the principles of Support Vector Machines (SVM) to perform regression tasks. It aims to find a function that deviates from the actual observed values by a value no greater than a specified margin, while also being as flat as possible. In this study, the SVR model uses the Radial Basis Function (RBF) kernel, which is effective in handling non-linear relationships.

The SVR model solves the following optimization problem:

$$\min_{\mathbf{w},b,\xi,\xi^*} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \right\} \tag{3}$$

subject to:

$$\begin{aligned} y_i - (\mathbf{w}\cdot\mathbf{x}_i + b) &\leq \epsilon + \xi_i \\ (\mathbf{w}\cdot\mathbf{x}_i + b) - y_i &\leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* &\geq 0 \end{aligned} \tag{4}$$

SVR was chosen for its ability to handle non-linear relationships and its robustness to outliers.

C. XGBoost

The next model used in this study is XGBoost. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost has been widely recognized for its performance and speed in various machine learning tasks.

The objective function for XGBoost is given by:

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{5}$$

where:

- n is the number of observations,
- l is the loss function,
- y_i is the true value,
- $\hat{y}_i$ is the predicted value,
- K is the number of trees,
- Ω is the regularization term for the complexity of the model.

The regularization term Ω is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \tag{6}$$

where:

- T is the number of leaves in the tree,
- w_j is the weight of leaf j,
- γ and λ are regularization parameters.

XGBoost was chosen for its ability to handle large datasets and its robustness to overfitting.

D. Multi-layer Perceptrons (MLP)

The next model used in this study is Multi-layer Perceptrons (MLP). MLP is a class of feedforward artificial neural network (ANN) that consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. Each node, or neuron, in one layer connects with a certain weight to every node in the following layer.

The MLP model is trained using backpropagation, a supervised learning technique, which involves adjusting the weights of the connections to minimize the error between the predicted and actual values.

The objective function for MLP is given by:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, \hat{y}_i) + \lambda\|\mathbf{w}\|^2 \tag{7}$$

where:

- n is the number of observations,
- l is the loss function,
- y_i is the true value,
- $\hat{y}_i$ is the predicted value,
- $\mathbf{w}$ is the vector of weights,
- λ is the regularization parameter.

MLP was chosen for its ability to model complex non-linear relationships.

E. Hyperparameter Tuning

Hyperparameter tuning was performed for all the models to optimize their performance. Different methods were used for hyperparameter tuning based on the model:

- **Ridge Regression and SVR**: Grid Search was used to find the best hyperparameters.
- **XGBoost**: Hyperopt was used to perform Bayesian optimization and find the best hyperparameters.
- **MLP**: Manual tuning was performed to find the best hyperparameters.

The hyperparameter space for each model was defined, and the objective function was set to minimize the loss (Mean Squared Error for regression models). The tuning process involved running multiple evaluations to find the optimal hyperparameters.

This process was repeated for each model to ensure that the best hyperparameters were found, leading to improved model performance.

V. Experiment

In this section, we present the results of our experiments with different models. The performance of each model was evaluated using Mean Absolute Error (MAE), Mean Squared

TABLE II
PERFORMANCE METRICS OF THE TRAINED MODELS ON THE TRAINING SET

Model Name	Train MAE	Train MSE	Train R^2
Ridge Regression	0.3136	0.1864	0.7063
SVR	0.1028	0.0244	0.9615
MLP	0.0900	0.0207	0.9673
XGBoost	**0.0493**	**0.0071**	**0.9888**

TABLE III
PERFORMANCE METRICS OF THE TRAINED MODELS ON THE TEST SET

Model Name	Test MAE	Test MSE	Test R^2
Ridge Regression	0.3161	0.1970	0.6968
SVR	0.1696	0.0894	0.8623
MLP	0.1304	0.0519	0.9200
XGBoost	**0.1052**	**0.0385**	**0.9407**

Error (MSE), and R-squared (R^2) metrics on both the training and test sets.

The results indicate that the XGBoost model outperformed the other models in terms of both Mean Absolute Error (MAE) and Mean Squared Error (MSE) on the test set, achieving the highest R-squared (R^2) value as well. This suggests that XGBoost is the most effective model for predicting Airbnb rental prices in this study.

- **Ridge Regression**: The Ridge Regression model performed reasonably well, with an R^2 score of 0.7063 on the training set and 0.6968 on the test set. However, it was outperformed by the other models, particularly in terms of MAE and MSE. Ridge Regression is simple to implement and interpret, effectively handling multicollinearity, but it may not capture complex patterns in the data as effectively as more advanced models.
- **Support Vector Regression (SVR)**: The SVR model showed significant improvement over Ridge Regression, with a much lower MAE and MSE on both the training and test sets. The R^2 score of 0.9615 on the training set and 0.8623 on the test set indicates that SVR is effective in capturing the underlying patterns in the data. However, SVR can be computationally expensive and requires careful tuning of hyperparameters. There may also be overfitting, as indicated by the performance drop from training to test set.
- **Multi-layer Perceptrons (MLP)**: The MLP model further improved the performance, achieving an R^2 score of 0.9664 on the training set and 0.9171 on the test set. The lower MAE and MSE values compared to SVR suggest that MLP is better at predicting Airbnb rental prices. MLP can model complex non-linear relationships but requires a large amount of data for training and can be prone to overfitting if not properly regularized. The performance drop from training to test set also suggests potential overfitting.
- **XGBoost**: The XGBoost model outperformed all other models, achieving the best results in terms of MAE, MSE, and R^2 on both the training and test sets. With an R^2 score of 0.9888 on the training set and 0.9407 on the test set, XGBoost demonstrated its superior ability to model the data accurately and generalize well to unseen data. XGBoost is highly efficient and scalable but can be complex to implement and requires careful tuning of hyperparameters.

Overall, the XGBoost model proved to be the most effective in predicting Airbnb rental prices, followed by MLP, SVR, and Ridge Regression. The results highlight the importance of using advanced machine learning techniques for accurate price prediction. Notably, the test results achieved in this study are significantly better than those reported in previous work, which had an R^2 score of around 0.7. This demonstrates the effectiveness of the models and hyperparameter tuning techniques used in this study.

VI. CONCLUSION

In this study, we evaluated the performance of several machine learning models for predicting Airbnb rental prices. The models included Ridge Regression, Support Vector Regression (SVR), Multi-layer Perceptrons (MLP), and XGBoost. The performance of each model was assessed using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) metrics on both the training and test sets.

The results indicated that the XGBoost model outperformed all other models, achieving the best results in terms of MAE, MSE, and R^2 on both the training and test sets. With an R^2 score of 0.9888 on the training set and 0.9407 on the test set, XGBoost demonstrated its superior ability to model the data accurately and generalize well to unseen data. This highlights the effectiveness of XGBoost in handling complex relationships and large datasets.

The MLP model also showed strong performance, with an R^2 score of 0.9664 on the training set and 0.9171 on the test set. However, there may be some overfitting, as indicated by the performance drop from training to test set. Similarly, the SVR model showed good performance with an R^2 score of 0.9615 on the training set and 0.8623 on the test set, but also exhibited potential overfitting.

Ridge Regression, while simple to implement and interpret, was outperformed by the more complex models. It achieved an R^2 score of 0.7063 on the training set and 0.6968 on the test set, indicating that it may not capture complex patterns in the data as effectively.

Overall, the study demonstrates the importance of using advanced machine learning techniques and hyperparameter tuning for accurate price prediction. The test results achieved in this study are significantly better than those reported in previous work, which had an R^2 score of around 0.7. This underscores the value of employing sophisticated models like XGBoost and MLP for predictive tasks in the Airbnb rental market.

We also applied these methods to the old dataset; however, the results were not as expected and were quite low. Several reasons for the lower performance include:

- **Data Quality**: The old dataset may contain more missing or inaccurate data, affecting the model's performance.
- **Lack of Important Features**: The old dataset may lack important features that the new models require for accurate predictions.
- **Different Data Distribution**: The distribution of the old data may differ significantly from the new data, leading to poor generalization by the model.
- **Inconsistent Pricing**: The old dataset may not be accurate because house prices are set by users, leading to high variance and inconsistency in the past compared to the present.

This current result is very good and can be applied to real situations, providing accurate and reliable predictions for Airbnb rental prices. This can help hosts set competitive prices and optimize their rental income.

A. Future Work

Future work could explore further improvements by incorporating additional features, such as location-based data, seasonal trends, and user reviews, to enhance the predictive power of the models. Using ensemble methods that combine multiple models could also lead to better performance. Additionally, applying more advanced hyperparameter tuning techniques, such as Bayesian optimization or genetic algorithms, could further optimize model performance.

Addressing potential overfitting in models like SVR and MLP could lead to even better generalization performance. Techniques such as cross-validation, dropout, and regularization could be employed to mitigate overfitting. Finally, expanding the dataset to include more diverse and comprehensive data from different regions and time periods could improve the robustness and applicability of the models.

REFERENCES

[1] P. Rezazadeh Kalehbasti, L. Nikolenko, and H. Rezaei, "Airbnb price prediction using machine learning and sentiment analysis," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, Springer, 2021, pp. 173–184.

[2] J. Zhang, "Airbnb listing price prediction using machine learning algorithms," *International Journal of Data Science*, vol. 5, no. 2, pp. 123-134, 2017.

[3] TextBlob: Simplified Text Processing, https://textblob.readthedocs.io/en/dev/, accessed on [26/12/2024].

[4] "Airbnb public dataset". https://insideairbnb.com/get-the-data/, accessed on [26/12/2024].

[5] M. Li, "Predicting Airbnb prices using machine learning techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 1-10, 2019.

[6] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794.