Predicting Airbnb Listing Prices of Popular Tourist Destinations in Europe

Moshiur Rahman
ID: 2131903642
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
moshiur.rahman21@northsouth.edu

Shahariar Ifti
ID: 2012632042
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
shahariar.ifti@northsouth.edu

***Abstract*—This project presents a machine learning approach to predicting Airbnb listing prices in ten popular tourist destinations across Europe. Several machine learning algorithms, including Linear Regression, Support Vector Regression, Decision Tree, Random Forest, and Gradient Boosting, are trained and evaluated to identify the optimal model. The goal of the project is to benefit individual hosts and travelers, as well as provide valuable insights for market analysis and strategic planning in the hospitality industry.**

I. Introduction

Airbnb is a platform that enables people to rent out their properties or spare rooms to guests seeking short-term accommodations. It connects hosts and travelers, offering a wide range of lodging options, from apartments and houses to unique stays like treehouses or castles. Predicting Airbnb listing prices for tourist destinations is an intriguing problem, due to dynamic price fluctuations influenced by demand, seasonality, and local regulations. It involves analyzing diverse data types like property features, location attributes, and external factors such as weather and events. Successful predictions offer insights into housing markets, tourism trends, and economic dynamics, benefiting travelers, property owners, and policymakers. Personalized recommendations enhance user experiences, while accurate pricing impacts business performance and competitiveness. European cities are popular tourist destinations for millions of people around the world. So, they are ideal for predicting Airbnb prices due to high tourist demand, seasonal variations, complex regulations, rich data availability, and cultural diversity. Understanding these factors enables accurate pricing models that are important for both travelers and hosts.

We analyze various data sources and select ten of the most visited European cities for our project: Paris, London, Berlin, Vienna, Rome, Lisbon, Amsterdam, Athens, Barcelona, and Budapest [1], [2]. The problem in hand is a regression problem, and in this project, our objective is to evaluate multiple machine learning algorithms on predicting Airbnb listing prices in the chosen European cities. The paper is structured as follows: in section I, we introduce the problem. Section II discusses related works on the topic. Details regarding the dataset, the models, and the overall design of our system are demonstrated in section III. Next, in section IV, we present the performance of our models, evaluated by standard machine learning model performance evaluation metrics. Finally, we conclude the paper by discussing the overall aspects of our project and possible future extension in section V.

II. Literature Review

There are a number of existing works that have addressed the problem of predicting Airbnb listing prices. Gyodi and Nawaro [3] analyze the determinants of Airbnb listing price in 10 major European cities, and find that Airbnb prices are spatially dependent, and accessibility of tourist attractions is a major price driver. Ahuja et al. [4] perform extensive feature engineering on their collected data, determine the top contributing features and focus on predicting rental prices of listings in various cities of California. Luo et al. [5] predict Airbnb prices in three cities: New York City (NYC), Paris and Berlin, and compare various models to determine the best performing one. Kalehbasti et al. [6] undertake a similar task, but they evaluate more models for an extensive comparison. However, they evaluate their models on a logarithmic scale of the price. Choudhary et al. [7] analyze Airbnb data of San Francisco and NYC to identify features that are more important for setting price of a new listing and the likelihood of listing availability.

From the literature review, we find that the most used models for the task are linear regression (with and without regularization), support vector regression (SVR), random forests, gradient boosting and neural networks. Based on the literature review, we predict listing prices in multiple European cities that are popular tourist destinations.

III. Methodology

A. System Design

The system design for the project consists of phases that are found in any standard machine learning project (Fig. 1). The first phase is data collection, where we collect the required dataset from a verified source. Next, the second phase is data preprocessing, where we prepare the data for analysis. The third phase of the project is feature engineering, where we perform operations such as one hot encoding to transform categorical features to numerical features and feature scaling to ensure all the features have similar values. Then, the fourth phase is model selection where we choose appropriate models for the task. In the fifth phase, we split the dataset into training set and test set, and we train the selected models on the training set and test them on the test set. In the final phase of the project, we take the output values and evaluate the models using standard performance metrics.

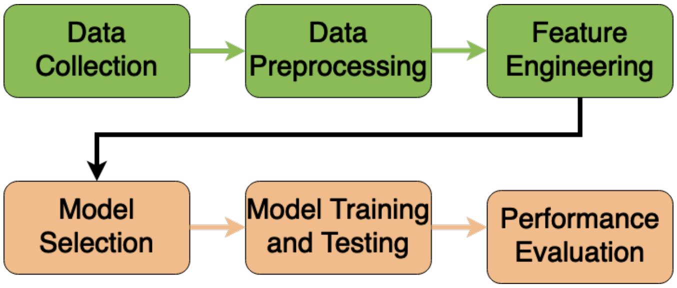

Fig. 1. System design flowchart of the project.

B. Required Software

We choose Python as the programming language to implement our project. For data analysis, we use the Python libraries pandas [8] and NumPy [9]. Pandas offers an extensive set of functions for data cleaning, transformation, and aggregation, making it essential for preparing data for analysis and modeling. On the other hand, NumPy offers powerful tools for performing efficient numerical computations, which are essential for scientific and engineering tasks. To implement the machine learning algorithms, we use the python library Scikit-learn [10], which provides a range of tools for data preprocessing, model selection, training, and evaluation. For data visualization, we rely on the Matplotlib [11] library, which is widely used for preparing publication quality graphics.

C. Dataset and Features

The dataset [12] is collected from Kaggle, and it is the dataset on which the work [3] is based. However, we process the dataset according to our specific task.

The dataset contains data of Airbnb listings in 10 European cities. The data is divided into weekday and weekends and organized in 20 comma separated values (CSV) files. There are three types of features: numerical, categorical, and boolean. We append two columns to each file, namely 'week_day' that describes whether the data is for a weekday or weekend, and 'city' that contains the name of the city the listing is in. Then we combine the 20 files into one to prepare the base dataset for our project. Description of the dataset is given in Table I. Then. we apply data preprocessing operations on this base dataset. first, we drop the index column, and the columns 'attr_index_norm' and 'rest_index_norm' as these are just normalized copies of 'attr_index' and 'rest_index' respectively. Then, we apply one hot encoding on the dataset to convert categorical features into numerical. After that, we determine

TABLE I
DESCRIPTION OF THE FEATURES IN THE DATASET.

Feature name	Description
realSum	total price: label (numeric)
room_type	type of room (categorical)
room_shared	whether the room is shared (boolean)
room_private	whether the room is private (boolean)
person_capacity	maximum number of people (numeric)
host_is_superhost	whether the host is a superhost (boolean)
multi	does host have $2-4$ listings (boolean)
biz	does host have > 4 listings (boolean)
cleanliness_rating	cleanliness rating (numeric)
guest_satisfaction_overall	overall guest satisfaction rating (numeric)
bedrooms	number of bedrooms (numeric)
dist	distance from city center (numeric)
metro_dist	distance from nearest metro station (numeric)
attr_index	attraction index of the listing (numeric)
rest_index	restaurant index of the listing (numeric)
attr_index_norm	attraction index: scale to 100 (numeric)
rest_index_norm	restaurant index: scale to 100 (numeric)
lng	longitude of the listing (numeric)
lat	latitude of the listing (numeric)
week_time	time of the week (categorical)
city	name of the city (categorical)

the correlation between the current features (Fig. 2). As there are no strong correlations, we don't drop or manipulate any of the current features. Next, we visualize the original numerical features by plotting histograms of each feature (Fig. 3). We notice that there are some features, including the label, which have a right skewed distribution. On these features, we perform a log transformation. Finally, we standardize all the numerical features using the StandardScaler module provided by Scikit-learn (Fig. 4), which removes the mean and scales to unit variance. The standard score of a sample x is calculated as:

$$z = \frac{x - \mu}{s} \tag{1}$$

where μ is the mean of the training samples and s is the standard deviation of the training samples.

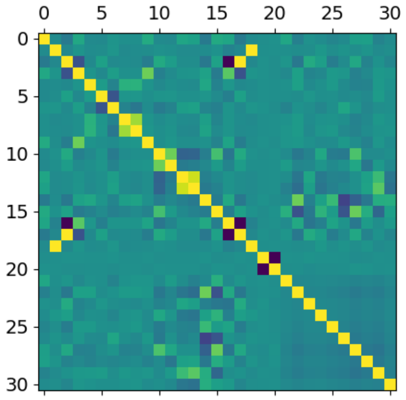

Fig. 2. Correlation between the label and the features, as well as among the features themselves. Index 0 represents the label.

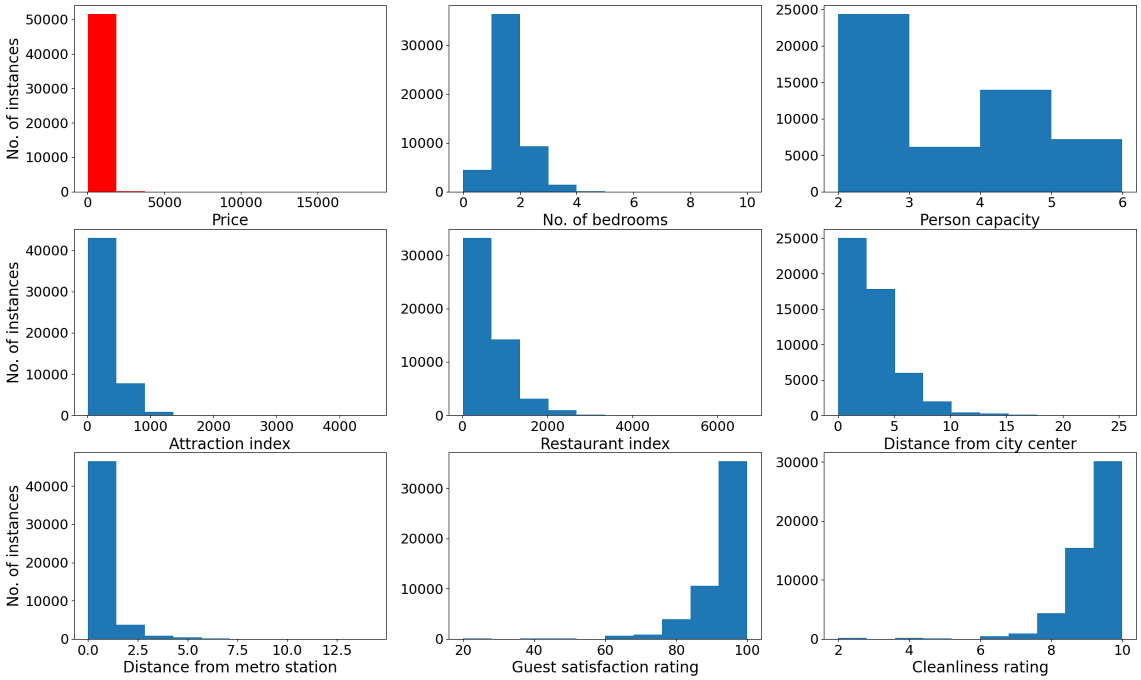

Fig. 3. Histogram plot of label and continuous features.

D. Models

We select four machine learning models for our task. A brief description of each of the selected models is given below.

1) Linear Regression: Linear regression is a supervised learning algorithm that provides a linear relationship between one or more predictor variable(s) and a response variable.

$$\hat{y}(x) = h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n \tag{2}$$

In our project, the response variable $\hat{y}$ corresponds to listing price, each predictor variable $x_1, x_2, \ldots, x_n$ corresponds to a feature that contributes to the price, and $\theta_0, \theta_1, \theta_2, \ldots, \theta_n$ are the regression coefficients. The best fitting model to the dataset is computed by minimizing the cost function defined as

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{3}$$

2) Ridge Regression: Ridge regression [13] is a technique for estimating coefficients in multiple regression models where the predictor variables are strongly correlated. It is mainly used to reduce errors caused by overfitting on training data. In case of ridge regression, the cost function is defined as

$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right] \tag{4}$$

where λ is the regularization parameter, which is a hyperparameter.

3) Support Vector Regression (SVR): Support Vector Regression (SVR) is a machine learning algorithm used for regression analysis, where the goal is to find a function $f(x) = wx + b$ that approximates the relationship between the input variables and a continuous target variable, and deviates from the actual observed values y by a value no greater than a specified margin ϵ. As defined in [14], the goal of SVR is the following optimization problem:

$$min_{w.b.\xi,\hat{\xi}} \quad C \sum_{i=1}^{n} (\xi_i + \hat{\xi}_i) + \frac{1}{2} ||w||^2 \tag{5}$$

subject to the following constraints:

$$y_i - (wx_i + b) \le \epsilon + \xi_i \tag{6}$$

$$(wx_i + b) - y_i \le \epsilon + \hat{\xi}_i \tag{7}$$

$$\xi_i, \hat{\xi}_i \ge 0 \tag{8}$$

where C is the regularization parameter, and $\xi, \hat{\xi}$ are the slack variables which represent the degree of deviation beyond the margin ϵ.

4) Decision Tree: Decision tree regression is a versatile machine learning algorithm that can perform both classification and regression tasks. It operates by recursively partitioning the data into segments depending on feature values, with the goal of minimizing errors in prediction at each step. For each node m, the weighted Mean Squared Error (MSE) is calculated, and

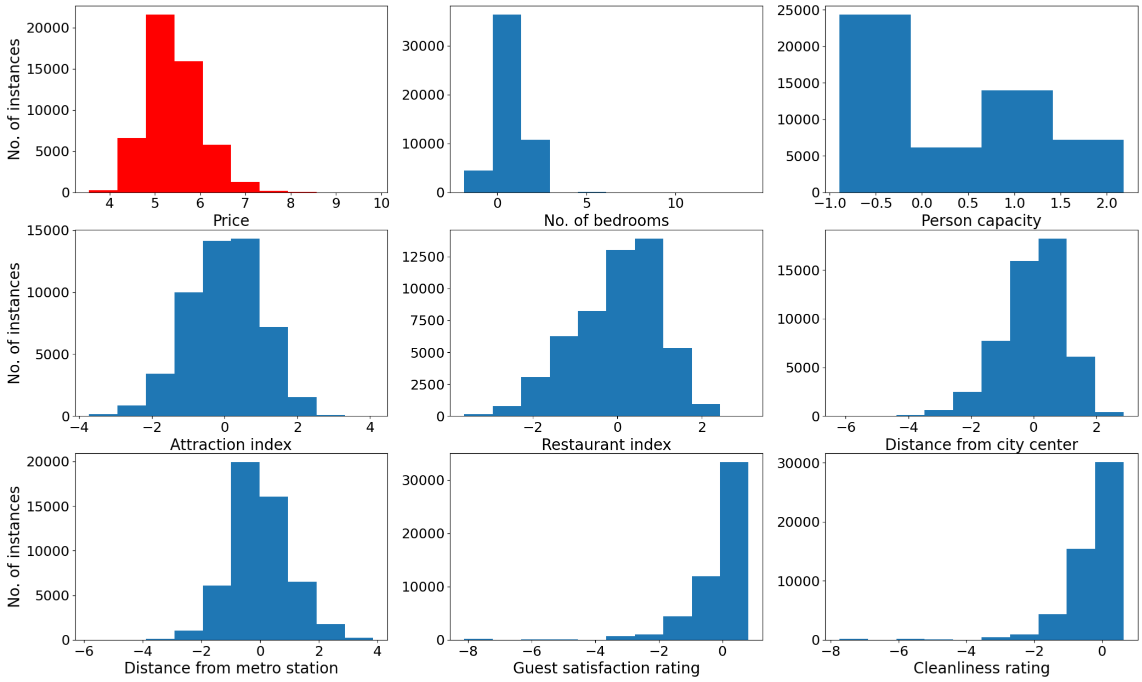

Fig. 4. Histogram plot of label and continuous features after log transformation and feature scaling.

the data is partitioned by choosing the split that minimizes the total MSE.

$$Total\ MSE_m = \frac{|D_{left,m}|}{D_m} \cdot MSE(D_{left,m}) + \frac{|D_{right,m}|}{D_m} \cdot MSE(D_{right,m}) \quad (9)$$

where $|D|$ is the number of samples in the dataset D, and $|D_{left}|$ and $|D_{right}|$ are the numbers of samples in the left and right subsets, respectively. MSE is defined as

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y})^2 \quad (10)$$

where y_i is the actual value, $\hat{y}$ is the predicted value, and N is the number of samples.

Besides these four models, we also choose two ensemble models. An ensemble model combines several individual models to produce more accurate predictions than a single model alone. The chosen models for this project are Random Forest and Gradient Boosting.

5) *Random Forest:* Random Forest [15] is an ensemble learning technique that uses multiple decision tree regressors to get a more reliable and accurate prediction. Random Forest Regressor usually improves predictive accuracy and controls overfitting compared to a single Decision Tree Regressor. The prediction equation used in Random Forest Regression is

$$\hat{y}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{y}_b(x) \quad (11)$$

where $\hat{y}(x)$ is the final prediction of the Random Forest model for a given input x, B is the total number of decision trees in the forest, and $\hat{y}_b(x)$ is the prediction made by the b-th decision tree for the input x. The final prediction $\hat{y}(x)$ is obtained by averaging the predictions from all B trees.

6) *Gradient Boosting:* Gradient Boosting [16] is an ensemble learning algorithm that combines several weak learners into strong learners, in which each new model is trained to minimize the loss function such as mean squared error or cross-entropy of the previous model using gradient descent. The prediction of the ensemble model at stage m is the sum of the predictions of the previous $m-1$ models plus the prediction of the current model.

$$\hat{y}_i^{(m)} = \hat{y}_i^{(m-1)} + \nu f_m(x_i) \quad (12)$$

where $\hat{y}_i^{(m)}$ is the prediction for the i-th instance at stage m, $\hat{y}_i^{(m-1)}$ is the prediction for the i-th instance from the previous stage, ν is the learning rate, a small positive number that controls the contribution of each model, and $f_m(x_i)$ is the prediction from the m-th model (weak learner).

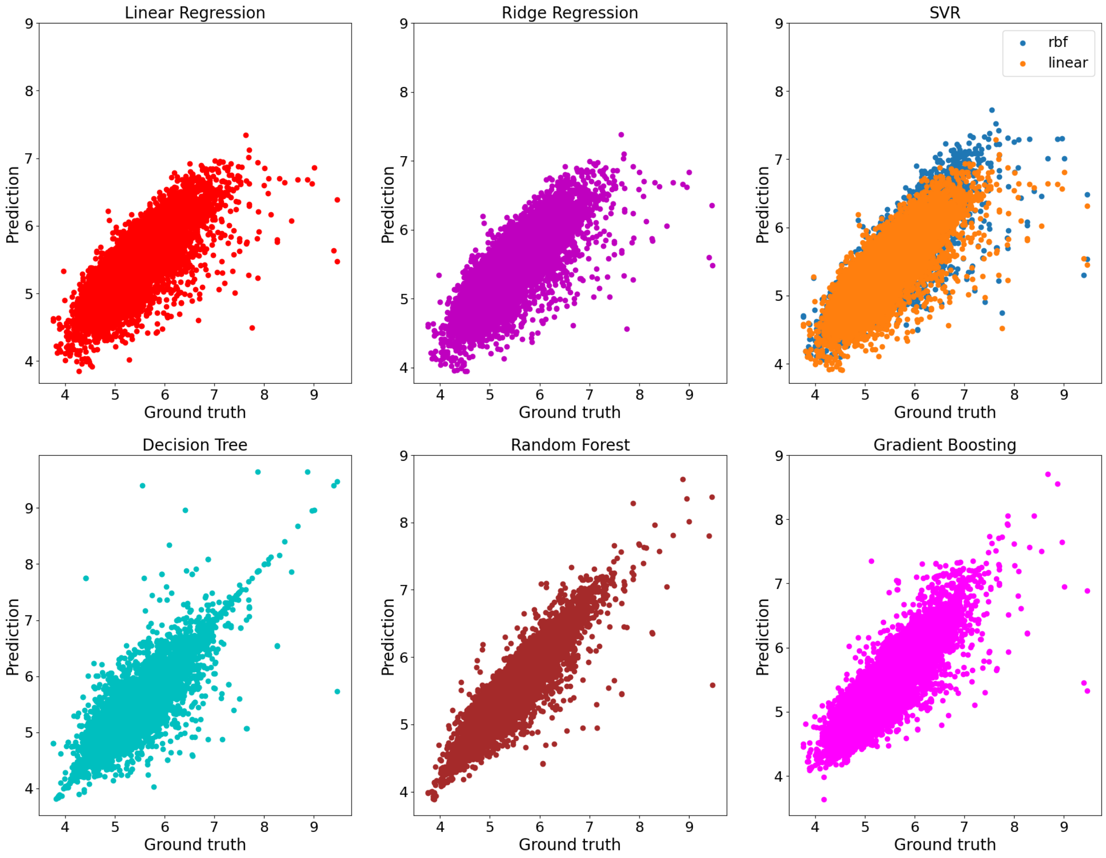

Fig. 5. Prediction vs ground truth on the test set, for each model.

E. Evaluation Metrics

We use three evaluation metrics to evaluate the performance of the models.

1) R-squared: R-squared is a measure that provides information about the goodness of fit of the regression model. In simple terms, it is a statistical measure that tells how well the plotted regression line fits the actual data. R squared measures how much the variation is there in predicted and actual values in the regression model.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y}_i)^2} \tag{13}$$

2) Root Mean Squared Error (RMSE): The Root Mean Squared Error (RMSE) measures the average difference between values predicted by a model and the actual values. It provides an estimation of how well the model is able to predict the target value (accuracy). A perfect model (a hypothetical model that would always predict the exact expected value) would have a Root Mean Squared Error value of 0.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{14}$$

3) Mean Absolute Error (MAE): Mean Absolute Error is a regressive loss measure looking at the absolute value difference between a model's predictions and ground truth, averaged out across the dataset. It is used when the goal is to evaluate the quality of predictions in terms of their absolute magnitude, rather than their relative magnitude.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{15}$$

IV. Results

We split the dataset into training set and test set by splitting it in a 80 : 20 ratio. Then, we trained and tested the models explained in section III-D. The experimental results are presented in Table II. We use Linear Regression as the

TABLE II

PERFORMANCE EVALUATION OF SELECTED MODELS ON TRAINING SET AND TEST SET.

		Train			Test		
Model	Hyperparameters	RMSE	MAE	R^2	RMSE	MAE	R^2
Linear Regression		277.060	78.523	0.664	301.513	83.227	0.656
Ridge Regression	α = 1.0	277.417	78.863	0.661	302.037	83.639	0.653
SVR	kernel = 'rbf'	263.861	64.237	0.760	288.201	72.986	0.729
	kernel = 'linear'	278.984	78.605	0.658	303.824	83.459	0.649
Decision Tree		1.74×10^{-14}	7.38×10^{-16}	**1.0**	276.548	52.958	0.766
Random Forest	n_estimators = 100	152.909	20.313	0.979	227.065	51.915	**0.854**
Gradient Boosting	n_estimators = 100, learning_rate= 1.0	245.376	65.435	0.769	270.408	73.368	0.734

baseline, and compare other models to it using the R^2 metric. We find that the performances of Linear Regression and Ridge Regression are very close, but Ridge Regression performs minutely worse. SVR with 'rbf' (radial basis function) kernel improves the performance by a good margin. However, SVR with 'linear' kernel further drops the performance. On the training set, Decision Tree achieves a perfect score on all metrics, and improves the performance on the test set. Random Forest also performs very well on the training set and achieves the best performance among all on the test set. Gradient Boosting improves the performance from the baseline, but falls short compared to Decision Tree and Random Forest. The results of each model is visualized in Fig. 5.

Thus, Decision Tree has the best R^2 score on the training set, and Random Forest has the best R^2 score on the test set. The results indicate that tree-based models are an ideal choice for the regression problem at hand.

V. CONCLUSION

In conclusion, the Random Forest algorithm emerged as the best-performing model for predicting Airbnb listing prices in popular tourist destinations across Europe, demonstrating superior accuracy and robustness in capturing the complex pricing dynamics. The model's high performance underscores its potential to provide valuable pricing insights to hosts and travelers alike. For future work, we aim to experiment with neural networks, which may further enhance prediction accuracy by capturing more intricate patterns within the data. Additionally, we plan to scale the project globally, extending our analysis to a broader range of tourist destinations worldwide, thereby increasing the model's applicability and utility in the global hospitality market. Through these advancements, we aspire to refine our predictive capabilities and offer even greater value to the Airbnb community and the hospitality industry at large.

DATA AND CODE AVAILABILITY

The dataset can be downloaded from this link. The code for this project can be found at this link.

ACKNOWLEDGEMENTS

Many of the concepts learned in the course CSE445: Machine Learning were used in this project, and we would like to express our sincere gratitude to the faculty Mr. Shahriar Hussain, Senior Lecturer, ECE Department, NSU for his expert guidance and skillful teaching.

REFERENCES

[1] K. Schmidt, "The 10 most visited cities in Europe," dw.com, 27-Sep-2023. [Online]. Available: https://www.dw.com/en/the-10-most-visited-cities-in-europe/g-66897729.

[2] L. D. Redman and L. Matthews, "The 10 best European cities to visit or live in - AFAR," AFAR Media, 05-Mar-2024. [Online]. Available: https://www.afar.com/magazine/europes-best-cities.

[3] K. Gyodi and L. Nawaro. "Determinants of Airbnb prices in European cities: A spatial econometrics approach," in Tourism Management, vol. 86, pp. 104319, 2021.

[4] A. Ahuja, A. Lahiri, and A. Das, "Predicting Airbnb rental prices using multiple feature modalities," arXiv.org, 13-Dec-2021. [Online]. Available: https://arxiv.org/abs/2112.06430.

[5] Y. Luo, X. Zhou, and Y. Zhou, "Predicting Airbnb listing price across different cities," 2019. [Online]. Available: https://www.semanticscholar.org/paper/Predicting-Airbnb-Listing-Price-Across-Different-Luo-Zhou/5aea98236bcc318a71f42c91312c5f948717d686.

[6] P. R. Kalehbasti, L. Nikolenko, and H. Rezaei, "Airbnb price prediction using machine learning and sentiment analysis," in Lecture notes in computer science, 2021, pp. 173–184.

[7] P. Choudhary, A. Jain, and R. Baijal, "Unravelling Airbnb predicting price for new listing," arXiv.org, 25-May-2018. [Online]. Available: https://arxiv.org/abs/1805.12101.

[8] W. McKinney, 'Data Structures for Statistical Computing in Python', in Proceedings of the 9th Python in Science Conference, 2010, pp. 56–61.

[9] C. R. Harris et al., 'Array programming with NumPy', Nature, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

[10] Pedregosa, F., et al. "Scikit-learn: Machine Learning in Python," in Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[11] J. Hunter. "Matplotlib: A 2D graphics environment," in Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.

[12] K. Gyódi and Ł. Nawaro, "Determinants of Airbnb prices in European cities: A spatial econometrics approach (Supplementary Material)". Zenodo, Jan. 13, 2021. doi: 10.5281/zenodo.4446043.

[13] A. Hoerl, R. Kennard. "Ridge regression: applications to nonorthogonal problems," in Technometrics, vol. 12, no. 1, pp. 69–82, 1970.

[14] C. M. Bishop, Pattern recognition and machine learning. Springer Verlag, 2006.

[15] L. Breiman. "Random forests," in Machine learning, vol. 45, pp. 5–32, 2001.

[16] J. Friedman. "Greedy function approximation: a gradient boosting machine," in Annals of statistics, pp. 1189–1232, 2001.