Estimating Airbnb prices using Machine Learning Algorithms

Haja Amir Rahman (p2100803)
School of Computing
Singapore Polytechnic, Singapore
amirrahman517804@gmail.com

***Abstract*—The study is focused on the estimation of Airbnb prices using Machine Learning Algorithms. The final pipeline achieves high performance of 0.55 R2 Test Score with a relatively complicated model with long training time.**

I. Introduction

Airbnb operates an online marketplace for lodging, primarily homestays for vacation rentals, and tourism activities. Based in San Francisco, California, the platform is accessible via website and mobile app. Airbnb does not own any of the listed properties; instead, it profits by receiving commission from each booking. Airbnb has regulations to follow in Europe, United States, Japan, Singapore, and China.

II. METHADOLOGY

A. Data Collection

The dataset used for this study was obtained from Kaggle. The dataset contains 22552 records of Airbnb bookings

B. Geojson Map Data

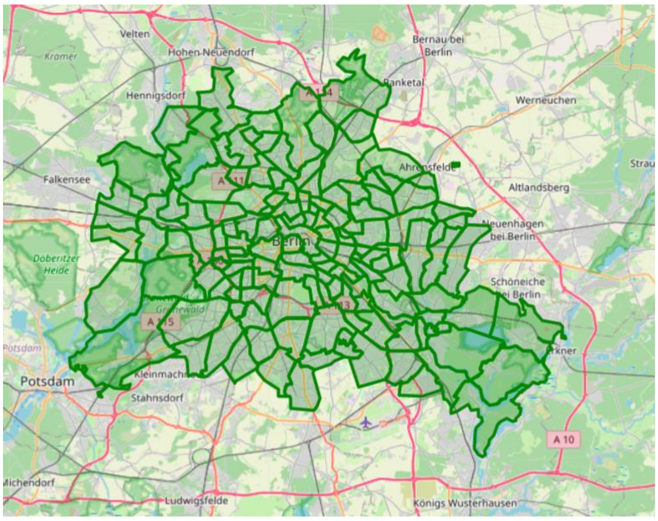


Fig 1 Map of Berlin, Germany

This map shows the geographical distribution of different places offering Airbnb in Berlin, Germany. We can see that there is a significant density of Airbnb outlets in the CBD (Central Business District) probably for people who are staying for a short while at their friends' place or other.

III. Exploratory Data Analysis (EDA)

Exploratory Data Analysis was performed using Pandas Profiling Report. The following are the finding results in the investigation of attributes that affects the prices of Airbnb prices.

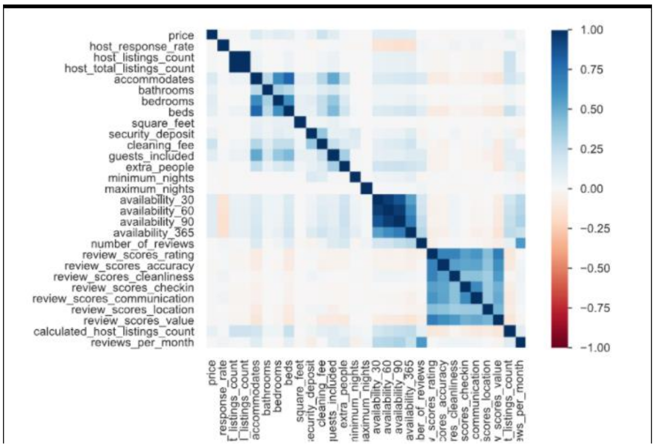


Fig 2 Pearson's Correlattion Plot of all the attributes against each other

A. Column Data Types

There were initially 96 columns in the dataset, but I manually removed the irrelevant attributes and only kept the relevant ones after doing some research. After the manual selection, I ended up with 48 columns with values of datatypes like integer, float64 and object.

B. Null Value Counts

To solve the problem of 0 or NaN values in the columns, I calculated the mean for the numerical columns and the mode for the categorical columns and appended the mean and mode into the 0 or NaN values in a 'for' loop. This way there will not be any outliers and all the values will be in range of the rest of the values.

C. Correlation Between Columns

Based on our feature importance table generated, the attribute that positively affect the price of Airbnb the most is 'property-type_Hotel' which translates to hotel type in layman terms.

D. Pandas Profiling Report

I generated a Pandas profiling report to simultaneously create all the distributions graphs as well as all the properties of all the attributes. The file is included as 'output_airbnb.html' inside the Section C folder.

E. Outlier Detection (Removed)

Isolation forest, local outlier factor, replacing outlier values with median (but made score worse so ignore). The reason

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

why no scaling was done was because when it was tried, it worsened the scores.

IV. Feature Engineering

A. One Hot Encoding
B. Normalization (removed)
C. Mathematical transformation (removed)

I used the ‘pd.getDummies()’ function to generate 0 ands 1 for all the categorical attributes containing non-numeric variables. Since this is a regression problem, all the values of all the attributes have to be numerical before being passed into the model to run.

V. Data Selection

Train test split

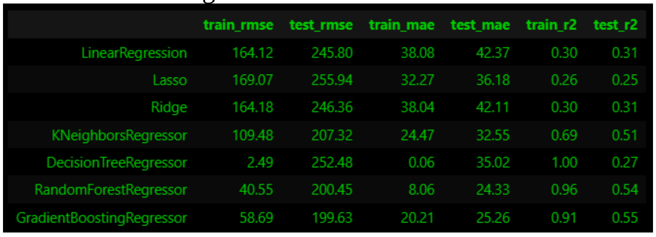
After splitting the datasets into train and test set with a ratio of 8:2, I bgean the process of model selection by trying out different family of models using the designed pipeline which is Features -> Numerical Features -> Estimator. I have decided to investigate the 4 different model families below:

1. Linear Models
   a. Linear Regression
   b. LASSO Regression
   c. Ridge Regression
2. Distance Based Models
   a. K-Nearest Neighbors Regressor
3. Tree-based Models
   a. Decision Tree Regressor
   b. Random Forest Regressor
   c. Gradient Boosting Regressor

The results of model selection and the decision for final predictor is included in the section below.

VI. Model Selection & Training

I created a custom function to perform model training on the train set to all the models mentioned above with out-of-the-box hyperparameters and evaluate the performance of the trained model using the test set.

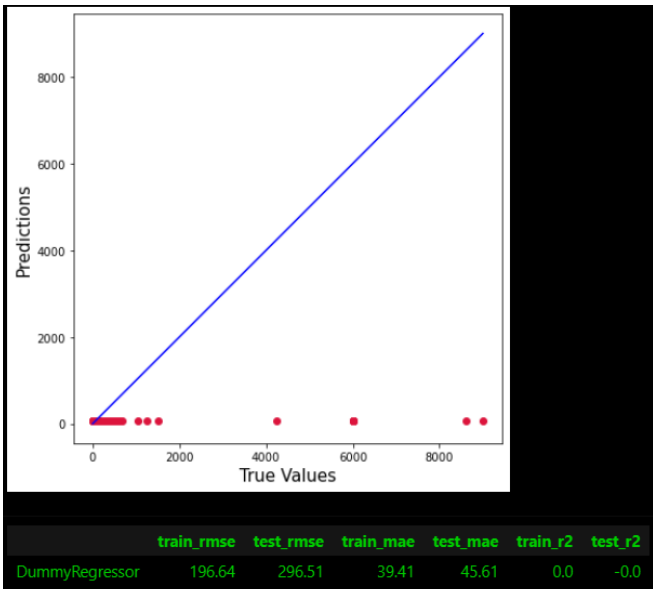


	train_rmse	test_rmse	train_mae	test_mae	train_r2	test_r2
LinearRegression	164.12	245.80	38.08	42.37	0.30	0.31
Lasso	169.07	255.94	32.27	36.18	0.26	0.25
Ridge	164.18	246.36	38.04	42.11	0.30	0.31
KNeighborsRegressor	109.48	207.32	24.47	32.55	0.69	0.51
DecisionTreeRegressor	2.49	252.48	0.06	35.02	1.00	0.27
RandomForestRegressor	40.55	200.45	8.06	24.33	0.96	0.54
GradientBoostingRegressor	58.69	199.63	20.21	25.26	0.91	0.55

Fig 3 Model selection

A. Baseline Model

For the baseline Model, I used Dummy Regressor which gave a score of -0.0 and to visualize and understand this, I created a function for Model Evaluation which outputs a graph of Prediction against True Values, and I could tell that the Dummy Regressor did very badly.



	train_rmse	test_rmse	train_mae	test_mae	train_r2	test_r2
DummyRegressor	196.64	296.51	39.41	45.61	0.0	-0.0

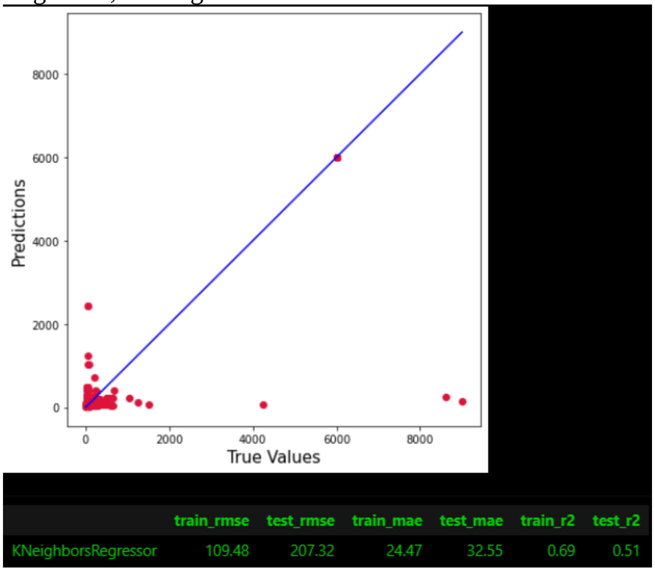
Fig 4 Dummy Regressor

B. Linear Model

For the linear model, I used Linear Regression which gave an r2 score of 0.31 for the test score

C. Distance-based Model

For the distance-based model, I chose K Neighbours Regressor, and it gave a r2 test score of 0.51



	train_rmse	test_rmse	train_mae	test_mae	train_r2	test_r2
KNeighborsRegressor	109.48	207.32	24.47	32.55	0.69	0.51

Fig 5 K-Nearest-Neighbors Regressor

D. Tree-based Model

For the tree-based models, they performed the best. I decided to use Random Forest Regressor and Gradient Boosting Regressor which gave a r2 test score of 0.54 and 0.55 respectively.

VII. Hyperparameter Tuning

With the aim of reducing overfitting of Gradient Boosting Regressor by increasing regularization, I've decided to perform Grid Search to search for the optimal hyperparameters to reduce high variance while retaining the low bias characteristics of our model.

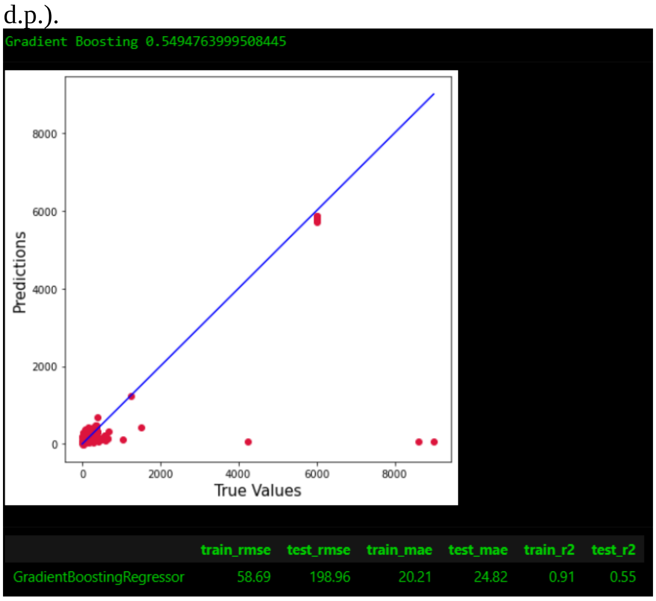
Due to time and computational resources constraint, I have only managed to perform grid search some models, although I wanted to try for SVR as well. The following is the set of hyperparameters that I have used to build my final Machine Learning pipeline.

a) n_estimators:100

For the hyper-parameter tuning, I created my own function to run a Grid Search CV on all the models I chose including Lasso and Ridge (which are not the main models so not mentioned). The best params for K Neighbors Regressor was 'n_neighbors:3', Random Forest Regressor was 'n_estimators: 100' and Gradient Boosting Regressor was 'n_estimators: 100'.

VIII. Model Evaluation

After performing the hyper-parameter tuning, the two best models that gave the best r2 test scores were Random Forest Regressor and Gradient Boosting Regressor of 0.54 and 0.55. The worst performing model was the Linear Regression Model with a r2 test score of 0.32 (rounded to 2 d.p.).



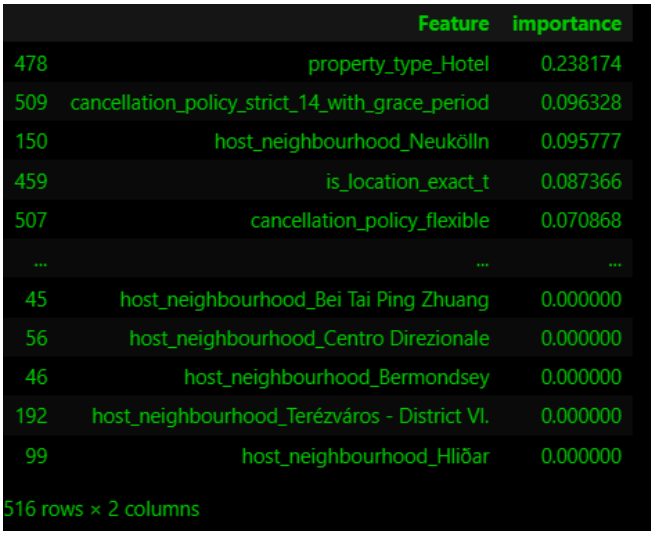
```
Gradient Boosting 0.5494763999508445
```

	train_rmse	test_rmse	train_mae	test_mae	train_r2	test_r2
GradientBoostingRegressor	58.69	198.96	20.21	24.82	0.91	0.55

Fig 6 Gradient Boosting Regressor Model Evaluation

Since I am using Gradient Boosting Regressor as my final model, we can visualize the model importance to investigate which features helps us better in modelling to investigate the relationship between the features and target variables.'

Fig 8 shows the Model Attributes like Hotel Property Type and Strict Cancellation Policy with Grace Period is significant in affecting the model's decision which is consistent with the findings by (Hati, S.R.H. et al., 2021). Moreover, Geographic Attributes like Longitude and Latitude are indispensable in the study of Airbnb Price Estimation.



	Feature	importance
478	property_type_Hotel	0.238174
509	cancellation_policy_strict_14_with_grace_period	0.096328
150	host_neighbourhood_Neukölln	0.095777
459	is_location_exact_t	0.087366
507	cancellation_policy_flexible	0.070868
...	...	...
45	host_neighbourhood_Bei Tai Ping Zhuang	0.000000
56	host_neighbourhood_Centro Direzionale	0.000000
46	host_neighbourhood_Bermondsey	0.000000
192	host_neighbourhood_Terézváros - District VI.	0.000000
99	host_neighbourhood_Hlištar	0.000000

516 rows × 2 columns

Fig 7 Feature Importance

IX. Conclusion

Overall, I have managed to build a machine learning model that can help us in estimating Airbnb prices with R2 test score of 0.55. Besides, based on EDA and Model Interpretation, I have identified several important attributes like Hotel Property Type and Strict Cancellation Policy and Geographical Attributes like latitude and longitude and CBD which are dominant towards the impact of Airbnb prices. The model can be improved with more computation resources for hyperparameter tuning as well as other Neural Network or Random Search CV that can better model non-linear relationships between the features and price.

References

[1] Bettendorf, B., 2019. Berlin Airbnb Data. *Kaggle.* Available at: https://www.kaggle.com/datasets/brittabettendorf/berlin-airbnb-data?select=listings_summary.csv [Accessed June 9, 2022].

[2] Anon, Get the Data. *Inside Airbnb.* Available at: http://insideairbnb.com/get-the-data/ [Accessed June 9, 2022].

[3] Anon, 2022. Airbnb. *Wikipedia.* Available at: https://en.wikipedia.org/wiki/Airbnb [Accessed June 9, 2022].

[4] Anon, Sign in. *RPubs.* Available at: https://rpubs.com/jeryl_goh/airbnb_SG [Accessed June 9, 2022].

[5] Hati, S.R.H. et al., 2021. A decade of systematic literature review on airbnb: The sharing economy from a multiple stakeholder perspective. *Heliyon.* Available at: https://www.sciencedirect.com/science/article/pii/S2405844021023252 [Accessed June 9, 2022].