Price Prediction for Airbnb Rentals

Akshay Rane*
arane1@stevens.edu
Stevens Institute of Technology
Hoboken, New Jersey

ABSTRACT

Airbnb community consists of hosts, who provide their guests with the unprecedented opportunity to explore places similar to a familiar neighborhood. When leasing out an apartment, the crucial element is fixing the price of the apartment. A variety of factors alter the price of Airbnb rentals rental namely Location, Type of room, the amenities provided by the host, etc. Machine Learning methods like Linear Regression, Ridge Regression, Random Forest Regression, K-Nearest Neighbors(kNN) Regression, and Decision Tree Regression can be used for the prediction and later, analyzed for the better results. It leverages machine learning models to predict the rental price by using the data from Inside Airbnb.

KEYWORDS

dataset, kNN, Linear Model, PCA, Random Forest, Decision Tree, nltk, MAE, RMSE

1 INTRODUCTION

Airbnb is an application that lets people rent apartments for short/long term to other people over the web.

This project will assist the host to predict the price of his rental.

2 RELATED WORK

In previous research, many different methods have been practiced. The one that has been cited implements two models General Linear Model (GLM) and Geographically weighted regression (GWR) to predict the prices for Airbnb rentals. The limitations of this work are that the root means squared error (RMSE) of both the models are relatively low; implying the features not examined thoroughly.[1] The cited paper predicts prices for Metro Nashville; this project will assist in predicting prices for Airbnb rentals in the city of Amsterdam(Netherlands). These prices differ across the city; this project will explore it.

3 DATA

Inside Airbnb site collects the data from sources, publicly available for the information of the Airbnb site. The data-set contains the price for each row; therefore, this classifies as a Supervised Machine Learning problem.

Number of rows: 19456
Number of columns: 105

Data Source

http://insideairbnb.com/get-the-data.html

Feature Selection

Selecting the features that are relevant to our prediction and omitting the features having primary key values (having distinct value), images, or sparse data.

*

Feature Name	Description
security_deposit	This field contains information about the security deposit for the rental.
cleaning_fee	This field contains information about the cleaning fee for the rental.
latitude & longitude	The latitude and longitude location of rentals
street	This field contains information about the city, state, and country of the rental location.
property_type	What is the type of Airbnb property? e.g. Apartments, Condominiums, Houses, etc.
accommodates	The number of guests the rental can accommodate.
bathrooms	Number of bathrooms included in the rental.
bed_type	Type of bed included in the rental like a futon, a real bed, airbed, etc.
amenities	The total number of amenities being offered by the rental, eg. T.V, A.C, Microwave, etc.
transit	The transportation available to reach the rental place.
beds	The number of bed that is included in the rental.
guest_included	The total number of guests that can reside in the rental place.

Target Variable

Variable Name: Price

Description: The price of rentals in dollars.

4 APPROACH

- Cleaning the data-set
  - The data-set is scanned for duplicate tuples to eliminate them and avoid redundancy.
  - The data-set is searched for derived attributes to exclude them; since these features can be deduced from others and their appearance might lead to over-fitting of the model.
  - Checked the data-set for features, that do not in any way is related to the price and eliminate them based on domain knowledge. Backward elimination is a viable option.
  - Derived new features like transit_count, amenities_count from textual data transit and amenity to have a meaningful impact on prediction.
  - Picked and eliminated the features that have more than 50% of sparse, missing or unknown values. Dropped features having low correlation with the target variable.
  - Outlier removal: Used the distance of neighborhood rentals, to eliminate the rentals not having the price within the range of its nearest neighbors(neighbors within 0.5 km of the specified rental).
  - Features such as names are categorical. The value of such a feature is not indicative of the price of the rental, but its category does. Therefore, transform these features into new sparse features (every discrete category).
  - Dimensionality reduction method PCA(Principle Component Analysis) is done to reduce the number of dimensions and project data to new k-dimensions, to avoid over-fitting.

- Running the model
  - Applied the regression models, Linear Regressor, Ridge Regressor, Decision Tree Regressor, kNN Regressor, and Random Forest Regressor by applying PCA on the dataset and also without it for comparing the results.
- Analysis:
  - The performance of the models implemented is identified by using Residuals or Root Mean square error.

5 EXPERIMENTAL DESIGN

5.1 Pre-Processing

5.1.1 Derived Features.

- **transit_count**
  This feature descends from the "transit" feature. The tokens for the modes of transport to the rentals are generated using nltk-tokenizer over the values in this feature. Every mode of transportation is assigned weights depending on its emphasis on price and summation of the weights of all modes of transportation available is assigned to the transit_count.

modes of transport	importance
bus	1
tram	1
foot	8
bicycle	8
bikes	1
cab	1
station	3
carrentals	1
bikerentals	2
tram-stop	4
metro	4
airport	3
taxi	2
transport	1
walk	8
walking	8
busstop	1

  The Nan values in this column are deduced by calculating the distance of such rentals to its neighborhood rentals (having a distance less than 0.5km) and assigning the value occurring most frequently in these neighbor rentals.
- **amenities_count**
  This feature descends from the "amenity" feature. The number of amenities mentioned in this feature is assigned as count.

5.1.2 Normalize features.

The features like cleaning_fee, security_deposit, and target variable price are of string type. These are converted to their float equivalents.

5.1.3 Categorical features.

The features like street, property_type, security_type, and bed_type are categorical features. Each of the categorical value is assigned numeric value describing the category by Label Encoder for regression.

5.1.4 Outlier removal.

The outliers are excluded from the data-set. They will cause dilemmas for the predictions. For every rental, calculate the closest neighborhood rentals (having a distance less than 0.5km) and check the target variable value, to be in the 10^{th} and 90^{th} percentile of prices of neighborhood rentals. If it doesn't lie in this range; it is classified as an outlier and eventually eliminated from the data-set.

5.2 Correlation of Features

5.2.1 HeatMap.

The heatmap below shows the correlation coefficients between all the features. From this, we learn that features like accommodates, bathrooms, and beds have a high correlation coefficient with the target variable price. Also, the feature room_type have high negative correlation coefficient stating that less luxurious rooms are priced at low prices.

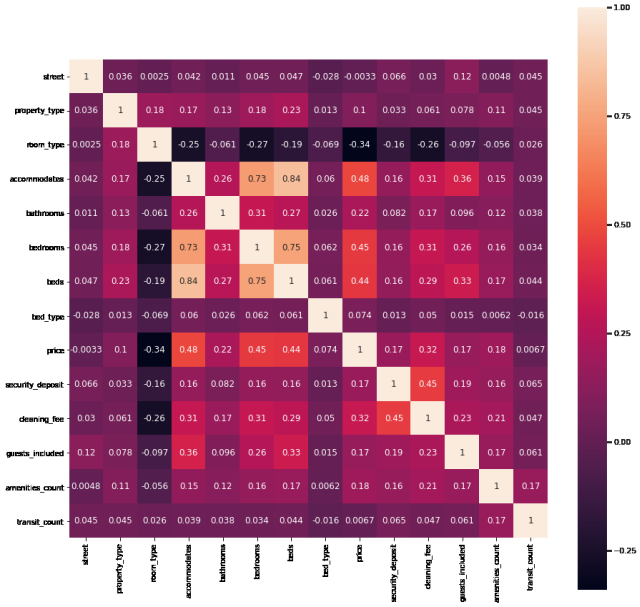

Figure 1: Heat Map
Heat map of chosen features

5.2.2 *Pairplot.*

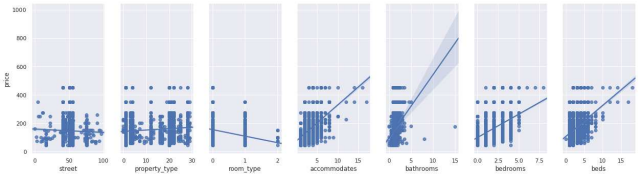

Figure 2: Pairplot
Pairplot of chosen features

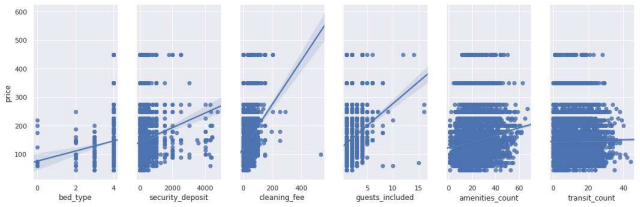

Figure 3: Pairplot
Pairplot of chosen features

The above pairplot representation gives us a picture of the feature values against price values. It also presents us with an idea of the spread of every feature and the frequency of different values. In general, we can observe that there is a general trend in each graph and there are a few points away from the general trend which can either indicate outlier or it can mean that those points have certain other features which contribute to their higher price.

5.3 Final Dataset

- Number of rows: 9860
- Number of columns: 14

	street	property_type	room_type	accommodates	bathrooms	bedrooms	beds	bed_type	price	security_deposit	cleaning_fee	guests_included	amenities_count	transit_count
0	55	1	0	3	1.0	1.0	1.0	4	125.0	300.0	40.0	2	19	5
1	55	21	1	2	1.0	1.0	1.0	4	150.0	0.0	0.0	1	21	14
2	55	1	0	4	1.0	3.0	3.0	4	219.0	0.0	60.0	4	26	5
3	51	1	0	2	1.0	1.0	1.0	2	145.0	300.0	0.0	2	22	1
4	55	1	0	2	0.0	1.0	1.0	4	180.0	150.0	40.0	2	31	13
5	55	3	1	2	1.0	1.0	1.0	4	159.0	300.0	35.0	1	22	17
6	55	1	0	4	1.0	2.0	4.0	4	210.0	0.0	35.0	2	14	1
7	55	1	0	2	1.0	1.0	2.0	4	100.0	300.0	50.0	4	35	7
8	55	1	0	3	1.5	1.0	1.0	4	250.0	100.0	60.0	2	28	13
9	51	1	0	4	1.0	2.0	3.0	4	200.0	0.0	50.0	2	30	5

Figure 4: Clean Data-set
Data-set after preprocessing

5.4 Hyper-parameters

5.4.1 *PCA.*

It uses the concept of eigenvalues and eigenvectors to find new orthogonal vectors which have minimum covariance between them but high variance, still keeping the effect of each column.
n_components = 12(Number of components to keep.[2])

5.4.2 *Linear Regressor.*

Ordinary least squares Linear Regression.
Hyper-parameter:
– fit_intercept=True
To calculate the intercept for this model.[7]
,

5.4.3 *Ridge Regressor.*

This model solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm.[3]
Hyper-parameter:
- alpha = 0.005
  Regularization strength improves the conditioning of the problem and reduces the variance of the estimates.[3]

5.4.4 *Decision Tree Regressor.*

Hyper-parameters:
- criterion=mse
  Measure the quality of a split.[4]
- max_depth=10
  The maximum depth of the tree., DTR
- max_leaf_nodes=500
  Grow a tree in best-first fashion.[4]
- min_samples_leaf=1
  The minimum number of samples required to be at a leaf node.[4]
- min_samples_split=2
  The minimum number of samples required to split an internal node.[4]
- splitter=best
  The strategy used to choose the split at each node.[4]

5.4.5 *kNN Regressor.*

Regression based on k-nearest neighbors.[5]
Hyper-parameter:
- n_neighbors = 10
  Number of neighbors to use for queries.[5]

5.4.6 *Random Forest Regressor.*

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.[6]
Hyper-parameter:
- n_estimators = 20
  The number of trees in the forest.[6]
- random_state=0
  Seed used by the random number generator.[6]

5.5 Evaluation

5.5.1 *K-Fold Cross Validation.*

K-Fold Cross Validation is a cross-validation technique that is widely used in machine learning. In K fold cross-validation the dataset is divided into k different subsets (or folds). (k-1) subsets of data are used to train the model and the last subset as test data.
The k-Fold value is set to 10 for carrying out all the experiments.

5.5.2 *RMSE.*

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.[8]

5.5.3 *MAE.*

Absolute Error is the amount of error in your measurements. It is the difference between the measured value and âĂIJtrueâĂİ value. The Mean Absolute Error(MAE) is the average of all absolute errors.[9]

6 EXPERIMENTAL RESULTS

6.1 Linear Regression

Measure	Without PCA	with PCA
RMSE	2475.03256	2475.2859
MAE	37.0081	37.0098

6.2 Ridge Regression

Measure	Without PCA	with PCA
RMSE	2475.03255	2475.2859
MAE	37.0081	37.0098

6.3 Decision Treee Regression

Measure	Without PCA	with PCA
RMSE	2946.8022	3474.6936
MAE	38.8128	41.4938

6.4 Random Forest Regression

Measure	Without PCA	with PCA
RMSE	2581.4773	2569.5409
MAE	37.8280	37.8312

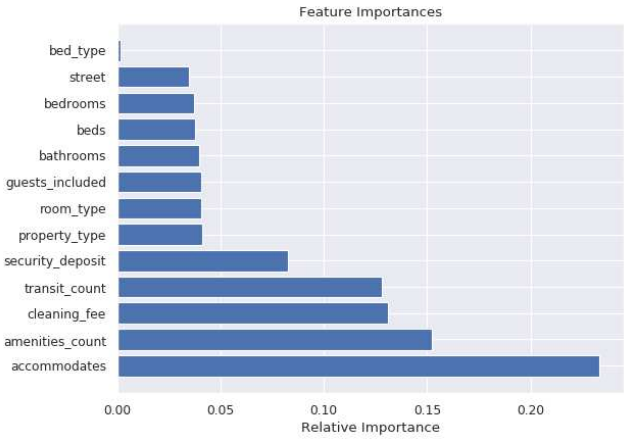

Figure 5: Without PCA
Random Forest Feature Importance

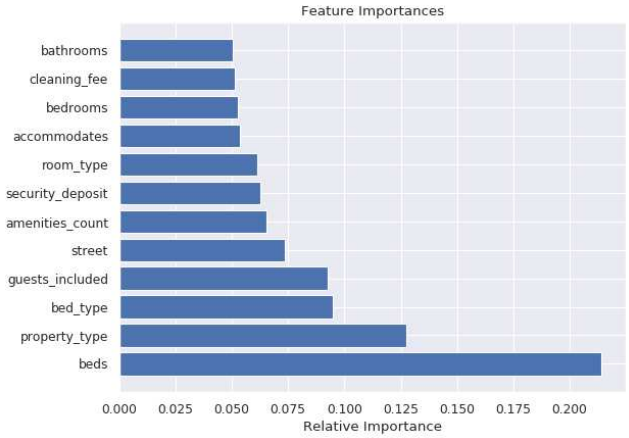

Figure 6: With PCA
Random Forest Feature Importance

6.5 k-Nearest Neighbor Regression

Measure	Without PCA	with PCA
RMSE	2550.4430	2571.5805
MAE	37.1830	37.1617

7 CONCLUSION

The result obtained from the experiments suggests that the Linear Regression Model and Ridge Regression Model perform better than others. These results are highly relevant; as we can observe that the features of data are linearly related to the target variable price. However, a more potent and practical argument is that the k-Nearest Neighbor Regressor also gives reliable performance. This model depicts how much the rental price is dependant on the rental prices of its nearby neighbors. In real-life scenarios, the rental prices should be set in par with neighborhood rentals. In this way, avoid losing to the competitors.

8 FUTURE WORK

In this research, the work has been done only on the rentals in the city of Amsterdam. It was observed from the heatmap that the correlation coefficient of the derived feature transit_count was very low. Therefore the feature wasn't able to influence the target variable as expected. Also, security_deposit and street had less influence. These two observations can be addressed by analyzing how the geographical location influence the price of rentals in any region. More knowledge of the region should be studied, to know popular places, most used means of transport, etc. By taking into account this consideration, the model will be able to predict better.

REFERENCES

[1] Zhihua Zhang 1, Rachel J. C. Chen 2,*, Lee D. Han 3,* and Lu Yang 1Key Factors Affecting the Price of Airbnb Listings: A Geographically Weighted Approach, https://www.mdpi.com/2071-1050/9/9/1635/pdf

[2] Principal Component Analysis, https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[3] Ridge Regressor https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

[4] Decision Tree Regressor, https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

[5] kNN Regressor, https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

[6] Random Forest Regressor, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[7] Linear Regressor, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[8] RMSE, https://www.statisticshowto.datasciencecentral.com/rmse/

[9] MAE, https://www.statisticshowto.datasciencecentral.com/absolute-error/