Nicholas Lichtsinn

IST 707

Project Report

**Airbnb Price Prediction**

**Introduction**

Airbnb is an online platform that allows homeowners and renters to put their properties online primarily as vacation rentals and homestays and has seen tremendous growth since its inception in August 2013 in San Francisco California. In December of 2020 Airbnb became a public company with an IPO that raised $3.5 billion, with stocks started the day at $68 a share and ending at $144.71 [1]. With competitors like VRBO and others also joining the niche home sharing market as well as hotel and other options there is a lot of competition to secure quests and keep them.

On Airbnb hosts set their own prices for their listings, although Airbnb and similar home-sharing sites provide recommendations based on a variety of factors, they cannot give an optimal price recommendation for each property that is listed on the platform. There are third-party options for hosts, however these require the hosts to input a base price to give any advice on seasonal or promotional pricing and cut into their bottom line. Airbnb pricing is an important aspect to each host as it can drastically change the number of guests each property has, especially in a crowded market.

Given the considerable number of options that customers have, and that 53% of respondents to the Morgan Stanley study use Airbnb because of the cost savings, this means that over half of Airbnb rentals are happening because of their price [2]. This coupled with the fact that only 11% of listings are reserved by customers on a typical night means that hosts who are not optimizing their pricing strategy are losing out on a lot of money, and there is clearly room for improvement [2]. This paper will attempt to solve this problem by using machine learning techniques with Python to predict the price bin for properties in six major cities in the United States.

**Analysis**

**The Data**

The data set used for this analysis is “Airbnb listings in major US cities” from a Deloitte machine learning competition on Kaggle.com [3]. The dataset consists of 74,111 rows and twenty-nine columns of features for each Airbnb listing. Some of the features include number of bedrooms, type of property, amenities, cancellation policy, reviews, and host information. The feature that will be predicted is log\_price. As the prices for this dataset are heavily skewed to the right creating a non-linear relationship between price and the other features that are being examined, the dataset takes the natural logarithm of the price to have log\_price follow a normal distribution as shown below.

**Chart, histogram

Description automatically generated**

Text

Description automatically generatedThere are twenty-eight features in the dataset excluding log\_price that will need to be addressed before any models are created. There are 24,392 missing values (na’s) for seven variables and variables that do not offer much information gain so they will be removed as well.

The features that were difficult to use, enumerate or proved unnecessary that were removed were id, description, amenities, first\_review, host\_response\_rate, last\_review, latitude, longitude, name, and thumbnail\_url. This left sixteen feature variables that could be used to predict log\_price.

For these sixteen remaining variables there were many missing observations shown below.

Graphical user interface

Description automatically generated with medium confidence

To manage the missing values for beds, bedrooms, and bathrooms a default of zero was used and for host\_has\_profile\_pic and host\_identity\_verified a default of False was used to maintain accuracy.

Neighbourhood had 6,872 counts of missing observations and there was not a default value that could be inserted without many hours of research or distorting the dataset, so these rows had to be dropped from the dataset.

Similarly review\_score\_rating had many missing observations (16,722), having a missing value here meant that this property had no reviews and would be fairly telling however to protect the relationship between log\_price and review\_score\_rating it was not possible to give them a default of 0 or the mean of the column and these observations were also dropped.

This left 52, 343 rows left for model creation and analysis after dealing with the missing observations.

For the machine learning models to work on this dataset all features needed to be represented numerically. To fix this, all Boolean features such as host\_identity\_verified, host\_has\_profile\_pic, instant\_bookable, and cleaning\_fee were transformed from True/False to 1 and 0.

The categorical features room\_type, property\_type, bed\_type, cancellation\_policy, city and neighbourhood were all enumerated using Label Encoding to assign a number for each unique type of the feature. Due to the substantial number of features in this dataset it was essential to determine which features were the most influential, a Correlation Matrix, Tree-based Feature Selection and Univariate Feature Selection models were used.

Text

Description automatically generatedThe Univariate model scores each feature based on univariate statistical tests. Using SelectKBest as the scoring program these were the top ten most influential features.

Chart, bar chart

Description automatically generatedFor the Tree-based feature selection the ExtraTreesClassifier was used to fit a randomized number of decision trees on sub-samples of the dataset to calculate the top performing features.

Chart

Description automatically generatedLastly the Correlation Matrix heatmap is below, showing a strong relationship between log\_price and room type, accommodates, bedrooms and beds.

The scores from each model were considered, and the following features were found to be the top ten important: bed\_type, city, instant\_bookable, room\_type, bedrooms, cancellation\_policy, review\_scores\_rating, bathrooms, beds, accommodates.

Because finding an exact price is not always necessary, determining an optimal price range will be used for these classification models. To do this log\_price was discretized into four categories low, medium, high, and very high. To do this both equal width and equal frequency binning methods were used, and the results will be shown for both methods. Equal width bins split the bins so that the width intervals are the same whereas equal frequency makes sure there are an equal number of observations in each bin.

**Decision Tree Models**

Text

Description automatically generatedGraphical user interface, application

Description automatically generatedFirst the Decision Tree Classifier was used to predict the bins based on all sixteen remaining features and on only the calculated top ten influential features. For these models’ multiple criteria were used to optimize the model, starting with max depths of 5 and 10 to limit overfitting, as the deeper the tree grows the more complex the model will become and the risk of overfitting the training sample increases. The default ‘best’ splitter was used to choose the best split at each node by evaluating all splits before splitting instead of randomly splitting at each node. Both Gini and Entropy criteria were used where Entropy maximized the information gain at each node and Gini measures the divergences between the probability distributions of the values to equivalent results.

Chart

Description automatically generated**Naïve Bayes Model**

Next a Gaussian Naïve Bayes model was used to predict log\_price, the advantage being fast processing and solving multiclass problems. This is a classification technique based on the Bayes theorem and the Gaussian model is used to support continuous values and assumes that each class is normally distributed. The algorithm predicts based on occurrences in the dataset so there is the ‘Zero Probability Phenomena’ when there is no class of something in the dataset. This occurred when using equal width bins and the Laplace transformation was used to smooth the results.

**Clustering Models**

Chart, line chart

Description automatically generatedK-Means clustering is one of the most popular methods used for classification problems. This model separates the data into clusters based on similar features and common patterns. K-Means Algorithm divides the dataset into k clusters based on similarity and mean distance from the centroid that subgroup formed. The optimal number of clusters was determined using the sum of squared error and elbow plots.

The elbow plots of all features and the top ten features were almost identical so only one is shown here.

Both indicated that 3-4 clusters would perform optimally. A K of 2, 3, 4, 6 and 10 was used in the analysis to show a range of performance.

Hierarchical clustering is another approach to clustering like k-means clustering for identifying groups in the dataset. For this model it is not required to specify the number of clusters to be generated as is required by the k-means approach. Hierarchical clustering results in a tree-based representation of the observations, called a dendrogram that uses Euclidean distance to show clusters.

**Results**

**Decision Tree Models**

Graphical user interface, table

Description automatically generatedAs can be seen from the figure shown here. The decision tree model performance was most impacted by binning technique due to a skewed number of occurrences landing in one bin so the model overfit and predicted most values to go in that bin. However, for the equal frequency bins there was a 1% increase in model performance when using only the top ten features while there was a decrease of 2% in the random forest model when using only the top ten features.

The models with equal frequency bins performed slightly better with a deeper tree and slightly worse with equal width bins. With all sixteen features and equal width bins the random forest model was the best performing model that was used and was the best performing when using all features with equal frequency as well performing 1% better than the individual decision trees.

**Naïve Bayes Models**

The Naïve bayes models performed slightly worse than the decision trees and random forest models. Interestingly for equal width bins the accuracy was higher with just the top ten features and increased by about 3% due to Laplace smoothing. For equal frequency bins the opposite was true with the model with all features performing 2% better and each model improved by only 1% after smoothing.

**Table

Description automatically generated**

**Cluster Models**

The K-Means models had an exceptionally low accuracy compared to both the decision trees and Naïve Bayes models. Clusters of 2, 3, 4 ,6, and 10 were used although with only two clusters both models had an accuracy of 0.0 so that is not shown.

**Table

Description automatically generated**

This is also the only model that saw a similar accuracy score between the equal width and equal frequency bins with the highest accuracy coming from four clusters and the equal width binning method. This model did not see a significant difference between using all features or only the top ten features, but with equal frequency binning there was some variance in accuracy using 6 and 10 clusters.

Hierarchical clustering was used and the dendogram out put is shown below. The dendogram can be divided into the four bins as the data shows however K-Means gives a much more intuitive result of the model accuracy as the dendogram utilized too many resources and shows more results in the lower two bins than expected.

Histogram

Description automatically generated

Using the dendogram to choose a two-cluster approach, the HAC plot shows that there is a lot of overlap between the clusters and could also explain why both clustering methods had a challenging time creating a model that was exceptionally accurate.

**Chart, scatter chart

Description automatically generated**

**Conclusion**

Airbnb prices are very volatile depending on several factors and there are still many things that can be done to find an optimal price point for each rental property. Given the dataset and the variables the vast majority of datapoints ended up in the high range of log\_price, this could be geographically measured from city center or using other location-based features like zip code with a lot more data processing on the front end to narrow down the data and create better results.

Most of the disparity in model accuracy comes from the binning method as equal width binning can group related points together better than equal frequency but can also fall victim to overfitting the model which was seen here. By taking the log\_price, prices are brought closer together and this can result in close observations being labeled differently which might be why the k-means model did not perform as well as the others.

Looking at the dataset and results from the different models, there are a few things that could have been done differently, one hot coding could have been used instead for label encoding but was not for fear of overfitting and running into the dummy variable trap. Using NLP techniques on the property description and reviews could also add some insight into the nature of the reviews and how a location is perceived to give higher accuracy to the models.

With the Decision Tree or Naïve Bayes model a host would be able to predict quite well the range of price that they should put their property for, especially if they live in a similar metro area to the cities that were in this dataset.

**References**

[1] “Airbnb by the Numbers: Usage, Demographics, and Revenue Growth” MuchNeeded [Online]. Available: https://muchneeded.com/airbnb-statistics. [Accessed March 2022]

[2] “Airbnb Now A $100 Billion Company After Stock Market Debut Sees Stock Price Double” NPR [Online]. Available: https://www.npr.org/2020/12/10/944931270/airbnb-defying-pandemic-fears-takes-its-company-public-in-ipo [Accessed March 2022]

[3] R. Mizrahi, “AirBnB Listings in Major US Cities.” Kaggle, March 14, 2018. [Online]. Available: https://www.kaggle.com/rudymizrahi/airbnb-listings-in-major-us-cities-deloitte-ml.