# CSCI-620 Data Mining with the Airbnb Dataset

[Exploring and Mining the dataset of NY Airbnbs]

Aishwarya Rao
ar2711@rit.edu

Apurav Khare
ak2816@rit.edu

Martin Qian
jq3513@rit.edu

Prateek Kalasannavar
pk6685@rit.edu

## ABSTRACT

The project aims to create a prediction model on the New York Airbnb Dataset that is capable of predicting which price group a particular house falls into. The approach we use is to build a classification model (such as decision tree) to predict a discrete version of the price (For instance, expensive vs cheap). One of the deliverable this project will include is determining what factors affect the price.

## 1. THE DATA MINING TASK

As mentioned in the previous phase, the data mining task selected was to build a classification model that is capable of predicting the price group a listing would fall into.

### 1.1 Targeted Knowledge

This data mining task is based on a number of factors. One, a model capable of predicting the price of a new listing has real-world applications. A landlord looking to put up a listing for a new house in New York would be able to use this model to determine the expected range of prices for his house and price it accordingly to make the most profit. Two, the exploration of the dataset revealed that there are a number of attributes in the data that correlate to the price. This indicates that these attributes do help determine the price and validates our hypothesis that the price can be predicted reasonably well by a data mining algorithm.

## 2. DATA EXPLORATION AND VISUALIZATION

Correlation plots were plotted on attributes against the price to see how they impact it. As a starting point, the attributes that have business meaning in the context of the data are used to plot the correlation plots. These attributes are: ”neighbourhood”, ”neighbourhood_group”, ”room_type”, ”minimum_nights”, and ”availability_365”.
Following are the notable observations:

1. The price of an Airbnb listing varies by the neighbourhood group that it is in. As observed from the correlation plot, the prices of the Airbnb listings in Brooklyn and Manhattan are relatively higher than those in Broknx, Queens, and Staten Island, with the listings in Staten Island being the least priced. Figures 1 depicts these result.

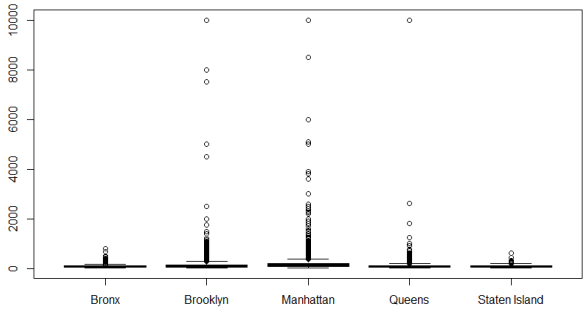


**Figure 1: Correlation between neighbourhood group and price**

2. The room type of the Airbnb listing affects its price. There are three room types in the dataset: Entire home/apartment, Private room, and Shared room. The correlation plot in Figure 2 clearly shows that Entire home/apartments are priced higher than private rooms, and Shared rooms are the least expensive of them all.

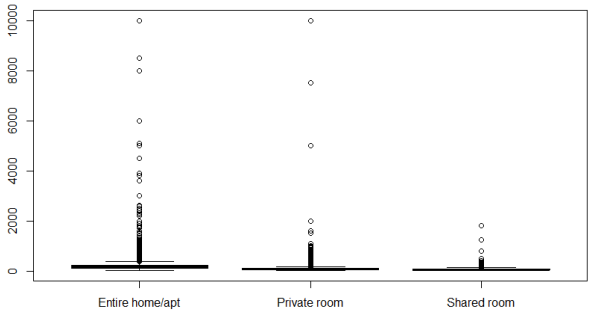


**Figure 2: Correlation between room type and price**

3. The neighbourhood impacts the price of the Airbnb listing as well. From the correlation plot, it is observed

that listings in neighbourhoods like "Upper West Side", "Upper East Side", and "Easts Harlem" have a higher price compared to those in other areas. Figure 3 shows the correlation plot.

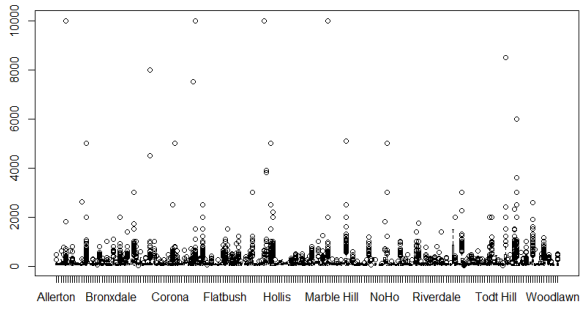


**Figure 3: Correlation between neighbourhood and price**

4. Some attributes which initially seemed to have business meaning did not have a strong correlation to the price. These attributes are "availability_365", and "minimum_nights".

Figure 4 depicts the summary of all the above correlation plots with the price, which will help choose the right features for the model.

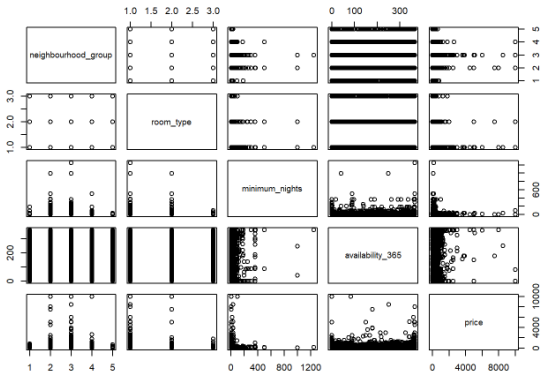


**Figure 4: Correlation of price between other attributes**

## 2.1 Outcomes of Data Exploration and Visualization

Data exploration and visualization provided a clear understanding of the type of the data and its quality, which provides a clear idea of the preprocessing steps required to make the data more suitable for the data mining task. These results are briefly described below:

**Type of the data:** The attributes that have an impact on the price are of varying types. For example, "neighbourhood" and "neighbourhood_group" contain discrete values which can be categorized, and "room_type" is a special categorical attribute where each category carries a meaning of "level" with it. In addition to these, the target label, "price", is itself a continuous attribute which can be categorized given the uneven spread of its values across the records. This can be observed in the histogram in Figures 5 and 6

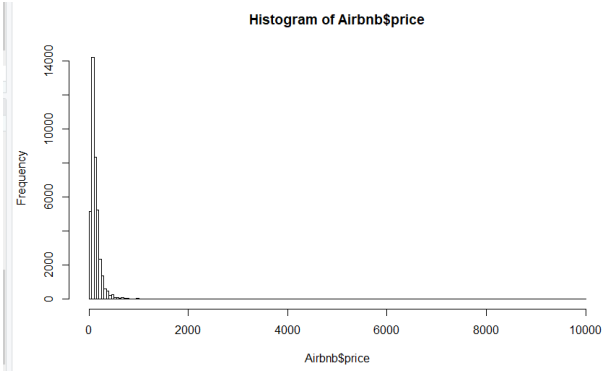


**Figure 5: Distribution of the price**

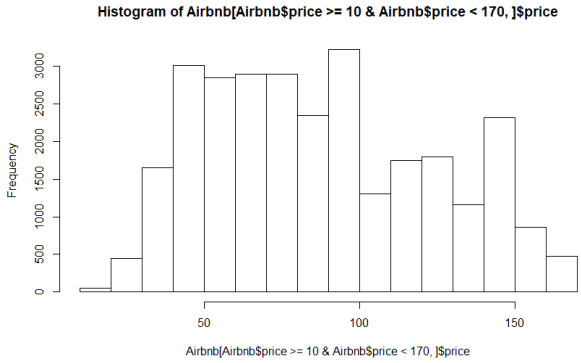


**Figure 6: Distribution of the price between the first and third quantiles**

**Quality of the data:** It was observed that the data contains certain records with data missing on attributes that might impact the data mining result, such as "room_type", and "availability_365". Also, a few records were found with 0 price. It was decided that these records would be considered spurious, and would be removed from the data fed to the classifier. The summary of price is described in Figure 7.

```
# price summary
summary(price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    69.0   101.0   142.3   170.0 10000.0
```

**Figure 7: Summary, and spurious values for price**

# 3. CATEGORIZING PRICE

Price the New York Airbnb Dataset is a continuous attribute with the following details -

Min. : 10.0
1st Qu.: 69.0
Median : 101.0
Mean : 141.8
3rd Qu.: 170.0
Max. :10000.0
To perform a classification task on the price, this attribute has to be made categorical. We used quartiles to create four different classes ranging from cheap to expensive. The quartiles were as follows, Category 1 : 25% (10 - 69)
Category 2 : 50% (70 - 101)
Category 3 : 75% (102 - 170)
Category 4 : 100% (171 - 10000)
Figure 8 shows the distribution of the data after the categorical split. As seen below, the distribution is well balanced with all classes having approximately the same number of instances, eliminating any class skewed dataset problems.

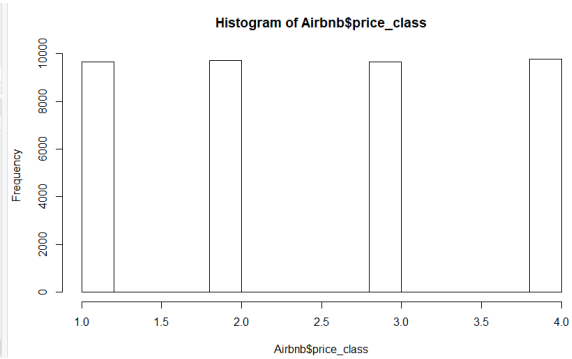


**Figure 8: Distribution of price after categorizing**

# 4. CLEANING THE DATASET

Since the dataset consists of both missing and inconsistent values, the first stage in the data mining process is to clean or eliminate these records. The total number of records in the dataset is 48,895 records. 10,052 records had missing values in last_review and reviews_per_month. However, since we foresee some significance in using these attributes to determine price, these missing records have to be eliminated to prevent incorrect data. The size of the dataset after eliminating the missing values is 38,843 records. Further, there are 10 records in the dataset that have the price value as 0. While these data records may be accurate, they form outliers that can interfere with the model's accuracy. The final dataset consists of 38,833 records.

# 5. FEATURE ENGINEERING

## 5.1 Dimensionality Reduction

In an attempt to reduce the dimensions of the dataset, attributes that are unlikely to contribute to the prediction of the price group without intensive preprocessing are eliminated. For instance, while the host name and the description of the listing could indirectly contribute to the price (An unknown pattern that could reveal that customers are more likely to pick a specific listing based on description - old ancient house versus new house in the suburb), these features are not easily found through simple classification techniques.

## 5.2 Factoring attributes

Factoring attributes allows R to encode the vector (in this case, the attribute column) into a categorical attribute. For the attributes room type, neighborhood, neighborhood group are all categorical values and hence these are factored to make them understandable to an R algorithm. The attribute availability 365 is a real valued number. Since this value would not easily be used to make a decision tree split, the attribute is split into 3 bins based on the 33rd and 66th percentile and then factored.

## 5.3 Leveling attributes

Leveling an attribute in R allows categorical attributes to be ordered in a meaningful manner. For instance, the attribute room type is ordered for 3 levels.

- Level 1 - Shared room
- Level 2 - Private room
- Level 3 - Entire home/apartment

# 6. TRAIN AND TEST SPLIT

The final stage of the preprocessing is splitting the dataset into train and test in a 80-20 ratio. The dataset is shuffled to ensure that there is representation of the data distribution in both the parts. As seen in figures 9 and 10, the distribution for all categories of price group remains the same in both train and test. All models that will be built will use train and test accuracy as a metric to determine performance. These accuracy measures will also be used to identify potential problems such as overfitting and underfitting if they occur and in turn work towards a model that eliminates such issues.

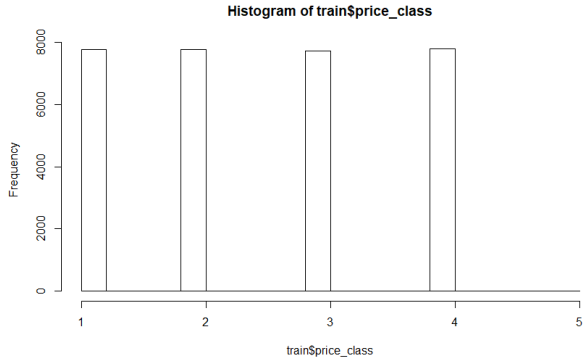


**Figure 9: Distribution of price in Train**

# 7. CHOOSING THE CLASSIFICATION MODEL

The next phase involved trying three different classification models to determine which one is more effective in modeling the dataset. The three considered were decision trees, K nearest neighbors, and Naive Bayes algorithm.
For the purpose of testing with the raw dataset but still performing a classification tasks, the following modifications were made to the dataset.

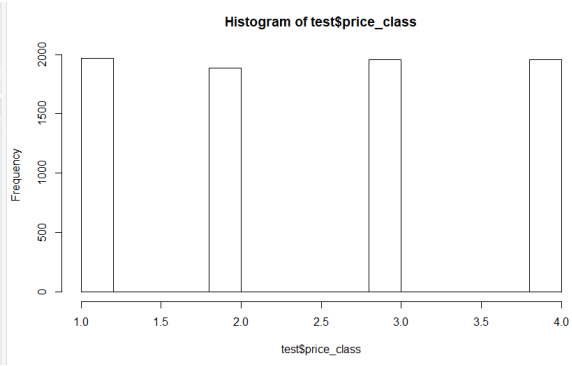


**Figure 10: Distribution of price in Test**

- The outliers in the price column were eliminated and price was split into two categories - no more than 100 and greater than 100.
- Only attributes with numerical values that required no preprocessing were used
- The train and test split ratio was 9:1.

The three algorithms were fitted to this dataset using R and the following documents the results obtained from the same.

- Decision Tree : Accuracy 72%
- K Nearest Neighbors : Accuracy 64% (k =20)
- Naive Bayes : Accuracy 68%

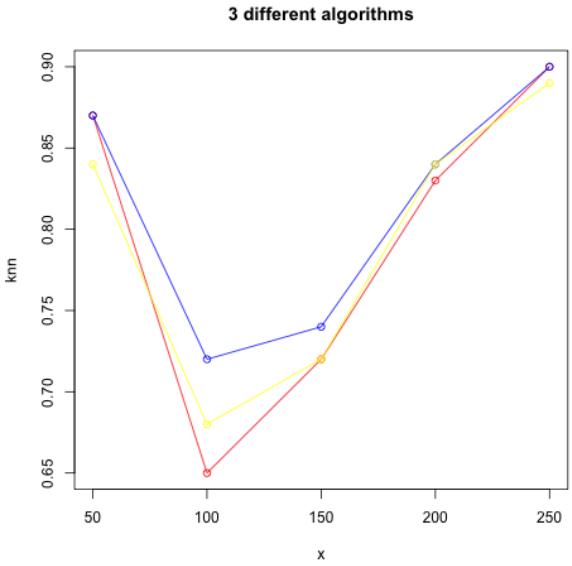


**Figure 11: Accuracy performance - Decision Tree (Blue), Naïve Bayes (Yellow), kNN (Red)**

Figure 11 depicts the performance of all three algorithms on the dataset. Through these results, we conclude that the decision tree algorithm is the most suitable for the New York Airbnb dataset and the next phase will be using only this algorithm.

# 8. THE DECISION TREE

A decision tree is constructed on the resulting data from our preprocessing stage. To summarize, the tree uses the attributes listed below to predict the outcome of the price group (ranges from 1 to 4)

- Neighbourhood
- Neighbourhood group
- Room type
- Minimum nights
- Number of reviews
- Number of listings by host
- Reviews per month

The decision tree is built using the rpart model available from R. RPart stands for Recursive Partitioning And Regression Trees and requires only one compulsory parameter to build the model.
Formula : This parameter indicates the value to be predicted and all the attributes that must be used to predict it

## 8.1 Refining the decision tree using preprocessed data

- Our first decision tree is built on only the raw categorical data to see how the model performs.
  The categorical attributes used for this is the neighbourhood group and the room type.
  Figure 12 depicts the decision tree built using only categorical data. Its accuracy is at 51.95056%.

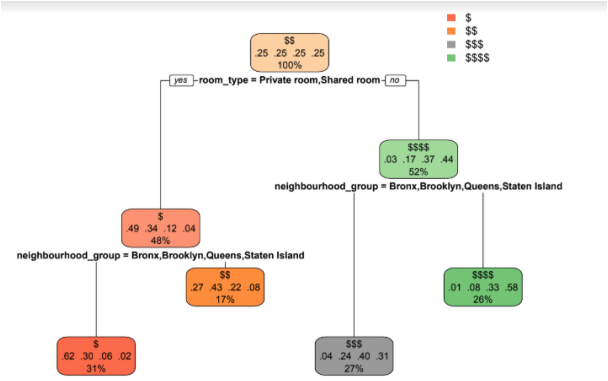


**Figure 12: Decision tree with only categorical attributes**

- In an attempt to improve this performance, we convert the numerical data into appropriate categorical values and build a new decision tree.
  The latitude attribute is divided into North and South, the longitude into East and West.
  Figure 13 depicts the decision tree built with these additions. Its accuracy is 52.98056%.

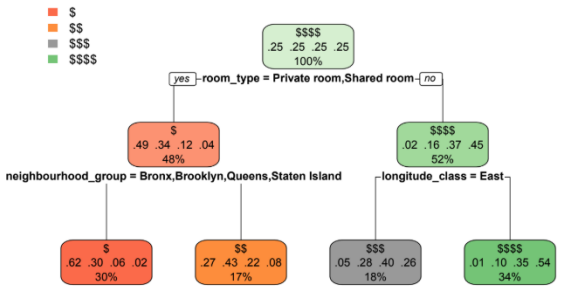


**Figure 13: Decision tree with categorical and numerical attributes**

- The next step in improving the performance of the decision tree is the cleaning the current data to remove any outliers or meaningless values. These changes affect the attributes minimum nights, availability_365 and host listing count
  Further, on visualization of the attributes, we noticed that the east and west classes were not balanced as seen in figure 14 and hence we shifted the split towards the east class in an attempt to balance the classes
  Figure 15 depicts the decision tree built with these additions. The accuracy of the tree built after these changes is 54.01713%

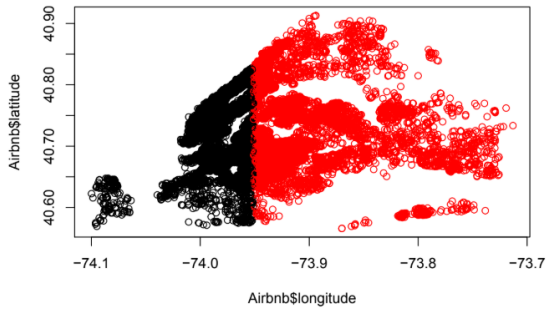


**Figure 14: Plot of longitude split on New York Map**

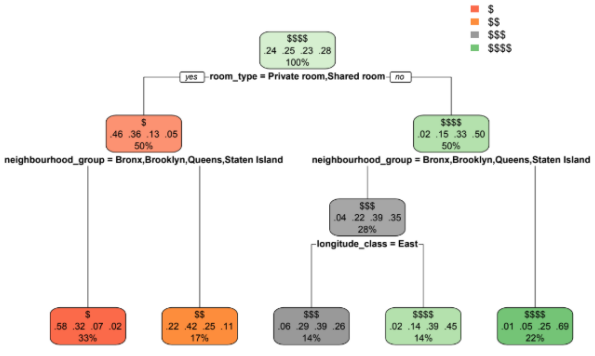


**Figure 15: Decision tree after cleaning outliers**

## 8.2 Refining the price split

The next attempt in improving the model's performance was by changing the price split. So far, the price was discretized into four categories based on quantiles. In this stage, the real values are categorized into 3 classes based on percentile - 0, 0.33, 0.66. All the remaining attributes are left unchanged and the decision tree is built.
Figure 16 depicts the decision tree built on the three way split. We note that this change positively impacts the accuracy and brings it up to 66.6821%

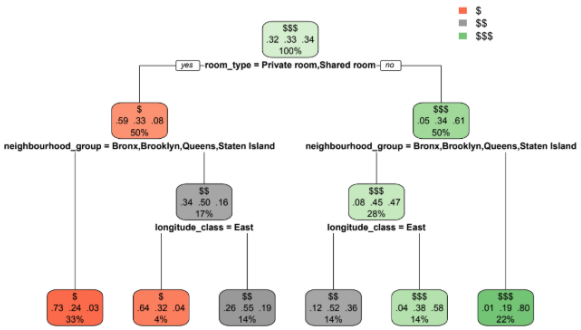


**Figure 16: Decision tree with three way price split**

## 8.3 Adding more attributes from another dataset

Airbnb is a publicly available dataset that contains even more features compared to the New York Airbnb dataset that we are currently using. To explore these features and see if they are good predictors for the price class, we merge the two datasets based on their unique name and id.
The new features obtained from this merge including the cleaning fee and the cancellation policy. Figure 17 show the distribution of these attributes. The cleaning fee is a real valued attribute and is hence categorized based on percentile into three classes.
The decision tree for the three way price split is then built

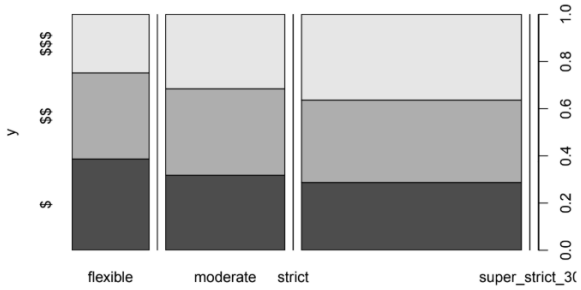


**Figure 17: Cancellation Policy Distribution based on price**

with these new attributes also added to the formula. Figure 18 is the result of this change.
However, we notice that the additional features do not positively impact the accuracy. In fact, the accuracy after these changes is at 54.54545%

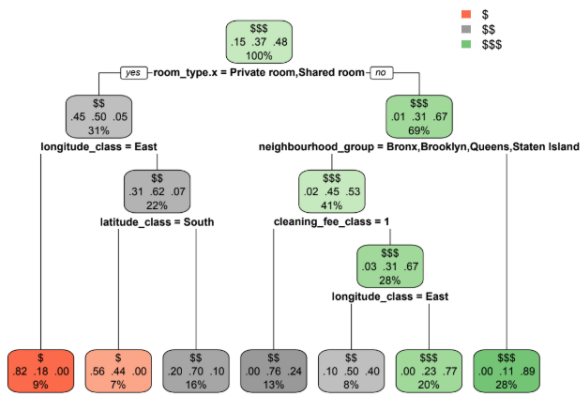


**Figure 18: Decision tree with more attributes**

# 9. A BETTER DIVISION

By checking the data roughly, we found that several certain prices like 99 and 199 have gathered a lot of entities. At the same time, median of all the price is exactly 100. These circumstances indicates that if we make classification based on these values like 100, it may not give us reasonable results, which also can be shown in the following table.

## 9.1 clustering

In order to fix this problem, we decided to use clustering to the prices to find out a better division.

## 9.2 K-means

First of all, we applied k-means clustering to this task, but it was proved to be a bad idea. K-means clustering is based on distance between points, yet prices are continuous, where distance lost their meanings. In the end, K-means gave us 4 clusters as following and it assumed the rest points as noise. This model is not a good model compared to next one.

## 9.3 density based clustering

As a matter of fact, what we really plan to find out is the prices' distribution, to be more specific, is how price looks like in different ranges. So density-based clustering is abetter choice here.

To do density-based clustering, there are two key parameters, one is epsilon ("eps") and the other is minimum points ("MinPts"). Eps defines the range of cluster and the higher the MinPts is, the more points are int the cluster. They both are hyper parameters, and best values of them depends on conditions. For our task, I tried different values and the results are shown:

From the table, we learned that range 91-105 have a highest density, which means that we'd better put them in same cluster. Despite that, we also get a few more clusters. So our 1st cluster is range 1-89.5, which proved to be correct as we can get around 80-85 accuracy in later tests.

However, the 2nd cluster is quite confusing, there are 3 main clusters above 105 which are 111-135, 140-160 and 190-210. In fact, every split has been tried and final results showed that they cannot get good accuracy at the same time.

In the end, we choose 140.5 as final second split point, and built our decision tree on it.

## 9.4 Gaussian Distribution

By anlysing the distribution of price, I realized that it is actually quite like Gaussian distribution (or Normal distribution).

By the formula of Gaussian distribution,

There is also a 3 u principal for Gaussian distribution, which is this is quite different from our results, yet after removing the noise and extreme values, final result is (79,135).

# 10. RESULTS