**Semester Project: New York City Airbnb**

**Zehuang Hong Chen**

**Vasinee Powthong**

**Junpeng Li**

**Xiao Yang**

**Amirali Abolhelm**

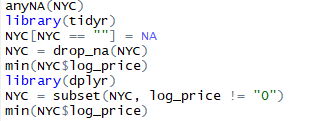
**MET AD 699 Data Mining for Business Analytics**

**Gregory Page**

**05/06/19**

**Step I: Data Preparation & Exploration**

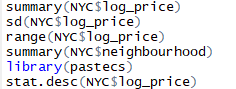
In order to analyze our data set, the first step is to simplified it. Our team is working with New York City, so we filter the city and name it as NYC. Then we View the data set, and remove the latitude, longitude and thumbnail\_url columns; they are –c(20,21,26) respectively, because we think these factors are not relevant to our dataset analysis for New York City. Lastly, we used droplevels to simplified unused levels of property\_type(apartment,house and loft).

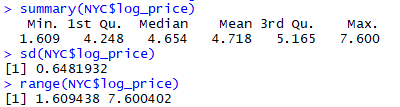


**I. Missing Values**

First, we input anyNA(NYC) to check if the data contains any missing value(NA), and the output is TRUE. Second we replace the empty cells with NA, we can combine these together and dropped the NAs. Last, we using drop\_na to delete any NA cells in NYC. On log\_price column, we testing the minimum log price which cannot be zero, because it is undefined and free listing is not possible. Then we using subset function to remove all the cells which have zero log price. Then we testing the minimum function again and got the number as 1.609438.

**II. Summary Statistics**

****We want to cursory review the variables we are interested in before we start analysing the dataset. In this part, we tried to summary the log price and neighborhood. Checking the range of log price, standard deviation, variance, and coefficient variance.

****

By using the summary function for log price in NYC dataset, the minimum price is 1.609, median 4.654, and max value 7.6. The standard deviation is 0.6481932. The range of log price is also presented in the summary of log price. If we want to see a further table of basic descriptive statistics for my variables, we using stat. desc checking the coefficient of variation, confidence interval mean and standard error mean. Then we want to count how many neighborhoods we have. These data analysis will be present in the data visualization section.

**III. Visualization**

In this part we used ggplot to visualize the connection between different variables.

We using Boxplot with bedrooms and log price. Top 10 Neighbourhood and price in barplot. Violin between accommodates and log price. Histogram between property type and frequency. Barplot for average log price and top10 neighborhood (The Graph screenshots are attached at end of this project paper).

**Boxplot with bedrooms and log price**

We want to see the number of bedroom influence the log price. In boxplot, from 0 bedrooms to 6 bedrooms. we can see there is a low rate growth of log price. And the range of each bedroom price is most likely the same.

**Barplot with Top10 neighborhood and frequency**

In this plot, we can see the frequency of the top 10 neighborhoods. Williamsburg and Bedford-Stuyvesant have more than half than the rest 8 neighbors.

**Violin within accommodates and log price**

A violin plot is a method of plotting numeric data. It is similar to a box plot, with the addition of a rotated kernel density plot on each side. As we can see, accommodates with 13 are different from others. The most accommodate have a soft log price.

**Histogram with property type and frequency**

In this chart, we want to see the frequency of the different type of property. As we can see, Apartment is higher than each other property type. I think it depends on city planning and population.

**Scatterplot with neighborhood and log price**

In this graph, we what is each neighborhood log price. As we can see, they all have the same scale of log price.

**Barplot with average log price and top 10 neighborhood**

In this plot, we want to see the average log price of the top 10 neighborhood. As we can see the average log price are mostly the same. Upper West Side has the highest average price.

**Step II: Prediction**

**I.** **Create a multiple regression model with the outcome variable ​*log\_price*​.**

Firstly, we decided to choose accommodates, review\_scores\_rating, number\_of\_reviews, property\_type, bathrooms, bedrooms, beds and room\_type as the independent variables due to several reasons. The number of accommodating per stay is one of the most important factors to set the price for Airbnb because it makes sense that the more people stay the higher the price and cost to the owner. Review rating and number of reviews are also affected Airbnb price since they are the guarantee that the place has a good quality or not. Moreover, different property types offer different convenient levels and different numbers of stay. Another factor to price is room type as the bigger the room is the price should be higher as well. However, we decided to remove bathrooms, bedrooms, and beds as independent variables. The reasons is that we do not include bathrooms, bedrooms, and beds is that we want to avoid the multicollinearity because they have a very high correlation. After creating ggpair with the independent variables chosen to assess correlation among them, it can be seen that they are no multicollinearity (as shown in figure 1).

Then, we also filter property\_type to analyze only apartments because it entails around 80% of the data set and in NYC is mostly composed of this property type. So, the others are not significant in impacting log\_price. Next, to prepare the variables for the model, we have to create a dummy variable for room\_type since it is a factor with 3 levels. Data partitioning is the next step before building the regression model. Then we ran the regression model and used the backward elimination approach to see which are the useful predictors. The result suggests that the number of reviews which has a high p-value of 0.718 so it is not significant to our model and should be eliminated.

Consequently, we built the regression model with review\_scores\_rating, accommodates, and room\_type. The model will be evaluated on the validation set. Figure 3 represents the summary measures for the forecast accuracy and it shows that the values of MAE, MPE, and MAPE of both training set and validation set are pretty close together so it can say that our model is acceptable.

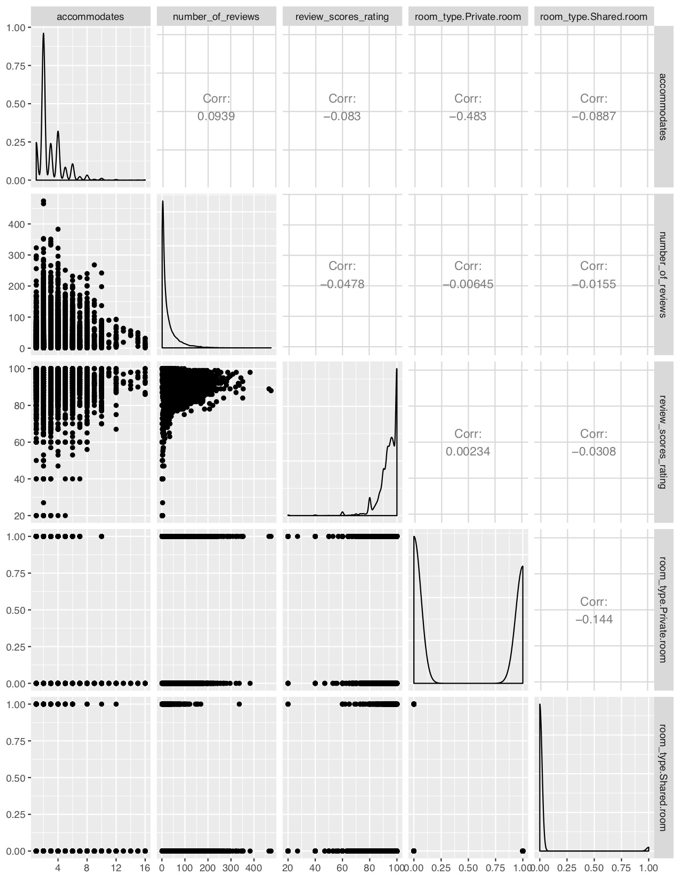


Figure 1.

Figure 2 shows regression model that can be used to predict log\_price from accommodates, review\_scores\_rating, and room\_type. The equation is

Log\_price = 4.2 + 0.11(accommodates) + 0.006(review\_scores\_rating) – 0.61(room\_type.Private.room) – 0.95(room\_type.Shared.room)

For both room types have a negative coefficient which mean that these variables have indirect variations to log\_price. In contrast, if accommodates and review\_scores\_rating is higher, log\_price will be higher as well.

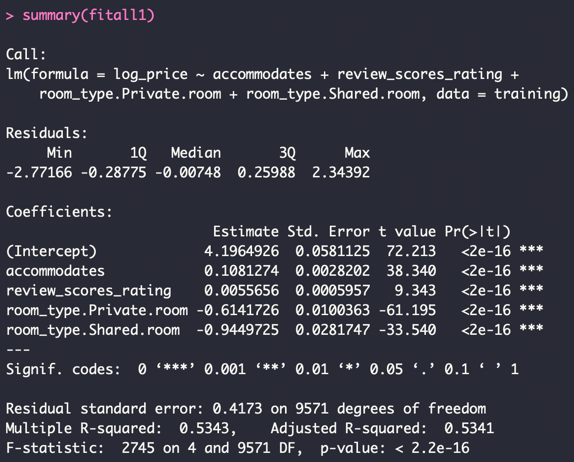


Figure 2.

R-square equals to 0.5343 which calculated from the square of the correlation coefficient between input and output and to compare with the adjusted r-square, at 0.5341 which is lower, so we can say that it indicates better fit.

RMSE equals to 0.418 which calculated from the square root of the average of squared differences between prediction and actual observation and can tell the error or the model whether positive or negative and here we can see from figure 3 that RMSE from the validation set is larger than the training set, so our model is acceptable.

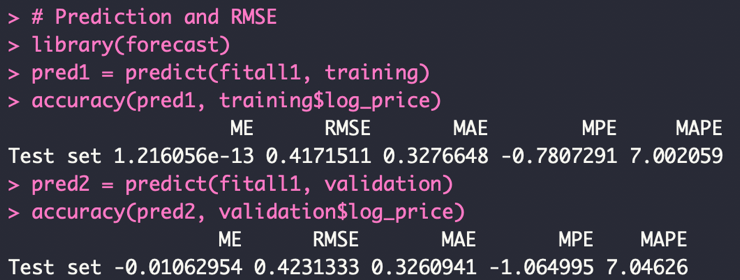


Figure 3.

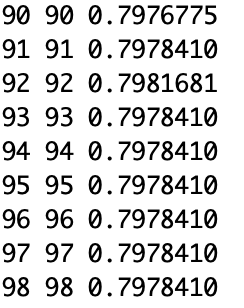
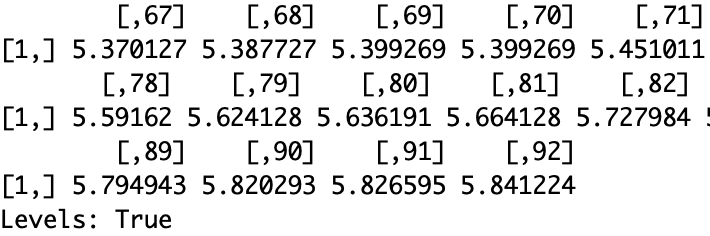
**Step III: Classification**

**K-nearest neighbors**

To predict whether our rental will have a cleaning fee in New York City, we first created a new dataframe with the following predictor variables: *log\_price, accommodates, number of reviews, review scores rating, and room type.* Our team came to the conclusion that these predictors are the most important to predict whether the rental will have a cleaning fee. For instance, a high log price usually has a cleaning fee included in its price. Additionally, the larger the number of people a listing accommodates, we should expect to also have a cleaning fee embedded on its price. Furthermore, we included review score rating because we expected from the beginning that a high review score rating will include cleaning fee, as the place will always be clean and neat before the next guests arrive. After selecting our predictors, we normalized the data so each predictor variable has the same weight, so we can avoid overweighting or underweighting the chosen input variables.

Our team chose to analyze 98 nearing neighbors at first because we took the square root of the total number of observations in the training data to obtain more accurate results. From this result, we used the summary function to obtain the number of nearest neighbors with the highest accuracy and it corresponds to k=92. After running our knn function with 92 nearest neighbors, it predicts that our rental would have a cleaning fee, please see below where the outcome prediction evaluates to True.

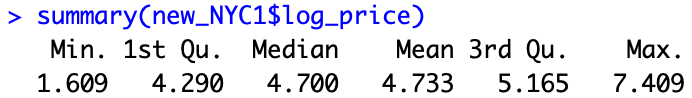




**Naive Bayes**

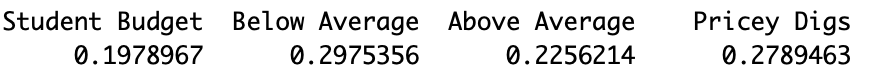
Before cutting log\_price variable into four similarly-sized bins, we called the summary function to understand how this variable is structured, so we can create more accurate bins. See below for how we cut the bins.

* “Student Budget” → [1.609 - 4.29]
* “Below Average” → [4.29 - 4.733]
* “Above Average” → [4.733 - 5.165]
* “Pricey Digs” → [5.165 - 7.409]

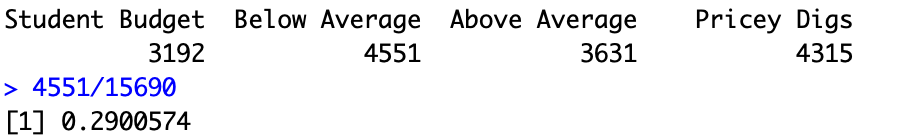


The five predictors we used to build our naive Bayes algorithm were as follows, *accommodates, number of reviews, review scores rating, private room, and shared room* after creating our dummy variables. First of all, accommodates is a very important variable that influences price because the log\_price is positively correlated with the number of individuals it can accommodate. Furthermore, a listing with a higher number of reviews has the capability to charge more, as it has records or reviews about people’s satisfaction staying at this particular Airbnb. Moreover, review score rating, private room and shared room are significant factors that affect the price of a listing because all these attributes gives the owner power to charge different prices based on features his place can offer.

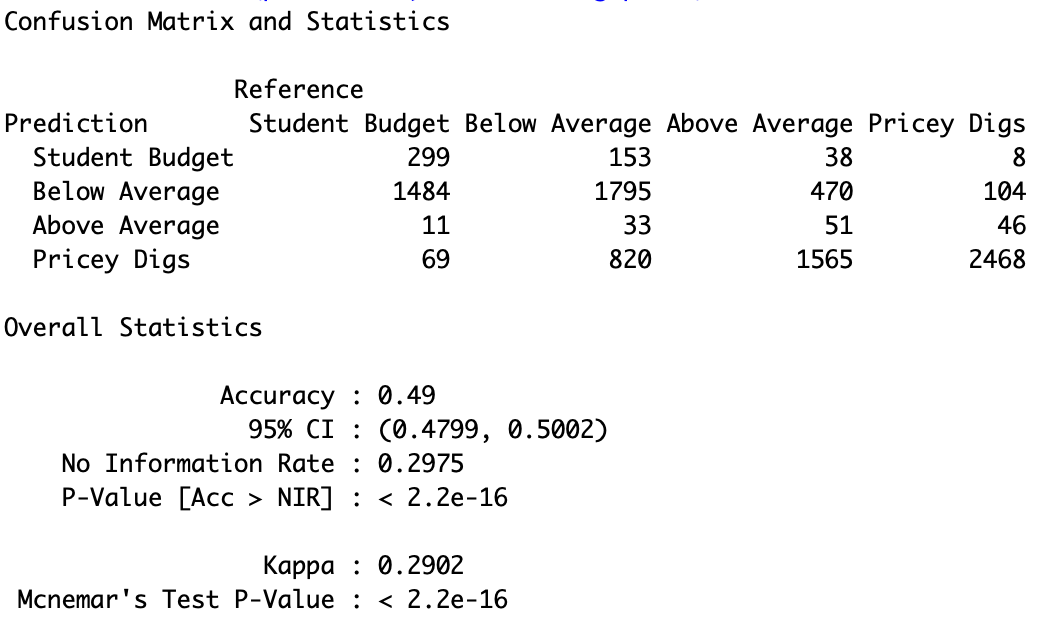
After selecting our predictors, we built a naive Bayes model with log\_price as the response variable. As we compared our model against both the training and validation dataset, we can see that the accuracy metric for training and validation is 0.49 and 0.47, respectively. In comparison, using the naive approach, we have an accuracy of 0.29 which is very different from the accuracy metric obtained from naive bayes. For our fictional apartment, using a naive approach would have classified our rental as Below Average because it contains majority of the records. On the other hand using naive bayes, our fictional apartment would have also been classified as Below Average but it is based on the predictors we chose; therefore, it provides more valuable insights as we can see the percentage of records for each bin based on the records chosen. Lastly, the accuracy for the training and validation are very similar, but the training data has a higher accuracy, which is expected because our model was built based on the training data. As the accuracy metric for both are very similar, this entails that we have a good model for predictive purposes and we are not capturing noise and overfitting the data.

***Naive Bayes Classification***

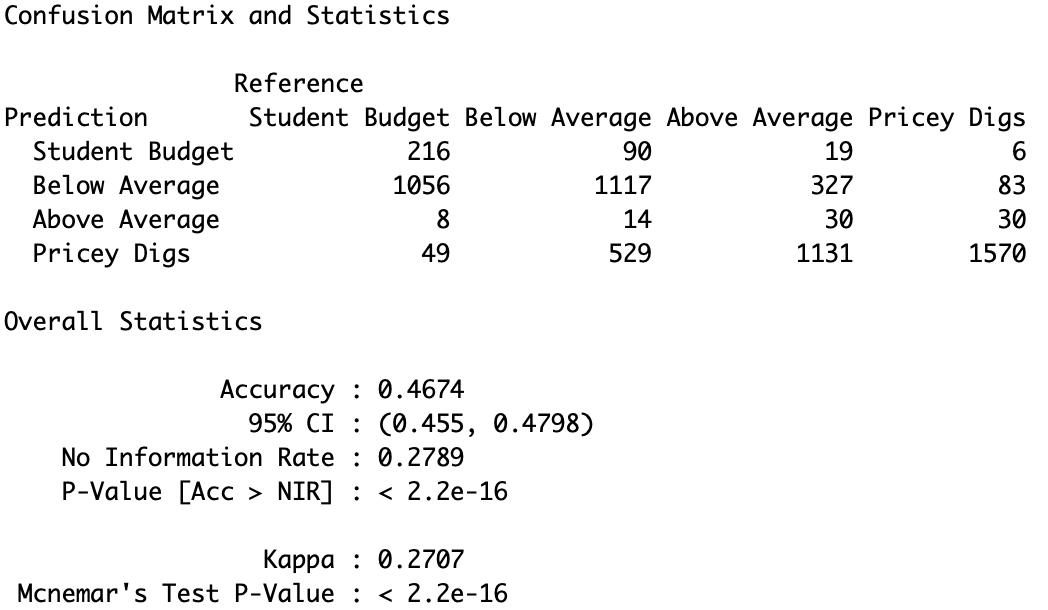
***Naive Approach***



***Training Matrix***

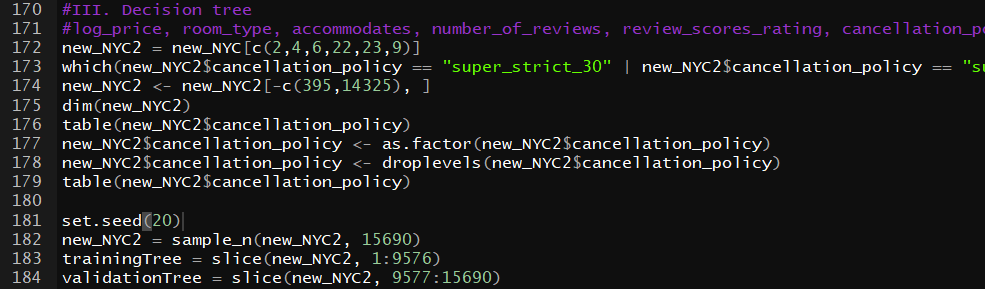


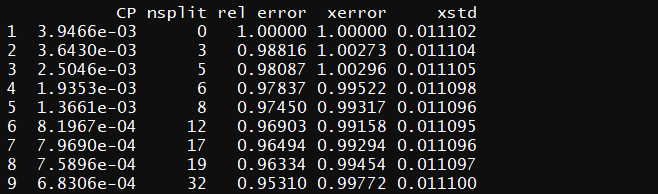
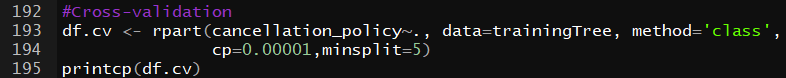
***Validation Matrix***

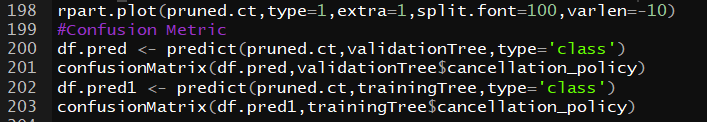
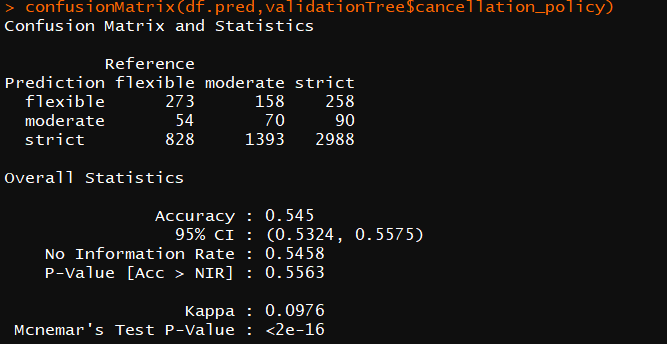
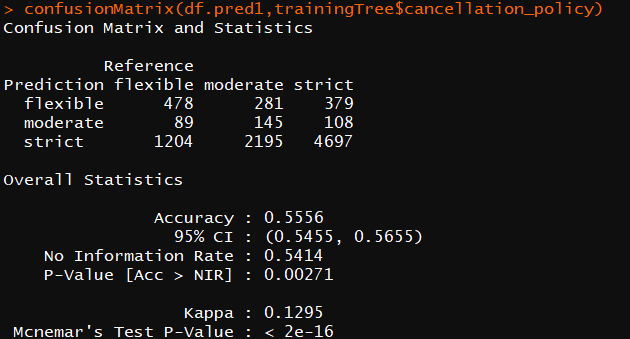
******

**Classification Tree**

The first step to build a tree is to get the data prepared. As what we did from the previous model, input variables and output variable will be selected. In this case, we picked log\_price, room\_type, accommodates, number\_of\_reviews, and review\_scores\_rating as our input variables, since we deemed these variables having a strong impact on cancellation policy. We dropped some of the variables on the other hand, for example, the massive levels of property type, bathroom, and all others that are less useful to our classification model. As for our output variable, cancellation policy, we need to drop the super\_strict\_30 and super\_strict\_60, because of their small count in the big data set. As what we did before, the next step is to slice the selected variables into 60% of training data and 40% of validation data. At this point, we are ready to implement our classification tree model (the codes are on the right).

Before we build our tree, we need to determine the optimal split of our tree. A cross validation is a proper approach to find a smaller tree with the smallest xerror. Below are the code and cv table we generate:

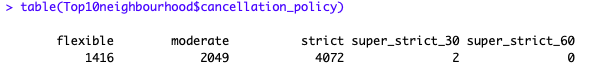
We can tell from the cv table that the optimal split should be 13, with the smallest xerror of 0.99158. Then we apply the picked split for our pruned tree, visualized it and then we evaluate the performance of the pruned tree using confusion matrix (the codes, and matrix are below, while the tree is in appendix).



As we can see the accuracy rates are 0.556 and 0.545 for training and validation sets respectively. These numbers look good to me since it is normal that the accuracy rate is lower for the validation set. New data brought in more noise inside the data.

**Step IV: Clustering**

**Clustering**

To cluster the data we decided to choose the top 10 neighborhood in NYC based on their frequency. "Williamsburg", "Bedford-Stuyvesant", "Bushwick", "Hell's Kitchen", "Harlem","Upper West Side", "Upper East Side", "Crown Heights", "Astoria", "East Harlem". The next step in the analysis was to get rid of the super strict cancelation policy for the top 10 neighbourhood since there was only two properties with that type. 

In order to do this, the drop level function was used. Please refer to code below.

*new\_NYC2$cancellation\_policy <- droplevels(new\_NYC2$cancellation\_policy)*

After, we are ready to dummify the categorical data for the data frame top 10 neighbourhood. We only have three categorical variable that we took into consideration. The three variables are shown as follow neighbourhood, cancellation policy and cleaning fee.

**Dummify**

Neighbourhood

dmy <- dummyVars("~neighbourhood", data = Top10neighbourhood, fullRank = FALSE)

transform1 = data.frame(predict(dmy, newdata = Top10neighbourhood))

Top10neighbourhood = cbind(Top10neighbourhood, transform1)

Cancellation Policy

dmy <- dummyVars("~cancellation\_policy", data = Top10neighbourhood, fullRank = FALSE)

transform2 = data.frame(predict(dmy, newdata = Top10neighbourhood))

Top10neighbourhood = cbind(Top10neighbourhood, transform2)

Cleaning fee

dmy <- dummyVars("~cleaning\_fee", data = Top10neighbourhood, fullRank = FALSE)

transform3 = data.frame(predict(dmy, newdata = Top10neighbourhood))

Top10neighbourhood = cbind(Top10neighbourhood, transform3)

The last step was to normalize the data to because this way we can scale all the variables and then compare them. The variables that were considered for the clustering were log\_price, accommodates, number\_of\_review, reviews\_score\_rating, bedrooms, beds, room type, neighbourhood, cancellation\_policy and cleaning\_fee. Please refer to code below for the process of normalizing.

**Normalizing**

Top10neighbourhood.norm <- sapply(Top10neighbourhood, scale)

row.names(Top10neighbourhood.norm) <- row.names(Top10neighbourhood)

str(Top10neighbourhood.norm)

View(Top10neighbourhood.norm)

Now the only thing left is to determine the optimal number of clusters thus, a good method is to use the elbow chart. Please refer to code below for the process of determining ‘k’.

**Elbow Chart**

km <- (nrow(Top10neighbourhood.norm)-1)\* sum(apply(Top10neighbourhood.norm,2,var))

for (i in 2:15) km[i] = sum(kmeans(Top10neighbourhood.norm, centers = i)$withinss)

km

plot(1:15,km,

type="b", pch = 19, frame = FALSE,

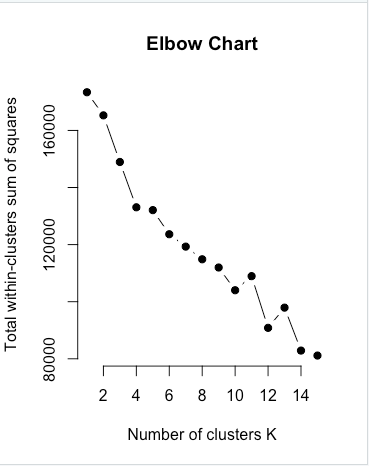
xlab="Number of clusters K",

ylab="Total within-clusters sum of squares",

main = "Elbow Chart")

km1 <- kmeans(Top10neighbourhood.norm, 7).

From the elbow chart below it could be observed that 7 clusters is the ideal number of k. This choice is very subjective but based on our analysis and using trial and error method we first started with 4 clusters and as we saw better patterns (more detailed) for each cluster we decided to add to the number of clusters and we ended up with k=7.



Now, by using the below code we are able to see each of the clusters.

km1$centers

Significant patterns were identified for each for each of the clusters. First, for each cluster we identified the neighbourhood that has the highest value. Therefore, using this strategy we could observe different characteristics for each of the neighbourhoods based on the variables.

**Cluster 1**

The screenshot below shows the 7 different clusters and all the variables chosen. Most of the cluster 1 belongs to the Upper East Side neighbourhood which has the value of 3.2. This neighbourhood is mostly well suited for shared rooms compared to the private rooms. In addition, the bedrooms have value of -0.44 which explains that there are not many bedrooms. Also, the price is relatively average compared to the other clusters. This cluster could be convenient for people who want to be at the heart of Manhattan and looking to share a place with others.

**Cluster 2**

It looks like that most of the cluster 2 is in Hell Kitchen Neighbourhood 2.85. The two variable that stands out for this cluster are price 0.65 and number of reviews 0.26. In fact, this cluster has the most number of reviews compared to the others. Also, the price is relatively high compared to other cluster. Thus, this cluster are the people who are relatively rich and they really care about reviews when choosing a place to stay,

**Cluster 3**

They are mostly in neighbourhood of Williamsburg and they have relatively high cleaning fee 0.48 and strict cancellation policy 0.45. Also, the price is relatively low -0.29 therefore, this cluster are the places that their owners are very strict and not so flexible with their guest. This cluster is good for people who have set plans and are sure that they will not cancel.

**Cluster 4**

This cluster are mostly in the neighbourhood of Bedford and they have very high prices of 1.19. Also, they accommodate value is 1.90 which means they can fit many guests. In addition, the bedroom value is 1.80 and beds 1.87 which means this place is very huge. Thus, it is recommended that this place is suited for big families who are wealthy.

**Cluster 5**

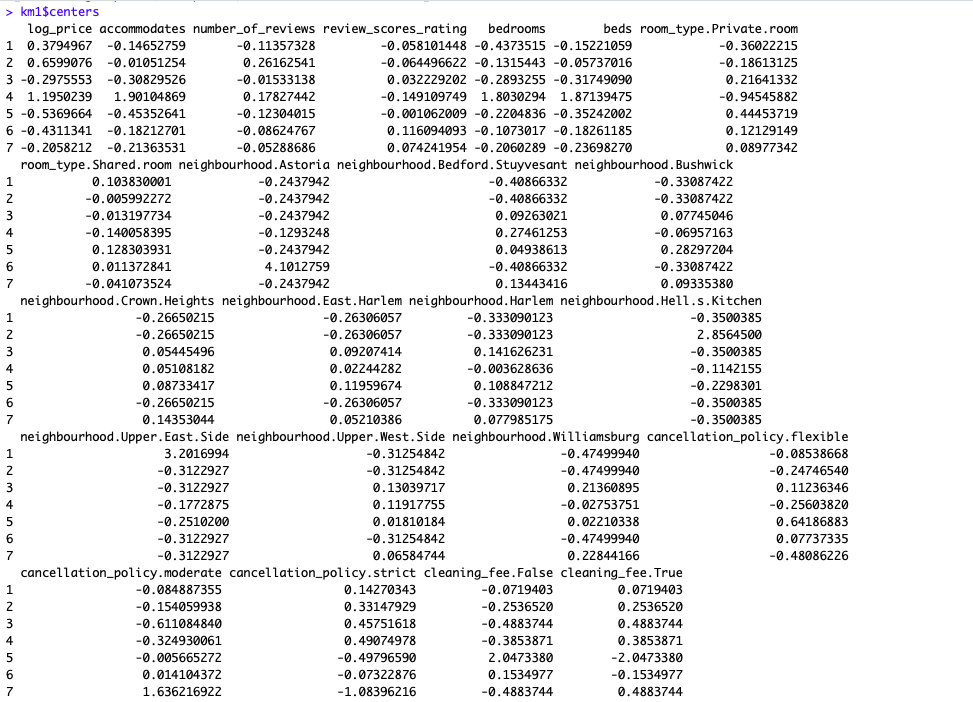
They are mostly in neighbourhood Bushwick. The price for this cluster is -0.53 which is the lowest compared to other clusters. They have the least amount of bed -0.35 and low value for accommodate -0.45 which means this place is very small and it is better for a single traveler or a student who does not have money.

**Cluster 6**

They are mostly in neighbourhood Astoria. The price is relatively low as well -0.43 and it actually has good score rating of 0.11. The moderate cancellation policy is f 0.014 and the flexible cancellation policy is 0.07. Thus, this place is suggested for people who are not sure about their travel plans and those who really care about the rating of the place when they book.

**Cluster 7**

They are mostly in Williamsburg. Their price is relatively low -0.20. The cancellation policy for this cluster is moderate 1.63 and it has relatively high cleaning fee of 0.48. It could be concluded that this cluster is also good for people who do not really care about paying the cleaning fee and looking for moderation cancelation policy.

**Cluster**

**Step V: Conclusions (10 points)**

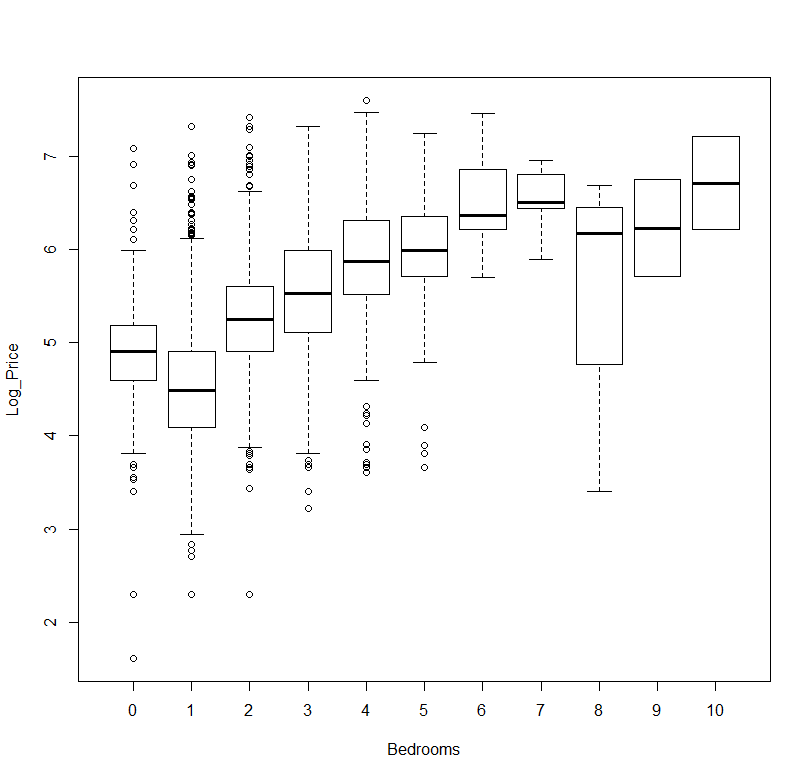
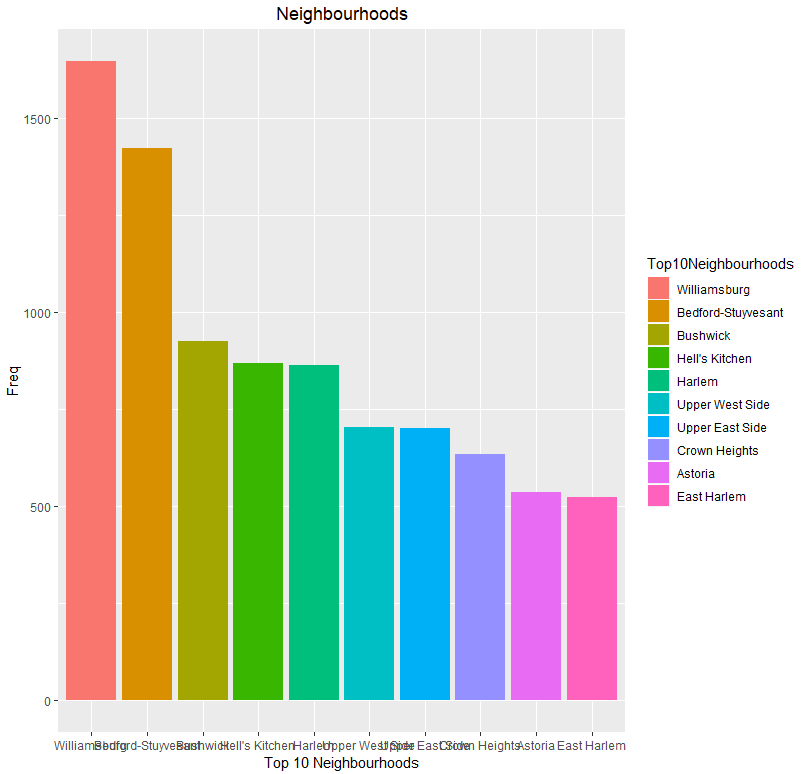
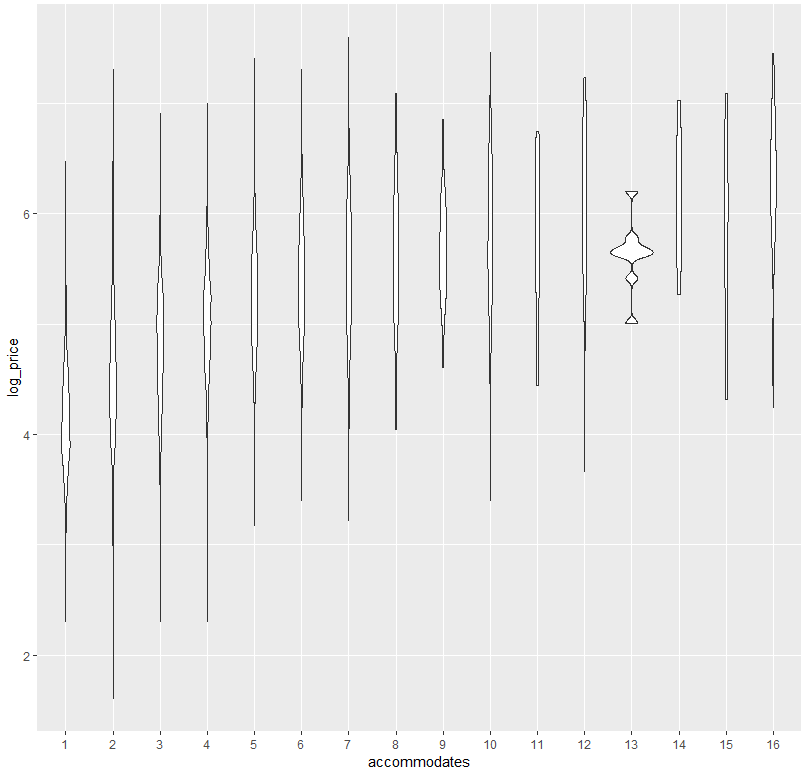
Data visualizations are a picture or graphical presentation about certain information. Data visualizations help us make decisions by seeing information visually. I believe for this reason data visualizations are important because they help us make a decision or get a main idea how the data are structured. This is why we think when the manager or supervisor asks us to prepare a presentation, they are more looking for data visualization than listening to programming slides because most people are visual learners and will easy to understand the story behind the graphs.

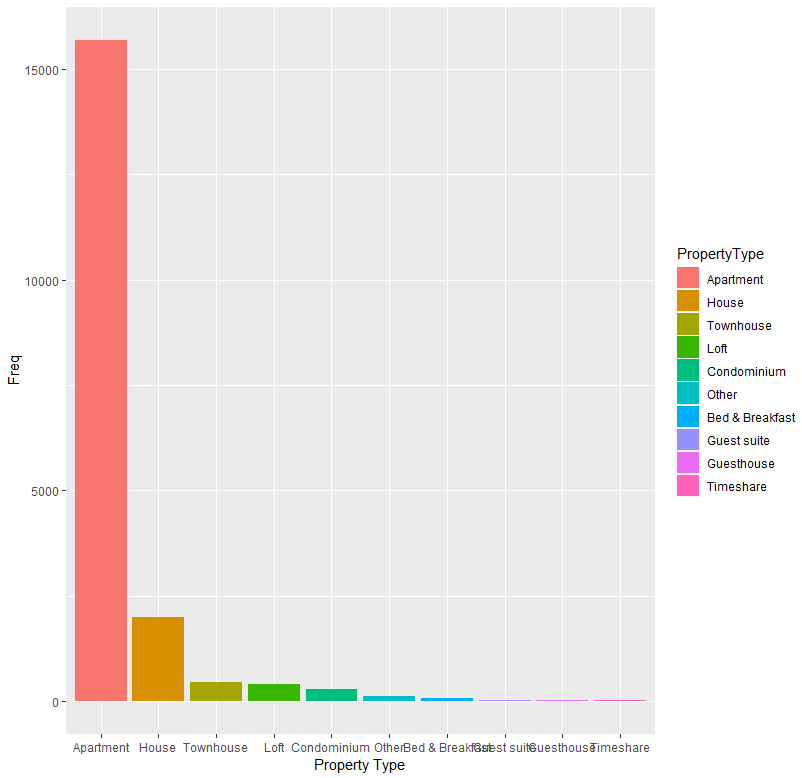
To predict log\_price, we need to build a multiple regression model which has log\_price as an outcome variable and we decided to use review\_scores\_rating, property\_type, and room\_type as independent variables based on ggpair chart and backward elimination approach. Firstly, we eliminated bathrooms, bedrooms, beds because of the result from ggpair that they have high correlation causing multicollinearity. In addition, we filtered the apartment from the whole property type since it takes 80%. Moreover, room type is a factor data that need to be transformed into dummy variables. Then running the backward elimination approach, the result suggests that we should eliminate the number of reviews due to high p-values, at 0.718. After running the model with four independent variables (revies\_scores\_rating, property\_type, room\_type.Private.room, and room\_type.Shared.room) and have log\_price as the outcome. The regression equation shows that accommodates and review\_scores\_rating have direct variations (positive coefficient) to log\_price despite the others having a negative coefficient. RMSE of validation set is larger than the training set and other accuracy values are close together, so our model is acceptable. Furthermore, the adjusted r-square of the model is higher than r-square which indicates a better fit.

For classification models, whether they are K-nearest, Naive Bayes for classification tree, their main function is to group data into different given categories and attributes. And this break down method tells a more in-depth story other than numeric data does, since classification reads binary data as good as it reads numeric data. Besides, to better enhance our classification models, the industry expertise matters since predictors are subjective to be selected by authority. If the decision maker failed to recognize what predictor stands out, the performance of classification models might fail as well. During the model implementation, we feel that it is better to compare how each model works and what they coincide with the outcomes. For instance throughout our models comparison, it was very effective to compare the accuracy to evaluate a model’s performance as we need it to be useful for predictive purposes and to avoid overfitting the data and capturing noise. Our team concluded that Naive Bayes is a very powerful tool for classification purposes because it gives an overall idea on the percentage of records belonging to each bin or category based on the predictors an individual chooses; in comparison, the naive rule does not provide this insight; it only categorizes based on the majority of records.

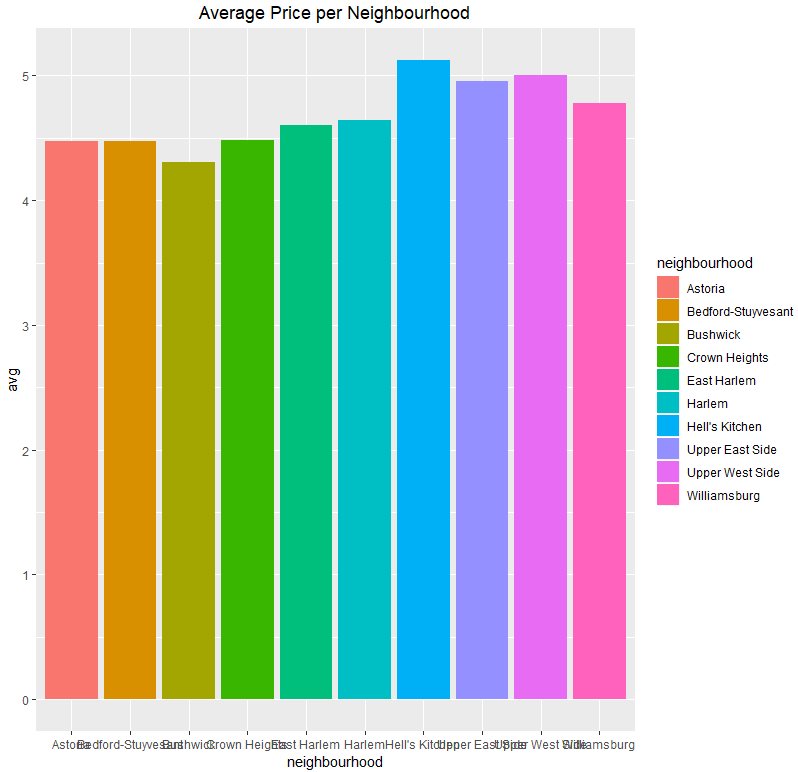
The purpose of clustering was to understand various characteristics of each neighbourhood. Clustering is actually helpful for Airbnb to understand how diversified are its properties across NYC. Based on our analysis that took into account 10 neighbourhoods and 7 clusters we compared each cluster. As a result, each cluster is unique because it is customized to the preference of the guests. Considering this Airbnb could tackle each customer independently. This will help them to grow and become popular among all types of travelers.

**Data Visualization plot**

1. Boxplot
2. Barplot
3. Violin Plot
4. Histogram



1. Barplot



6. Classification tree