Airbnb Rating Prediction

Project Report

Group 10

Arjun Chanda

Ateeksha Chaudhary

Garrett Chaffey

Yuexi Li

Qiuhan Li

BANA 273

Machine Learning Analytics

Professor Mingdi Xin

1.1 Executive Summary

Airbnb is a company that specializes in empowering small time property owners to rent their properties for short term rentals (and generate a secondary source of income). The company is especially popular with tourists and individuals/groups who want to visit a new locale but who do not want a long term commitment which a lease would force them into.

On the Airbnb platform, users give these properties thousands of reviews providing a wealth of information for both property managers and future tenants. Many properties however have hundreds or even thousands of reviews, making it both time consuming and difficult for future tenants and property managers to determine if a renter had or will have a positive experience or not. Accordingly, most users do not read all the reviews before booking a property, instead tenants prefer to look at the aggregate star rating of the property to make their decision on where to stay. The relationship between the sentiment of text reviews, property attributes and average property rating is not always immediately obvious. While there does seem to be a relation between the wording of a review and a property's attribute scores on a per review basis, it is difficult for property owners to determine what exactly drives the customer's review and what factors influence the rating of their property. We created and examined word clouds derived from customer reviews and extracted attribute scores from properties to create models which will predict what rating a property will receive on a better than the competition/worse than the competition scale (ie pass/fail or 0/1) where the cutoff between the binary classification will be the average rating or 4.7.

1.2 Business Idea

Our group built a model that predicts whether an Airbnb property in New York receives a “better than the competition” score or not by extracting attributes that are determined to be important, and performing a sentiment analysis on the reviews for existing properties. Smaller Airbnb hosts (with a small number of rental units) will be able to use our model to better understand what factors drive the rating system and to hone in on and make improvements/changes to existing amenities/features, per stay preparations (like cleaning, putting out shampoo, etc), pricing (potentially informing on pricing decisions so that renters feel like they get “their money’s worth”) and other changes with the goal of increasing the most important factor which determines whether a property is chosen by a potential renter: the average review score (in tandem with the number of reviews). Thereby making listed properties more attractive when being viewed by potential renters. Additionally, our model will also guide the decisions of Airbnb hosts by providing an easy to understand pass/fail metric which indicates whether or not their property is better than the competition in New York city.

Larger Airbnb hosts (as in hosts which manage a larger number of properties) can benefit from our model in all of the above ways that smaller Airbnb hosts can with the additional benefit of also predicting whether or not a new property they do not currently own is worth investing in/purchasing as they can examine the features that are currently exhibited by the property in question, use our model to predict an average rating based on the observed features and then decide on whether or not to buy the property based on how attractive the predicted rating will be (and by extension how attractive the property will be in the eyes of Airbnb users).

On the corporate side Airbnb has an interest in vetting properties and ensuring that high quality locations are added while minimizing the number of properties added which will consistently receive poor scores as it hurts their platforms image; with this goal in mind Airbnb (corporate) could use our model when determining whether or not to accept a new host’s application to list a new property rental as they already gather most

of the information required for our predictive model (all that is missing is the sentiment analysis of the comments/reviews) and they could then use our model to accept/decline applications.

Our goal is primarily to make ratings more scientific, and precise, as well as help hosts understand more about the key factors that will influence their business and revenues; while at the same time giving an easy to understand metric on whether a new property will pass and receive a “good” score or fail and receive a “poor” score.

2.1 Data Acquisition, Description & Cleaning

A. Acquisition

In this project we will be examining the corporate provided datasets from the Airbnb website (http://insideairbnb.com/get-the-data.html) to look for attributes which are significant drivers for Airbnb property ratings. We will be using the following datasets: 1) Listings - which contains the detailed listing’s data for New York, and 2) Reviews - detailed customer review data for listings in New York.

B. Data Cleaning

Our step by step process for cleaning and processing the reviews dataset was as follows:

1. We started cleaning the data by fetching the comments from the Reviews dataset containing 876,200 records.
2. After pulling the dataset, we converted the entire comments column to lowercase and imported the ‘stopwords’ package from the nltk library to remove all the unnecessary words from the sentences leaving only the keywords in a sentence.
3. Once we were down to only the keywords, we removed punctuation from the sentences.
4. Finally, we eliminated the reviews which were in languages other than English in order to make all the sentences readable by the model. The number of comments after cleaning the reviews dataset totaled 15,722.

With the cleaning done for the ‘Reviews’ dataset, we moved to the ‘Listings’ dataset. We started cleaning the data by using Python to remove all non-ascii characters using the “string.isascii()” method to find non-alphabet and non-numeric values in the reviews and other columns of our data to allow for their removal. From there we removed columns which we deemed unimportant or which we decided would not provide any relevant information to us, such as ‘last_review', 'host_neighbourhood', and 'bathrooms' (this column only had null values in it). Before removal of these attributes our dataset contained seventy-four columns, however after removal of what we had deemed extraneous we had pared down the number of attribute columns to forty-one.

With our new refined dataset we then began the process of converting non numeric data into more easily manageable data types by creating dummy variable columns. Some of our columns, such as ‘Superhost’, ‘Availability’, and ‘Bookable’, contained values of either “True” or “False”, we turned these values into ones and zeros to better be able to use them in our models without problem. Additionally for some of the columns there were percentage and dollar signs that were associated with the values (specifically in the columns ‘Price’, ‘Host Response Rate’, and ‘Host Acceptance Rate) and we solved this issue by extracting the desired numbers from these fields and dropping the dollar and percentage signs. The final step for value extraction was to check for string type numerics, and when found convert them into true numerics. One example of this was the bathroom count for the properties in the dataset as the column would sometimes list strings like “half bath” to describe the total restrooms in the property. These string type values were then extracted as a numeric value (0.5 in the case of “half bath”) and added to the final count of numeric form bathroom data. After adding the dummy variables, we then dropped the original columns. We also joined the two datasets (Listings and Reviews) horizontally by property ID.

C. Description

The dataset is divided into 2 different datasets having Listings and Reviews. We have 37,713 observations in New York’s Listings and 876,200 observations in the

Reviews dataset. Running a forward selection listed the following ten attributes as being the most influential on average review score:

- Number_of_reviews: the number of reviews for a particular listing.
- Review_scores_accuracy: Score depicting how accurately the listing has been described on airbnb.
- Review_scores_cleanliness: Score representing the cleanliness of the property.
- Review_scores_checkin: Score representing the efficiency of the check-in process of the property.
- Review_scores_communication: Score depicting the accessibility to communicate with the staff and property owner
- Review_scores_location: Score depicting how close the property is to nearby stores and tourist spots.
- Review_scores_value: Score representing whether the tenants think the overall rental experience is worth the money they spend
- Rent: daily price in local currency
- Polarity: facilities provided by the host
- Subjectivity: review comments written by the tenants

The review scores for various factors are ordinal variables ranging between the score of 1(worst) to 5(best). Amenities and comments are the only two nominal categorical variables describing the rental facilities and tenants reviews.

2.2 Exploratory Data Analysis

Sentiment Analysis

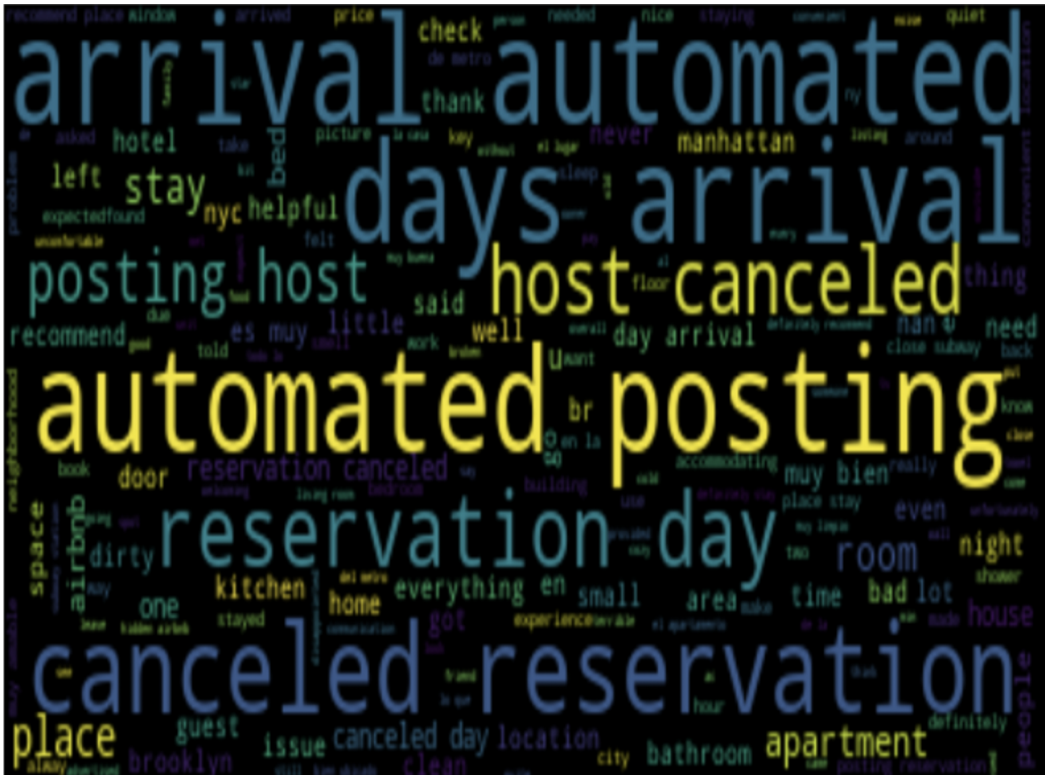
Sentiment analysis is performed on the textual data ‘comments’ to help determine whether the information is positive, negative or neutral. Based on the information collected, we can decipher the mood and emotions behind the text and better predict the correlation between sentiment of the comment and property rating.

Word Cloud Generation

Numpy is one of the most popular and helpful libraries used for multi-dimensional arrays and matrices. We used Numpy in combination with Pandas to perform our data analysis. We generated **word clouds**, or a visualization showing which words are used most frequently in a given text block. We created positive and negative sentiment word clouds of the words extracted from the filtered/cleaned comments based on the sentiment of the review. We used matplotlib to display these clouds.

Positive Sentiment Word Clouds

Negative Sentiment Word Clouds

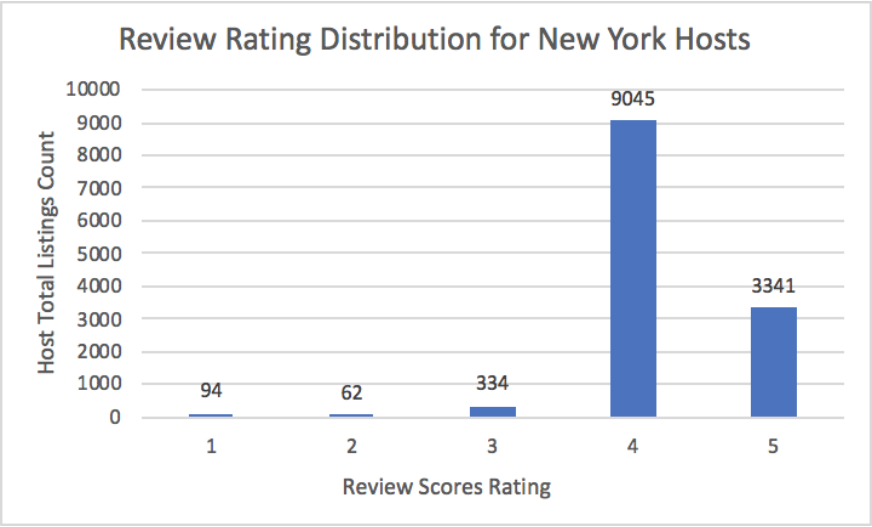


Textblob

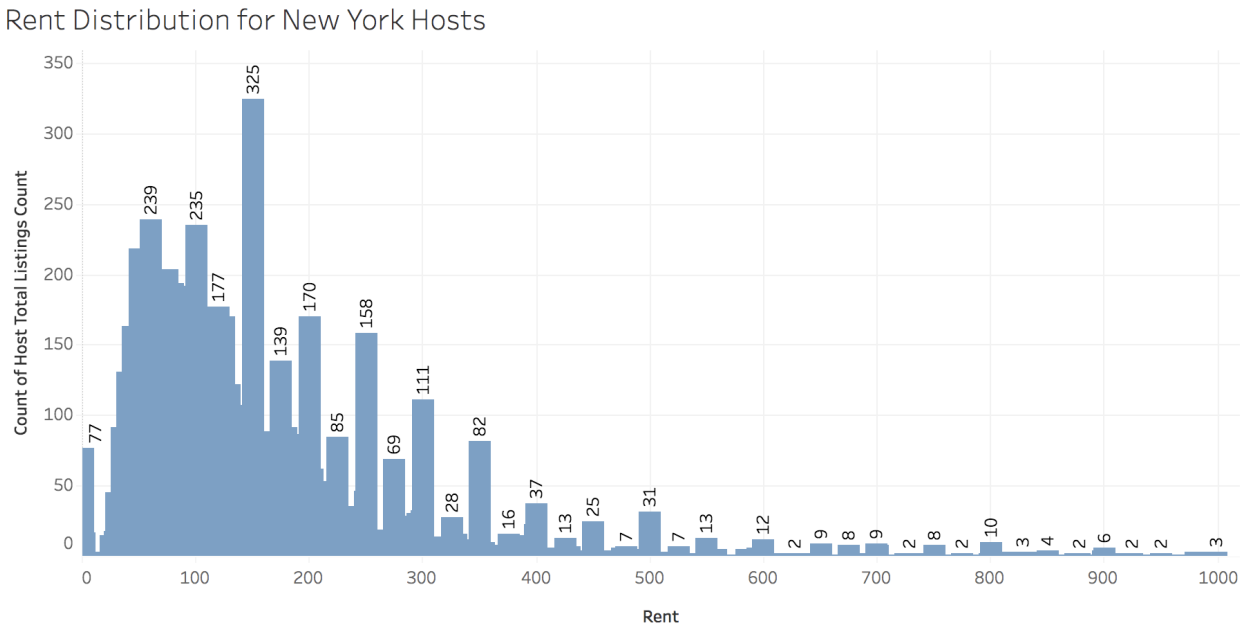
Textblob is another python NLP library used for supporting complex analysis and operations on textual data. We used textblob to determine the numeric values for **polarity** and **subjectivity**. “Polarity” values exist on a -1 to 1 scale, with -1 being a very negative sentiment and 1 being a very positive sentiment (0 of course is neutral). Subjectivity values lie on a 0 to 1 scale. Subjectivity describes how objective or subjective text is. Higher subjectivity (closer to 1) means the text contains personal opinion rather than factual information (where an “objective” score would be 0). Textblob also handles modifiers known as intensifiers (ie words like “very”, “extremely” etc) which intensify the meaning of text according to its pattern. Whenever an intensifier word is used, TextBlob will ignore polarity and subjectivity and just use intensity to compute the sentiment of the text.

Visualization

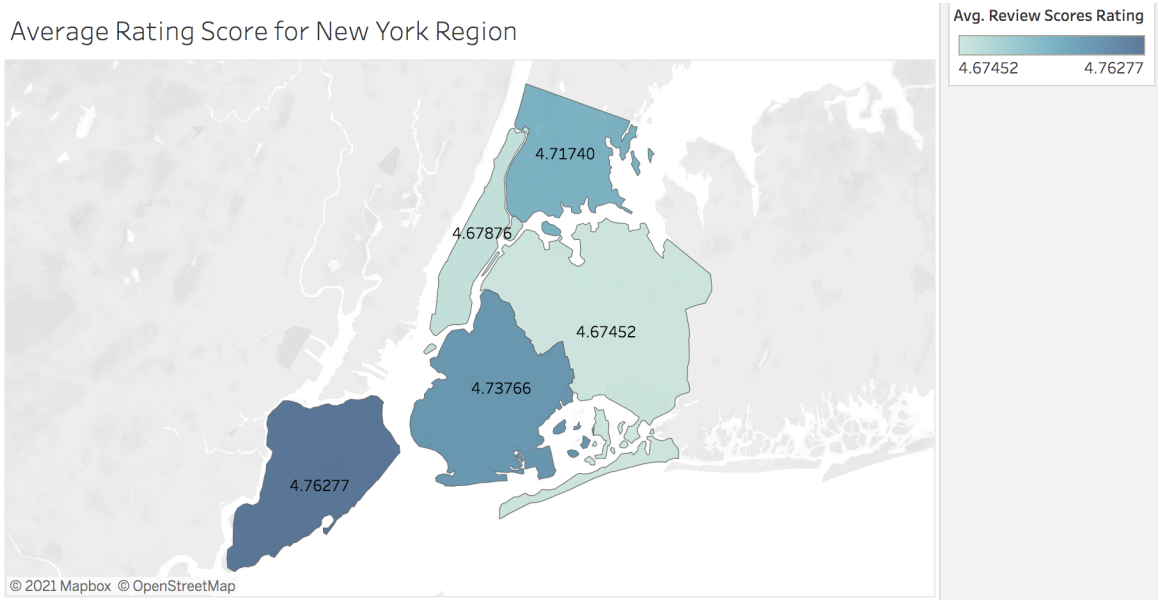
The bar chart indicates the amount of review ratings by the total host listings in New York. In general, tenants tend to leave good reviews, 4 stars and 5 stars mostly, rather than negative comments.



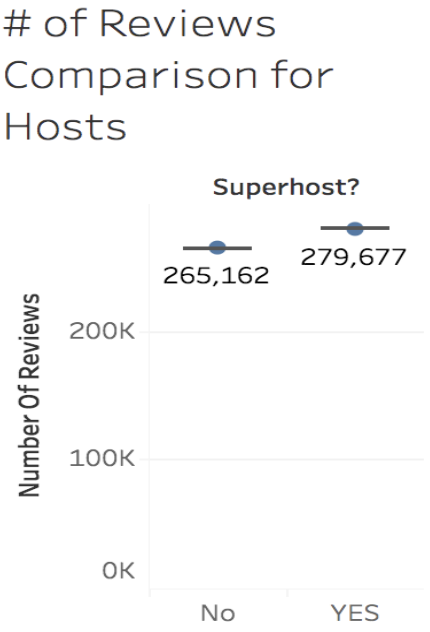
Rent distribution is shown in the distribution graph below. Most of the listing prices are between $90 to $350 per night, with the most expensive listings for $999. Factors affecting the rent pricing are features such as locations, accommodates, and amenities.



The map plot is used to show the average rating score for different New York regions: Bronx, Brooklyn, Manhattan, Queens, and Staten Island. The average rating score for the 5 regions are all above 4.6, with Queens having the worst average rating at 4.67452 and Staten Island having the highest average rating at 4.76277.



The relation between the number of reviews and whether the host is a superhost is presented in the chart below. Superhosts, top-rated and most experienced hosts, are likely to receive more reviews than normal hosts.



2.3 Data Engineering

With the above shown analysis, we can see that this is a classification problem. Hence, we will be using the below mentioned models to analyze the results:

1. Logistic Regression
2. Decision Tree
3. Random Forest

Using these models, we will be able to predict if a property will receive a good or bad rating based on sentiment analysis and average rating score.

The models have been trained with 70% of the dataset and the predictions have been made using the remaining 30% of the test dataset.

All the models above are using categorical variables as input which are various attributes of a listing for AirBnB. The model giving the best overall accuracy will be selected for prediction of a property receiving a good or bad review. Our dataset from

listings and reviews has various columns out of which we have converted the columns in listings dataset that are needed from nominal to ordinal or binary as mentioned in the data cleaning. Also, after cleaning the reviews dataset, we will be taking the average of polarity and subjectivity of all reviews for each listing.

In order to classify the ratings as good or bad for each listing, we will be storing the final rating as a binary variable depicting 0 as bad and 1 as good rating based on the average sentiment score as well as the average score rating for each listing.

2.4 Modelling Results

After performing the pre-processing and analysis on the datasets of listings and reviews, a binary variable named 'Ratings' has been created where an **average rating less than or equal to 4.7 is denoted by 1 and an average rating greater than 4.7 is denoted by 0.** Sentiment score is considered as the predictor variable. We are using three models to predict if the overall rating received by a property is "good" or "bad" based on sentiment analysis score and average rating score.

The accuracy rate of logistic regression with all variables is 68%
The accuracy rate of logistic regression with significant variables is 82%
The accuracy rate of decision tree is 87%
The accuracy rate of random forest is 89%

Logistic Regression

- **Benchmark Model**

The first machine learning model that we will be using is Logistic regression as we have the dependent variable as a binary variable denoting 0 for ratings greater than 4.7 and 1 for ratings lesser than or equal to 4.7. After importing the Logistic Regression module, we created a logistic regression classifier object using the LogisticRegression()

function. Afterwards we will fit the model on the train set using the function fit() and then perform prediction on the test set using predict() function.

Based on the logistic regression model, we are trying to answer the following queries:

- Which predictor variables have a significant impact on the probability if a property has a good or bad rating?

Below mentioned is the summary of logistic regression with all variables which has the accuracy of 68%

The classification matrix such as precision, accuracy and recall for the logistic regression **with all variables** are as mentioned below:

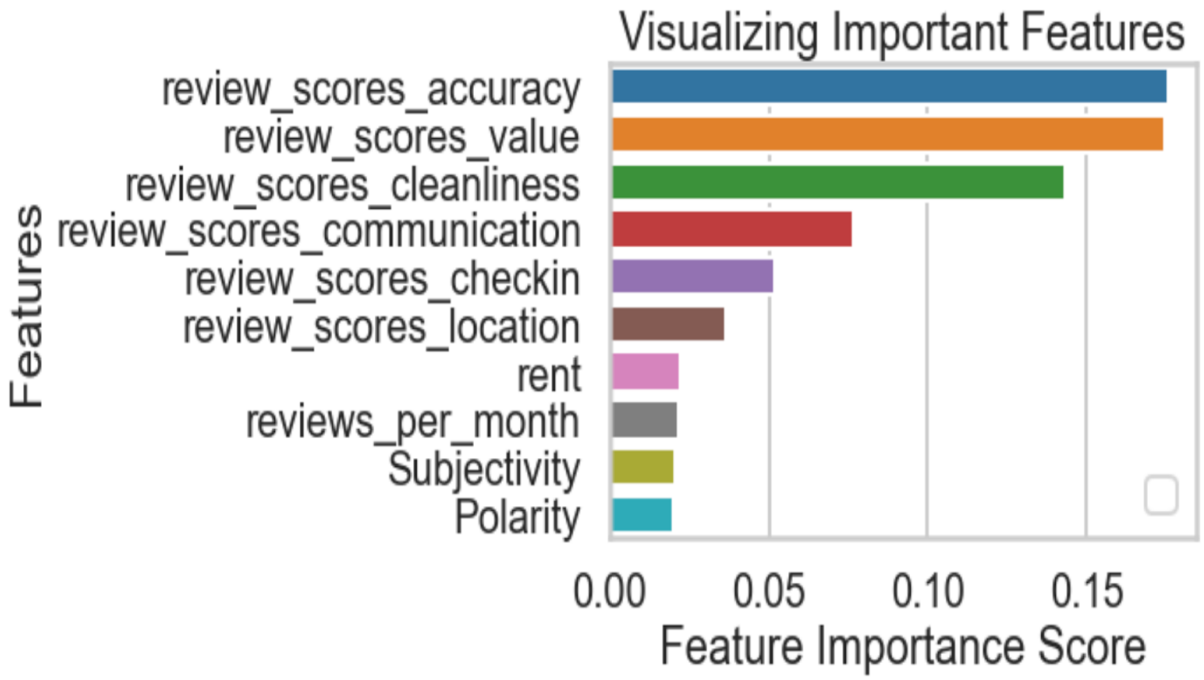
```
Accuracy of logistic regression classifier with all Variables: 0.68
Accuracy: 0.6833930704898447
              precision    recall  f1-score   support

           0       0.69      0.94      0.80      1121
           1       0.58      0.16      0.25       553

    accuracy                           0.68      1674
   macro avg       0.64      0.55      0.52      1674
weighted avg       0.66      0.68      0.62      1674
```

The key features which are highly impacting the accuracy of prediction were identified based on the p-values obtained from running the logistic regression model.

However, the below mentioned logistic regression has more accuracy of 82% as the model is run taking **only significant variables** into consideration.

```
Optimization terminated successfully.
         Current function value: 0.292887
         Iterations 8
                           Logit Regression Results
==============================================================================
Dep. Variable:                Ratings   No. Observations:                 5578
Model:                          Logit   Df Residuals:                     5568
Method:                           MLE   Df Model:                            9
Date:                Mon, 29 Nov 2021   Pseudo R-squ.:                 0.5375
Time:                        18:11:46   Log-Likelihood:                -1633.7
converged:                       True   LL-Null:                       -3532.7
Covariance Type:            nonrobust   LLR p-value:                     0.000
================================================================================================
                                   coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Intercept                      97.9725      3.001     32.648      0.000      92.091     103.854
review_scores_checkin          -1.8481      0.384     -4.819      0.000      -2.600      -1.096
reviews_per_month               0.0454      0.016      2.781      0.005       0.013       0.077
review_scores_accuracy         -5.9947      0.414    -14.463      0.000      -6.807      -5.182
review_scores_cleanliness      -3.9142      0.221    -17.718      0.000      -4.347      -3.481
review_scores_communication    -3.8983      0.415     -9.403      0.000      -4.711      -3.086
review_scores_location         -1.6668      0.203     -8.225      0.000      -2.064      -1.270
review_scores_value            -3.4218      0.290    -11.815      0.000      -3.989      -2.854
Subjectivity                    0.8386      1.064      0.788      0.431      -1.247       2.925
Polarity                       -0.5390      0.940     -0.574      0.566      -2.381       1.302
================================================================================================
```

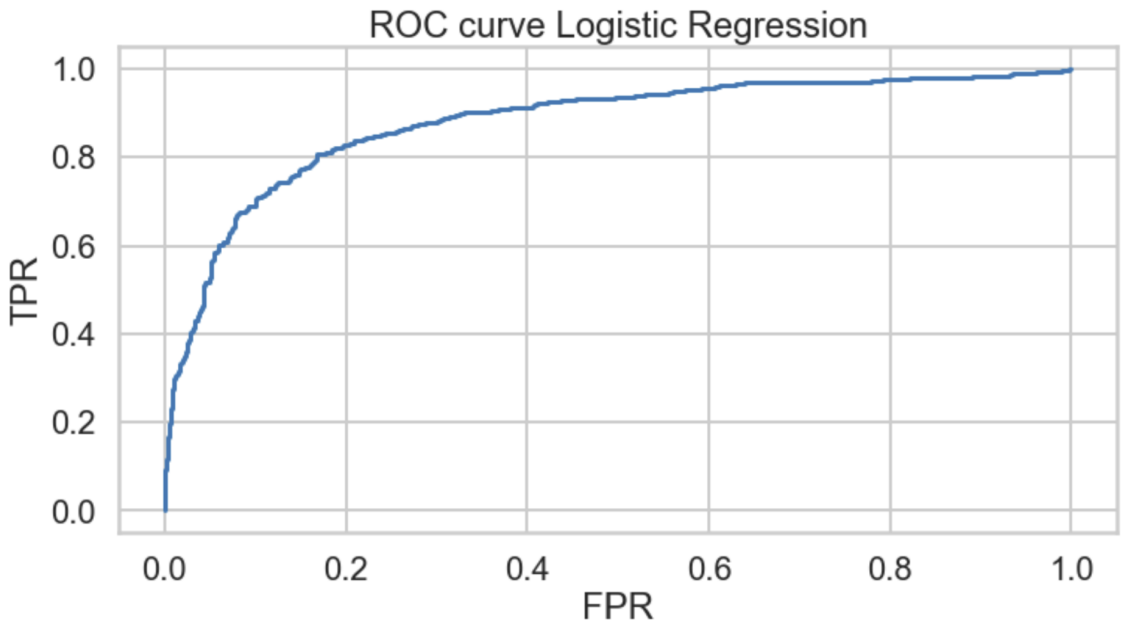
The most significant features identified for this model are:



The precision, accuracy and recall for the logistic regression with significant variables are as mentioned below:

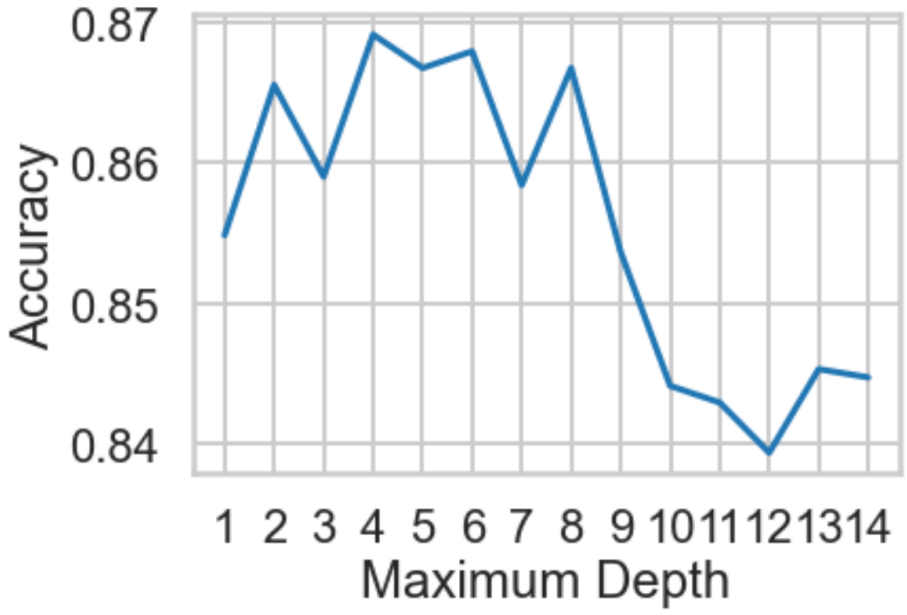
```
Accuracy of logistic regression classifier with significant variables: 0.82
Accuracy: 0.8225806451612904
              precision    recall  f1-score   support

           0       0.82      0.95      0.88      1121
           1       0.84      0.57      0.68       553

    accuracy                           0.82      1674
   macro avg       0.83      0.76      0.78      1674
weighted avg       0.82      0.82      0.81      1674
```

Receiver Operating Curve (ROC) providing the information regarding true positive rate against false positive rate is shown below and the AUC score is 0.88:

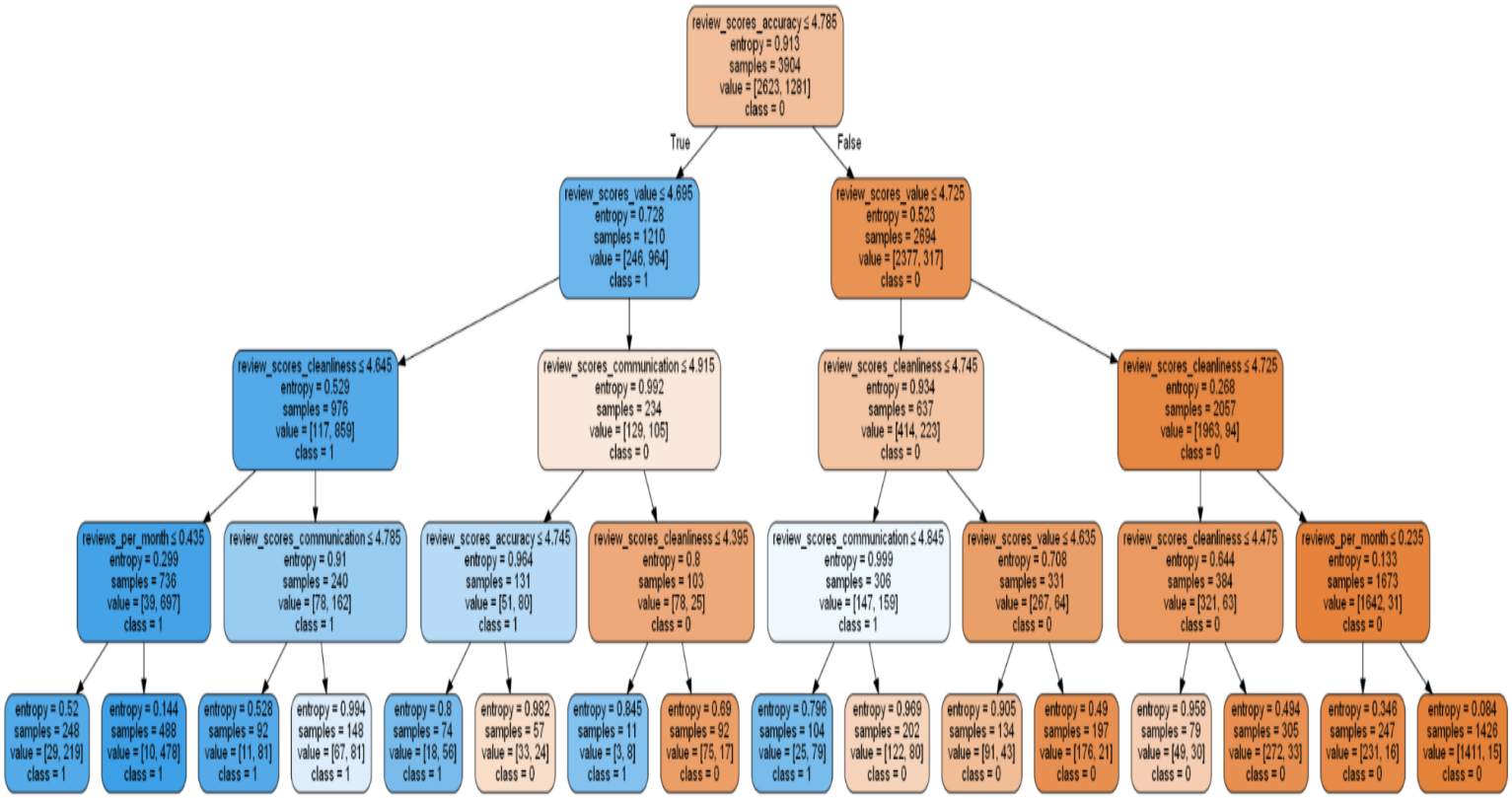


Decision Tree

Decision tree is another machine learning model used here as it splits observations into increasingly purer sub groups. Entropy is used to measure the impurity of the input set. To avoid overfitting, we used pre-prune, stopping the tree before it had completed classifying the training set, by specifying the maximum depth of a tree. The output graph below indicates that a tree with 4 layers should be created.



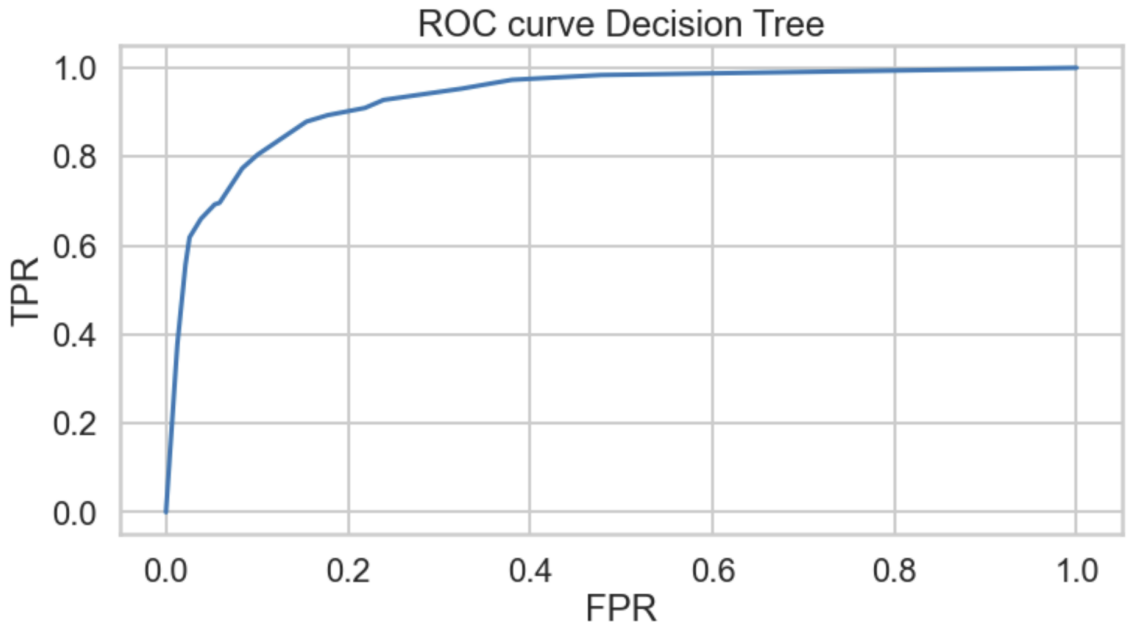
The decision tree is shown below:


review_scores_accuracy ≤ 4.785
entropy = 0.913
samples = 3904
value = [2623, 1281]
class = 0
True
False
review_scores_value ≤ 4.695
entropy = 0.728
samples = 1210
value = [246, 964]
class = 1
review_scores_value ≤ 4.725
entropy = 0.523
samples = 2694
value = [2377, 317]
class = 0
review_scores_cleanliness ≤ 4.645
entropy = 0.529
samples = 976
value = [117, 859]
class = 1
review_scores_communication ≤ 4.915
entropy = 0.992
samples = 234
value = [129, 105]
class = 0
review_scores_cleanliness ≤ 4.745
entropy = 0.934
samples = 637
value = [414, 223]
class = 0
review_scores_cleanliness ≤ 4.725
entropy = 0.268
samples = 2057
value = [1963, 94]
class = 0
reviews_per_month ≤ 0.435
entropy = 0.299
samples = 736
value = [39, 697]
class = 1
review_scores_communication ≤ 4.785
entropy = 0.91
samples = 240
value = [78, 162]
class = 1
review_scores_accuracy ≤ 4.745
entropy = 0.964
samples = 131
value = [51, 80]
class = 1
review_scores_cleanliness ≤ 4.395
entropy = 0.8
samples = 103
value = [78, 25]
class = 0
review_scores_communication ≤ 4.845
entropy = 0.999
samples = 306
value = [147, 159]
class = 1
review_scores_value ≤ 4.635
entropy = 0.708
samples = 331
value = [267, 64]
class = 0
review_scores_cleanliness ≤ 4.475
entropy = 0.644
samples = 384
value = [321, 63]
class = 0
reviews_per_month ≤ 0.235
entropy = 0.133
samples = 1673
value = [1642, 31]
class = 0
entropy = 0.52
samples = 248
value = [29, 219]
class = 1
entropy = 0.144
samples = 488
value = [10, 478]
class = 1
entropy = 0.528
samples = 92
value = [11, 81]
class = 1
entropy = 0.994
samples = 148
value = [67, 81]
class = 1
entropy = 0.8
samples = 74
value = [18, 56]
class = 1
entropy = 0.982
samples = 57
value = [33, 24]
class = 0
entropy = 0.845
samples = 11
value = [3, 8]
class = 1
entropy = 0.69
samples = 92
value = [75, 17]
class = 0
entropy = 0.796
samples = 104
value = [25, 79]
class = 1
entropy = 0.969
samples = 202
value = [122, 80]
class = 0
entropy = 0.905
samples = 134
value = [91, 43]
class = 0
entropy = 0.49
samples = 197
value = [176, 21]
class = 0
entropy = 0.958
samples = 79
value = [49, 30]
class = 0
entropy = 0.494
samples = 305
value = [272, 33]
class = 0
entropy = 0.348
samples = 247
value = [231, 16]
class = 0
entropy = 0.084
samples = 1426
value = [1411, 15]
class = 0

The summary of the decision tree mentioned below has an accuracy of 87%. The model evaluation matrix such as accuracy, precision and recall for the decision tree are as mentioned below:

```
Accuracy: 0.8691756272401434
Precision: 0.8199233716475096
Recall: 0.7739602169981917
              precision    recall  f1-score   support

           0       0.89      0.92      0.90      1121
           1       0.82      0.77      0.80       553

    accuracy                           0.87      1674
   macro avg       0.86      0.85      0.85      1674
weighted avg       0.87      0.87      0.87      1674
```

ROC providing the information regarding true positive rate against false positive rate is shown below and the AUC score is 0.93:



Random Forest

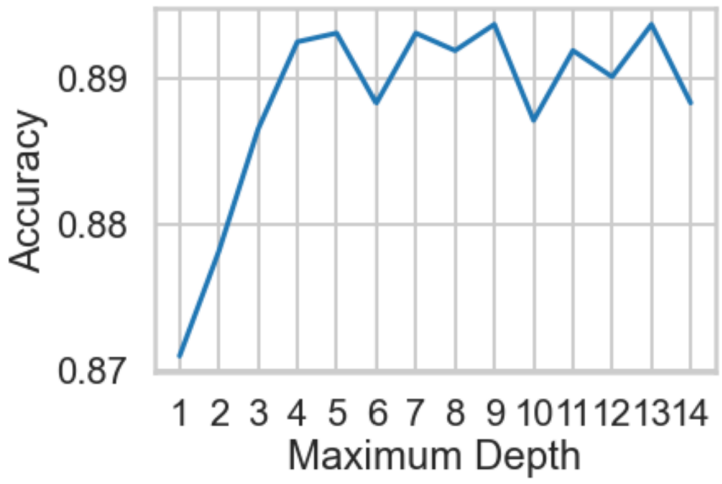
In order to overcome the limitations of a single decision tree, we also tried using Random Forest machine learning model which has 2 key concepts:

1. Random sampling of training data points when building trees
2. Random subsets of features considered when splitting nodes

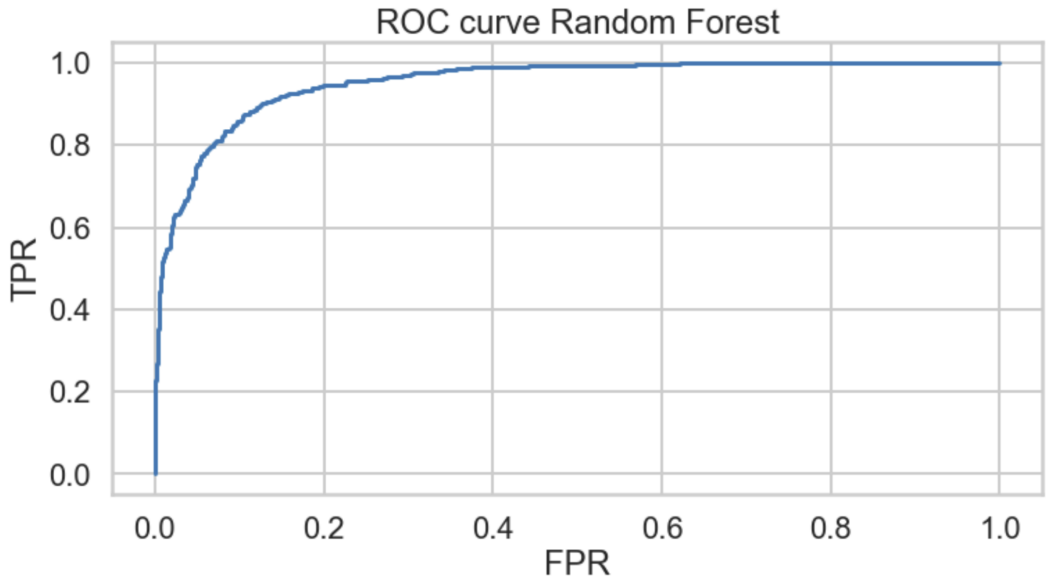
Random Forest has the maximum accuracy of 89% among all the models. The model evaluation matrix such as precision, accuracy and recall for the Random Forest are as mentioned below:

```
Accuracy: 0.8918757467144564
Precision: 0.8381818181818181
Recall: 0.833634719710669
              precision    recall  f1-score   support

           0       0.92      0.92      0.92      1121
           1       0.84      0.83      0.84       553

    accuracy                           0.89      1674
   macro avg       0.88      0.88      0.88      1674
weighted avg       0.89      0.89      0.89      1674
```

The depth of random forest model here is 9 i.e. the output of 9 layers should be created:



Receiver Operating Curve(ROC) providing the information regarding true positive rate against false positive rate is shown below and AUC score is 0.95:



3.1 Inference and Key Takeaways

For logistic regression analysis, the regression model was run twice. The benchmark was set by the logistic regression with all the variables. It was run with all the variables in the first trial but since the model evaluation matrix such as precision, recall score, accuracy is not quite as low when compared to the second regression analysis we ran our analysis again using the “significant” variables. However, this model is the weakest among all the models that we have used based on the recall score which is the model’s ability to correctly predict the positives out of actual positives.

Decision Tree analysis is a bit better than Logistic Regression as the recall score is more in case of Decision tree model. But if we try running other models, we see that other models are performing better than the decision tree for the given dataset.

For Random Forest analysis, the precision score that measures the model’s ability to predict the positive outcomes out of all the predictions is the highest. The

accuracy of predicting the variables out of actual outcomes, which is the recall score, is also the highest. Therefore, this model performed the best among all the models we used.

	Precision	Recall	F1-Score	AUC
Logistic Regression(BenchMark)	0.58	0.16	0.25	0.669504
Logistic Regression	0.84	0.57	0.68	0.880348
Decision Tree	0.82	0.77	0.8	0.931682
Random Forest	0.84	0.83	0.84	0.953648

3.2 Conclusion

Our goal was to create a method by which Airbnb managers would be able to identify the drivers of average star rating in order to make more informed choices surrounding their properties; whether that was a decision on if they should buy a new property based on qualities exhibited by the for sale rental in question or not, adjusting prices on units to match renter expectations, ensure important features/amenities were on hand during tenancies and ultimately to boost Airbnb rental average scores and determine whether or not their property would be able to compete with the best properties New York has to offer (or properties which have a 4.7 or higher average review score on the Airbnb platform) on a pass/fail score based on whether or not a property is above the average rating of our dataset or not. By implementing machine learning methods we learned in our classes we were able to successfully create multiple models which accomplish this; with our highest accuracy model being the random forest model with an 89% accuracy rate.