PRACTICAL BUSINESS ANALYTICS-COMM053

Group Name – Data Alchemists(Group7)

Coursework Report

Airbnb Price Analysis

Team Members:

Jithin Jaison	6700210
Binayak Rout	6675500
Krishnapriya Chitharanjan	6700508
Sahla Marjan	6711257
Ayona Biju	6707640

1.PROBLEM DEFINITION

To build a model to analyse the price of listed Airbnb's and determine if the price is reasonable or overpriced based on other exogenous features from the chosen dataset.

Hosting sites like Airbnb has grown tremendously over the past few years. Airbnb, Inc is a California based online platform which lists rental sites. It is a household property that holds rent on a short-term basis to visitors. It has provided temporary housing solutions for a variety of people over variety of places. The whole platform can be accessed via website and App which makes it convenient and hassle-free for both hosts and guests. The platform builds profit by the commission received from the hosts of rental properties.

The presupposition of the group aim is to observe if any of the factors in the data set has determinative relationship with the price of rentals. Also, the other relationships between different features in the dataset, if any. The Model will also try to find any non-linear relationship between the aspects of database and the price. The model should also be able to classify a new set of data points into 2 categories when a new set of inputs are given.

In the project, the group is trying to build a model that gives suggestions to the host of a property if the price they gave is reasonable or overpriced. This feedback mechanism will help the host to price the property competitively so that probability of occupancy is increased. This is beneficial to both the host and the company. This is because more and more people will be encouraged to list their property in Airbnb as it provides general trend of market price of similar properties. This in turn increases the revenue of the business.

The dataset used for this project is obtained from Kaggle (https://www.kaggle.com/chadra/ab-nyc-2019). The dataset has different parameters for the year 2019 over a number of cities in Newyork.

2. DATA DESCRIPTION

The data set includes 3 main parts:

- Description of each listing : This dataset contains information about hosts, Airbnb houses and price. The attributes available are id (listings ID), name (name of the Airbnb houses), host_id (ID of the host), host_name (name of the host), room_type (Type of the room. .i.e private room, shared room or Entire home) and price (in dollars), calculated_host_listings_count (Properties owned by the particular host).
- Reviews : Reviews given by the customers. This includes number_of_reviews (number of reviews for each listing), last_review (date of the last review received), reviews_per_month (average number of reviews received per month)
- Location : Provides information about location of the AirBNB listings in the NewYork city. This includes neighbourhoodgroup (name of the borough in which it is located), neighbourhood(name of the area), latitude and longitude.

2.2 Data Dictionary for Dataset

The input fields (predictor, explanatory, or independent variables) fields are:

Input Field	Description
id	Unique ID of each AirBnB property
name	Name of AirBnB property
host_id	ID of property's host
host_name	Name of the host
neighbourhood_group	Boroughs of New York City
neighbourhood	Distinct neighbourhoods in each Borough
latitude	Latitude of the property
longitude	Longitude of the property
room_type	Types of Room of the property
minimum_nights	Minimum no of nights to book the property
number_of_reviews	Number of reviews for each property
last_review	Date of latest review
reviews_per_month	Reviews per month per property

calculated_host_listings_count	No of AirBnB properties owned by the host
availability_365	No of days the property is available in an year

Output Field	Description
price	Price of each property of AirBnb

3.DATA CLEANING & EXPLORATORY ANALYSIS

When data is collected from different sources and put together, the dataset may contain lot of anomalies. These anomalies needs to be removed or modified before data analysis. The processing of data for analysis by eliminating or altering data which is inaccurate, insufficient, unimportant, or incorrectly formatted is called Data Cleaning. This data is unnecessary in terms of data analysis as it may negatively affect the results. The processing might also be time consuming if data is not cleaned.

The data set contains 48895 records with 16 features. The data set needed substantial amount of work to clean as it was filled with inaccurate, incomplete, unbalanced and irrelevant data. These types of data will lead to false conclusions and mislead the model into making erroneous fiscal decisions. In order to identify these inconsistencies, several visualisation techniques were used. The following steps were undertaken as a part of cleaning and exploration:

- Some rows in the dataset had room price as zero dollars which is an untrue value as in reality the price of rentals cannot be zero. Since the number of records that had this anomaly were 0.02%, it was evident that removal of these data points will not highly affect the purpose of the data set. Hence it was removed.

- For the better understanding of data set, the distribution of different room types in each neighbourhood is plotted using a bar graph.

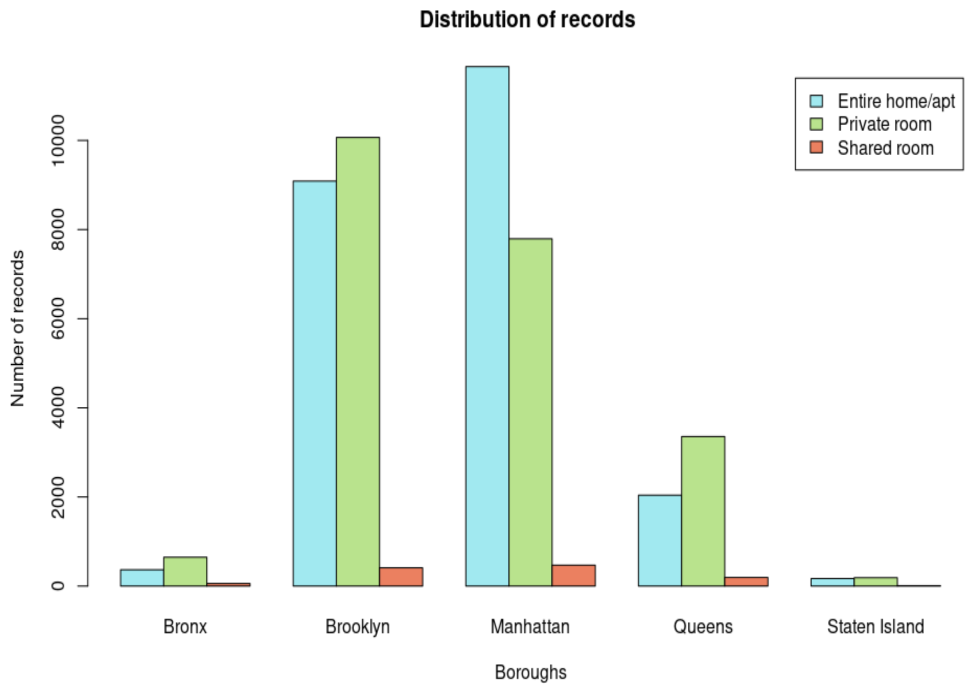


Fig:3.1.Distribution of types of rooms in different locations

Fig3.1, shows that Manhattan and Brooklyn have more apartments and rooms which represents a high demand.

- Removal of outliers play a significant part in data cleaning process. These are values that lies at an abnormal distance from other values in the data set. The presence of these values will make it difficult to generalise the relationships. The visualization technique used for identifying the outliers is called box plots. The points beyond the outer fence are considered as an outlier.

Boxplot is used for graphically visualising groups of data into different sections called quartiles to find out outliers. It is constructed by creating a box between upper and lower quartiles and a median separating both quartiles. Fences are required to determine outliers. A point beyond an inner fence on either side is considered a **mild outlier**. A point beyond an outer fence is considered an **extreme outlier**.

To determine the outliers in price field the prices are plotted as shown in the fig 3.2:

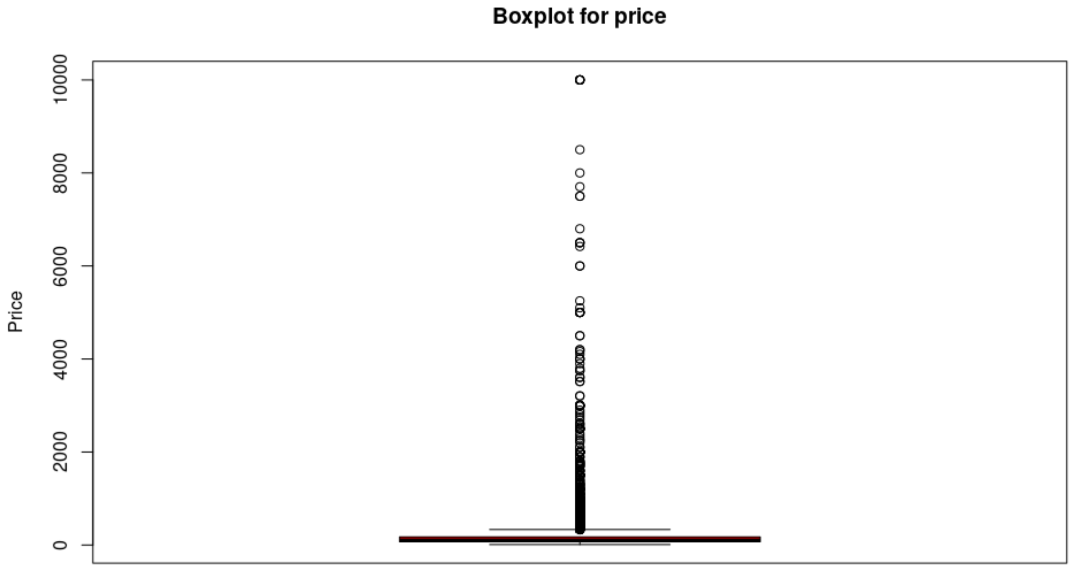


Fig 3.2: Box plot of price values

Removal of outliers based on this analysis alone may lead to elimination of valuable information. Based on the domain knowledge, it is logical to conclude that different boroughs may have different price for rentals based on the popularity of the place. For the purpose of validating that such useful information is not lost, the box plots of price per boroughs are plotted as shown in Fig.3.3.

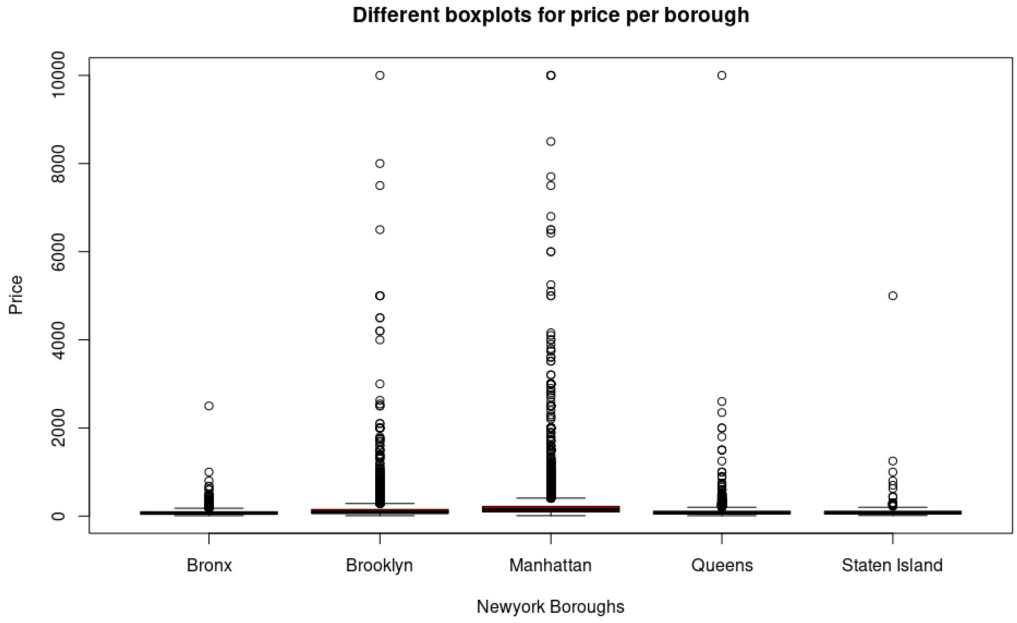


Fig:3.3: Box plot of price values for each borough

After comparing the plots in Fig 3.2 and Fig 3.3, it is evident that the prices above and below the upper and lower whiskers does not provide any vital knowledge. Since these values comprise of less than 5% of the entire data and hence, they were removed.

- Feature selection: ID, host name, host ID ,name and last review were removed to reduce dimensionality and as they do not significantly impact the model.
- The records showed NULL values in the data set in the column reviews_per_month, which could be because the number of reviews received for the data point is zero. These points were replaced with zero value in the reviews_per_month column.
- A visualisation method called heat map was used to examine the correlation between the features in the data set. It determined if there was any linear trend between variables.

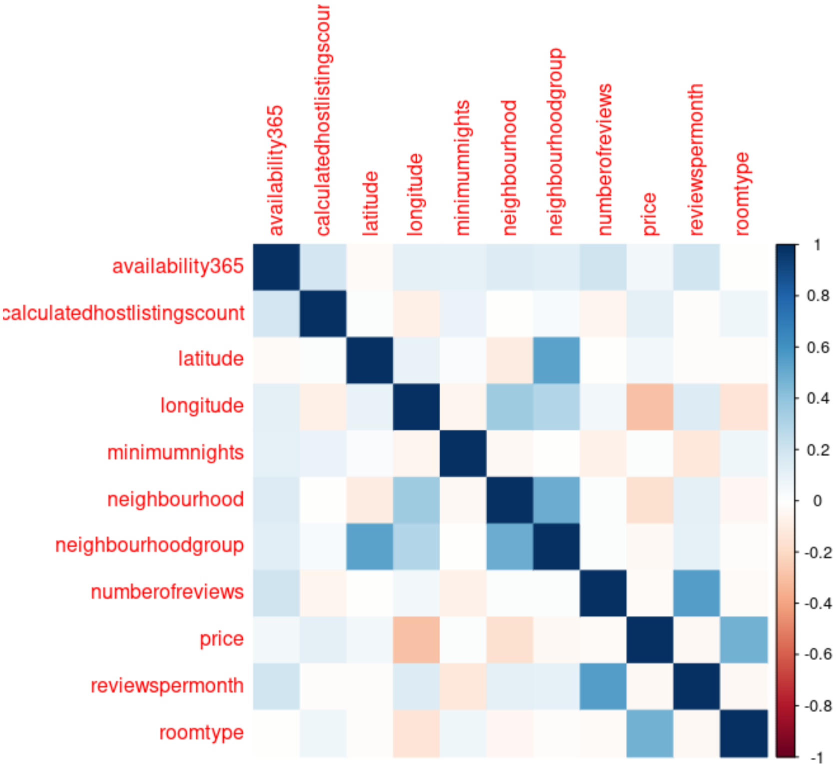
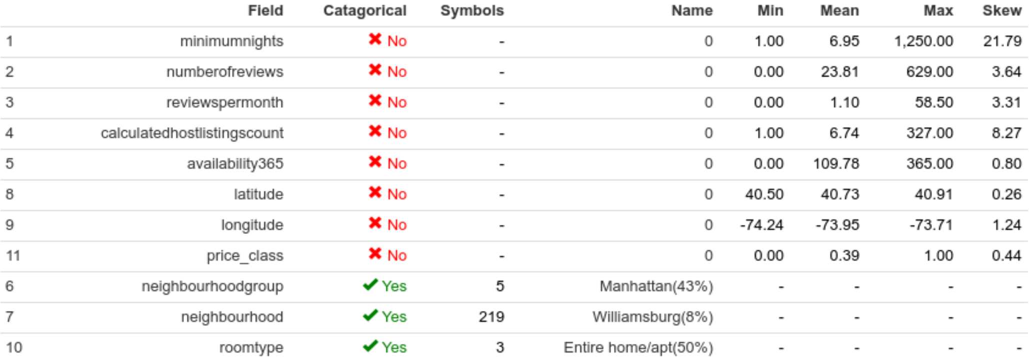


Fig:3.5: Heat map of features

The Fig 3.5 shows the correlation between different feature. The variation in colour gives the intensity with which the variables are related. If the grid shows colour corresponding to zero, this signifies that there is no linear relationship between the attributes. In order to find the correlation, Pearson correlation coefficient is used.

The initial assumption was that the price of room had a close relationship with the number of reviews. But it is noticeable from the Fig.3.5 that the number of reviews and price did not have a large positive correlation. This could be because number of reviews is an absolute value and does not give the information about the number of positive and negative reviews. So, the number of reviews does not necessarily have a relationship with the price. However, the price and room type have an observable correlation.

- The data set columns need to be analyzed to determine which all are symbolic and categorical. This is important as the decide which all fields need to be encoded.

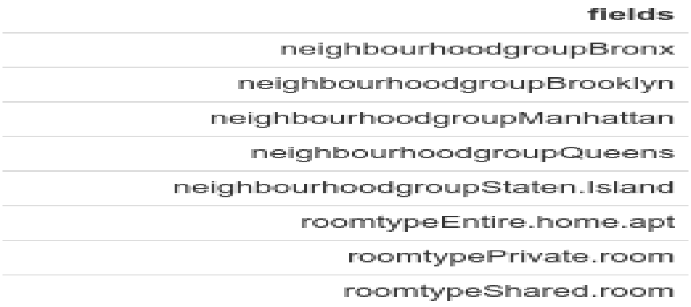


	Field	Catagorical	Symbols	Name	Min	Mean	Max	Skew
1	minimumnights	✖ No	-	0	1.00	6.95	1,250.00	21.79
2	numberofreviews	✖ No	-	0	0.00	23.81	629.00	3.64
3	reviewspermonth	✖ No	-	0	0.00	1.10	58.50	3.31
4	calculatedhostlistingscount	✖ No	-	0	1.00	6.74	327.00	8.27
5	availability365	✖ No	-	0	0.00	109.78	365.00	0.80
8	latitude	✖ No	-	0	40.50	40.73	40.91	0.26
9	longitude	✖ No	-	0	-74.24	-73.95	-73.71	1.24
11	price_class	✖ No	-	0	0.00	0.39	1.00	0.44
6	neighbourhoodgroup	✔ Yes	5	Manhattan(43%)	-	-	-	-
7	neighbourhood	✔ Yes	219	Williamsburg(8%)	-	-	-	-
10	roomtype	✔ Yes	3	Entire home/apt(50%)	-	-	-	-

Fig 3.6: Field description

The Fig3.6 shows what is the current field types. 8 fields are non-categorical and 3 are categorical. The symbolic fields are neighbourhood group, neighbourhood and room type.

1-hot encoding can be used to encode the data set. This uses a single binary value for each category and more dimensions will be added to the existing data set. The group has decided to encode neighbourhood group and room type. The neighbourhood information will be available in latitude and longitude fields. So retaining the location information is not necessary. Moreover, neighbourhood attribute has 219 unique values hence it cannot be encoded using 1 hot encoding. Hence this field can be entirely removed.



fields
neighbourhoodgroupBronx
neighbourhoodgroupBrooklyn
neighbourhoodgroupManhattan
neighbourhoodgroupQueens
neighbourhoodgroupStaten.Island
roomtypeEntire.home.apt
roomtypePrivate.room
roomtypeShared.room

Fig:3.7: Added fields after 1-Hot encoding

The Fig 3.7 shows the new fields added to the existing data set after 1- hot encoding.

The original fields *neighbourhood group* and *room type* are removed from the data set.

- Threshold selection: In order to separate the price values into reasonable and overpriced, it is important to find a threshold value. Hence, following assumptions were made to derive a logical conclusion.

If the prices are comparable in each neighbourhood, then it is rational to take the average price value as cut-off. Thus, box plot is used for understanding the distribution of records for each neighbourhood group as shown in Fig 3.8.

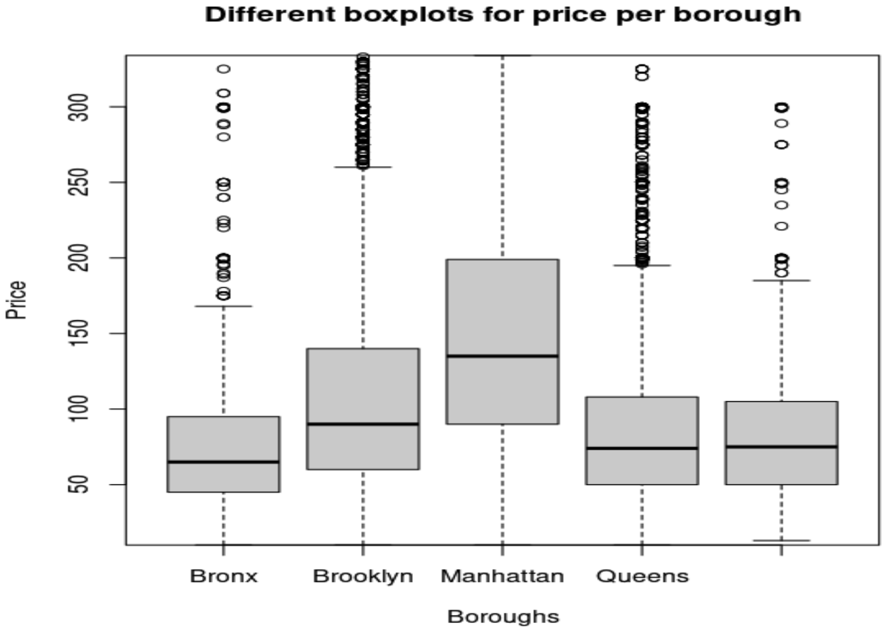


Fig:3.8 Box plot of prices in each Boroughs

From the Fig.3.8, it can be determined that Bronx, Queens and Staten Island has comparable room price, whereas, Manhattan and Brooklyn have higher values. Thus, a general average of prices will not provide an acceptable threshold.

Thus, it's not reasonable to make a single threshold for price distinction. Hence, the group utilized both room type and neighborhood group to compute the threshold. The average of each neighbourhood group is considered to classify the data point into reasonable and overpriced.

- Sampling: For better efficiency of the model, it is important to feed a balanced data set to prevent the model from over or under fitting. Bar plots are used to determine if the data set available is balanced or not.

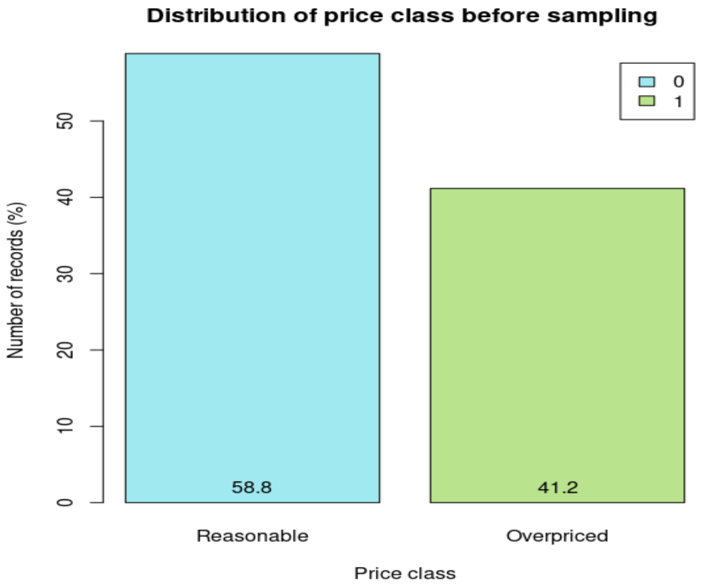


Fig: 3.9 Distribution of price class before sampling

In the Fig.3.9 it can be recognized that 58.8% are reasonably priced and 41.2% are overpriced. There is a clear difference of 17% of data. Hence, data sampling needs to be done as a part of preprocessing.

Basically, there are 2 types of sampling, over sampling and under sampling.

1) In **over samplin**g, the data in the minority data set are chosen and are added again to the data set multiple times. If the copies of same data are added again and again, there is a possibility of overfitting.

2) **Under sampling**, is decreasing the data from majority class and keeping the data in minority class as such. If this is done, there is possibility of removing data with vital information.

Therefore, both under and over sampling were done to the data set. The data distribution after sampling can be seen in the Fig.3.10

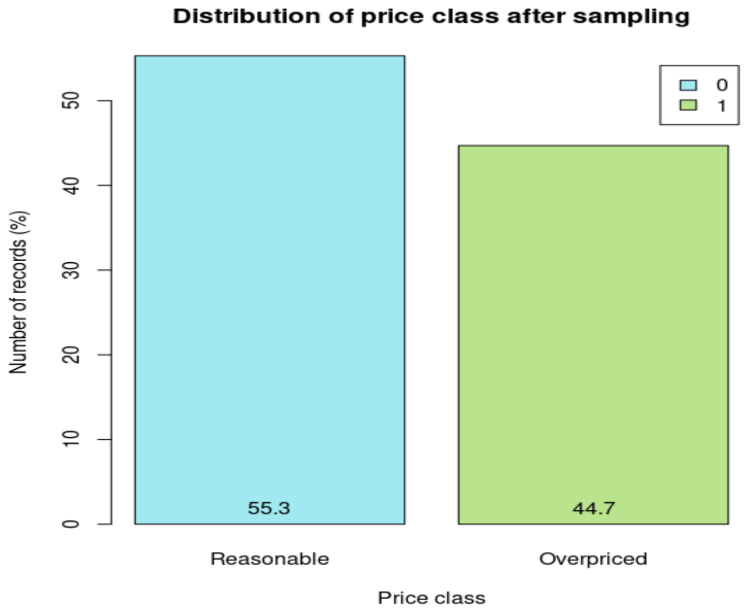


Fig:3.10. Distribution of price class after sampling

From the bar plot, Fig 3.10, the difference in data point distribution in both the classes are significantly reduced.

4. SELECTION OF MODELLING APPROACHES

After completing the preprocessing of data set by removing the outliers, encoding the required fields, selecting the threshold value, and sampling the data set, the group selected suitable model algorithms for regression and classification.

As mentioned before, the prime objective of the project is to predict if the price is overpriced or reasonable based on all other parameters. This falls under binary classification problem in which we are segregating the classes in to either of the 2 models. Hence, logistic regression is taken as the initial model which would give the probability of a certain data point belonging to a particular class. Since, it is evident that the data is not linearly separable, a nonlinear classifier needs to be used. Thus, decision tree is being chosen as the 2nd model. The 3rd model used is random forest which works well on large data sets and suitable for both categorical and numerical data. Also, it has lower tendency to overfit. As the random forest is a collection of several decision trees, it is a longer and slower process but typically more accurate than decision trees. The final model selected is deep neural network which uses circuit of artificial nodes.

5. MODELS & EVALUATION

5.1 Logistic regression

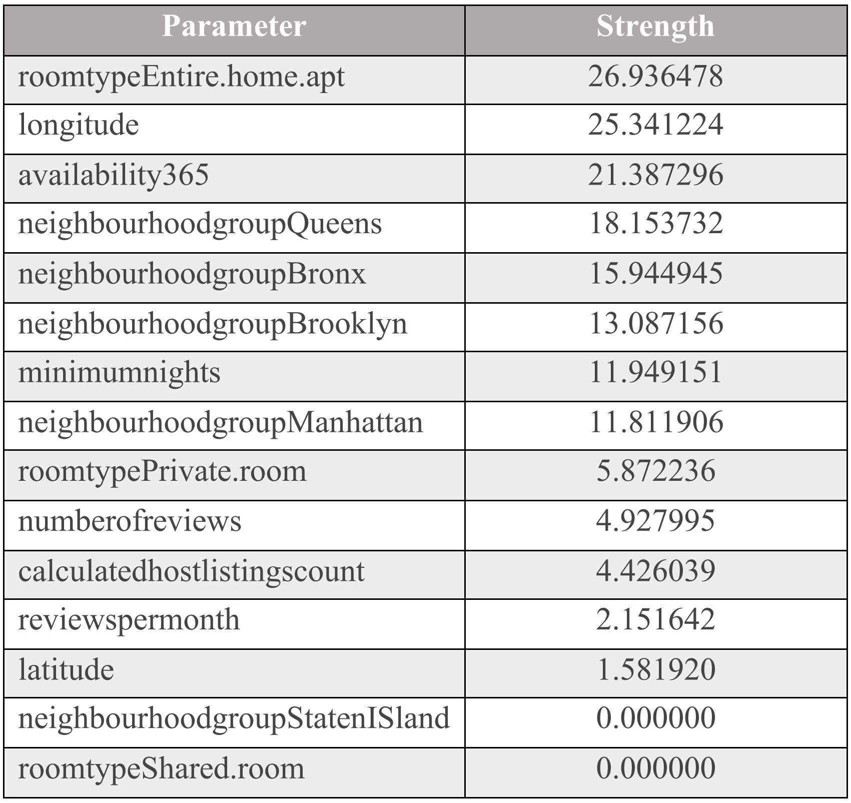
Logistic regression is a commonly used method to predict binary classes. The outcome or target variable is dichotomous in nature.

Since the dependent variable is divided into overpriced and reasonable which is a binary classification, the most appropriate model to start with is logistic regression. Logistic regression with a train to test ratio of 70/30 is used. The performance of the model can be explored using the below table 5.1.1

Measure	Value
Precision	0.7343
Sensitivity	0.8158
Specificity	0.7919
F-score	0.7727
Accuracy	80.18%
AUC of ROC chart	85.39%
MCC	0.6005
TPR	81.58%
FPR	20.80%
TNR	79.19%
FNR	16.57%
Threshold	0.57
pgood	73.43%
pbad	85.91

Table 5.1.1

The resulting ROC determined a threshold of 0.57 with a TPR = 81.58 and FPR = 20.80



Parameter	Strength
roomtypeEntire.home.apt	26.936478
longitude	25.341224
availability365	21.387296
neighbourhoodgroupQueens	18.153732
neighbourhoodgroupBronx	15.944945
neighbourhoodgroupBrooklyn	13.087156
minimumnights	11.949151
neighbourhoodgroupManhattan	11.811906
roomtypePrivate.room	5.872236
numberofreviews	4.927995
calculatedhostlistingscount	4.426039
reviewspermonth	2.151642
latitude	1.581920
neighbourhoodgroupStatenISland	0.000000
roomtypeShared.room	0.000000

Table 5.1.2 strengths of the variables in the logistic model

The significance of the variables can be shown a bar plot as below:

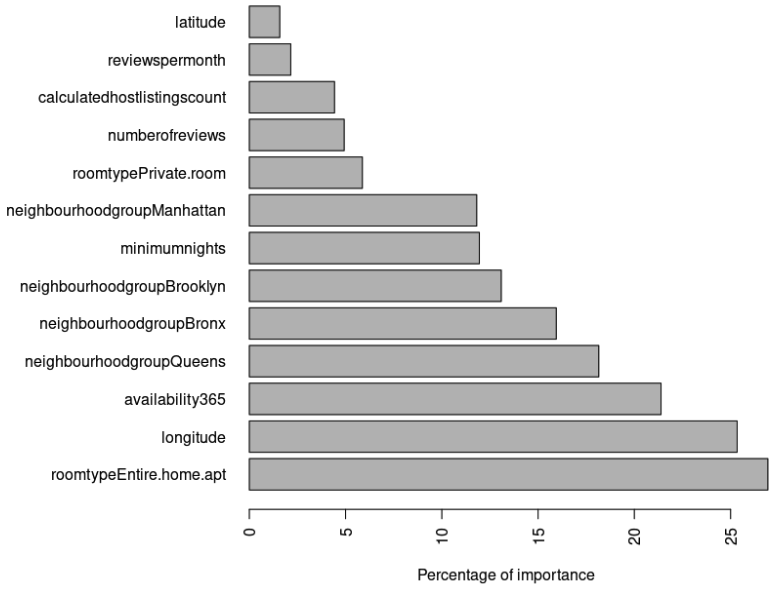


Fig:5.1.1 Strengths of the variables in logistic model

ROC for the logistic model is as shown below

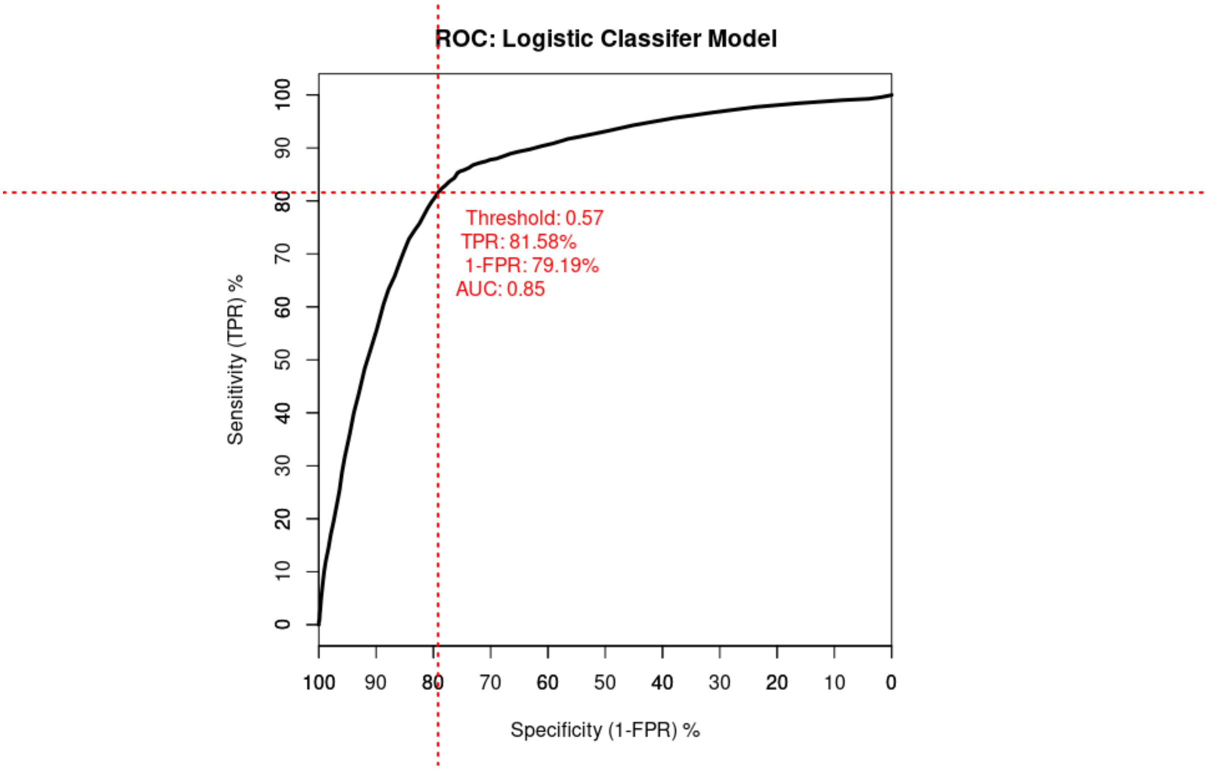


Fig:5.1.3 ROC for Logistic classifier model

Confusion matrix for the testing data set is as shown below:

n = 13774		**Actual**	
		Class 0	Class 1
Predicted	Class 0	6398	1681
	Class 1	1049	4646

5.2 Decision Tree

Decision Tree is a method of supervised learning. Both regression and classification problems can be solved by using decision tree algorithm. The objective of utilizing a Decision Tree is to make a preparation model that can be used to anticipate the class or worth of the objective variable by taking in basic decision guidelines deduced from earlier data (training information). In Decision Trees, for predicting a class name for a record we start from the base of the tree. We think about the upsides of the root quality with the record's characteristic. Based on correlation, we follow the branch comparing to that value and jump to the following node.

A C5.0 decision tree is used in which 51 rules were derived using the full set of variables.

Measure	Value
Precision	0.7642
Sensitivity	0.8093
Specificity	0.824
F-score	0.7861
Accuracy	81.79%
AUC of ROC chart	86.06%
MCC	0.6286
TPR	80.93%
FPR	17.60%
TNR	82.39%
FNR	18.00%
Threshold	0.4
pgood	76.42%
pbad	85.97%

Table 5.2.1 decision tree result measurements

Confusion matrix for the model is given below:

n = 13774		**Actual**	
		Class 0	Class 1
Predicted	Class 0	6657	1422
	Class 1	1086	4609

Parameter	Strength
roomtypeEntire.home.apt	100
calculatedhostlistingscount	93.22
longitude	90.11
availability365	78.46
latitude	62.52
minimumnights	57.45
neighbourhoodgroupBronx	6.42
neighbourhoodgroupQueens	4.39
neighbourhoodgroupBrooklyn	4.23
numberofreviews	4.0

neighbourhoodgroupManhattan	3.70
roomtypePrivate.room	3.59
reviewspermonth	3.5
neighbourhoodgroupStaten.Island	0.0
roomtypeShared.room	0.0

Table 5.2.2 strengths of the variables in the Decision tree model:

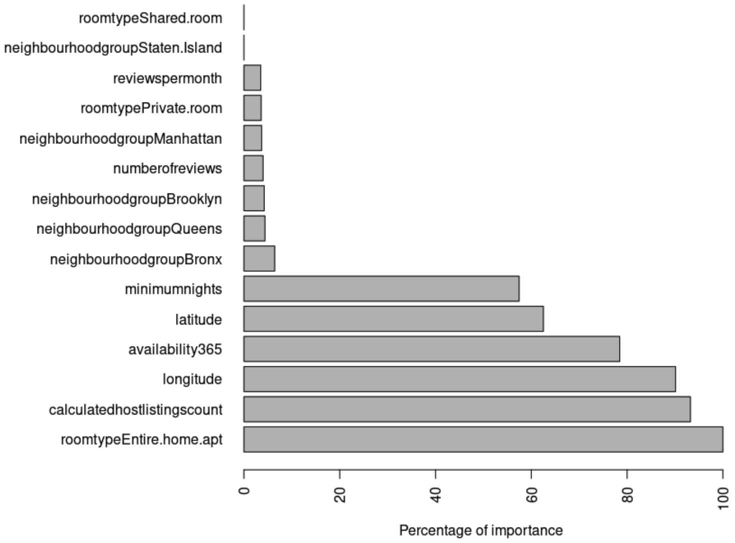


Fig:3.3: Strengths of the variables in Decision tree model

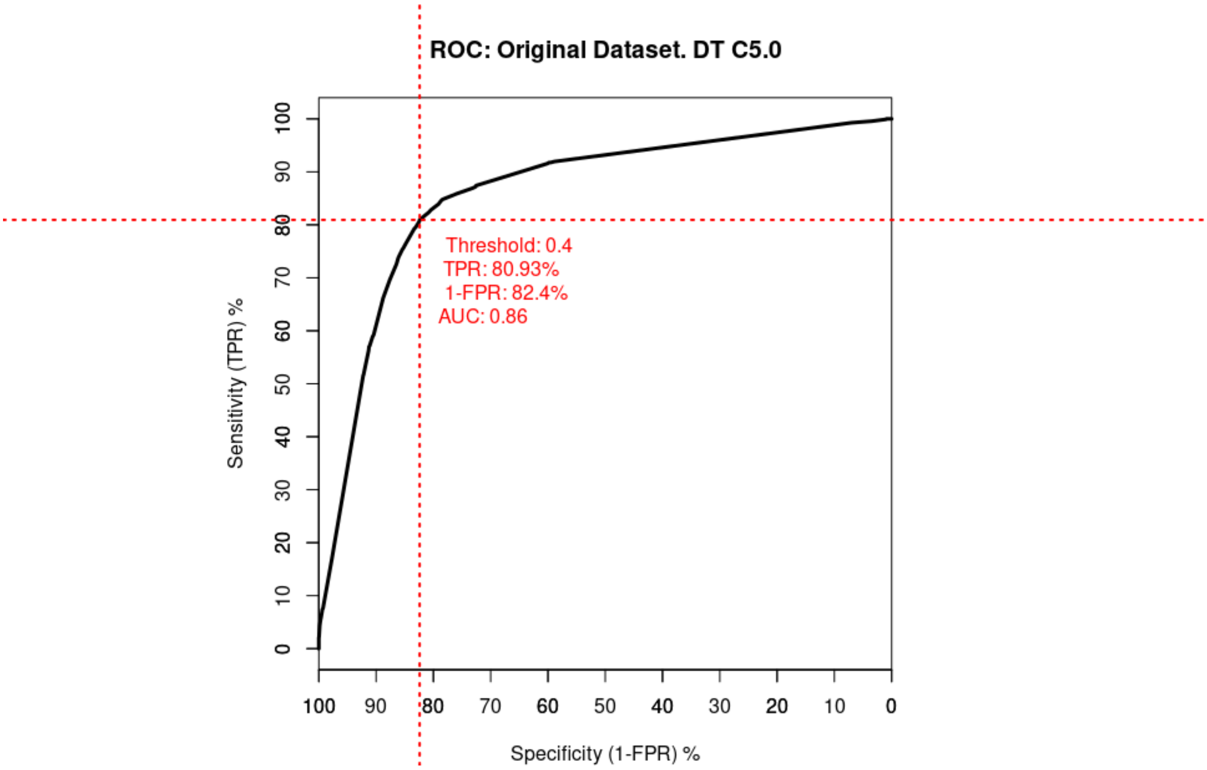


Fig:3.4: ROC of the decision tree C5.0

5.3 Decision tree C5.0 boosted

The type of Boosting used here is Adaptive Boosting. It is a type of Boosting ensemble which works well with decision tree. It learns from the mistakes by increasing the weight of misclassified data points

A C5.0 decision tree with boost=20 produced the following results.

Measure	Value
Precision	0.7614
Sensitivity	0.8248
Specificity	0.8178
F-score	0.7918
Accuracy	82.07%
AUC of ROC chart	88.65%
MCC	0.6363
TPR	82.47%
FPR	18.22%
TNR	81.77%
FNR	16.177%
Threshold	0.44
pgood	76.13%
pbad	86.87%

Table 5.3.1 Resulting Measurement

The confusion matrix for the decision tree boosted is given below:

n = 13774		**Actual**	
		Class 0	Class 1
Predicted	Class 0	6607	1472
	Class 1	998	4697

Parameter	Strength
roomtypeEntire.home.apt	100
roomtypeShared.room	100
minimumnights	100
calculatedhostlistingscount	100
availability365	100
longitude	99.84
latitude	99.74
numberofreviews	98.73
reviewspermonth	98.72
neighbourhoodgroupBronx	97.62
neighbourhoodgroupManhattan	86.44
neighbourhoodgroupQueens	81.23
neighbourhoodgroupBrooklyn	75.82
roomtypePrivate.room	43.36
neighbourhoodgroupStaten.Island	28.21

Table 5.3.2 The strengths of the input fields

In particular, it can be noted that 5 features registered 100% importance, indicating that Boosted Decision tree was effectively producing a mini-forest of 5 trees of varying size.

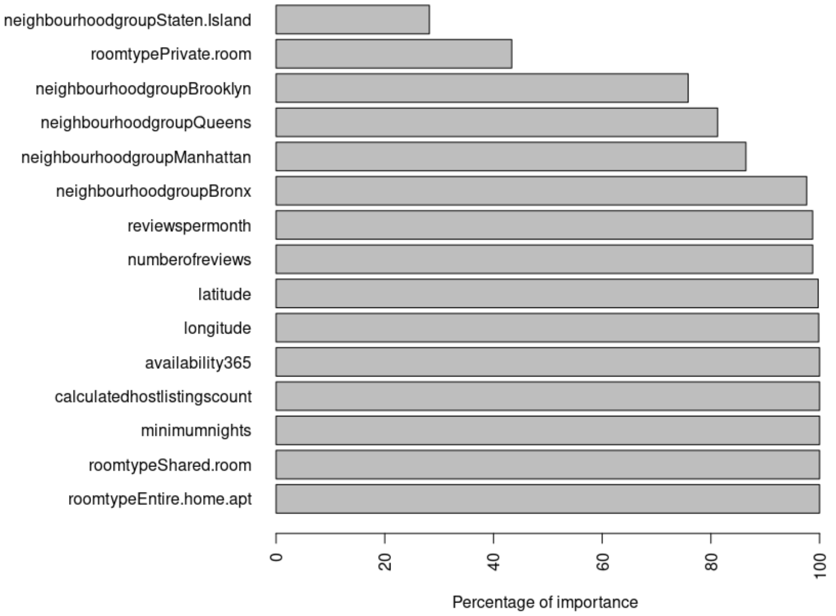


Fig 5.3.1: Strengths of the variables in D tree Boosted model

ROC for the decision tree boosted is shown below:

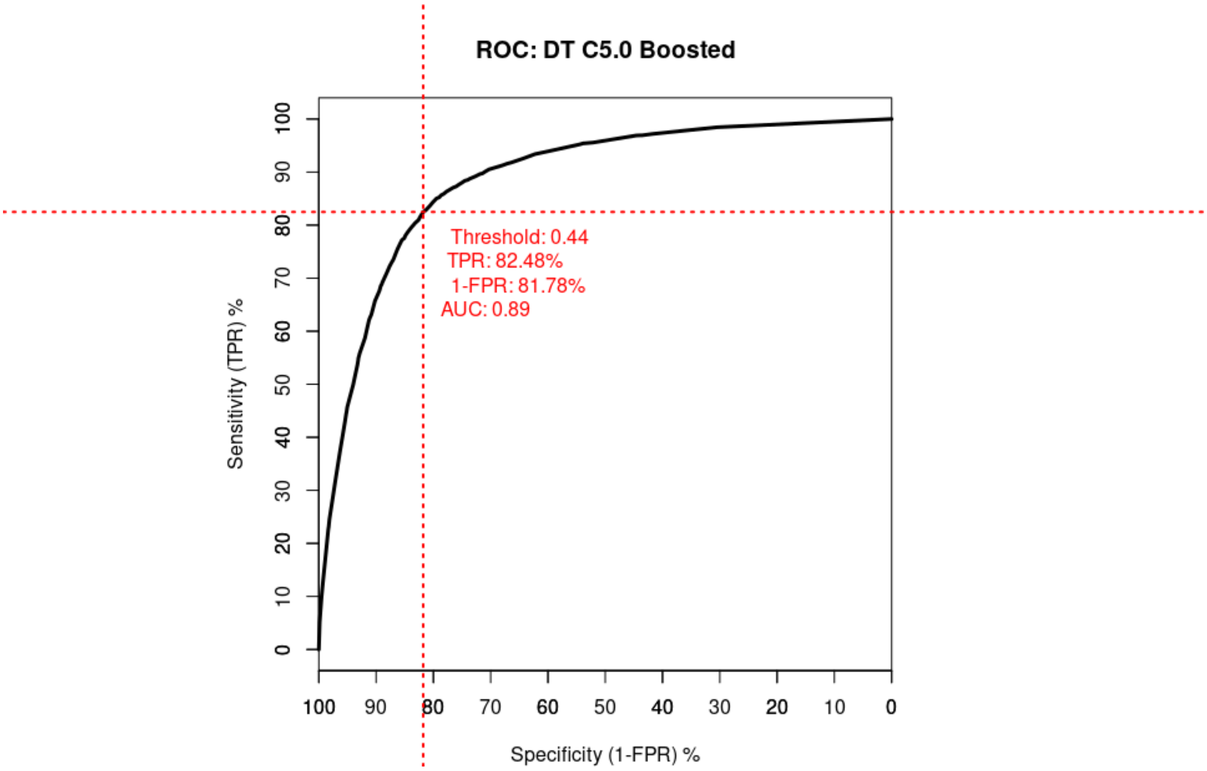


Fig: 5.3.2: ROC of the decision tree C5.0 boosted = 20

5.4 Random Forest

A random forest is constructed from decision tree algorithms, it is a supervised machine learning algorithm. This algorithm is used to predict behavior and outcomes in various industries such as banking and e-commerce. In random forest, a number of decision trees are created during the training time. The random forest model is being used to classify the data set.

The number of trees was 500 and the number of candidate variables at each split was the square root of the number of variables, which was 15, giving an mtry=$\sqrt{15}$.

Measure	Value
Precision	0.7649
Sensitivity	0.8242
Specificity	0.8214
F-score	0.7934
Accuracy	82.26%
AUC of ROC chart	88.45%
MCC	0.6396
TPR	82.42%
FPR	17.86%
TNR	82.13%
FNR	16.31%
Threshold	0.48
pgood	76.48%
pbad	86.89%

Table 5.4.1 Average Measures

The confusion matrix for the decision tree boosted is given below:

n = 13774		**Actual**	
		Class 0	Class 1
Predicted	Class 0	6636	1443
	Class 1	1001	4694

The strengths of the input fields are given below:

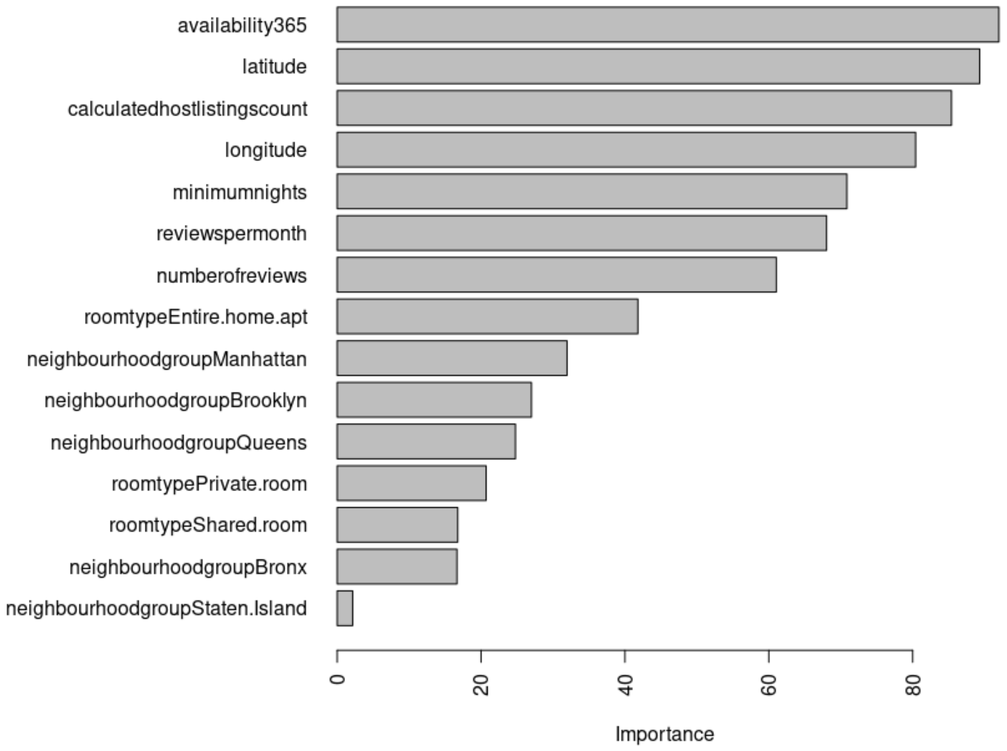


Fig 5.4.1: Strengths of the variables in Random forest model

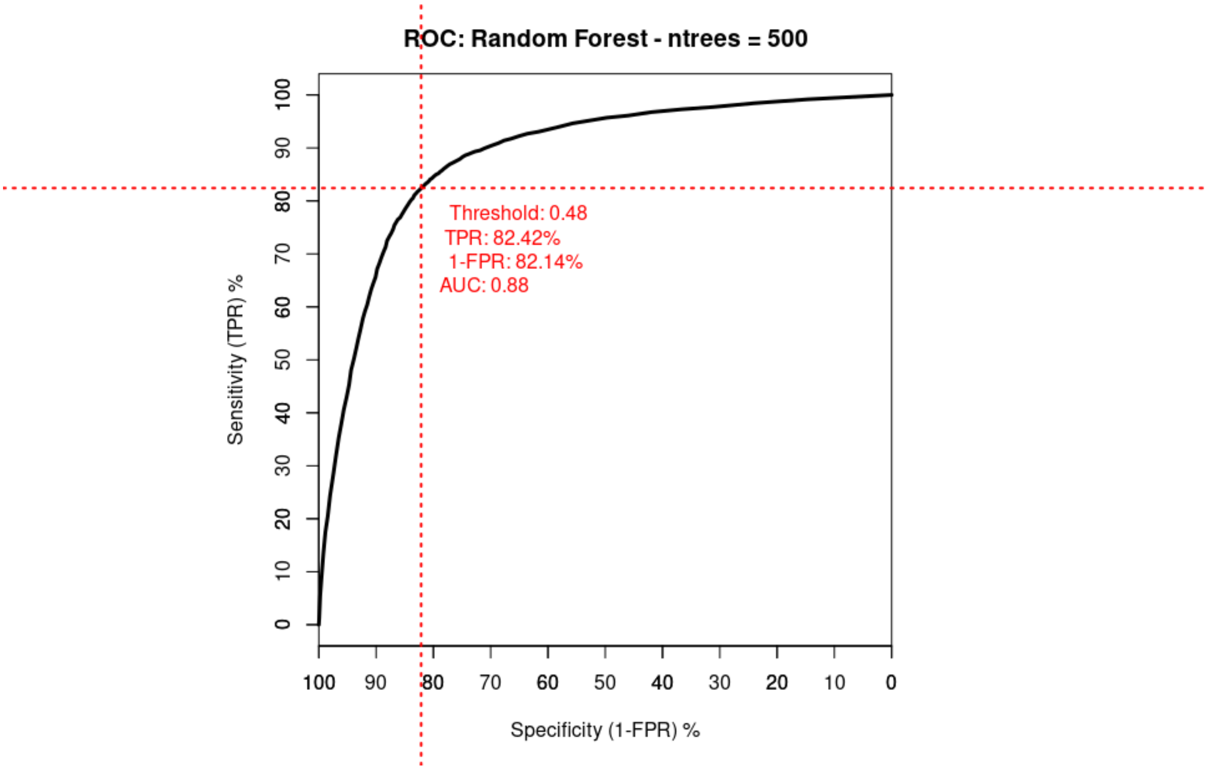


Fig 5.4.2: ROC of Random Forest

5.5 Neural Network

Neural Network are at the heart of deep learning algorithms. A deep learning neural network model for classification was generated using the standard H2O R library with two hidden layers of 20 neurons.

Parameter	Value
Activation function	Rectifier
Epoch	30
Hidden layers	2
Neurons in each layer	20

The averaged measures are shown in the table below:

Measure	Value
Precision	0.7377
Sensitivity	0.8389
Specificity	0.7871
F-score	0.7851
Accuracy	80.87%
AUC of ROC chart	87.76%
MCC	0.6181
TPR	83.89%
FPR	21.29%
TNR	78.71%
FNR	14.16%
Threshold	0.41
pgood	73.77%
pbad	87.25%

Table 5.5.1 Averaged measures

The confusion matrix for the neural network is given below

n = 13774		**Actual**	
		Class 0	Class 1
Predicted	Class 0	4813	1711
	Class 1	924	6326

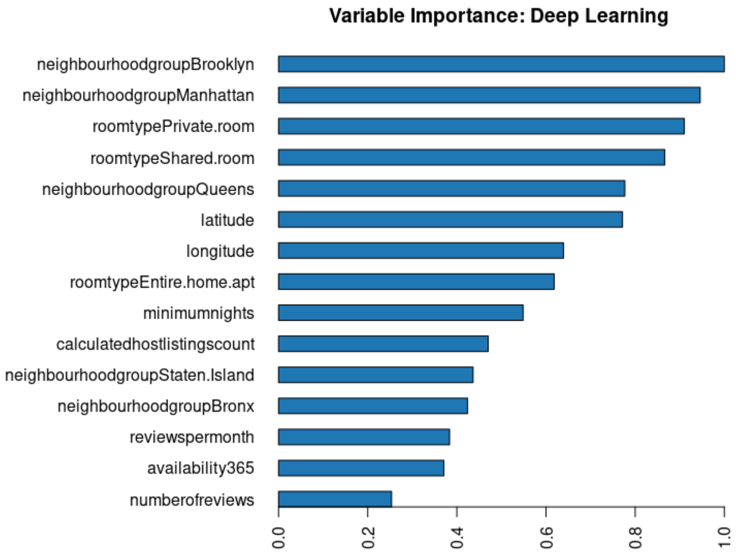


Fig 5.5.1 strengths of the input field

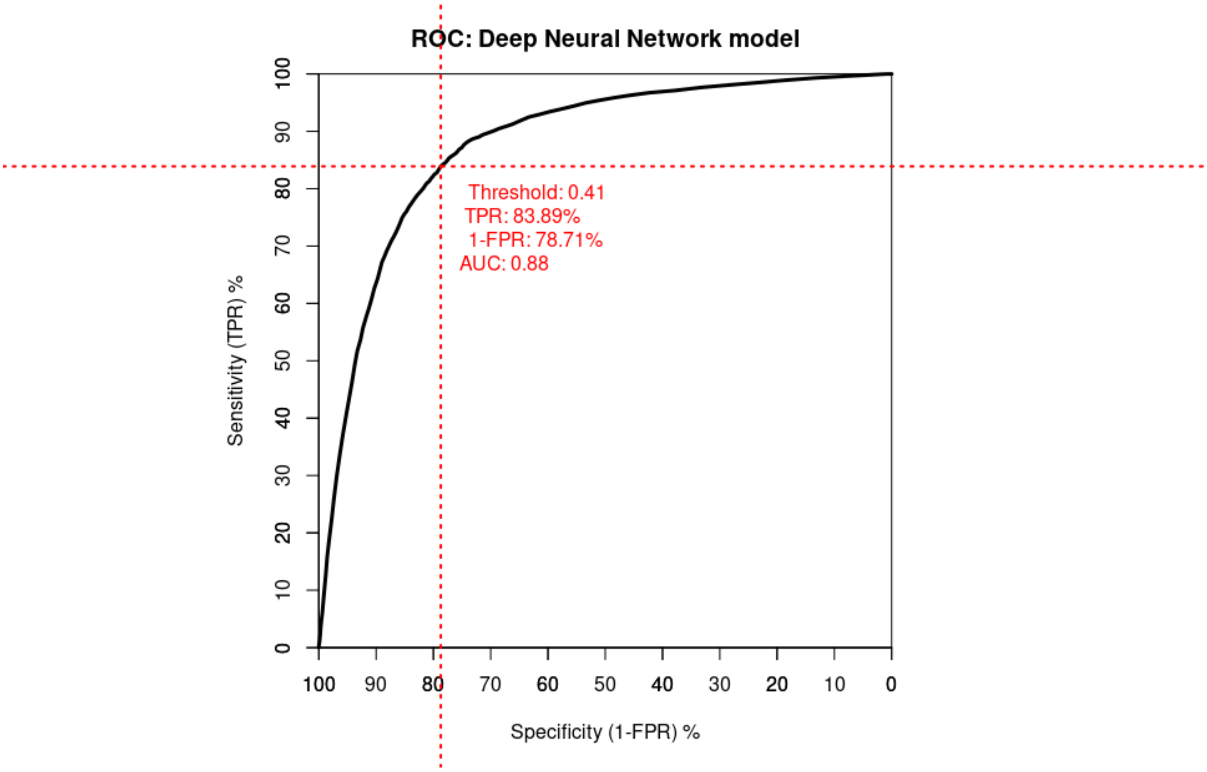


Fig 5.5.2 ROC for neural network

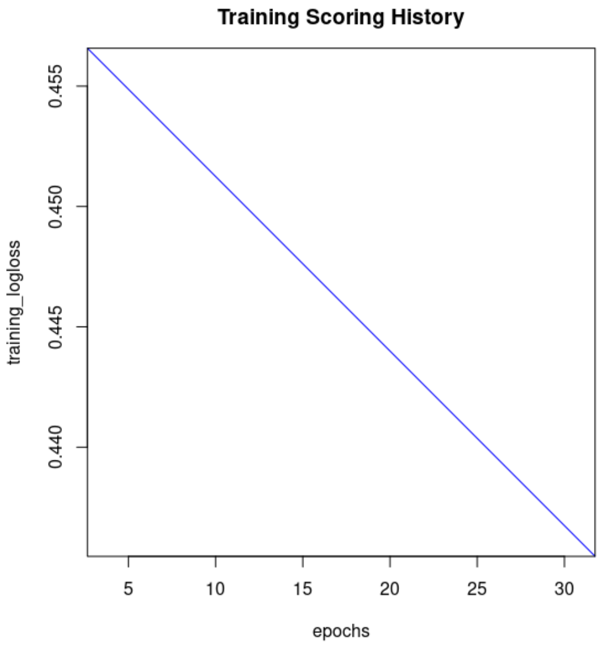


Fig 5.5.3 Epoch vs Logloss

The choosing of number of epochs is significant as the figure goes higher, there is a possibility of over fitting. To reduce over fitting and to increase generalization of the model, an optimal value for epochs is used. It is observed that when the value is above 30, the model performance starts to depreciate.

6. Model Assessment and Summary of Results

The following table was produced summarising the averaged results across the five models used to predict case outcome of the data:

	TPR	FNR	TNR	FPR	Accuracy	Precision	AUC	MCC	threshold
Random Forest	82.42	16.31	82.13	17.86	82.26	0.7649	88.45	0.6396	0.548
DT_boosted	82.47	16.177	81.77	18.22	82.07	0.7614	88.65	0.6363	0.44
Neural network	83.89	14.16	78.71	21.29	80.87	0.7377	87.76	0.6181	0.41
Decision tree C5.0	80.93	18.00	82.39	17.60	81.79	0.7642	86.06	0.6286	0.4
Logistic regression	81.58	16.57	79.19	20.80	80.18	0.7343	85.39	0.6005	0.57

Table 6.1 Average results across five models

From initial evaluation, the accuracy for random forest and DT boosted were higher and comparable whereas the other models had slightly lower accuracy.

For precision, the metrics for Random Forest and Decision trees were in the same range. Since the data set is unbalanced and a false classification cost more than a true classification, precision and accuracy are not sufficient to compare the models.

Thus, Matthew’s correlation coefficient (MCC) is used and more reliable since it only produces high score if all the four confusion matrix categories show good results. Random forest and boosted decision tree have the highest MCC (~0.64) which is approximately 7% higher than the worst model. A similar result is observed while comparing the area under the Receiver Operator characteristics curve.

It is also noteworthy that only for logistic and random forest model, the optimum threshold was above 0.5.

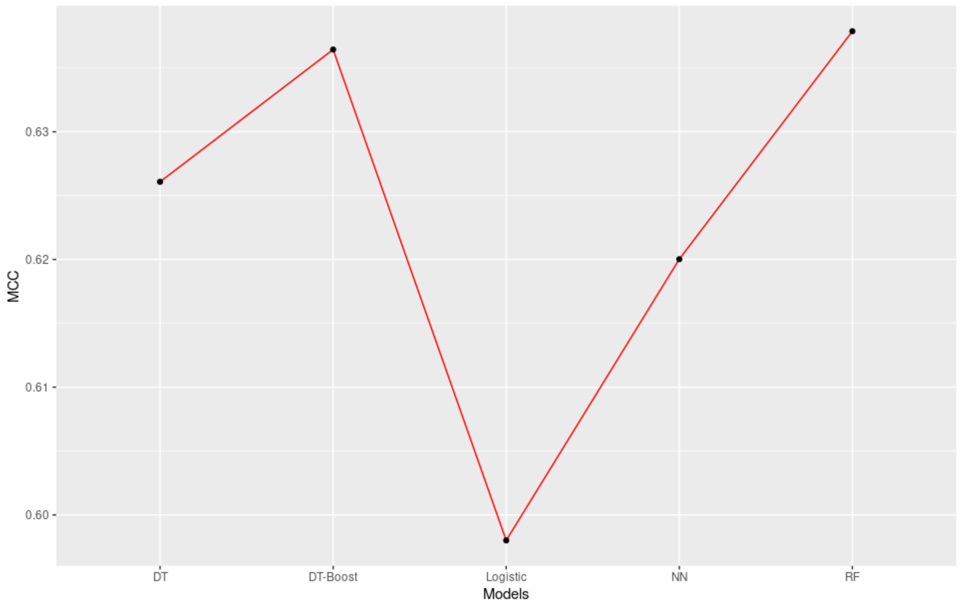


Fig 6.1 MCC values for each model

7. Conclusions and Suggestions for further research:

In this project, the group has built different machine learning models to classify the property price listed in Airbnb. The models created showed promising results and can be further improved by incorporating the following ideas.

- Since random forest and decision tree gave better results than neural network, it will be interesting to observe how will a support vector machine model will perform.
- Another possible method of improvement can be done by using a regression model to predict the price and compare the predicted value with a the actual price. Comparing the prices predicted and actual, the data points can be classified into overpriced and reasonable.

REFERENCES

[1] https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/deep-learning.html

[2] https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/

[3] https://www.analyticsvidhya.com/blog/2015/12/improve-machine-learning-results/

[4] https://surreylearn.surrey.ac.uk/d2l/le/lessons/227155/units/2249140

APPENDIX

The previous expectation according to the plan was to predict the change in price if one or more parameters were increased or decreased. The initial approach was by checking if the features had any linear relation to price. After multiple trials using different linear models such as Linear Regression and Multiple Linear Regression, it was found that there was no linear relationship between price and other features. This led to the conclusion that the linear models will not be able to achieve the previous expectations. The objective was later changed to classification of the price across various parameters rather than a prediction of price.