

## Assignment A1: Classification

Student Name	Abhishek Vinod Rajput				Student No	219266543	
Problem attempted	Complex Model 80-100%	Simple Model 40-79%			Student Id	avra@deakin.edu.au	
Place "Yes" in one only	YES						
Partial Submission	Exceptional	Very Good	Good	Acceptable	Improve	Unaccept.	
Exec Problem							
Data Exploration							
Final Submission	Exceptional	Very Good	Good	Acceptable	Improve	Unaccept.	
Exec Solution							
Data Preparation							
Model Development							
Model Evaluation							
Brief Comments						Total	
						0 to 100	

## Executive Problem Statement

Airbnb guests have a wide variety of preferences, but there are few aspects that a guest pays close attention to and an understanding of these preferences is what allows a company to retain an advantage over its rivals as well as to remain economically viable, i.e. rentals will produce enough revenue to allow both the property owner and Airbnb as a business to continue their services. Getting such an economically viable listing and being able to predict exactly whether a listing will be economically viable will help Airbnb maintain the required Rental Quality and hence the definition of rental attractiveness and market understanding is really relevant.

In the point of view of customers, a neighbourhood is considered attractive if the rental is closer to its place of interest while at the same time offering several good choices where a person can choose to stay in an entire house, a private room or live in a shared accommodation while at the same time the most rentals are getting a good number of reviews every month. AirbnbAI would like to know the most attractive neighbourhood and the reasons for the same.

Understanding the market deals with understanding what kind of property is booked more often thereby helping us in understanding the target audience i.e. are the tenants looking for shared accommodation, a private room or an entire house for their use and also at the same time understand if the owners with more than one property make better arrangements and can get more number of reviews. This will also help us in understanding how much money is the tenant ready to spend on a particular type of home and if there is a relationship between the type of room, the cost associated with it and its popularity.

The Neighbourhood Attractiveness Review shows the most attractive neighbourhood to be either in Manhattan or Brooklyn. A potential reason for this is due to the enormous demand in the area that occurs mainly due to Office and bookings in the region and bookings made by customers on business trips. Customers often tend to live in an entire house or a private room and seldom want to live in shared rooms.

The Market Study found that consumers tend to book costly rentals that are often either whole house or private rooms, and hosts with multiple listings offer cheaper rental choices, and can therefore earn a large number of reviews.

Airbnb needs to clarify the following issues based on the above evaluations

- Price impact on Reviews
- Is there a relationship between the rental name and its economic viability.
- Is there a neighbourhood group consistent in receiving higher ratings, likewise a community group that earns consistently poor ratings and potential solutions to the problems.

The Following is the procedure involved in answering the above problems.

## Data exploration

The data with 49000 examples provided by AirbnbAI was used for this analysis

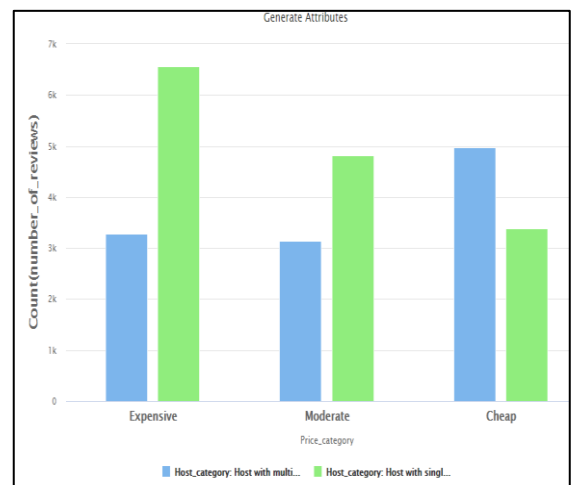
The data consists of 16 unique attributes and a Rapidminer model was prepared using the following a

Attribute Name	Description	Missing Value	Characteristics of the Attribute	Action taken
<b>reviews_per_month</b>	The average number of reviews per month.	10052	Real type. Values ranging from 0.010 to 58.5. Left skewed with most data in the range of 0-5	All the missing values are filtered. And Since the data is skewed median is taken to calculate the true average value.
<b>number_of_reviews</b>	The total number of reviews the rental has received in its lifetime.	0	Integer type. Values range from 0 to 629. Left skewed with most data in the range of 0-50.	Only those rentals are accepted for the analysis whose number of reviews are more than 0
<b>neighbourhood</b>	The Area in which the rental is	0	Polynomial type. 221 unique Neighbourhood.	None
<b>Price</b>	The price for each rental in dollars	0	Integer Type. Values range from 0 to 10000. Heavily left-skewed with most data lying between 0 and 1000.	Only rentals are chosen with a price between 1 and 1000. The prices are divided into two categories Cheap, Expensive and Moderate depending on the neighbourhood group.
<b>Room type</b>	The type of room listed for renting	0	Polynomial Type. 3 values. With an entire home being the most popular and shared home being the least popular	None
<b>calculated_host_listings_count</b>	The number of listings a host has.	0	Integer type. Values range from 1 to 327. With most people having between 1 -5 listings.	Host category is created by making use of this attribute and split into hosts with single listing and hosts with multiple listing
<b>name</b>	The Name of the listing	16	Polynomial type. Each name being unique.	None
<b>last_review</b>	The date when the rental received its last review	10052	Date_type. Containing dates between 28/03/2011 and latest date being 9/07/2019	Only dates after 2015 are considered. Since all previous dates skew the data towards low reviews per month and it provides relevance to the model.
<b>minimum_nights</b>	The minimum nights a rental need to be booked.	0	Integer type. Values range from 1 to 1250. With most properties in the range of 1-5 nights.	None
<b>Longitude and latitude</b>	Two separate attributes to provide the geolocation.	0	Real type.	None
<b>Availability_365</b>	The number of days the property is available to book	0	Integer. Ranging from 0 to 365 with dates even spread.	None
<b>Host_id</b>	Unique Id by the host is identified	0	Integer. Random Values assigned.	None

Table 1. Attributes Exploration



**Figure 2.1 Attractive Neighbourhoods by reviews\_per\_month**



**Figure 2.2. Price Category by host listing count**

The figure shows the most attractive neighbourhood and figure shows the relationship between price and number of reviews for hosts with a single listing and host with multiple listings.

## Executive Solution Statement

The economic viability is predicted through the creation of Rapidminer models for all those rentals which have no reviews.

As shown in figure 6.4 most New York properties are economically viable and a small part of it is not viable, which is a positive sign for Airbnb.

The model provides ample evidence of the relation between the property's name and likeliness to make it economically viable (Figure 4.3). Using an appealing name is really important for a rental so that more people are likely to book the rental. Airbnb should further explore what are the most widely used keywords and the keywords that draw customers most likely.

The model also showed a rental's ability to draw customers is virtually independent of the type of room and the price associated with it, suggesting the assumption derived from the figure 2.2 is incorrect (Figure 4.1).

The Ratings received per month, although they have a much lower relationship with the neighbourhood group, Manhattan and Brooklyn are able to sustain many of their properties. A potential reason for such a case is due to the existence, as seen from figure 2.1, of several attractive neighbourhoods within them. Bronx performs the worst of all neighbourhood groups, this can be explained by low traffic or maybe the properties need more appealing names in order to draw new customers.

The findings obtained will enable Airbnb to maintain its competitive edge and market sustainability.

## Data Preparation

The model aims at finding the economically viable rentals and uses the number of reviews a rental is able to get as the factor to check its viability, therefore a new attribute “**Viability**” is created by discretising the attribute “**reviews\_per\_month**” into “Viable” and “Non-Viable”. All those examples having “**reviews\_per\_month**” more than 1 are Discretised using *Discretise* operator as economically viable and all others are discretised as non-viable. This newly created attribute is used as the label for the model.

A *Correlation Matrix* was used to understand the relationship between the attributes. The result of the correlation matrix is as shown.

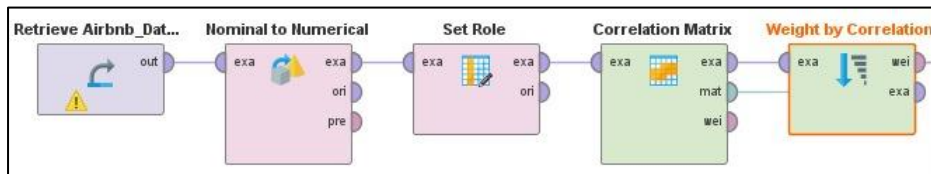


Figure 4.1 Correlation Matrix and Weight Analysis

The matrix shows “**reviews\_per\_month**” to have positive Correlation with most of the attributes, “**number\_of\_reviews**” to be most strongly positively correlated while “**price**”, “**room\_type**” and “**minimum\_nights**” show a negative correlation.

*Weight by Correlation* operator is used to understand the relevance of different attributes with the label. The operator accepts only numerical data therefore the attributes need to be converted to nominal values using *Nominal to Numerical* operator and then fed into the operator. The figure shows the weights of different attributes with respect to **reviews\_per\_month** and top 9 attributes with highest attributes are used as predictors in the model since the lower attributes have very low weights to cause any changes in the result.

Attributes	reviews_per_month ↓
reviews_per_month	1
number_of_reviews	0.505
host_id	0.251
name	0.248
availability_365	0.216
host_name	0.127
longitude	0.119
neighbourhood	0.093
neighbourhood_group	0.081
calculated_host_listings_count	0.011
price	-0.017
room_type	-0.019
latitude	-0.021
minimum_nights	-0.131

Figure 4.2 Correlation with labels

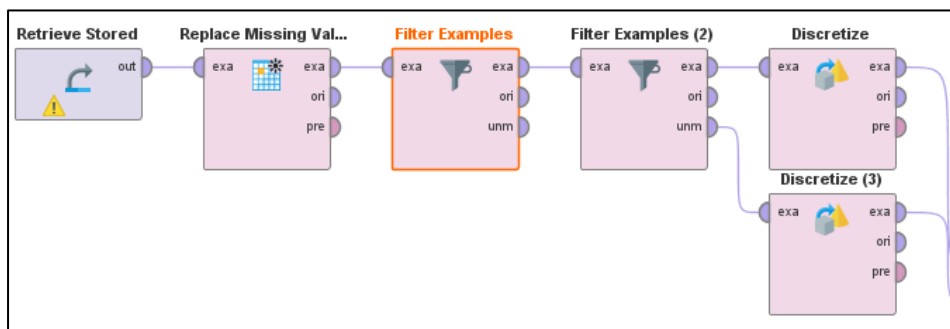


Figure 4.4 Data Preparation Process

Missing **names** are replaced by “No names” and extreme prices above \$2000 are filtered and examples having **hostnames** missing are filtered.

attribute	weight
name	0.966
last_review	0.328
host_name	0.323
number_of_reviews	0.233
availability_365	0.132
minimum_nights	0.059
id	0.042
host_id	0.038
neighbourhood	0.036
longitude	0.013
calculated_host_listings_count	0.011
neighbourhood_group	0.009
latitude	0.002
room_type	0.001

Figure 4.3 Weight of Attributes

## Model Development

The data after preparation is retrieved and the data with reviews is used to train and test the model. *Set Role* operator is used to assign “**reviews\_per\_month**” as the label and the weights of each attribute are calculated using the *Weight by Information Gain* operator and the top 9 attributes are selected as predictors. And data is then sent to the *Cross Validation* operator wherein either KNN or Gradient Boosted Tree classification is applied to the model.

### Prediction Using KNN Classification

The KNN model uses Euclidean distance in order to calculate the distance between query sample and its nearest K samples and therefore the data needs to be normalised, which is done using the *Normalize* operator and Z transformation is used for the process since this transformation preserves the distribution of the data. The Training data as seen in figure 5.1 is imbalanced since Viable class is more compared to Non-Viable class and hence in order to balance the two classes *Smote Upsampling* operator is used and the training data after upsampling is balanced as seen in Figure 5.2.

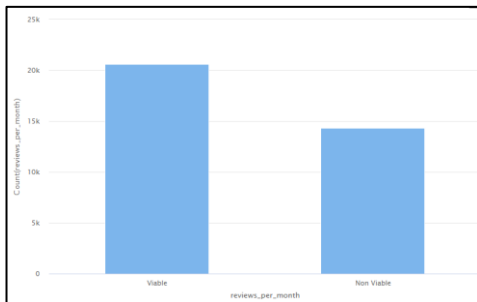


Figure 5.1 Class Imbalance in Training Data

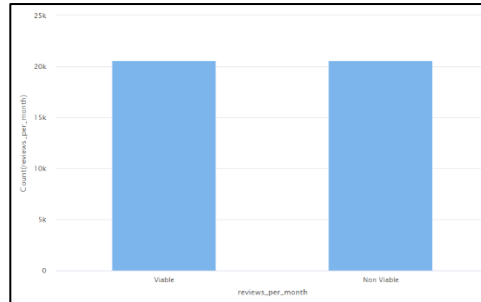


Figure 5.2 Balanced Class in Training Data

iteration	k-NN.k	acc... ↓
3	21	0.840
4	31	0.840
5	41	0.839
2	11	0.838
6	51	0.837
7	60	0.836
8	70	0.836

Figure 5.3 Optimise Parameter

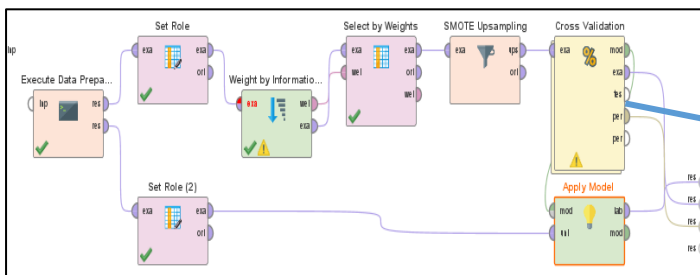
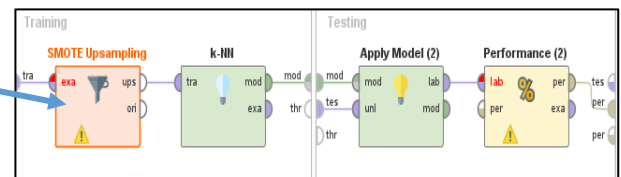


Figure 5.4 KNN Classification Model



The examples with the selected attributes are then passed through the *Optimise Parameter* operator to find the best possible parameters for the model. When the Maximum value of K is at 100 optimisation results are shown in figure 5.3 and the parameters for the optimum results are obtained.

Examples are fed in the model and the value of K is set as 21 since it gives the best possible accuracy and Kappa and their performance are calculated using cross-validation.

### Prediction Using Gradient Boosted Classification

Gradient Boosting Relies on previous results to get the best possible results for the next model in order to minimise the error by progressively improving the results. The values of number of trees and depth need to be decided since in order to understand the sweet spot where enough information is captured to give accurate results at the same time reduce processing time. An *Optimise Parameter* Operator is used to get the best possible value of depth with parameters set as maximum depth =100 and the maximum number of trees = 20. As shown in figure 5.5, the optimum number of trees is 18 and optimum depth is 31 to 95.8% accurate results.

iteration	Gradien...	Gradien...	acc... ↓
43	18	31	0.958
21	18	11	0.957
31	16	21	0.957
44	20	31	0.957
19	14	11	0.957
62	12	51	0.957
16	9	11	0.957

Figure 5.5 Optimise Parameter

The examples are fed in the model and the performance for the model is calculated using Cross-Validation.

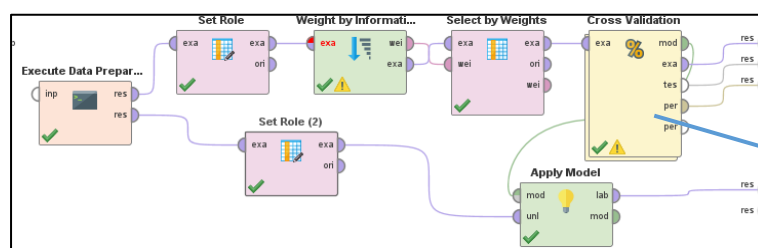
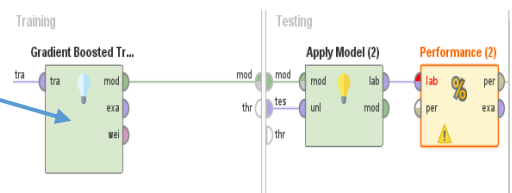


Figure 5.6 Gradient Boosted Classification Model



## Model Evaluation

The performance of both models is calculated using cross-validation.

The below Figure shows the confusion matrix for the two models with viable as true positive

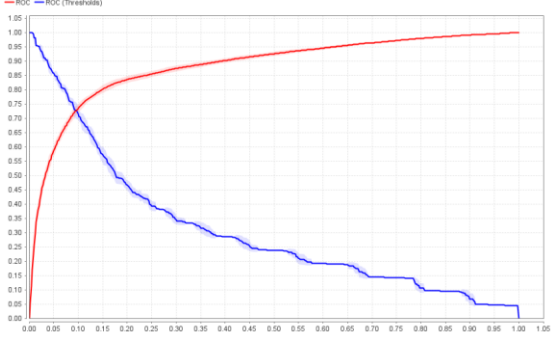
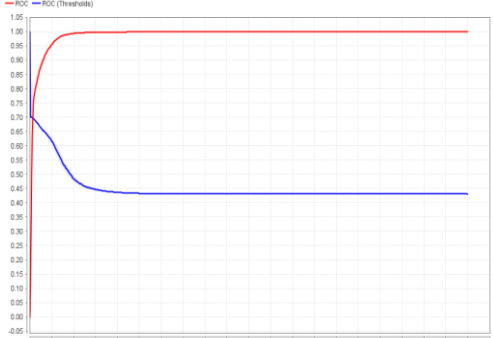
	true Viable	true Non Viable	class precision
pred. Viable	18872	2835	86.94%
pred. Non Viable	4008	13062	76.52%
class recall	82.48%	82.17%	

Figure 6.1 Confusion Matrix KNN Classification Model

	true Viable	true Non Viable	class precision
pred. Viable	22185	927	95.99%
pred. Non Viable	695	14970	95.56%
class recall	96.96%	94.17%	

Figure 6.2 Confusion Matrix Gradient Boosted Classification Model

The Above confusion matrices are used to calculate the key performance metric.

	KNN Classification Model	Gradient Boosted Classification Model
Accuracy	82.35% +/- 0.83%	95.82% +/- 0.26%
Kappa	0.639 +/- 0.017	0.913 +/- 0.005
Sensitivity	82.48% +/- 1.04%	94.17% +/- 0.65%
Classification_error	17.65% +/- 0.83%	4.18% +/- 0.26%
AUC	0.883 +/- 0.007	0.988 +/- 0.001
Specificity	82.17% +/- 1.19%	96.96% +/- 0.70%
AUC-ROC Curve		

In order to achieve better results, Airbnb should aim at minimising False Negatives and therefore a better model is the one having better Kappa i.e. better reliability and is able to correctly identify the viability of the property. In comparison to the KNN Classification model, the Gradient Boosted Classification model is a better choice in terms of performance matrices like Accuracy, Kappa, Sensitivity, Classification\_error, AUC and Specificity.

Using all the above criteria Gradient Boosted Classification Model will be more efficient in identifying the viability of the property. The results are 95.82% accurate when all the predictors are present but inconsistent when any of the predictors is absent.

The final Results using Gradient Boosted Classification are plotted

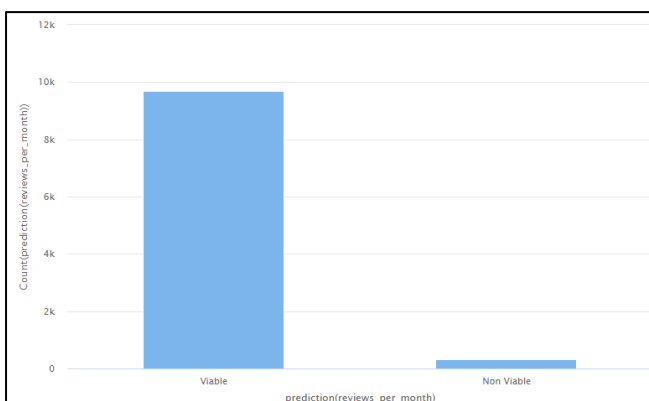


Figure 6.3 Viable properties vs Non-Viable Properties

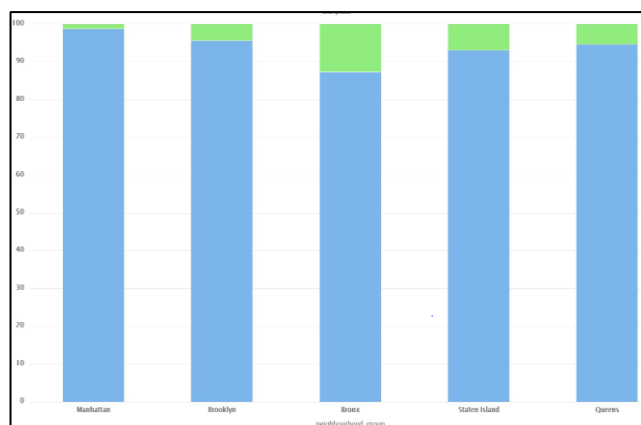


Figure 6.4 Spread of Viability by Neighbourhoodgroup