QUALITATIVE ANALYSIS OF NEW YORK YELLOW TAXI: AN ATTEMPT IN FORMULATING AND CLASSIFYING RATINGS FOR TAXI TRIPS

Nathanael Luira Yoewono
Bachelor of Science
The University of Melbourne
Student ID: 1000582

November 1, 2020

ABSTRACT

This report will demonstrate an attempt in identifying clusters in New York yellow taxi trips. These clusters are used to rate trips based on five different criteria: fare per miles, fare per minute, tip amount, miscellaneous cost, and trip frequency. These ratings may be important for both passengers and the taxi driver. One the one hand, this rating may be more intuitive to inform the passenger whether the trip may be costly or not. On the other hand, the taxi driver might know which pickup and drop off location are popular. The result shows four distinct ratings that may separate the taxi trip quality. Moreover, a random forest classifier is used to classify these ratings using a real-world scenario attributes. The model managed to get 72% classification accuracy.

1 Introduction

To begin with, there may exist several metrics to measure the quality of taxi trips. For instance, the rate of trip frequency, the tip amount, the estimated cost, and many more. However, it may not be effective to judge each taxi trip using these metrics separately. Thus, clustering the trips based on the combination of these metrics and turn it into ratings may be more intuitive for both passengers and the taxi driver. These clusters may be significant, as it may help passengers to judge whether the trip is costly or not. Moreover, this may also prevent disputes over the fare amount between passenger and taxi driver [4]. These ratings may also assist passenger and taxi driver in deciding the fair tip amount based on the median tips in each rating. Ultimately, the aim of this analysis is to find a method to cluster the trip and to test whether these ratings can be predicted using real-world scenario attributes.

The analysis will use the TLC New York yellow taxi data [1] with additional external data sets, which is gas fuel price from U.S. Energy Information Administration website [2], and weather from wunderground.com [3]. The period will be limited to 2019, which is from January to December. The reason for these limitations are first, 2020 was affected by the covid-19 pandemic. Second, 2018 data below may have lurking confounding factors that may affect the taxi trip rating metrics. Henceforth, the cluster evaluation will be limited to one year prior to 2020.

In addition, this report will be divided into two parts, that is unsupervised and supervised modeling. The unsupervised is used to identify and learn the cluster group labels for the taxi trips. The attributes of interest here are the fare amount over distance, the fare per minute, the miscellaneous amount percentage, the tip amount percentage, and the trip frequency for each combination of pickup and drop off location trips. All of these attributes were extracted from the original taxi using the feature engineering process, and this will be explained further in the preprocessing step. The fare amount here is turned into ratios to adjust the fare based on the trip distance and time. The government also specifies a fare calculation that is based on distance and time [5]. Moreover, the fare over distance is reduced by the gas fuel per miles cost in order to adjust the fare amount price. The miscellaneous amount is the combination of both extra and toll charges, that is the additional spending a passenger needs to pay aside from the taxi fare. Taking the percentage amount of both miscellaneous and tip over the total amount will help indicate the impact of these two attributes on the total cost for the passenger, as there may be several trips where it may be costly due to the payment of the toll. The trip frequency may indicate which destination trip is popular, and which is rare. All of these five attributes will be used to identify the ratings for the taxi trip using unsupervised methods.

Moreover, for the supervised section, certain models will be tested to make a classification prediction on the ratings based on real-world scenario attributes. These attributes are the day and hour of the trip, rate code id, the pickup and drop off location, estimated trip distance and time, and other external factors, such as temperature and precipitation. They are chosen as they are known before the trip happens. However, the attributes that are used to make the ratings will not be included, as these attributes may only be available after the trip is taken by the passenger.

2 Preprocessing

2.1 TLC Data

The preprocessing was divided into two main parts, one is the main preprocessing, where each data from each month in 2019 will go through a series of guards based on some specific values in certain attributes. This is done to exclude anomalies in the TLC yellow taxi data. Moreover, feature engineering is also included in this section. The second part is aggregating all of the cleaned data into one large data set.

To start with, most of the preprocessing time was done in the main section. There are several guards that are implemented to exclude data anomalies. Firstly, the passenger attribute indicates the number of passengers that are in the taxi. However, some of the passenger values are 0; hence, the data frame will be sliced based on values that are bigger than 0. Second, the data scope was limited to only credit payment type, as dispute and no charge payment type has negative values in its fare amount, and this may distort the analysis. Also, cash type payment does not record any of the tip amount [6]. Third, there was some undefined rate code id, such as 99; thus, these rows were being excluded. Next, all trip distance should be greater than 0, and the minimum fare is $2.5 [5].

Additionally, some predefined rules were listed on TLC website regarding fare rates, especially for JFK and Newark, rate code id 2, and 3. For JFK, the rate is fixed, with $52 as its cost [5], whereas for Newark, the rate was set to a minimum of $17.5 for each ride [5]. Ergo, an additional slicing guard was implemented to ensure consistency in the data quality. Next, the negative tip amount will be excluded, but not zero, as it is still possible that some passengers did not give any tips to the taxi driver. Lastly, there are some errors spotted in the dates of the taxi trip; thus, data with false data, those that do not belong to the restricted period, will be excluded from the data set.

In addition, after this selective process, the data set will go through a feature engineering stage. This stage will extract the time taken for the trip (in an hour), speed (miles/hour), and the hour when the passenger was picked up by the taxi driver. Time taken is constructed by taking the differences between the pickup date time minus drop off date time. Speed is the trip distance (miles) over the time taken for the trip (hour). While the hour and time is used as predictors in the analysis part, speed is only used to reduce data anomalies further, as there are some errors in the recorded date time pickup and drop off data, resulting in negative time traveled for the taxi trips. Not only that, but some instances also have a large trip distance, which results in a large ratio between trip distance and time taken. Even though the speed here does not mean the exact speed of the taxi, it is assumed that this speed is an indication of the estimated speed for the taxi, and it will be limited to below 65 mph, as it is the maximum speed in New York state [7], and by inspection, there may be a steady and normal ratio from this value below. Time taken will also be restricted to 5 minute-2 hours, as values outside of this range are in the 1% of the quantile, which is very rare; thus, they will be treated as outliers.

At this stage, it is assumed that the TLC data may not contain any anomalies; hence, the next stage would be to group them by the date, hour, pickup, drop off and rate code id of the trip and take the mean to retain as much information as possible from the data. Trip frequency, another feature engineered predictors, is attained from another group by with the same keys, but summing them instead of taking the mean. The trip frequency will be added to the mean group by. Moreover, since the data has been reduced tremendously at this stage, the day of the week attribute was elicited at this point, as it requires more time. After extracting the day, the fare ratio was made, which is the fare/distance and fare/minute. The minute was taken from the time taken for the trip by multiplying it with 60 to get the total minute. After all of this was done, the last part of the main preprocessing is to adjust the fare/distance from the TLC data with the gas price per gallon, and the assumption here is that a gallon of gas could accommodate 18 miles for each taxi, which is 0.055 gallon per miles [8]. By multiplying the 0.055 to the gas price and subtracting the fare/miles with this value, the fare/distance became a clean fare/distance. The resulted data set will be saved in feather form. This main preprocessing flow will be run for each month in 2019; thus, there will be 12 feather files containing clean yellow taxi trips from January to December 2019.

The final stage of the preprocessing is the concatenation of these 12 files. The preprocessing manage to reduce the data size from around 7.5 GB to 2.16 GB, with around 17 million taxi trips. However, this is only the early part of the preprocessing, as further minor preprocessing will be done in the analysis part, that is after clusters are found in the taxi trips.

2.2 Gas data

Obtaining the gas data was not a challenge, as the data can be downloaded. This data contains a weekly gas price, starting from around 1993 to 2020. Since the period is limited to 2019, the date of the gas is sliced to match the stated period. Moreover, since it is a weekly price, a backfill method was done to fill all of the nan rows when combining this gas data with the TLC, as the TLC data is a daily data. After the fare adjustment was done as in the previous subsection, the gas price was dropped, as it is no longer needed.

2.3 Weather

Getting the weather data was not straight forward, as there is no option to download the data in csv format; thus, web scraping was done using selenium [10] to get the hourly data. The attribute of interest here is the precipitation and temperature. Minor preprocessing was done to change the precipitation and temperature data format to float instead of a string. This data will be combined with the original data in the last part of this analysis, as it is part of the real-world scenario analysis.

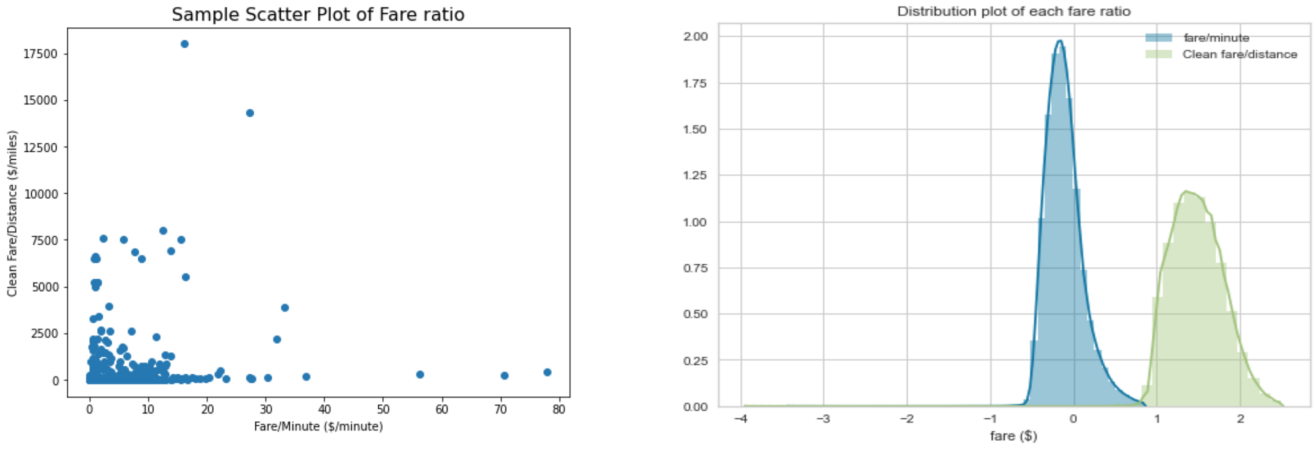
3 Identifying Clusters

The first step in identifying these taxi trip clusters is to find which attribute may best distinguish these trips. Thus, it is decided that fare amount/distance, fare amount/minute, tip amount %, miscellaneous amount % and trip frequency may be a decent indicator for clustering this taxi trips.

3.1 Summary Statistics

3.1.1 Fare/distance and fare/minute

The fare amount of taxi trips are being adjusted based on their distance and minute of the trip in order for them to be comparable. Further adjustment on the gasoline price was done to better judge whether the fare is actually expensive or not.



(a) Outliers for fare/distance vs fare/minute

(b) Log normalised distribution plot

Figure 1: General plots for fare/distance and fare/minute

By looking at figure 1a, it can be seen that there may be outliers in both of these fare ratios. For instance, there is a fare/distance of $17,500 per mile, which may not be accurate. This plot was generated over 1,000 sample data that was taken from the original data; hence, it can be expected that more outliers exist. Therefore, an outliers detection using the whiskers method was done to exclude these points, as it may distort the clustering process. In addition, after excluding outliers, a distribution plot was generated over all of the data, which can be seen in figure 1b. Both ratios here were log normalized to generate a better bell curve shape. The plot may indicate that the fare/minute may have a smaller standard deviation compared to the fare/distance, which means the fare/mintue ratio is more stable. Their fare mean was around $0.9 per minute and $4.6 per mile, both without the log normalized.

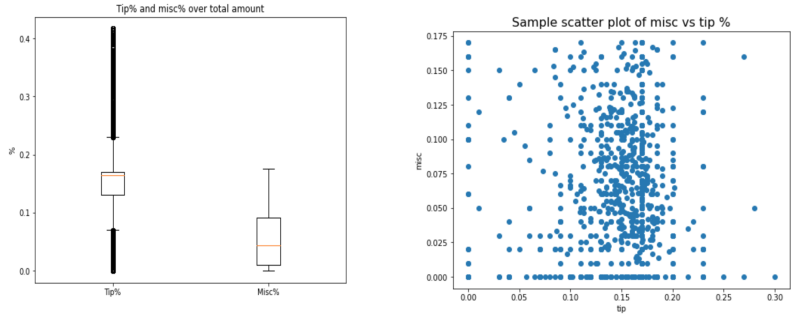
3.1.2 Miscellaneous and Tip amount

The miscellaneous amount attribute consists of the amount of the toll combined with the extra charges that are available in the original taxi data. The tip amount is the tip that the passenger gave to the driver, via credit charge. Both of these attributes were turned into percentage over the total amount to see their impact on the total cost of the taxi trip.

	tip_amount	misc_amount
count	1.698075e+07	1.698075e+07
mean	1.467774e-01	6.382542e-02
std	4.981317e-02	6.001246e-02
min	0.000000e+00	0.000000e+00
25%	1.300000e-01	1.000000e-02
50%	1.651429e-01	5.000000e-02
75%	1.700000e-01	1.000000e-01
max	9.800000e-01	9.800000e-01

Figure 2: Correlation plot between selected attributes in TLC data

By just looking at the summary table above, it seems that both attributes may contain outliers too, as seen from the gap between 75% quantile and the maximum value. While it may be possible that a passenger would be willing to give a huge tip to a driver, with around 98% of its overall trip cost, it may be rare; hence, these records were considered outliers and removed from the data.



(a) Boxplot of tip and misc % over total cost

(b) Scatter plot of misc vs tip

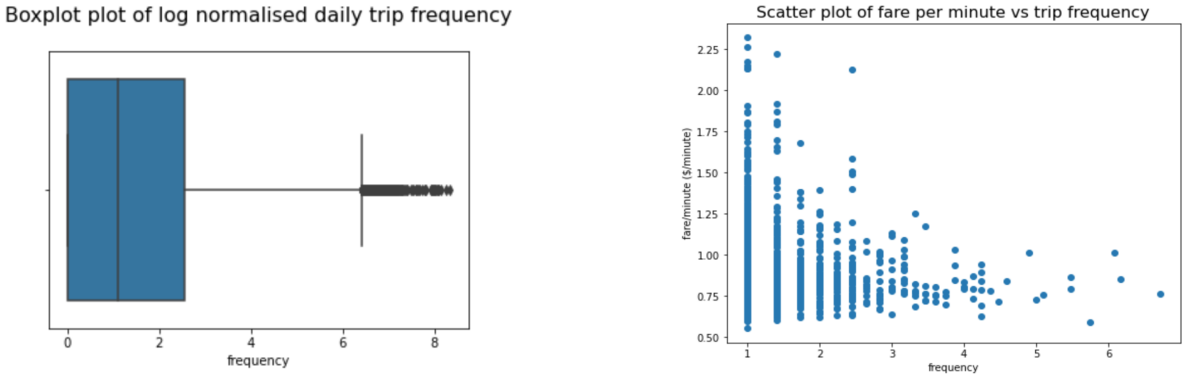
Figure 3: General plot for tip and miscellaneous amount

Figure three above displays a summary of what these attributes look like. The boxplot at 3a shows that tips may be more dispersed compared to the miscellaneous costs, which may indicate that the value range may vary. The median for both tip % and miscellaneous cost % are 0.165% and 0.05% respectively. Moreover, figure 3b indicates that this pair of attributes may not have a linear relationship. In fact, they may have a very low correlation, as seen by the sample scatter plot. Hence, it seems that the passenger may still be willing to give generous tips even though the miscellaneous cost is quite costly.

3.1.3 Trip frequency

The last attribute that will be discussed to identify these clusters is trip frequency. From This attribute may indicate which trips are more popular than others. This popularity may be affected due to the trip locations, such as Manhattan, JFK, and LaGuardia airports, or due to cheaper fare rates.

Figure 4a displays the overall distribution of the log normalized daily trip frequency within 2019 that is based on each pickup location area around New York. It seems that there may be a few favorable pickup spots, as seen from the data outside the whiskers. Also, based on the exploratory data analysis in the previous assignment, there may be evidence that each specific trip may have different frequencies. Thus, this may indicate that taxi trips may be distinguished based on the historical trip frequency, as it may reflect certain trip's popularity amongst New York yellow taxi passengers. The median of daily trip frequency is around 3, and the maximum is bigger than 2980 trips. Furthermore, other interesting findings can be seen from figure 4b, where it shows that trips with lower fare per minute tend to have a higher historical daily trip frequency. This might be one of the indicator of why certain trips are more popular.



(a) Boxplot of trip frequency

(b) Scatter plot of trip frequency vs fare

Figure 4: General plot for tip and miscellaneous amount

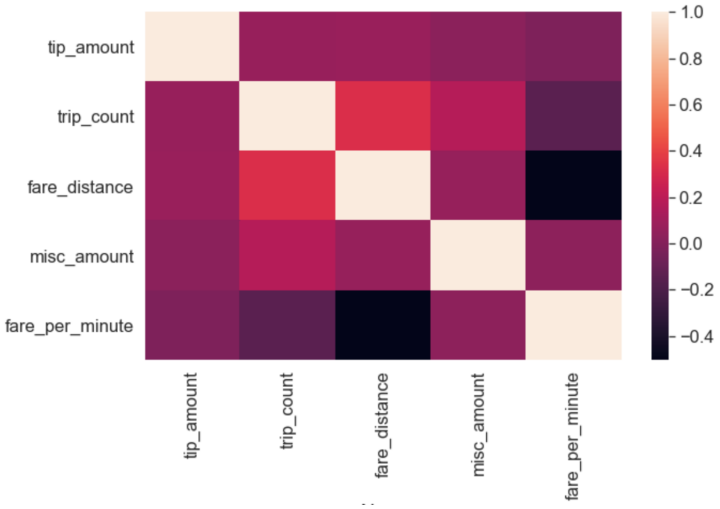


Figure 5: Correlation plot between selected attributes in TLC data

3.1.4 Correlation

Since all of the attributes here are continuous, a Pearson correlation that is visualized using a heat map plot may assist in identifying any linear relationship between these pairs of attributes. From the figure above, it can be seen that the highest correlation is achieved between fare/distance and fare/minute, with -0.5 for the correlation values approximately. Thus, an increase in fare/distance may lead to a decrease in fare/minute. This might be explained by the standard meter fare rules, but correlation does not mean causation. Moreover, the trip count may also bring values to this correlation, as it has 0.33, 0.17, and -0.15 correlations with fare/distance, miscellaneous amount, and fare/minute, respectively. It is interesting to see that trip frequency correlated positively to fare/distance, and negatively to fare/minute. One reason that may explain this behavior is due to the minimum charge of $2.5 for the standard rate regardless of the distance. Hence, a shorter trip may seem to have a lower fare, but when the fare is adjusted to its distance, it might not be considered low.

All in all, these attributes may have their own merits in distinguishing the yellow taxi trips. However, the objective of this report is to combine all of the information contained in these attributes in order to generate clusters for the New York yellow taxi trips. Thus, the apparent solution for this problem may be unsupervised clustering using equal-width binning and k-means clustering.

3.2 Clustering Methods

Unsupervised learning is a method in machine learning to identify the hidden clusters in a given data set with no labels available [11]. In other words, this method's intention is to learn the patterns of how the data may interact with each other, and find similarities using the available attributes. Thus, this learning may help identify the hidden clusters inside these taxi trips. There are two unsupervised methods that will be used to solve this problem, K-means, and Equal Width Binning.

3.2.1 K-means

The idea behind k-means is to initialize k random centroids points inside the data, and calculate the distance between each point to those centroids [12]. These k random points will be the k clusters or labels for the data. In each iteration, points should be allocated to its nearest center, and the center itself will be updated by taking the means of each value in the cluster. Next, allocate the points to its nearest updated centroids and re-update the centroids again. This process will be repeated until are no more cluster changes in the points assignment process. The distance metric that is normally used for continuous variables is the euclidean distance [13].

3.2.2 Equal width binning

Equal width binning is an unsupervised binning method. This method will bin a given data attribute into k bins with equal width [14]. It is called unsupervised as, without domain knowledge, one may not know immediately what might the best number of bins to group the data. The bin width can be easily calculated by subtracting the maximum and minimum values in the data and divide it with the number of bins specified.

3.2.3 Metrics

The metrics that are used to evaluate the cluster are the ratio between cohesion and separation of the cluster. The cohesion here can be calculated using the within Sum of Squared Errors (W_{SSE}), and the separation using the between SSE (B_{SSE}).

$$W_{SSE} = \sum_{i=1}^{k} \sum_{x \in C_i} (x - \mu_i)^2 \tag{1}$$

$$B_{SSE} = \sum_{i=1}^{k} n_i (\mu - \mu_i)^2 \tag{2}$$

The intuition behind the W_{SSE} is that it is measuring the closeness of each point to its specific cluster. In contrast, B_{SSE} is measuring the distance between each cluster. The n variable in equation two here indicates the total number of points in each cluster. By taking the ratio of W_{SSE}/B_{SSE}, one may get a ratio of how close each point is in each of their own central clusters over how far each cluster is from each other. Henceforth, a low ratio means that each point is closed to their central cluster, and each cluster is well separated.

3.2.4 Training

To start with, one may need to specify the k, which is the number of clusters. This can be done using the elbow method [15], which is to find the optimal number of k for the given data. This is done using the B_{SSE} as its metric. The optimal k can be found by looking at the reduction in the B_{SSE} with each increase in k. Choose k where the reduction of B_{SSE} is still significant.

Figure 6 below displays the elbow method that was done on the k-means method. Here, each point consists of a single taxi trip with five dimensions, that is fare/distance, fare/minute, trip frequency, tip amount, and miscellaneous amount. The optimal k here is 4, which means there may be 4 clusters that minimize the cohesion for each cluster well. After getting the optimal k, one may only need to specify the k in the k-means and let the algorithm finds the hidden cluster. However, before running the k-means, the data need to be scaled first, as imbalanced scaling may drag the cohesion and separation metrics, which resulted in less accurate cluster quality. The scaling that is used here is the min-max scaler [16].

The other method is the equal width bin. This second method may need some processing steps, as one may need to aggregate the result from each attribute bin into one single score metric. Thus, here are the steps to get the clusters:

1. Use the elbow method to get the optimal number of bins in each attribute.
2. Divide each attribute with k bins based on the results from the elbow method (1.)
3. Each labeled bin values now act as a score. For instance, fare/distance with labeled bin 1 may indicate a low fare per miles compared to label 4.
4. Do a row sums on each of the labeled attributes. This will result in a single scoring value. For instance, a taxi trip with fare/distance: 4, fare/minute: 2, trip frequency: 1, tip amount: 1, and miscellaneous amount: 4 will get a score of 12.
5. Do an equal 4 width binning on the final scoring value to match the bins in the previous method.

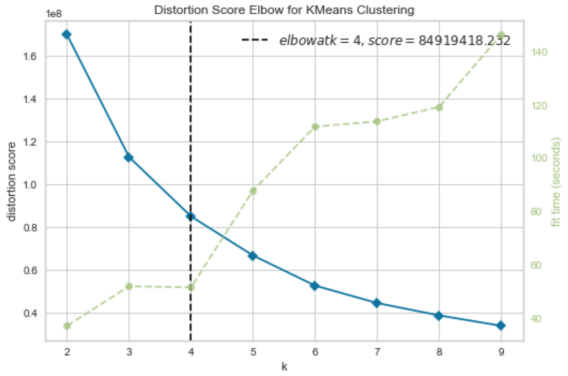


Figure 6: Elbow method to find the optimal k in k-means method

The results for the elbow method in each attribute are 4 for fare/distance, fare/minute, tip and miscellaneous amount, and 5 for trip frequency and they all have similar graph visualization with figure 6.

TABLE OF RESULTS

	k_means	ewb
1	4655823	3464923
2	1581894	11437350
3	3613913	1173877
4	6226875	2355

(a) Elbow method to find the optimal k in k-means method

	score
count	1.607850e+07
mean	9.449215e+00
std	1.325737e+00
min	5.000000e+00
25%	9.000000e+00
50%	9.000000e+00
75%	1.000000e+01
max	1.800000e+01

(b) Equal width score statistics

Figure 7

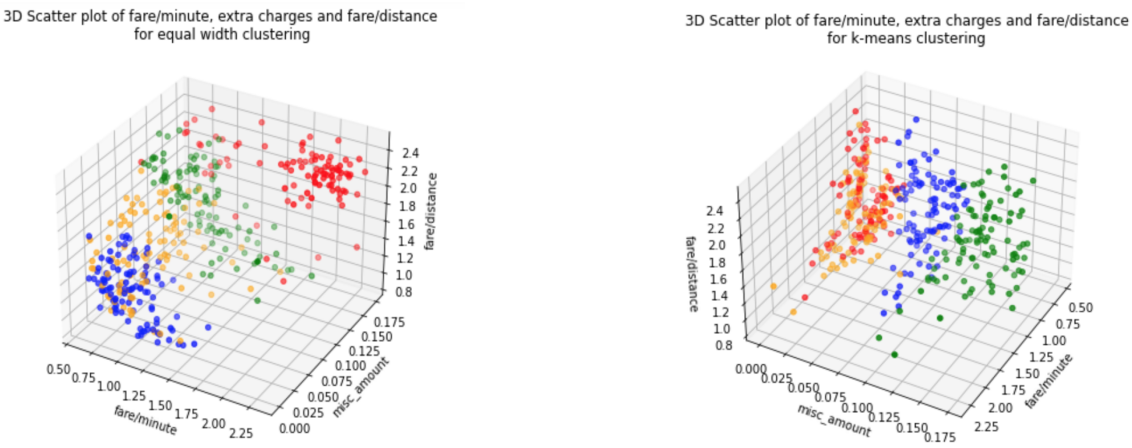
The table in figure 7a shows the number of points allocation in each cluster. It can be seen that the points are more equally distributed in k-means compared to the equal width bin. This is due to the nature of the k-means algorithm, as it grouped points with its closest centroids. In contrast, the equal width binning method was done in a predefined scoring metrics. Figure 7b indicates that the maximum score was achieved at 18, whereas the minimum is at 5. This means the range for each bin is at 3.25. Hence, based on the equal width binning result, it is quite rare for a trip to get a score that exceeds 14.75. However, this does not apply to the k-means cluster, which means those points clustered at label 4 does not mean that they will have a high score on the taxi trip.

Metrics	EWB	K-Means
Cohesion	4339630.906	3006309.004
Seperation	1809600.669	4052307.335
Cohesion/Seperation	2.398	0.742

Figure 8: Cluster metrics result

In addition, figure 8 above displays the clustering metrics for each method. It appears that K-means indeed produce better clusters compared to the equal width binning, with W_{SSE}/B_{SSE} of 0.742. The equal width binning method has a larger cohesion

compared to k-means by 1.44, and less separation by 0.44. However, numbers may not be enough to judge the quality of the clusters; thus, a 3d plot of a sampled data may show the structure of the clusters within the data.



(a) 3d plot of equal width cluster

(b) 3d plot of k-means cluster

Figure 9: Label color: blue=1, orange=2, green=3, red=4, 500 samples from each clusters

Judging from the visuals above, it turns out that the equal width clusters in figure 9a may have a better cluster compared to the k-means clusters. This can be seen in how the points are well clustered based on fare/minute, miscellaneous amount, and fare/distance. The most obvious one is cluster 4, where these trips are considered expensive in terms of both fare ratios, and costly miscellaneous amount. Nonetheless, the cluster in k-means is more compact in comparison with equal width clusters. However, they are only well separated from the miscellaneous amount. It seems that the result from k-means does not cluster the taxi trips based on the fare ratios, but more towards the miscellaneous amount. Since the first intention of creating the labels is to give ratings for the taxi trip based on its cost, it is decided that the clusters from equal width binning will be used in this report analyst. However, it does not mean that the clusters from k-means are worst, as in this case, their interpretations may not be well suited for this problem. It may be best to investigate further on the k-means clusters in future studies.

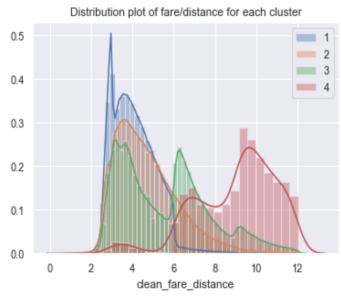
The result of the clusters can be seen in figure 10a-10e. Here, it can be seen that the clusters are well distinguished using miscellaneous amount. Cluster 3 and 4 may have the highest cost in terms of the miscellaneous cost, with a median of around 0.135% of the total cost of the taxi trip. The least is in cluster 1 and 2 with a median of around 0.01% and 0.055%. Moreover, cluster 4 may have the most expensive fare ratios compared to the other clusters, as seen from the distribution plot in figure 10a and 10b. In terms of trip frequency, it is expected that the median in each of these clusters are low, as high hourly trip frequency may only exist in certain areas of New York, such as Manhattan and airports. However, it can be seen cluster 4 may not have any high-frequency trip routes, as its maximum frequency is only at 7 trips per hour. Most of the popular trip routes may be included inside cluster 2 and 3, leaving cluster 1 as neutral. Unfortunately, one may not be able to distinguish each cluster well in terms of tip amount, as seen from the boxplot in figure 10e.

3.2.5 Model Refinement

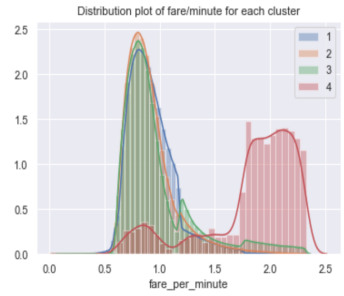
In addition, further refinement of the cluster was done by excluding the least effective attributes for the cluster construction. Hence, the tip amount was excluded. The score for each trip was re-calculated, and the final score for each trip was binned again using the same method, which is an equal width bin with 4 groups. Unfortunately, the ratio for W_{SSE}/B_{SSE} increased to 2.52, which is 0.122 greater than the old cluster. This might indicate that the tip amount may still be relevant in constructing the clusters for this taxi trip ratings. Moreover, figure 10f below shows that cluster 4 now may have similar fare/distance with the other clusters, as compared to figure 9a where the former was still located at the high-end scale of the fare ratios. For these reasons, it is decided to keep the tip amount as part of the cluster's dimension.

3.2.6 Final Result

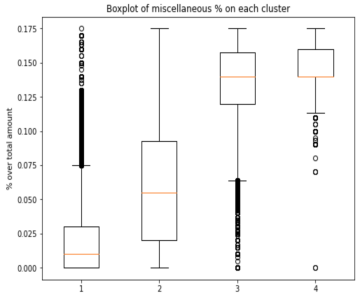
The final result for the clusters is seen from figure 9a and 10 with W_{SSE}/B_{SSE} of 2.398. The best attribute to distinguish taxi trip clusters is the miscellaneous amount. Fare/distance and fare/minute may only separate cluster 4 from the other clusters. Since the fare in cluster 4 might be expensive, this may result in low trip frequency for the trip routes. Nonetheless, the tip amount may be varied in each cluster. This is because tips depend on the passenger's discretion. All in all, this equal width scoring method was able to distinguish each trip based on the taxi trip's miscellaneous amount. This rating method was able to



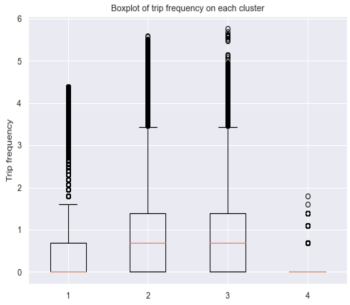
(a) Fare/distance distribution in each label



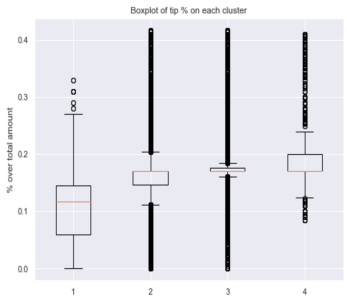
(b) Fare/minute distribution in each label



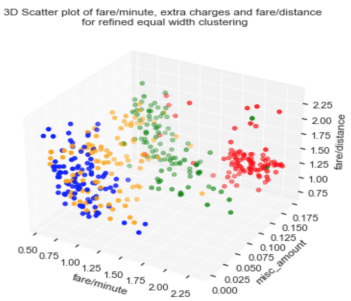
(c) Miscellaneous amount in each label



(d) Log normalised trip frequency in each label



(e) Tip amount in each label



(f) Equal width method cluster refinement

Figure 10: Each cluster results for each attribute (a-e) and model refinement (f)

separate the extreme cases of high fare ratios trips from its peers, which is cluster 4. Thus, for the passenger's point of view, rating 1 is considered to be the cheapest trip compared to others, as it has the lowest miscellaneous cost, followed by rating 2. Rating 3 may have higher fare/distance and miscellaneous cost compared to rating 1 and 2. Finally, cluster 4 is considered to be the most expensive trip. However, these clusters may not be good enough to be the final metrics to rate a taxi trip, as cluster 1, 2 and 3 may still overlap with each other. Nonetheless, if one's intention is to just separate trips based on the miscellaneous amount, then k-means may give the best result for the clusters, as seen in figure 9b.

4 Classifying Clusters

In the previous section, a method has been developed to identify clusters to rate taxi trips based on its fare/distance, fare/minute, trip frequency, tip amount and miscellaneous amount. The next step would be to develop a model to classify or rate taxi trips based on the defined clusters. However, the attributes used in constructing the clusters can only be retrieved after the trip was done. Thus, one may need to formulate a new set of attributes to predict the ratings of the trip, and these attributes may need to be available before the trip happens. Henceforth, these selected attributes are historical trip distance and time that acts as an estimated trip distance and time for the trip, rate code id, the hour the passenger is picked up, the day of the week, the trip routes (pickup and drop off location), and two external attributes, that is precipitation and temperature. However, this might be quite challenging since none of these attributes was used as the dimension for the clusters.

4.1 Preparation

There is minor preparation that was conducted to prepare the training and testing dataset. Listed below are the steps for the further preprocessing:

1. Combine both pickups and drop off location id attributes as single attributes called destination. This can be done by creating a dictionary of all combination of pickup and drop off location id and turn them into nominal values.
2. Concatenate the precipitation and temperature data based on the date.
3. Concatenate the clusters found in the previous section.
4. Split the data into train and test set.
5. Remove all of the trip distance and time from the test set, and change them with historical trip distance and time from the median of each unique trip destination routes based on each hour.

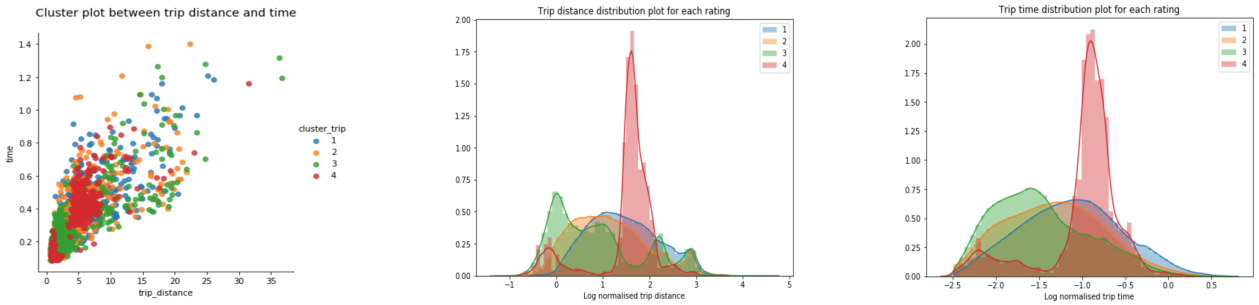
The test data reflect the real-world case scenario, where both trip distance and time are only estimates. In this way, one may be able to test whether it is feasible to get the ratings of the trip without knowing any of the taxi cost.

4.2 Brief Summary Statistics

4.2.1 RatecodeID, Hour, Day of Week

In the previous assignment, it has been shown that different hours and days of the week may have different trip frequency trends. Moreover, rate code id 5 on average may have higher fare per distance compared to the standard fare - rate code id 1. Thus, these attributes might have some contribution in detecting the trip ratings.

4.2.2 Trip distance and time

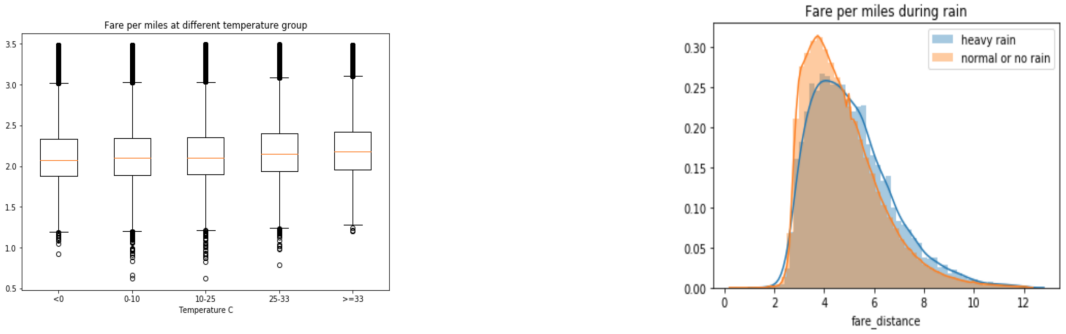


(a) Scatter plot of trip distance vs time (b) Trip distance distribution (c) Trip time distribution

Figure 11: Summary statistics for trip distance and time in each rating

The figure above indicates that taxi with a rating 4 may have a higher probability that the trip distance will be around 2.71-7.41 miles and trip time around 22-36 minutes. Rating 4 most of the time may have higher trip distance compared to the other ratings. However, this is not the case for trip time. Rating 4 trip time may be similar to rating 1. The lowest amongst all is maintained by rating 1, with high probability density that the trip distance is around 0.74-1.35 miles and trip time around 8-13.38 minutes. Nonetheless, by looking at figure 11a, it turns out that these rating clusters may not be separable just by using these two attributes. But, it seems that rating 4 may be clustered around the lower end of both attribute's scale.

4.2.3 Temperature and Precipitation



(a) Boxplot of fare per miles in each temperature group

(b) Fare per miles during heavy rain and normal or no rain

Figure 12

For temperature and precipitation, it may be easier to display their contribution through their relationship with fare per distance. Intuitively, during high rain, there may be more traffic. Thus, the fare per distance should have some deviation compared to normal or no rain. However, figure 12b shows that the deviation may not be significant at all. In fact, they tend to be similar. This phenomenon can also be seen in temperature, where the different temperatures may not affect the fare per miles. Nonetheless, in the previous assignment, the extreme temperature may influence the trip frequency instead of the taxi fare.

4.3 Decision Tree and Random Forest Classifier

The first two proposed models are decision tree and random forest classifier. Since these attributes may not correlate directly to the cluster of the taxi trips, applying a decision rules to these attributes based on a given trip rating may help define the separability of these clusters. In addition, a random forest is a collection of decision trees, where each tree may learn a different subset of attributes and training data. This may reduce the over-fitting of the data, which may result in a lower bias. Moreover, one of the fortes of using these tree classifier is that may be able to handle non-linear data.

4.4 Linear Support Vector Machine

Linear Support Vector Machine (Linear SVC) is best known for its ability to cluster data based on its geographical location in the vector space. However, the main assumption for this model is that the data should be linearly separable. That is why this model will also be used to check whether using the real-world case attributes may linearly separate the taxi trip rating cluster.

4.5 Training and Development

The training was done on the training data set, which will be split further into train and validation set. The test data will not be used for any development, as this set may represent the model's performance on unseen data. The whole training set consist of around 10 million instances, while the test set has around 5 million instances.

The first step of the training process is to train each of the proposed classifiers without any tuning on the parameters. The model was trained on the training set after it was split further for the validation set with 75% and 25% stratified split [17], respectively.

However, there may be a problem in this experiment. As seen in figure 7a, the equal width binning column denotes the number of instances in each cluster. It can be seen that there is a huge imbalance distribution between each cluster. As a result, it is hypothesised that the model may be able to learn rating 2 well, but not for other ratings due to the lack of training records. For this reason, a simple baseline model 0R will be included, as this model classifies all instances with the most popular label.

Classifier	Accuracy
0R	70.48%
DT	64%
Random Forest	73%
Linear SVC	69%

Table 1: Accuracy result for each classifier

By looking at the results from the baseline model, it can be seen that rating 2 dominates the validation set by 70.48%. The only classifier that could beat this baseline classifier is Random Forest with 73% accuracy. Hence, it can be concluded that in this case, the performance of a single tree may be inferior compared to the collection of trees. Moreover, the result from the Linear SVC classifier might indicate that these attributes may not be linearly separable for the clusters. Consequently, it may be best to continue improving the model that beat the baseline model, which is the random forest classifier.

4.6 Error Analysis

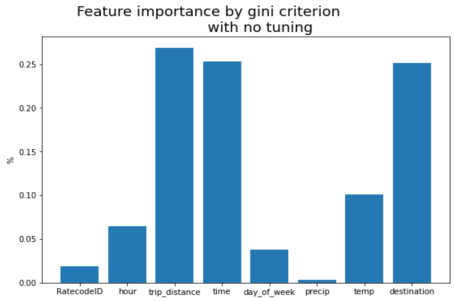
4.6.1 Evaluation Metrics

The next part of the analysis would be to evaluate the random forest model performance. For this, one may need to look at the accuracy, precision, recall and f1-score. Precision, recall and f1-score may be a better justification for evaluating model performance in imbalance label distribution [19].

4.6.2 Original Result

It can be seen that in figure 13b, the model was able to detect ratings 2 with precision, recall and f1-score of 74%, 94% and 83%. This result indicates that the model might be too confident in predicting most of the instances with label 2, as it turns out that 26% of its prediction for label 2 was incorrect. This result actually supports the hypothesis that was defined earlier in the training and development section. Moreover, low recall means high false-negative classification. An example of false negative is if the classifier said that the instance is not 3, even though the ground truth label is 3. This phenomenon can be seen in the result from label 1, 3 and 4. This may be detrimental, as miss-classifying a rating might even lead to a conflict between passenger and taxi driver. This could happen when the trip was rated lower than its ground truth label.

In addition, the feature importance of the random forest classifier can be seen in figure 13a. Intuitively, the feature importance is calculated based on how well the attribute split is in each of its nodes to classify an instance to a class. This is done using



(a) Feature importance

	precision	recall	f1-score	support
1	0.58	0.22	0.32	575132
2	0.74	0.94	0.83	1813328
3	0.65	0.20	0.30	184007
4	0.66	0.17	0.27	348
accuracy			0.73	2572815
macro avg	0.66	0.38	0.43	2572815
weighted avg	0.70	0.73	0.68	2572815

(b) Classification report

Figure 13: Random forest classifier performance without tuning

the gini impurity from each attribute [9]. The result shows that trip distance, trip time and destination routes may be the most effective attributes to classify trip ratings. The sum of the feature importance % for these three attributes are around 75%, which may indicate most of the instances are labelled based on these features' splits.

Moreover, it seems that precipitation may have the lowest contribution in determining the feature classification, with near 0% of feature importance. However, since there are not many attributes that are available in this model, the precipitation will not be dropped from the attribute set. Instead, it will be turned into boolean values where 1 means the precipitation is greater than 0.2 inch, and 0 otherwise. The reason being is that precipitation bigger than 0.2 means heavy rain [20].

Another thing to be considered is the training size. In the first run, the time taken for the model to learn and predict was quite long. Figure 14 shows that reducing the training size may not give a huge impact on accuracy. Moreover, the relationship is linear for both accuracy and the training size. Hence, it is decided to cut the training size to 0.4 from the original size, and allocate the other 0.6 to the validation set. The assumption here is that tuning the model at lower training size may not impact the linear relationship between the training size and the validation accuracy of the model.

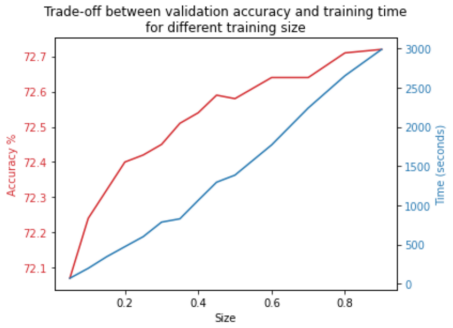


Figure 14: Trade-off between validation accuracy and training time for different train sizes

4.7 Refining Classifier

The random forest was now trained on around 1.5 million instances that were stratified sampled [17] from the original training set. The precipitation was also turned into the boolean value specified in the previous section. The result can be seen in the figure below.

It seems that binning the precipitation may even decrease some of the model accuracies. This can be seen from the rating 2 and 4's recall, which means there is less correctly classified ratings for 2 and 4 over their actual total number. Thus, using continuous precipitation is still preferable.

The last attempt to improve the model is to tune the random forest. This can be done by changing the number of trees in the forest, the maximum number of the sampled attribute used in each tree, the maximum samples to be trained in each tree and the random state. To speed up the process, the random search was used [18]. With a total of 50 iterations, the result for the best model is using the following hyperparameter: number of estimator = 125, random state = 42, maximum samples = 0.5 and maximum features = log2.

	precision	recall	f1-score	support
1	0.57	0.20	0.30	1955450
2	0.74	0.95	0.83	6165314
3	0.65	0.17	0.27	625623
4	0.13	0.47	0.21	348
accuracy			0.72	8746735
macro avg	0.52	0.45	0.40	8746735
weighted avg	0.70	0.72	0.67	8746735

(a) Continuous precipitation value

	precision	recall	f1-score	support
1	0.57	0.20	0.30	1955450
2	0.74	0.94	0.83	6165314
3	0.65	0.17	0.27	625623
4	0.13	0.46	0.21	348
accuracy			0.72	8746735
macro avg	0.52	0.45	0.40	8746735
weighted avg	0.69	0.72	0.67	8746735

(b) Binned precipitation value

Figure 15

	precision	recall	f1-score	support
1	0.68	0.50	0.58	1955450
2	0.83	0.91	0.87	6165314
3	0.86	0.71	0.78	625623
4	0.30	0.78	0.44	348
accuracy			0.81	8746735
macro avg	0.67	0.73	0.67	8746735
weighted avg	0.80	0.81	0.80	8746735

Figure 16: Hyper parameter tuned random forest

Surprisingly, the result of the tuned random forest increased the accuracy by 9%. Although rating 2's recall decrease, other rating's recall did increase quite significantly. Moreover, the f1-score for each rating also escalated. This signifies that both recall and precision of the classifier has improved significantly. The lessen recall from rating 2 can still be off-set by the improvement from other ratings. Ultimately, the model improvement has come to an end; thus, the next step is to test it using the untouched testing data.

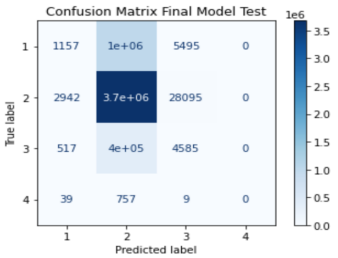
4.8 Final Test

Finally, the model will now be tested in a real-world case scenario. The result from the best-tuned hyperparameter was specified in a new random forest model that contains the whole set of the original training data. Here, the assumption is that the accuracy of this bigger training size may increase, as seen in figure 13.

The test set size has 5 million instances from October-December 2019. Here, both the trip distance and time are the median from its historical records, that is trips from January-September. The median was chosen instead of the mean as mean is sensitive to outliers. This test set may test on the model performance in dealing with the real-life condition, where even the real trip distance and time is not available.

	precision	recall	f1-score	support
1	0.25	0.00	0.00	1023147
2	0.72	0.99	0.84	3708741
3	0.12	0.01	0.02	400117
4	0.00	0.00	0.00	805
accuracy			0.72	5132810
macro avg	0.27	0.25	0.21	5132810
weighted avg	0.58	0.72	0.61	5132810

(a) Classification report



(b) Confusion matrix of the tuned random forest final performance

Figure 17: Final result tested on test data

As expected, the result is not satisfactory. Although it is stated in the classification report that the accuracy is 72%, one should also look at the performance of each rating. From figure 17a, it can be seen that the model failed to classify any rating 4. This can also be noticed from figure 17b on the fourth-row fourth column, as none of rating 4 is correctly labelled. Figure 17b also indicates the behavior of the model on classifying the trips. The model tends to classify most of the trips with rating 2. This can be seen from column 2, where it dominates every other column in the matrix. Nonetheless, the model may still be able to classify some of the trips with rating 1 and 3 correctly. However, its false-negative rate for rating 1 and 3 may out weight the true positive result.

Moreover, by using 0R as a baseline classifier, it can be seen that the final model is on par with the baseline classifier. This is because 72% of the trips in the test set have rating 2. Thus, further improvement for both the model and feature selections are advisable.

5 Discussion

This report has demonstrated that developing a rating system for taxi trips may not be easy. One of the main challenges is to define the number of clusters from historical taxi trips' performance. Setting the number of clusters to 4 may be able to show some separation in each group. Unfortunately, this separation may not be applicable to both trip frequency and tip amount. For trip frequency, taking all combinations for each trip routes resulted in around 30,000 combinations. However, most of these routes may not be popular. For instance, a taxi that is taking pickups and drop off around Staten Island is very rare. This kind of trip may swarm the trip frequency attributes. Henceforth, finding the separation in trip frequency when combined with other attributes may be hard. In addition, the tip amount may depend mostly on the passenger's discretion. Thus, the distribution value for this attribute may be random. Additionally, it can be seen that combining the trip frequency to the other taxi cost metrics may not be effective. Hence, in future studies, one may need to separate trip frequency and make different ratings based on popularity for the taxi driver.

Moreover, the distribution of the rating cluster is dominated by rating 2 and 1. This may indicate that the rule for the New York yellow taxi fare cost is fair, as the fare per miles may be similar across different trip distances. Thus, passengers should not judge whether a trip is expensive or not just by looking at the fare amount, as it should be adjusted with the distance as well.

Nonetheless, this rating method manages to separate some of the trips that may have a more expensive fare ratio compared to the others, as seen in figure 10a and 10b. This is the rating 4 trips, with a higher probability that the fare/distance is around $8-12, and fare/minute around $1.7-2.4. However, these trips are quite rare in 2019. Due to this scarcity, the model may need more rating 4 training data to be able to differentiate it from the other ratings.

This report also has demonstrated that using the real world scenario attributes to classify the trip may not work. Although these attributes may some relations with the attributes that are used to formulate the trip ratings, they should not be used directly to classify the trip ratings. The reason being is that these attributes may not give enough information for the model to separate the ratings. For instance, the trip distance may not be enough to indicate which trip may have a higher fare amount per mile. The model may need the estimated fare to help distinguish these trip ratings. Henceforth, for future studies, one may try to make estimates on each of the metrics that are used to formulate the trip ratings using the real world scenario attributes. Then, the result from these estimates may be used as a predictor variable for the model to rate the trips.

6 Conclusion

In conclusion, the proposed rating method managed to separate the New York yellow taxi trips in 2019 into four different ratings. Rating 1 may have the least cost for the passenger in terms of fare per miles, fare per minute, and miscellaneous cost. This is followed by rating 2 and 3. Rating 4 is the most expensive trip compared to the other ratings. However, it is also the least popular trip in New York. Moreover, the historical tip % in each rating varies. A tuned random forest was used to predict this rating using real-world scenario attributes, and it has an accuracy of 72%. However, the model accuracy here is on par with the 0R baseline model. Henceforth, more research needs to be conducted to further improve the model.

References

[1] Www1.nyc.gov. 2020. TLC Trip Record Data - TLC. [online] Available at: <<https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>[Accessed30September2020]>.

[2] Eia.gov. 2020. Weekly U.S. All Grades All Formulations Retail Gasoline Prices (Dollars Per Gallon). [online] Available at: <<https://www.eia.gov/dnav/pet/hist/LeafHandler.ashx?n=PET&s=EMM_EPM0_PTE_NUS_DPG&f=W>[Accessed28September2020]>.

[3] Wunderground.com. 2020. Local Weather Forecast, News And Conditions | Weather Underground. [online] Available at: <<https://www.wunderground.com/>[Accessed28September2020]>.

[4] Newyork.cbslocal.com. 2020. Passenger Shoots Cabbie In Neck During Fare Dispute In Brooklyn, Police Say. [online] Available at: <<https://newyork.cbslocal.com/2020/09/18/cab-driver-shot-brooklyn/>> [Accessed 28 September 2020].

[5] Www1.nyc.gov. 2020. Taxi Fare - TLC. [online] Available at: <<https://www1.nyc.gov/site/tlc/passengers/taxi-fare.page>[Accessed30September2020]>.

[6] Www1.nyc.gov. 2020. Fact Book - TLC. [online] Available at: <<https://www1.nyc.gov/site/tlc/about/fact-book.page>[Accessed30September2020]>.

[7] Weiss, M., 2012. Fastest Road In America Maximum Speed Limits In New York. [online] Weiss Associates, P.C. Available at: <https://nytrafficticket.com/fastest-road-in-america-and-maximum-speed-limits-in-new-york/> [Accessed 29 August 2020].

[8] Caruso, D., 2020. Taxi Drivers Struggle To Cover Cost Of Gas. [online] The Washington Post. Available at: <<https://www.washingtonpost.com/archive/politics/2005/09/25/taxi-drivers-struggle-to-cover-cost-of-gas/d0b24ffe-079b-4690-bb70-930e771dd3d2/>[Accessed28September2020]>.

[9] Scikit-learn.org. 2020. Feature Importances With Forests Of Trees — Scikit-Learn 0.23.2 Documentation. [online] Available at: <<https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html>[Accessed2October2020]>.

[10] Selenium-python.readthedocs.io. 2020. Selenium With Python — Selenium Python Bindings 2 Documentation. [online] Available at: <<https://selenium-python.readthedocs.io/>[Accessed30September2020]>.

[11] Mishra, S., 2017. Unsupervised Learning And Data Clustering. [online] Medium. Available at: <<https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a>[Accessed30September2020]>.

[12] Stanford.edu. 2020. CS221. [online] Available at: <<https://stanford.edu/~cpiech/cs221/handouts/kmeans.html>[Accessed30September2020]>.

[13] Sciencedirect.com. 2020. Euclidean Distance - An Overview | Sciencedirect Topics. [online] Available at: <<https://www.sciencedirect.com/topics/mathematics/euclidean-distance>[Accessed30September2020]>.

[14] Datacadamia.com. 2020. Statistics - (Discretizing|Binning) (Bin). [online] Available at: <<https://datacadamia.com/data_mining/discretization>[Accessed30September2020]>.

[15] Scikit-yb.org. 2020. Elbow Method — Yellowbrick V1.1 Documentation. [online] Available at: <<https://www.scikit-yb.org/en/latest/api/cluster/elbow.html>[Accessed30September2020]>.

[16] Scikit-learn.org. 2020. Sklearn.Preprocessing.Minmaxscaler — Scikit-Learn 0.23.2 Documentation. [online] Available at: <<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>[Accessed30September2020]>.

[17] Scikit-learn.org. 2020. Sklearn.ModelSelection.Stratifiedshufflesplit — Scikit-Learn 0.23.2 Documentation. [online] Available at: <<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html>[Accessed30September2020]>.

[18] Scikit-learn.org. 2020. Sklearn.ModelSelection.Randomizedsearchcv — Scikit-Learn 0.23.2 Documentation. [online] Available at: <<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html>[Accessed30September2020]>.

[19] Shung, K., 2020. Accuracy, Precision, Recall Or F1?. [online] Medium. Available at: <<https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>[Accessed1October2020]>.

[20] ThoughtCo. 2020. What "Chance Of Rain" Really Means. [online] Available at: <<https://www.thoughtco.com/chance-of-rain-3444366>[Accessed2October2020]>.