FLIGHT DELAY RATE ANALYSIS AND PREDICTION

ANLY 501 Group Project Part Two

NOVEMBER 11, 2018

Guanzhi Wang, Boyang Wei, Xi Yang, Mengtong Zhang, Mengzhi Zhou

1. Exploratory Analysis

1.1 Basic Statistical Analysis

As table 1.1.1 suggests, comparing the mean and median for all numerical attributes in our dataset, only subtle differences exist between these two measures. However, the standard deviation for each attribute is relatively large, which suggests high variations occur in the weather attributes.

The mode of each categorical attributes, as table 1.1.2 suggests, indicates the most frequent airlines, aircrafts, destination cities and other features. Since the sample is derived from the top 30 airports, the most frequent airlines and aircraft type correspond to the biggest airline companies and manufacturers. Therefore, normalization is not needed in most cases, since the data represent the population of airlines in general. However, in machine learning prediction analysis part, normalization is important for training and preventing the effects of bias.

In the first glance, high values of standard deviation, for example, attribute 'cloud_altitude' whose deviation weight is almost 73 percent of its mean, imply high variation and instability of data features. Therefore, it is important to visualize these abnormal distributions of features and see if these abnormal patterns are due to high number of outliers.

features	mean	median	std
cloud_altitude	15453.2	15000	11425.62
temp	22.82524	23	5.755422
dewpoint	22.82524	23	5.755422
visibility	9.640777	10	1.390022
wind_speed	7.403883	7	2.94641
gust_speed	0.848544	0	3.793962
DurationMin	150.0303	107.2333	133.5355

Table 1.1.1 Statistical Summary of Numerical Attributes: (Mean/Median/Standard Deviation)

features	mode
ident	NKS559
aircrafttype	A320
originCity	New York, NY
destinationCity	Chicago, IL
airline	American Airlines
aircraft_manuf	Boeing
Class (Delay)	0

Table 1.1.2 Statistical Summary of Categorical Attributes: (Mode)

As mentioned before, the information generated by Basic Statistical Analysis, especially in terms of mean, median and standard deviation, is limited. Therefore, detection of outliers and visualize the distribution can help better understand the features.

Feature	Number of Outliers
cloud_altitude	9
temp	0
dewpoint	0
visibility	147
wind_speed	3
gust_speed	75
DurationMin	186

Table 1.1.3 Count of Outliers for Numerical Features in Flight Dataset (25-75 percentile)

1.2 Histograms

According to table 1.1.3 above, feature visibility, gust speed and flight duration tend to have large numbers of outliers. Specifically, for flight duration, as histogram 1.2.1 and boxplot 1.2.2 suggest: there are many extremely long flights duration times which lie above 75 quantiles within the feature.

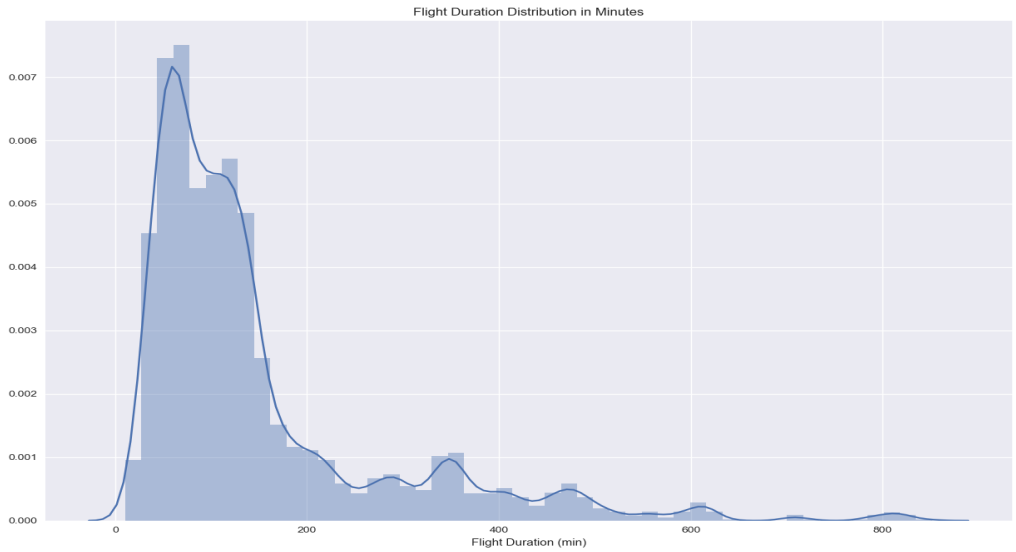


Figure 1.2.1 Histogram of Flights Duration in Minutes

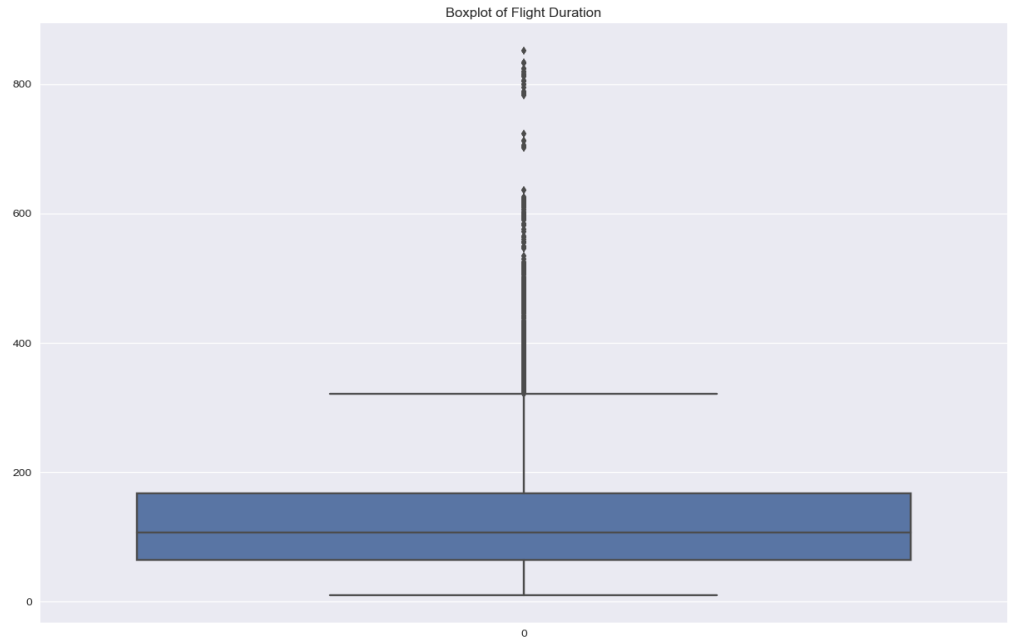


Figure 1.2.2 Boxplot of Flights Duration in Minutes

However, these outliers, also known as long flights duration, tend to have association with the delay rate. As histogram 1.2.3 suggests, after binning the flights duration into 5 duration groups, ranging from '1-2 hours' to '8 hours and above', we can see there are around 600 observations of long-time flights (more than 8 hours) in our dataset. The major reasons of binning flights duration are that this feature is continuous and consists of a range of data from 0 to more than 1000, and binning the time can help distinguish the airlines as short-duration to long-duration flights contextually, which also serve as an interesting factors that correlated with delay rate as the paper will illustrate in later part.

Since these observations are derived from real flights dataset, which are correct and justifiable since long-time flights do exist in major airports.

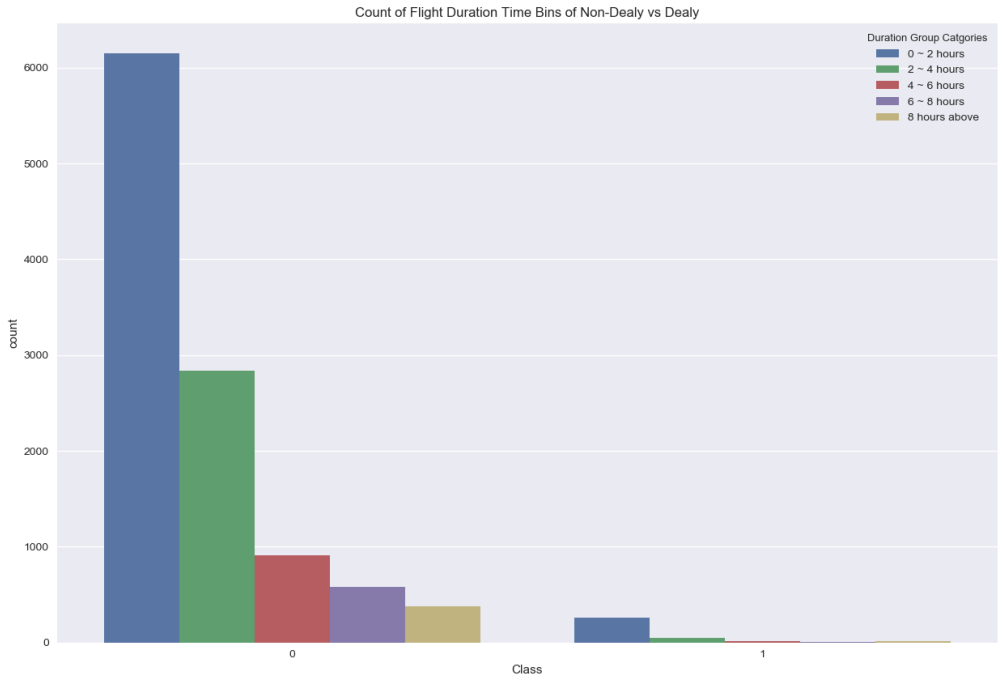


Figure 1.2.3 Distribution of Flights Duration Bins and Number of Delays

Surprisingly, the distribution of delayed (Class 1) and not-delayed (Class 0) in term of flights duration time are similar, which suggests that there is a potential correlation between flights duration and likelihood of delay. Therefore, the outliers in flight duration time feature can not be removed and are valuable in future machine learning predication analysis.

Figure 1.2.4 and 1.2.5 show the distribution of another two features visibility and gust speed whose outliers' numbers are high. As the distribution of 'Visuality', the distribution of discrete levels suggests that there are few levels between 2 to 8, and rather have more 10 and 0, 2 in the dataset. The same logic goes with the distribution of 'Gust Speed' in Figure 1.2.5.

Therefore, the mean and standard deviations can not be used as the only indicators to explain the outliers. In this case, the outliers can not be removed due to the distribution below.

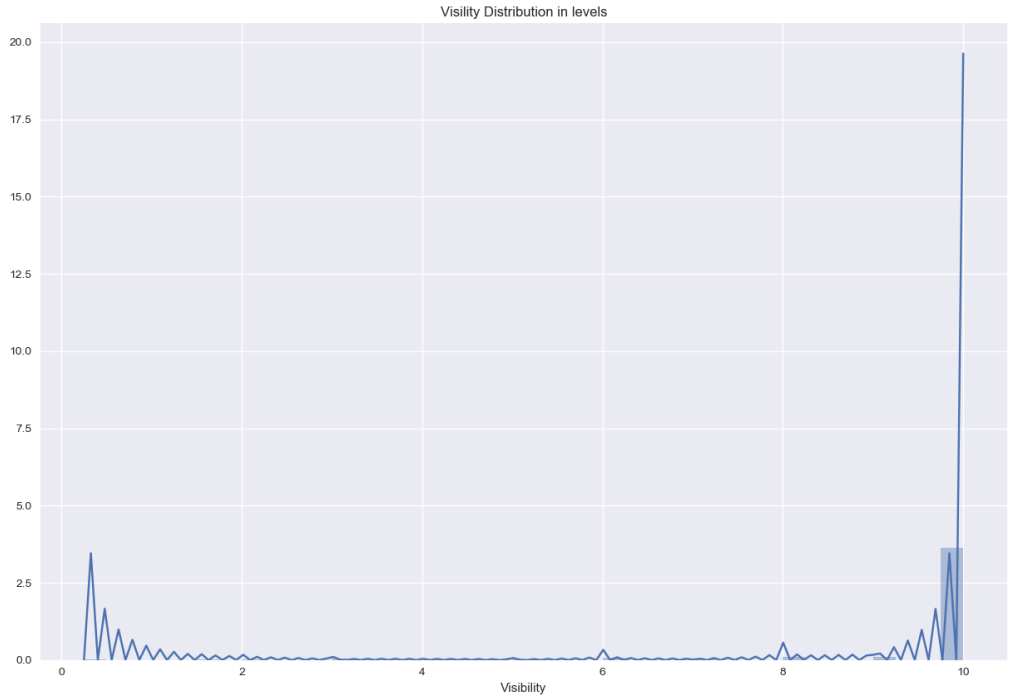


Figure 1.2.4 Distribution of Visibility Levels in Flights Dataset

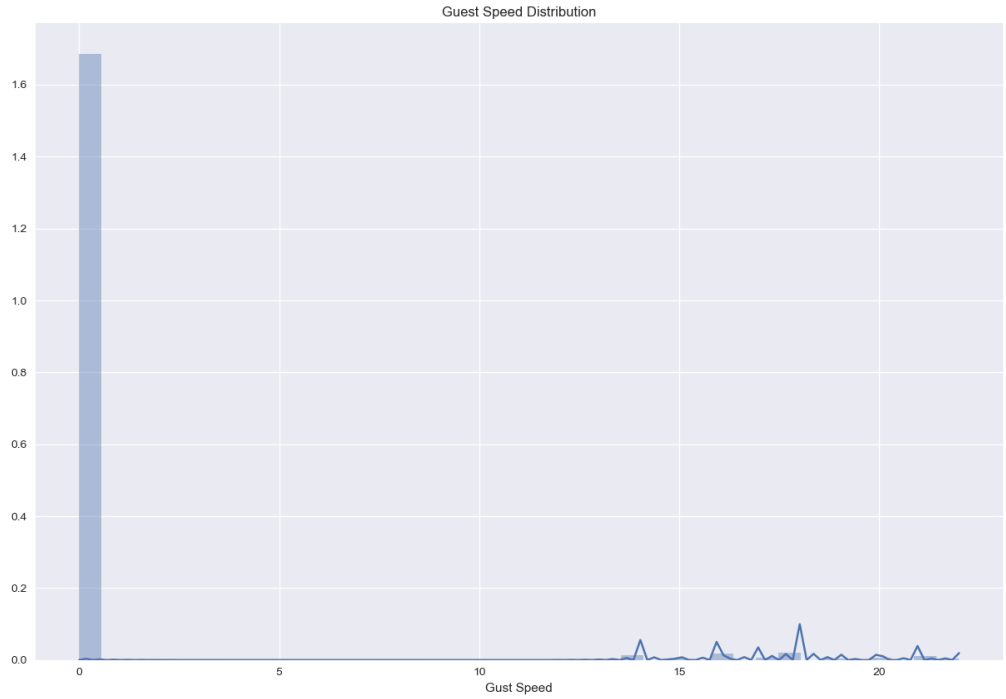


Figure 1.2.5 Distribution of Guest Speed in Flights Dataset

In order to further proceed to analyze flights dataset, the missing values can have negative effects for detecting correlations and building machine learning models. As shown in Figure 1.2.6, the weather features tend to have many missing rows compared to the categorical features related to the information of airlines, destinations, aircraft types and so on. However, removing all the rows containing missing values of weather features will also affect the completeness of other features. Instead, keeping the missing cells under weather features and separating these two types of features to analyze separately is a better option.

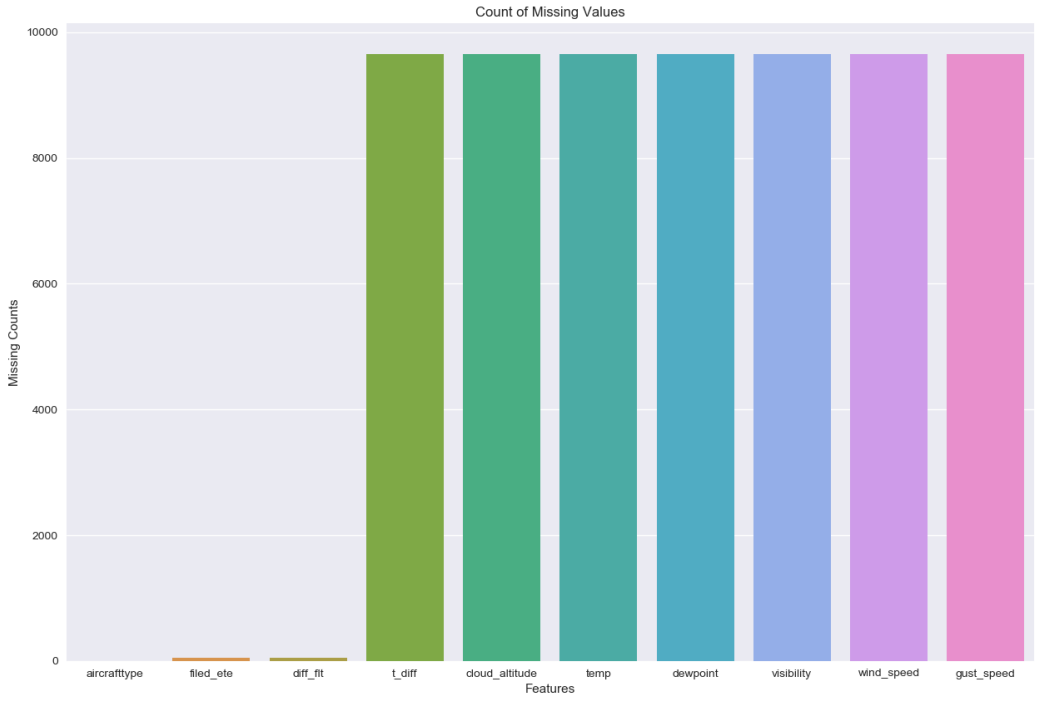


Figure 1.2.6 Count of Missing Values in Flights Dataset

1.3 Correlation Analysis

In order to reduce the dimensionality of flights dataset, which contains 41 features in the raw cleaned data frame, finding the correlated features is important to generate better results in machine learning models, such as Naïve Bayes whose assumption is the independency of all features.

Figure 1.3.1 is the heat map of all numerical features, mostly from weather features that shown the correlation ratio ranging from negative 0.1 to positive 0.3. As the Figure suggests, features such as temperature and dewpoint, wind speed and gust speed are highly correlated, with ratio more than 0.2. Such observations can be further analyzed by using Pearson's R test.

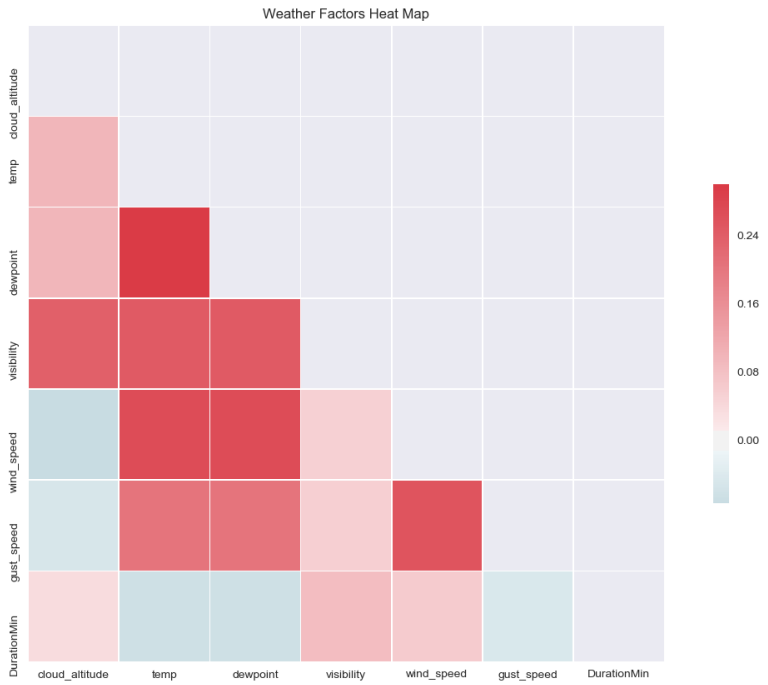


Figure 1.3.1 Weather Features Correlation Heat Map

Table 1.3.2 also suggests the Pearson's R ratio between top three pairs of weather features (Temperature, Wind Speed and Gust Speed); surprisingly, many pairs of weather features are highly correlated. These correlations imply that the weather factors tend to influence each other in a regular pattern, and the dimensionality can be reduced by reducing the highly correlated features.

Correlation	Temperature	Wind Speed	Gust Speed
Temperature	1	0.2695	0.20389
Wind Speed	0.26952	1	0.25490
Guest Speed	0.20389	0.25490	1

Table 1.3.2 Pearson's R rations between top three pairs of weather features

Figure 1.3.3 below also suggests the same results for correlations. The scatter plots between temperature and wind speed shows a positive linear correlation, even the correlation is not as strong as a linear approximation. Besides, the temperature and gust wind/guest wind and wind speed form a straight line of scatter points, as shown on the third subplots in first and second lines and first/second subplots in third lines. This observation is interesting because it seems to suggest the independency, as gust speed will always remain to be 0-2 no matter how temperature and wind speed changes. However, the Pearson's R results show that these pairs are not independent and further analysis will be needed to visualize these discrepancies.

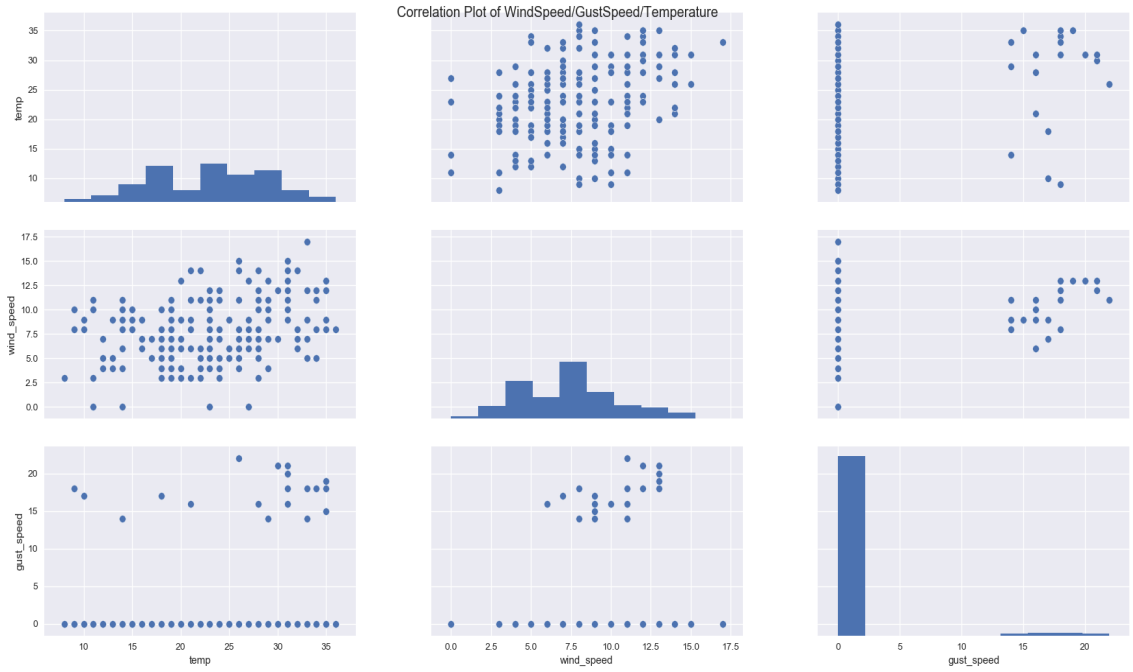


Figure 1.3.3 Correlation Matrix of Top Three Correlated Weather Features in Flights Dataset

1.4 Clustering Analysis

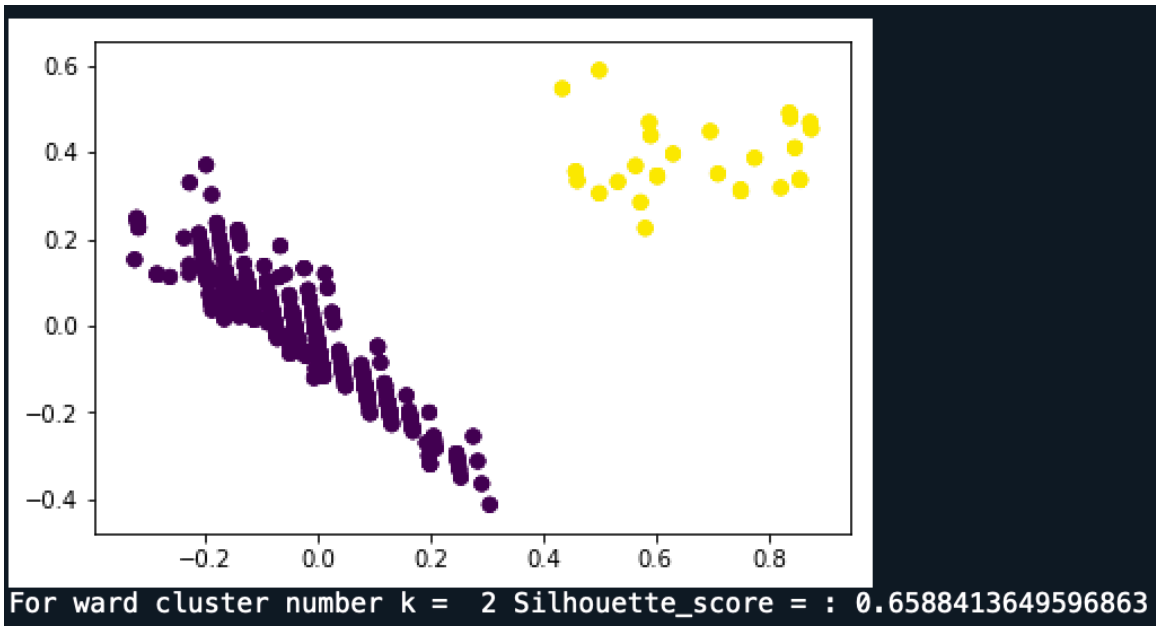
In this part, Ward, K-means and DBscan clustering methods are used to detect special patterns and clusters for collected flight dataset. After analyzing of each attribute in dataset, there are several attributes may contribute to flight delay, such as: wind speed, gust speed, visibility and air speed. These attributes are selected to do clustering analysis in the following step.

Based on outputs and PCA plots generated by Python, we could find there is obvious pattern for collected dataset even we didn't do clustering analysis. Most of points are located on the left side of plot as a line shape and several points spread on the right side. To get precise cluster analysis, Ward, K-means, and DBscan methods need to be processed.

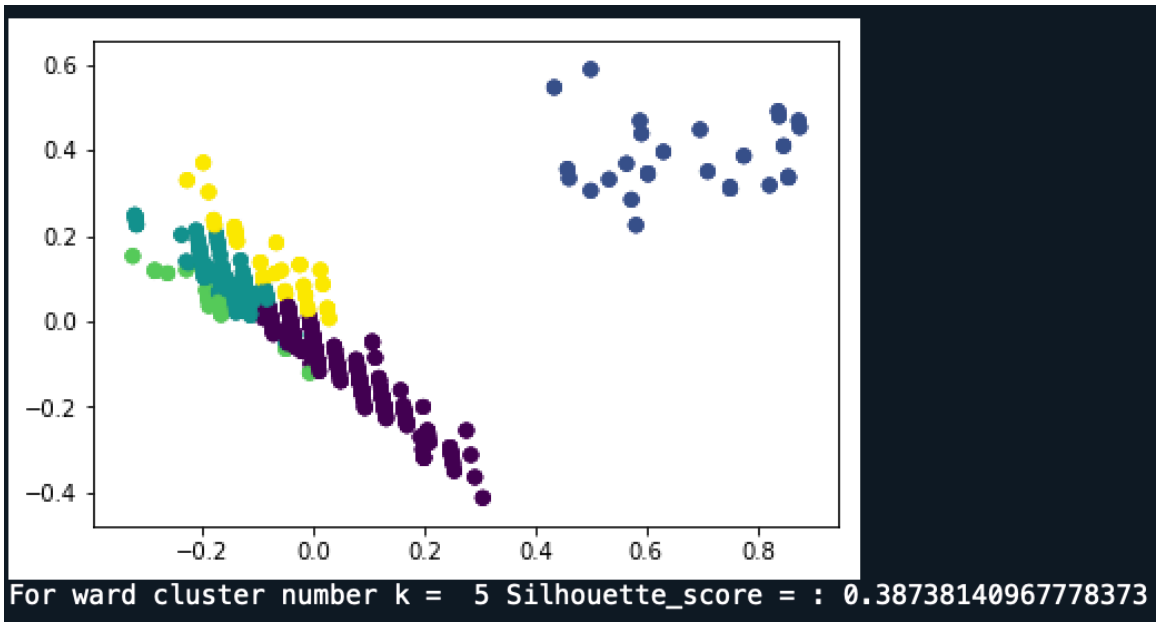
Also, some data-derived insights from clustering analysis are generated with attributes like flight speed , flight altitude and time of the departure delay. The results of clustering these three attributes show departure delay appears more frequently with high flight altitude and high flight speed.

1.4.1 Ward method:

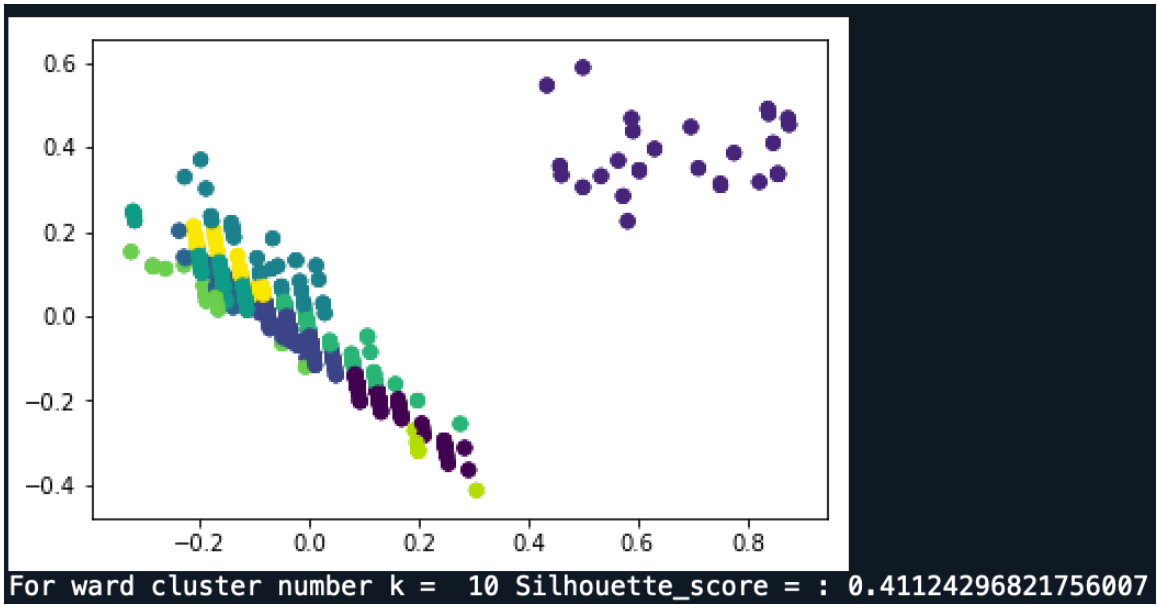
Ward method used variance of clusters as standard to separate data points. Minimizing variance of clusters is goal of ward method. During ward clustering process, cluster number k is set to 2, 5, and 10. Among these 3 different cluster numbers, Silhouette score of ward method is the highest one and equals to 0.6588 when cluster number equals to 2. The PCA of plot demonstrates that cluster number should be 2 which give the best clustering result.



Graph 1.4.1.1 Cluster Result Visualization for k=2 using Ward Method



Graph 1.4.1.2 Cluster Result Visualization for k=5 using Ward Method

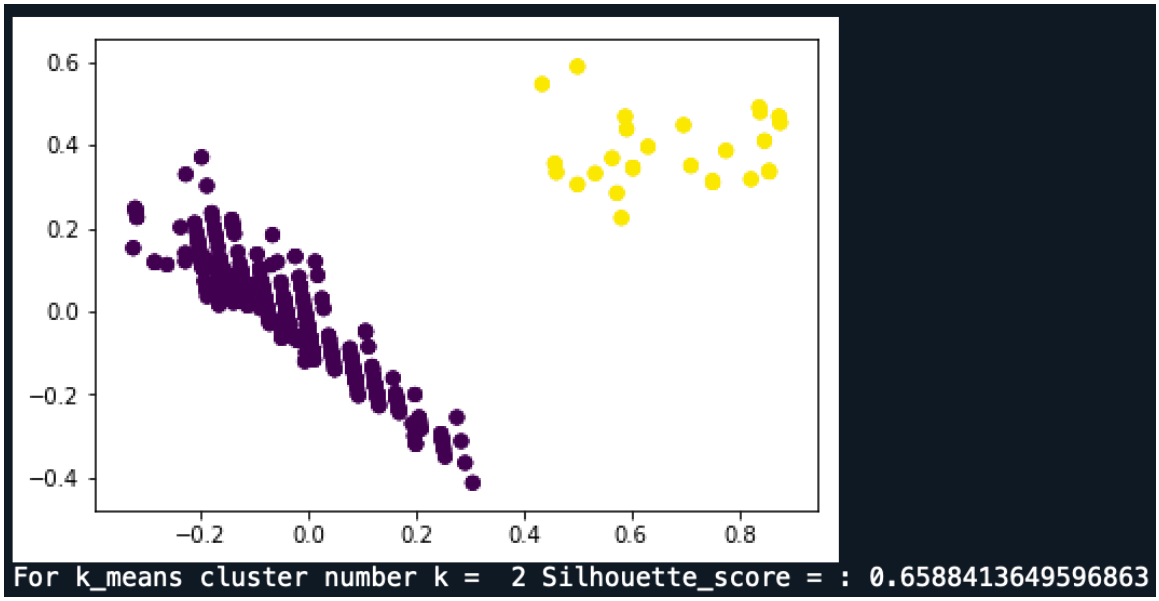


Graph 1.4.1.3 Cluster Result Visualization for k=10 using Ward Method

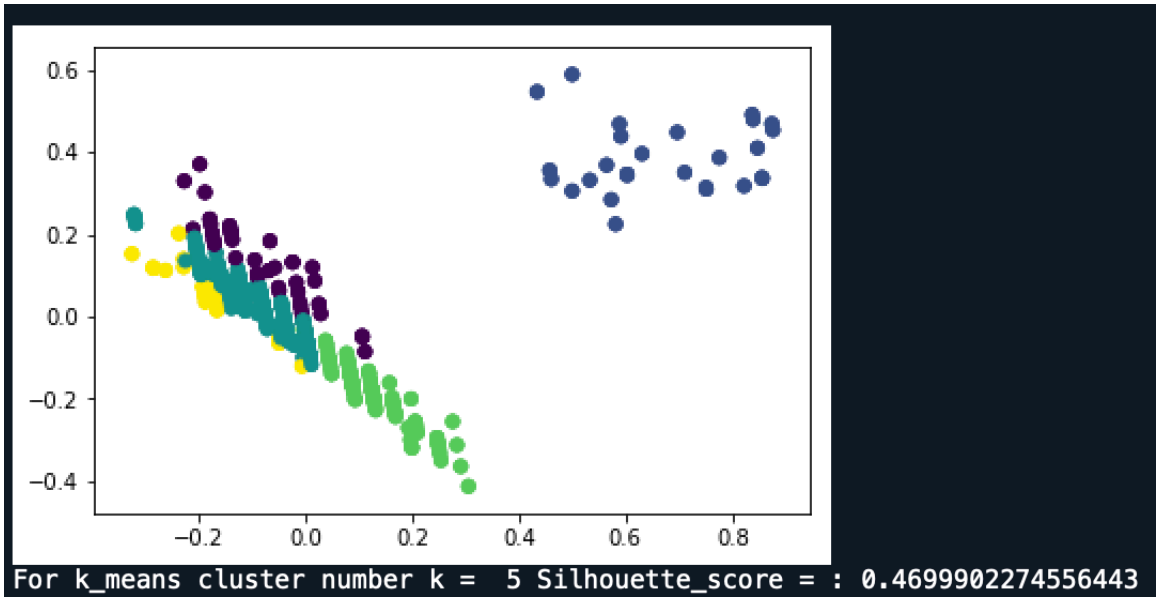
1.4.2 K-means method:

K-means method used distance between points and centroid to separate data points into clusters. During K-means clustering process, cluster number k is also set to 2, 5, and 10 which is same as Ward

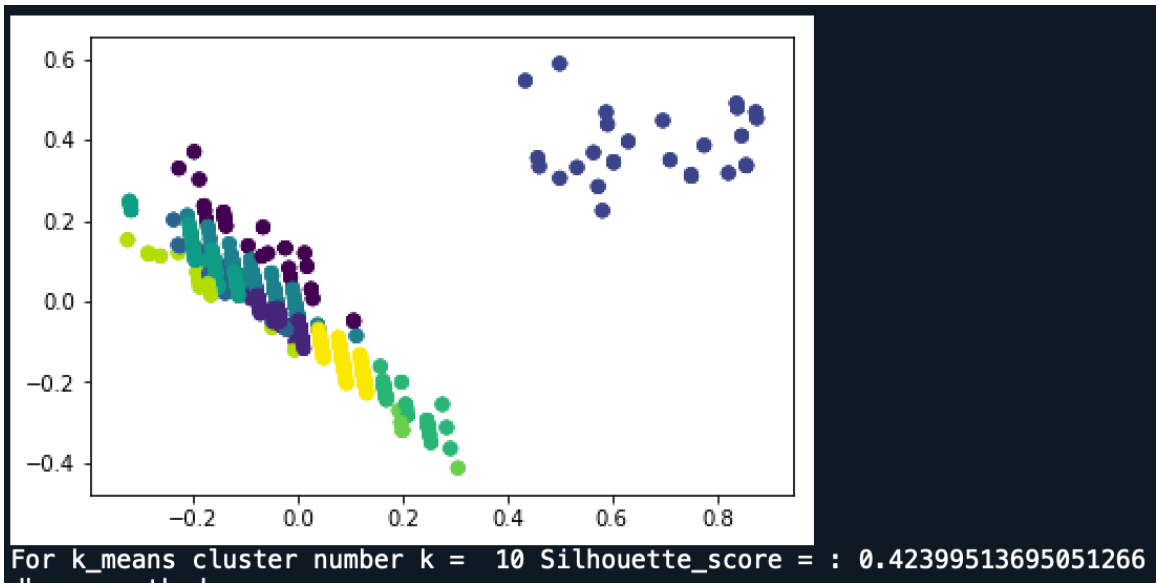
method's cluster numbers. Among these 3 different cluster numbers, Silhouette score of K-means method is the highest one and equals to 0.6588 when cluster number equals to 2. The PCA plot of K-means method also demonstrates that cluster number should be 2 which give the best clustering result.



Graph 1.4.2.1 Cluster Result Visualization for k=2 using Kmeans Method



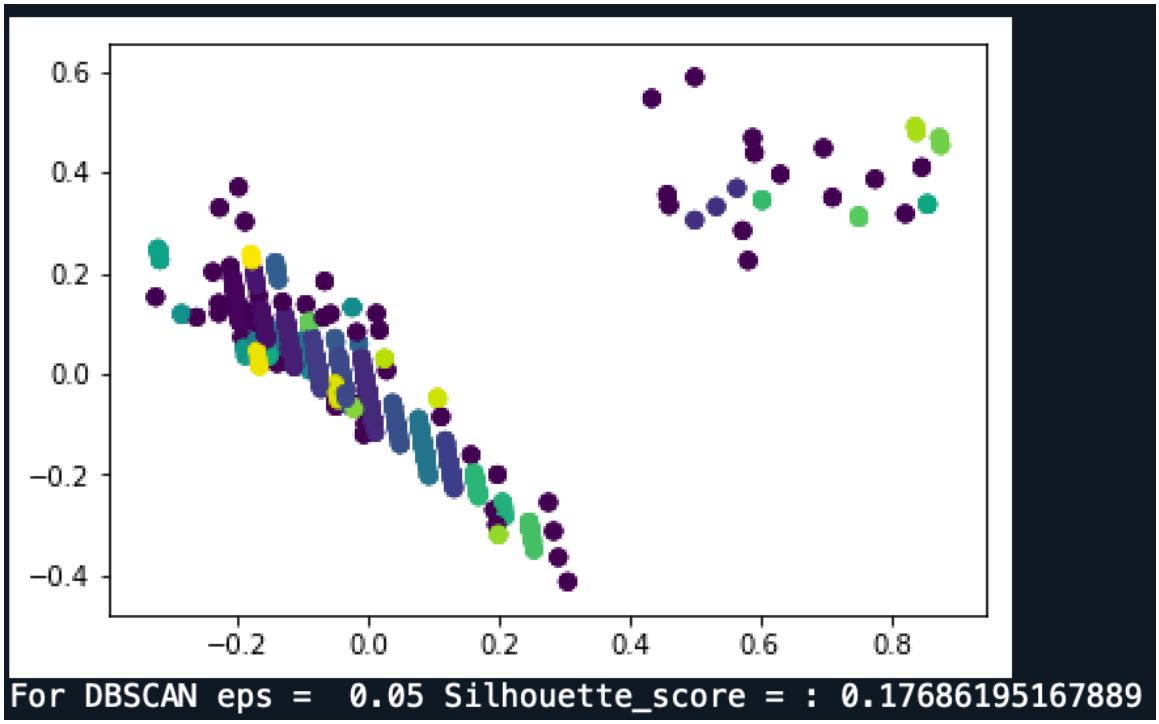
Graph 1.4.2.2 Cluster Result Visualization for k=5 using Kmeans Method



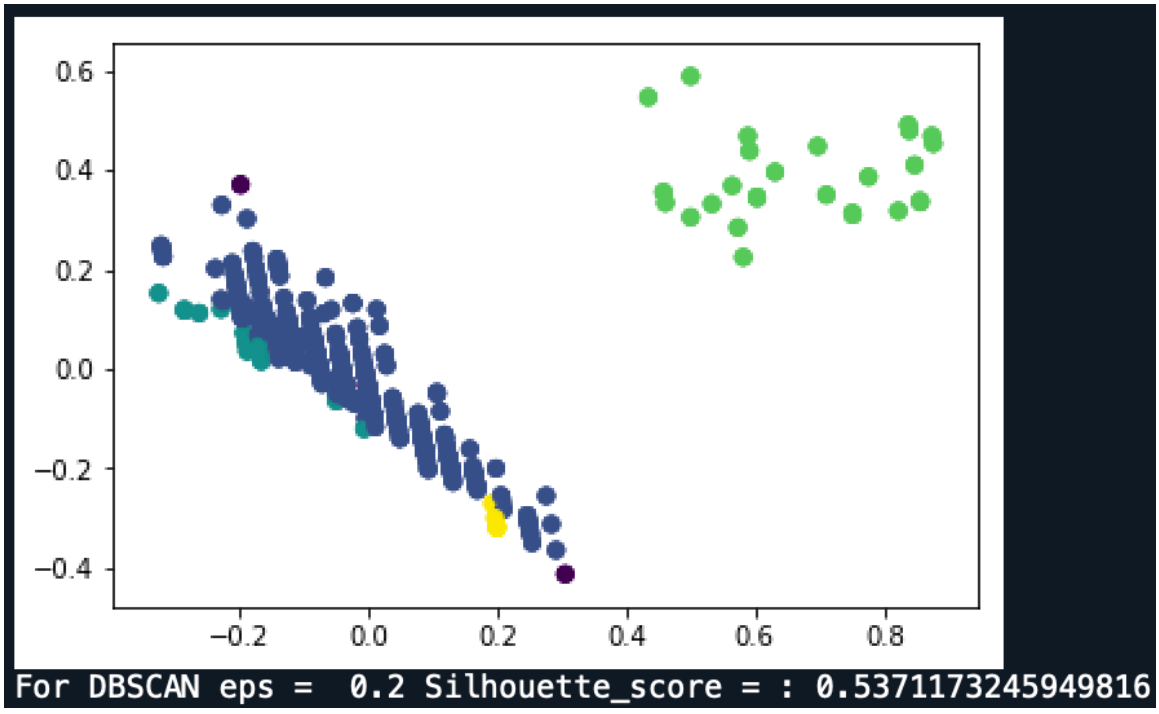
Graph 1.4.2.3 Cluster Result Visualization for k=10 using Kmeans Method

1.4.3 DBscan Method:

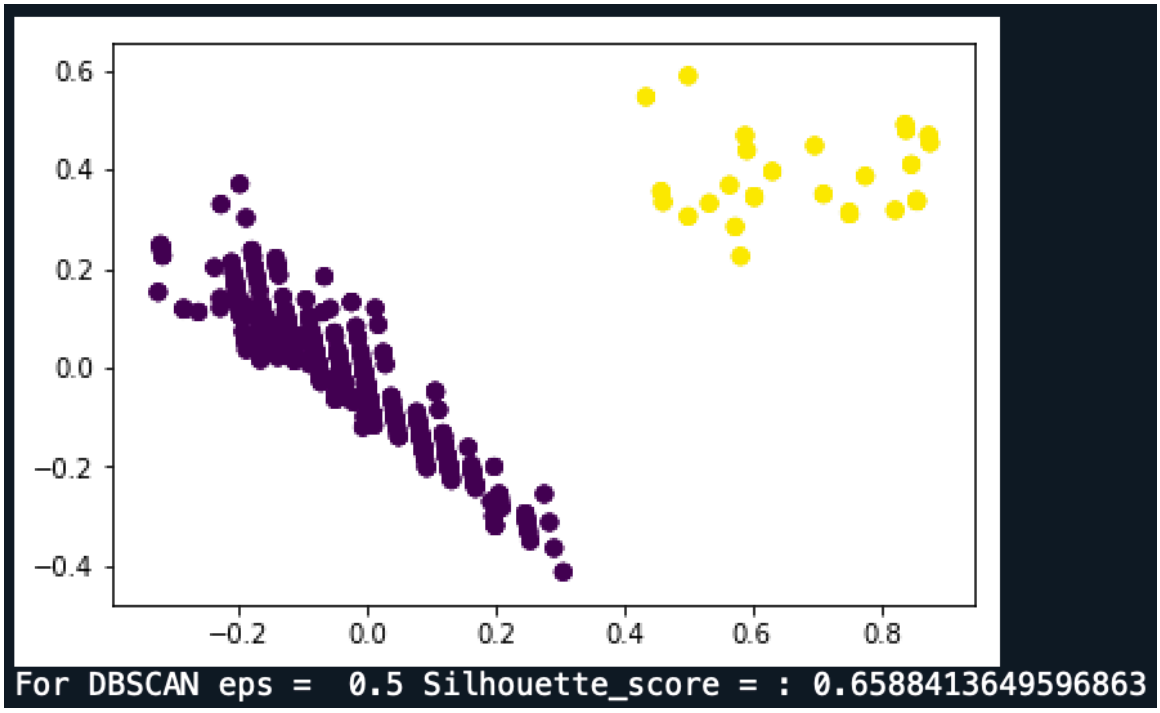
DBscan method used neighborhood distance to separate data points. During DBscan clustering process, maximum distance that to be consider as same neighborhood: eps is set to 0.05, 0.2, and 0.5. Among these 3 different eps values, Silhouette score of DBscan method is the highest one and equals to 0.6588 when eps equals to 0.5. The PCA of plot of DBscan method also shows that two cluster number gives the better clustering result.



Graph 1.4.3.1 Cluster Result Visualization for eps=0.05 using DBscan Method



Graph 1.4.3.2 Cluster Result Visualization for eps=0.2 using DBscan Method



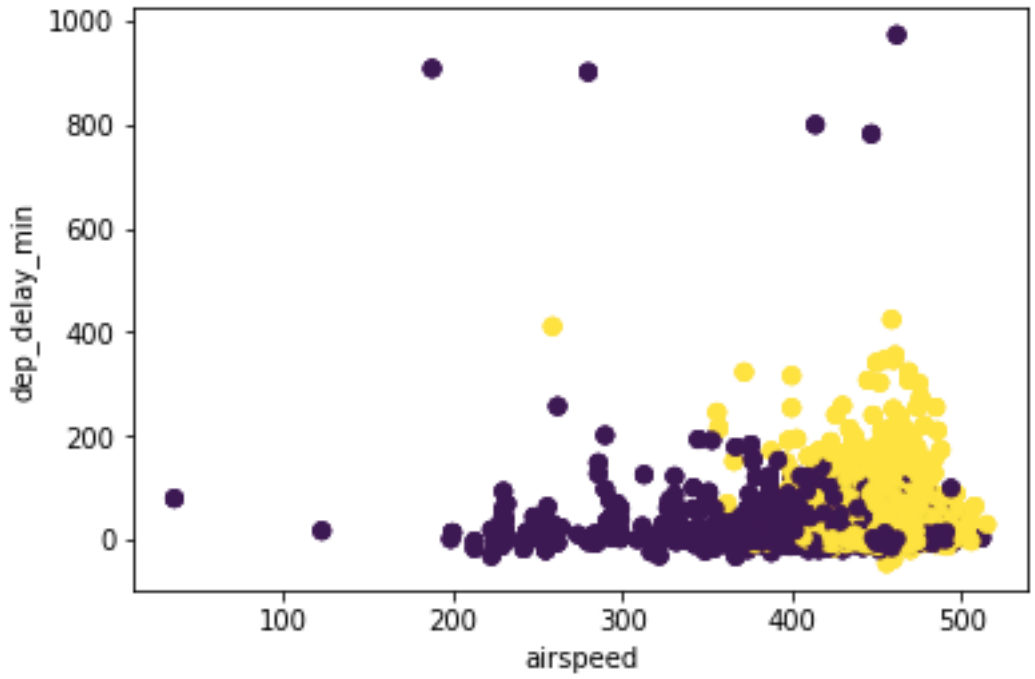
Graph 1.4.3.3 Cluster Result Visualization for eps=0.5 using DBscan Method

All three clustering methods showed that Silhouette score is the highest when cluster number is two. PCA graphs and outputs of Python shows that there are distinct clusters and patterns in our data. Therefore, we could use those attributes: wind speed, gust speed, visibility and air speed to train the machine learning and do the further step.

1.4.4 More Insights from Clustering Analysis

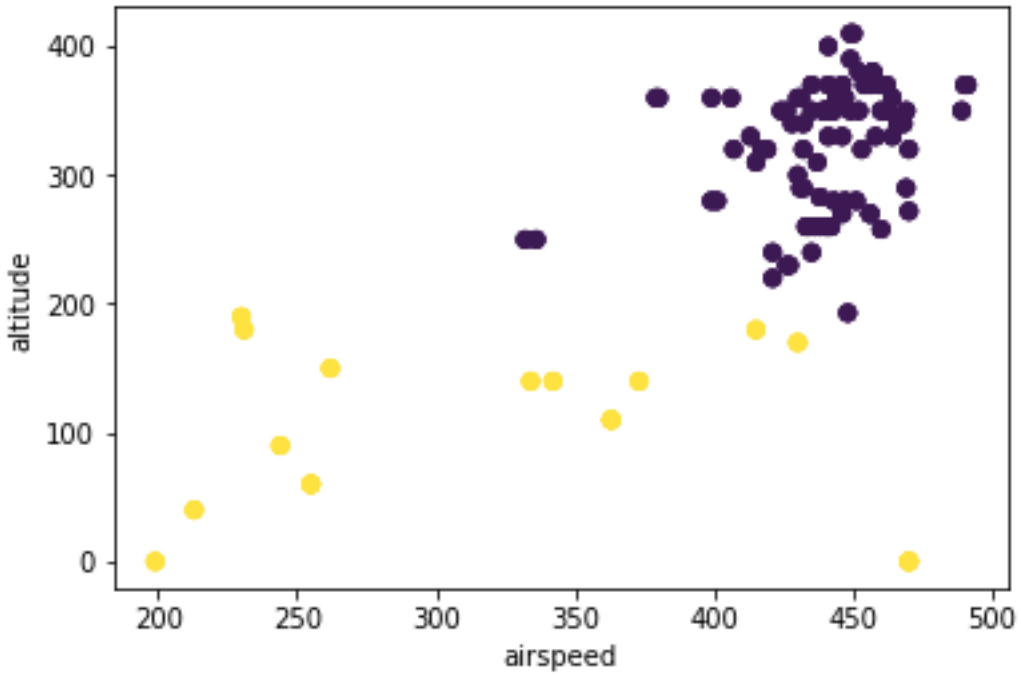
In the first 3 sections of 1.4, attributes related to weather are selected to do clustering analysis. And the result provides guarantee to use these attributes in prediction analysis. Next, attributes related to flights are used to conduct clustering analysis to generate some insights for future prediction analysis.

Departure delay is of great concern for many travelers, and departure delay can be related with many factors such as the altitude and speed of flight. These two factors combined with the time of departure delay are clustered using Ward Method. Best number of clusters is selected using Silhouette score, which turns out to be 2 just as the first 3 sections in 1.4. To tell the practical meaning of clustering method, clustering result is visualized on departure delay time and flight speed instead of PCA graph.



Graph 1.4.4.1 Cluster Result Visualization for Ward Method (using departure delay, flight speed and flight altitude)

It is obvious that the main clusters lies on the right of graph, which means a departure delay typically appears with high flight speed. To further determine the correlation of departure delay with flight speed and flight altitude, the altitude and speed of flights with departure delays are extracted for clustering analysis. This time Kmeans method is used.



Graph 1.4.4.2 Cluster Result Visualization for Kmeans Method (using departure delay, flight speed and flight altitude, only for flights with departure delays)

The results show that there are separated 2 clusters for flight altitude and flight speed. The main class of departure delayed flights show a high speed and high altitude, which indicates a large aircraft and long airline. Thus, in future analysis, flight altitude, flight speed, aircraft type and airline can be factors of interests.

1.5 Association Rules

Association rule mining is a technique to identify underlying relations between different items. In this part, two different Python packages were selected to use the Apriori algorithm.

1.5.1 Basic idea

As the previous work shows, the flights delay could be regarded as “arrival delay” and “departure delay”. In order to find representative rules, the algorithm would be implemented to diverse subsets aimed at different kinds of delay. In this case, two subsets were set up, and their attributes were not all exactly the same.

1.5.2 Features and thresholds selection

There are two fundamental criteria for selecting the features of the subsets. First, the features should be essential and related with the machine learning part to keep the coherence. Second, the data type should be factor. The binning methods were used to process the numeric data. Finally, for arrival delay subset, the attributes are “aircrafttype”, “filed_airspeed_kts_bin”, “destination”, ”actualarrivaltime_week”, “airline” and “arr_delay_sig”. For departure delay subset, the attributes are “aircrafttype”, “filed_airspeed_kts_bin”, “origin”, ”actualdeparturetime_week”, “airline” and “dep_delay_sig”.

For thresholds, as required, at least three different support levels were chosen. They are:

min_support	min_confidence	min_lift	min_length
0.05	0.2	3	5
0.01	0.2	3	5
0.005	0.2	3	5

Table 1.5.2.1 Chosen minimum values for support, confidence, lift and length for rules

Also, when using mlxtend package to find the association rules, min_support=0.002 is the new threshold.

1.5.3 Apyori Package

To do the association rules, the dataset needed to be changed to an appropriate format. To use the Apyori 1.1.1 package, the data was transformed from Table 1.5.3.1 to 1.5.3.2.

Index	aircrafttype	ed_airspeed_kts_t	destination	tualarrivaltime_we	airline	arr_delay_sig
0	A320	D	KBOS	Sun	Spirit Airlines	False
1	A320	D	KBWI	Sun	Spirit Airlines	False
2	A320	D	KBOS	Sat	Spirit Airlines	False
3	A320	D	KBWI	Sat	Spirit Airlines	False
4	A320	D	KBOS	Thu	Spirit Airlines	False
5	A320	E	KBWI	Thu	Spirit Airlines	False
6	A320	E	KBOS	Sun	Spirit Airlines	False
7	A320	E	KBWI	Sun	Spirit Airlines	False
8	A320	D	KBOS	Sat	Spirit Airlines	False
9	A320	E	KBWI	Sat	Spirit Airlines	False
10	A320	D	KBOS	Thu	Spirit Airlines	False
11	A320	E	KBWI	Thu	Spirit Airlines	False

Table 1.5.3.1 Data Before transforming to transaction data

Type	Size	Value
list	6	['A320', 'D', 'KBOS', 'Sun', 'Spirit Airlines', 'False']
list	6	['A320', 'D', 'KBWI', 'Sun', 'Spirit Airlines', 'False']
list	6	['A320', 'D', 'KBOS', 'Sat', 'Spirit Airlines', 'False']
list	6	['A320', 'D', 'KBWI', 'Sat', 'Spirit Airlines', 'False']
list	6	['A320', 'D', 'KBOS', 'Thu', 'Spirit Airlines', 'False']
list	6	['A320', 'E', 'KBWI', 'Thu', 'Spirit Airlines', 'False']
list	6	['A320', 'E', 'KBOS', 'Sun', 'Spirit Airlines', 'False']
list	6	['A320', 'E', 'KBWI', 'Sun', 'Spirit Airlines', 'False']
list	6	['A320', 'D', 'KBOS', 'Sat', 'Spirit Airlines', 'False']
list	6	['A320', 'E', 'KBWI', 'Sat', 'Spirit Airlines', 'False']

Table 1.5.3.2 Transaction data after transformation

After that, apply the command “apriori(data, min_support, min_confidence, min_lift, min_len)”. The outcome is with the class generator, so use “list( )” to change the data type. The part of list with min_support=0.01 in arrival delay subset shown below:

```
[RelationRecord(items=frozenset({'B737', 'Southwest Airlines'}),
support=0.06597671410090557,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'B737'}
), items_add=frozenset({'Southwest Airlines'}),
confidence=0.7434402332361516, lift=9.007512543754627),
OrderedStatistic(items_base=frozenset({'Southwest Airlines'}),
items_add=frozenset({'B737'}), confidence=0.7993730407523512,
lift=9.007512543754627)]),
RelationRecord(items=frozenset({'Southwest Airlines', 'Sat'}),
support=0.07373868046571798,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'Sat'})
, items_add=frozenset({'Southwest Airlines'}),
confidence=0.3084415584415584, lift=3.7370740544721732),
OrderedStatistic(items_base=frozenset({'Southwest Airlines'}),
items_add=frozenset({'Sat'}), confidence=0.8934169278996865,
lift=3.7370740544721737)]),
RelationRecord(items=frozenset({'Southwest Airlines', 'B737',
'False'}), support=0.06390685640362224,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'B737',
'False'}), items_add=frozenset({'Southwest Airlines'}),
confidence=0.7373134328358208, lift=8.933280306929303),
OrderedStatistic(items_base=frozenset({'False', 'Southwest
Airlines'}), items_add=frozenset({'B737'}),
confidence=0.7942122186495176, lift=8.949359256794127)]),
```

Figure 1.5.3.1 List with min_support=0.01 in arrival delay subset

However, we can find that the “min_length” attribute in the command is actually invalid, and the sizes of the rules were from 2 to 5. Also, the list is still bewildering, so the further step is necessary. After the several steps’ processing, six “.txt” files were wrote. The files names were like “A_B.txt”, where “A” represent the subset’s type and “B” is the minimum support number. The basic format of these files is like:

Rule: ['False', 'Sat', 'KBWI', 'D'] -> Southwest Airlines

Support: 0.016597510373443983

Confidence: 0.6666666666666667

Lift: 3.3472222222222228

==========

Rule: ['KFLL', 'Spirit Airlines', 'False', 'Mon'] -> D

Support: 0.016597510373443983

Confidence: 1.0

Lift: 3.3943661971830985

==========

And for all files, the sizes of the rules follow the ascending order. Under this circumstance, to find the meaningful rules, it is efficient to search from the bottom of the text.

1.5.4 Mlxtend Package

However, during the coding, there were several disadvantages of the "Apyori 1.1.1" package. For example, when setting a small min_support, the speed of the code is quite slow. Also, the output is difficult to process, and it is hard to constrain the rules' sizes and select the frequent patterns. As a further attempt, the "mlxtend" package is selected and the command "from mlxtend.frequent_patterns import apriori" can import the Apriori algorithm function.

Although the algorithm is the same, this new package requires Dataframe as input data format. Also, it needs all elements of the subset as the new columns of the new Dataframe and use "True" or "False" as the class (see the Figure below).

Index	0V7	320	3HF	57B	737	A	A319	A320	A321
15	False	False	False	False	False	False	False	False	False
16	False	False	False	False	False	False	False	False	False
17	False	False	False	False	False	False	False	False	False
18	False	False	False	False	False	False	False	False	False
19	False	False	False	False	False	False	False	False	False
20	False	False	False	False	False	False	False	False	False
21	False	False	False	False	False	False	False	False	False
22	False	False	False	False	False	False	False	False	False
23	False	False	False	False	False	False	False	False	False
24	False	False	False	False	False	False	False	False	False
25	False	False	False	False	False	False	False	False	False
26	False	False	False	False	False	False	False	False	False
27	False	False	False	False	False	False	False	False	False
28	False	False	False	False	False	False	False	False	False

Table 1.5.4.1 Converting data to the required format of Mlxtend Package

Then, use following codes, the outcome with required minimum support and rule size can be obtained. The threshold in this test is min_support=0.002, since it is much more easier to find the rules and frequent sets with long size.

```
frequent_itemsetsarr = apriori(arrdf, min_support=0.002, use_colnames=True,max_len=6)

frequent_itemsetsarr['length'] = frequent_itemsetsarr['itemsets'].apply(lambda x: len(x))

frequent_itemsetsarr[ (frequent_itemsetsarr['length'] >= 5) &
                   (frequent_itemsetsarr['support'] >= 0.002) ].sort_values(by=['support'])
```

Figure 1.5.4.1 Code for implementing association rules

Four ".csv" files were created to save the output, with type "arrival" and "departure" and frequent itemset size 5 and 6.

	support	itemsets	length
5768	0.002070	(E, American Airlines, KDFW, True, Sat)	5
5719	0.002070	(A321, E, Delta Air Lines, True, Fri)	5
5718	0.002070	(A321, E, Wed, American Airlines, True)	5
5857	0.002070	(Horizon Air, KPDX, DH8D, C, Fri)	5
5859	0.002070	(True, KPDX, DH8D, C, Fri)	5
5860	0.002070	(Mon, Horizon Air, KPDX, DH8D, C)	5
5861	0.002070	(Thu, Horizon Air, KPDX, DH8D, C)	5
5792	0.002070	(E, KPHX, False, Southwest Airlines, B737)	5
5854	0.002070	(E, KDEN, True, B788, United Airlines)	5
5791	0.002070	(E, KPHX, False, Sat, B737)	5
5709	0.002070	(D, A321, American Airlines, True, Sat)	5
5787	0.002070	(D, KDEN, Southwest Airlines, Sat, B737)	5
5707	0.002070	(D, A321, American Airlines, True, KPHL)	5
5786	0.002070	(D, KBWI, True, Southwest Airlines, B737)	5
5705	0.002070	(Sun, KBWI, True, A320, Spirit Airlines)	5
5871	0.002070	(Horizon Air, True, KPDX, C, Fri)	5
5703	0.002070	(Sun, E, True, A320, United Airlines)	5
5710	0.002070	(A321, E, American Airlines, False, Sat)	5
5701	0.002070	(E, True, A320, KSEA, United Airlines)	5

Table 1.5.4.2 Rules sorted by support values

The itemsets in the files were sorted by support values.

1.5.5 Findings and Results

By analyzing the experimental data, the higher the minimum support is, the smaller frequent itemset sizes of the outcome. And the number of rules would also decline rapidly. Focus on the frequent itemsets which have no less than 5 elements inside, there are some results:

In arrival subset, the most frequent patterns with length 5 is (E, False, Southwest Airlines, Sat, B737), with support 0.036999. The second one is (E, False, Southwest Airlines, Sat, B738), with support 0.013972. That means if a person taking Southwest Airlines' boeing 737 or 738 planes in Saturday with airspeed catalog "E", there is large probability that this flight will not has arrival delay.

When the patterns' length is 6, there are only 5 itemsets, they are

	support	itemsets	length
6107	0.002070	(KLAX, E, False, Southwest Airlines, Sat, B737)	6
6108	0.002070	(E, KMDW, False, Southwest Airlines, Sat, B737)	6
6110	0.002329	(E, False, Southwest Airlines, Sat, KSAN, B737)	6
6106	0.002846	(D, KLAS, False, Southwest Airlines, Sat, B737)	6
6109	0.003105	(KOAK, E, False, Southwest Airlines, Sat, B737)	6

Table 1.5.5.1 Top 5 Rules (patterns' length=6)

Still the Saturday, Southwest Airlines, Boeing 737, E type airspeed and no delay. The supports became much smaller, but the patterns are consistent with former.

In departure subset, when length equals to 5, the (E, True, Southwest Airlines, Sat, B737) is still the most frequent with support 0.025097. However, the second one is now (D, True, E75L, Mesa Airlines, KIAH) with support 0.017076. Notice that in this pattern, the delay option is "True", which means the high frequency of delay for all Mesa Airlines' flights with E75L plane arriving at KIAH with airspeed type D. When the length is 6, there are in total 11 patterns, they are

support	itemsets	length
0.002070	(E, KPHX, False, Southwest Airlines, Sat, B737)	6
0.002070	(E, KLAS, True, Southwest Airlines, Sat, B737)	6
0.002070	(Horizon Air, True, KPDX, DH8D, C, Fri)	6
0.002329	(E, KMDW, True, Southwest Airlines, Sat, B737)	6
0.002587	(D, True, E75L, Mesa Airlines, KIAH, Fri)	6
0.002587	(D, True, E75L, Sat, Mesa Airlines, KIAH)	6
0.002846	(D, Mon, True, E75L, Mesa Airlines, KIAH)	6
0.002846	(Thu, D, True, E75L, Mesa Airlines, KIAH)	6
0.003105	(E, KBWI, False, Southwest Airlines, Sat, B737)	6
0.003105	(Sun, D, True, E75L, Mesa Airlines, KIAH)	6
0.004140	(E, KBWI, True, Southwest Airlines, Sat, B737)	6

Table 1.5.5.2 Total 11 Rules (patterns' length=6)

The most frequent one still contains the Southwest Airlines, B737 and Saturday. The interesting thing is, in this chart, most patterns are with "delay=true" while in arrival part, all patterns are with "delay= False" if the length is 6. So it might be reasonable to say that the when considering all features, departure delay is more likely to occurs than arrival delay.

Then thinking about the rules. The real key rules should be the rules that demonstrate if the flights will delay or not. Same as the frequent patterns, only the rules whose sizes are as long as possible will be taken into account.

For arrival delay subset, there are 5 key rules:

Rule: ['Southwest Airlines', 'Sat', 'B737'] -> False

Support: 0.05717981888745149

Confidence: 0.9608695652173913

Lift: 11.64188360365272

==========

Rule: ['Southwest Airlines', 'B737'] -> False

Support: 0.06390685640362224

Confidence: 0.7373134328358208

Lift: 8.933280306929303

==========

Rule: ['KFLL', 'A320', 'Fri', 'Spirit Airlines'] -> False

Support: 0.016597510373443983

Confidence: 1.0

Lift: 3.3943661971830985

==========

Rule: ['KBOS', 'Mon', 'JetBlue Airways'] -> False

Support: 0.012448132780082987

Confidence: 0.6

Lift: 4.131428571428571

==========

Rule: ['KBOS', 'Fri', 'JetBlue Airways'] -> False

Support: 0.016597510373443983

Confidence: 0.6666666666666667

Lift: 4.590476190476191

==========

For departure delay subset, the 5 key rules are:

Rule: ['Southwest Airlines', 'Sat', 'B737'] -> False

Support: 0.024062095730918498

Confidence: 0.9789473684210526

Lift: 11.860914040587362

==========

Rule: ['Southwest Airlines', 'E', 'B737'] -> False

Support: 0.01500646830530401

Confidence: 0.8656716417910447

Lift: 10.488466757123474

==========

Rule: ['E', 'Sat', 'B737'] -> False

Support: 0.01371280724450194

Confidence: 0.7910447761194029

Lift: 3.2490840166859645

==========

Rule: ['Southwest Airlines', 'D', 'B737'] -> False

Support: 0.010090556274256144

Confidence: 0.8125

Lift: 9.844239811912226

==========

Rule: ['D', 'Sat', 'B737'] -> False

Support: 0.01034928848641656

Confidence: 0.8333333333333335

Lift: 3.4227771873893027

==========

The rules are generally come from the most frequent patterns that showed before. Basically, the results are not surprising since they agree with the results in machine learning part.

2. Predictive Analysis

2.1 Hypothesis tesing

2.1.1 Testing the departure delay rate and airlines

To rank the carriers with the best and worst on-time arrival records, delay rates for 5 most popular airlines are evaluated. Based on the data collected, there are 5 airlines with largest number of records, which are 'American Airlines', 'Delta', 'United Airlines', 'Southwest Airlines', 'Skywest Airlines'. To compute the departure delay rate, simply take the mean of the 'dep_delay_sig' value with respect to each of the airlines. The bar Figure of delay rates is shown below:

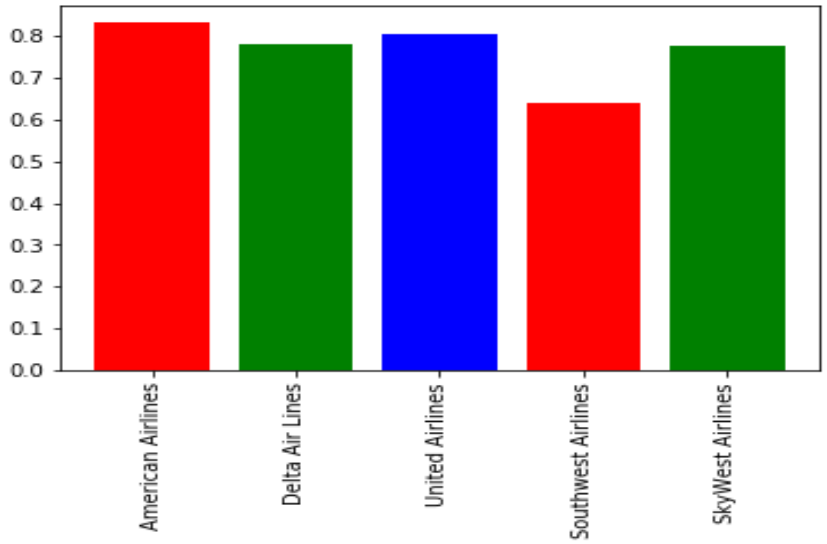


Figure 2.1.1.1 Departure delay rates of top 5 airlines

For these 5 airlines, departure rates are within the range of 0.6 to 0.8. SouthWest Airlines shows the lowest departure delay rate while American Airlines shows the highest. Next step is to conduct an ANOVA test with respect to the delay records of these 5 airlines.

One-way ANOVA

Null hypothesis: The departure delay rates are same for 5 airlines.

Significance level: 0.05

```
F,p=stats.f_oneway(d_data[airline[0]],d_data[airline[1]],d_data[airline
[2]],d_data[airline[3]],d_data[airline[4]])
    ...: print('The F statistics is',F,'\n', 'The p-value is',p)
The F statistics is 36.94303914503367
 The p-value is 1.4246493687256603e-30
```

Figure 2.1.1.2 Result of ANOVA test

The p-value of this ANOVA test is 1.42e-30, which is far smaller than 0.05. Thus, null hypothesis of this test can be safely rejected. Departure delay rates show a significant difference between the 5 most popular airlines.

2.1.2 Testing the arrival delay rate and aircraft manufacturer

Based on data collected, there are 4 kinds of dominant aircraft manufacturers, which are 'Airbus', 'Boeing', 'Embraer' and 'Canadair Regional Jet'. Many of the popular airliners in service worldwide are built by Airbus and Boeing, and these two manufacturers typically provide large passenger airliners. Embraer and Canadair Regional Jet are small jet plane makers and they provide regional jets with less than 130 passenger seats. Also, the mean of the attribute 'arr_delay_sig' is used to estimate the arrival delay rates for these 4 aircraft manufacturers.

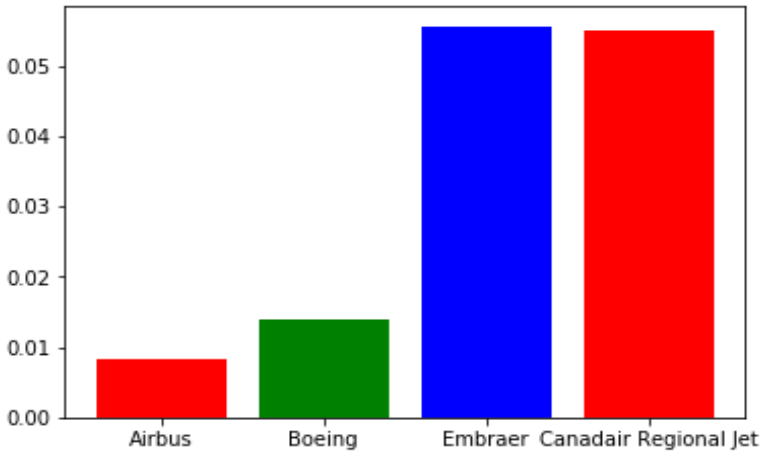


Figure 2.1.2.1 Arrival delay rates of top 4 airlcraft manufacturers

As the bar Figure shown above, Airbus and Boeing show relatively lower arrival delay rate around 0.01, while Embraer and Canadair Regional Jet show a higher rate around 0.05. Airbus' aircrafts are most unlikely to be delayed according to the result.

Next, ANOVA test is used to judge whether the arrival delay rates show a significant difference on the aircrafts manufactured by these 4 manufacturers.

One-way ANOVA

Null hypothesis: The arrival delay rates are same for 4 manufacturers.

Significance level: 0.05

```
In [17]: F,p=stats.f_oneway(Airbus,Boeing,Embraer,CRJ)
    ...: print('The F statistics is',F,'\n', 'The p-value is',p)
The F statistics is 62.58851442808704
 The p-value is 4.115264797673178e-40
```

Figure 2.1.2.2 Result of ANOVA test

2.2 Machine Learning

In this part, the prediction models are trained with a set of flight data and then another set of data is used to validate the model. Since there are two kinds of delay, departure delay and arrival delay, we applied machine learning techniques for both kinds of delay. Moreover, delay could be a binary class (delay or not) or a multi-level class (delay class). Thus, prediction models are applied to predict both binary class and multi-level class for delay.

2.2.1 Predict delay or not with binary class (True or False)

The ultimate goal of this study is to accurately predict if a flight will be delayed or not, and in some situations, we cannot obtain complete and accurate weather attributes such as wind gust speed, dew point temperature, and atmosphere pressure. While missing weather data, it may be hard to predict delay. However, we use a portion of our data with missing weather data for machine learning to predict delay and research the results. This section contains four sub-sections: departure delay prediction with weather data, arrival delay with weather data, departure delay prediction without weather data, arrival delay prediction without weather data.

-Departure delay prediction using binary class (including weather data)

In Figure 2.2.1.1 below, the 10-fold cross-validation results of prediction models during training is shown. Decision Tree and Random Forest preform the best, while Gradient Boosting method becomes the third best. KNN, Gaussian Naïve Bayes, and SVM classifiers do not perform well. This results is anticipated because many attributes of our data are categorical like airplane manufacturer, airline, origin, and destination. Thus, SVM, Gaussian Naïve Bayes, and KNN may not perform well.

```
Accuracy during training:

Decision Tree: 0.983258 (0.012608)
Random Forest: 0.980951 (0.016073)
KNN: 0.827695 (0.033826)
Gaussian Naive Bayes: 0.744610 (0.048545)
SVM: 0.808669 (0.020469)
Gradient Boosting Classifier: 0.891759 (0.019838)
```

Figure 2.2.1.1 Cross-validation results of prediction models (departure delay prediction with weather data).

In Table 2.2.1.1 below, the testing results of six classifiers are shown. From the confusion, matrix, Random Forest classifier accurately predict all on time and delayed flights, and Decision Tree classifier also performs well with high accuracy. Gradient Boosting Classifier is the third best, which is the same as the results during training. The other three classifiers are unreliable though they still have accuracy score up to about 0.8.

Test results using Decision Tree is :

Accuracy score: 0.9817629179331308

Confusion Matrix:
[[70 0]
[6 253]]

	precision	recall	f1-score	support
0.0	0.92	1.00	0.96	70
1.0	1.00	0.98	0.99	259
micro avg	0.98	0.98	0.98	329
macro avg	0.96	0.99	0.97	329
weighted avg	0.98	0.98	0.98	329

Decision Tree

Test results using Random Forest is :

Accuracy score: 1.0

Confusion Matrix:
[[70 0]
[0 259]]

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	70
1.0	1.00	1.00	1.00	259
micro avg	1.00	1.00	1.00	329
macro avg	1.00	1.00	1.00	329
weighted avg	1.00	1.00	1.00	329

Random Forest

Test results using KNN is :

Accuracy score: 0.8115501519756839

Confusion Matrix:
[[34 36]
[26 233]]

	precision	recall	f1-score	support
0.0	0.57	0.49	0.52	70
1.0	0.87	0.90	0.88	259
micro avg	0.81	0.81	0.81	329
macro avg	0.72	0.69	0.70	329
weighted avg	0.80	0.81	0.81	329

KNN

Test results using Gaussian Naive Bayes is :

Accuracy score: 0.7446808510638298

Confusion Matrix:
[[17 53]
[31 228]]

	precision	recall	f1-score	support
0.0	0.35	0.24	0.29	70
1.0	0.81	0.88	0.84	259
micro avg	0.74	0.74	0.74	329
macro avg	0.58	0.56	0.57	329
weighted avg	0.71	0.74	0.73	329

Gaussian Naïve Bayes

Test results using SVM is :

Accuracy score: 0.7872340425531915

Confusion Matrix:
[[0 70]
[0 259]]

	precision	recall	f1-score	support
0.0	0.00	0.00	0.00	70
1.0	0.79	1.00	0.88	259
micro avg	0.79	0.79	0.79	329
macro avg	0.39	0.50	0.44	329
weighted avg	0.62	0.79	0.69	329

SVM

Test results using Gradient Boosting Classifier is :

Accuracy score: 0.8936170212765957

Confusion Matrix:
[[40 30]
[5 254]]

	precision	recall	f1-score	support
0.0	0.89	0.57	0.70	70
1.0	0.89	0.98	0.94	259
micro avg	0.89	0.89	0.89	329
macro avg	0.89	0.78	0.82	329
weighted avg	0.89	0.89	0.88	329

Gradient Boosting Classifier

Table 2.2.1.1 Testing results of prediction models (departure delay prediction with weather data).

The ROC curves of all six models are shown in Figure 2.2.1.2 below. By using the area under the curve and combining the results above, it is not hard to conclude that when we have all the attributes including weather data, **Random Forest is the best classifier to predict departure delay**.

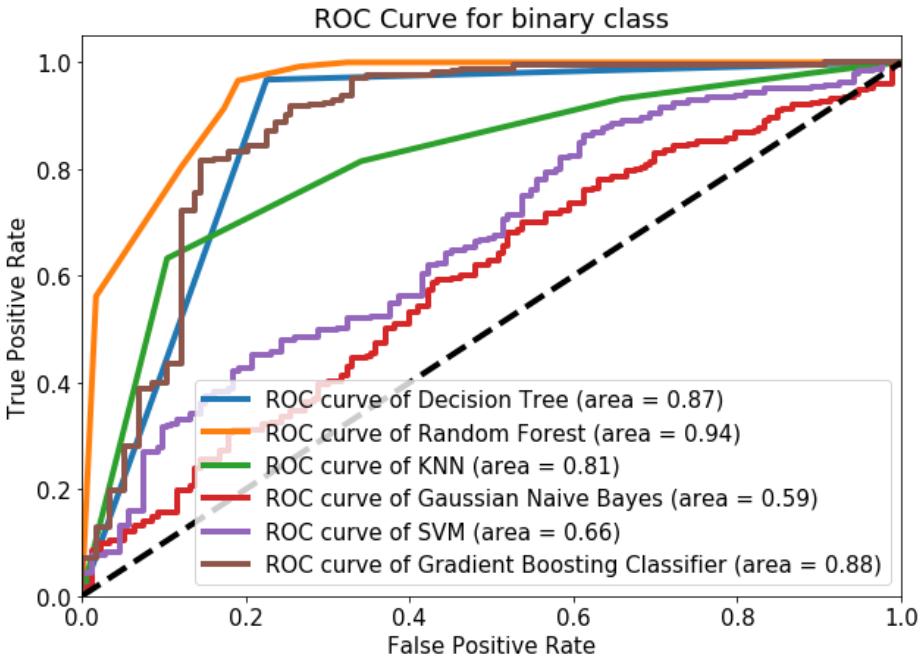


Figure 2.2.1.2 ROC curves of prediction models (departure delay prediction with weather data).

-Arrival delay prediction using binary class (including weather data)

In Figure 2.2.1.3 below, the 10-fold cross-validation results of prediction models during training is shown. Decision Tree, Random Forest, and Gradient Boosting classifiers preform the best. KNN and SVM yield acceptable results during cross-validation. Gaussian Naïve Bayes classifier performs the worst because none of our attributes follow Gaussian distribution. (Other Naïve Bayes classifiers like complement Naiive Bayes are tested but results are also unacceptable).

```
Accuracy during training:

Decision Tree: 0.997384 (0.003997)
Random Forest: 0.997376 (0.004008)
KNN: 0.981693 (0.016797)
Gaussian Naive Bayes: 0.650633 (0.042068)
SVM: 0.980839 (0.016436)
Gradient Boosting Classifier: 0.999130 (0.002609)
```

Figure 2.2.1.3 Cross-validation results of prediction models (arrival delay prediction with weather data).

From testing results shown in Table 2.2.1.2 below, the confusion matrices and accuracy reports are useful to rank the classifiers. It is worth to notice that the accuracy score is not useful here because we need to keep cost in mind because most of the flights in test dataset are not delayed here. Thus, though some models like SVM and KNN predict most on time flights correctly, they do not predict 8 delayed flights correctly. Random Forest and Gradient Boosting classifiers predict all flights accurately, while Decision Tree classifier perform relatively worse by mis-classifying 3 delayed flights as on time.

Test results using Decision Tree is :

Accuracy score: 0.9939148073022313

Confusion Matrix:
[[482 3]
[0 8]]

	precision	recall	f1-score	support
0.0	1.00	0.99	1.00	485
1.0	0.73	1.00	0.84	8
micro avg	0.99	0.99	0.99	493
macro avg	0.86	1.00	0.92	493
weighted avg	1.00	0.99	0.99	493

Decision Tree

Test results using Random Forest is :

Accuracy score: 1.0

Confusion Matrix:
[[485 0]
[0 8]]

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	485
1.0	1.00	1.00	1.00	8
micro avg	1.00	1.00	1.00	493
macro avg	1.00	1.00	1.00	493
weighted avg	1.00	1.00	1.00	493

Random Forest

Test results using KNN is :

Accuracy score: 0.9817444219066938

Confusion Matrix:
[[483 2]
[7 1]]

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	485
1.0	0.33	0.12	0.18	8
micro avg	0.98	0.98	0.98	493
macro avg	0.66	0.56	0.59	493
weighted avg	0.98	0.98	0.98	493

KNN

Test results using Gaussian Naive Bayes is :

Accuracy score: 0.6064908722109533

Confusion Matrix:
[[291 194]
[0 8]]

	precision	recall	f1-score	support
0.0	1.00	0.60	0.75	485
1.0	0.04	1.00	0.08	8
micro avg	0.61	0.61	0.61	493
macro avg	0.52	0.80	0.41	493
weighted avg	0.98	0.61	0.74	493

Gaussian Naïve Bayes

Test results using SVM is :

Accuracy score: 0.9837728194726166

Confusion Matrix:
[[485 0]
[8 0]]

	precision	recall	f1-score	support
0.0	0.98	1.00	0.99	485
1.0	0.00	0.00	0.00	8
micro avg	0.98	0.98	0.98	493
macro avg	0.49	0.50	0.50	493
weighted avg	0.97	0.98	0.98	493

SVM

Test results using Gradient Boosting Classifier is

Accuracy score: 1.0

Confusion Matrix:
[[485 0]
[0 8]]

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	485
1.0	1.00	1.00	1.00	8
micro avg	1.00	1.00	1.00	493
macro avg	1.00	1.00	1.00	493
weighted avg	1.00	1.00	1.00	493

Gradient Boosting Classifier

Table 2.2.1.2 Testing results of prediction models (arrival delay prediction with weather data)

The ROC curves of prediction models is shown in Figure 2.2.1.4 below. If the accuracy of these models are ranked only by area, Gradient Boosting is the best while SVM becomes the second place. However, combining the confusion matrices above, it is not hard to see that SVM is not suitable for flight delay prediction. Though Gradient Boosting classifier performs well on predicting arrival delay here, it is not the best while predicting departure delay (discussed above). Thus, for both departure and arrival delay, Random Forest classifier is the first choice while all desired attributes including weather data are available.

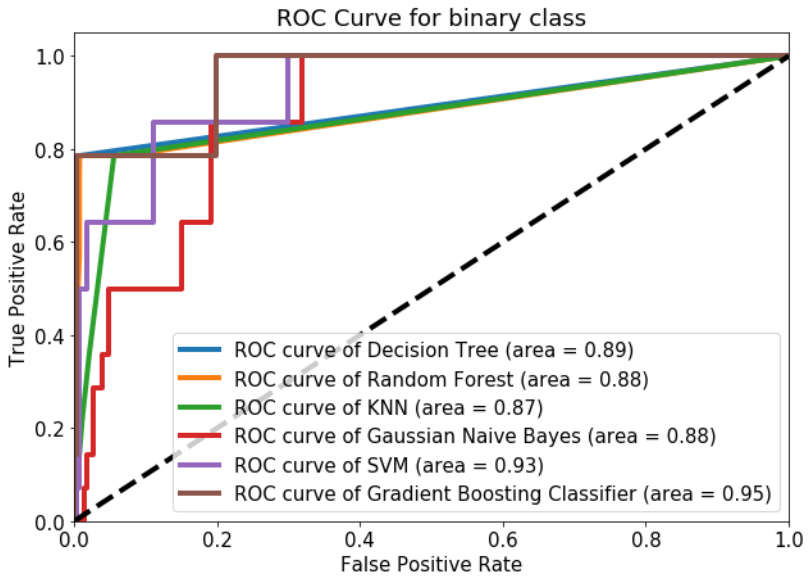


Figure 2.2.1.4 ROC curves of prediction models (arrival delay prediction with weather data).

-Departure delay prediction using binary class (without weather data)

```
Accuracy during training:

Decision Tree: 0.962614 (0.006503)
Random Forest: 0.960263 (0.008065)
KNN: 0.800695 (0.017497)
Gaussian Naive Bayes: 0.772964 (0.011310)
SVM: 0.782866 (0.012963)
Gradient Boosting Classifier: 0.803170 (0.012854)
```

Figure 2.2.1.5 Cross-validation results of prediction models (departure delay prediction without weather data)

While weather data are missing, the classifier models are perform relatively worse compared to their performance while weather data are provided. However, Decision Tree and Random Forest are still reliable while 10-fold cross-validation during training as well as testing shown in Figure 2.2.1.5 and Table 2.2.1.3 below. The other four classifiers including Gradient Boosting, are not reliable though they may still have high accuracy score around 0.8. The confusion matrices show their disability to predict departure delay while weather data are not provided.

Test results using Decision Tree is :

Accuracy score: 0.9711233034940803

Confusion Matrix:
[[670 51]
[49 2693]]

	precision	recall	f1-score	support
0.0	0.93	0.93	0.93	721
1.0	0.98	0.98	0.98	2742
micro avg	0.97	0.97	0.97	3463
macro avg	0.96	0.96	0.96	3463
weighted avg	0.97	0.97	0.97	3463

Decision Tree

Test results using Random Forest is :

Accuracy score: 0.9717008374241987

Confusion Matrix:
[[664 57]
[41 2701]]

	precision	recall	f1-score	support
0.0	0.94	0.92	0.93	721
1.0	0.98	0.99	0.98	2742
micro avg	0.97	0.97	0.97	3463
macro avg	0.96	0.95	0.96	3463
weighted avg	0.97	0.97	0.97	3463

Random Forest

Test results using KNN is :

Accuracy score: 0.8071036673404562

Confusion Matrix:
[[330 391]
[277 2465]]

	precision	recall	f1-score	support
0.0	0.54	0.46	0.50	721
1.0	0.86	0.90	0.88	2742
micro avg	0.81	0.81	0.81	3463
macro avg	0.70	0.68	0.69	3463
weighted avg	0.80	0.81	0.80	3463

KNN

Test results using Gaussian Naive Bayes is :

Accuracy score: 0.7837135431706613

Confusion Matrix:
[[64 657]
[92 2650]]

	precision	recall	f1-score	support
0.0	0.41	0.09	0.15	721
1.0	0.80	0.97	0.88	2742
micro avg	0.78	0.78	0.78	3463
macro avg	0.61	0.53	0.51	3463
weighted avg	0.72	0.78	0.72	3463

Gaussian Naïve Bayes

Test results using SVM is :

Accuracy score: 0.7917990181923188

Confusion Matrix:
[[0 721]
[0 2742]]

	precision	recall	f1-score	support
0.0	0.00	0.00	0.00	721
1.0	0.79	1.00	0.88	2742
micro avg	0.79	0.79	0.79	3463
macro avg	0.40	0.50	0.44	3463
weighted avg	0.63	0.79	0.70	3463

SVM

Test results using Gradient Boosting Classifier is :

Accuracy score: 0.8149003753970546

Confusion Matrix:
[[142 579]
[62 2680]]

	precision	recall	f1-score	support
0.0	0.70	0.20	0.31	721
1.0	0.82	0.98	0.89	2742
micro avg	0.81	0.81	0.81	3463
macro avg	0.76	0.59	0.60	3463
weighted avg	0.80	0.81	0.77	3463

Gradient Boosting Classifier

Table 2.2.1.3 Testing results of prediction models (departure delay prediction without weather data).

The ROC curves for all six models are plotted in Figure 2.2.1.6 below. According to the shape of the curves and the area calculated under the curve, Random Forest and Decision Tree both performs well in predicting departure delay while weather attributes are missing.

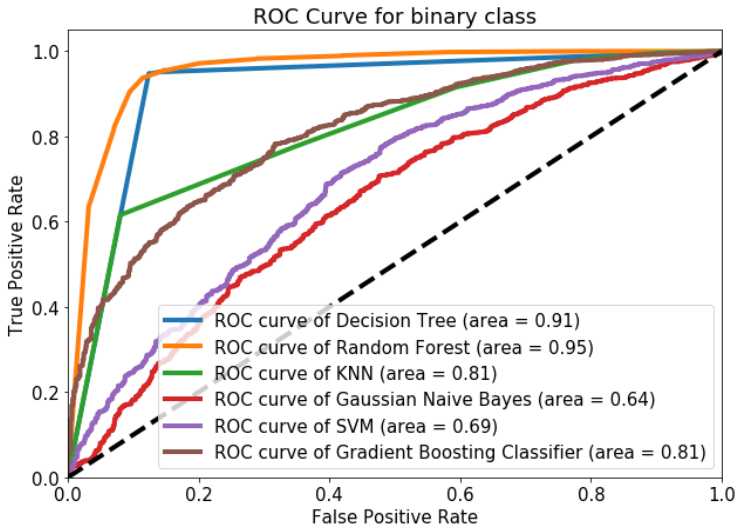


Figure 2.2.1.6 ROC curves of prediction models (departure delay prediction without weather data).

-Arrival Delay Prediction using Binary Class (without weather data)

```
Accuracy during training:

Decision Tree: 0.989354 (0.004261)
Random Forest: 0.989848 (0.004063)
KNN: 0.968927 (0.008449)
Gaussian Naive Bayes: 0.911364 (0.008044)
SVM: 0.972516 (0.007222)
Gradient Boosting Classifier: 0.976849 (0.007777)
```

Figure 2.2.1.7 Cross-validation results of prediction models (arrival delay prediction without weather data).

In Figure 2.2.1.7, all classifiers except Gaussian Naïve Bayes perform well during 10-fold cross-validation. However, this results is unreliable because of over-fit and other limitations. Combining the confusion matrices of testing in Table 2.2.1.4 below, only Decision Tree and Random Forest yield acceptable results while predicting arrival delay without weather data.

Test results using Decision Tree is :

Accuracy score: 0.9898931562229281

Confusion Matrix:
[[3333 22]
[13 95]]

	precision	recall	f1-score	support
0.0	1.00	0.99	0.99	3355
1.0	0.81	0.88	0.84	108
micro avg	0.99	0.99	0.99	3463
macro avg	0.90	0.94	0.92	3463
weighted avg	0.99	0.99	0.99	3463

Decision Tree

Test results using Random Forest is :

Accuracy score: 0.9887380883626913

Confusion Matrix:
[[3346 9]
[30 78]]

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	3355
1.0	0.90	0.72	0.80	108
micro avg	0.99	0.99	0.99	3463
macro avg	0.94	0.86	0.90	3463
weighted avg	0.99	0.99	0.99	3463

Random Forest

Test results using KNN is :

Accuracy score: 0.9607276927519491

Confusion Matrix:
[[3311 44]
[92 16]]

	precision	recall	f1-score	support
0.0	0.97	0.99	0.98	3355
1.0	0.27	0.15	0.19	108
micro avg	0.96	0.96	0.96	3463
macro avg	0.62	0.57	0.59	3463
weighted avg	0.95	0.96	0.96	3463

KNN

Test results using Gaussian Naive Bayes is :

Accuracy score: 0.9055732024256425

Confusion Matrix:
[[3122 233]
[94 14]]

	precision	recall	f1-score	support
0.0	0.97	0.93	0.95	3355
1.0	0.06	0.13	0.08	108
micro avg	0.91	0.91	0.91	3463
macro avg	0.51	0.53	0.51	3463
weighted avg	0.94	0.91	0.92	3463

Gaussian Naïve Bayes

Test results using SVM is :

Accuracy score: 0.9688131677736067

Confusion Matrix:
[[3355 0]
[108 0]]

	precision	recall	f1-score	support
0.0	0.97	1.00	0.98	3355
1.0	0.00	0.00	0.00	108
micro avg	0.97	0.97	0.97	3463
macro avg	0.48	0.50	0.49	3463
weighted avg	0.94	0.97	0.95	3463

SVM

Test results using Gradient Boosting Classifier is

Accuracy score: 0.9705457695639619

Confusion Matrix:
[[3346 9]
[93 15]]

	precision	recall	f1-score	support
0.0	0.97	1.00	0.98	3355
1.0	0.62	0.14	0.23	108
micro avg	0.97	0.97	0.97	3463
macro avg	0.80	0.57	0.61	3463
weighted avg	0.96	0.97	0.96	3463

Gradient Boosting Classifier

Table 2.2.1.4 Testing results of prediction models (arrival delay prediction without weather data).

In Figure 2.2.1.8 below, the shape of curve and area under curves below show that Decision Tree, Random Forest, KNN, and Gradient Boosting all yield acceptable result. However, combining the confusion matrices above, we conclude that only Decision Tree and Random Forest are useful to predict arrival delay while weather data are missing.

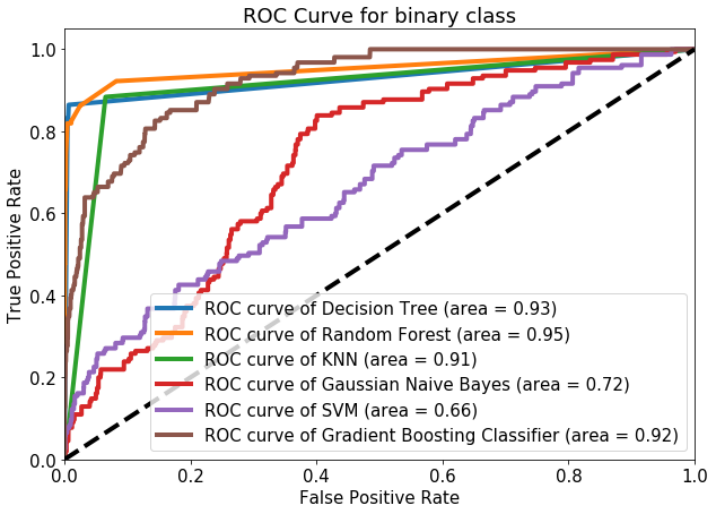


Figure 2.2.1.8 ROC curves of prediction models (arrival delay prediction without weather data).

2.2.2 Predict delay with multi-type class (Delay Level)

In Section 2.2.1 above, all results show that Random Forest and Decision Tree are two useful classifiers for predicting departure and arrival delay no matter if the weather data are available or not. Thus, in this section, we divide delay to four levels: 0. Early flights, 1. Slightly delay (within 10 min), 2. Medium delay (Under 30 min), 3. Long delay (Longer than 30min). Random Forest and Decision Tree classifiers are used for performing this multi-type classification.

-Departure delay prediction with multi-type class (including weather data)

The testing results including accuracy scores, confusion matrices, and reports are shown in Table 2.2.2.1 below, and multi-type class ROC curves are shown in Figure 2.2.2.1 below. According to these results and curves, it is clear to conclude that both Decision Tree and Random Forest classifiers perform well in predicting departure delay while weather data are completely available. Decision Tree performs even better than Random Forest under this situation.

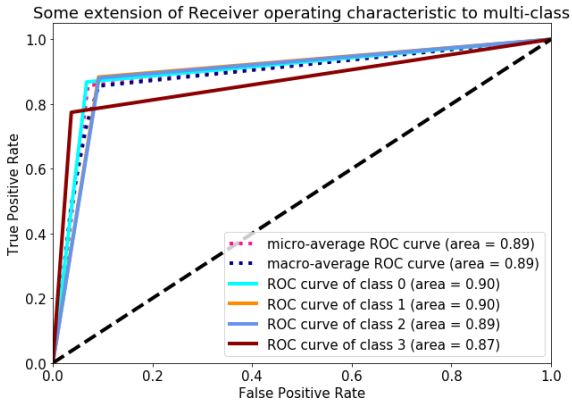
```
Test results using  Decision Tree  is :


Accuracy score:  0.9726443768996961

Confusion Matrix:
 [[ 67   0   3   0]
 [  0 118   0   0]
 [  3   0 100   0]
 [  0   0   3  35]]
              precision    recall  f1-score   support

         0.0       0.96      0.96      0.96        70
         1.0       1.00      1.00      1.00       118
         2.0       0.94      0.97      0.96       103
         3.0       1.00      0.92      0.96        38

   micro avg       0.97      0.97      0.97       329
   macro avg       0.98      0.96      0.97       329
weighted avg       0.97      0.97      0.97       329
```

Decision Tree

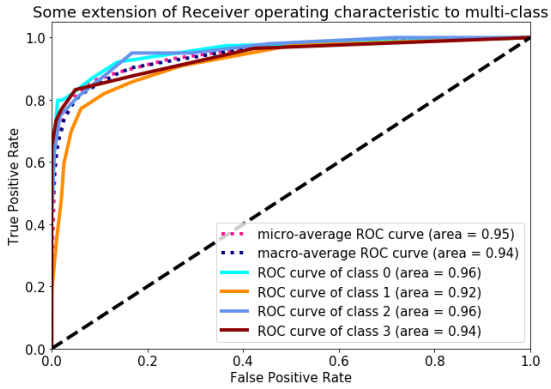
```
Test results using  Random Forest  is :


Accuracy score:  0.9665653495440729

Confusion Matrix:
 [[ 70   0   0   0]
 [  0 115   3   0]
 [  2   0  98   3]
 [  0   3   0  35]]
              precision    recall  f1-score   support

         0.0       0.97      1.00      0.99        70
         1.0       0.97      0.97      0.97       118
         2.0       0.97      0.95      0.96       103
         3.0       0.92      0.92      0.92        38

   micro avg       0.97      0.97      0.97       329
   macro avg       0.96      0.96      0.96       329
weighted avg       0.97      0.97      0.97       329
```

Random Forest

Table 2.2.2.1 Testing results of Decision Tree and Random Forest classifier (departure delay prediction with weather data).



Decision Tree



Random Forest

Figure2.2.2.1 ROC curves of multi-type class (departure delay prediction with weather data).

-Arrival delay prediction with multi-type class (including weather data)

The testing results including accuracy scores, confusion matrices, and reports are shown in Table 2.2.2.2 below, and multi-type class ROC curves are shown in Figure 2.2.2.2 below. According to these results and curves, it is clear to conclude that both Decision Tree and Random Forest classifiers perform well in predicting departure delay while weather data are completely available. Though Random Forest classifier's ROC curves shown some advantage compared to the curves of Decision Tree, their confusion matrices show that they perform the same in predicting arrival delay level.

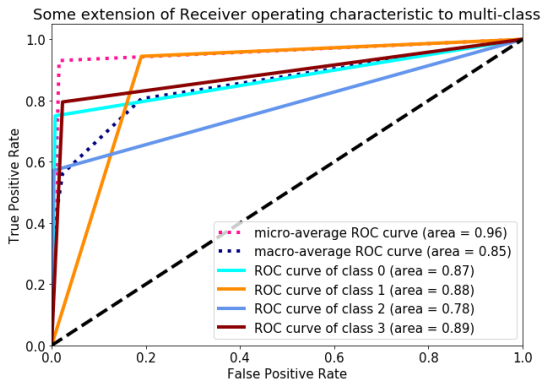
```
Test results using  Decision Tree  is :


Accuracy score:  0.9908814589665653

Confusion Matrix:
 [[ 16   0   0   0]
 [  0   1   0   0]
 [  0   3 303   0]
 [  0   0   0   6]]
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00        16
         1.0       0.25      1.00      0.40         1
         2.0       1.00      0.99      1.00       306
         3.0       1.00      1.00      1.00         6

   micro avg       0.99      0.99      0.99       329
   macro avg       0.81      1.00      0.85       329
weighted avg       1.00      0.99      0.99       329
```

Decision Tree

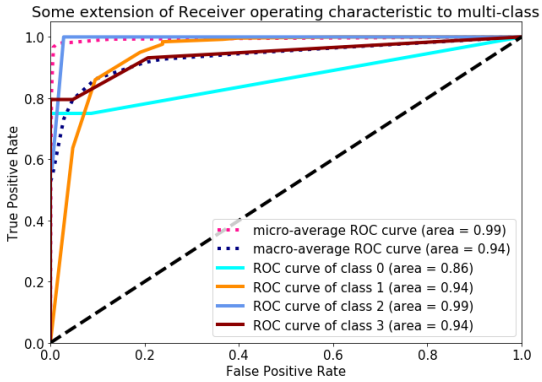
```
Test results using  Random Forest  is :


Accuracy score:  0.9908814589665653

Confusion Matrix:
 [[ 16   0   0   0]
 [  0   1   0   0]
 [  3   0 303   0]
 [  0   0   0   6]]
              precision    recall  f1-score   support

         0.0       0.84      1.00      0.91        16
         1.0       1.00      1.00      1.00         1
         2.0       1.00      0.99      1.00       306
         3.0       1.00      1.00      1.00         6

   micro avg       0.99      0.99      0.99       329
   macro avg       0.96      1.00      0.98       329
weighted avg       0.99      0.99      0.99       329
```

Random Forest

Table 2.2.2.2 Testing results of Decision Tree and Random Forest classifier (arrival delay prediction with weather data).



Decision Tree



Random Forest

Figure 2.2.2.2 ROC curves of multi-type class (arrival delay prediction with weather data).

-Departure delay prediction with multi-type class (without weather data)

The testing results including accuracy scores, confusion matrices, and reports are shown in Table 2.2.2.3 below, and multi-type class ROC curves are shown in Figure 2.2.2.3 below. According to these results and curves, while weather data are missing, both classifiers perform slightly worse compared their performance while full weather data attributes are available. However, there is no doubt that both Decision Tree and Random Forest classifiers still perform well, and the prediction results are acceptable.

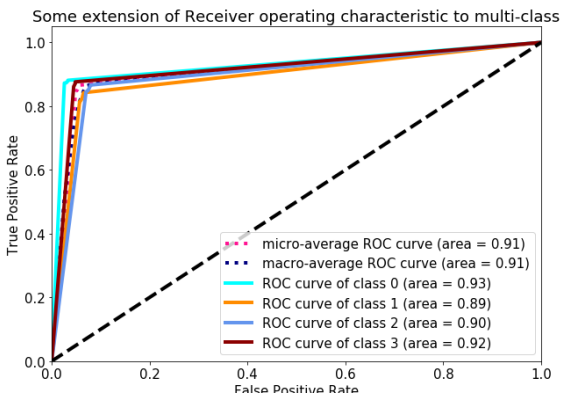
```
Test results using  Decision Tree  is :


Accuracy score:  0.9631875270679948

Confusion Matrix:
 [[463   9   7   3]
 [  6 668  25   0]
 [  6   9 565   4]
 [  3   4   9 528]]
              precision    recall  f1-score   support

         0.0       0.97      0.96      0.96       482
         1.0       0.97      0.96      0.96       699
         2.0       0.93      0.97      0.95       584
         3.0       0.99      0.97      0.98       544

   micro avg       0.96      0.96      0.96      2309
   macro avg       0.96      0.96      0.96      2309
weighted avg       0.96      0.96      0.96      2309
```

Decision Tree

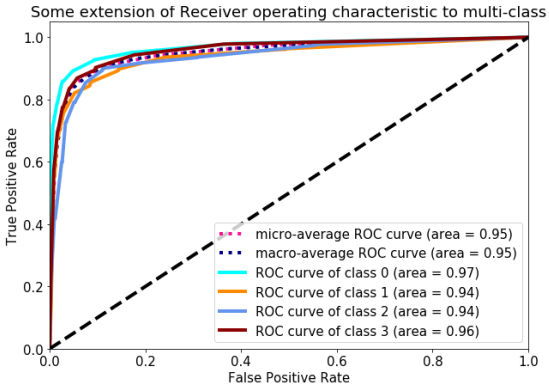
```
Test results using  Random Forest  is :


Accuracy score:  0.9558250324815938

Confusion Matrix:
 [[465  10   4   3]
 [  4 668  23   4]
 [  6  18 551   9]
 [  3   3  15 523]]
              precision    recall  f1-score   support

         0.0       0.97      0.96      0.97       482
         1.0       0.96      0.96      0.96       699
         2.0       0.93      0.94      0.94       584
         3.0       0.97      0.96      0.97       544

   micro avg       0.96      0.96      0.96      2309
   macro avg       0.96      0.96      0.96      2309
weighted avg       0.96      0.96      0.96      2309
```

Random Forest

Table 2.2.2.3 Testing results of Decision Tree and Random Forest classifier (departure delay prediction without weather data).



Decision Tree



Random Forest

Figure 2.2.2.3 ROC curves of multi-type class (departure delay prediction without weather data).

-Arrival delay prediction with multi-type class (without weather data)

The testing results including accuracy scores, confusion matrices, and reports are shown in Table 2.2.2.4 below, and multi-type class ROC curves are shown in Figure 2.2.2.4 below. According to these results and curves, it is clear to conclude that both Decision Tree and Random Forest classifiers yield acceptable results while predicting arrival delay level while weather data are missing. However, different than three situations above, Decision Tree shows obvious robust accuracy while predicting class 1 short delay than Random Forest.

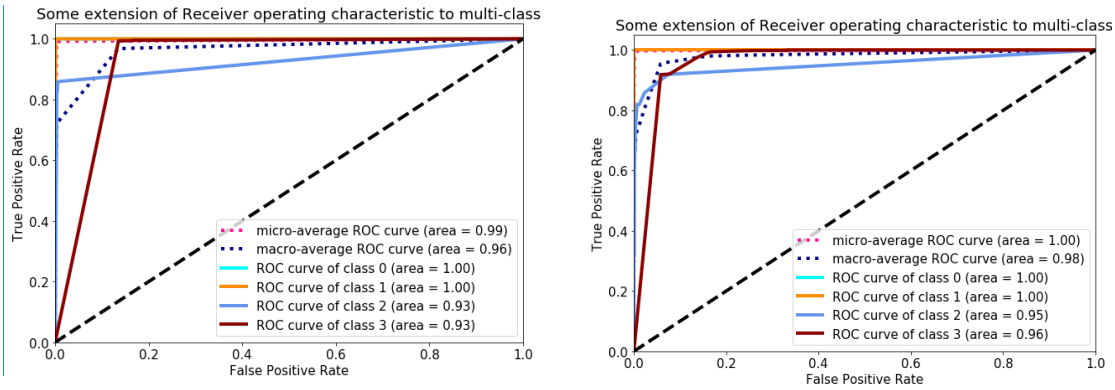
```
Test results using  Decision Tree  is :


Accuracy score:  0.9939367691641403

Confusion Matrix:
 [[2228    7    0    0]
 [   4   63    0    0]
 [   3    0    0    0]
 [   0    0    0    4]]
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00      2235
         1.0       0.90      0.94      0.92        67
         2.0       0.00      0.00      0.00         3
         3.0       1.00      1.00      1.00         4

   micro avg       0.99      0.99      0.99      2309
   macro avg       0.72      0.73      0.73      2309
weighted avg       0.99      0.99      0.99      2309
```

Decision Tree

```
Test results using  Random Forest  is :


Accuracy score:  0.989172802078822

Confusion Matrix:
 [[2233    2    0    0]
 [  20   47    0    0]
 [   3    0    0    0]
 [   0    0    0    4]]
              precision    recall  f1-score   support

         0.0       0.99      1.00      0.99      2235
         1.0       0.96      0.70      0.81        67
         2.0       0.00      0.00      0.00         3
         3.0       1.00      1.00      1.00         4

   micro avg       0.99      0.99      0.99      2309
   macro avg       0.74      0.68      0.70      2309
weighted avg       0.99      0.99      0.99      2309
```

Random Forest

Table 2.2.2.4 Testing results of Decision Tree and Random Forest classifier (arrival delay prediction without weather data).



Decision Tree Random Forest

Figure 2.2.2.4 ROC curves of multi-type class (arrival delay prediction without weather data).

2.2.3 Logistic Regression

In order to rank and select from all the features, recursive feature elimination (RFE), which works by recursively removing attributes and building a model on those attributes that remain, is implemented. In another word, RFE uses the model accuracy to identify which feature contributes the most to predicting the target class. Finally, based on RFE test results, the ranking of attributes for departure and arrival delay are shown below in Table 2.2.3.1 and 2.2.3.2. From Table 2.2.3.1, for departure delay, the weather attributes are highly ranked, and airline is the second important attribute that delay. For arrival delay shown in Table 2.2.3.2, it is expected that origin and destination are important, but air manufacturer becomes the first place that affecting arrival delay. Though this observation may has some limitations because our data like aircraft manufacturer are categorical and has been normalized for logistic regression. We still draw some useful conclusion from the RFE logistic regression. The logistic regression results shown in Figure 2.2.3.1 confirms the conclusions from ranking attributes by RFE. The attributes x1 to x10 in the logistic regression report in Figure 2.2.3.1 are the ten corresponding flight attributes in Table 2.2.3.1 and 2.2.3.2.

Flight Attributes	Rank	Weather Attributes	Rank
Aircraft Type	7	Cloud Altitude	4
Planned Cruise Speed	12	Temperature	9
Planned Cruise Altitude	14	Dew Point	3
Origin	5	Visibility	10
Destination	15	Wind Speed	1
Departure Week Day	16	Wind Gust Speed	8
Departure Time Hour	6		
Arrival Time Hour	13		
Airline	2		
Aircraft Manufacturer	11		

Table 2.2.3.1 Attributes ranking for departure delay

Flight Attributes	Rank	Weather Attributes	Rank
Aircraft Type	9	Cloud Altitude	6
Planned Cruise Speed	5	Temperature	13
Planned Cruise Altitude	7	Dew Point	11
Origin	3	Visibility	2
Destination	4	Wind Speed	12
Departure Week Day	14	Wind Gust Speed	10
Departure Time Hour	8		
Arrival Time Hour	16		
Airline	15		
Aircraft Manufacturer	1		

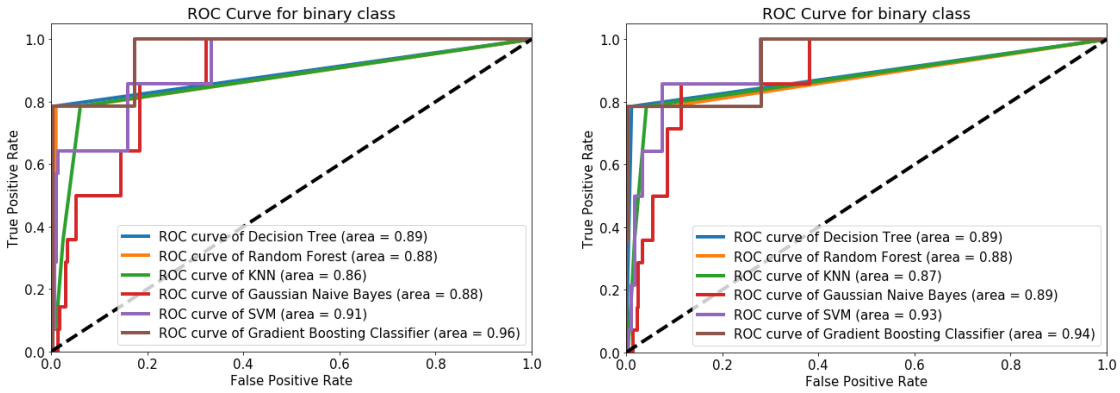
Table 2.2.3.2 Attributes ranking for arrival delay.

```
Optimization terminated successfully.
         Current function value: 0.123059
         Iterations 8
                          Results: Logit
================================================================
Model:              Logit            No. Iterations:   8.0000
Dependent Variable: y                Pseudo R-squared: 0.052
Date:               2018-11-11 19:39 AIC:              2860.4506
No. Observations:   11541            BIC:              2933.9872
Df Model:           9                Log-Likelihood:   -1420.2
Df Residuals:       11531            LL-Null:          -1498.2
Converged:          1.0000           Scale:            1.0000
----------------------------------------------------------------
          Coef.    Std.Err.      z    |  P>|z|     [0.025    0.975]
----------------------------------------------------------------
x1       -0.7290     0.4103   -1.7770   0.0756   -1.5331    0.0750
x2       -5.0511     0.5209   -9.6963   0.0000   -6.0721   -4.0301
x3        0.9186     0.3884    2.3649   0.0180    0.1573    1.6799
x4       -2.8985     0.4925   -5.8850   0.0000   -3.8638   -1.9332
x5        1.0983     0.2579    4.2580   0.0000    0.5927    1.6038
x6       -0.3726     0.1772   -2.1026   0.0355   -0.7198   -0.0253
x7       -0.5372     0.4315   -1.2448   0.2132   -1.3829    0.3086
x8        0.1883     0.2450    0.7682   0.4423   -0.2920    0.6685
x9        1.5474     0.3014    5.1338   0.0000    0.9566    2.1382
x10       1.2498     0.3384    3.6938   0.0002    0.5867    1.9130
================================================================
```

Figure 2.2.3.1 Logistic Regression Result.

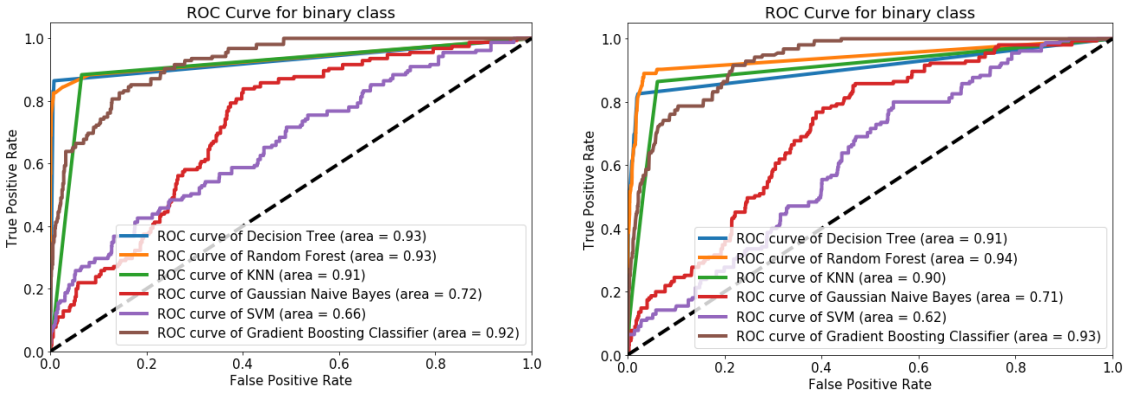
Another point to research for logistic regression is that after selecting attributes based on logistic regression results, how will classifiers perform. Thus, we show ROC curves (predicting arrival delay) of all classifiers in Figure 2.2.3.2 and 2.2.3.3 for comparison before and after feature elimination. Eight

attributes are kept for data with weather and six attributes are kept for data without weather attributes. Both sets of figures show that the feature elimination does not affect the classifiers performance.



Before Feature Elimination Selecting 8 attributes

Figure 2.2.3.2 ROC curves of classifiers with weather data for arrival delay.



Before Feature Elimination Selecting 6 attributes

Figure 2.2.3.3 ROC curves of classifiers without weather data for arrival delay.

2.2.4 Conclusion on prediction models

In this section, prediction models are implemented on our data set. Both binary class and multi-type class to be predicted have been applied. Situations that weather data are missing is also researched. According to all the results, we can draw the conclusion that for flight delay prediction, if all attributes listed above including weather attributes are provided, Random Forest is the first choice. If weather attributes are missing, both Random Forest and Decision Tree perform well. However, Decision Tree is preferred based on its advantage on accurately predict delay level.