Project 4 - Anomaly Detection and Outlier Analysis

Celine Chiou(Yun-Chyau, Chiou) 300230747, Xinyu Xie 0300239933, Deonne Millaire 300260913

December 10, 2021

Introduction

Nowadays,the aviation industry is a rapidly growing sector, there is a huge amount of data in this field used for forecasting or model predicting. One of the most recurrent problems in predictive modeling tasks is the presence of noise in the data. Some of the observations are not reliable and often are generated by different patterns. In this report, we try to identify the outliers/anomalies by using different methods and evaluate their performances.

Initial Exploratory Data Analysis

Data Visualization and Observations Appear to be Anomalous or Outlying

The original dataset contains 583985 observations for flights which departed in January 2019 with 18 variables in total, and contains information about the landing and take-off of each flight, for example, the departure date, departure city and airport, destination city and airport. . . ect. We can safely remove any observations with missing values, since the size of the dataset is really large. Moreover, we remove the last attribute called X since the last one is uninformative.

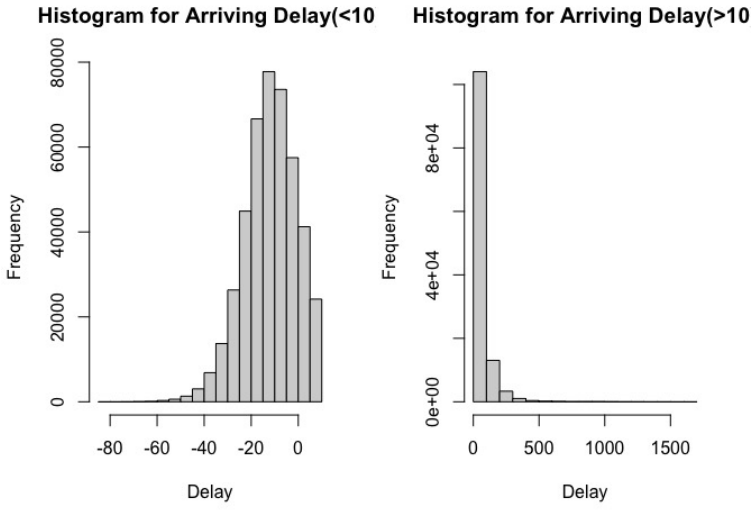

Figure 1: The distribution of arrival delay

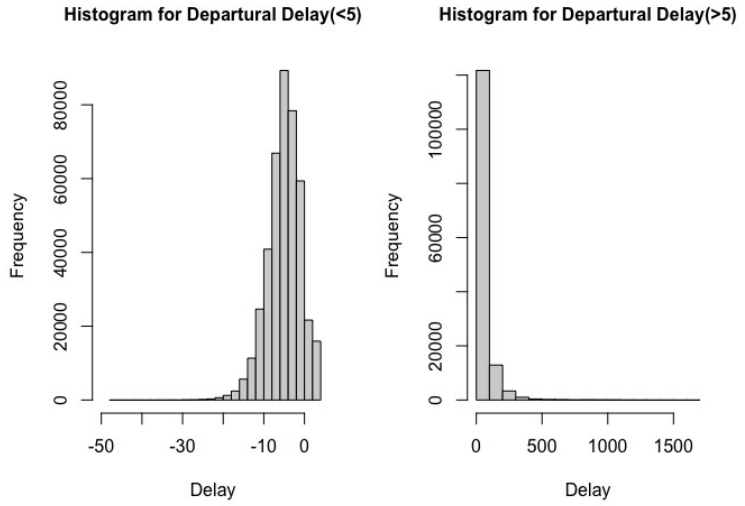

Figure 2: The distribution of departure delay

Figure 1 and Figure 2 The distributions for departure and arriving delay are really right skewed, which are difficult for us to interpret by using one histogram, so we decided to separate our histogram into two parts for both arriving delay and departure delay. To have further understanding of distribution of arrival delay and departure delay, we created the summary table that represents descriptive statistics for these two variables. The overall means are 4.26 and 9.68 for arriving delay and departure delay respectively. When we compared the average delay time between different day of week, we found that the values are largest in Thursday(6.8 min for ARR_DELAY and 11.6 min for DEP_DELAY)

	1 (N=76805)	2 (N=89682)	3 (N=93343)	4 (N=96473)	5 (N=78654)	6 (N=60146)	7 (N=70860)	Overall (N=565963)
ARR_DELAY								
Mean (SD)	5.51 (56.9)	2.44 (50.2)	5.05 (49.8)	6.79 (50.4)	3.08 (44.2)	1.68 (53.1)	4.19 (53.9)	4.26 (51.2)
Median [Min, Max]	-7.00 [-75.0, 1380]	-8.00 [-85.0, 1450]	-6.00 [-76.0, 1510]	-5.00 [-70.0, 1390]	-6.00 [-74.0, 1610]	-8.00 [-81.0, 1640]	-7.00 [-85.0, 1500]	-7.00 [-85.0, 1640]
DEP_DELAY								
Mean (SD)	11.2 (53.8)	8.35 (47.8)	10.0 (47.2)	11.6 (47.9)	8.45 (41.7)	7.71 (50.0)	9.74 (50.7)	9.68 (48.4)
Median [Min, Max]	-3.00 [-35.0, 1380]	-3.00 [-44.0, 1430]	-3.00 [-38.0, 1500]	-2.00 [-41.0, 1390]	-2.00 [-47.0, 1640]	-3.00 [-47.0, 1650]	-2.00 [-34.0, 1490]	-3.00 [-47.0, 1650]

Figure 3: Summary table of statistics for dep_delay and arr_delay

Figure 3 To have further understanding of the distribution of arrival delay and departure delay, we create plots above to represent the delay rate on each day. The second one shows the rate for delay which is more than 15 min. They are extremely similar to each other. There are no clear trend shown in these two graphs throughout the whole period of January. We reached lowest delay rate 0.22 in Jan 08 and have relatively high delay rate in Jan 2, Jan 21, Jan23 and Jan 24. Both extreme weather conditions and holiday periods could cause the delay.

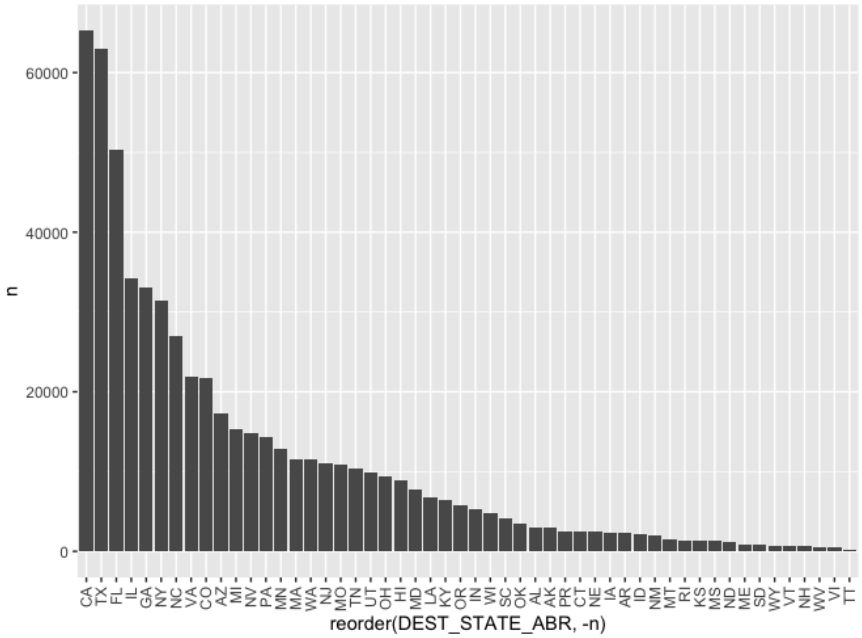

Figure 4: The distribution of destination stste

Figure 4 The whole dataset contains observations from 52 states, 11.24% observations have California as the destination state and 10.96% observations whose destination state is Texas and 8.73% for Florida. The rest of the states whose percentage are all below 6%. There are more observations in coastal states.

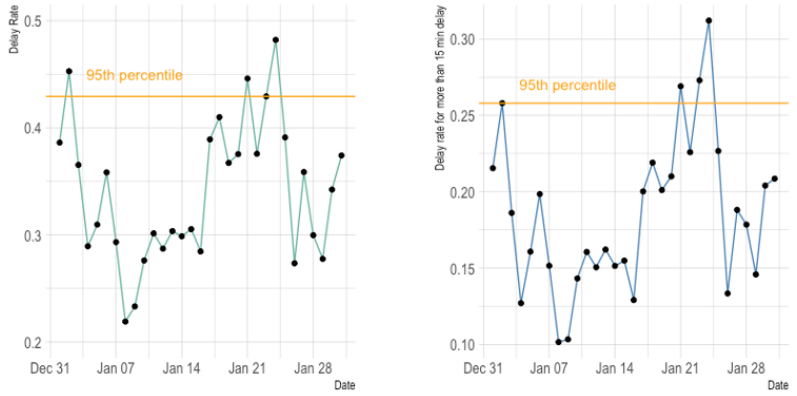

Figure 5: The distribution of destination stste

Figure 5 Plots above represent the delay rate on each day. The second one shows the rate for delay which is more than 15 min. They are extremely similar to each other. There is no clear trend shown in these two graphs throughout the whole period of January. We reached lowest delay rate 0.22 in Jan 08 and have relatively high delay rate in Jan 2, Jan 21, Jan23 and Jan 24. Both extreme weather conditions and holiday period could cause the delay.

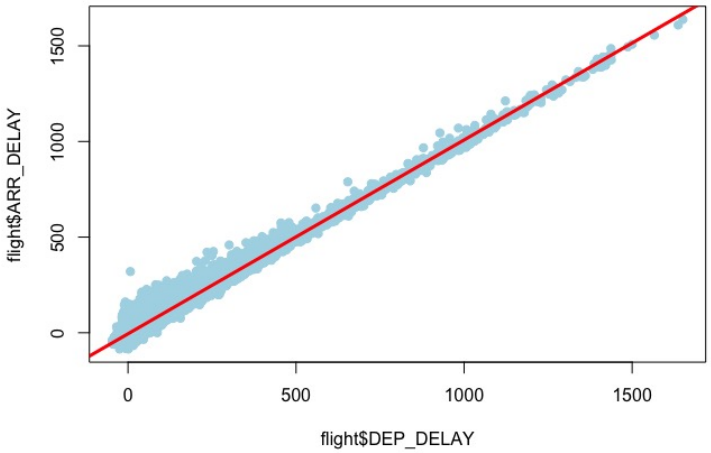

Figure 6: Arrival dely verses departure delay

Figure 6 According to the plots, we are interested in the correlation of APR_DELAY and DEP_DELAY. Furthermore, we could observe that the observations are getting less while the delay time is getting longer. To be more specific, this might mean that while predicting the outliers, the plots located in the long-delay time area would probably be the group of outliers.

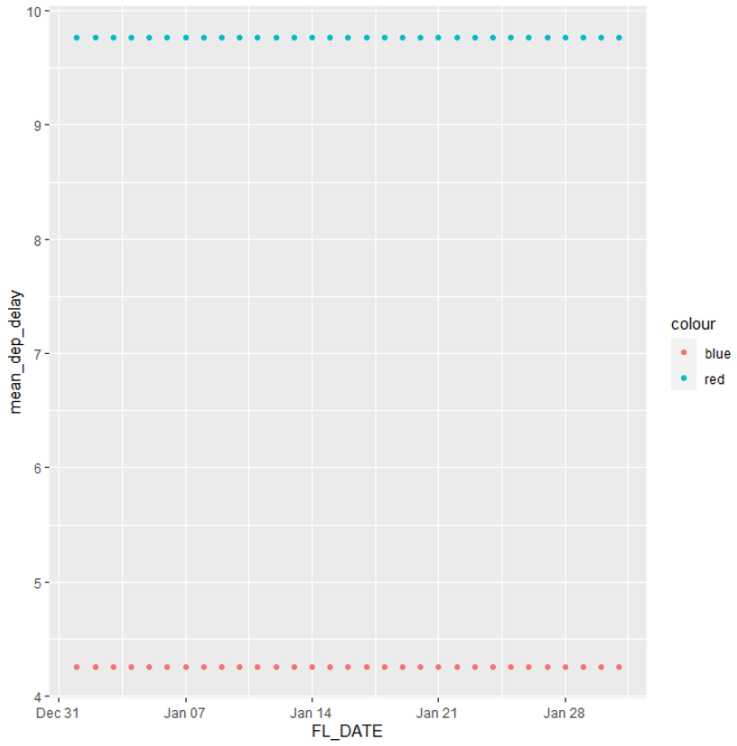

Figure 7: Flight date verses mean departure delay

Figure 7 The mean arrival delays and departure delays appear uniform throughout every day in January.

No specific day pops out as needing further investigating, however, this may not be the case for every single airport. However, if there was something such as a snow storm in the northern states, or in a very busy airport, we would expect a bit more movement from the means.

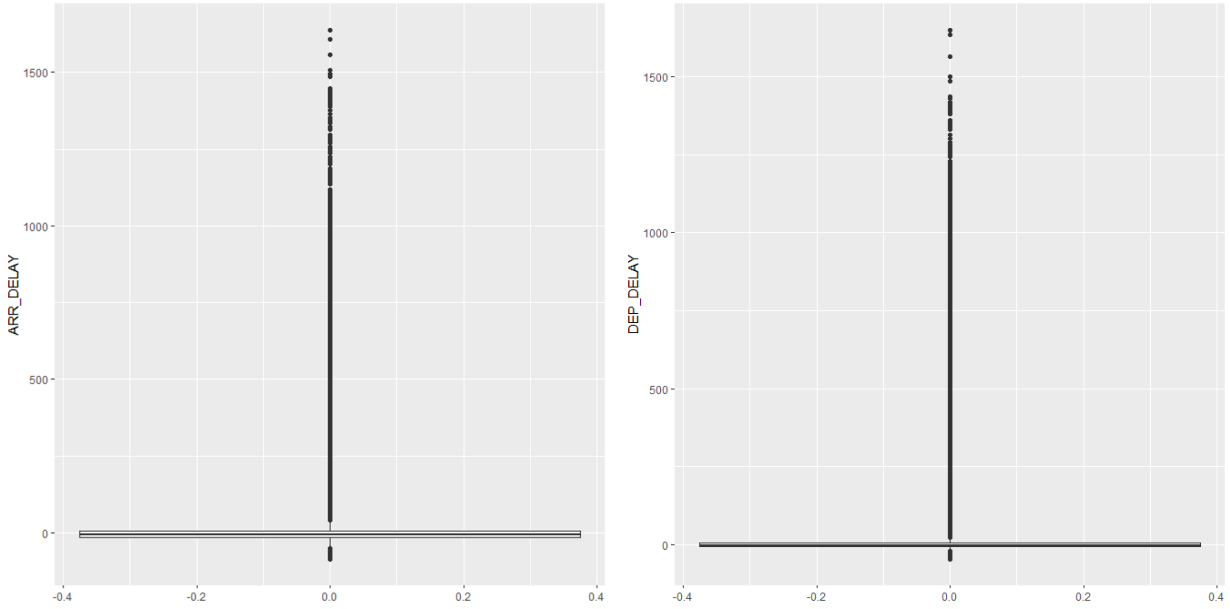

Figure 8: DBSCAN cluster plots

Figure 8 We noticed that the distributions for continuous variables are really right-skewed, but we cannot conclude that the points beyond the outer fences as outliers by simply using boxplot, since these large values might make sense under some certain scenario. Working on a transformed scale might be useful since we have a large range for the delay time. Observations that are identified as outliers might be in line with the rest of the data. Because of the large dataset, we cannot identify the potential outliers for categorical outliers based on the data visualization, we will identify them in later sections.

Dimension Reduction

Since our main goal is to identify the anomalies, choosing features beforehand might result in loss of information at the stage of outlier detection. After outlier detection, feature extraction/selection can be done in the last part of data pre-processing just before the model fitting. Moreover, some feature selection algorithms like PCA are really sensitive to the outliers, so it's essential for us to remove the outliers before applying such feature selection algorithms beforehand. Instead of choosing the subset of the original dataset, we decided to choose different subsets for different anomaly detection algorithms.

```
sum(flights$ARR_DELAY<0, na.rm = T)
```

```
## [1] 362339
```

```
sum(flights$ARR_DELAY>0, na.rm = T)
```

```
## [1] 193194
```

Since 362339 observations have an arrival delay less than 0, which is more than half of the data set, we felt it was better to use the original arrival delay in our outlier detection in order to spot outliers under these conditions.

Anomaly Detection Algorithms

Average Distance to kNN

The K-Nearest neighbors determine whether or not a point lies in a sparse region of the feature space by computing the average of the distances to the k- nearest neighbors of the point. We refer to this quantity as the kNN score for a point. Intuitively, the points in dense regions will have many points near them and will have a small kNN score. We might assume that observations that are much far from other observations are more likely to be outliers, in other words, increasing distance being increasingly suggestive of anomalous status, then the kNN score is useful for detecting outliers. To calculate the distance, we might apply different distance functions, which definity give different results. We decided to choose the default one, which is Euclidean distance. We choose k = 5.

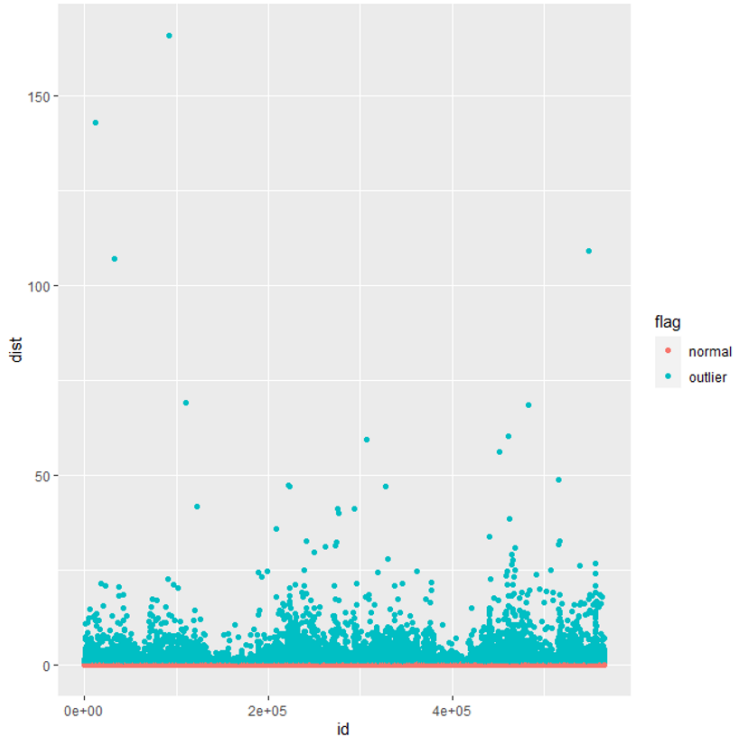

Figure 9: Distance to 5 neareat neighbor

DBSCAN

Since DBSACN cannot work with the categorical data, we are interested in the outlier between arrival delay and departure delay. This algorithm contains two main parameters: ϵ is specified as the maximum distance between two points, which is also called eps, and minPts is specified as the minimum number of neighbors within "eps" radius.

The first step for the DBSCAN is to find the ϵ and the minPts respectively. For the ϵ, we used the KNN plots to find the maximum distance. The KNN plots analyze the average distances of every point to its k nearest

neighbors. These k distances are then plotted in ascending order. When the curve occurs a sharp change in the distance, it indicates this point serves as the optimal "eps" value. According to the plots (sampling 100,000 observations), we could conclude that when $\epsilon = 0.25$ would be an optimal value for the eps.

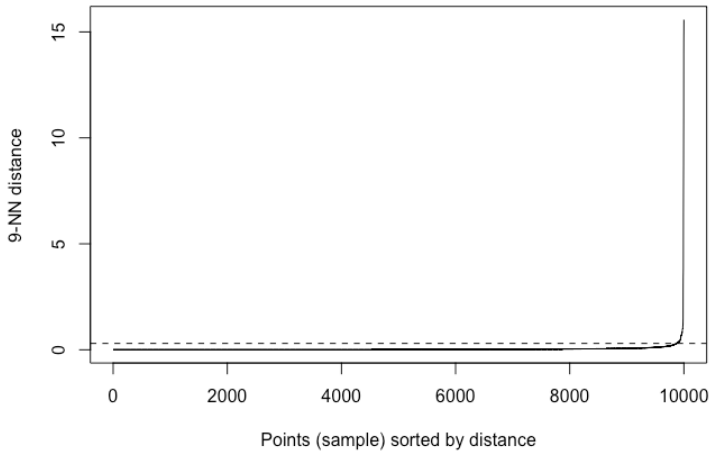

Figure 10: 9-knn plots

For the MinPts, it should be larger as the size of the dataset increases. A low minPts will result in building more clusters rom outlier, therefore we used ln(n) to decide the minimum number of neighbors within "eps" radius, where n is the total number of points to be clustered. As a result, the MinPts would be q as shown following for each 116797 observations.

```
#find the MinPts with ln()
ln(116797)
```

```
## [1] 11.66819
```

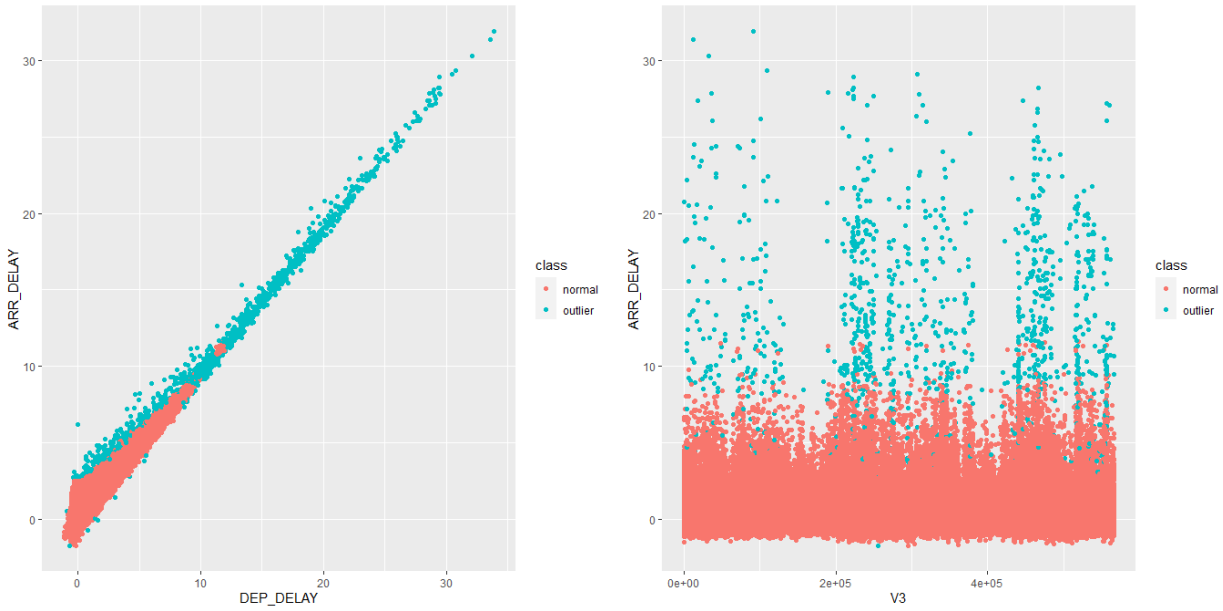

Figure 11: DBSCAN cluster plots

Figure 11 Because the dataset is too large to analyze, we first separated the observations into five piles. And after finishing analyzing for each, we combined all of the points together to get the plot above. According to figure 11, we could see that the blue plots indicate the outliers. And apparently, As we assumed before,it could not detect the points locate in the long delay time which have a large distance to each other.

Pros and Cons for DBSCAN

Pros:

- Do not need to specify how many clusters beforehand
- Only need two parameters
- Scales well and performs well in arbitrary shapes clusters.

Cons:

- Sometimes, it's hard to determine the eps without domain knowledge
- Since DESCAN only uses distance to determine, sometimes it is not well suited to define clusters.
- In our case, because some of the points are far from each other, we lose the idea of density, which means DBSCAN cannot detect the cluster on the right-hand side of the plot.

Isolation Forest

We decided to try using an Isolation Forest model in order to identify anomalies. It is also a density based method as DBSCAN was, however the philosophy of the isolation forest is that it tries to identify outliers under the assumption that there are few outliers, and that they have unique attributes compared to the rest of the data.

One benefit of the isolation forest method is that it can work with both numeric and categorical data. The problem however, is that many of our categories contain many levels. For example, there are 346 levels of

the origin airport ID. This could lead to identifying outliers solely based on coming from a less busy airport, which is fine, however we will leave this kind of outlier detection to a solely categorical based algorithm. We would like our isolation forest to attempt to balance outliers based on both the numeric values as well as more balanced categories.

With tree based methods, you cannot have observations with missing values, therefore we only consider the 565963 complete cases.

We use the Solitude package in R in order to create the isolation tree. Due to computational constraints, we set the number of trees in the forest to 500 (the default parameter is 100 trees). We then use the numeric variables departure delay and arrival delay, and the categorical factors date and destination state abbreviation.

	id	average_depth	anomaly_score
1:	270564	3.966	0.7646523
2:	243024	3.974	0.7642385
3:	377414	3.996	0.7631018
4:	3433	4.002	0.7627921
5:	269789	4.014	0.7621730

565959:	366564	8.000	0.5820092
565960:	366664	8.000	0.5820092
565961:	366695	8.000	0.5820092
565962:	366986	8.000	0.5820092
565963:	367111	8.000	0.5820092

From this, we flagged the observations with an anomaly score of over 0.74 as outliers, and below to be normal. This resulted in 290 observations being outliers according to the isolation forest we created. The summaries of the outlier observations and normal observations were as follows

Outliers:

DEP_DELAY	FL_DATE	DEST_STATE_ABR
Min. : 256.0	2019-01-18: 28	VA : 30
1st Qu.: 600.0	2019-01-21: 22	MN : 22
Median : 745.0	2019-01-16: 20	UT : 17
Mean : 784.2	2019-01-01: 18	IL : 16
3rd Qu.: 959.2	2019-01-20: 16	MO : 14
Max. :1637.0	2019-01-25: 16	TN : 13
	(Other) :170	(Other):178
ARR_DELAY	id	flag
Min. : 261.0	Min. : 479	Length:290
1st Qu.: 596.0	1st Qu.:239139	Class :character
Median : 746.0	Median :362229	Mode :character
Mean : 784.2	Mean :345282	
3rd Qu.: 958.0	3rd Qu.:476534	
Max. :1609.0	Max. :565837	

Normal:

DEP_DELAY	FL_DATE	DEST_STATE_ABR
Min. : -47.000	2019-01-02: 20166	CA : 63616
1st Qu.: -6.000	2019-01-07: 19847	TX : 62034
Median : -3.000	2019-01-11: 19824	FL : 49427
Mean : 9.283	2019-01-25: 19810	GA : 32602
3rd Qu.: 5.000	2019-01-10: 19755	IL : 30220
Max. :1651.000	2019-01-17: 19710	NY : 30060
	(Other) :446561	(Other):297714
ARR_DELAY	id	flag
Min. : -85.000	Min. : 1	Length:565673
1st Qu.: -16.000	1st Qu.:141457	Class :character
Median : -7.000	Median :282946	Mode :character
Mean : 3.858	Mean :282950	
3rd Qu.: 7.000	3rd Qu.:424415	
Max. :1638.000	Max. :565963	

From this, we can see that the outlier observations have much higher departure delay and arrival delay with the means of those being 784.2 for both vs. the non-outlier points have means of 9.2 and 3.8. However, it is interesting to note that the observation with delays of 1651 minutes was not flagged as an outlier using our anomaly score criteria.

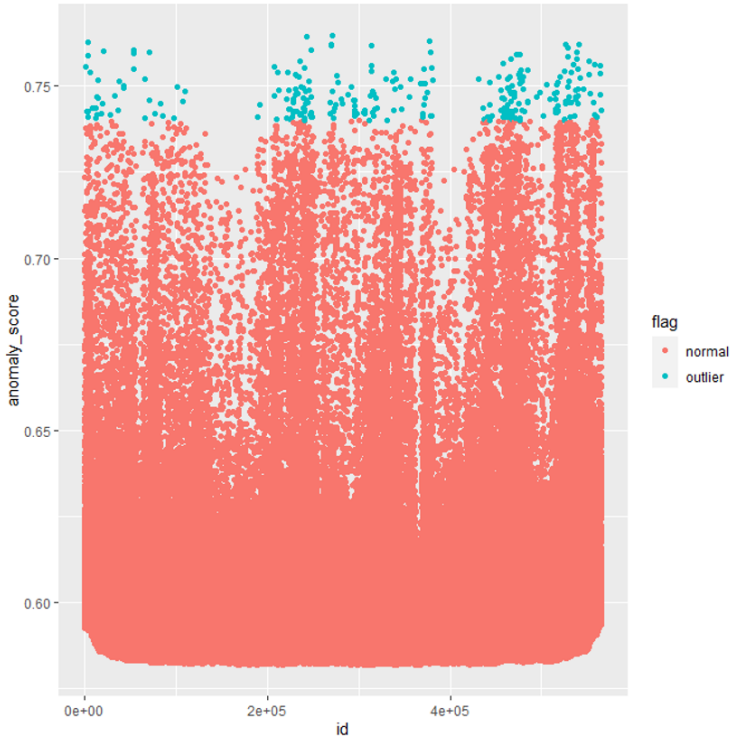

Figure 12: id verses anomaly score

Pros and Cons for Isotation Forest

Pros:

- Could deal with dimensional data
- Do NOT need labelled data
- Do Not Require data to be normally distributed
- Small time and memory requirements

Cons:

- Do not work well with elements with fewer data points

Attribute Value Frequency (AVF)

After conducting outlier analysis for continuous variables, such as arrival delay and departure delay, by using quantitative methods, we also try to detect outlying observations in categorical data.

The Attribute Value Frequency (AVF) algorithm is one of fastest and simplest outlier detection methods applied in a categorical data, since the AVF is based on a simple concept that outliers are those points

which have very low frequencies and it even does not need to create or search through different combinations of attribute values. We calculated the frequencies of each attribute value and assigned an AVF score to each data point. We categorized the arrival time based on its period of day. (Morning:5:00-12:00, Afternoon:12:00-17:00, Evening: 17:00-21:00, Night: 21:00-5:00).We calculate the AVF score for each observation based on the variables "DAY_OF_WEEK", "DEST_STATE_ABR", "TIME_OF_DAY"and "ORIGIN_AIRPORT_ID". We classify the observations with score lower than 45,000 as outliers and others to be normal points.

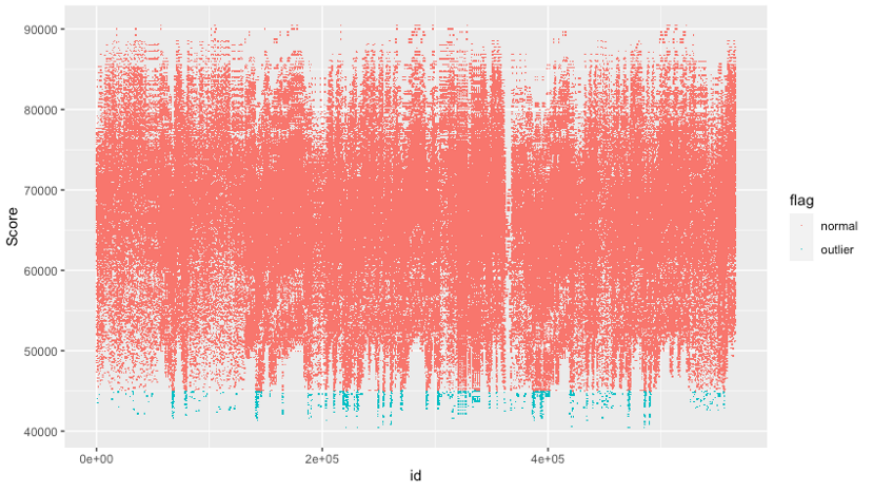

Figure 13: id verses score

Conclusion

After completing the four algorithms below, we now start to compare the results for them. We combined the results of the four algorithms into a single data frame and calculated for each observation, how many algorithms declared the observation was an outlier. The results of this are on figure 14.

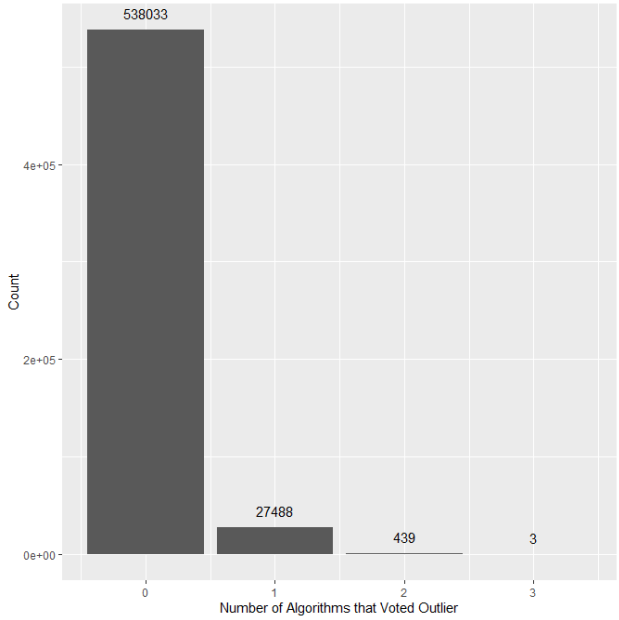

Figure 14: Four algorithms conclusion

According to the plots, we could see that only 3 observations are voted as outliers for three algorithms, 439 observations were voted as outliers for two algorithms, and 27488 observations for one algorithm. We were surprised to see that no observation was voted as an outlier by all four of the algorithms.

Votes	AVF	KNN	DBSCAN	Isolation.Forest
3 (3 total outliers)	2	3	1	3
2 (439 total outliers)	102	435	54	287
1 (27488 total outliers)	1567	24827	1094	0

From the vote breakdown, the distance to KNN model declared the most outliers for all 3 votes, followed by AVF and then DBSCAN. Isolation forest as a model works under the assumption that there are very few outliers, which is why there were only 290 outliers identified.

Limitations and Future Considerations:

One limitation we had with the algorithms was that we could not fully implement DBSCAN on the whole data set, and thus had to split it up into 5 subsets in order to carry out the algorithm. While we do believe that the subsets represented the population well, it is possible that some outliers may have been missed using this method. Another limitation is that, while we were interested in testing various algorithms, such as distance, density and category based algorithms, in the end comparing them and using them to vote on which observations are outliers has its difficulties. The categorical algorithm AVF identifies outliers on a different set of features compared to distance to KNN for example. While they may identify the same observation as being an outlier, it is for different reasons and further analysis/domain expertise is probably required in verifying the findings.

A future consideration we would like to test out if given further time is to somehow compare the outlier votes using a metric from each model. Rather than just considering which algorithm identified what as an outlier and taking the vote, we would like to quantify how confident the model is in a certain observation is an outlier and use a score based system rather than the vote. An example where this could be useful is a case where an observation has 2 votes, from say AVF and distance to KNN, if the AVF score was very low and distance very high, our current methodology could not tell this and we may conclude that a lower scored 3-voted observation is more likely.