UNIVERSITY of HOUSTON

College of Natural Sciences and Mathematics

Department of Computer Science

Fall 2020

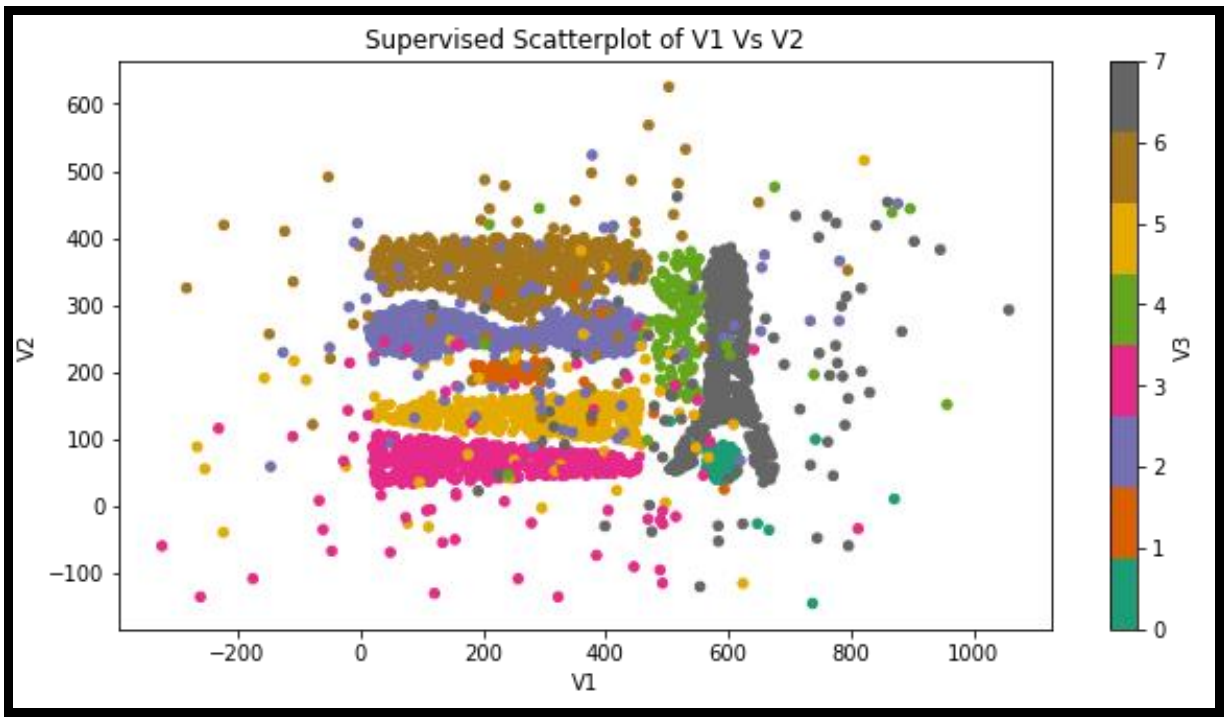
COSC6335 – Data Mining

Task 5: Design, Implementation and Evaluation of an Outlier Detection Techniques for a Spatial Dataset Report

Submitted By-
Pratik Ghatake
PSID- 1897033

Submitted to-
Dr. Christoph Eick
Professor, Data Mining

Task a:



- The above scatterplot is for V1 Vs. V2. This is a supervised scatterplot as it is coloured with class labels for respective data points.
- From the legend at the right-hand side, we can see that there are 8 class labels starting from class 0 to class 7.
- These are labelled with different colours as shows in the right-hand side. In the plot, roughly there are 8 clusters getting formed so are 8 class labels.
- There are 5 clusters at slight left side of the plot, whereas there are three clusters at slight right of the plot.
- The 5 clusters at the left are horizontal in nature. The zeroth cluster or the class with label 0(green), is circular in shape. The remaining two clusters are vertical in nature.
- The 6^{th} cluster(brown) appears to be the biggest one, having major area. Conversely, the 0^{th} cluster appears to be the smallest one. Precisely, we can't figure out whether

cluster is big or not because of the density, we can't form hypothesis about no. of data points and cluster size. Its very hard to determine no of points belonging to one cluster from the scatterplot. What can be depicted is the area spanned by the cluster.

- The horizontal clusters are roughly spanning in the area for $0 < V1 < 450$ and $30 < V2 < 420$. Clusters 2,3,5,6 have almost same length.
- The cluster no. 4 (Green) appears to be distorted little bit with slightly sparse density unlike other clusters.
- The noise or the outliers from the data can be easily detected. Around all the 8 clusters there are noisy data points. The density for noisy data points at left hand side is lower as compare to the density of noisy data points at right hand side.
- At the bottom and lower left corner there are many noisy data points belonging to the class 3 (Pink). Similarly, there are many noisy data points at right hand side belonging to class 7. At the top, we can see noisy data points for class 6.
- If looked closely, there is a gap between horizontal clusters and verticals clusters. This gap is also filled with noisy data points.
- Overall, the data is clustered in the region $0 < V1 < 680$ and $0 < V2 < 420$.

Task d:

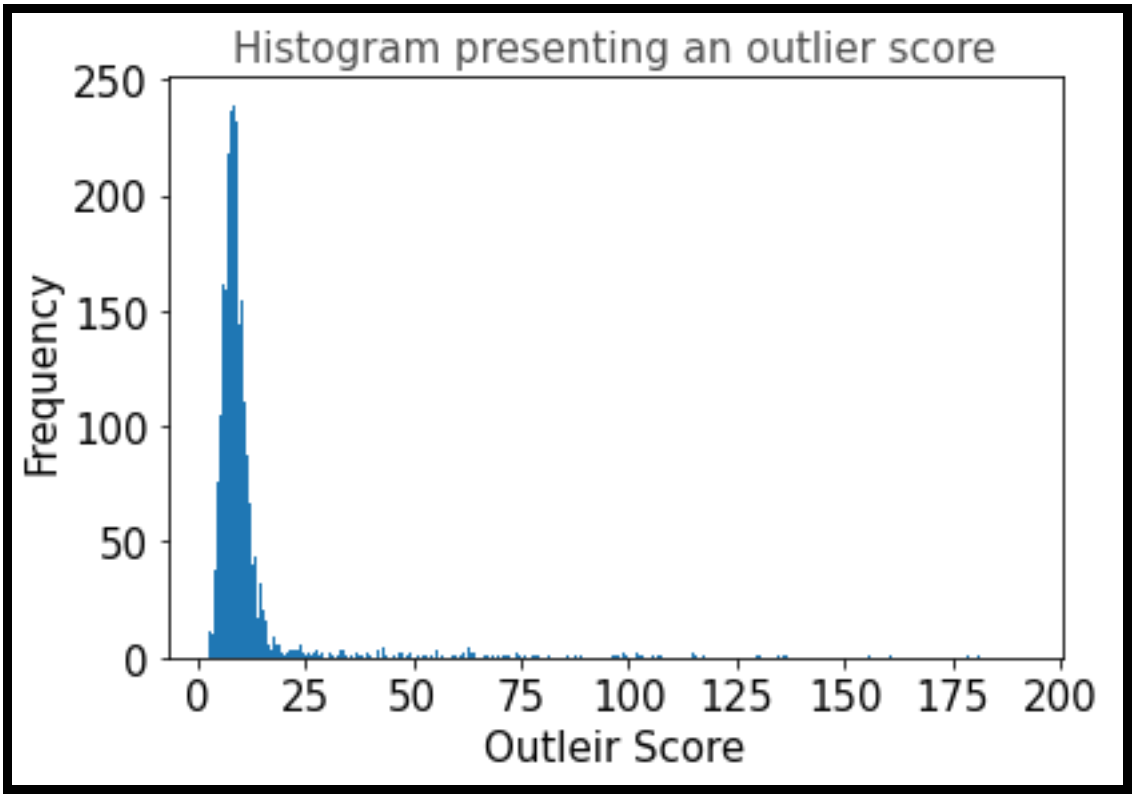
Outlier Detection Algorithm-

For the outlier detection algorithm, I have incorporated the KNN algorithm with different techniques to enhance the algorithm's detection capability. The PyOD library was used to detect the outliers, specifically KNN from PyOD was used. I have tried to streamline the process pf algorithm by dividing it in 3 steps. First one being the model building then comes is to determine the sensible threshold. Last step is presenting the summary statistic of the technique used. Since the task is unsupervised, training took

place only on attributes V1 andV2. The V3 or the class labels are not considered during the training part. If considered otherwise, the task becomes supervised machine learning which is not in our case. The class labels (V3) are used only for validation and visualization purpose.

Therefore, our data for training the model is unlabelled. The algorithms identify the anomalies or outliers in the data when we fit the model. For the PyOD KNN model, the points which belong to the low-density regions are considered as outlier and those who do not would be considered as signal.

Primarily I have used simple KNN model from PyOD library which is proximity-based algorithm. The outliers scores are calculated by using the distance if each data point to its Kth nearest neighbour. After fitting the model, the I have saved the results of the outlier scores in an array called 'dfscores'. The score is calculated as explained above. Higher is the score, more abnormal data point is. Therefore, outliers tend to have higher scores. When the model is fitted to the data, the decision scores (Outlier scores) are available to view. For the algorithm, I have used the default distance matrix which is Eucledian. Since we are performing unsupervised KNN, it doesn't have any learning involved in it. For computing the distance of very data points, target variable or V3 in our case is not required. After fitting the model, I have predicted the model on whole dataset. After prediction it returns the outlier scores in decision_function ( ). Each point s has an outlier score and the main focus is to find out the points above threshold view in order to consider them as an outliers or noise. Also, we can see there are two classes formed, 1 and 0 where 1 belongs to class of outliers. I have visualized the outlier scores using histogram so that I could decide what would be the reasonable boundary for threshold.



The histogram as plotted for outlier scores of each data point. From the above histogram, we can see that frequency is decreasing around 25, after which it is gradually decreasing. Therefore, I set the threshold of 25 for outlier score. For 0 to 25 range, there are high number of points, representing high frequency and are signal. From the threshold the points which have OLS score greater than 25 would be considered as outliers.

Then, in the last step I have calculated and printed the summary statistics for outlier scores of signal points and outliers. The cluster 0 belongs to the signal points whose average outlier score is 8.86 which is pretty low. ON the other hand, cluster 1 belongs to outliers and therefore has high average outliers score 66.34. The high score for outlier was expected and is getting verified from the summary statistics. Therefore, summary statistic showed us the dramatic differences between the signal and noise data points (Cluster 0 and Cluster 1 resp.)

When it comes to the unsupervised machine learning tasks, the models are easily susceptible for overfitting. Therefore, the PyOD library recommends combining various detector outputs, by averaging or by other method. This improves the robustness of the algorithm. Following the recommendation, I tried to enhance the stability of the model using different approaches provided by the PyOD library. I have tried to use following methods –

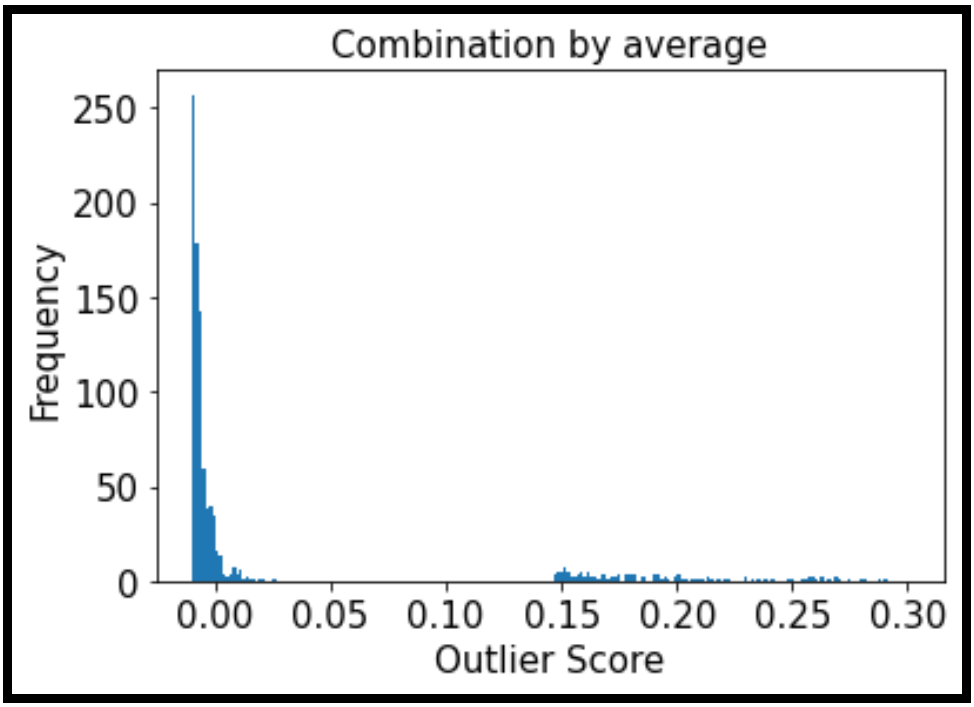
1. **Average**
2. **Maximum of Maximum (MOM)**
3. **Average of Maximum (AOM)**
4. **Maximum of Average (MOA)**

Before, applying all these 4 methods, I have created 20 knn modes by specifying different no. of neighbours. This is nothing g but creating gird search space. I have created a list starting from 10 to 200 for KNN. The model will get trained with different no. of K's and I have collected all the predictions in one dataset. Therefore, in other words, all the data points (2933) will have 20 predictions by using each K given in the list. Also, I have used standardization so as to avoid overfitting. For KNN, method used is 'largest' which means that the distance to the Kth neighbour would be considered as an outlier score. It also has another, methods like mean and median. As before, I have saved the outlier scores or decision scores in an array.

Average method takes the average score of all detectors whereas maximum method takes the maximum score from all the detectors. The third method Average of maximum (AOM) divides the base detectors and takes the maximum score of each subgroup. The final score is the average of all such subgroups scores. The maximum of average (MOA) uses similar approach

that is dividing the base detectors into subgroups and taking the average score for each subgroup . Final score would be the maximum score of all such subgroups. For all the methods, I have standardized the decision score so the outlier scores from these approaches as this is the crucial step. The standardization would transform them into zero average and unit std. This step is performed since the output of different approaches varies, and it becomes difficult to compare them if not on same scale. After applying the algorithm, I have plotted the histogram for outlier score and threshold is decided. Once deciding the thresholds, clusters are fremd. At the end, the statistical summaries are printed so as to give an idea of cluster and its average outlier score.

1. Average



Histogram for average method is as shown above. The threshold was decided to be 0.00 in this case since there is a high peak of values belonging to low scores of points. These points are nothing but signals. And there are few points with high score greater than 0

which are outliers. Also, we can see after major gap, there are few points which have outlier score greater than 0.15.

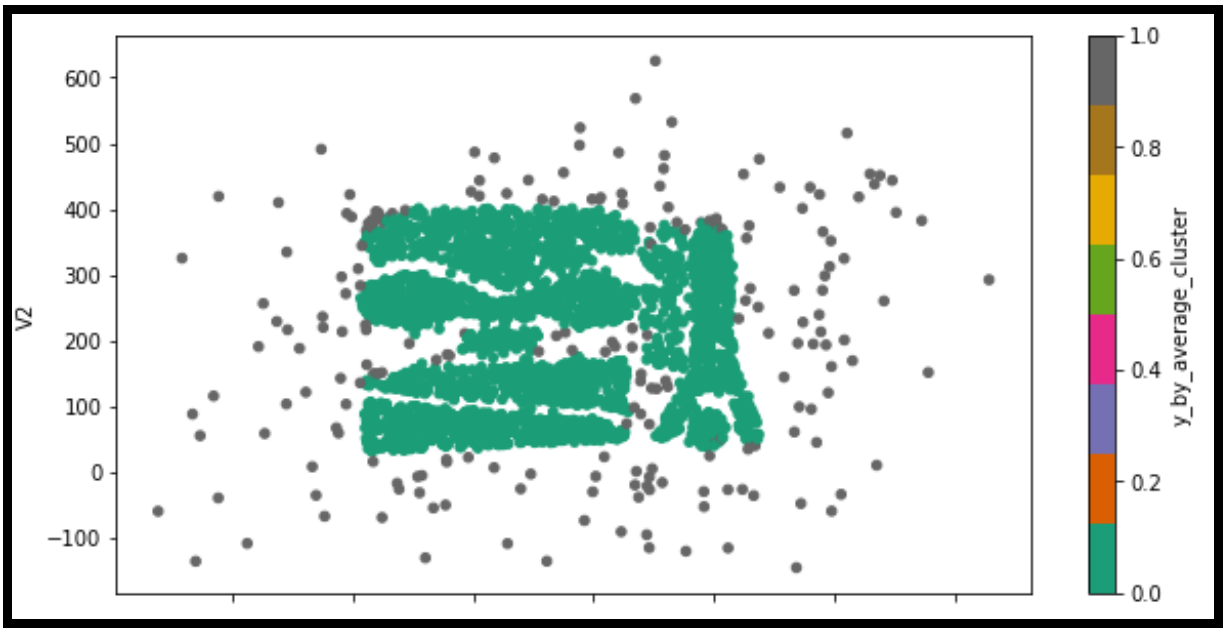
```
0     2718
1      215
```

We get 215 points as an outlier after specifying the threshold.

The summary statistics are as below-

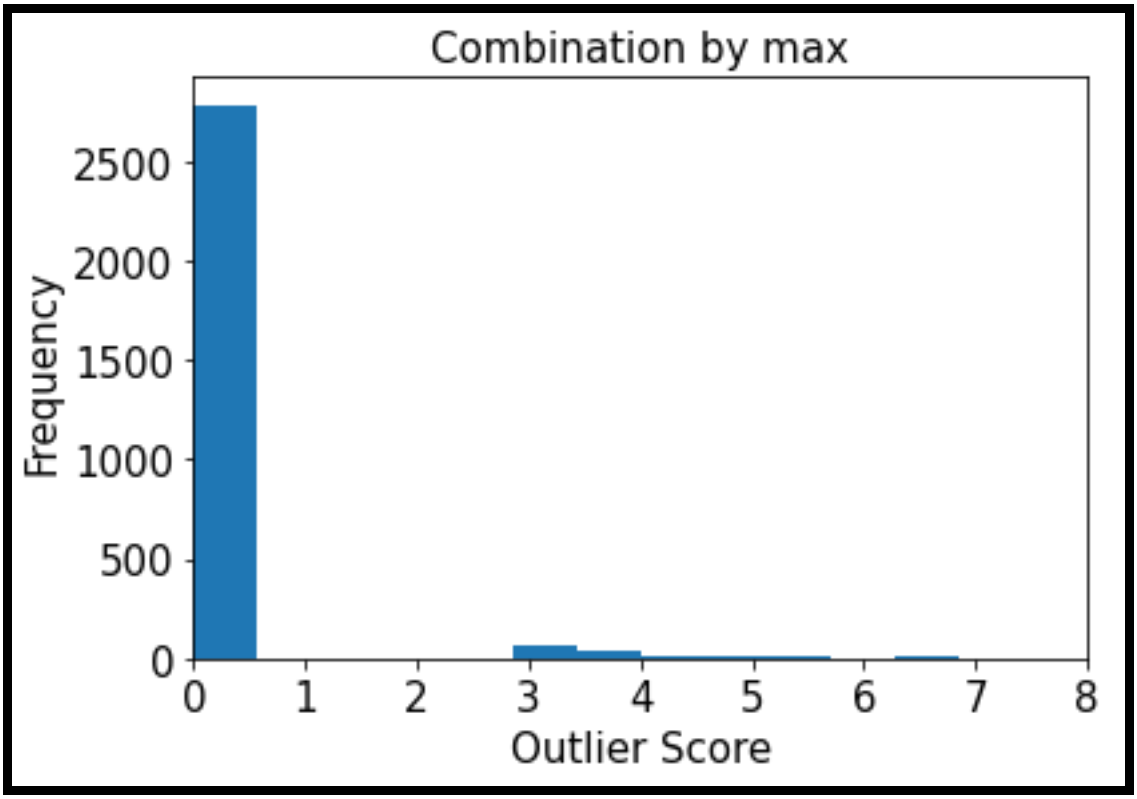
y_by_average_cluster	V1	V2	OLS	Cluster	y_by_average_score
0	325.542211	207.119476	8.652379	0.000000	-0.011496
1	369.874502	205.674690	51.828161	0.702326	0.145327

For the class 0, outlier score seems to be pretty low, and for cluster 1 the outlier scores are high which is pretty obvious fact as these points are noise data points. The OLS score is the score calculate using default KNN whereas y_by_average_score is the score obtained after applying the average technique on the algorithm.



The histogram is plotted for V1 and V2 attributes after applying average score technique. We can see that this approach is predicting the outliers nicely with few exceptions. For ex. There are certain data points in top left portion which are a nearby a cluster but are considered as outliers. Also, there are few data points which are considered as outliers for vertical clusters at right. The approach also predicts the points in the gap of horizontal and vertical clusters as an outlier which depicts its high quality of removal of noise from the data.

2. Maximum of Maximum (MOM)



For each of the technique same method is used with only difference in functions. The maximization would take simple combinations by taking maximum scores. From the

histogram it can be easily noted that threshold is 0.5. Therefore, the points with outlier scores greater than 0.5 would be considered as an outlier. There are certain gaps in the data between 0.5 to 2.8, after 7 etc. The high peak at left hand side of the histogram shows skewed distribution and its pretty obvious since we have many observations which are normal as compared to the noise in data.

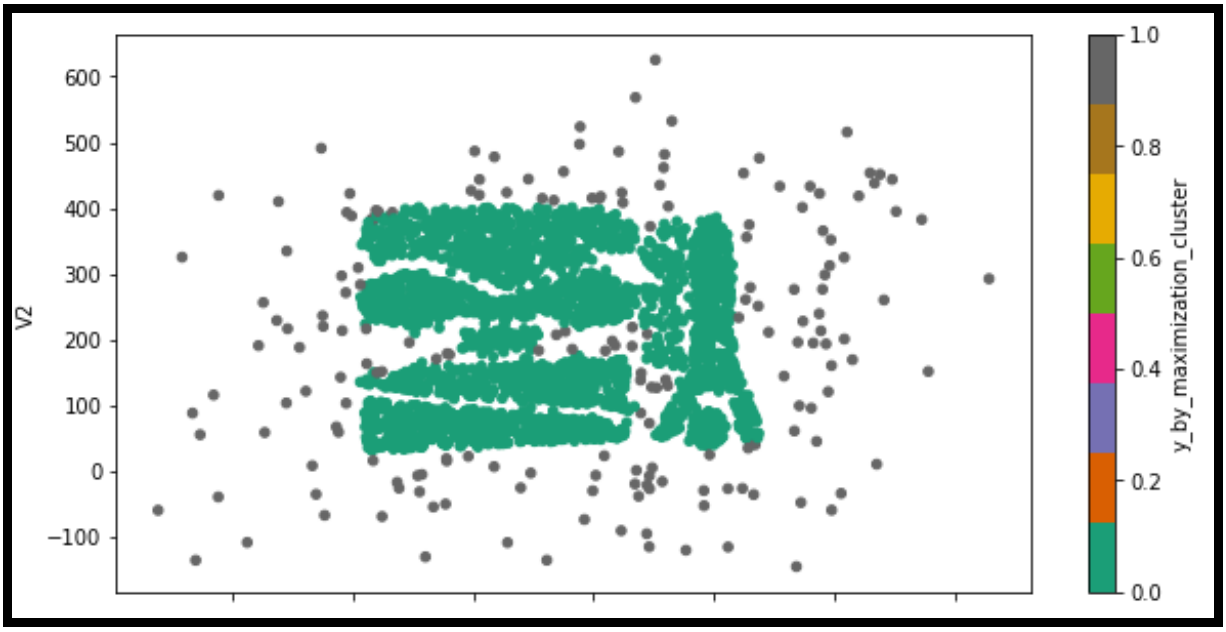
```
0    2745
1     188
Name: y_by_maximization_cluster, dtype: int64
```

After specifying threshold, the above result is obtained. I am succeeded to predict 188 points as an outlier belonging to cluster 1.

Summary statistics are as shown below-

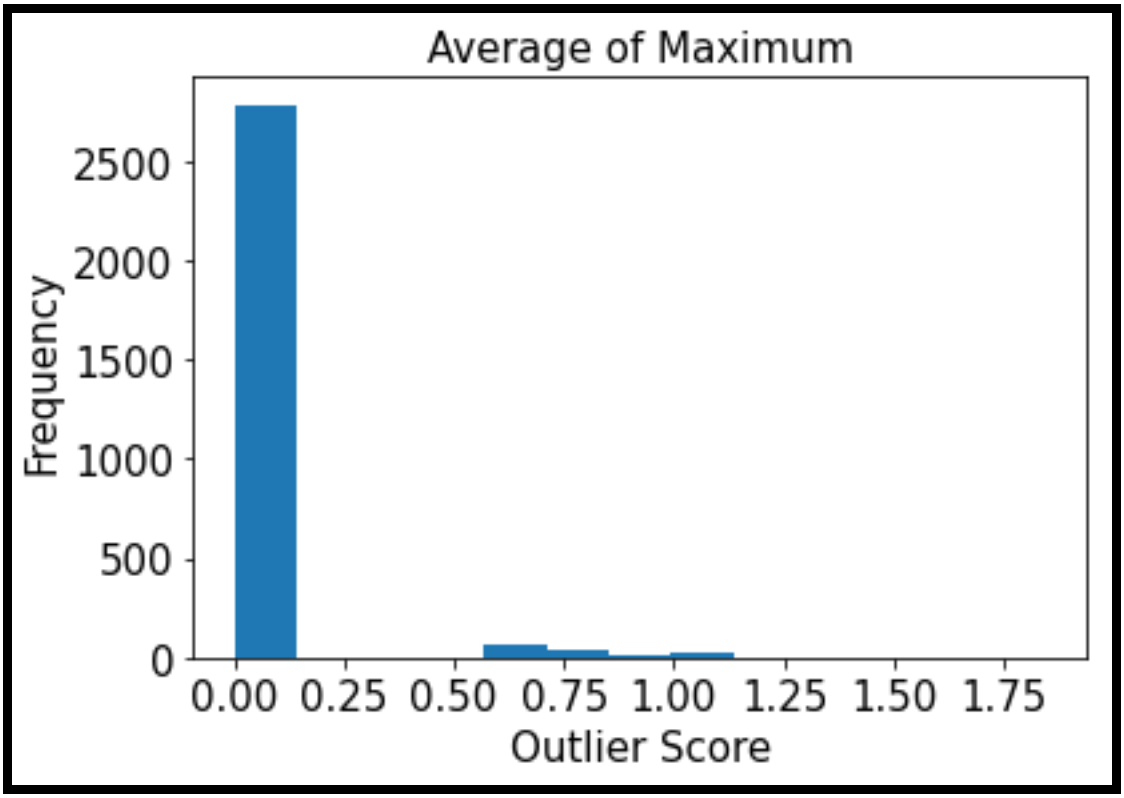
y_by_maximization_cluster	V1	V2	OLS	y_by_maximization_score
0	325.474908	207.916058	8.700370	0.000223
1	377.224075	193.836239	57.328217	3.320725

The summary statistics explicitly states that points belonging to cluster 1 have high OLS score. Again, the terminology is similar to the average approach. ‘y_by_maximization_score’ presents the score produced by this technique.



As compared to scatter plot by average approach this seems to be more accurate. Since it has tried to predict many points correctly as outlier. Also, the exceptions/ mistakes observed in average scatterplot are greatly reduced in this approach, which can be verified by the above graph.

3. Average of Maximum (AOM)



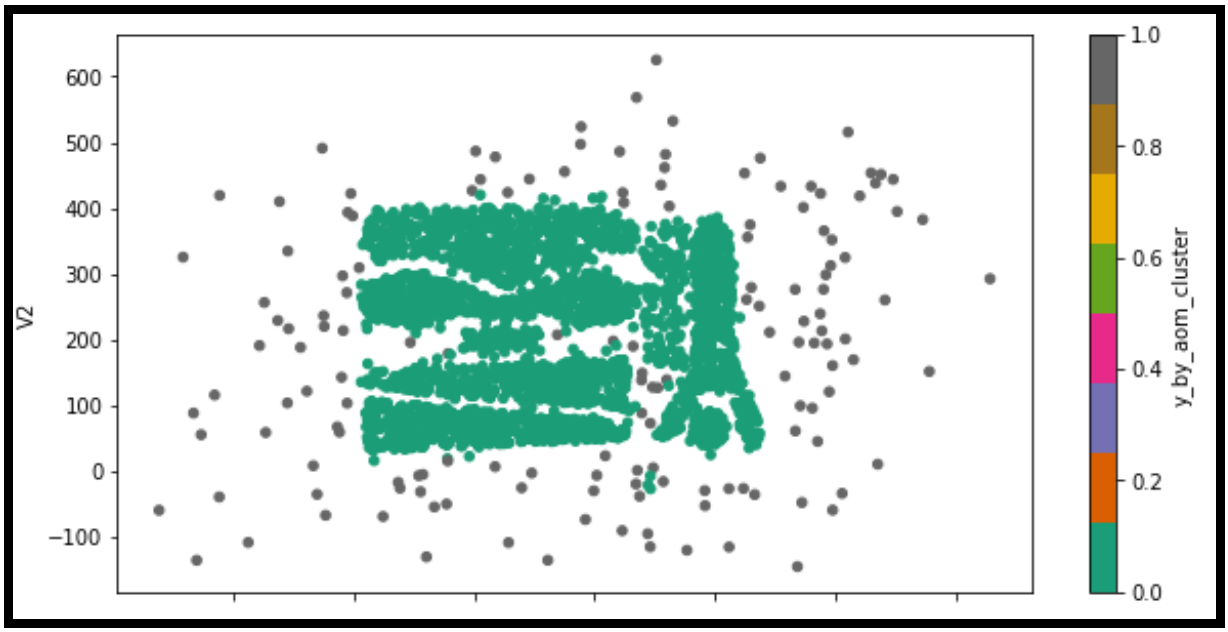
The approach as explained before, subdivided the detectors and takes the average of all subgroup scores. From the histogram, we can specify threshold to be around 0.125, given the high peak at left hand side. All the histograms for all the approaches are highly skewed to the right presenting high no of population for low outlier scores, i.e., normal data points, signal.

```
0    2782
1     151
Name: y_by_aom_cluster, dtype: int64
```

We able to predict 151 points as an outliers and 2782 as normal data points or signals. Summary statistics are as shown below-

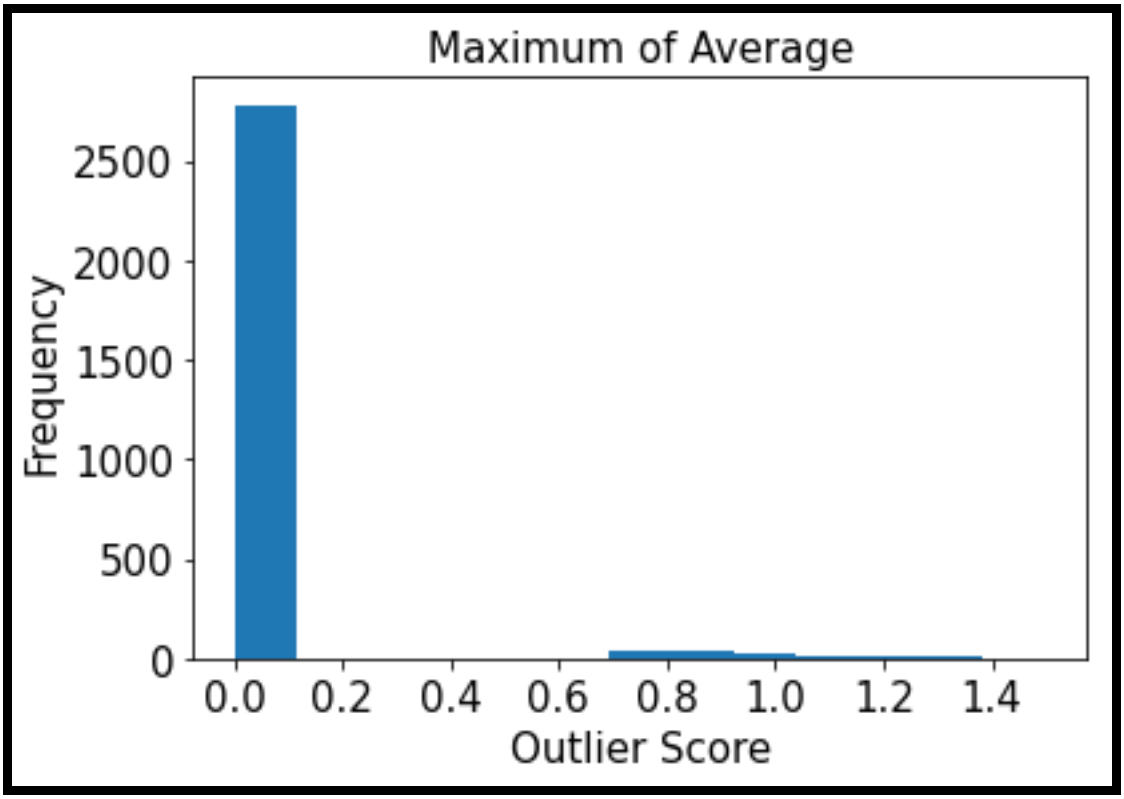
y_by_aom_cluster	V1	V2	OLS	y_by_aom_score
0	325.105726	208.019181	8.857743	0.000542
1	396.706076	188.486294	66.344230	0.817709

As seen from the above summary, the cluster 1 belongs to outliers and has high score 0.817709. Whereas class 0 is having very low score 0.000542.



This approach is also able to predict most of the outliers but I think, its overestimating the points as an outliers. Some points are part of cluster but are considered to be an outlier. Also, there is one exception involved at the bottom where accidentally 2 points are considered as normal data points or signal which are actually outliers. Therefore, the quality of this approach seems to be slightly low as compared to maximization approach.

4. Maximum of Average (MOA):



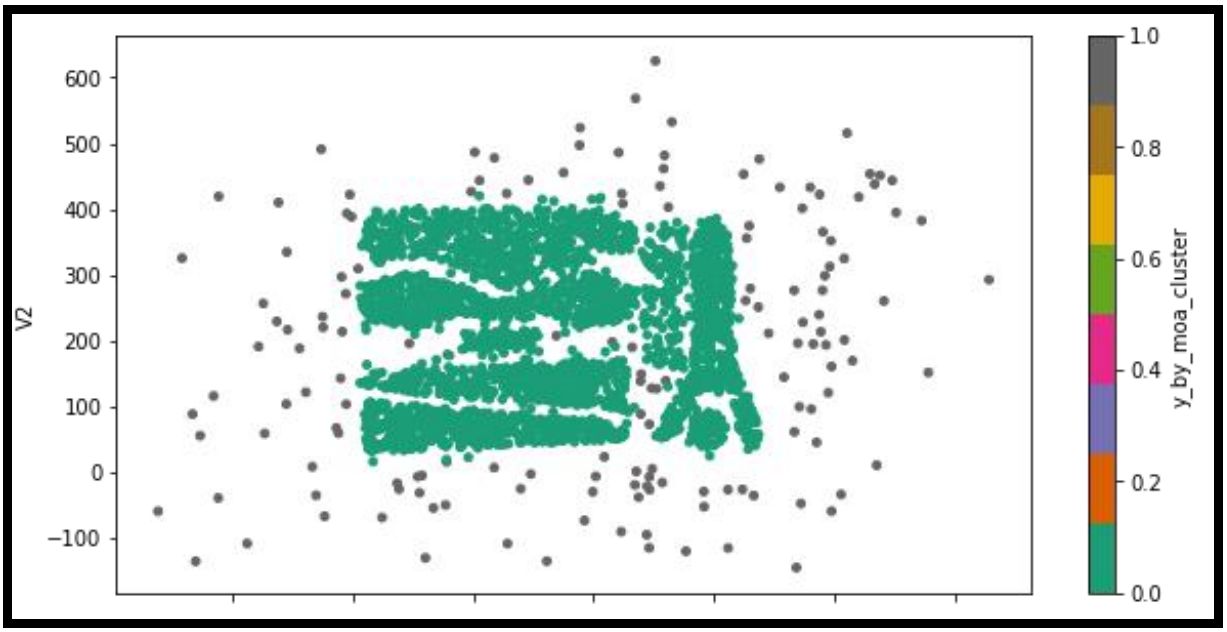
In this approach, detectors are divided into subgroups and then average score for each subgroup is calculated. Final score is the maximum of all average scores of subgroups. As the histogram indicates, threshold for deciding the point to be a signal or noise is set to be 0.10. After a gap for 0.4 units we can again see some frequencies at 0.8 till 1.4 which are assumed to be outliers.

```
0    2779
1     154
Name: y_by_moa_cluster, dtype: int64
```

I got 154 points as outliers and 2779 as signal points as sown in above summary. Summary statistics are as shown below-

y_by_moa_cluster	V1	V2	y_by_moa_score
0	324.925704	208.263489	0.000559
1	398.559853	184.458159	1.004373

From the statistical summary we can easily interpret that points belonging to cluster 1 has high OLS score and points belonging to cluster 0 has low score (0.000559).

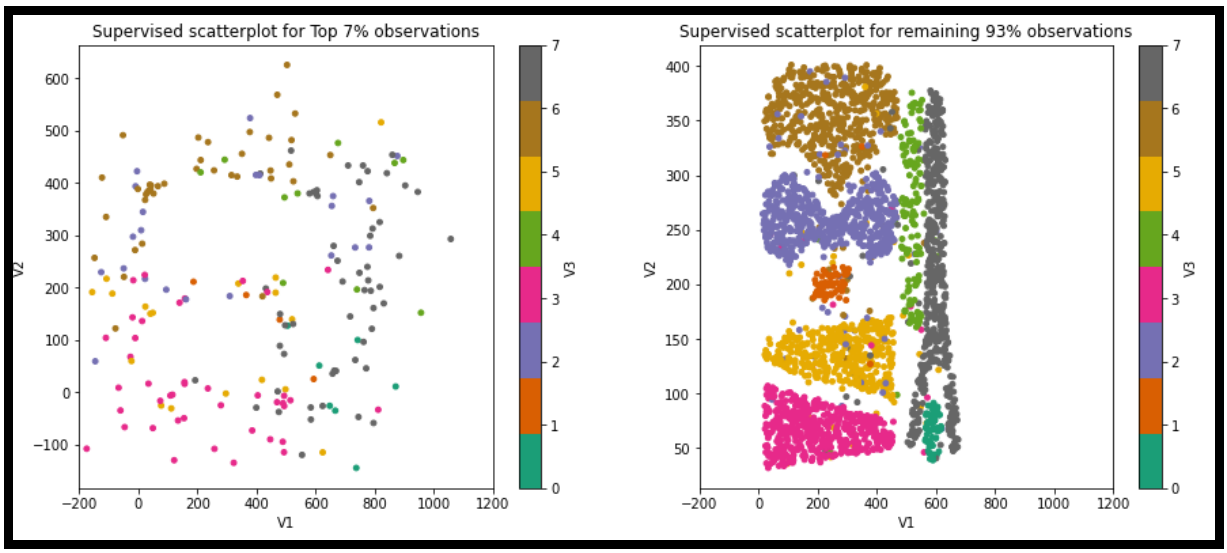


As compared to AOM, this works well in the terms of quality of outlier detection. Most of the data points are successfully detected as outliers. Also, the gap between horizontal and vertical cluster is detected well.

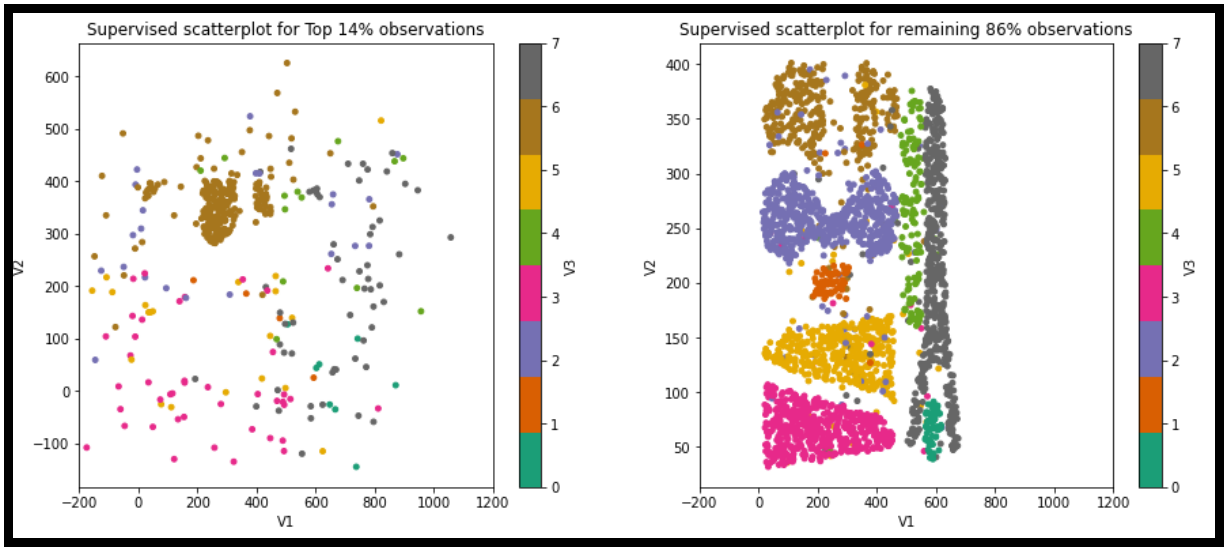
From all the 4 approaches, maximization approach seems to have highest quality and was able to detect maximum outliers successfully. Therefore, I have chosen to go ahead with this approach. I have created a final dataframe with V1, V2, V3, OLS score

(y_by_maximization_score) and cluster to which they belong to. If the cluster is 0 then it is a signal or normal data point whereas cluster 1 represents outlier cluster. I have append the scores to this final dataframe and sorted the rows as per the OLS score in descending order.

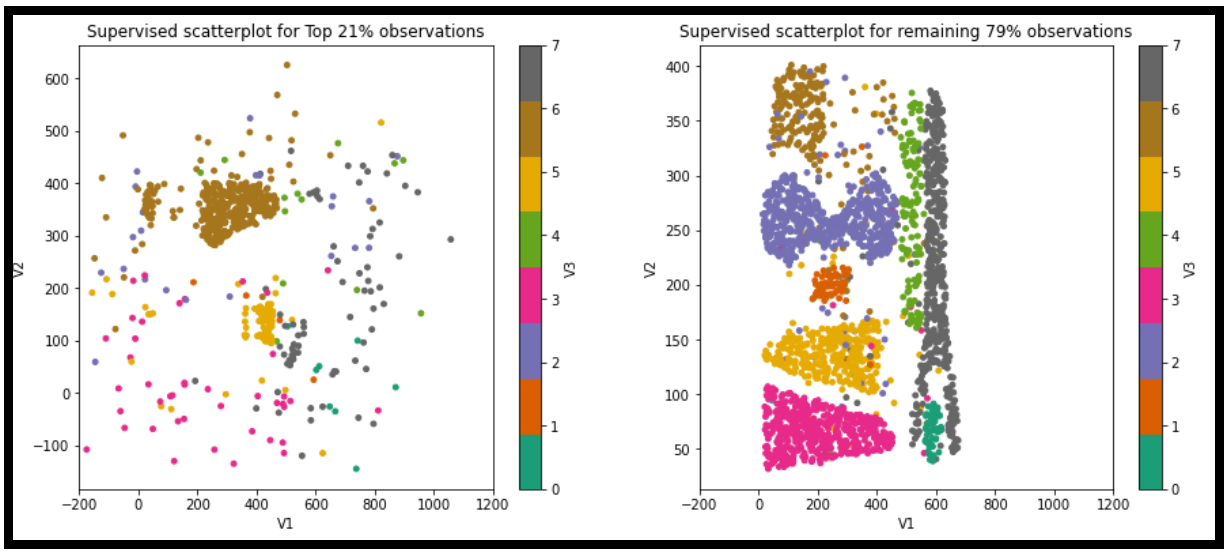
Task f:



From the above scatterplot, we can see that when we consider top 7% most points are belonging to outliers or cluster 1 as per our algorithm. Notice that both plots have different range for X. For ex. points are scattered from -200 to 1100 approx. For top 7% plot. Conversely, points are scattered only between 0 to approx. 700 for remaining 93% plot. Most of the points are in the sparse region of the plot and since the algorithm considers the density of the points to its K neighbours. The points on the edges also seems to have removed because of low density. Since these points will have many points on one side and very few on other. Remaining 93 % points are the points which re not in plot on left side. These are mostly signal points with high density and forming cluster as shown in the figure.



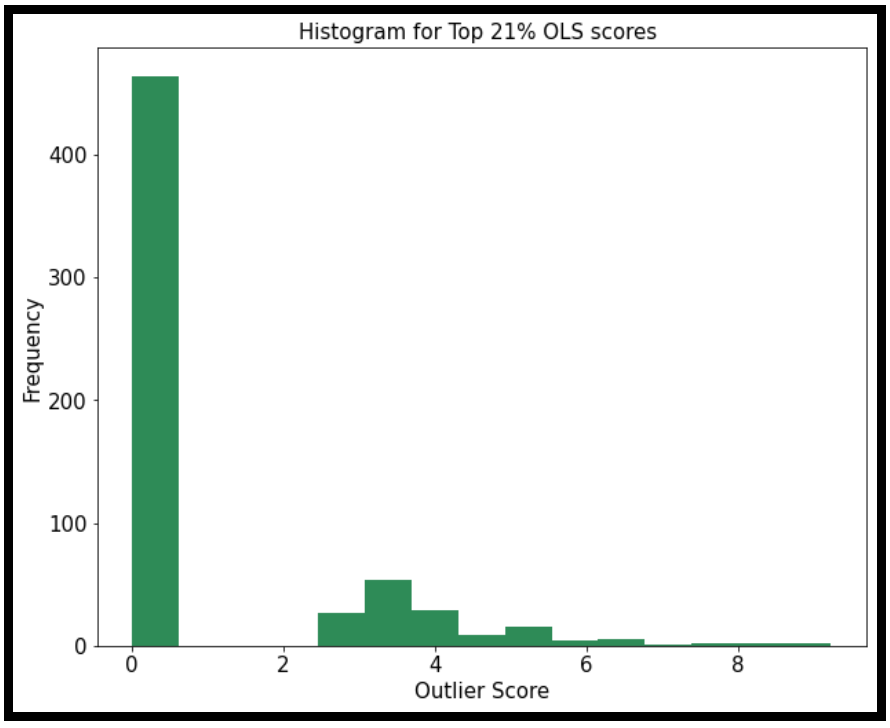
The supervised scatterplot for Top 14% is shown on left hand side. We can quickly observe that there's one cluster belonging to class 6 which can be seen from left plot. These are the points whose OLS score is higher as per algorithm but are not considered as an outlier. Therefore, we can make a hypothesis that actually no of outlier data points has to be less than 14%. When considered for top 14%, majority of pints form cluster 6 got removed. Therefore, another hypothesis could be that points belonging to this class are having higher OLS scores as compared to points belonging to other class.



The scatterplot for top 21% and remaining 79% is as shown in the figure. Now, after top 14%, majority of points from cluster 6 and 5 have high OLS scores and therefore is the reason they are sowing up in left side plot. From the remaining 79% we can see that cluster for class 6 almost loosen half of it part and so does class 5. There are very few points for class 1 with high OLS scores as they don't show up in top 21% observations plot.

Overall, if we compare the plots, as we decrease the remaining percentage graphs go on decreasing its density and clusters are therefore not tight. Cluster for class label 6 and 5 are the too clusters to lose their points when percentage is increased for; left side plot. Therefore, these are the points which have high OLS scores but aren't really outliers. 7% seems to be reasonable boundary or cut for visualizing outliers. If points are getting considered as noise as like in other plots, it would lead to information loss.

Task g:



- The above histogram shows the Outlier scores for Top 21% record from all observations.
- It can be easily noticed that the histogram is highly right skewed.
- The OLS values seems to be positive for all observations. Hence, histogram starting from 0.
- There is only one gap in the histogram, one being from 0.5 to apporx.2.3. In between 6 to 8 there are very few observations but is not a gap.
- The threshold or the cut point from this histogram seems to be 0. 5, since after this peak for frequency falls.
- The extreme right skewed nature of the plot is pretty obvious given the fact of outlier detection. Many points belonging to the histogram are signal points or normal data points whose outlier score is low.

Software Description:

For the above task Python is used as a programming language since I was most comfortable with this language. Also, Pandas and numpy library is used which are inevitable part of any data analysis project. These libraries help in data wrangling and data engineering part and are very convenient tow work with. For visualization purposes, I have used matplotlib library which is a comprehensive graphing solution for python when it comes to Exploratory Data analysis (EDA). Also, for pre-processing and KNN sklearn is used. Pre-processing involves standardization and normalization functions.

Last but not least, I have used PyOD package throughout this task. Specifically, Unsupervised KNN model is used as an outlier detection algorithm for the task, it has serval different models such as Local Outlier Factor, Isolation Forest, Supervised KNN, Histogram based outlier score etc. PyOD is scalable Python toolkit which is vastly used for anomaly

detection in multivariate data. This library has a scikit style API which includes numerous detection algorithm implementations. Also, I have leveraged the model combinations part of this library by using functions such as average, max, AOM, MOA.

References:

1. Official Documentation of PyOD: https://pyod.readthedocs.io/en/latest/index.html
2. Functions of PyOD: https://pyod.readthedocs.io/en/latest/api_cc.html
3. https://pyod.readthedocs.io/en/latest/example.html?highlight=knn#model-combination-example
4. https://www.kdnuggets.com/2019/02/outlier-detection-methods-cheat-sheet.html
5. https://www.analyticsvidhya.com/blog/2019/02/outlier-detection-python-pyod/
6. https://towardsdatascience.com/anomaly-detection-with-pyod-b523fc47db9