Technical University of Denmark

02450 Introduction to Machine Learning and Data Mining

Year 2019-2020

Group Project 3

PROJECT 3

Group Members

Name	Student Number
Altug Tosun	s181314
Mahsa Eskandarzadeh	s192933
Sukru Han Sahin	s192136

Contributions

Section	Author
1	Altug Tosun, Mahsa Eskandarzadeh, Sukru Han Sahin
2	Altug Tosun, Mahsa Eskandarzadeh, Sukru Han Sahin
3	Altug Tosun, Mahsa Eskandarzadeh, Sukru Han Sahin

1 Clustering

In LA Ozone data, there are 10 attributes that has been explained in the first report. Ozone attribute is selected as a response variable. Day of the year attribute is not used in analysis and the remaining 8 attributes selected as explanatory variables according to findings from first report. Since the day of the year is related with time series, it was set to date object in previous report. Therefore, the inspection with day of the year can be made by considering ozone attribute as a time series but it is not relevant for report so it will not be done. LA ozone data-set does not have any categorical attribute so one-of-K coding isn't applied. Nonetheless, since regularization will be used, all of the attributes in the data-set are standardized to have standard deviation 1 and mean 0. It is done by subtracting mean of each column from their column and dividing these columns to their standard deviation values. For the section 3, (Association Mining) data isn't standardized as in the example of exercises.

Question 1 :

Before applying the required steps, first the y output, which is ozone level, classified into 3 classes with same method used in previous project. To do this split in a more statistical way, mean and standard deviation values of Ozone attribute is being used. Standard deviation is subtracted from mean and values which are less than 3 are classified as "low" class. Standard deviation is added to mean and values which are more 19 are classified as "high" class. And observations which doesn't belong to these classes are classified as "medium" class. Then our output variable is transformed into **multi-class** variable. After that, hierarchical clustering of our data using a suitable dissimilarity measure and linkage function is performed as follows;

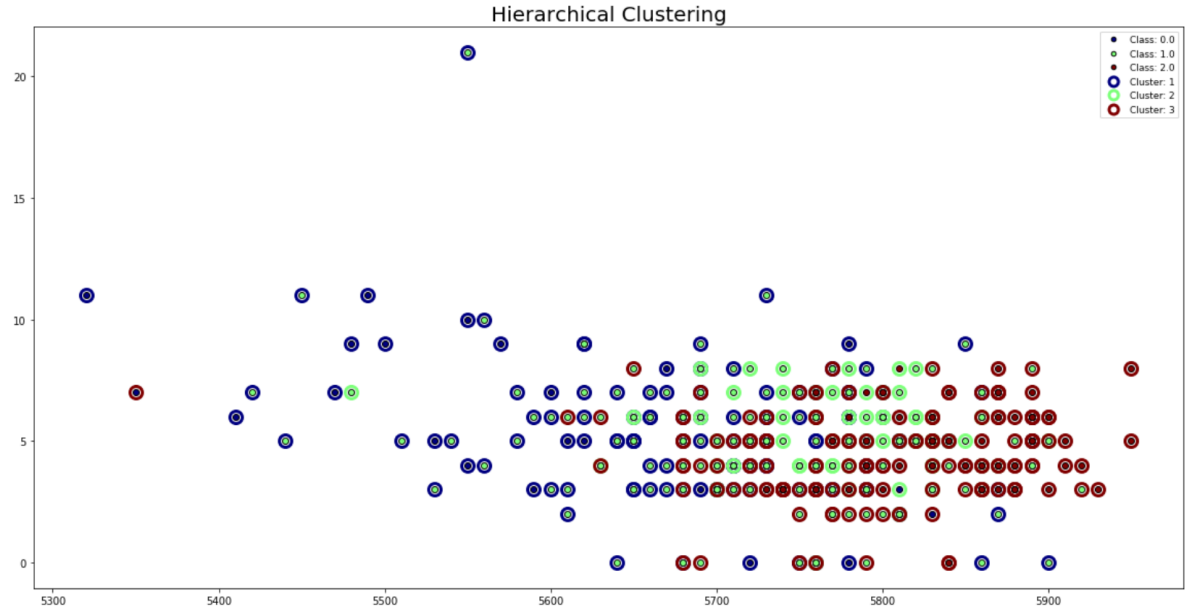

Figure 1: Hierarchical Clustering: There are 3 clusters. Since our output data has 3 classes, maximum number of clusters are given as an input of 3.

Above plot is being created with the "clusterplot" function from the toolbox. Since the input dimension is greater than 2, data is projected onto the first two principal components. Data objects are plotted as a dot with a circle around. The color of the dot indicates the true class and the circle indicates the cluster index.

Our first problem is that; number of clusters is given as an input, not optimized yet. Since our input data has 8 dimension, while applying PCA, some important information is lost and figure includes some of the variation from data, but not all of the variation. Therefore it is hard to comment clear results from the plot above. Label of the classes doesn't represent exact characteristics but clustering is extracting natural structure of data. (Labels are as; "low", "medium" and "high" level of ozone.) In the end, from the above plot, onto the PC-1 and PC-2 space, when the cluster indices are examined, blue dots are spread around more and green darkred dots are more closer to each other.

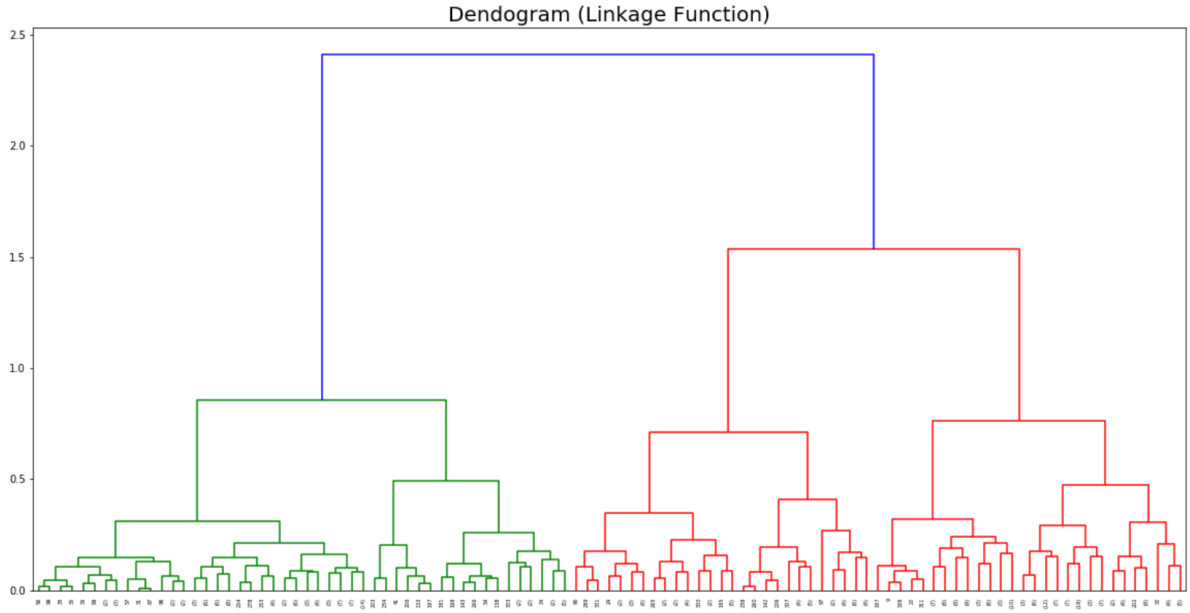

Figure 2: Dendogram (Linkage Function): Dendogram has 6 display levels. This diagram shows hierarchical relationship between data points.

Above dendogram, created as an output from hierarchical clustering. With the help of dendrogram, data points are allocated to clusters. Complete method is used with euclidean metric. Levels above are easily noticeable. Hierarchical clustering of data into two then four isn't matching with our 3 classes, as it is an input to Hierarchical Clustering plot at Figure 1.

Question 2 :

At this part, data is clustered by the Gaussian Mixture Model (GMM) and crossvalidation is used to estimate the number of components in the GMM. Density of data is calculated as the weighted sum of "k" number of mutivariate normal distribution in GMM. "k" is the number of clusters or number of components in GMM. "k" is going to be determined with crossvalidation. GMM model is created with full covariane matrix, repetitions for convergence is set to 3. "k" value is set to between 1 to 10 including both. For each "k" values, "BIC", "AIC" and log-likelihood values are obtained. Important point here; log-likelihood values are obtained via holding out cross validation method. Log-likelihood values are calculated on test data. Then, below figure is created.

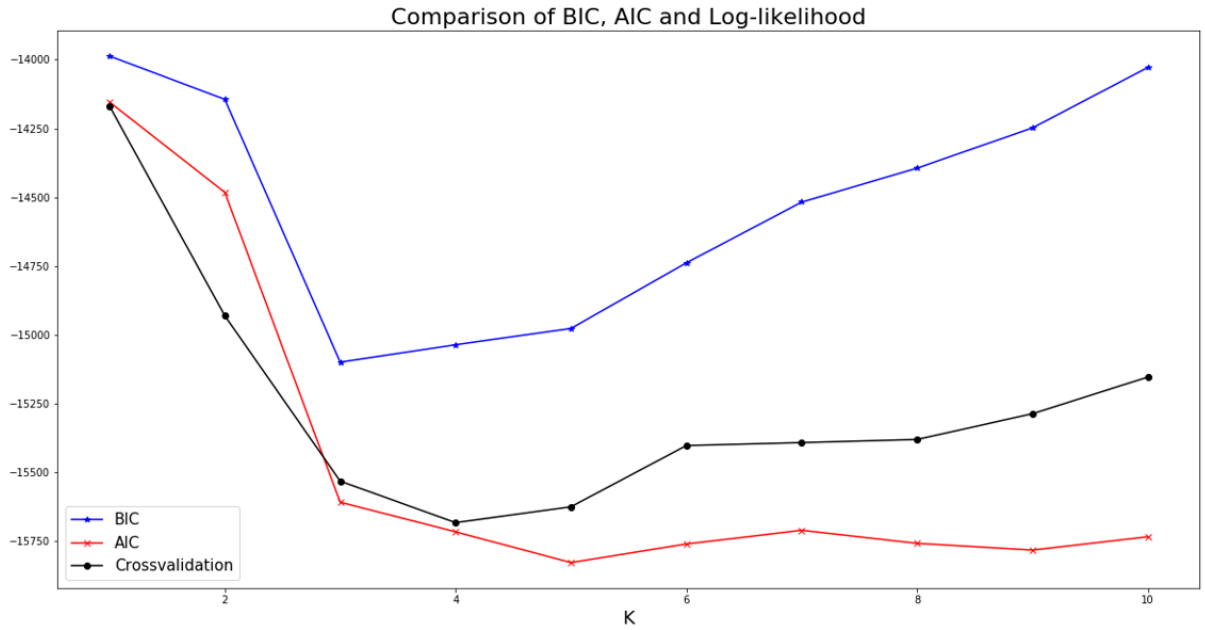

Figure 3: AIC, BIC, and Log-Likelihood Comparison: "k" is optimized as seen.

BIC and log-likelihood has a very similar moving trend. they find a minimum value "k" around 3,4 and then starts to increase. For AIC, it seems that, it is not as robust as BIC log-likelihood in terms of overfitting. Because it shows more frequent ups and downs. With the help of $KRange[CVE.argmin()]$ function, "k" is calculated as 4. Number of cluster is set to 4.

When the dendogram in the Figure 2 is considered, now it is more meaningful to consider 4 classes. Because in dendogram, clustering data into 2 or 4 seemed more logical at which the GMM yieleded similiar result 4, as well.

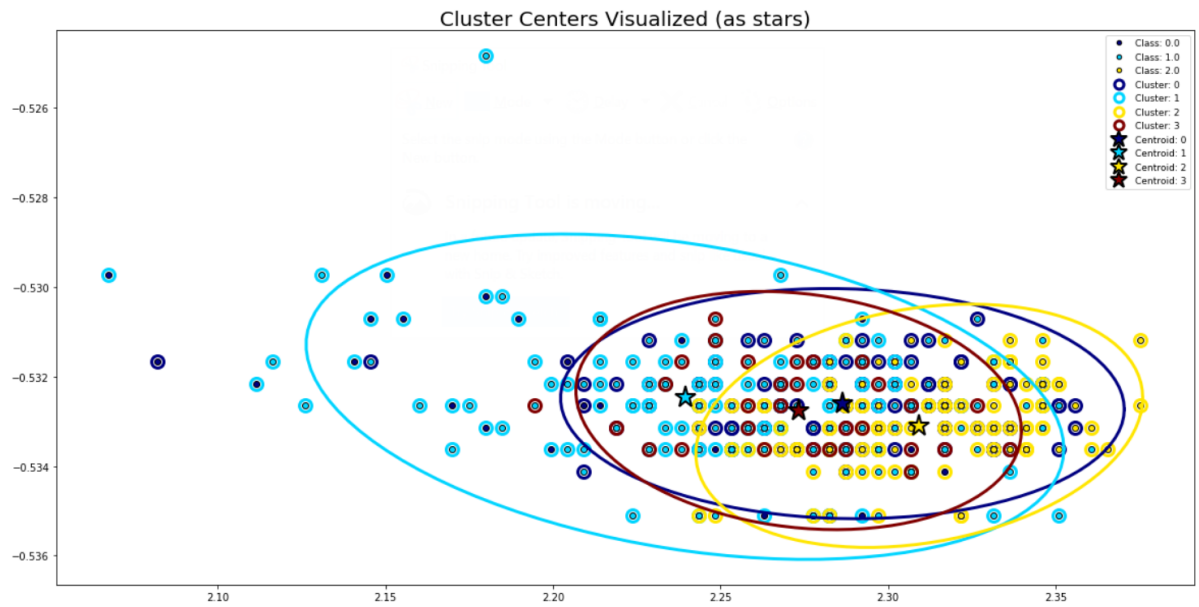

Figure 4: Plot of Cluster Centers: Arbitrarily, very first 2 attributes are chosen in order to visualized cluster centers just for these attributes. (Attributes are; "vh" and "wind".)

Figure 4 above is just for the visualization. Two attributes are chosen arbitrarily and then plotted with cluster centroids and with ellipsoids corresponding to covariance matrices of each cluster. Clusters are really integrated within each other and there isn't an obvious pattern to differentiate them with two dimensions. But light blues colored dots are a little bit accumulated more away than others.

	vh	wind	humidity	temp	ibh	dpg	ibt	vis
Cluster 1	2.286399	-0.532606	-0.501248	-0.502719	0.415335	-0.518082	-0.447892	-0.486497
Cluster 2	2.239400	-0.532452	-0.511666	-0.510692	1.910732	-0.530105	-0.497284	-0.450859
Cluster 3	2.309341	-0.533101	-0.506663	-0.500773	-0.215082	-0.534418	-0.420763	-0.485732
Cluster 4	2.273324	-0.532760	-0.512735	-0.507328	0.942830	-0.526564	-0.463784	-0.464758

Figure 5: Table of Cluster Centers: Cluster Centers for each attribute is calculates as in table.

At the Figure 5, since the data is standardized, there aren't huge gaps between cluster center values. To have more significantly distinct classes which are more far away than each other, it is logical to want to have clusters which has different values for each cluster and for each attribute. In that sense, from the table above, significant attribute is "ibh" which helps most to locate dots into different regions to have distinct clusters. Other attributes have really close values for different clusters, therefore they aren't as effective as "ibh" for clustering.

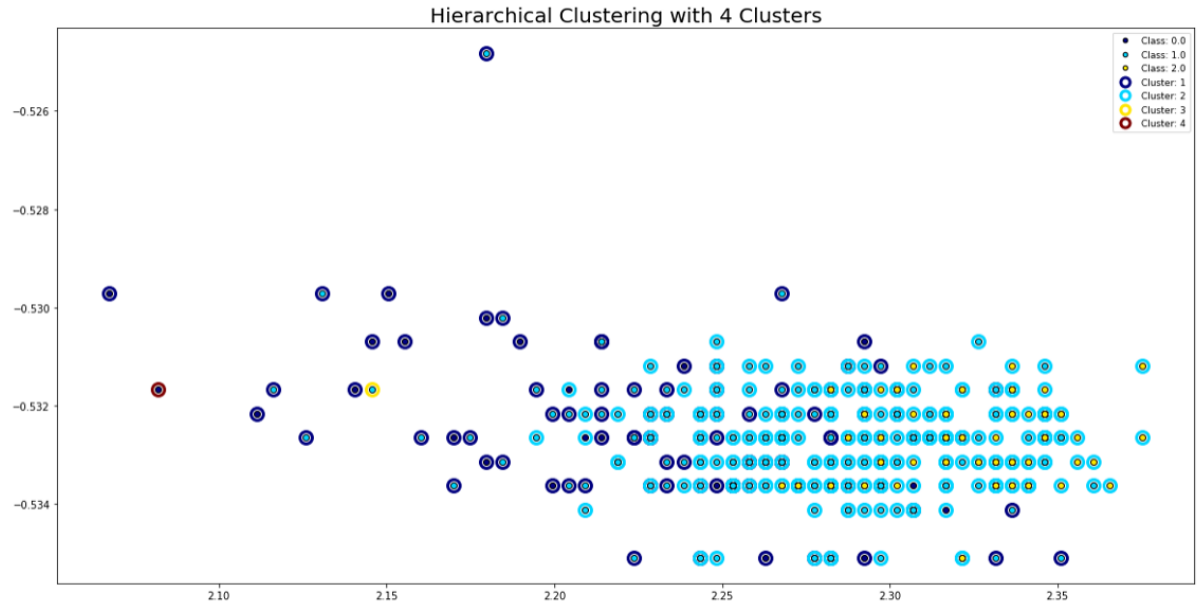

Figure 6: Hierarchical Clustering: "k" is optimized to 4. Plot shows the results on PC-1 and PC-2

Again, above plot is being created with the "clusterplot" function from the toolbox and data is projected onto the first two principal components. Number of clusters is optimized as 4. Other problems related to PCA still exists like important information is lost. Therefore it is again, hard to comment clear results from the plot above. From the above plot, onto the PC-1 and PC-2 space, as the dendogram yielded, clustering data into 2 or 4 is more reasonable and this can be seen from the Figure 6 as well. Mainly there are dark blue and light blue cluster indices which mostly identifies 2 clusters. But this might be happening because of PCA as well.

Question 3 :

	Rand	Jaccard	NMI
GMM	0.505038	0.249029	0.152491
Hierarchical	0.536447	0.414014	0.166287

Figure 7: Clustering Comparison acc. to Performance Measures: Performance Measures are as follows; Rand: Random Index, Jaccard: Jaccard Similarity, NMI: Normalized Mutual Information

To compare the GMM and Hierarchical Clustering methods, at Figure 7, table of clustering comparison is created. (Like it is aforementioned, labels are created as "low", "medium" and "high" from the calculations of standard deviation and mean. This is just a way to create classes from regression problem.) Output variable labels aren't clustered good for both of the models. For a good clustering, these performance measures should be close to 1. But none of them close to 1. Between performance measures, Hierarchical Clustering performed better, since its performance measure values are closer to 1. One suggestion to have better clustering is as follows; instead of labeling into 3 categories, our output data might be labeled into 2 or 4 categories. Maybe, after that process, these clustering methods, would have better results in terms of performance measures.

2 Outlier detection/Anomaly detection

Question 1 and Question 2 :

According to three methods used, there are likely outliers in the LA ozone data-set. The degree of likelihood of certain values are in terms of density is determined by using distance between samples. The samples were sorted by estimated density in ascending order for the all of three methods. In addition to that, the lowest 20 values were plotted on bar graphs. A large jump in density to the next lowest value and a combination of a low absolute density denote a potential outlier. According to the Gaussian Kernel density estimates in Figure 8, the least 2 values have densities extremely close to 0. Therefore they hava a large distance to the other samples. Also, between samples 15 and 16 in Figure 8, there is a considerable jump in the density. This might mean that all samples lower than the value given by sample could be outliers.

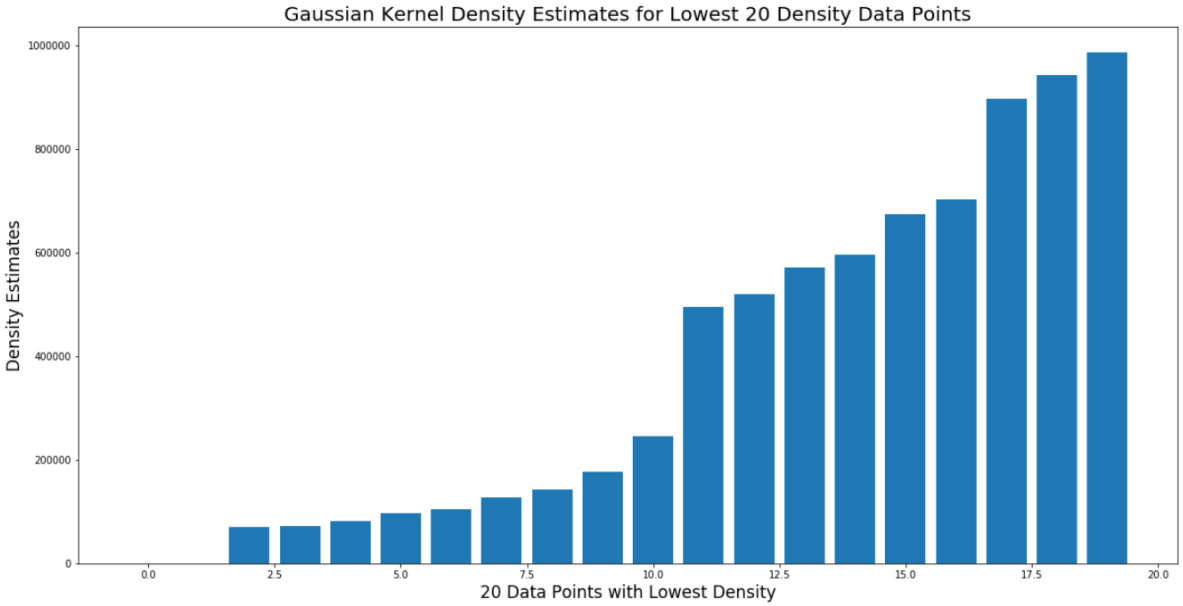

Figure 8: Gaussian Kernel Density

KNN density is used as a next metric for outlier detection and the indicators of an outlier are very similar. As it can be seen from the Figure 9, the lowest 2 samples might be outliers since there are large jumps in density between them and next lowest density samples. After sample 2, the graphs seem to be gradually rising and this can mean that the other samples are admirably valid.

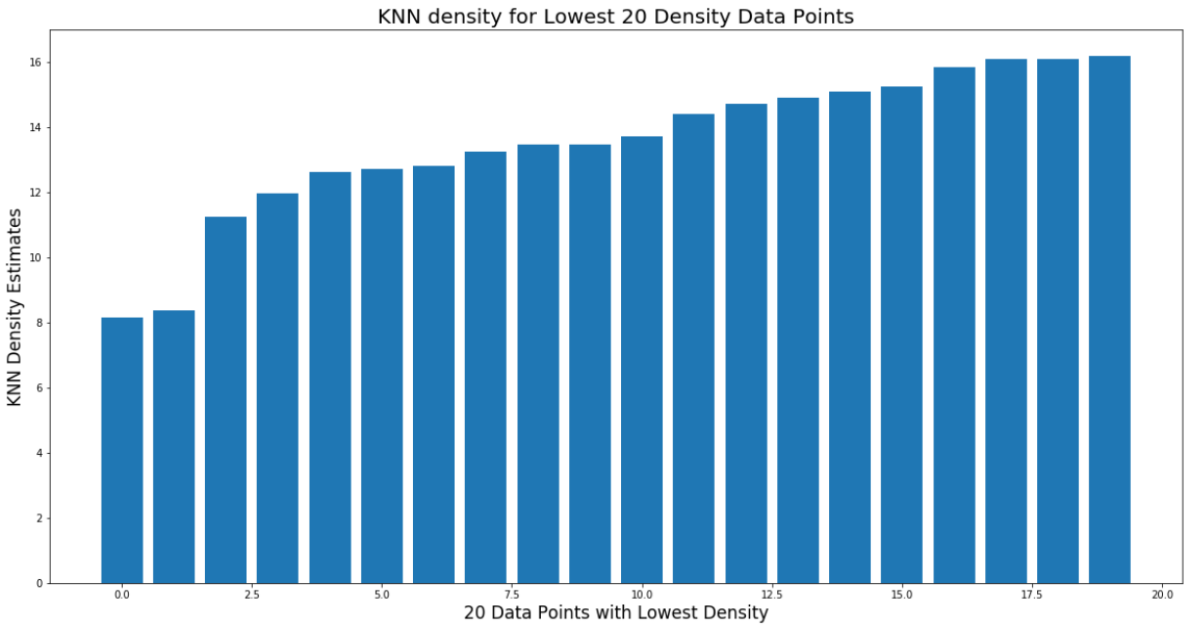

Figure 9: KNN Density

The KNN average relative density used as a final metric for outlier detection. This method takes into account the densities of the closest K neighbours to the sample being evaluated and generate a value in relation to those values. Therefore, it is smoother than the basic KNN method and it is more difficult to classify a sample as an outlier. In this regard, the graph in the 10 shows that the lowest relative density value might be an outlier as its value is far lower than other values.

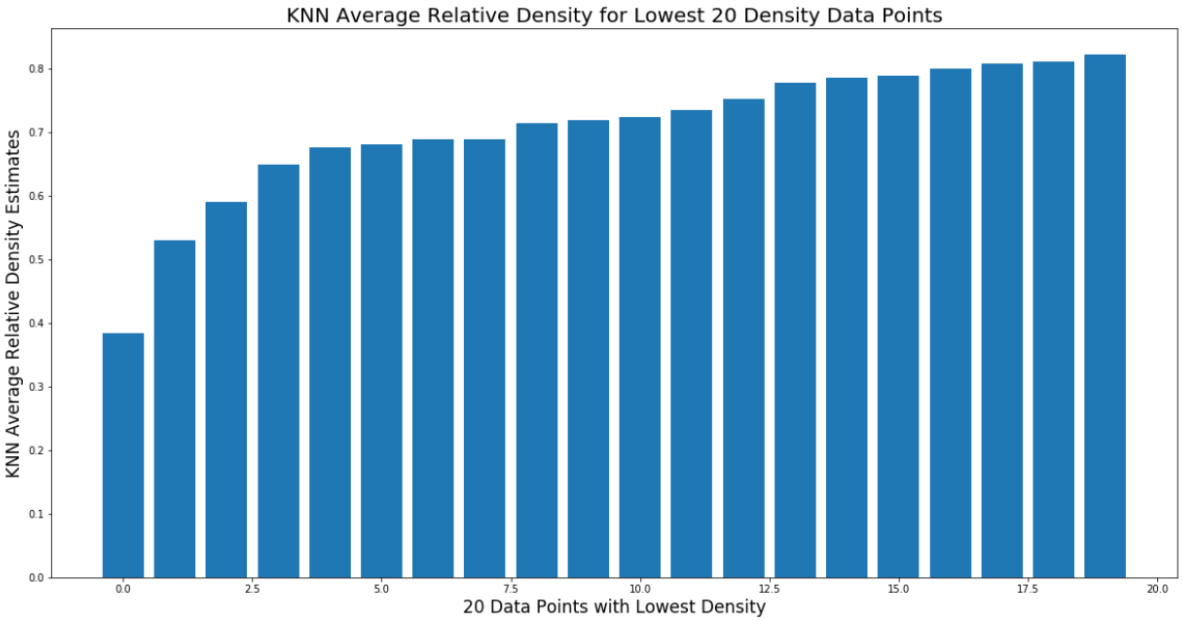

Figure 10: KNN Average Relative Density

3 Association mining

Question 1:

In this part, associations among the attributes of our data is examined based on association mining. Before running Apriori algorithm there are steps to be followed. First, **X** matrix and **y** column is appended to obtain all data. Since our data doesn't contain categorical variables or binary variables, one out of K coding isn't applied. All of the attributes are continuous. To binarize the attributes, a threshold value is decided for the continuous variables to obtain the percentiles of from 0^{th} to 50^{th} and from 50^{th} to 100^{th}. Since the percentiles are used the best fit threshold value is median value of each attribute. All of the attributes are binarized. For example, ozone attribute has been binarized as; `ozone_`0^{th}-50^{th} and `ozone_`50^{th}-100^{th}. Median values of ozone is 10. In the column of `ozone_`0^{th}-50^{th} has values which are "1" if at the same row, ozone column has the values less than or equal to 10. If the value in ozone column is greater than 10, it is "0" in the `ozone_`0^{th}-50^{th} column.

	{X} ->	{Y}	Support (>0.3)	Confidence (>0.95)
0	[ozone_50th-100th, temp_50th-100th, vh_50th-10...	[ibt_50th-100th]	0.315152	0.981132
1	[temp_50th-100th, vh_50th-100th, vis_0th-50th]	[ibt_50th-100th]	0.300000	0.980198
2	[vh_0th-50th, ibh_50th-100th]	[ibt_0th-50th]	0.324242	0.972727
3	[temp_0th-50th, ozone_0th-50th, ibh_50th-100th]	[ibt_0th-50th]	0.321212	0.963636
4	[ozone_50th-100th, vh_50th-100th, ibt_50th-100th]	[temp_50th-100th]	0.315152	0.962963
5	[temp_50th-100th, vh_50th-100th]	[ibt_50th-100th]	0.363636	0.960000
6	[temp_0th-50th, ibh_50th-100th]	[ibt_0th-50th]	0.345455	0.957983
7	[ibh_0th-50th, vh_50th-100th]	[ibt_50th-100th]	0.309091	0.953271

Figure 11: Dataset association rules sorted by confidence

Question 2

It seems like there is reliable relationships among the attributes. For instance, as it can be seen from Figure 11, it is almost the case that if temp is higher than median(between 50 and 100), vh is higher than median(between 50 and 100), and vis is lower than median(between 0 and 50), then ibt will also be higher than median(between 50 and 100) for the sample. These associations can be very useful in either supporting previously discovered meteorological phenomena or suggesting potential new ones. In addition to that, finding these relationships could allow researchers to fill in missing attributes with values that make sense for the purpose of certain machine learning methods. Yet, that is not to suggest that these fabricated values should be regarded as sound data.