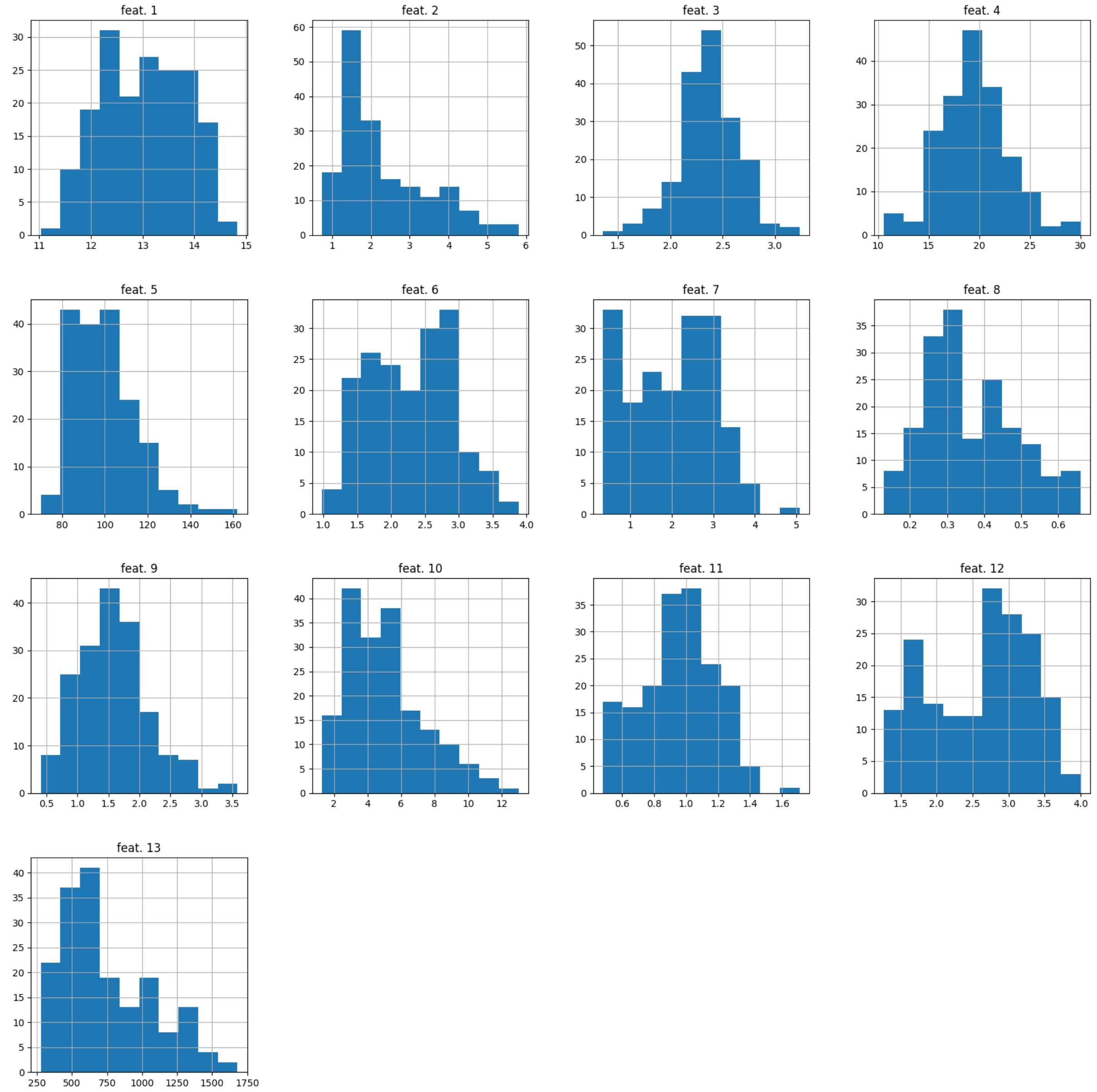
Report of wine data set

We are provided with multivariate data with real and integer values. Each wine has class represented by values from 0 to 2. There are 178 instances of wines, each with 13 attributes and there are no missing values.

To improve clarity of reading this report I will not be using name of features but I will be using number of column passed in data set.

feat. 1	alcohol	feat. 8	nonflavanoid phenols
feat. 2	malic acid	feat. 9	proanthocyanins
feat. 3	ash	feat. 10	color intensity
feat. 4	alcalinity of ash	feat. 11	hue
feat. 5	magnesium	feat. 12	od280/od315 of diluted wines
feat. 6	total phenols	feat. 13	proline
feat. 7	flavanoids		

Firstly, I have checked whether data is normally distributed using histograms.



As we can see the only features which are close to be normally distributed are feature 3rd, 4th and 9th but they are not. The cause of that is that the data set is real and like the most things in the world it is not normally distributed.

Thus we shall not use mean to calculate center point but use median instead.

Stats	feat. 1	feat. 2	feat. 3	feat. 4	feat. 5	feat. 6	feat. 7	feat. 8	feat. 9	feat. 10	feat. 11	feat. 12	feat. 13
count	178	178	178	178	178	178	178	178	178	178	178	178	178
mean	13,00	2,34	2,37	19,49	99,74	2,30	2,03	0,36	1,59	5,06	0,96	2,61	746,89
median	13,05	1,87	2,36	19,50	98,00	2,36	2,14	0,34	1,56	4,69	0,97	2,78	673,50
STD	0,81	1,12	0,27	3,34	14,28	0,63	1,00	0,12	0,57	2,32	0,23	0,71	314,91
mode	12,37	1,73	2,28	20,00	88,00	2,20	2,65	0,26	1,35	2,60	1,04	2,87	520,00
mad	0,69	0,92	0,21	2,60	11,00	0,54	0,86	0,10	0,45	1,84	0,19	0,61	259,33
var	0,66	1,25	0,08	11,15	203,99	0,39	1,00	0,02	0,33	5,37	0,05	0,50	99166,72
min	11,03	0,74	1,36	10,60	70,00	0,98	0,34	0,13	0,41	1,28	0,48	1,27	278,00
max	14,83	5,80	3,23	30,00	162,00	3,88	5,08	0,66	3,58	13,00	1,71	4,00	1680,00
range	3,80	5,06	1,87	19,40	92,00	2,90	4,74	0,53	3,17	11,72	1,23	2,73	1402,00
25%	12,36	1,60	2,21	17,20	88,00	1,74	1,21	0,27	1,25	3,22	0,78	1,94	500,50
75%	13,68	3,08	2,56	21,50	107,00	2,80	2,88	0,44	1,95	6,20	1,12	3,17	985,00
IQR	1,32	1,48	0,35	4,30	19,00	1,06	1,67	0,17	0,70	2,98	0,34	1,23	484,50

Median is center point of the set (since the set is not normally distributed).

Standard deviation is close to 0 only for the 3rd, 8th and 11th feature, so the set is quite spread out and quite far from the mean.

Variance is not equal to squared standard deviation for all features. Moreover values are high so it implicates that points are quite far in sense of distance from median especially for the last feature so the data is quite spread out as we observed from the standard deviation.

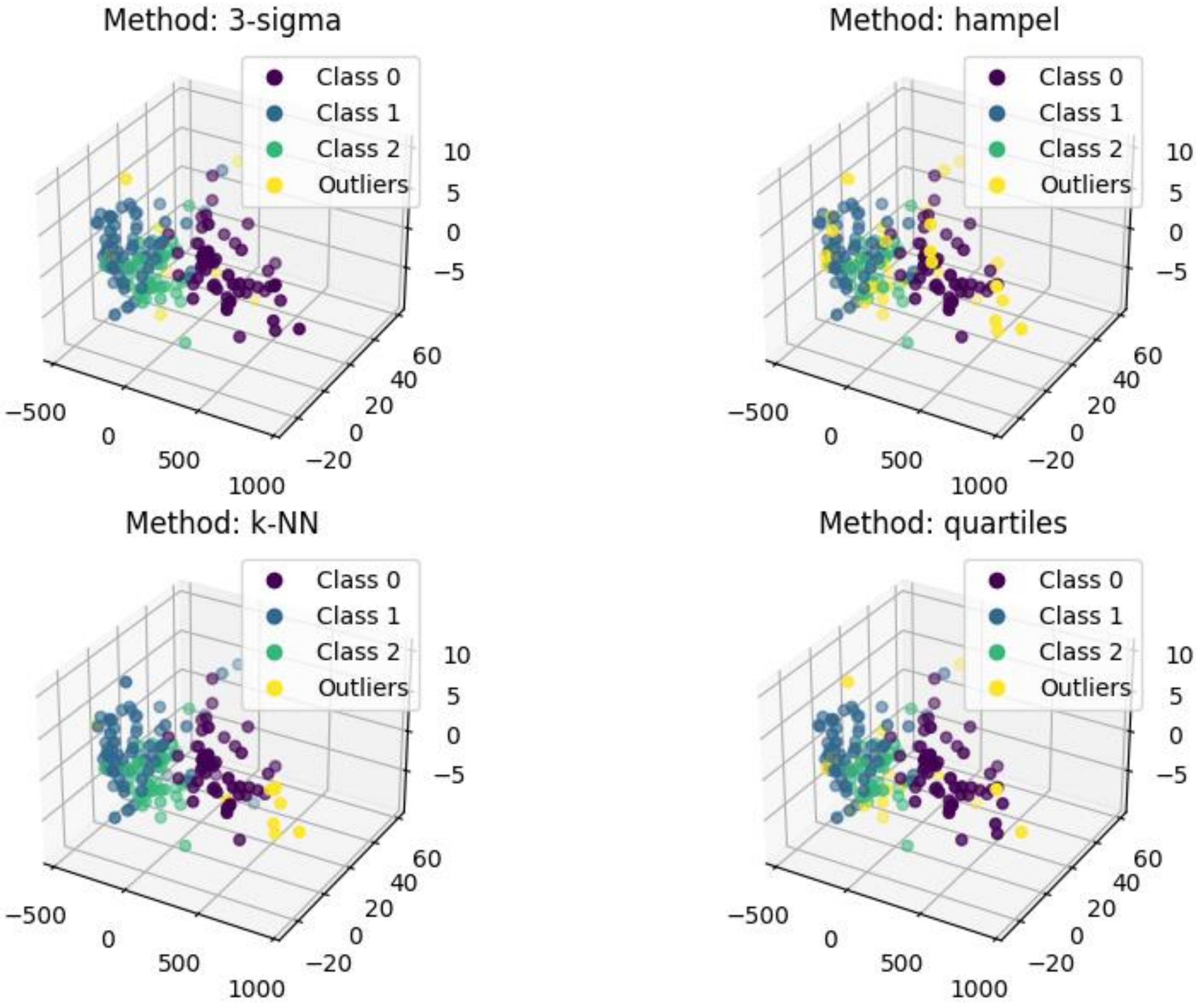
The point which occurs the most is mode.

Variability of data set is high as we have seen that from variance, standard deviation and also we will see that from the plots.

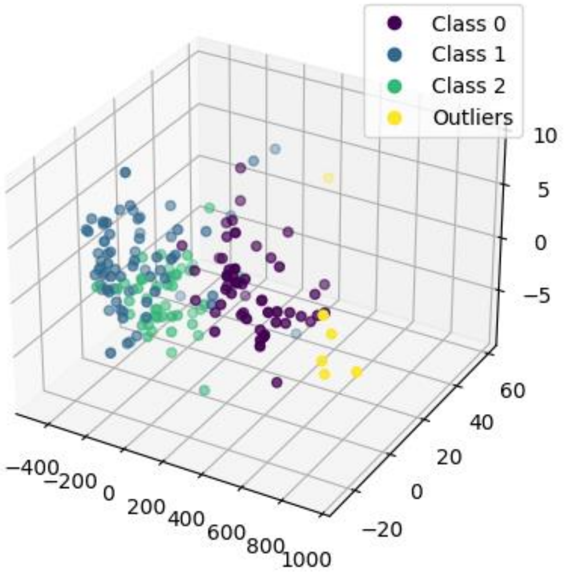
The range also verifies that our set is spread out as well as IQR does.

Using Hampel, k-Nearest Neighbors algorithms, 3-Sigma rule and quartiles and then taking intersection of results from k-NN and Hampel algorithms I have found 6 outliers of indices {3, 10, 14, 18, 31, 95}. I have decided to use particularly these algorithms since the data is not normally distributed and also results from these algorithms are the most common I have observed (chatGPT suggested to do so as well).

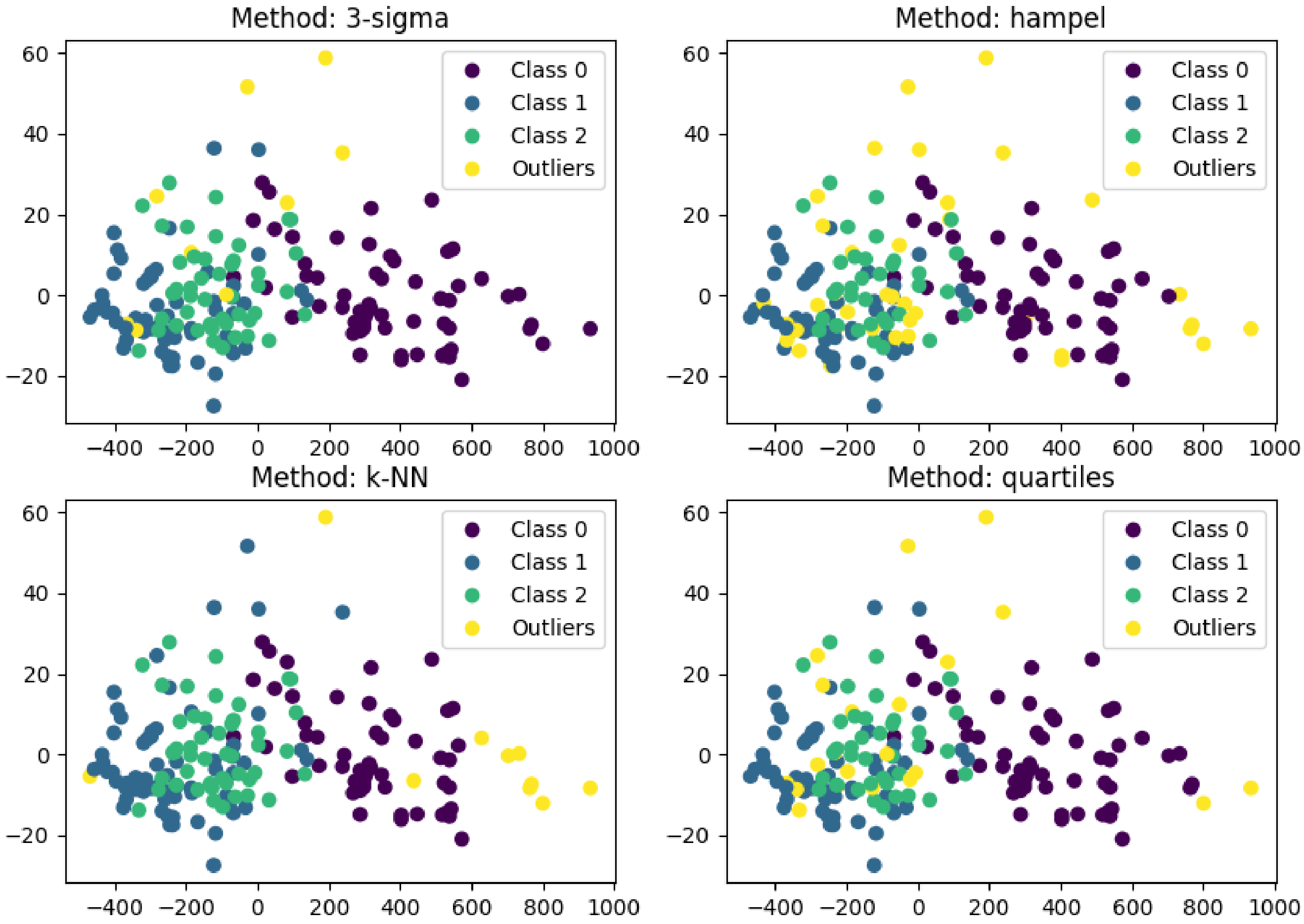
Using PCA algorithm to reduce dimensionality to 3 dimensions I have visualized the data set and also marked the outliers which are the results from each algorithm.



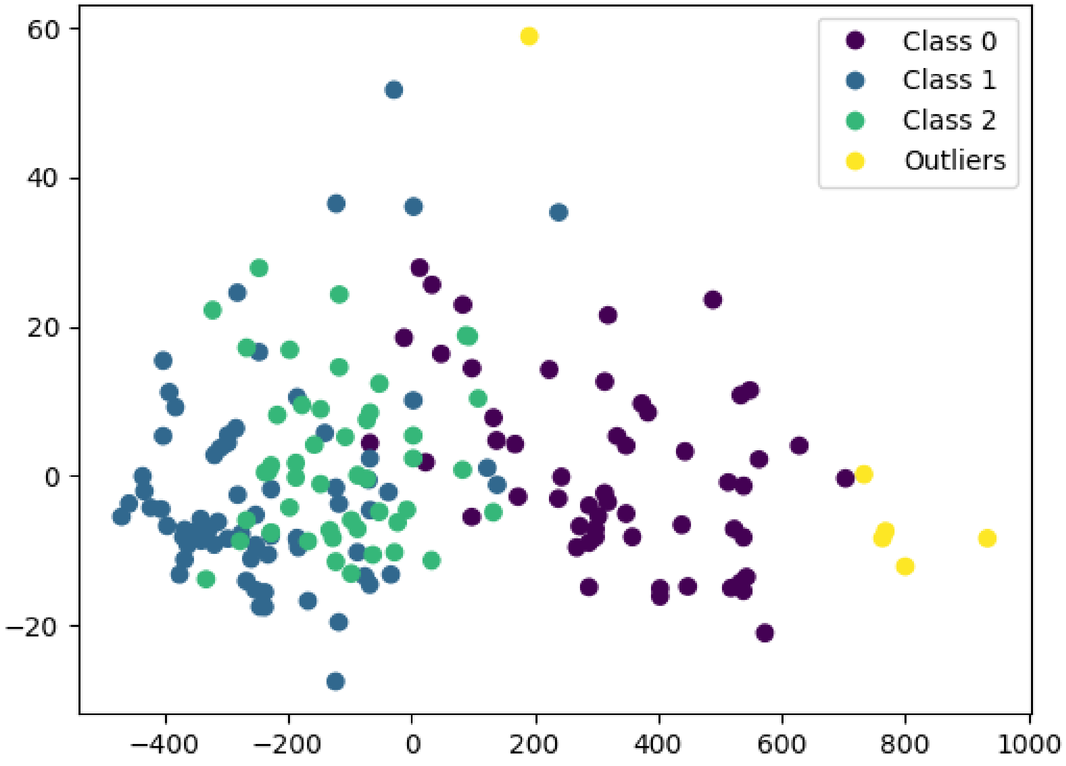
This is a plot with the final result:



Same results using PCA to reduce to 2 dimensions:

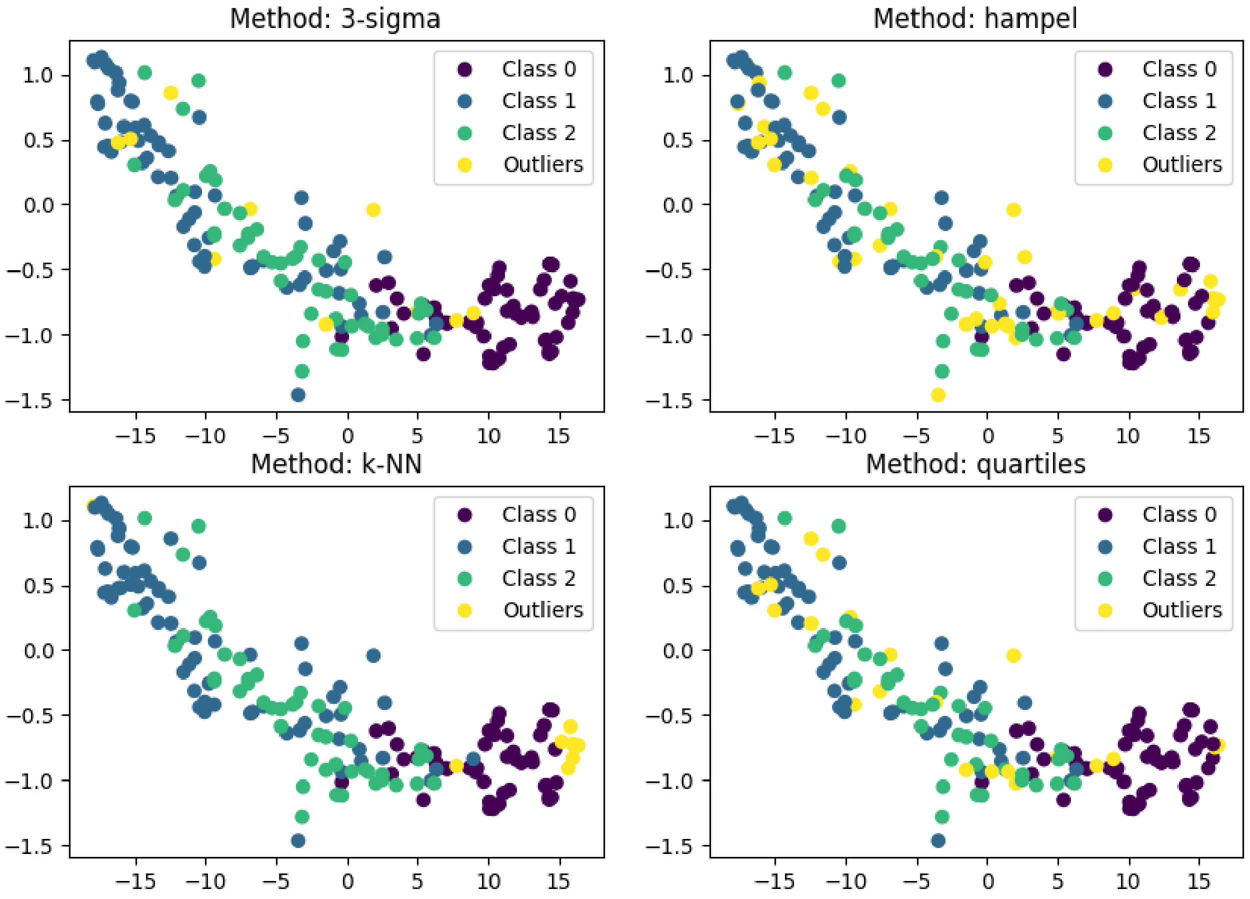


And this is the final result:

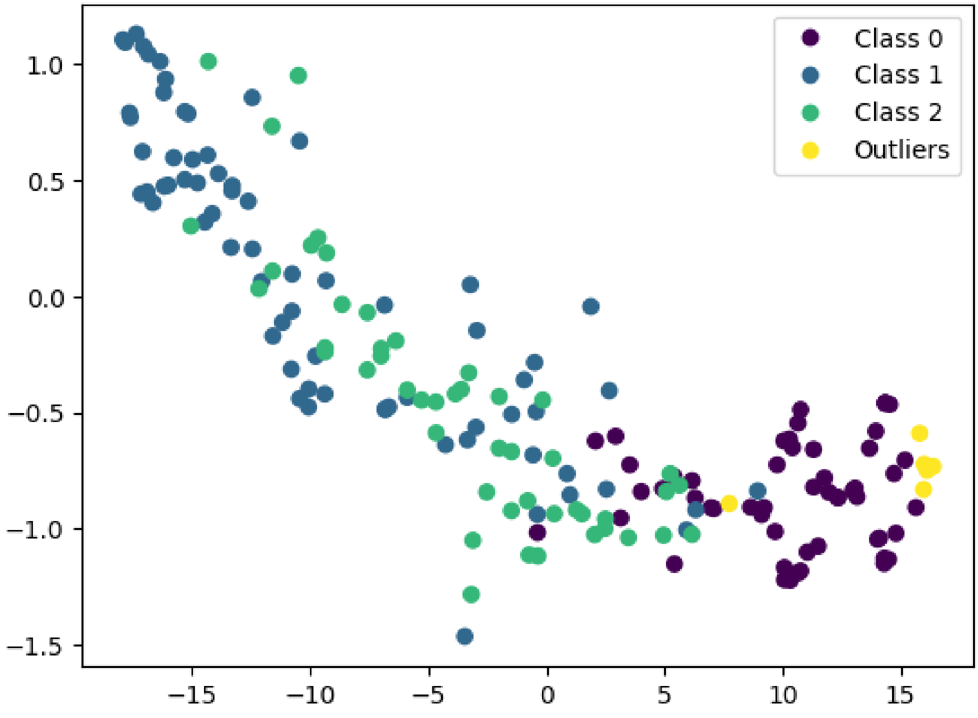


In my opinion the data set should be visualized in 3 dimensions because in 2 dimensions variables layer on each other.

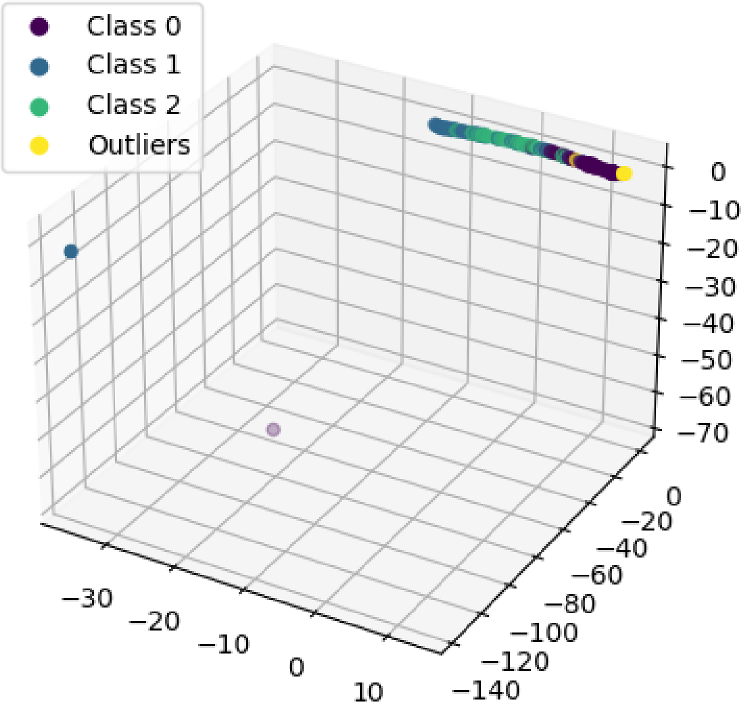
These are the results of reducing dimensionality to 2 and 3 using t-SNE algorithm:



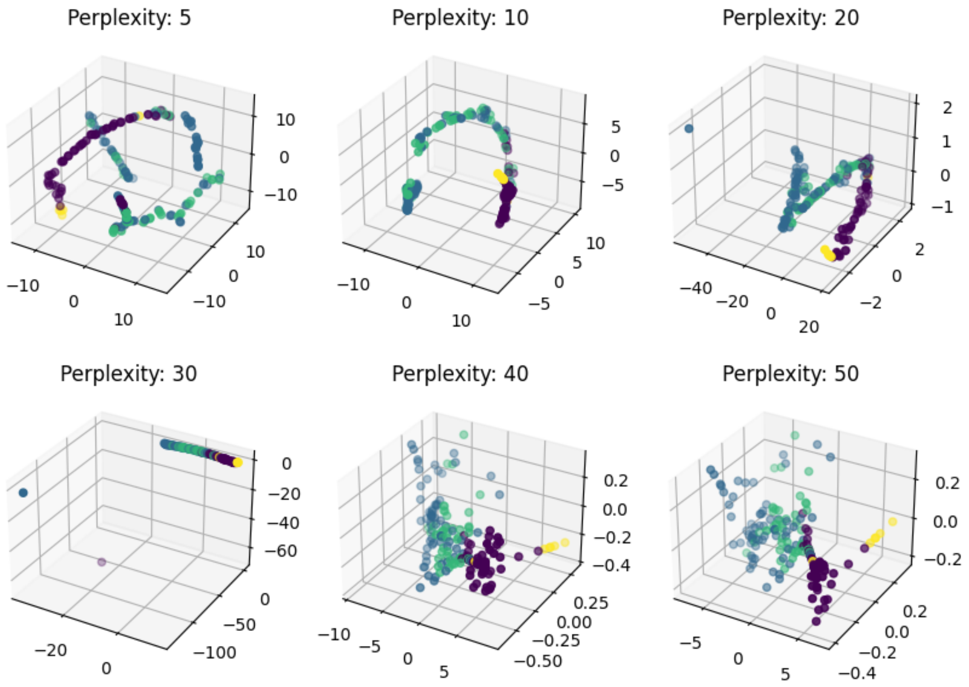
Final result:



While plotting t-SNE result on 3 dimensions I have encountered a problem:



I have asked chatGPT what could be the problem and I have got an answer that t-SNE is trying to keep the data structure so t-SNE packed up almost every point in one line and had problems with scaling it. I have tried plotting the same data for different perplexities variable in t-SNE algorithm trying to get some satisfying results and here is what I have got.


Perplexity: 5

Perplexity: 10

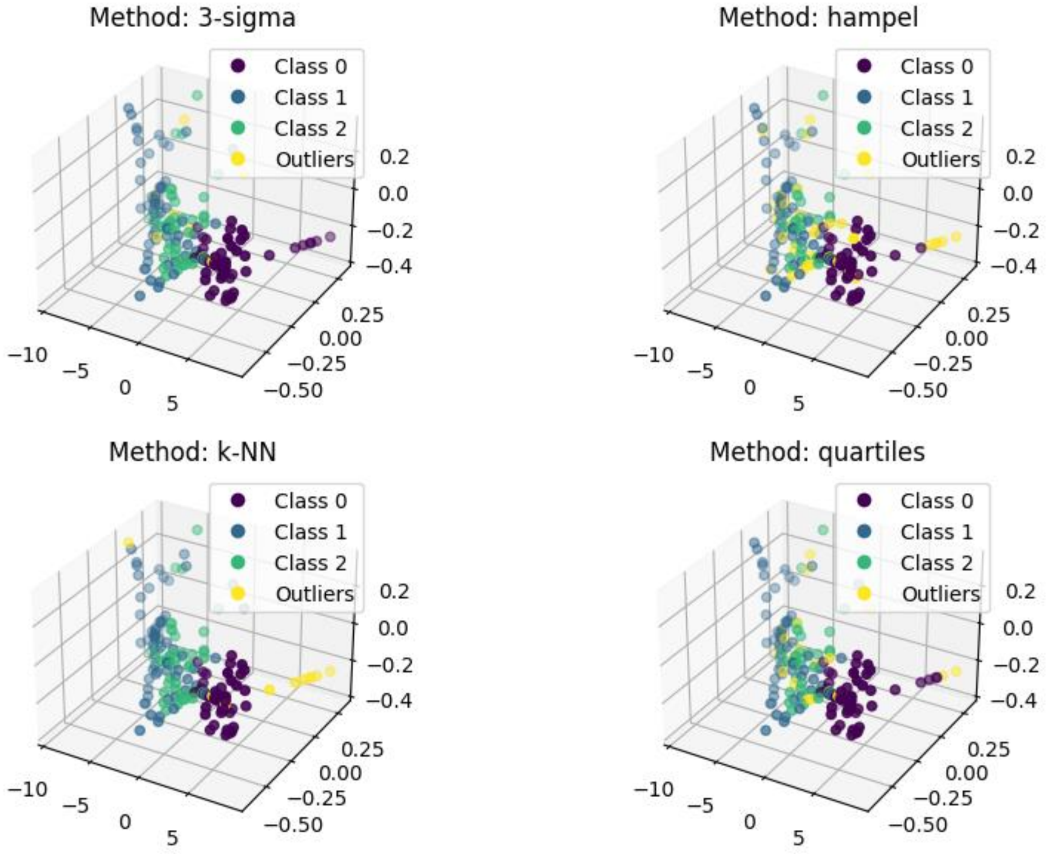
Perplexity: 20

Perplexity: 30

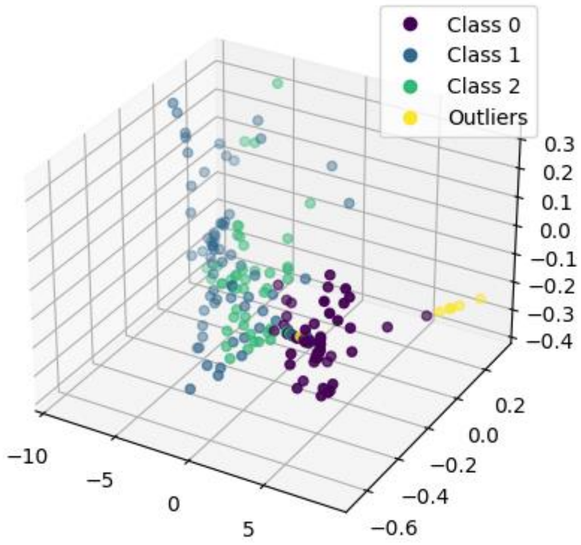
Perplexity: 40

Perplexity: 50

Plots for perplexity equal to 40 and 50 looks fine for our predictions from the second page. I will use plot with perplexity equal to 40 because it is more lucid than the last one, in my opinion.



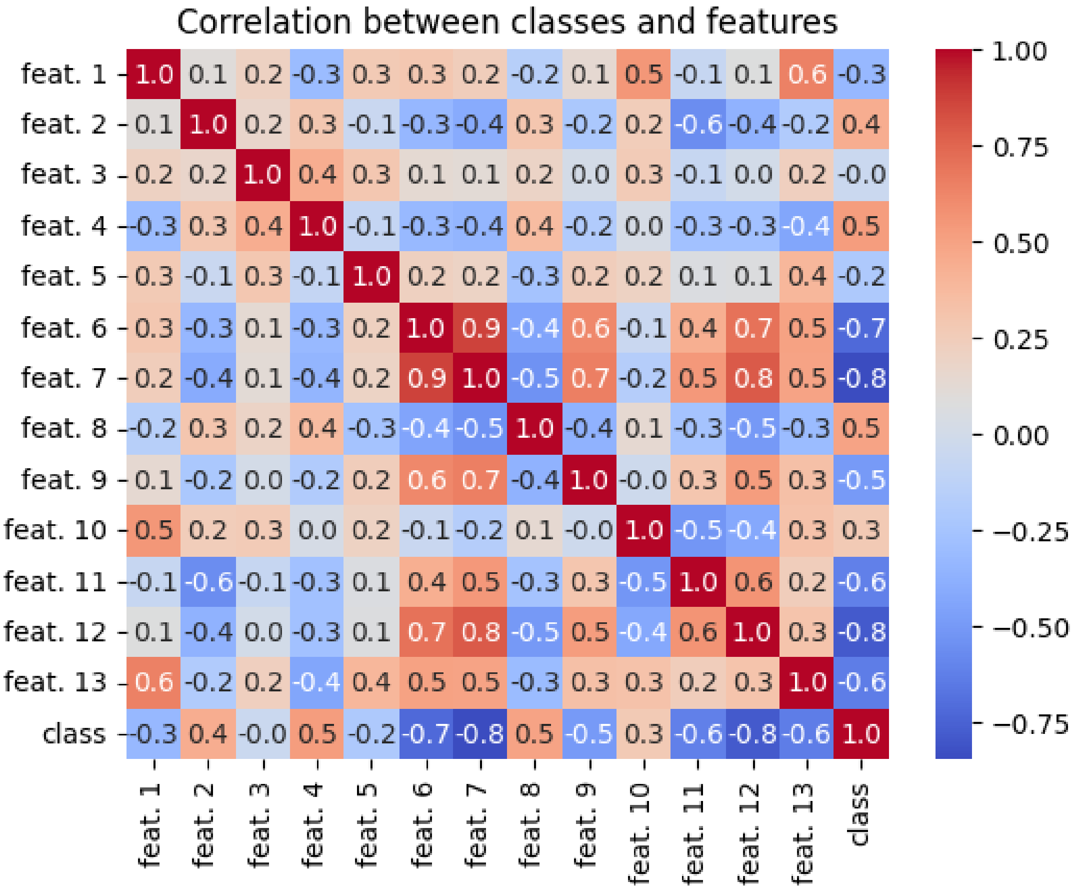
Final result:



The same goes for t-SNE in preferred dimensionality as in PCA.

In conclusion of reducing and visualizing data set, in 2D most of the points layer on each other so the best result we can obtain from 3D visualization. t-SNE has some problems with keeping data structure and the result of that is it is returning packed up points in one line with 2 points being separated from each other and the line. This problem can be resolved by searching best perplexity for t-SNE algorithm.

Keeping in mind that finding correlations between features is not the main task, I think finding correlations between classes and features will be quite useful thing in terms of classification.



	feat. 1	feat. 2	feat. 3	feat. 4	feat. 5	feat. 6	feat. 7	feat. 8	feat. 9	feat. 10	feat. 11	feat. 12	feat. 13	class
feat. 1	1.0	0.1	0.2	-0.3	0.3	0.3	0.2	-0.2	0.1	0.5	-0.1	0.1	0.6	-0.3
feat. 2	0.1	1.0	0.2	0.3	-0.1	-0.3	-0.4	0.3	-0.2	0.2	-0.6	-0.4	-0.2	0.4
feat. 3	0.2	0.2	1.0	0.4	0.3	0.1	0.1	0.2	0.0	0.3	-0.1	0.0	0.2	-0.0
feat. 4	-0.3	0.3	0.4	1.0	-0.1	-0.3	-0.4	0.4	-0.2	0.0	-0.3	-0.3	-0.4	0.5
feat. 5	0.3	-0.1	0.3	-0.1	1.0	0.2	0.2	-0.3	0.2	0.2	0.1	0.1	0.4	-0.2
feat. 6	0.3	-0.3	0.1	-0.3	0.2	1.0	0.9	-0.4	0.6	-0.1	0.4	0.7	0.5	-0.7
feat. 7	0.2	-0.4	0.1	-0.4	0.2	0.9	1.0	-0.5	0.7	-0.2	0.5	0.8	0.5	-0.8
feat. 8	-0.2	0.3	0.2	0.4	-0.3	-0.4	-0.5	1.0	-0.4	0.1	-0.3	-0.5	-0.3	0.5
feat. 9	0.1	-0.2	0.0	-0.2	0.2	0.6	0.7	-0.4	1.0	-0.0	0.3	0.5	0.3	-0.5
feat. 10	0.5	0.2	0.3	0.0	0.2	-0.1	-0.2	0.1	-0.0	1.0	-0.5	-0.4	0.3	0.3
feat. 11	-0.1	-0.6	-0.1	-0.3	0.1	0.4	0.5	-0.3	0.3	-0.5	1.0	0.6	0.2	-0.6
feat. 12	0.1	-0.4	0.0	-0.3	0.1	0.7	0.8	-0.5	0.5	-0.4	0.6	1.0	0.3	-0.8
feat. 13	0.6	-0.2	0.2	-0.4	0.4	0.5	0.5	-0.3	0.3	0.3	0.2	0.3	1.0	-0.6
class	-0.3	0.4	-0.0	0.5	-0.2	-0.7	-0.8	0.5	-0.5	0.3	-0.6	-0.8	-0.6	1.0

Let’s focus on last row. We can see that the classes have high negative correlation with feature 7th, 8th and 12th.

We should see that at the end of this report that some classes have values of those features in a very close range.

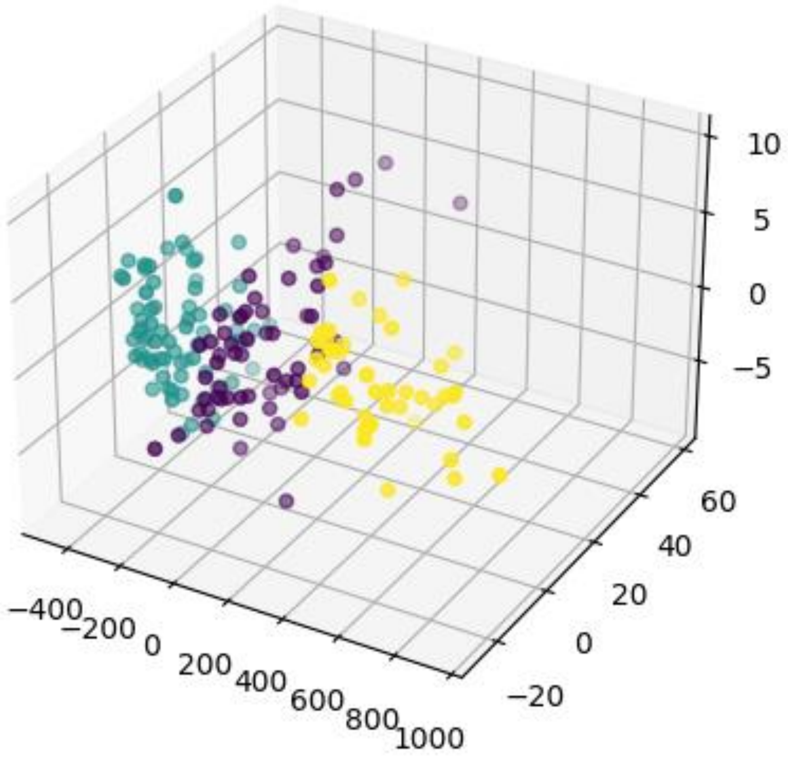
Features 7 with 6, and 12 with 7 are in positive high correlation (threshold = 0.7).

Features 11 with 2, 8 with 7, 12 with 8, 11 with 10 are in negative high correlation (threshold = -0.5).

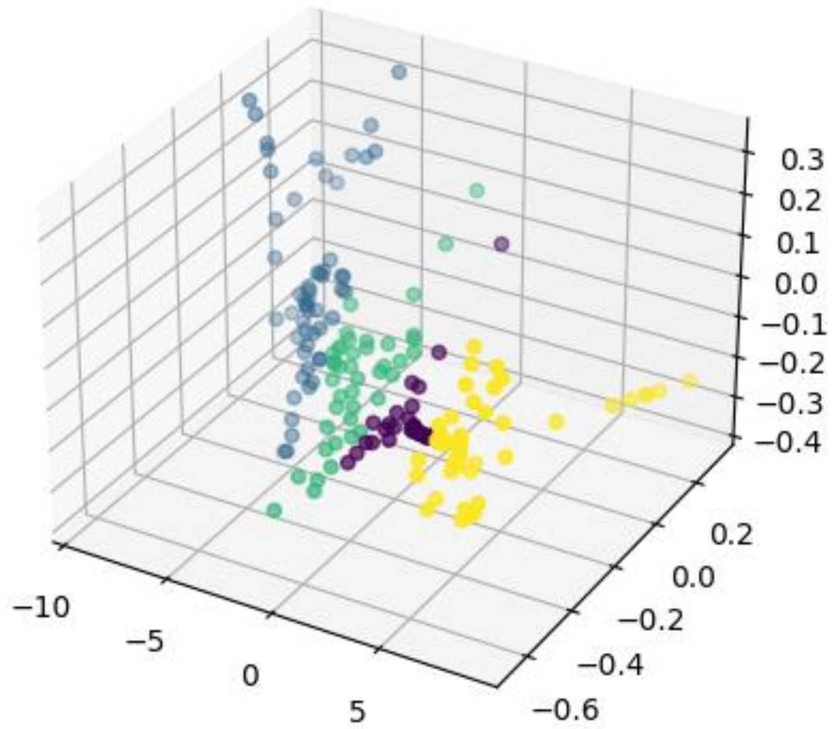
In conclusion of statistical summary of data set we have obtained that the data is spread out, not normally distributed and variability of data is high. We can find 6 instances of wines which outlie from the rest. There are 3 features with high positive correlation and 5 features with high negative correlation. There is also a correlation between class and feature 7th, 8th and 12th.

Using KMeans algorithm I have conducted clustering and the results were as I expected. Number of clusters for raw data and for preprocessed data with PCA algorithm does not differ. In both results there are 3 clusters.

This is a result:

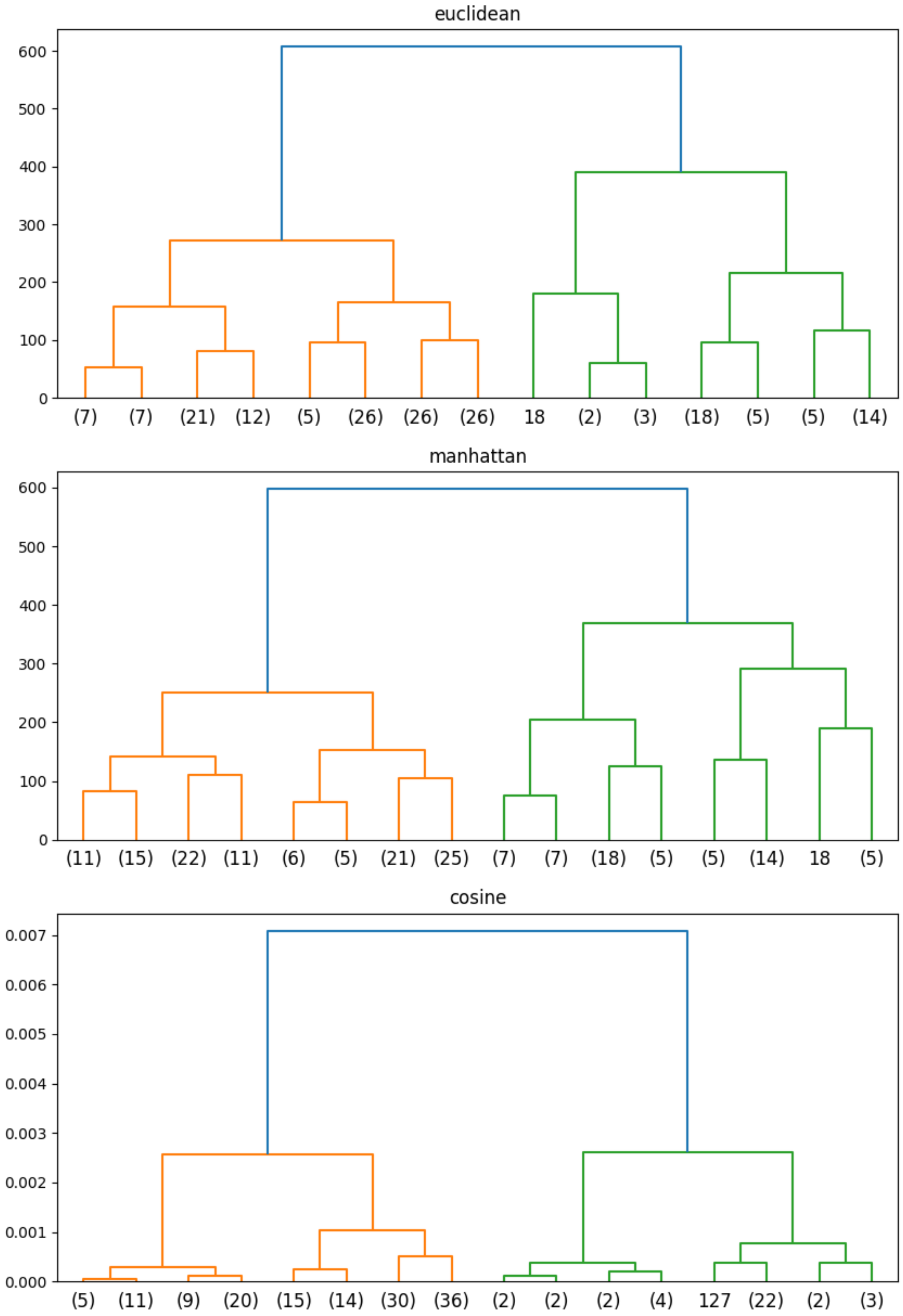


For data preprocessed with t-SNE the best solution was 4 clusters.



The best result was for t-SNE in 3D (I also did it in 2D) for which rand index was equal to 0.745. From here we can presume that clustering results in wine data set is going to be weak in terms of matching true classes.

Results of Agglomerative Clustering algorithm on raw data set with 3 different metrics (Euclidean, Manhattan, Cosine):



Those are the best results after grid search for the best number of clusters. For Euclidean it is 4 clusters, for Manhattan it is 6 and for Cosine it is 11. Those results are weak as we have predicted because rand index for this metrices are 0.709 for Euclidean, 0.697 for Manhattan and the best one 0.721 for Cosine.

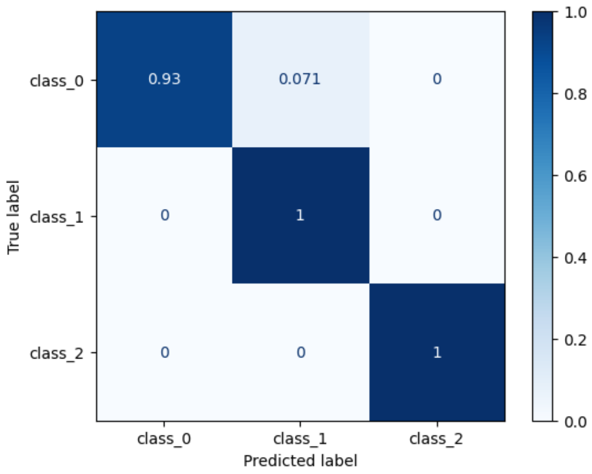
In conclusion for clustering, the data is not well separated and weak in terms of matching true class labels .

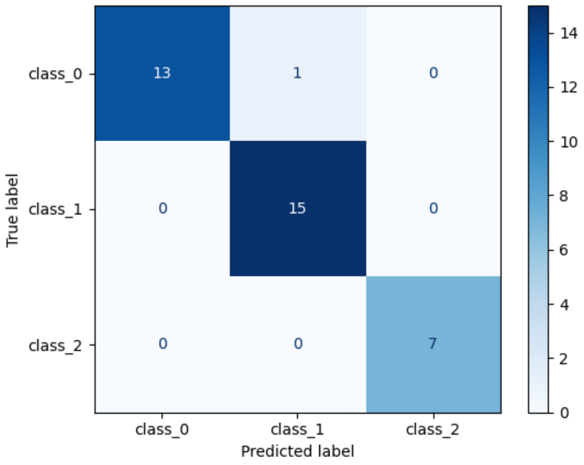
I have split the set into training and testing in 4:1 ratio and shuffled it to get the best results of classification.

I have conducted classification using Naïve Bayes, k-Nearest Neighbors (with different metrices) and Decision Tree algorithms.

For Naïve Bayes result is pretty high because accuracy score is equal approximately to 0.972.

This is a visualization of result:



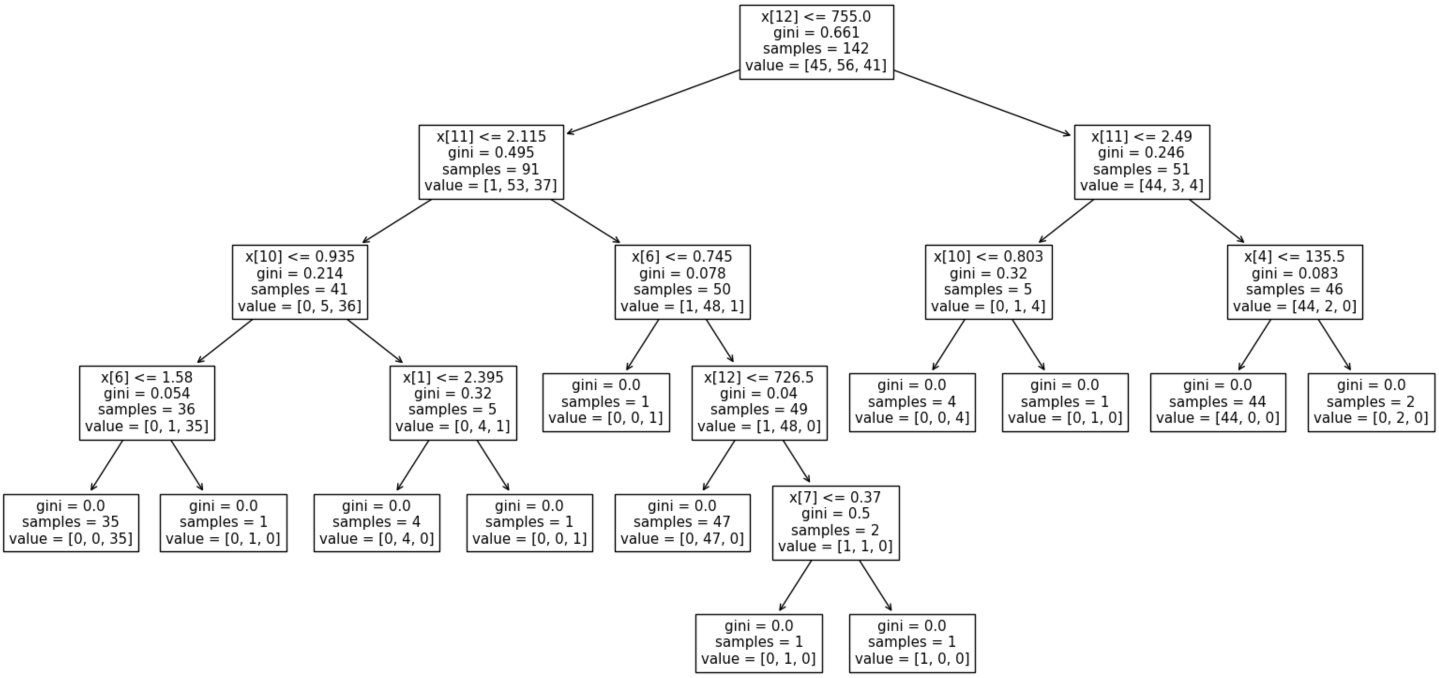


As we can see the only point from testing set was misclassified as class 1.

Mean results of k-NN algorithm for each metric are 0.708 for Euclidean, 0.759 for Manhattan and 0.814 for Cosine. These results are weaker than Naïve Bayes because of the distribution of the data set.

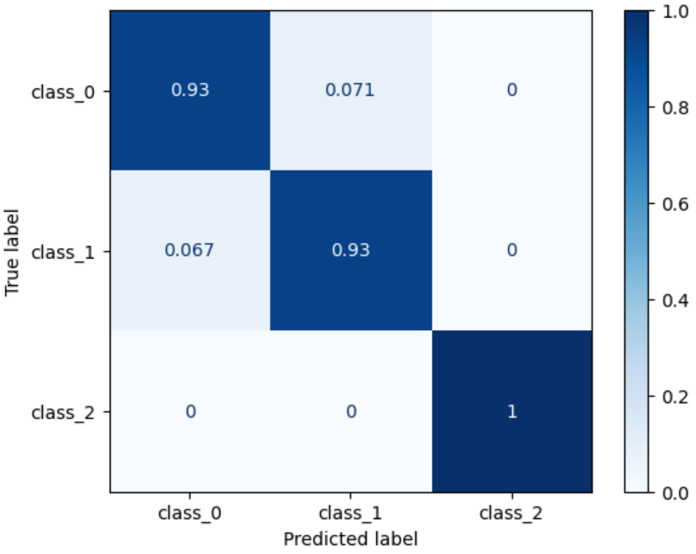
From Decision Tree Classifier we have got a result equal to 0.944.

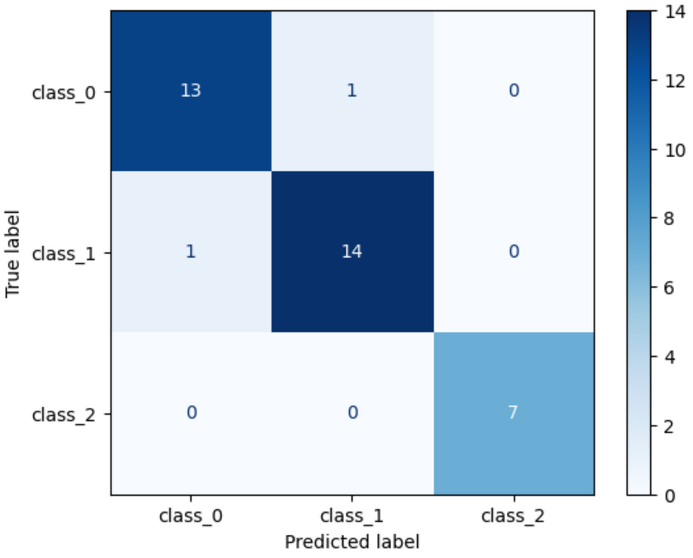
This is a visualization of decision tree:



For all nodes and leaves we can see that gini index shows us almost all features are well distributed and none of the features belong to only one class.

I have also visualized the confusion matrix to see exact results of how many instances of classes were misclassified.





There were only 2 misclassifications, one of class 0 and one of class 1.

In conclusion to classification of the data set is very well classified among the features and correlation with features did not influence very much classification as we could have seen in Decision Tree that gini index was below 0.5 for almost every node.

In conclusion, wine data set is well distributed, some features are close to be normally distributed and variability is high. There are 6 outliers. Using PCA and t-SNE to reduce dimensionality to 2D and 3D I have decided to use 3D because it is more lucid. Clustering results were relatively weak in terms of matching the true class labels. Classification results were very good for most of used algorithms.

Using chatGPT to check whether my results are correct, chatGPT confirmed my statistical summary and chose to use 2D rather than 3D visualization. Clustering results were the same as mine and the classification was as well. ChatGPT also proposed to use different algorithms in the future work with data set.

The source code is provided in directory along with copy of my chat with chatGPT.

In the source code there is also an attempt of using DBSCAN algorithm as chatGPT proposed to do so for the clustering which was quite different from what I have obtained from the algorithms I have presented here. There are also some visualizations in 2D for comparation purposes to 3D in each section.

Developed by Wiktor Szewczyk