Franciszek Ruszkowski w1787351
Data Science and Analytics

Clustering

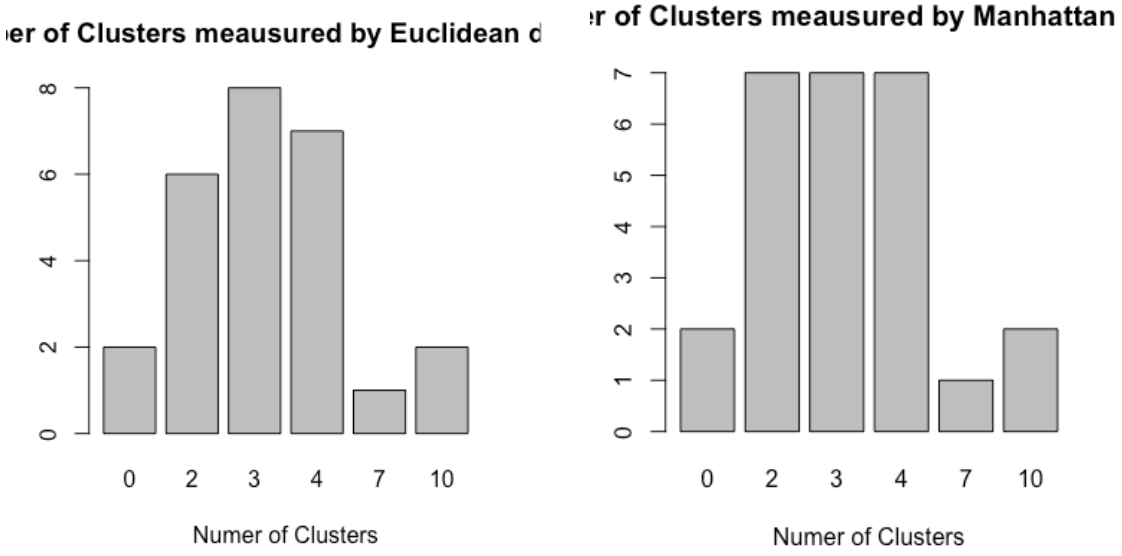
We look at the data and we can check for the correlations matrix which will indicate which variables are correlated. It is to avoid multicollinearity, and remove variables that have no impact on the class result. However, as it was said, we will use all 19 features in this particular example. The “Samples” feature will be removed however, as it is simply nothing else but the index column, therefore it can be omitted. Moreover, theoretically if we could remove features, we would have used principal components analysis in order to detect the features that can be removed in order to reduce dimensionality.

After that I began the preprocessing stage by outlier removal. I conducted a graphical visualization with boxplots. I used the Rosner test in order to detect multiple outliers out at once, with the test giving me true or false values - regarding if the datapoint is indeed an outlier or not. I chose that over a manual histogram method, as it is more precise. However, in some cases, when checking the (Skew Maxis , Skew maxis Max.L.Ra) the Rosner test gave False value - as if the outliers on the boxplots were not actually outliers. Therefore I proceeded to continue with the dataframe data_v4, but also compared my results with data_v8 (which is after removing the outliers purely on the argument that visually they significantly stood out from the rest. I conducted a correlation matrix, which was not that useful, since I was not supposed to remove any features, however it gave me a better understanding of the entire dataframe.

I normalized the dataframe, and also used scaling on every column one by one. The better approach in this case was scaling was more appropriate. If this was a linear regression task for example, I would have normalized the data as it provides a bell curve, however in this case it was not fully necessary.

In search for the optimal number of K clusters, I used the NbClust method, in order to give me the right suggestion. I did the NbClust using both euclidean and manhattan distances. For the Euclidean distance, it suggested that 2 or 3 or 4 clusters will be best, with a small indication that 3 will be the most appropriate decision. The Manhattan distance on the other hand, gave equally great chances for 2 or 3 or 4 being the optimal number.

These are results for NbClust i have obtained:

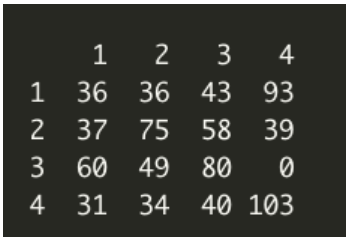


Therefore I proceeded to the next step, which is doing k means, for k=2 , k=3 , k=4. I compared the plot later on with fviz_cluster. I got the following results :
(between_SS / total_SS = 59.3 %), # (between_SS / total_SS = 55.2 %) , #(between_SS / total_SS = 42.7 %) for k=4,3,2 respectively.

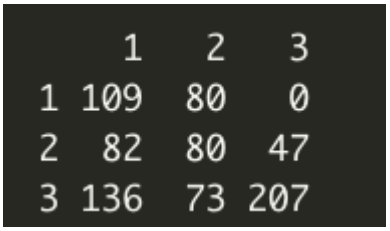
Now came the tricky part, which is comparing the results obtained with the original classes from the dataset. This is not straightforward, as the confusion matrix has to be an NxN matrix. The number of classes has to match the number of clusters. I looked at a plot with the scaled X and original classes as inputs and saw that opel and saab can be easily merged into one class. So for k = 3 I merged those two into one category and proceeded doing the confusion matrix. Analogly, I did the same thing for k = 2 with making ‘bus’ and ‘van’ one category. Moreover, the confusion matrix can be far from correct, with a low accuracy, as the clusters “1” or clusters “2” are assigned as random. Therefore it can mismatch the Class categorical variable which corresponds to 1 or corresponds to 2. Moreover, K-Means clustering is an unsupervised algorithm, making the comparison to the original class values somewhat unusual.

Nonetheless, with putting a lot of focus on labeling the categorical variables accordingly I managed to obtain 3 confusion matrices. Accuracy 0.3612 Accuracy, 0.4865 Accuracy 0.6978]

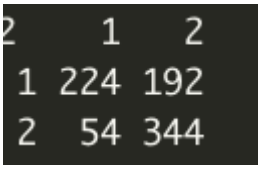
These are the 3 confussion matrices I obtained for k=4 k=3 and k=2



	1	2	3	4
1	36	36	43	93
2	37	75	58	39
3	60	49	80	0
4	31	34	40	103



	1	2	3
1	109	80	0
2	82	80	47
3	136	73	207



2	1	2
1	224	192
2	54	344

With accuracy being equal to : 36,12% ; 48,65% ; 69,78% respectively.

Based on the accuracy k = 2 is the most appropriate model, given the big similarities between some of the classes. Although the sum of squares does not make it look like the best solution, the accuracy is important, along with some other statistics : Sensitivity : 0.8058, Specificity : 0.6418, Pos Pred Value : 0.5385, Neg Pred Value : 0.8643. Accuracy is the total number of correctly labeled classes. Recall is the proportion of actual positives identified correctly. Precision os the number of true positives divided by the number of true positives + false positives - essentially just focusing on the positive values. All measures are vital in classification.

Some more information on the final choice of k means (k = 2)
Cluster means:

	Comp	Circ	D.Circ	Rad.Ra
1	1.0793803	1.1066728	1.1523161	1.0414205
2	-0.5598279	-0.5739833	-0.5976565	-0.5401398

	Pr.Axis.Ra	Max.L.Ra	Scat.Ra	Elong
1	0.2568545	0.6459941	1.2257381	-1.1687378
2	-0.1332193	-0.3350492	-0.6357373	0.6061737

	Pr.Axis.Rect	Max.L.Rect	Sc.Var.Maxis	Sc.Var.maxis
1	1.2295496	1.0236493	1.1883365	1.2318659
2	-0.6377141	-0.5309226	-0.6163387	-0.6389155

	Ra.Gyr	Skew.Maxis	Skew.maxis	Kurt.maxis
1	1.0142560	-0.08584552	0.1078164	0.2294956
2	-0.5260507	0.04452436	-0.0559197	-0.1190294

	Kurt.Maxis	Holl.Ra
1	0.05111642	0.2001828
2	-0.02651187	-0.1038261