Applied data science coursework

Alberto Plebani
ap2387

Preliminary information

I run the entire code on a Ubuntu-22.04 WSL on my Huawei MateBook 16 laptop with Windows 11. To run the code I used Anaconda 23.9.0. I have allocated 8 GBs of RAM to the WSL, half of the total laptop's available RAM. The laptop has an AMD Ryzen 5 5600 Hz CPU with Radeon Graphics.

- Part 1 took 19 seconds to run with the `--features` flag, and 3 seconds without
- Part 2 took less than 1 second to run
- Part 3 took 19 seconds with the `--heatmap` flag and only 3 seconds without
- Part 4 took less than 4 seconds to run
- Part 5 took 5 seconds to run

Therefore, running all the steps require less than a couple of minutes.

1 Section A

In this section I will be presenting the work done on parts 1,2 and 3.

A-1

The density plot for the first 20 features can be found in Figure 1, whereas Figure 2 displays all 20 density features overlaid in the same plot. We can see how the majority of features have a very narrow peak centred at 0, with only features 5, 11, 14, 18, 19, and 20 displaying a different behaviour.

Afterwards, I applied a 2-dimensional PCA on all 500 features, and I obtained the distribution in Figure 3. We can clearly see two different clusters based on the value of PC1. Therefore, we can infer that PC1 at low values represents the variables which peak at 0 and at high values represents the other variables.

Next, I split the dataset in two training sets of equal sizes, and applied k-means to to each training set, using 8 clusters. In each case, the unused data was then mapped

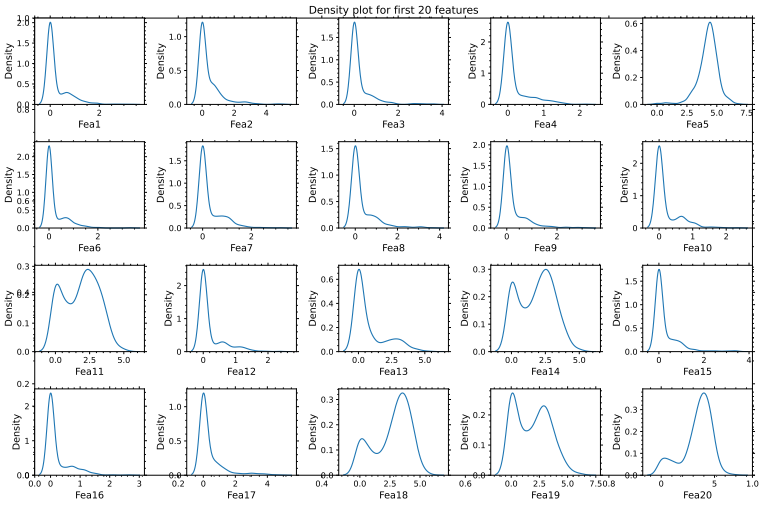


Figure 1: Density plot for all 20 features

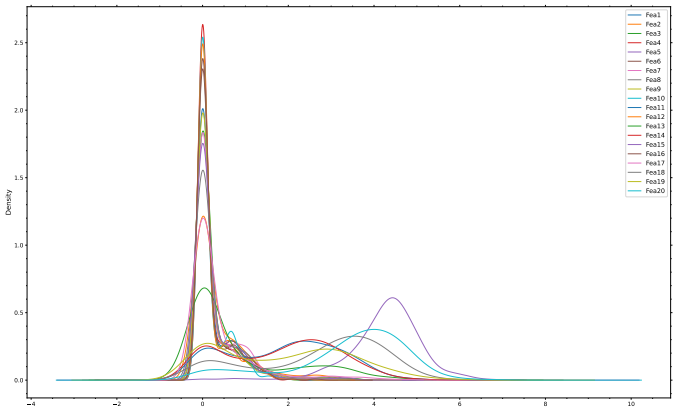


Figure 2: Density plot for all 20 features, overlaid

onto the learned clusters. This is possible because the clustering assignment is based on minimising the distance between the object and the centroid. Because the centroid is the mean position of all the events belonging to the cluster, it is then straightforward to assign the unused data in the nearest centroid. In the code, this is done by fitting k-means to the subset, and then predicting to the entire dataset.

The contingency table is presented in Table 1. We can see that there are many clusters which contain only one event, and we can also note that the contingency matrix is not diagonal. A diagonal contingency matrix implies that the k-means clustering is stable, because the events in a certain cluster fitted on a test set are located in the same cluster also when fitting on the other set. Additionally, by looking at the silhouette plots in Figure 4, we can see how the clusters are not stable as there are many negative single silhouette values.

Therefore, I tested different number of clusters, from 2 to 10. I obtained the best

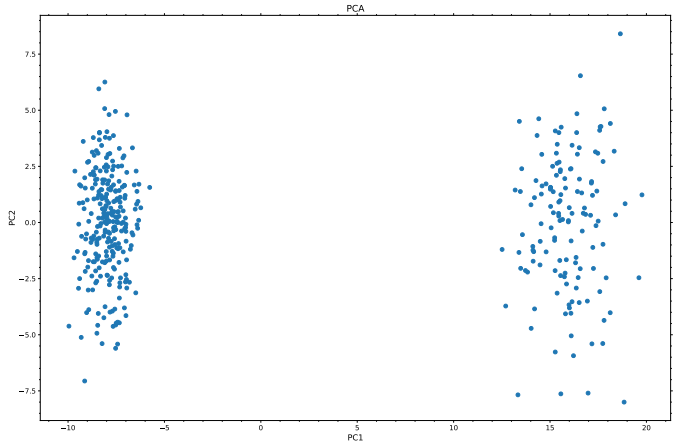


Figure 3: Visualisation of the PCA in a 2D space

		k-means on test 2								
	cluster	0	1	2	3	4	5	6	7	Total
k-means on test 1	0	0	12	0	0	102	0	0	0	114
	1	0	0	0	3	0	0	0	0	3
	2	14	0	6	38	0	1	3	1	63
	3	8	0	18	36	0	0	4	0	66
	4	1	0	0	0	0	0	0	0	1
	5	1	0	0	0	0	0	0	0	1
	6	0	71	0	0	87	0	0	0	158
	7	0	0	0	2	0	0	0	0	2
	Total	24	83	24	79	189	1	7	1	408

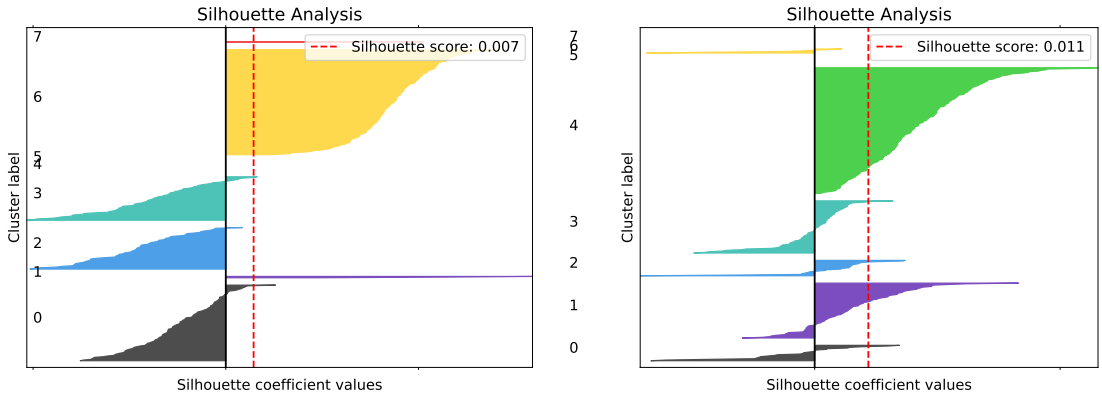
Table 1: Contingency matrix displaying the entries in the different clusters for the two subsets

configuration with 2 clusters, because the contingency matrix was diagonal (Table 2) and the silhouette scores were the highest Figure 5.

		set 2		
	cluster	0	1	Total
set 1	0	136	0	136
	1	0	272	272
	Total	136	272	408

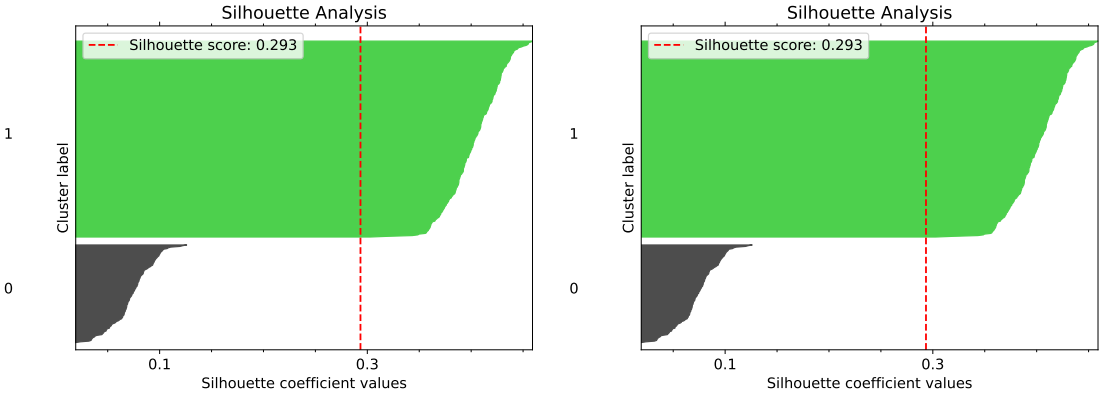
Table 2: Contingency matrix displaying the entries in the different clusters for the two subsets

Finally, I identified the clusters within the PCA, as displayed in Figure 6. We can



(a) Set 1 (b) Set 2

Figure 4: Silhouette scores for both sets



(a) Set 1 (b) Set 2

Figure 5: Silhouette scores for both sets. We can see that they are identical.

clearly see the separation between the clusters within the PCA. I also tried performing the PCA before applying k-means, and I obtained the same figure. In general, is better to do k-means before because in this way all the information contained in the dataset are used, whereas applying to the PCA means using only a smaller fraction of that. The PCA should be used after the clustering has been performed, as a way to better visualise the data.

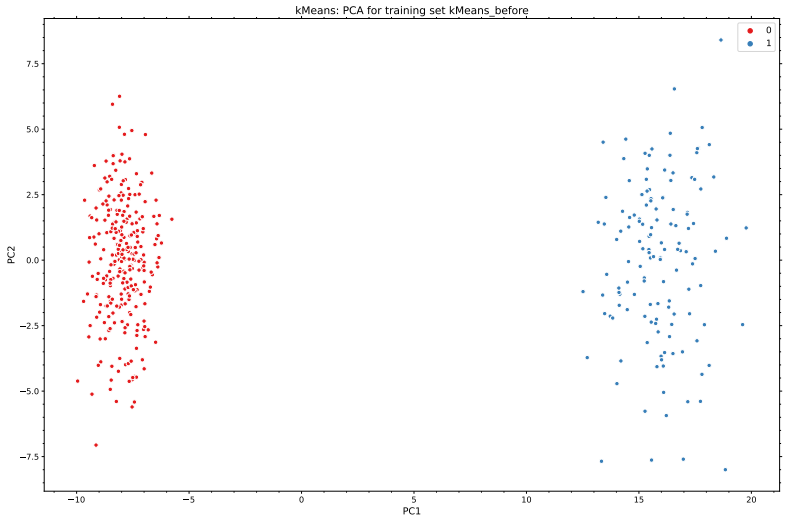


Figure 6: k-means clusters visualised within the PCA

A-2

The frequency of the labels is presented in Table 3. We can see how we have 20 unlabelled observations out of the 428 events. Among the labelled, the label 1 is the most represented, followed by label 2 and 4.

Label	Number	% on total	% on labeled data
1	179	41.8%	43.9%
2	157	36.7%	38.5%
4	72	16.8%	17.6%
Unlabelled	20	4.76%	-

Table 3: Labels frequency

I have then scouted the dataset for duplicated observations, by looking at rows which were identical up to the label. The duplicated observations are presented in Table 4. If also the label was equal, then I just removed one of the two, because these two observations were identical in every aspect. On the other hand, if the label is different, I removed both observations, because there was no way of knowing which one was the correct one. This method is conservative, and it implies a significant loss of data, but since the duplicated observations with wrong labels were only 10, I considered this loss negligible with respect to the total size of the dataset.

The missing data can either by at random (MAR) or not at random (MNAR). The former refers to situations where the missing data is related to a predictor variable that itself is fully recorded, whereas the latter refers to when the value of the missing data is related to the reason for it being missed. When there are observations with missing labels, two different approaches can be considered: omission and imputation. The pros

Obs 1	Obs 2	Label 1	Label	Action
73	145	1	1	Removing 73
219	290	2	4	Removing both
146	408	1	2	Removing both
43	192	1	1	Removing 43
65	252	1	1	Removing 65
27	259	1	2	Removing both
383	395	2	1	Removing both
187	248	1	1	Removing 187
174	310	2	2	Removing 174
165	423	1	1	Removing 165
343	388	1	1	Removing 343
116	296	4	4	Removing 116
350	351	2	1	Removing both
119	358	1	2	Removing both
118	381	4	4	Removing 118
99	172	4	1	Removing both
29	100	2	4	Removing both
45	106	1	1	Removing 45
209	304	1	4	Removing both

Table 4: Summary of the duplicated observations and the action applied

for omission are that it's the simplest method, in that it consists of removing the missing labels. Furthermore, if the missing is completely at random (MCAR), omitting the data will not introduce any bias. However, if the data is not missing at ranodm (MNAR), omitting the data will cause a bias, which is why in these cases it is better to impute the data. This approach is more complicated since it requires the generation of new data, and it can introduce variance, because the imputed data may not be generated in the correct way. On the other hand, the additional gain of using imputation is that no amount of data is wasted, which is useful especially when there aren't many observations available.

I have then predicted the missing labels using k-nearest-neighbour (KNN). The confusion matrix can be found in Table 5 and can be visualized in Figure 7. This method was then used to predict the 20 missing labels, with the predictions being:

		True		
	Label	0	1	2
Pred	0	0.41	0.01	0.003
	1	0.13	0.26	0.01
	2	0.05	0	0.12

Table 5: Confusion matrix for the KNN classifier

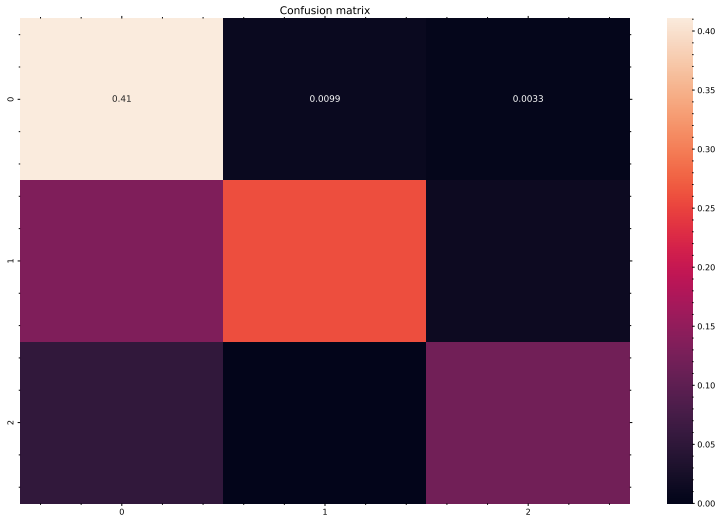


Figure 7: Visualisation of the confusion matrix

- Label 1: 15 entries
- Label 2: 3 entries
- Label 4: 2 entries

Therefore, the final labels are displayed in Table 6, with the last column displaying the frequency before the new labels were added. We can see that the difference is minimal, because only 4.76% of the labels were missing.

Label	Number	% now	% before
1	194	45.3%	43.9%
2	160	37.4%	38.5%
4	74	17.3%	17.6%

Table 6: Labels frequency

A-3

There are 55 missing observations, coming from 5 samples (138, 143, 231, 263, 389) and 11 features (58, 142, 150, 233, 269, 299, 339, 355, 458, 466, 491), as displayed in **??**.

The imputation of missing data can be done either with a static or with a model-based method. The static approach can be used for time-series data, simply by carrying forward the last observation. On the other hand, a model-based approach requires the generation of the missing data based on the available data, by predicting it with a generative model. The static approach has the disadvantage of not being able to detect variability, because we force the data to be the same as the last observation, but has the advantage of being the

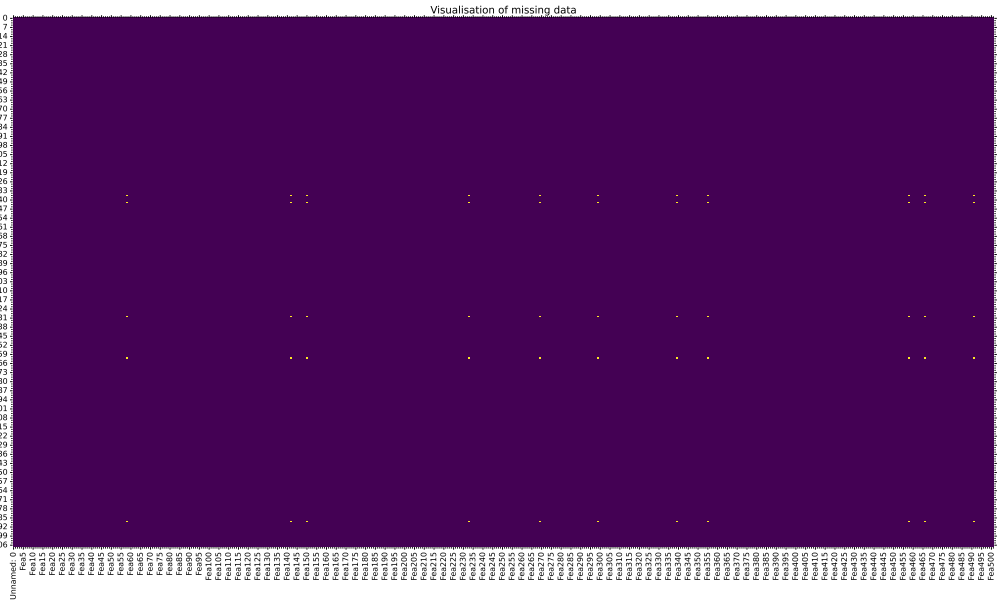


Figure 8: Visualisation of the missing data

simplest method. The model-based approach instead allows to preserve the relationships which are present in the data, and it reduces the bias especially in situations where the missing is not at random. The con of this method is that it may introduce a variance, which can be reduced with multiple imputations, where the uncertainty associated with the imputation is taken into account.

I decided to impute the missing data using a Gaussian Mixture Model (GMM). The choice was motivated by the features density distributions, where it was clear