S1 Principles of Data Science Coursework Report

CRSiD: tmb76

University of Cambridge

Section A

Question 1

In this question we are given a dataset which contains 408 observations, each with 500 features. All features seem to be continuous variables. The dataset contains a classification column, containing 3 categories: 1, 2 and 4. The aim of this question is to explore the dataset and conduct some processing of the data. Conducting a check for missing values, which there were none, the data is explore in the following ways.

(a)

The data first needs to be visualised in some way. Now because there are 500 features, and therefore 500 dimensions in the feature-space, this means that for now, only visualisations of each feature separately is useful. Thus a density plot of the first 20 features is obtained and shown below (Figure 1). In Figure 1, the features were grouped by similarity and not by number. The main observation is just that: some features are very similar to each other, following similar distributions. Some are bimodal, others have a stronger peak on just one of bimodal peak locations. A rough assumption to be made here would be that there are overall 34 groups of highly correlated features for the entire dataset. The most common one has a very strong peak at 0, with a much smaller one at around 1. Another is bimodal, with peaks at 0 and 3. And the last one is also bimodal, with peaks at 0 and a stonger one at approximately 4. This is not yet a firm conclusion but is a good starting point for further analysis. If the features can be grouped then one could reduce the dimensionality of the feature-space to 1 representative feature from each group.

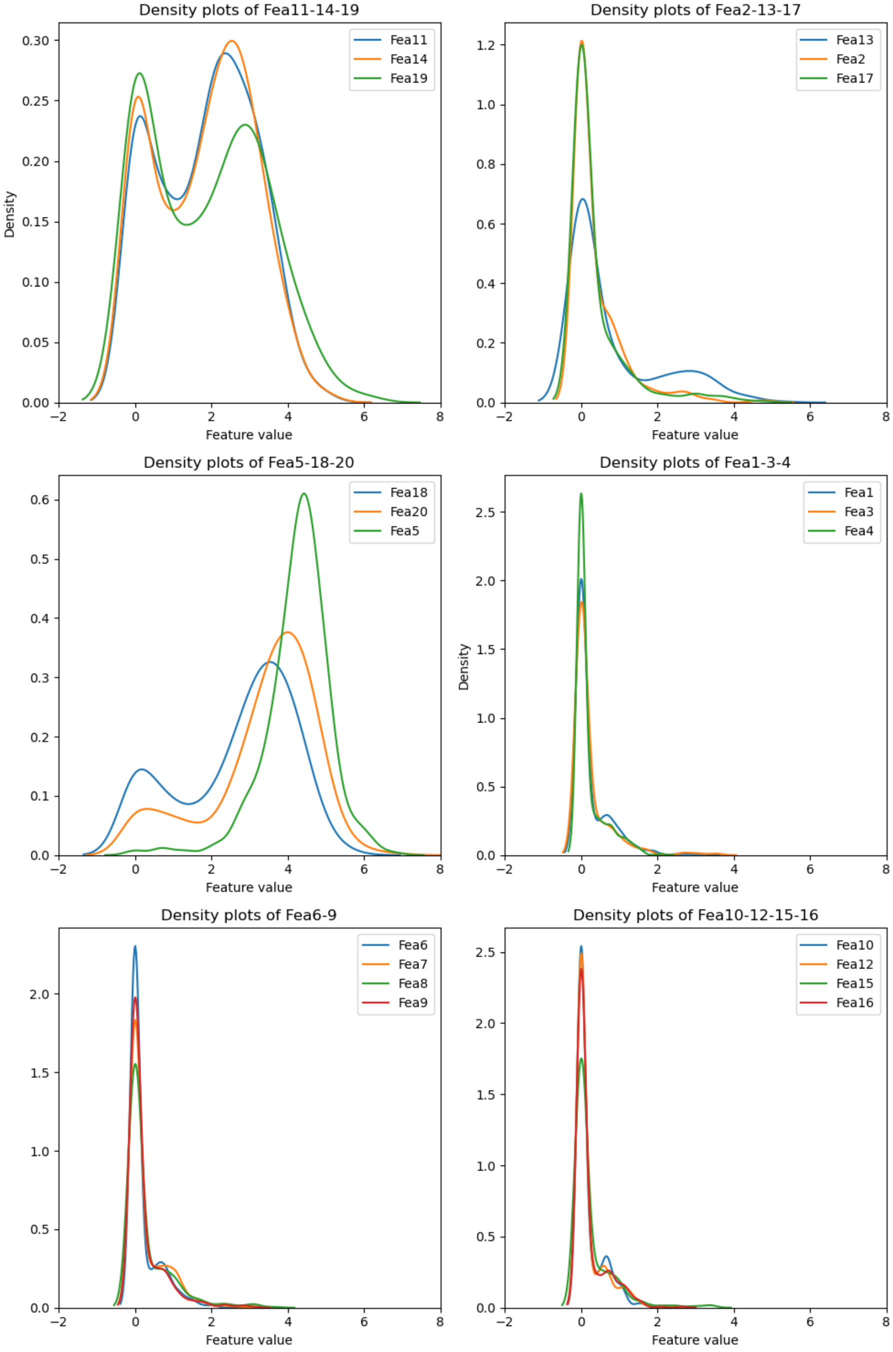

Figure 1: **Density plots of the first 20 features, grouped by similarity**. The x-axis was set to the same scale for all plots, for easire comparison. The y-axis is however different for all plots.

(b)

Thus, Principle Component Analysis (PCA) is conducted on the dataset. PCA enables one to derive a lower-dimensional set of features from the full **A** dataset. This is done by finding the direction in which the observations have the greatest variance, for the full feature space [1, pp. 255-257]. This is done on the entire dataset, not just the first 20 features. Dropping the classification column, `scikit-learn`'s PCA function is used, setting the number of components to 2 to get the first 2 Principles Components (PC). The dimensionaly reduced dataset was obtained and the data was plotted in the following scatter plot (Figure 2).

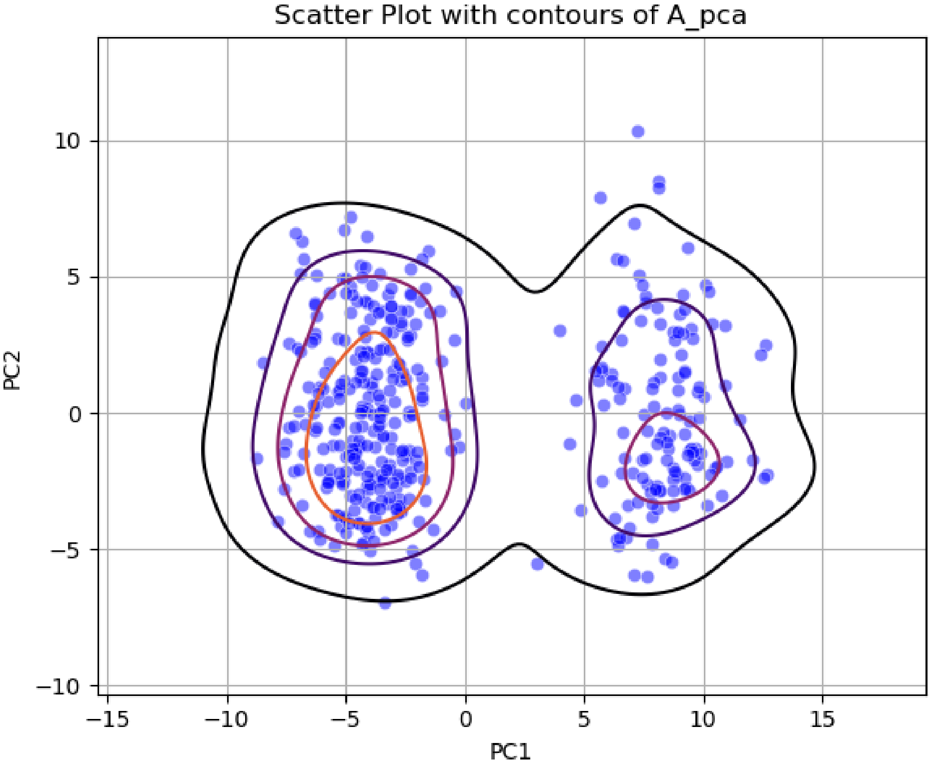

Figure 2: **Scatter plot of the dataset after PCA.** Density contours are also shown.

From Figure 2, the bimodal separation seen in some of the first 20 features is seen again in the 1st PC. SMTHG ABOUT LINEAR SEPARATION. The 2nd PC also resembles the more commonly observed feature in Figure 1, with a very strong peak and a much weaker one at a slightly higher value, more from the right-hand side group. This is hard to see in the scatter plot, but the desnity contours do show the negative skewness of the data's distribution for that PC.

(c)

From this visualisation, clustering could be run to try and visualise the how the clusters would compare to these groups that are observed. In this case, the actual classification is known but it is worth seeing how this would perform if it was not. The `scikit-learn` KMeans clustering algorithm was used, with all parameters set to their default values. This is done by not inputting any parameters to the function, apart from the random state so the same result is obtained at every run. The default number of clusters is 8. Now from Figure 1 and figure 2, it seems unlikely that there are 8 clusters in the data. This is one of the risks of K-means clustering as it will "find" 8 clusters, even if there are not 8 clusters in the data. To assess if "default" K-means clustering performs adequatly, the data is split into 2. For both these splits, clustering is run and then applied to the other split. The reason this can be done is that K-means clustering's main result is the position of points called centroids, which are the geometrical centres of each clusters (when defining the centroids from the feature means). In terms of their significance they are each the most representative points of each cluster. One of the `scikit-learn` K-means output is those `cluster_centers_`. Then when using the `.predict` method after fitting the model, i.e. clustering one half of the date, the other half can be clustered base on the distance of each point to the centroids of the clusters. This is done for both splits of the data and the results are shown in the tables below:

Clusters	**0**	**1**	**2**	**3**	**4**	**5**	**7**
0	4	9	11	4	2	9	2
1	4	5	18	5	3	3	4
2	4	3	12	4	2	2	7
3	6	8	9	4	0	0	5
4	3	5	5	2	0	3	1
5	4	2	9	5	2	5	5
6	0	0	0	0	0	0	2
7	0	2	0	0	0	0	0

Table 1: Contingency table for the 2 clusterings.

(d)

As can be seen, cluster 6 for the 2^{nd} clustering (columns) is empty. This can be explained by the initialization of the centroids. Indeed the K-means clustering has to place the centroids somewhere at the start, and they are then updated based on which points are closest to it and their mean. If the centroid for cluster 6 in the 2^{nd} clustering happened to have no points closer to it than other centroids, it stayed where it was for the whole clustering process [2]. Overall, the fact that a too large cluster number was selected for the model means there is almost no agreement between clusters (Table 1). This means that the clusters are not very stable. Stability here refers to getting similar clusters when clustering the same or very similar datasets. Furthermore, the number used to designate the cluster is not the same in the 2 clusterings. In other words, it could be the case that the relatively greater agreement between cluster 1 from the 1^{st} clustering (rows) and cluster 2 from the 2^{nd} clustering (columns), is due to the fact they have centroids that are close to each other. But they happened to have different numbers. Regardless the agreement is small since all values in the table are small.

Thus, we repeat the clustering for a lower number of clusters, 3, based on the visualisation of the data, Fig 2, and the actual knowledge of there being 3 clusters. We get the following results:

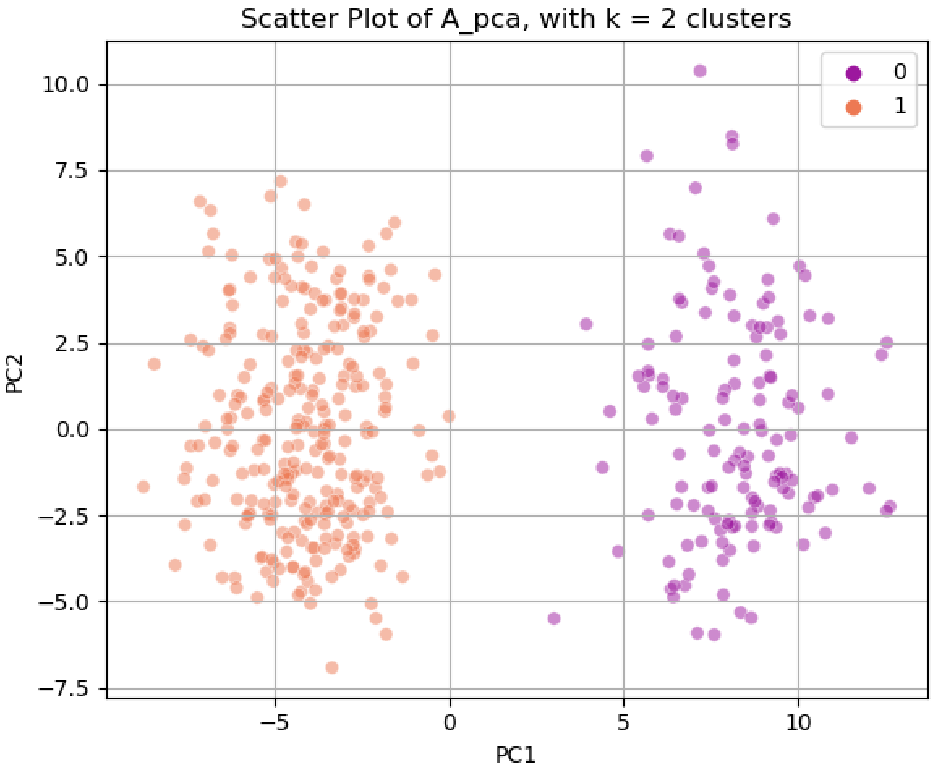

Figure 3: **Scatter plot of the 2 clusterings** The colors indicate the cluster membership.

Again, the number of the clusters are not the same but it is clear there is a much better agreement between the 2 clusterings. This is also reflected in the contingency table below:

Bibliography

[1] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. *An Introduction to Statistical Learning with Applications in Python*. Springer Texts in Statistics. Springer, 2013.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. scikit-learn: Machine learning in python. `https://scikit-learn.org`, 2011.