Exploration of Five Unsupervised Learning Algorithms

Adedamola Adesoye
College of Engineering
Northeastern University
Toronto, ON
adesoye.a@northeastern.edu

Abstract

In this report, I explored the performance of unsupervised learning algorithms on two interesting datasets. The five learning algorithms include k-means, Gaussian Mixture Model (GMM), Principal Component Analysis (PCA), Independent Component Analysis (ICA) and Uniform Manifold Approximation and Projection (UMAP). The two datasets were a bank marketing data and adult dataset. I applied all algorithms to both datasets. I then applied a neural network classifier in combination with the unsupervised learning algorithms to run some experiments on the adult dataset.

1 Datasets

The two datasets are like the ones used for the supervised learning assignment: bank marketing data and the adult data. The only difference is that the bank marketing data is more balanced with fewer instances than the previously used one. The bank marketing data describes the results of a direct marketing campaign by an unnamed Portuguese bank to convince customers to subscribe to a term deposit, which is a type of deposit account where money is locked up for an agreed time duration. These locked up funds allow banks to invest their customer's money in profitable ventures. The adult data contains demographic information about individuals from a 1994 U.S.A. Census database, and the target column indicates if an individual earns more or less than $50,000 per year, which can help tailor marketing or political campaigns for better conversion rates.

Table 1: The basic feature of both datasets.

	Data Set Characteristics	Attribute Characteristics	Associated Tasks	Number of Instances	Number of Attributes
Adult	Multivariate	Real, Categorical	Classification	32,561	14
Bank Marketing	Multivariate	Real, Categorical	Classification	11,163	17

1.1 Data characteristics

The histograms of classes from both datasets are shown in Figure 1. Class unbalance is evident in the adult dataset.

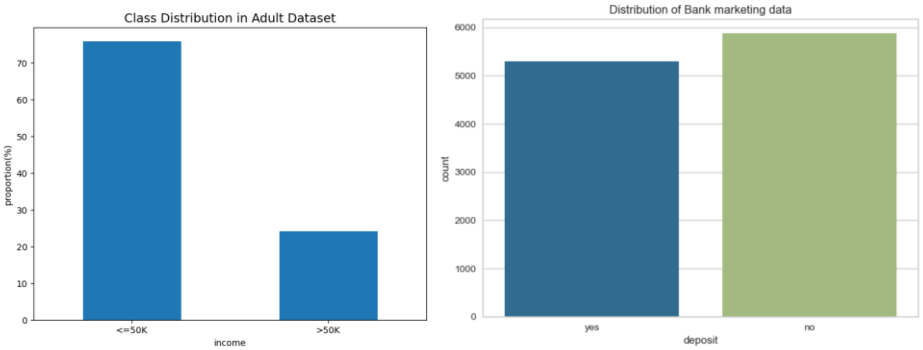


Figure 1: The class frequency of both datasets.

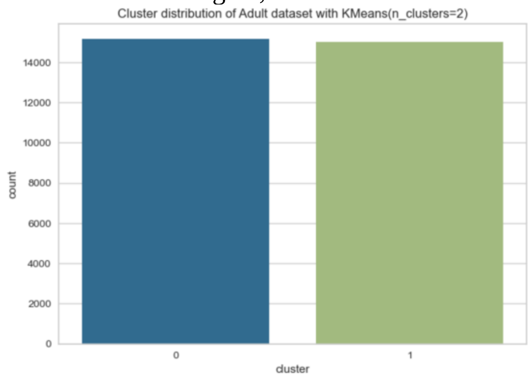
2 Adult data

2.1 Pre-processing

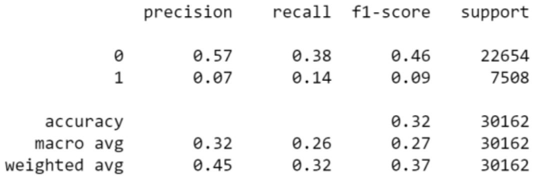
I started by dropping missing values, leaving me with 30,162 instances. I transformed the *income* feature to a numerical variable, making *<=50K* equal to *0* and *>50K* equal to *1*. Although I dropped this feature before building my clustering models, I did this to be able to later evaluate the results of my models with two clusters. To avoid issues with multicollinearity, I dropped *education* feature which provides the same information as *education-num* but in a different format. I one-hot encoded my categorical variables, scaled all my numerical variables (except *education-num*) to zero mean and unit variance; I scaled *education-num* (an ordinal variable) to a range between 0 and 1.

2.2 K-Means

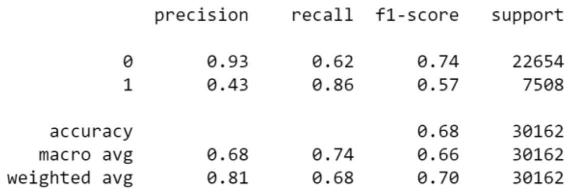
I started by initializing my K-Means with two clusters. I did this because the initial target feature was a binary variable. The resulting clusters were relatively evenly distributed with *cluster 0* having 15,148 instances and *cluster 1* having 15,014 instances.



I then went on to treat the *income* label as the original target feature of a test set and the cluster labels as the predicted labels of the test set. This yielded the classification report shown below.


```
              precision    recall  f1-score   support

           0       0.57      0.38      0.46     22654
           1       0.07      0.14      0.09      7508

    accuracy                           0.32     30162
   macro avg       0.32      0.26      0.27     30162
weighted avg       0.45      0.32      0.37     30162
```

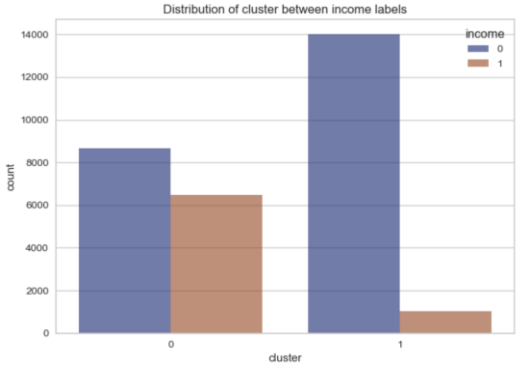
Because K-Means is unsupervised and *0* & *1* are just ways to differentiate two different clusters, I changed *cluster 0* to *cluster 1* and vice-versa. Again, treating the cluster labels as the prediction labels of a test set, I got the following results.



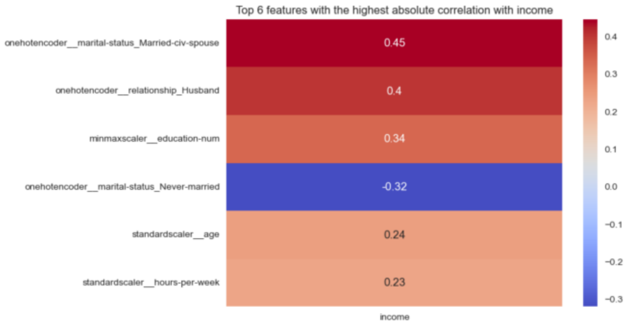
```
              precision    recall  f1-score   support

           0       0.93      0.62      0.74     22654
           1       0.43      0.86      0.57      7508

    accuracy                           0.68     30162
   macro avg       0.68      0.74      0.66     30162
weighted avg       0.81      0.68      0.70     30162
```

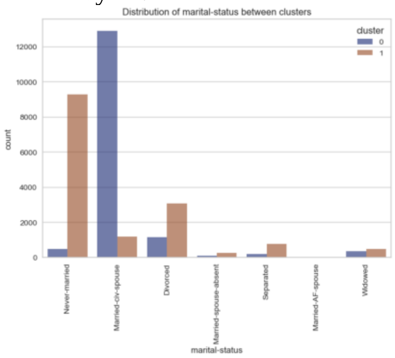
Every metric improves significantly but this model is still unsuitable for labelling the original manually labelled data, one reason, because the label distributions are unsimilar (3:1 in the original data compared to 1:1 in the clustered data).

It can be noted from the count plot below that *cluster 1* is mostly made up of residents with income less than or equal to 50k/year and *cluster 0* contains 86% of the residents whose *income* is greater than 50K/year.

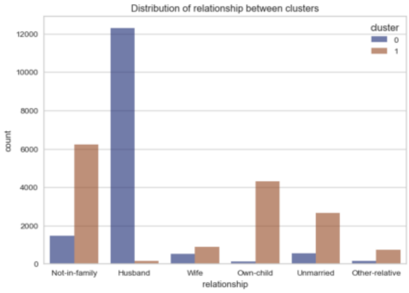


Taking a closer look at some of the features with the highest absolute correlation with *income* (shown in the figure on the left below), there's some relatively high correlation between being married to a civilian spouse and earning more than 50K/year and it's no surprise that most of these residents are grouped in *cluster 0* as shown in the figure on the right below. Also, having a *marital-status* of *never-married*, has some correlation with not earning more than 50K/year and it's no surprise that nearly all these instances are grouped in *cluster 1*, the cluster containing the larger proportion (61.7%) of the residents not earning more than 50K/year.



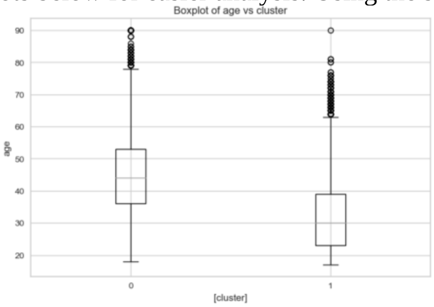


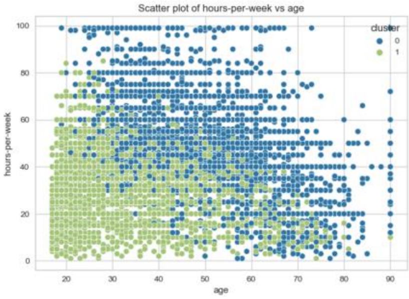
As shown below, another relatively strong indicator of earning more than 50K/year, being a *Husband*, is mostly contained in *cluster 0*, the cluster containing 86% of the residents earning more than 50K/year.



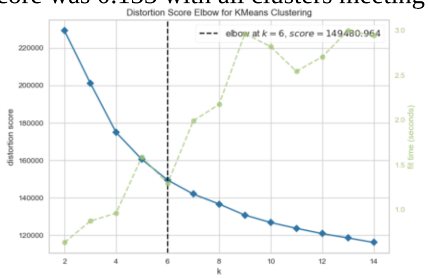
Other notable mentions are working *hours-per-week* and *age*; the higher the working hours and age, the more likely they are in *cluster 0*. The feature *education-num* displays no visible pattern in the data.

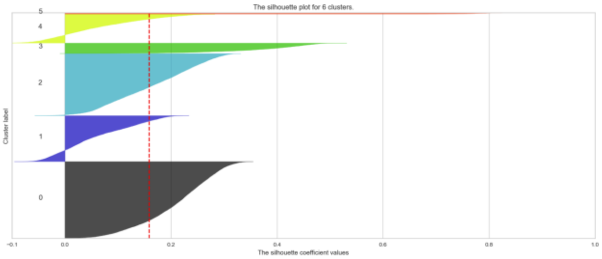
<u>N.B</u>: The features were scaled before clustering. The original scale is retained for the x and y ticks of the plots below for easier analysis. Using the scaled ticks produces the same pattern in the data.



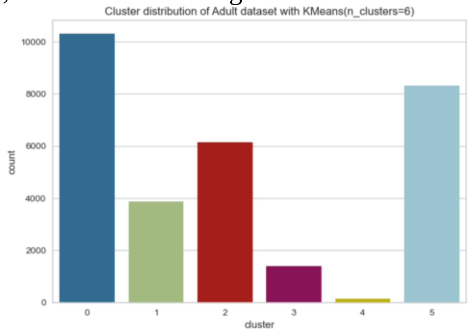


Moving on from n_clusters = 2 and looking for the optimal k using elbow and silhouette plots, the optimal k was 6 and 5 respectively. The silhouette average scores were relatively low with the highest (between two and nine clusters) being 0.162. For n_clusters = 2, the average silhouette score was 0.133 with all clusters meeting the average.

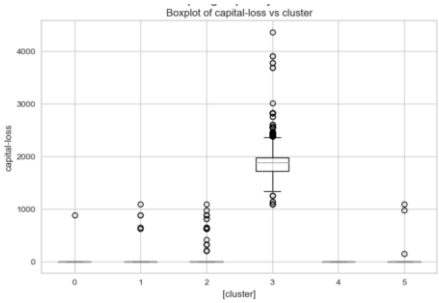


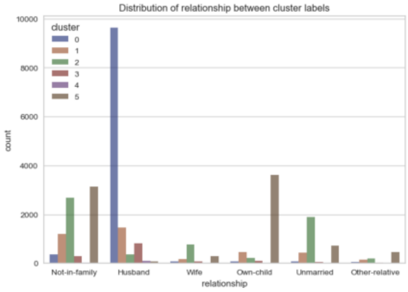


I chose to experiment with k = 6, which had an average silhouette score of 0.159. The clusters were distributed as follows, with *cluster 4* having the lowest number of instances, 148.



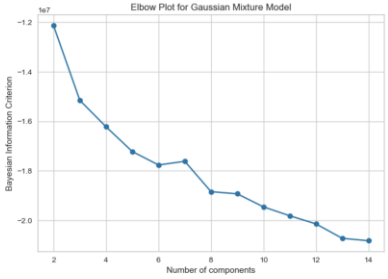
The most notable patterns were residents in *cluster 3* having *capita-loss* way higher than other clusters, *cluster 0* mostly made up of *Husband* in a *relationship*, and *Own-child* mostly contained in *cluster 5*.

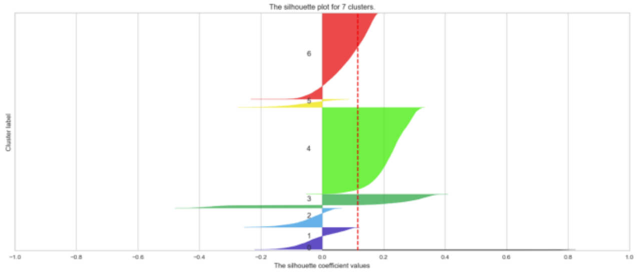




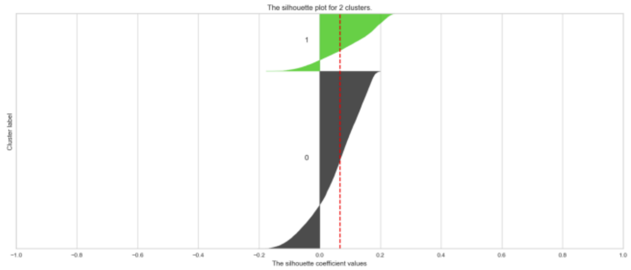
2.3 Gaussian Mixture Model (GMM)

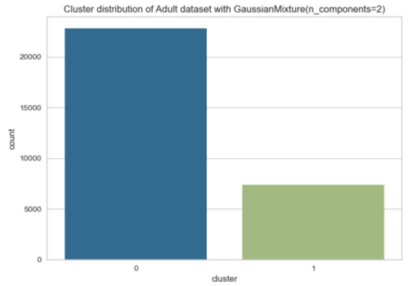
To search for the optimal number of components for the GMM, at first, I looked at an elbow plot with a range from two to nine components. The Bayesian Information Criterion(BIC) didn't look like levelling off within the selected range, so I widened the range to 14 and there was still no sign of BIC levelling off.





There were similar complications when using the silhouette plot as well. Within the range of two and nine, using seven components had the highest (0.116) silhouette score but as shown in the silhouette plot above, it is a bad pick because at least two clusters do not meet the average silhouette score. Also, the average silhouette score did not seem to have levelled off within the range of two and nine; eight n_components dropped to 0.010 average silhouette score while nine n_components jumped to 0.072. To save time, I decided to experiment with n_components = 2 which had the following silhouette plot and an average silhoette score of 0.007.

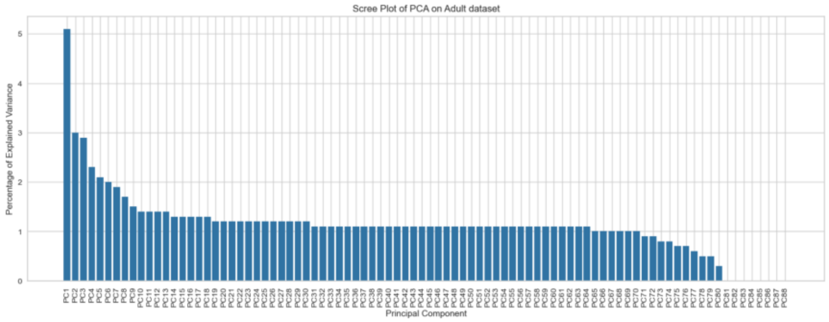




Evidently, there is a high imbalance (3:1) very similar to the proportion of the original target labels. Comparing the cluster labels with the original labels shows that 95% of the residents who earned more than 50K/year were contained in the first cluster.

2.4 Principal Component Analysis (PCA)

Before applying PCA to my dataset, I scaled every feature using scikit-learn's StandardScaler to ensure each feature was on the same zero mean and unit variance scale and to avoid PCA being biased towards any one feature. I applied PCA to the scaled data and got the same number of components as original features, 88. The scree plot below shows how much variation each component accounts for.

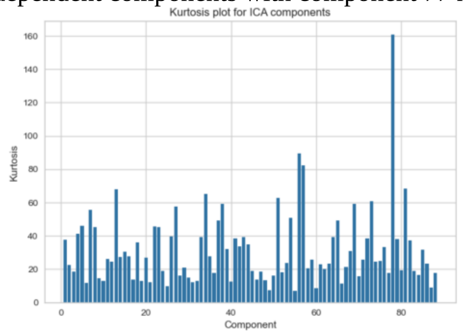


The last eight components, together, account for less than 1% of variation in the dataset. The clustering result of retaining all the principal components was the same with the clustering result of the original dataset.

It seems the initialization of the PCA influences how much variation the resulting components will account for because when I initialised PCA with n_components = 10, I noticed that the resulting ten components accounted for a slightly higher variation (0.35%) than did the first ten components of the PCA with n_components set to the default “None”. On the other hand, when I initialized PCA with n_components = 0.90, I got the same 67 components with the first 67 components of the PCA with n_components set to the default “None”. Using only these 67 components and clustering my data with GMM(n_component=2), I got the same cluster labels as when I kept all components. Using GMM(n_components =3) yielded different results; only 12,343 instances had the same labels. I also checked and confirmed that this was not a case of *cluster 0* in one result being *cluster 2* in another result; the data distribution was different among clusters for both results.

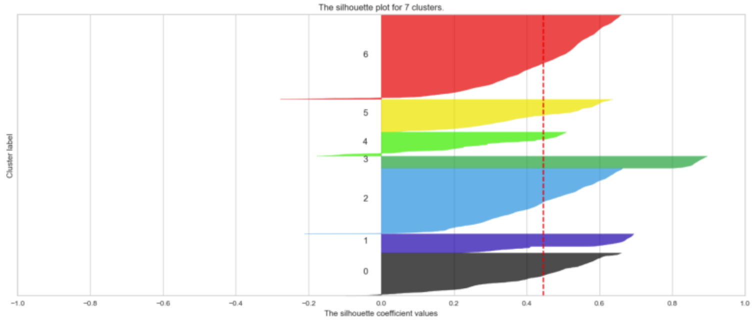
2.5 Independent Component Analysis (ICA)

Fitting the FastICA with its default parameters yielded a warning with the description, “FastICA did not converge. Consider increasing tolerance or the maximum number of iterations.” I increased the maximum iterations to 1000 and the tolerance to 0.001 but I got the same warning. Adding *whiten = False* also yielded the same warning. I compared GMM(n_components=2) clustering of FastICA(max_iter=1000, tol=0.001, random_state = 26, whiten=False) fitted and transformed data with FastICA() fitted and transformed data. The results were different. The former yielded the same clustering results as the original data. The following plot the distribution of kurtosis values of the independent components with component 77 having the highest value.



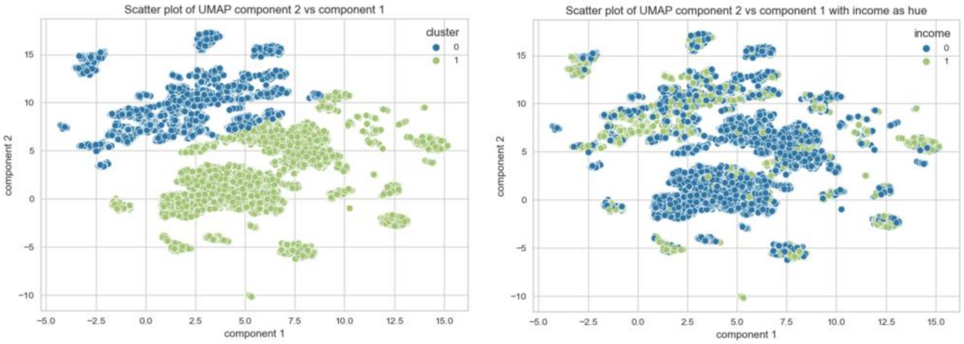
2.6 Uniform Manifold Approximation and Projection (UMAP)

After reducing my data’s dimensions to two using UMAP, I combined silhouette plot and GMM to find the optimal number of components betweeen two and nine. Seven clusters had the highest average silhouette score (0.445) with all clusters exceeding the average silhouette score.



I used these number of clusters in running experiments with the neural network.

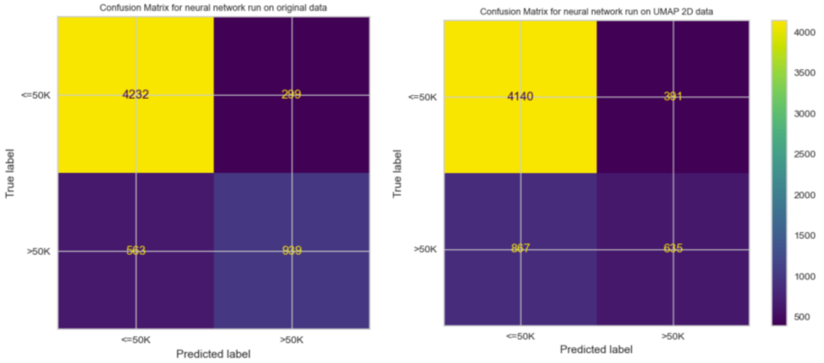
The plot on the left below shows the result of using GMM to group 2D UMAP data into two clusters. The plot on the right shows the same 2D data with the original target variable as the hue.



2.7 Neural Network

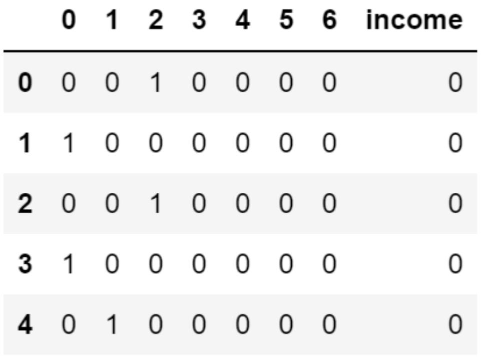
2.7.1 Results of Neural Network on UMAP 2D data

An MLP with these parameters ({'solver': 'adam', 'hidden_layer_sizes': (100,), 'alpha': 0.1, 'activation': 'relu'}) yielded an f1-score of 0.69 for the original data and 0.50 on the UMAP 2-D data. The following confusion matrix shows the classification results on both datasets.

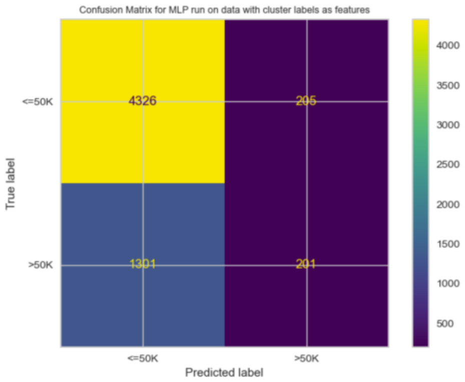


2.7.2 Result of Neural Network run on data with cluster labels as features

I used the result of GMM grouping the UMAP 2D data into seven clusters for this part of the experiment. I one hot encoded the cluster labels and got data frame shown in the picture on the left below.



	0	1	2	3	4	5	6	income
0	0	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	1	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0



Running the neural network learner on the data yielded an f1-score of 0.21. The results of the classification is shown in the confusion matrix above.

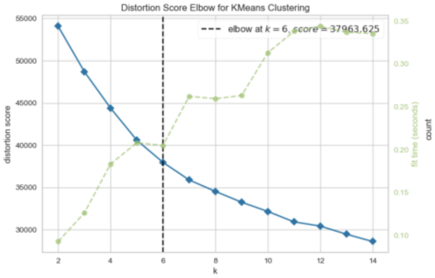
3 Banking data

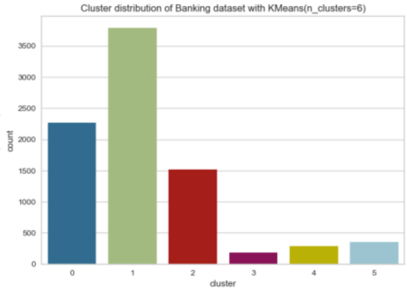
3.1 Preprocessing

As in the adult dataset, I dropped the instances with unknown values and was left with 8,393 instances. I one-hot encoded the categorical variables, scaled the numerical variables with StandardScaler and used MinMaxScaler for the the one ordinal variable contained in the data. Before applying PCA, I scaled all the features using StandardScaler.

3.2 K-means

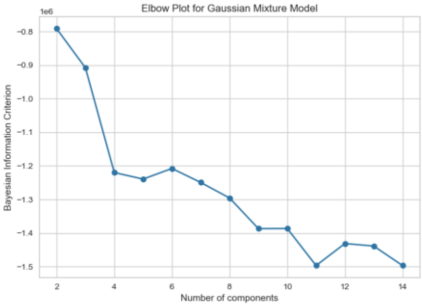
Running K-Means (n_clusters=2) on the data yielded clusters distribution very different (73:27) from the original data's target label distribution (54:46).
Based on the result of the elbow plot in the diagram on the left below, I ran a K-Means(n_components = 6) on the dataset which yielded the cluster distribution in the diagram on the right below.





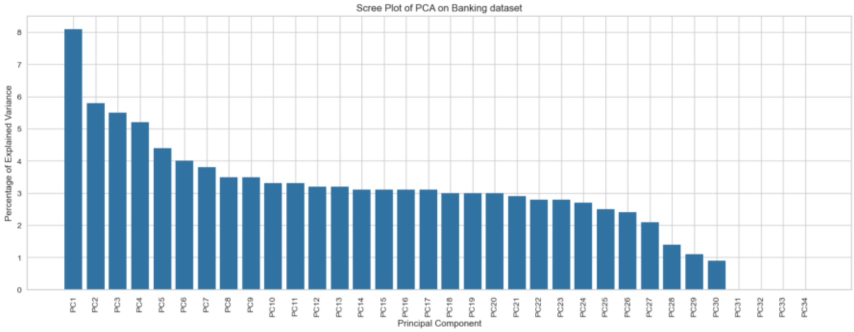
3.3 GMM

A GMM(n_components=2) on the dataset yielded a (65:35) data distribution between the clusters. Like in the case of the adult dataset, the BIC for the elbow plot didn't seem to level off in the range of k equals two and 14 as shown in the elbow plot below.



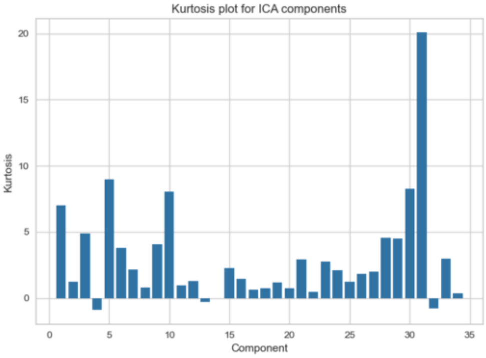
3.4 PCA

Applying PCA to the dataset yielded the following plot showing the components and the amount of variation they account for. The last five components account for less than 1% variation.



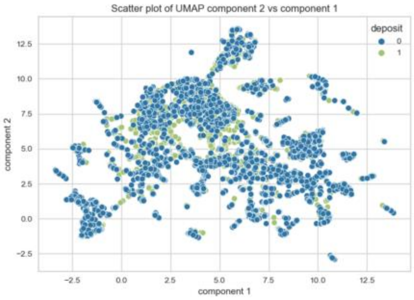
3.5 ICA

Applying ICA to the data yielded the following kurtosis for each component. Notably, they result in a lower average kurtosis than those derived with the adult dataset.



3.5 UMAP

Reducing the data to two dimensions and visualizing it like in the diagram below show no clear pattern in the data.



4 Conclusion

In this report I have applied five unsupervised machine learning algorithms to two datasets.

Acknowledgments

The learning code was produced with multiple discussions with OpenAI's ChatGPT.

References

[1] Chen, J. (2022, March 20). Time Deposit: Definition, How It's Used, Rates, and How to Invest

Investopedia. https://www.investopedia.com/terms/t/termdeposit.asp#:~:text=What%20Is%20a%20Term%20Deposit,levels%20of%20required%20minimum%20deposits.