ML assignment 4

By Rupal Shrivastava

Table of Contents

Introduction

In this assignment, I will implement Feature Selection, Feature Transformation and Unsupervised Learning on the GPU runtime and Bank Marketing dataset. I will also implement Artificial neural network after feature transformation.

Dataset Description

The GPU dataset used can be found at: https://archive.ics.uci.edu/ml/datasets/SGEMM+GPU+kernel+performance
The dataset consists of 14 features (independent variables) and 241,600 rows. First 10 features are ordinal and last 4 as binary variables. There were 4 runs performed on this dataset, which corresponds to 4 runtime variables in the dataset. More details on the dataset can be found in the link mentioned above.
The Bank Marketing dataset can be found at: http://archive.ics.uci.edu/ml/datasets/Bank+Marketing#
This dataset contains Direct Marketing (cold calling) campaign information of a Portuguese Bank. The goal is to predict if the client will subscribe to a term deposit. I chose this dataset because it is interesting to see how many conversations does cold calling bring to the business. This dataset contains 15 features, and 45211 rows. It has a combination ordinal, nominal and interval variables.
The assignment is divided in different parts: Data Preprocessing (which remains the same as the previous assignment for both the datasets), Feature Reduction, Unsupervised Algorithm and ANN Experiments.

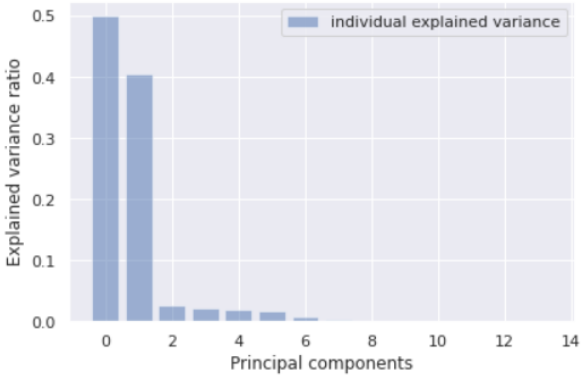
Feature Transformation

Here we are performing feature selection and feature transformation on all the features, to use in the following algorithms and ANN.

Dataset 1 – GPU Runtime

Feature Selection

From feature selection, I am trying to pick the 7 most relevant features from the dataset. To achieve this, I am using Random Forest Algorithm, and got ['MWG', 'NWG', 'KWG', 'MDIMC', 'NDIMC', 'VWM', 'SA'] as the most relevant ones for this dataset.

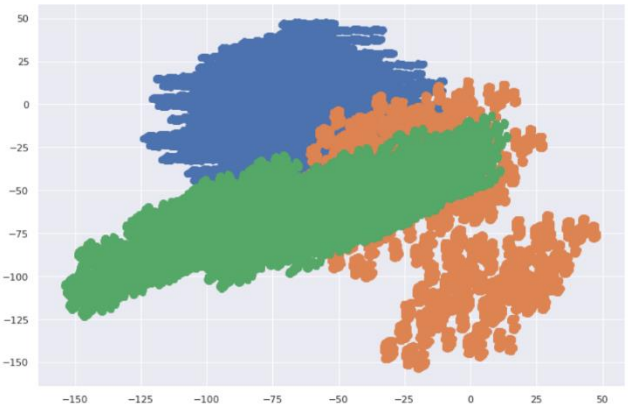


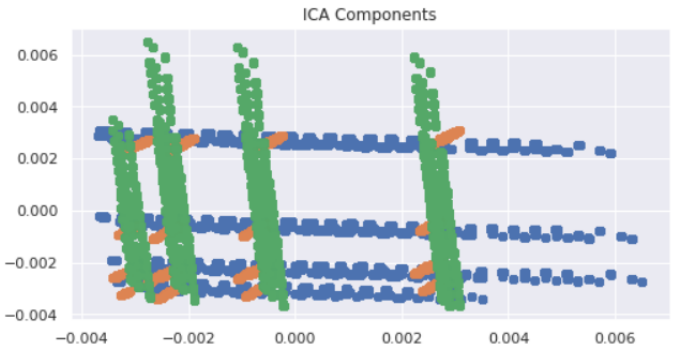
Principle Component Analysis (PCA)

Now I am applying PCA to figure out the number of Principle Components which cover the maximum variance. From the graph we can see that just 2 PCs are covering the maximum variance of the dataset. First and second PC is covering 50% and 40% of variance of the dataset.

Independent Component Analysis (ICA)

In ICA, I am selecting the mutually independent components of the dataset. Using 3 components, we can see the distribution of the data in the graph.





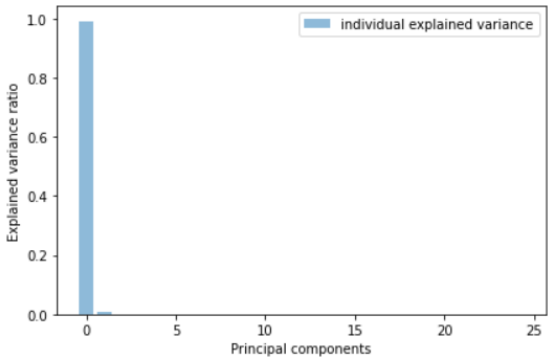
Random Component Analysis (RCA)

Similarly applying RCA, we are getting the following 3 clusters in the dataset. These cluster are compact.

Dataset 2 – Bank Marketing

Feature Selection

From feature selection, I am trying to pick the 12 most relevant features from the dataset. To achieve this, I am using Random Forest Algorithm, and got ['age', 'balance', 'housing', 'loan', 'duration', 'campaign', 'job_admin.', 'job_self-employed', 'job_technician', 'marital_single', 'education_primary', 'education_tertiary'] as the most relevant ones for this dataset.

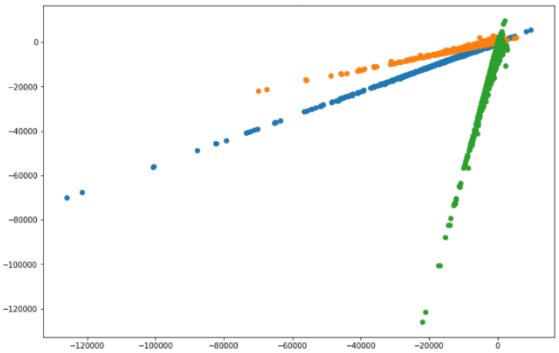


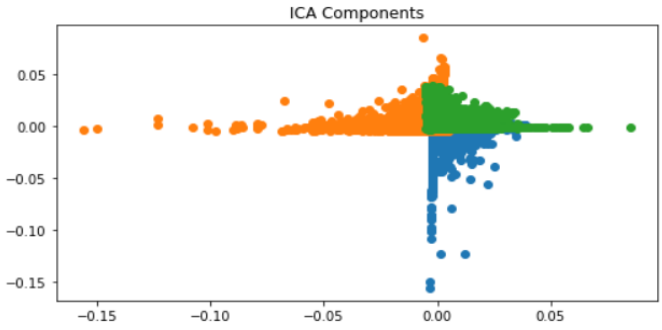
Principle Component Analysis (PCA)

Now I am applying PCA to figure out the number of Principle Components which cover the maximum variance. From the graph we can see that just 1 PCs is covering the 99% variance of the dataset. But we select 2 PCs to perform the algorithms.

Independent Component Analysis (ICA)

In ICA, I am selecting the mutually independent components of the dataset. Using 3 components, we can see the distribution of the data in the graph.





Random Component Analysis (RCA)

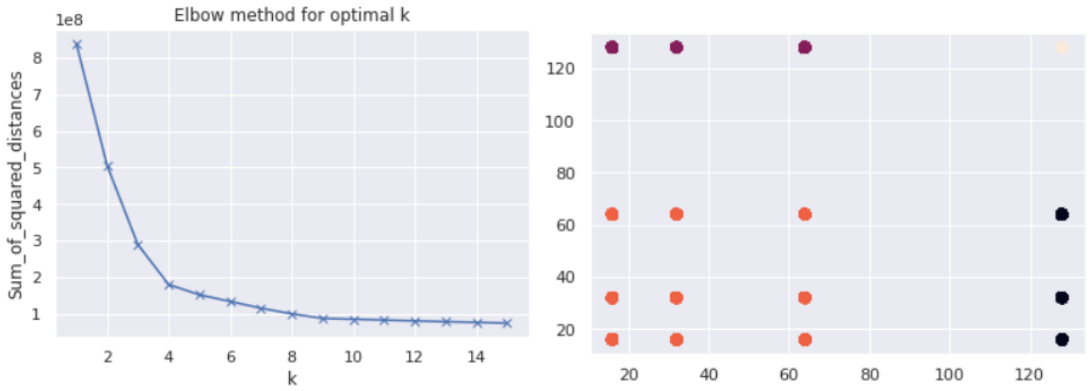
Similarly applying RCA, we are getting the following 3 clusters in the dataset.

Algorithms

Dataset 1 – GPU Runtime

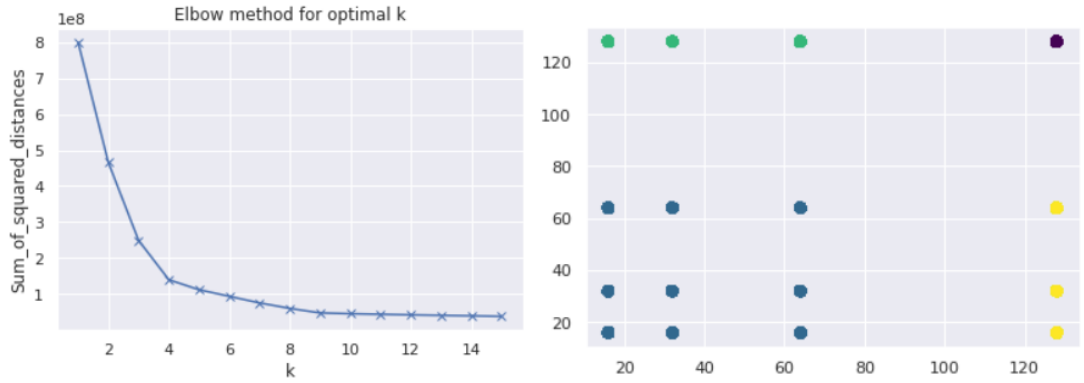
K-means

In K-means, I am using the elbow method to choose the number of clusters. From the graph, we can see the elbow at k=4. So, using 4 clusters, we get the following clustering results, with the center at each corner of the graph, well separated.



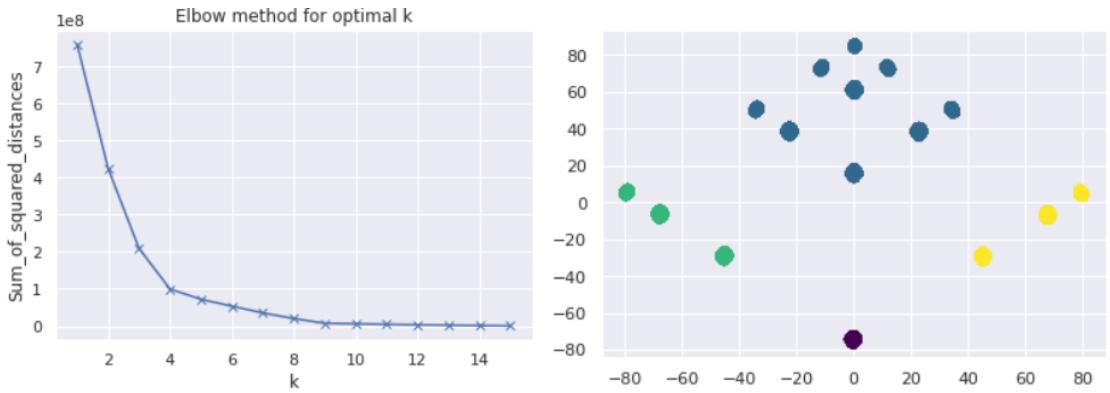
With Feature Selection

After using the features that we found using Random Forest, again we get 4 clusters from the elbow method, but now the SSE value has reduced from being ~2 to ~1.2. The clusters are still well separated, so we are getting better results with Feature Selection.



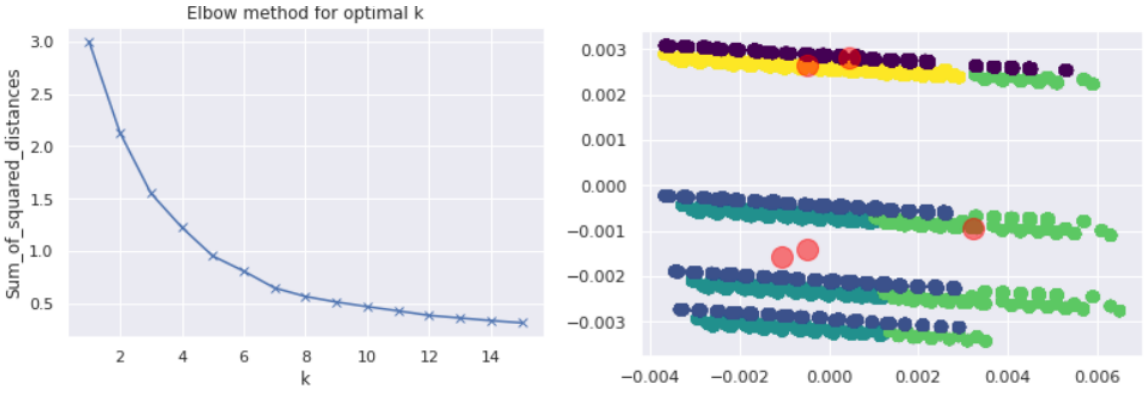
With PCA

Now I am passing my dataset with 2 PCs that we had selected. Here from the elbow graph we can see the SSE value has further reduced 1. The number and distribution of clusters remain the same, but everything is now orthogonal to the original data points as we would expect in PCA. This validates we are using the right number of clusters for this dataset.



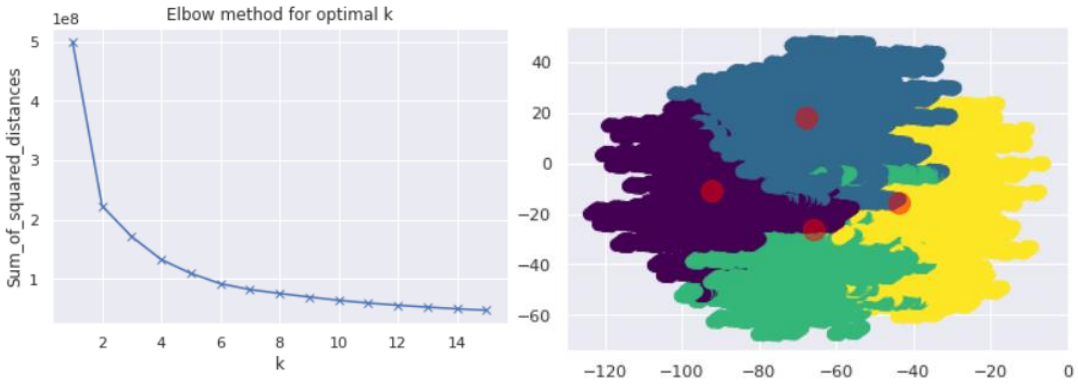
With ICA

In ICA, we are getting 5 clusters from the elbow graph, with slightly lower value of SSE than PCA. But the clusters do not look well separated, with cluster centers being close to one another. This means, in this dataset we cannot successfully identify the independent components.



With RCA

In this, using the elbow curve we are just getting 2 components, but the SSE value is higher than PCA. So, I selected 4 components as it has equivalent SSE value. But again, we see from the graph that the clusters are not well separated, and centers are not as far from one another.

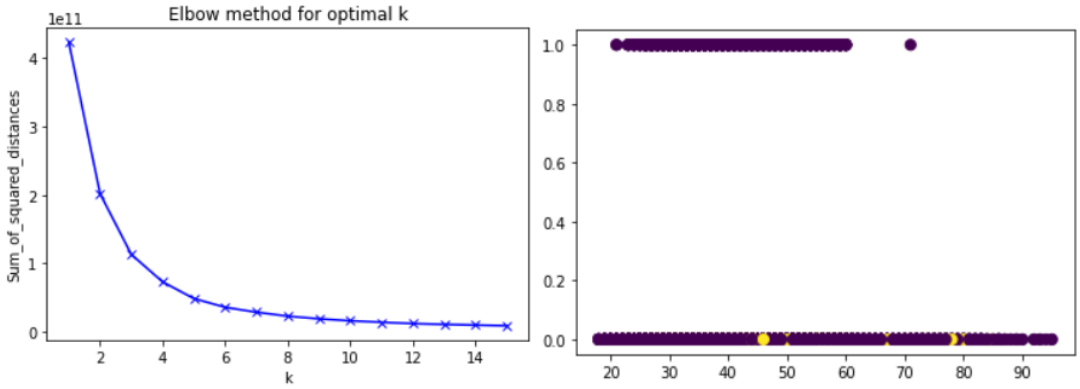


Overall, PCA works best this GPU dataset.

Dataset 2 – Bank Marketing

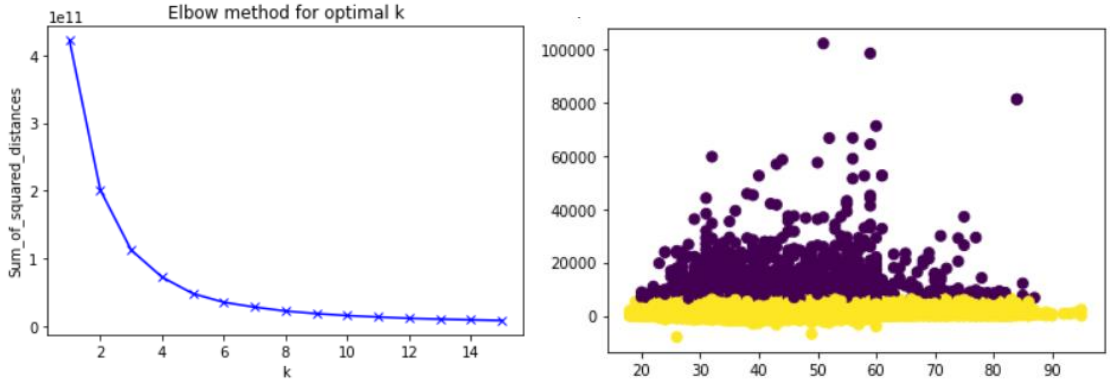
K-means

In K-means, I am using the elbow method to choose the number of clusters. From the graph, we can see the elbow at k=2, which makes sense as we are doing binary classification. So, using 2 clusters, we get the following clustering results, with the center at top left and bottom right corner of the graph, well separated.



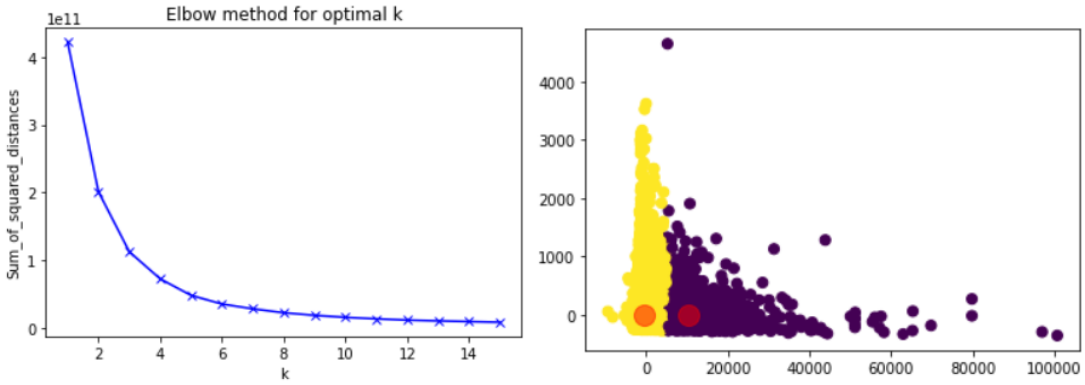
With Feature Selection

After using the features that we found using Random Forest, again we get 2 clusters from the elbow method, but now the SSE value has slight reduced. But the clusters are well separated, though closer to one another. So, we are getting better results with Feature Selection. The cluster centers have now shifted to top right and bottom left corner, making them far apart.



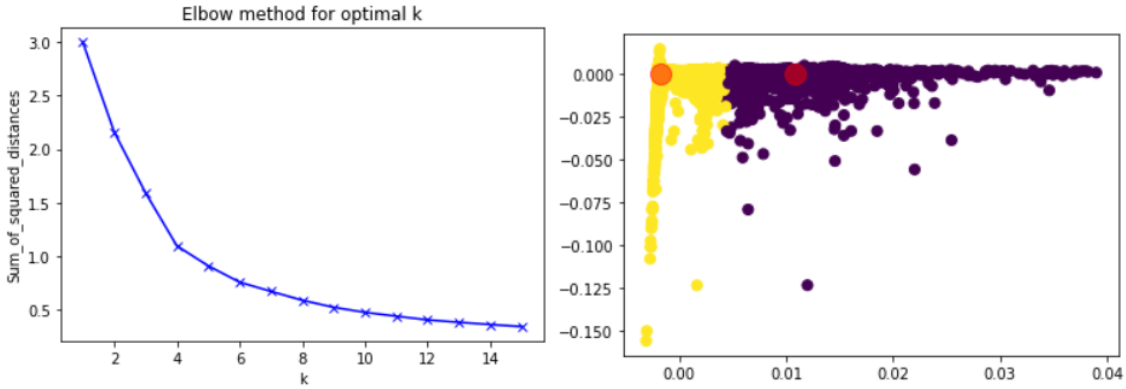
With PCA

Now I am passing my dataset with 2 PCs that we had selected. Here from the elbow graph we can see the SSE value is still the same. The number and distribution of clusters remain the same as feature selection, but everything is now orthogonal to the original data points as we would expect in PCA. But the cluster centers are now closer to each other as we can see in this graph.



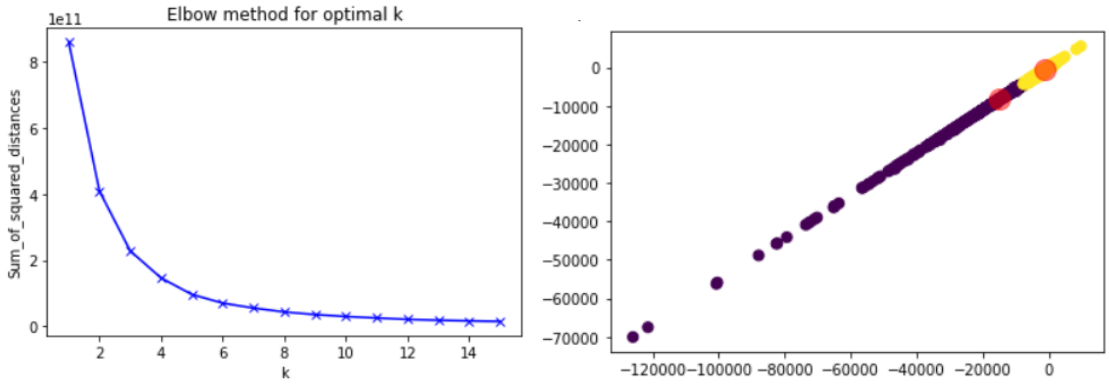
With ICA

In ICA, we are getting 4 clusters from the elbow graph, with lower value of SSE than before. But I am sticking to 2 components as this is a binary classification problem. The clusters do look well separated, with cluster centers being not too close to one another. This means, in this dataset have the independent components as the SSE is lower and distribution is better than PCA.



With RCA

Again, we are using 2 components, but the SSE value is higher than before. We see from the graph that the clusters are not well separated, and centers are not as far from one another. This does not perform well for this dataset.



Overall, ICA works best this Bank Marketing dataset.

Dataset 1 – GPU Runtime

Expectation Maximization

In this, we will look at I am using 2 components as they were giving the best separation of the clusters. We get the clusters as shown in the graphs.

Expectation Maximization	**Inference**
With Feature Selection	After using the features that we found using Random Forest, again we take 2 clusters and they look pretty similar to the previous case. (Figure 1).
With PCA	Using the PCs that we found earlier, we get a similar distribution of the 2 clusters, but they are now orthogonal to the original dataset. (Figure 2).
With ICA	Using the ICA results from earlier and 2 clusters, the data looks well separated in the graph. (Figure 3).
With RCA	Using RCA, we get slightly overlapping clusters but they represent the data well. (Figure 4).

Since this is a part of soft clustering and there is no accuracy or performance metric to compare the results, we can simply say from the graph if this transformation is working out for this dataset. Here I can choose both PCA and ICA to be providing good clusters but cannot comment on which one is better.

Also, we can clearly see better results using K-Means over Expectation Maximization for this dataset.

Figure 1 & 2

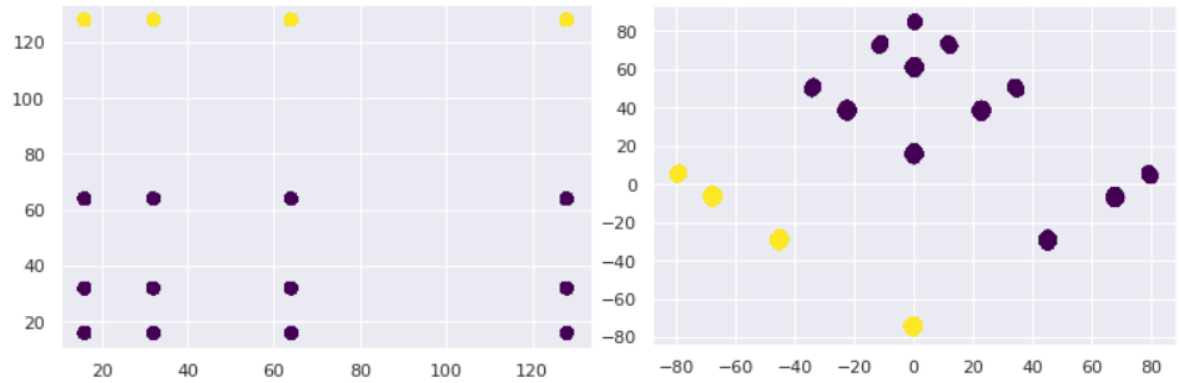
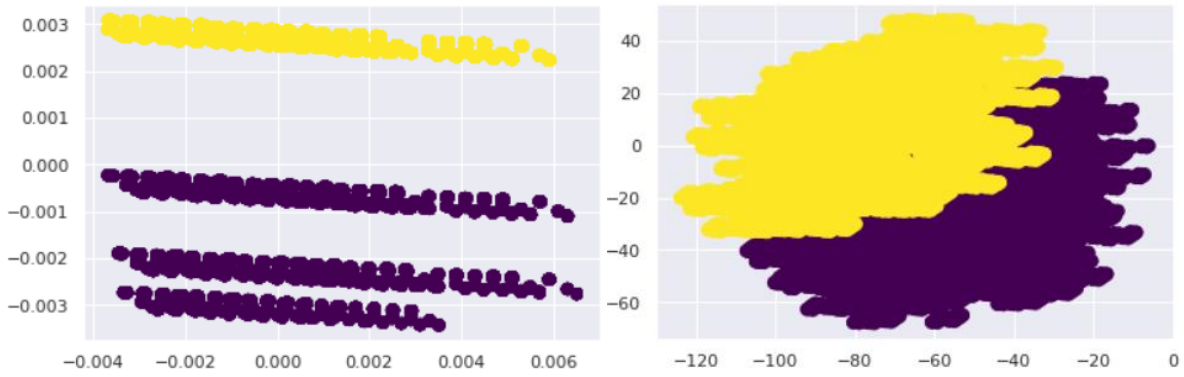


Figure 3 &4



Dataset 2 – Bank Marketing

Expectation Maximization

In this, we will look at I am using 2 components as they were giving the best separation of the clusters. We get the clusters as shown in the graphs.

Expectation Maximization	**Inference**
With Feature Selection	After using the features that we found using Random Forest, again we take 2 clusters and they are separating the data pretty well. (Figure 1).
With PCA	Using the PCs that we found earlier, we get a similar distribution of the 2 clusters, but they are now orthogonal to the original dataset. (Figure 2).
With ICA	Using the ICA results from earlier and 2 clusters. Here ICA is not performing well as the data is not well separated in the graph. Ironically it was working well for K-Means (Figure 3).
With RCA	Again, RCA is not providing any conclusive results as it was in the case of K-Means. (Figure 4).

Since this is a part of soft clustering and there is no accuracy or performance metric to compare the results, we can simply say from the graph if this transformation is working out for this dataset. Here I can choose both Feature Selection and PCA to be providing good clusters but cannot comment on which one is better.

Also, the best clustering is done by K-Means for this dataset, using ICA for data transformation.

Figure 1 & 2

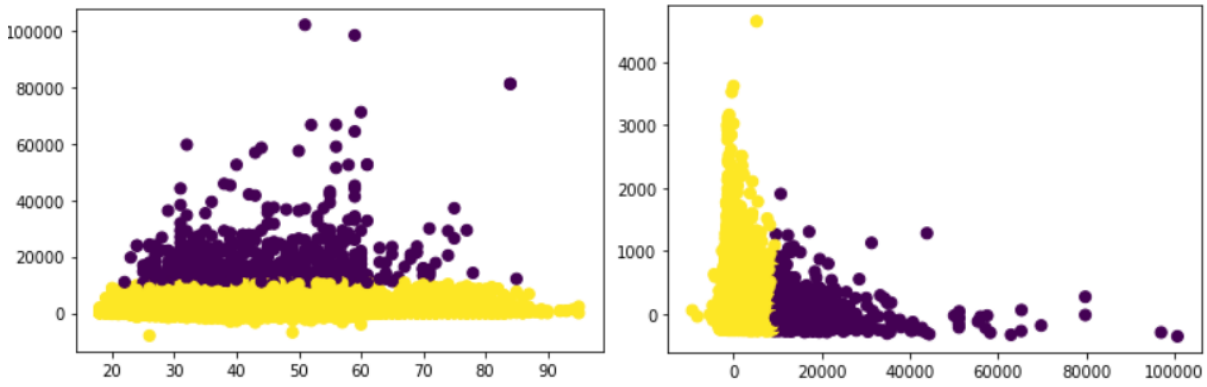
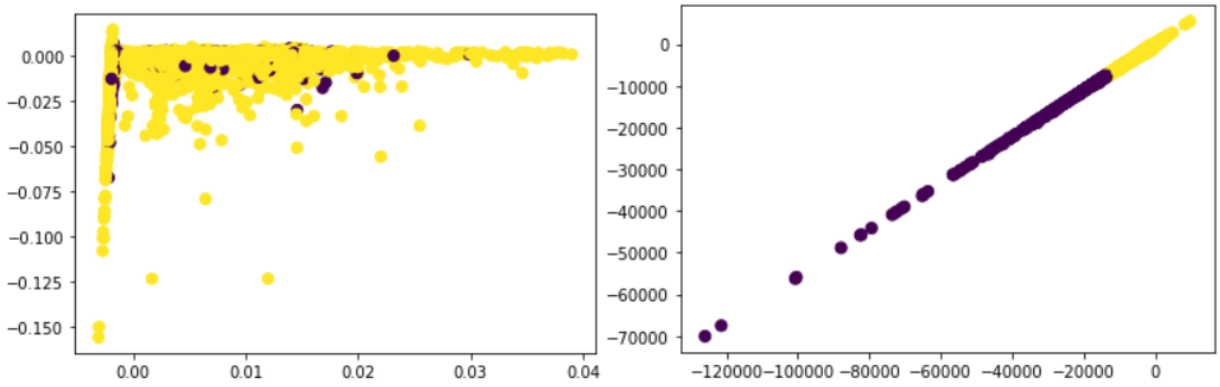


Figure 3 &4

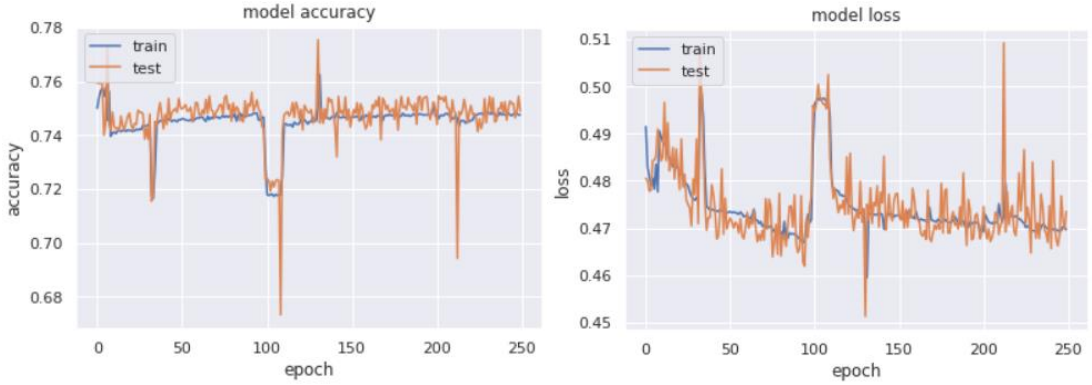


ANN with PCA

Now I am implementing ANN on the best model from the previous assignment using PCA components input dataset.

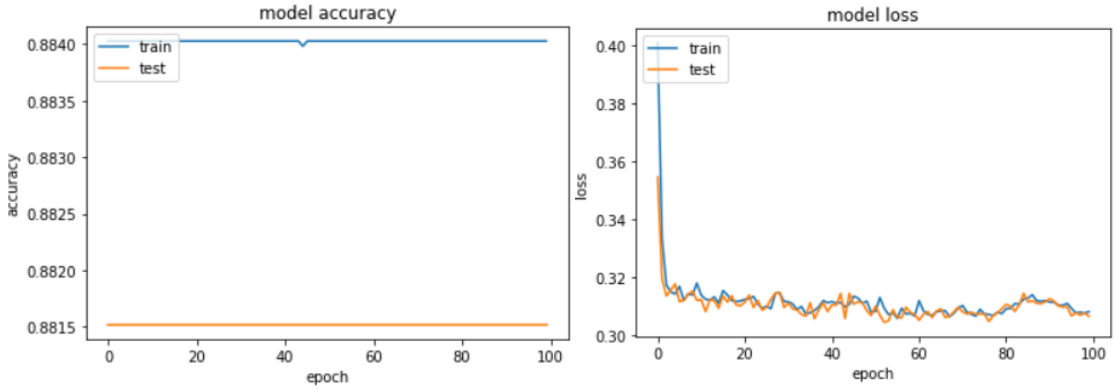
Dataset 1 – GPU Runtime

The model accuracy has gone down by ~20% from the previous neural network, now being 74%. The precision is also reduced to 50%. But if we look at the learning curves, both test and train curves are perfectly overlapping each other, making it a generalized model. So, experimenting with hyperparameters might help improve the accuracy and precision.



Dataset 2 – Bank Marketing

The model accuracy is almost the same as the previous neural network, i.e. 88%. The precision is also the same ~11%. But if we look at the learning curves, both test and train curves are perfectly overlapping each other, making it a generalized model. So, ANN is working better with data transformed by PCA for this dataset.



ANN with K-means and EM results as inputs

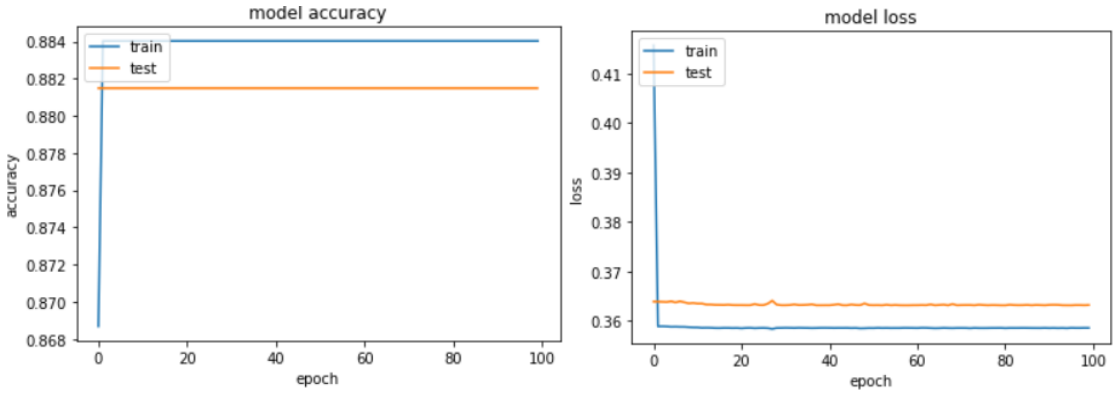
Now we are using the predictions from the K-Means and Expectation Maximization as the input features and using the same target variables as before. Now we have converted our unsupervised problem into supervised learning problem. And since we are using the results of unsupervised learning as the input to the neural network, accuracy of this network will help us judge the performance of the unsupervised algorithms.

Dataset 1 – GPU Runtime

The model accuracy has gone down to 70% from the previous neural network. The precision is also reduced to 46%. But if we look at the learning curves, both test and train curve become constant and are close to one another, making it a generalized model. So, experimenting with hyperparameters might help improve the accuracy and precision.

Dataset 2 – Bank Marketing

The model accuracy is almost the same as the previous neural network, i.e. 88%. The precision is also the same ~11%. But if we look at the learning curves, both test and train curves are perfectly straight and close to one another, making it a generalized model. So, according to ANN results, our K-Means and Expectation Maximization is performing well.



Conclusion

Using data transformation method, we can improve the model by identifying if the data is orthogonally separable or has independent variables. It also helps in generalizing the model well, especially in case of unsupervised learning where there is no performance metric.