Sujeeth Shetty 05/05/2020

BUAN 6341 Applied Machine Learning

ASSIGNMENT 4

Implementation of K-Means & Expectation Maximization

Introduction:

The objective of this project is to implement K-Means & Expectation Maximization clustering algorithms on two Datasets to classify data points into specific groups. This report details the various experiments conducted using the two datasets to understand the effect of feature selection and dimensionality reduction algorithms. It also shows the effect of dimensionality reduction on Artificial Neural Network.

About the Data:

The first data set is of SGEMM GPU kernel performance which consists of 14 features and 241600 records. This data set measures the running time of a matrix-matrix product A*B = C, where all matrices have size 2048 x 2048, using a parameterizable SGEMM GPU kernel with 261400 possible parameter combinations. Out of 14 features, the first 10 are ordinal and can only take up to 4 different powers of two values, and the 4 last variables are binary.

The second dataset is Bank Marketing UCI Dataset published by Banco de Portugal which consists of 20 features and 45307 records. This dataset is used to measures the success of Bank Telemarketing where the goal is to predict if the client will subscribe to a bank term deposit. There are few missing values, and all are coded with the “unknown” label. These missing values are treated as a possible class label. One of the reasons for selecting this dataset is because of the imbalance in class distribution.

Project Outline:

Algorithm Implementation

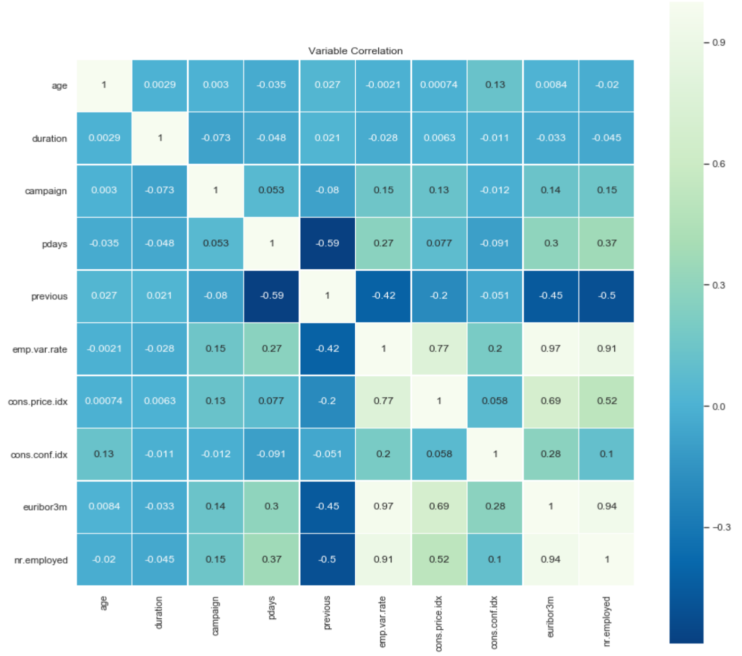
The K-Means and Expectation maximization algorithms are implemented using python “scikit-learn”. Sklearn is a very popular package because of its ease of use. It provides a layer of abstraction on top of Python. Therefore, to make use of the K-Means algorithm & Expectation-Maximization, it’s enough to create an instance of KMeans & GaussianMixture respectively. To measure the performance of the clustering algorithm, Sklearn provides a metric called **Silhouette Score**. This measure checks how much the clusters are compact and well separated. The more the score is closer to one, the better the clustering is.

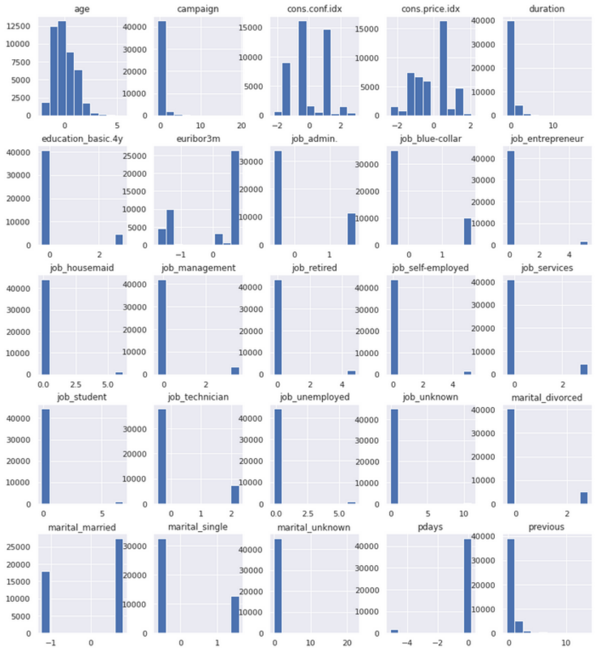
The clustering algorithms are implemented for two different datasets which were originally used for binary classification. The algorithms are implemented with feature selection & dimensionality reduction algorithms like Random Forest, PCA, ICA & Random Projection with options to change the hyperparameters: n_components, n_clusters, etc.

Data Preparation and Exploratory Data Analysis

For **SGEMM GPU Kernel Performance Dataset**, the goal was to cluster the performance of GPU Kernel.

The objective of **Bank Marketing UCI Dataset** is to identify weather the clients will subscribe to a bank term deposited or not. The correlation plot indicated high correlation between employment variation rate, euribor 3-month rate and number of employees(nr.employeed). To deal with multicollinearity, the variables nr.employeed & emp.var.rate are removed from the dataset.





Experimentation and Results

Five tasks were undertaken in which we ran the clustering algorithm K-Means and Expectation Maximization. At least two experiments were conducted on each algorithm where we explored the effect of feature selection & dimensionality reduction. The dimensionality reduction was tested even on Neural Network to compare the performance. These tasks & experiments were repeated both for SGEMM GPU Kernel Performance Dataset and Bank Marketing Dataset. Finally, the results of the experiments were discussed in the results section and further improvement opportunities were discussed.

Tasks & Experimentation:

1 Clustering Algorithms

1.1 SGEMM GPU Kernel Performance Dataset

The goal was to understand the clusters of our class labels by using various clustering algorithms such as K-Means and Expectation Maximization. K-Means and Expectation Maximizations were run on the dataset without any dimensionality reduction.

1.1.1 K-Means & Expectation Maximization

The Elbow method was used to determine the optimal number of clusters. In this method, we run the K-Means clustering on the dataset for a range of values of k and calculate the WCSS (within-cluster sum of squares) for each value of k. Our goal is to choose a small value of k that has a low WCSS and marks the 'elbow' of an arm. As from the figure above, the optimum value of k is 4.

Expectation-Maximization algorithm comes under Gaussian models and is a way of thinking and modeling rather than an algorithm. In this algorithm, clusters are modeled as Gaussian distributions and not by their means. There is some correspondence between all data points and all clusters rather than correspondence between each data point to its cluster, as in the case of k-means clustering. We use the gradient of Bayesian Information Criterion(BIC) to determine the number of clusters. This criterion gives us an estimation of how much is good the GMM in terms of predicting the data we have. We can see that from the starting of cluster five, there is no steep increase in gradient, so we will pick 4 clusters.

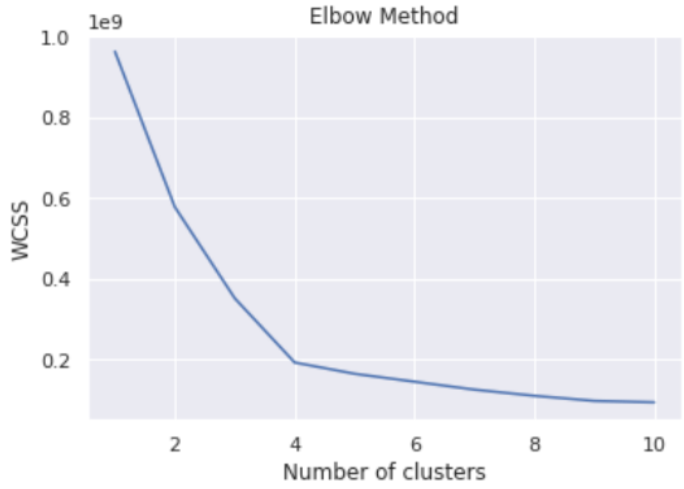


Figure 1 K-Means Elbow Method

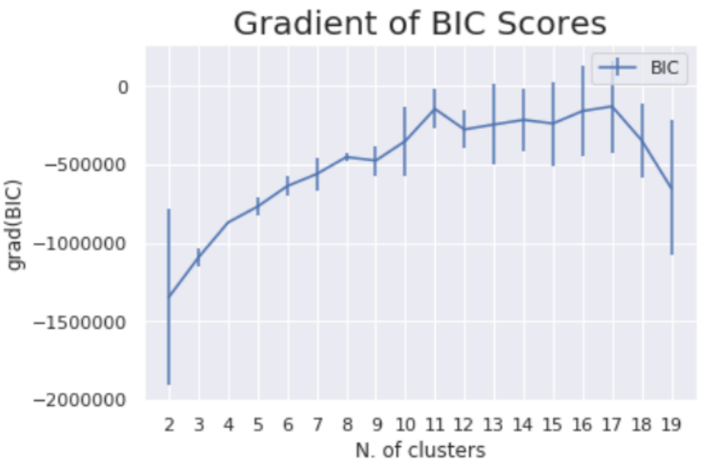


Figure 2 EM - Gradient of BIC Scores

Both the clustering algorithms were implemented on entire data set and its performance was measure using Silhouette Score.

Algorithm	K-Means	Expectation Maximization
Silhouette Score	0.5296	0.5296

1.2 Bank Promotion Dataset

1.2.1 K-Means & Expectation Maximization

Based on the knowledge of the dataset, we just need 2 clusters since we need to identify whether the customers will opt for the Bank Term Deposit or not. But for experimenting, we will check the Elbow Method/Gradient of BIC Scores to find the optimum clusters and evaluate the performance. From the below figure, the optimum value of K is 3.

Model	K-Means(k=2)	K-Means(k=3)	EM (clusters=2)	EM (clusters=3)
Silhouette Score	0.6729	0.7323	0.7143	0.7651

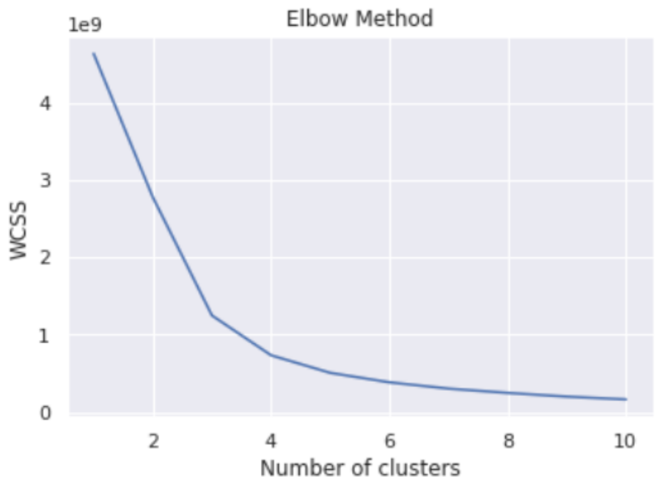


Figure 1 K-Means - Elbow Method

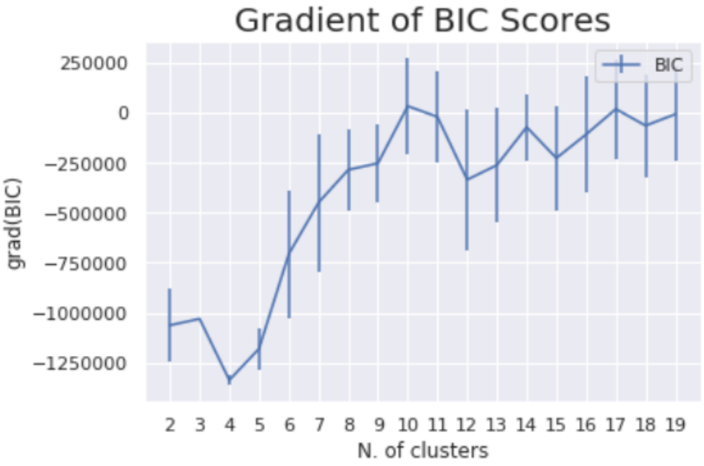


Figure 2 EM - Gradient of BIC Scores

The two-cluster model output was verified with the labeled dataset to measure the performance in term of predictions. The K-Means had a better accuracy of 86% compared to 10.86% of EM.

2 Feature Selection & Dimension reduction

2.1 SGEMM GPU Kernel Performance Dataset

2.1.1 Principal Component Analysis

The PCA was applied to the entire dataset and the variance ratio was plotted. The first two components contribute to 90.6% of the total variance. So, it's good enough to choose only 3 components. Now with these first 2 components, we can jump to one of the most important applications of PCA, which is data visualization. Now, since the PCA components are orthogonal to each other and they are not correlated, we can see four distinct classes.

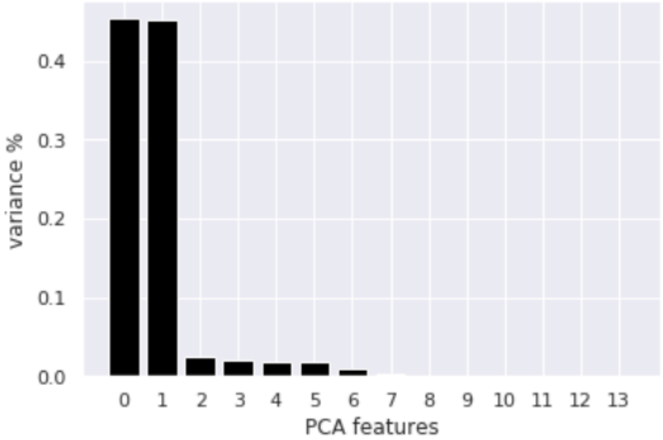


Figure 3 PCA - Variance Information

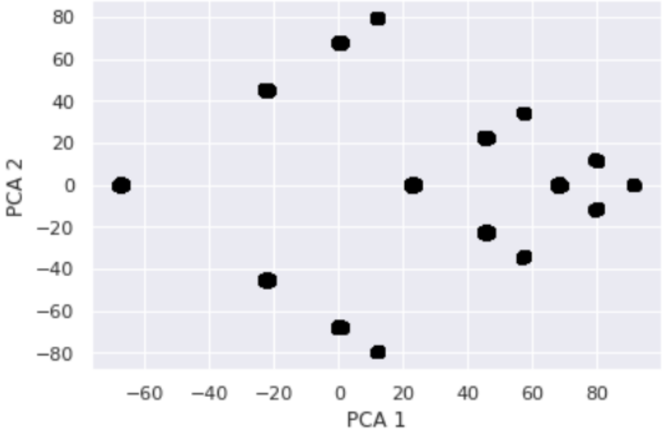
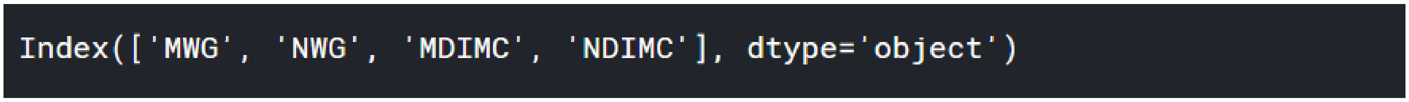


Figure 4 PCA1 vs PCA2

2.1.2 Feature Selection using Random Forest

Random forests consist of 4 –12 hundred decision trees, each of them built over a random extraction of the observations from the dataset and a random extraction of the features. Each tree is also a sequence of yes-no questions based on a single or combination of features. At each node (this is at each question), the three divides the dataset into 2 buckets, each of them hosting observations that are more similar among themselves and different from the ones in the other bucket. Therefore, the importance of each feature is derived from how “pure” each of the buckets is.

After performing feature selection on the original dataset, following features were obtained: ‘MWG’, ‘NWG’, ‘MDIMC’ ,’NDIMC’.



```
Index(['MWG', 'NWG', 'MDIMC', 'NDIMC'], dtype='object')
```

2.2 Bank Promotion Dataset

2.2.1 Principal Component Analysis

The first two components contribute to 99.84% of the total variance. So, it’s good enough to choose just 2 components and from the plot, we can make out two distinct classes.

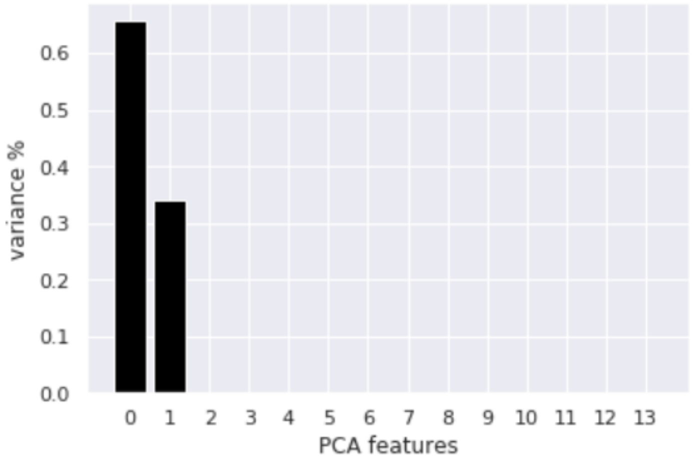


Figure 5 PCA - Variance Information

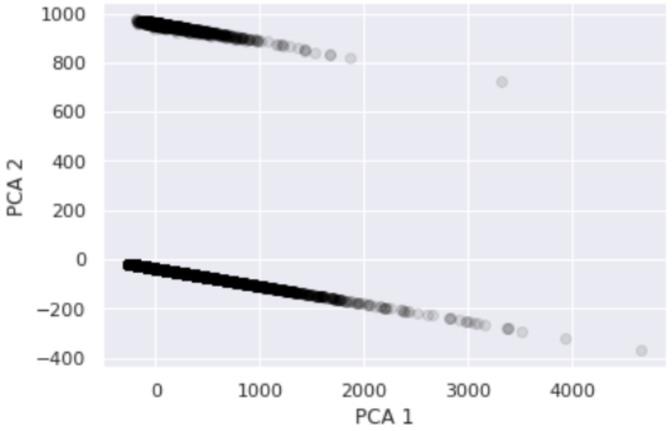
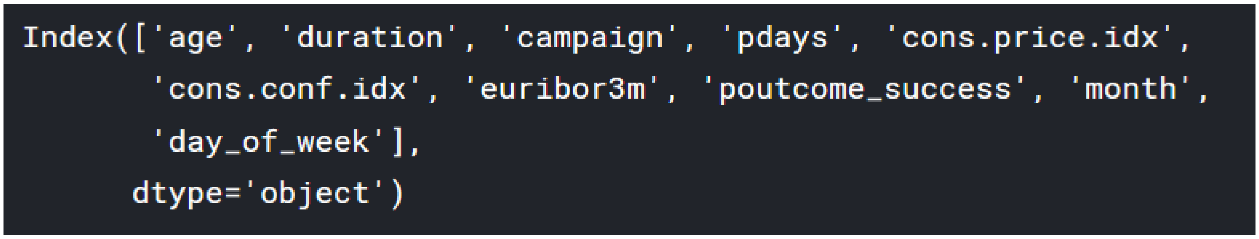


Figure 6 PCA1 vs PCA2

2.2.2 Feature Selection using Random Forest

After performing feature selection on the Bank Promotion dataset, following 10 features were obtained:



```
Index(['age', 'duration', 'campaign', 'pdays', 'cons.price.idx',
       'cons.conf.idx', 'euribor3m', 'poutcome_success', 'month',
       'day_of_week'],
      dtype='object')
```

3 Application of Dimension Reduction on Clustering Algorithm

3.1 SGEMM GPU Kernel Performance Dataset

3.1.1 K-Means

Three components of PCA with a total variance of 93% were applied on K-Means, which gave a clear distinct cluster which can be seen in the 3D plot below. Based on the Silhouette Score we can conclude that applying the ICA dataset on K-Means gave a better result. Also, we can note that K-Means clustering with Dimension Reduction performed better than K-Means on the raw dataset.

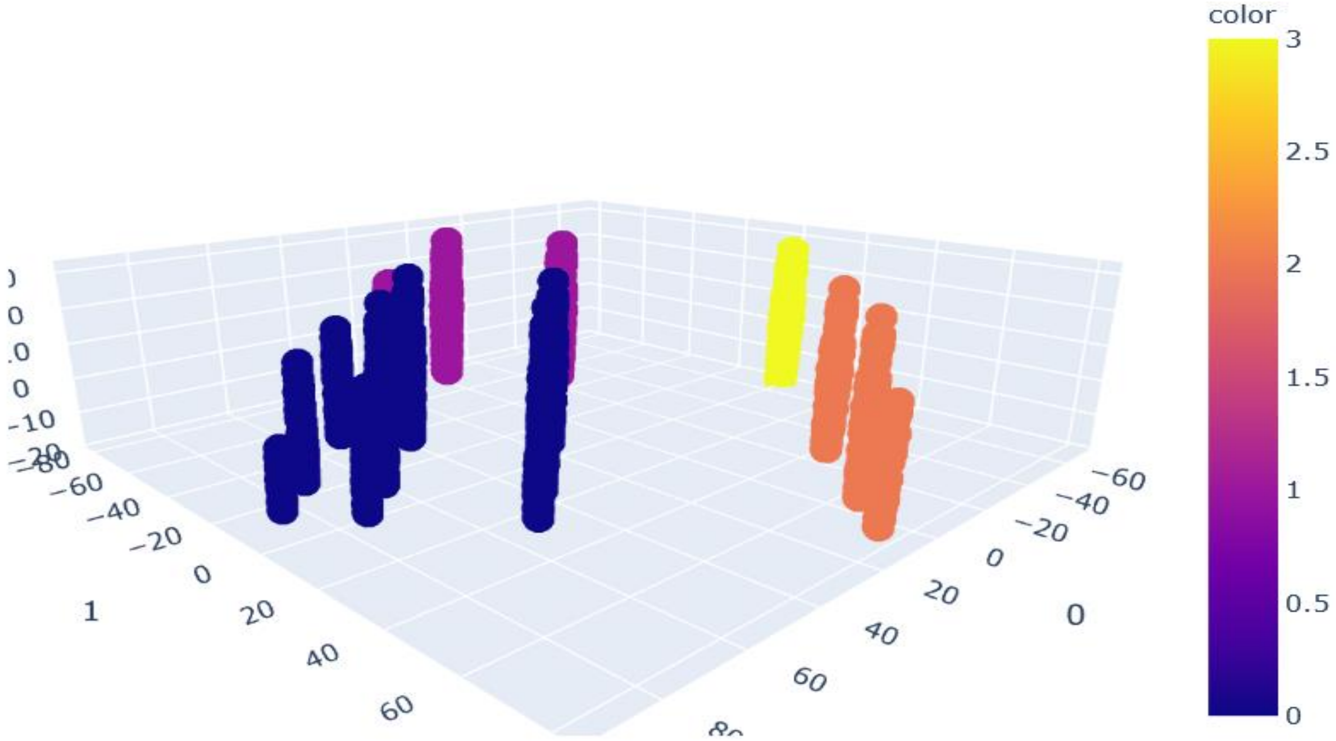


Figure 7 PCA - 3 Components - Cluster Plot

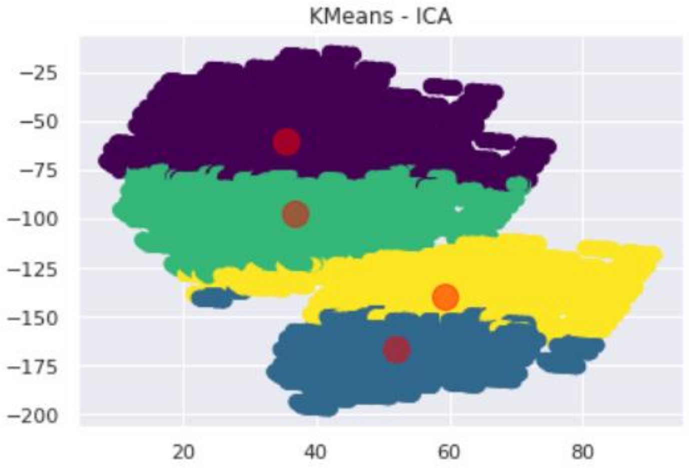


Figure 8 RCA - 2 Components – Clusters

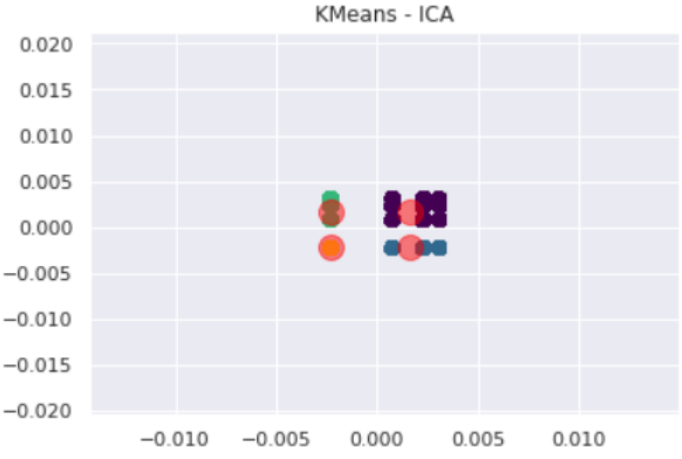
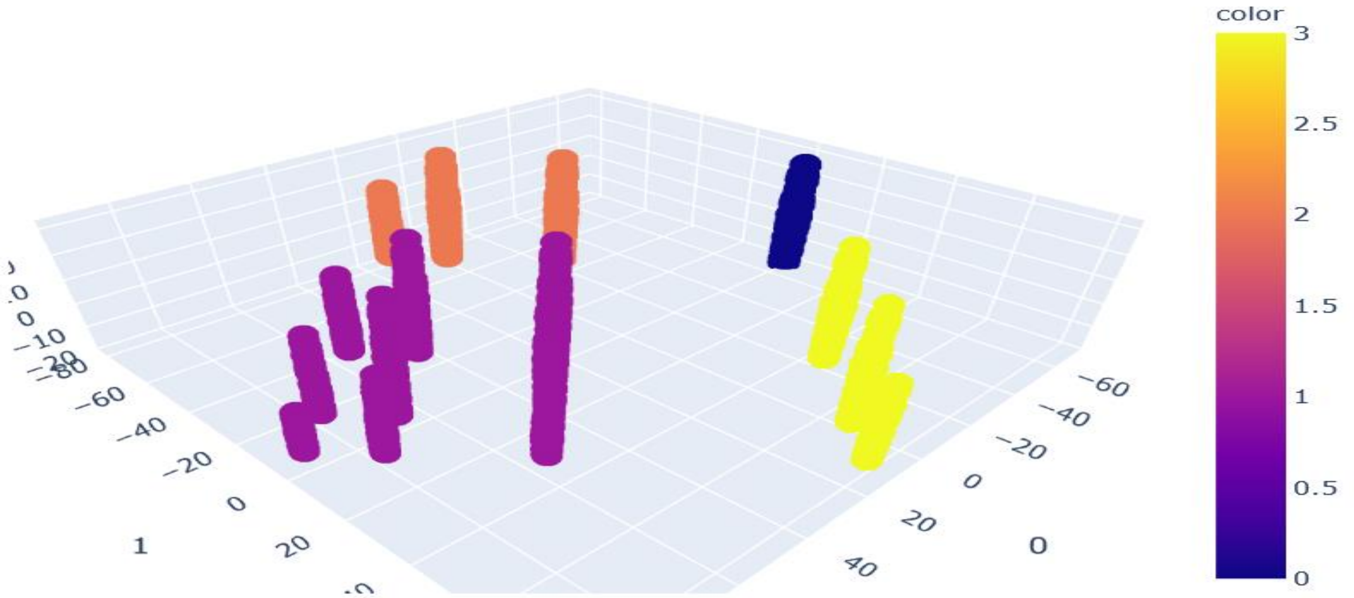


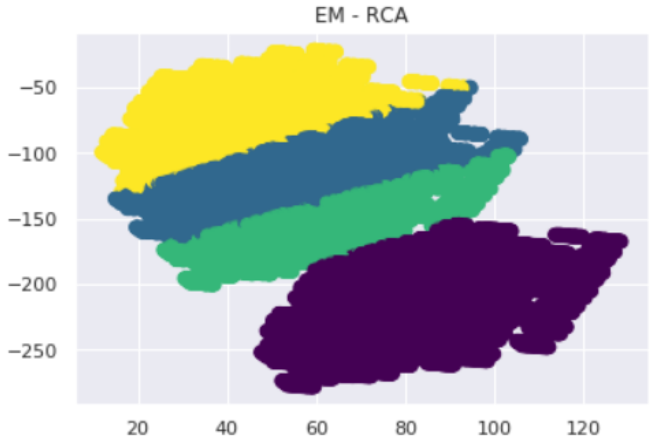
Figure 9 ICA - 2 components - Clusters

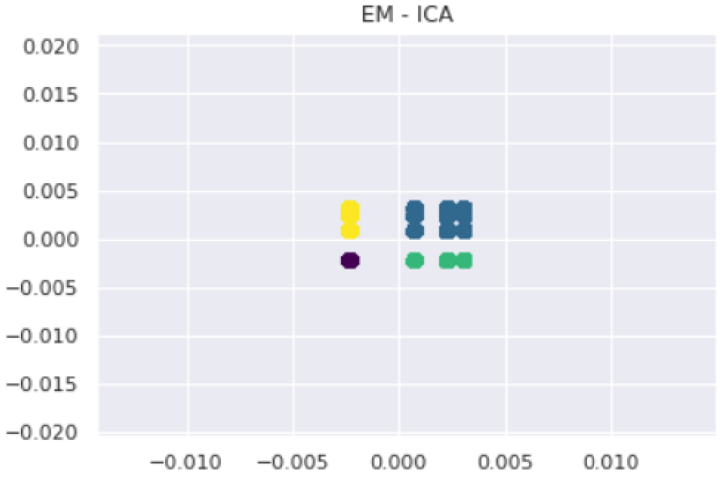
Model	PCA	RP	ICA	Random Forest
Silhouette Score	0.63944	0.1971	0.7161	0.6279

3.1.2 Expectation Maximization

Three components of PCA with a total variance of 93% were applied on K-Means, which gave a clear distinct cluster which can be seen in the 3D plot below. PCA, ICA, RCA, and Random Forest were equally good in clustering the data points. Based on the Silhouette score we can determine that all Dimension Reduction techniques gave well separate and tight clusters.





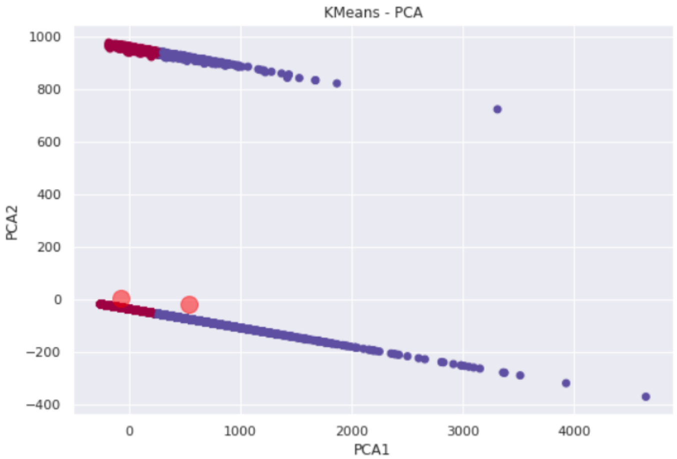


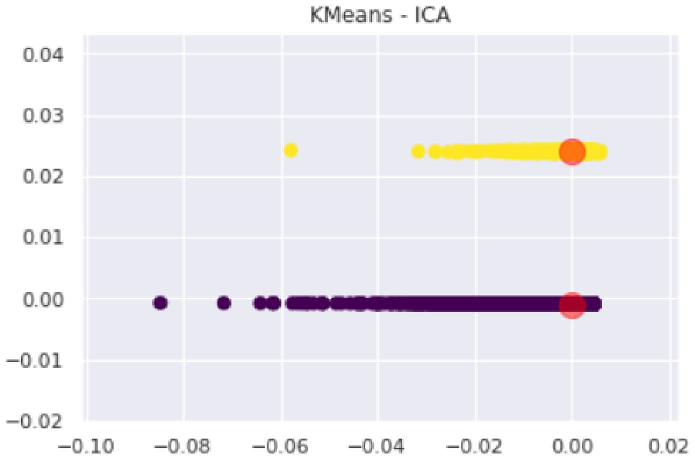
Model	PCA	RP	ICA	Random Forest
Silhouette Score	0.63944	0.6394	0.6394	0.6394

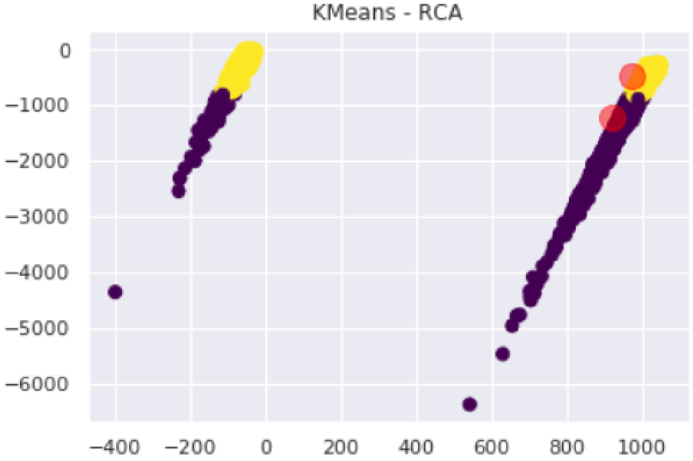
3.2 Bank Promotion Dataset

3.2.1 K-Means

Two components of PCA with a total variance of 99.84% were applied on K-Means, which gave a clear distinct two clusters which can be seen in the plot below. Based on the Silhouette Score we can conclude that applying the ICA dataset on K-Means gave a better result. Also, we can note that K-Means clustering with Dimension Reduction performed better than K-Means on the raw dataset. When the K-Mean ICA cluster output was compared with the existing labels on the dataset, an accuracy of 89.77% & precision 63.7% is achieved.







Model	Silhouette Score
PCA	0.6762
RP	0.6739
ICA	0.8378
Random Forest	0.673

3.2.2 Expectation Maximization

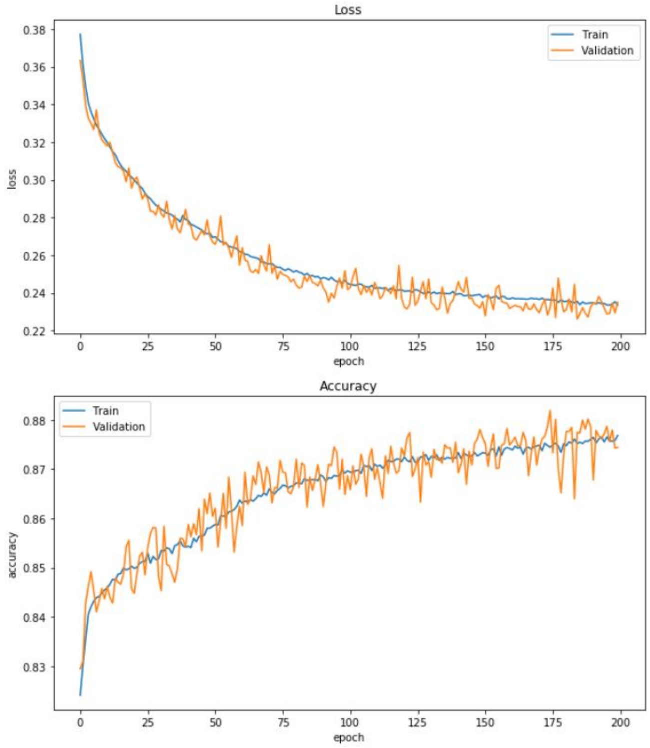
A similar experiment was conducted on Expectation-Maximization and the corresponding Silhouette Score is noted below. Unlike K-Means the performance was relatively consistent.

Model	PCA	RP	ICA	Random Forest
Silhouette Score	0.7174	0.7174	0.7174	0.7174

4 Classification using Neural Network with Dimension Reduction

4.1 SGEMM GPU Kernel Performance Dataset

The best-performed model with 3 layers from the previous assignment has been picked for this experiment. From the accuracy and loss, it is evident that Feature & Dimension Reduction deteriorated the performance. Comparing all four techniques, Random Forest with 4 features performed better with 90.04% accuracy.



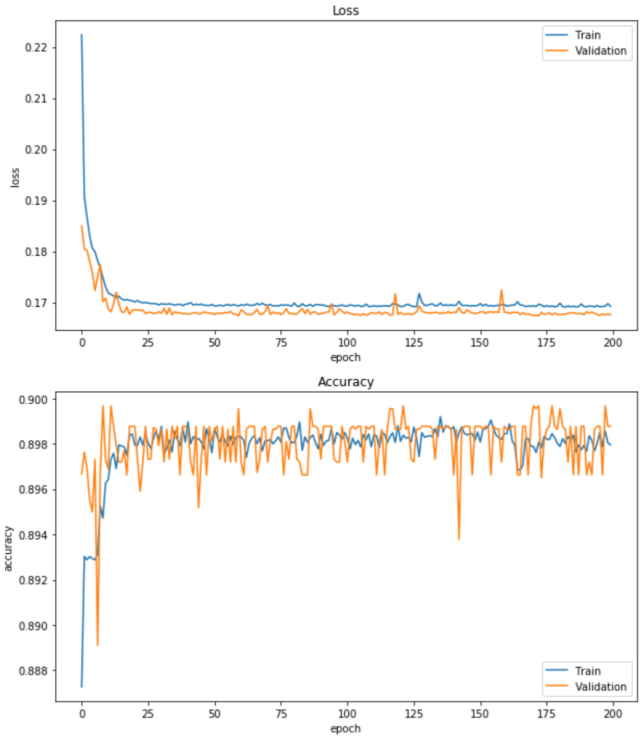


Figure 10 Accuracy & Loss – PCA

Figure 11 Loss & Accuracy Plot - Random Forest

ANN – 3 Layers	PCA	RP	ICA	Random Forest
Train Accuracy	87.68	79.67	81.31	89.80
Validation Accuracy	87.44	80.16	81.49	89.88
Test Accuracy	87.35	79.86	81.25	90.04

4.2 Bank Promotion Dataset

A similar best performing model has been used here and the dimension reduction didn't play any role in improving the performance. Rather the Precision dropped to a maximum of 68% with Random Forest.

ANN – 3 Layers	PCA	RP	ICA	Random Forest
Train Accuracy	90.82	90.81	90.83	91.35
Validation Accuracy	90.54	90.51	90.58	90.44
Test Accuracy	90.48	90.48	90.46	90.92
Precision	0.64	0.64	0.63	0.68

5 Artificial Neural Network on Clustering Data

5.1 SGEMM GPU Kernel Performance Dataset & Bank Promotion Dataset

Using the clustering results from Expectation-Maximization as the new features and applied Neural Network learner on this new data consisting of only clustering results as features and class label as the output. This experiment was performed on both datasets and the train and test accuracy of the algorithms were 100%, which is impressive.

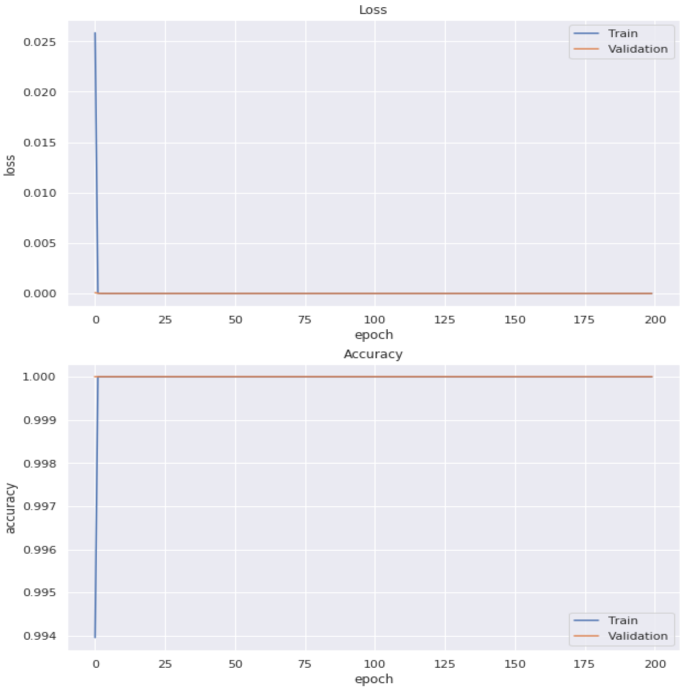


Figure 12 Accuracy & Loss - SGEMM Dataset

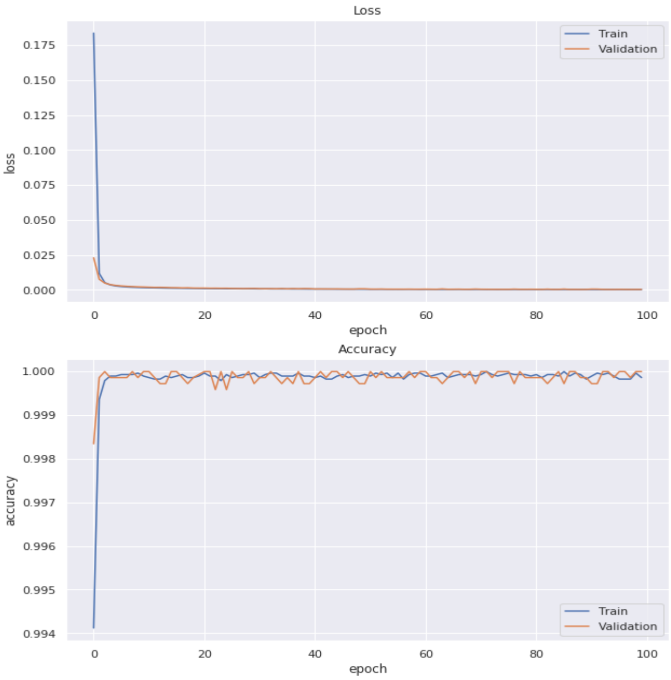


Figure 13 Accuracy & Loss - Bank Promotion Dataset

Results

Since there is no definite answer for finding the right number of clusters as it depends on (a) Distribution shape (b) scale in the data set (c) clustering resolution required by the user. Although finding the number of clusters is a very subjective problem. For example, in **Bank Marketing UCI Dataset** we applied domain knowledge as we have to find whether the customer will opt for bank term or not and determined k=2. But in the case of **the SGEMM GPU Kernel Dataset**, we used a data-driven approach like Elbow Curve to determine the number of clusters. Comparing the performance of K-Means & EM concerning various Feature Selection & Dimension Reduction; K-Means with ICA gave better clustering results for both **SGEMM GPU Kernel Dataset** & **Bank Marketing UCI Dataset**.

In conclusion, Dimensionality Reduction plays a really important role in machine learning, especially when you are working with hundreds of features. This technique, in addition to making the work of feature manipulation easier, it still helps to improve the results of the clustering, as we saw in this post. However, these techniques didn't play a vital role in improving the performance of ANN. In the future, experiments like increasing the components of Dimension Reduction can be conducted to check whether it has any impact on the performance.

Citation

[Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, In press, http://dx.doi.org/10.1016/j.dss.2014.03.001

Rafael Ballester-Ripoll, Enrique G. Paredes, Renato Pajarola. Sobol Tensor Trains for Global Sensitivity Analysis. In arXiv Computer Science / Numerical Analysis e-prints, 2017 (https://128.84.21.199/abs/1712.00233)