ASSIGNMENT 4 REPORT – MANIKANDAN LALITPRASAD (mxl190001)

Dataset 1- SGEMM GPU Kernel Performance

This data set measures the running time of different parameter combinations of parameterizable SGEMM GPU kernel. The dataset contains 241600 records of these feasible combinations. For each tested combination, 4 runs were performed and recorded (measured in milliseconds)

There are 14 parameters, the first 10 are ordinal and can only take up to 4 different powers of two values, and the 4 last variables are binary.

Attribute Information:

Independent variables:
1-2. MWG, NWG: per-matrix 2D tiling at workgroup level: {16, 32, 64, 128} (integer)
3. KWG: inner dimension of 2D tiling at workgroup level: {16, 32} (integer)
4-5. MDIMC, NDIMC: local workgroup size: {8, 16, 32} (integer)
6-7. MDIMA, NDIMB: local memory shape: {8, 16, 32} (integer)
8. KWI: kernel loop unrolling factor: {2, 8} (integer)
9-10. VWM, VWN: per-matrix vector widths for loading and storing: {1, 2, 4, 8} (integer)
11-12. STRM, STRN: enable stride for accessing off-chip memory within a single thread: {0, 1} (categorical)
13-14. SA, SB: per-matrix manual caching of the 2D workgroup tile: {0, 1} (categorical)

Output:
15-18. Run1, Run2, Run3, Run4: performance times in milliseconds for 4 independent runs using the same parameters. They range between 13.25 and 3397.08.
We take the average of these four runs as our target variable and added as a column in our data frame

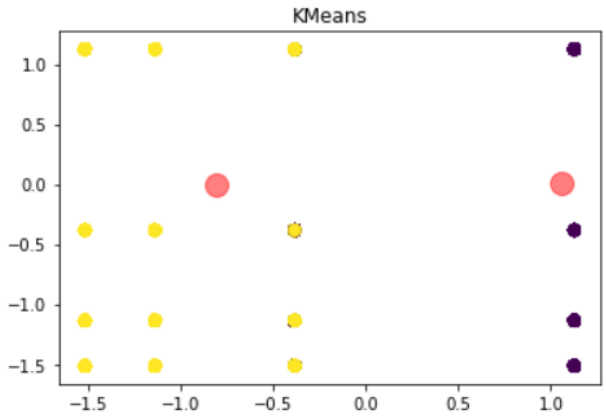
OBJECTIVE:
To implement two learning algorithms using this dataset

1. K-Means
2. Expected Maximization

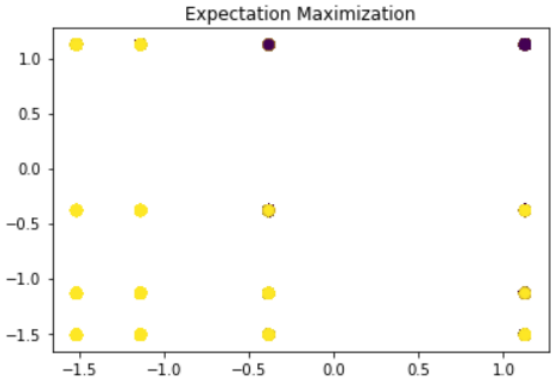
Task 1: K-Means and Expectation Maximization

Our class labels are run_class('1')when Average Run >= Mean and else run_class('0'). We perform cluster analysis to understand the types of cluster formed by the data for both the classes. We run the KMeans and Expectation Maximizations on the dataset without any dimensionality reduction.

```
0.2956815673289183
[[16806 36892]
 [14157  4625]]
```




```
[[13988 39710]
 [ 4974 13808]]
0.3834988962472406

Text(0.5, 1.0, 'Expectation Maximization')
```



The above graphs show the two clusters formed by K-Means and Expectation Maximization algorithms along with their respective confusion matrix. As accuracy cannot be less than 0.5, whenever it goes below 0.5, we predict 0 as 1 and 1 as 0 to get the exact accuracy.

The accuracy of K-means and Expectation Maximization is 29.56% and 38.34%. In this dataset Expected Maximization algorithm performs the better of both the models. Clearly the data does not suit for the K-means and expectation maximization algorithm. Both the class labels are very much merged causing difficulties for the algorithm to separate them.

Task 2: Feature Selection

We perform feature selection algorithm forward selection for selecting the major features based on the best performing features in random forest algorithm.

Filtered Features: 'MWG', 'NWG', 'KWG', 'MDIMC', 'NDIMC', 'KWI', 'VWM', 'VWN', 'STRM', 'STRN', 'SA', 'SB'

The filtered features are normalized. The features are furthered reduced to 6 main features using feature transformation algorithms such as Principal Component Analysis (PCA), Independent Component Analysis (ICA) and Random Component Analysis (RCA). We get separate feature sets for PCA, ICA and RCA for analysis.

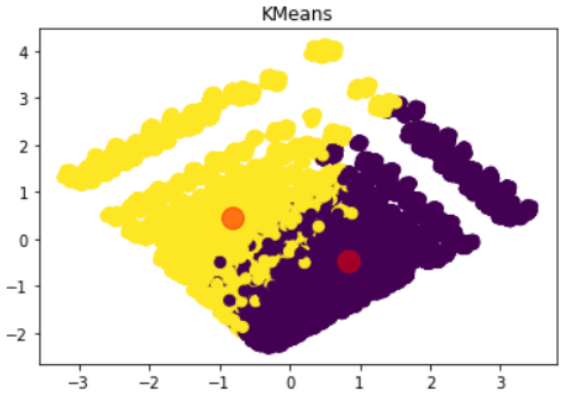
Task 3: Clustering Algorithms after Dimensionality Reduction

We perform the clustering analysis again with features we got through dimensionality reduction algorithms in task 2.

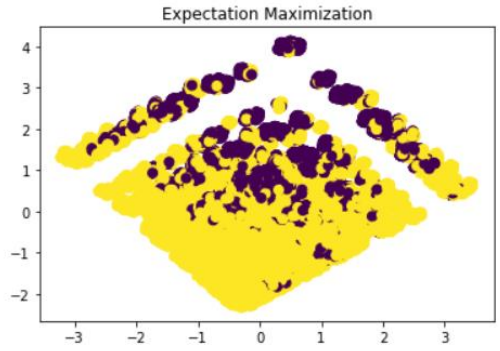
PCA:

The below graph shows cluster group created by the features from PCA

```
0.5865342163355408

Text(0.5, 1.0, 'KMeans')
```



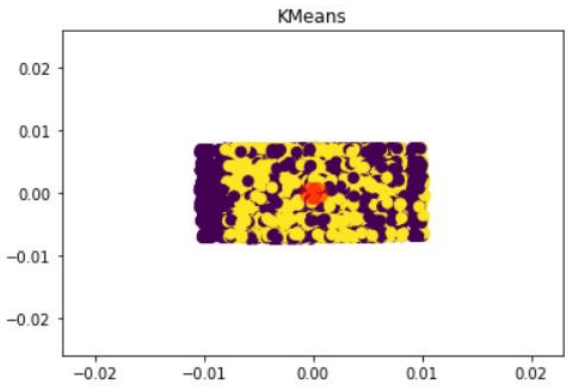

```
[[37798 15900]
 [ 6341 12441]]
0.69314293598234

Text(0.5, 1.0, 'Expectation Maximization')
```



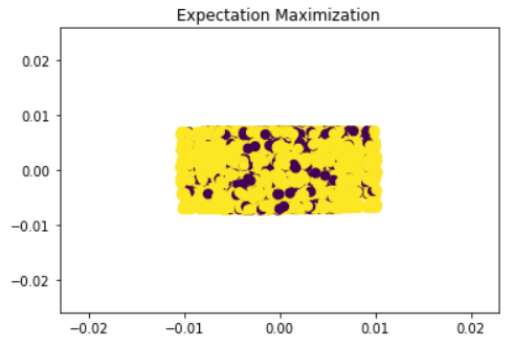
ICA:

The below graph shows the cluster group created by the Features from ICA

```
0.3318846578366446

Text(0.5, 1.0, 'KMeans')
```

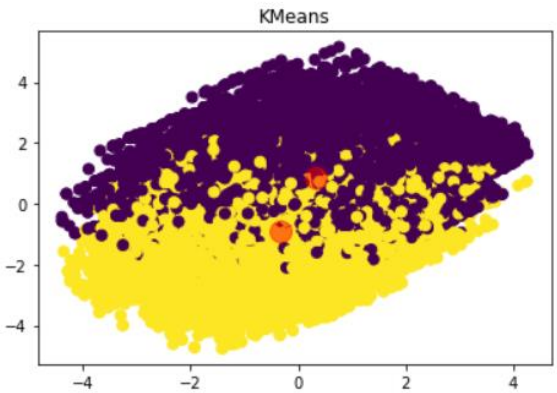



```
[[24109 29589]
 [11340  7442]]
0.43530629139072846

Text(0.5, 1.0, 'Expectation Maximization')
```

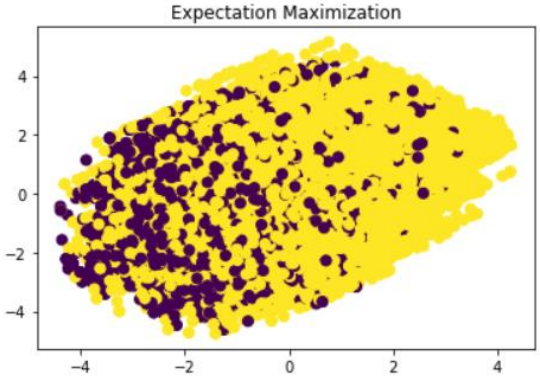


RCA: The below graph shows the cluster group created by the Features from RCA

```
0.5561534216335541

Text(0.5, 1.0, 'KMeans')
```




```
[[22785 30913]
 [10854  7928]]
0.42374448123620306

Text(0.5, 1.0, 'Expectation Maximization')
```



The red dots represent the centers of the respective clusters.

From the graphs, we can clearly see that PCA gives a compact cluster whereas ICA gives a very dispersed cluster. RCA gives optimum cluster of mildly dispersed and compact at the same time. All the clusters from K-means clearly distinguishes the class labels whereas the clusters from Expectation maximization is well dispersed and both the class labels merge into each other causing more errors.

The accuracy of features after performing PCA is 58.65% for K-Means and 69.31% for Expectation Maximization.

The accuracy of features after performing ICA is 33.18% and 43.53% for Expectation Maximization.

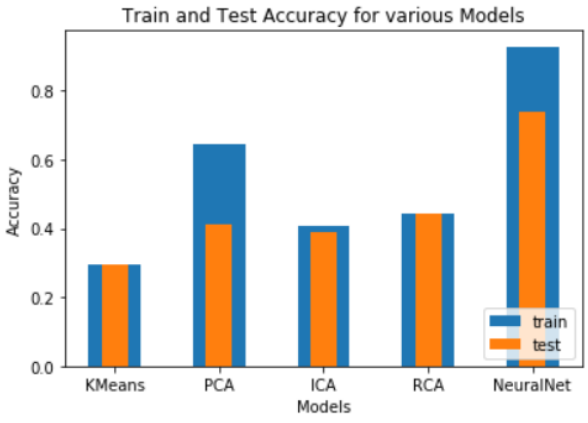
The accuracy of features after performing RCA is 55.61% and 42.37% for expectation Maximization.

K-means performs better on the dataset.

PCA performs the best from the feature transformation algorithm. PCA is the best algorithm which retains the most information from the feature set in most of the cases.

Task 4:

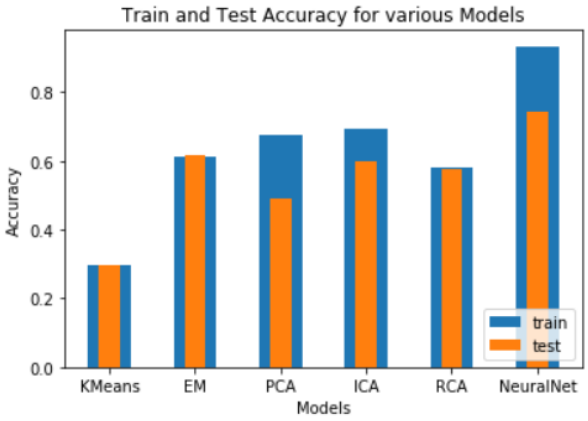
We perform Artificial Neural Networks on the transformed features based on the best model hyper parameters we decided from assignment 3. Since PCA retains the most information, we will use the features from RCA for the neural networks and compare the same with the K-means. The below graph gives the Test and Train accuracy for all the algorithms including Neural Networks.



From the graph, we can clearly see that the transformed features perform way better in Neural Networks than in K-means or Expectation Maximization. The Test accuracy for Neural network is 74.90% and train accuracy is 93.54%. When we perform Neural Networks, we see an astonishing jump in inaccuracy by 15% from the second-best model K-means with PCA. One of the major reasons neural network performs better is due to its ability to form much complex decision boundaries as hypothesis. This ability enables it to classify more data points correctly. The neural network ran comparatively faster than running on the features without dimensionality reduction. Hence dimensionality reduction enables us to increase the speed of few algorithms.

Task 5:

We performed neural network analysis with features from algorithm K-means and expectation maximization from Task 1. The train and test accuracy of the algorithms are 94.12% and 74.91%. The model performed decently when compared with other clustering algorithms.



Data in Transformed Domains:

In SGEMM dataset, both PCA and RCA fluctuated hugely from +10 to -10 whereas ICA ranged mostly between -5 to +5. This may due to the shortage of individual elements available in the dataset. Apparently, ICA performed the poor of all the 3 feature transformation algorithms.

Conclusion:

In clustering algorithms, K-means with PCA performs the best on the dataset. Overall Neural Networks with PCA features performs exceptionally well giving a significant boost to the accuracy and enabling us to form complicated hypothesis functions.

Dataset 2- Australia Weather

The objective is to predict whether rain will come tomorrow or not based on the location's weather conditions in Australia. The dataset provides us with various variables such as Wind speed, humidity, temperature, pressure etc. Occurrence of rain is indicated by 1 and non-occurrence is indicated by 0.

Reason for selecting the Dataset: Rain in Australia is an interesting dataset because we study the weather patterns and the features are diverse and challenging to understand and build models. It has the right number of training samples and features for building various learning algorithms and experiment on them. It provides the right learning platform.

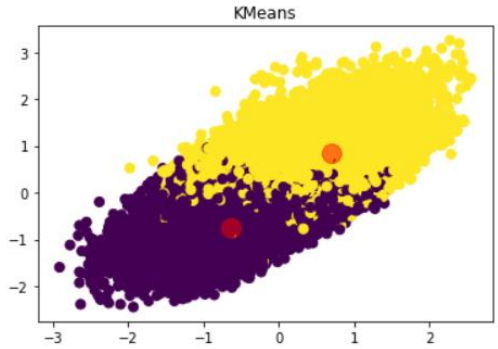
There are null values in the dataset and they are removed using dropna () function. Min-max normalization is done on all the selected features. The dataset is divided into 70/30 split using train_test_split from sklearn. The 70 % of the data is used for the training the model and remaining 30% is used for testing.

Task 1: K-Means and Expectation Maximization

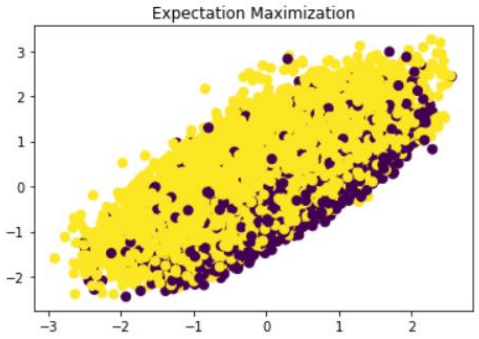
Our class labels are occurrence of rain ('1') and non-occurrence of rain ('0'). We perform cluster analysis to understand the types of cluster formed by the data for both the classes. We run the K-Means and Expectation Maximizations on the dataset without any dimensionality reduction.

Independent Variables: Mintemp, Maxtemp, Rainfall, WindGustSpeed, WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm, Pressure9am, Pressure3pm, Cloud9am, Cloud3pm, Temp9am, Temp3pm, Raintod

```
0.46898263027295284
[[6556 6711]
 [2277 1382]]
```




```
[[6183 7084]
 [1489 2170]]
0.4935011225333806

Text(0.5, 1.0, 'Expectation Maximization')
```



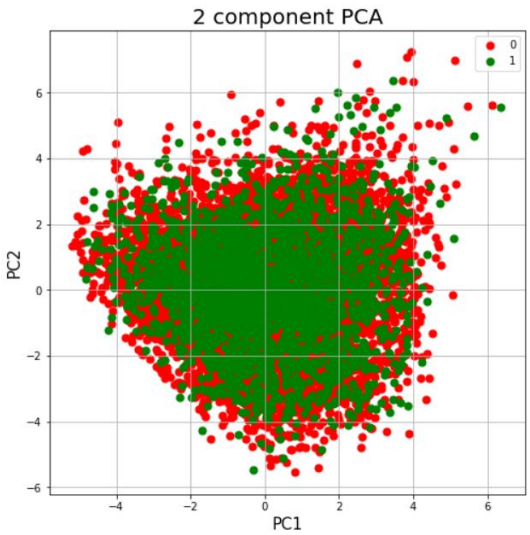
The above graphs show the two clusters formed by K-Means and Expectation Maximization algorithms along with their respective confusion matrix. As accuracy cannot be less than 0.5, whenever it goes below 0.5, we predict 0 as 1 and 1 as 0 to get the exact accuracy.

The accuracy of K-means and Expectation Maximization is 46.89% and 49.35%. In this dataset Expected Maximization algorithm performs the better of both the models. Clearly the data does not suit for the K-means and expectation maximization algorithm. Both the class labels are very much merged causing difficulties for the algorithm to separate them.

Task 2: Feature Selection

We perform feature selection algorithm forward selection for selecting the major features based on the best performing features in random forest algorithm.

Filtered Features: MinTemp, MaxTemp, RainFall, WindGustSpeed, Windspeed3pm, Humidity9am, Humidity3pm, Pressure9am, Pressure3pm, Cloud9am, Cloud3pm, RainTod.



The filtered features are normalized. The features are furthered reduced to 6 main features using feature transformation algorithms such as Principal Component Analysis (PCA), Independent Component Analysis (ICA) and Random Component Analysis (RCA). We get separate feature sets for PCA, ICA and RCA for analysis.

The graph shows the scatter plot between first 2 components in PCA

The graph is plotted between two major components which explains the major variances in our dataset. The graph clearly explains for the poor performance of our models, our data does not form separate clusters and at the same time they are not linearly separable as well. Which makes normal linear functions and clustering functions to segregate the data.

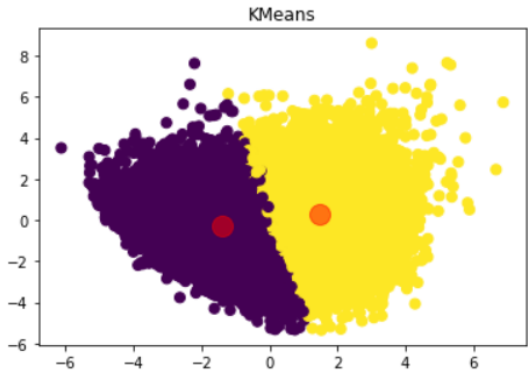
Task 3: Clustering Algorithm after applying Dimensionality Reduction

We perform the clustering analysis again with features we got through dimensionality reduction algorithms in task 2.

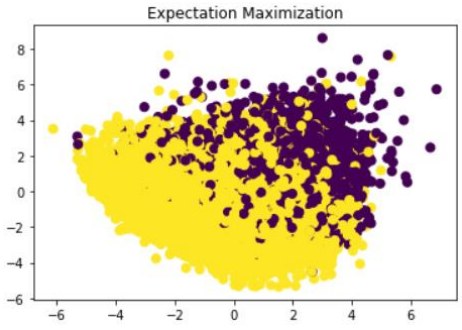
PCA:

The below graph shows cluster group created by the features from PCA

```
0.6490015360983102

Text(0.5, 1.0, 'KMeans')
```



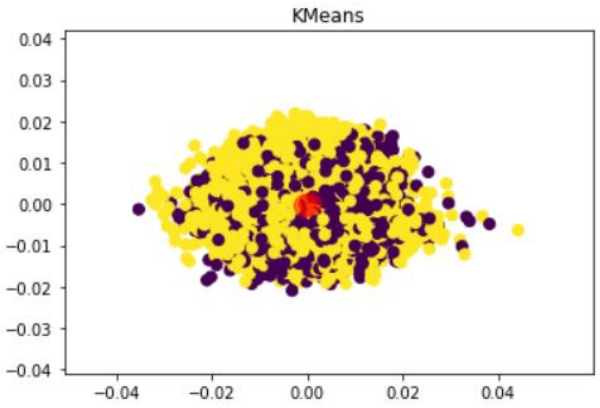

```
[[6919 6348]
 [ 507 3152]]
0.5950017724211273

Text(0.5, 1.0, 'Expectation Maximization')
```



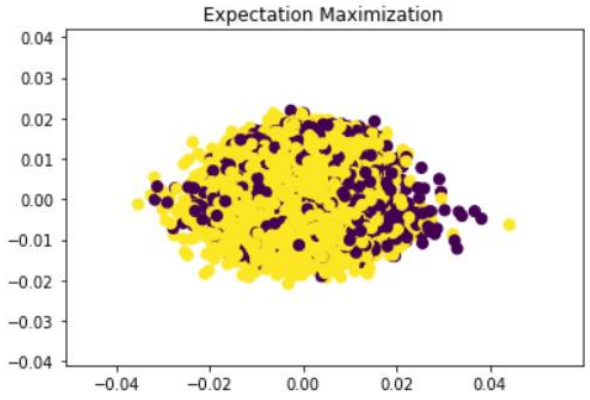
ICA:

The below graph shows the cluster group created by the Features from ICA

```
0.500708968450904

Text(0.5, 1.0, 'KMeans')
```



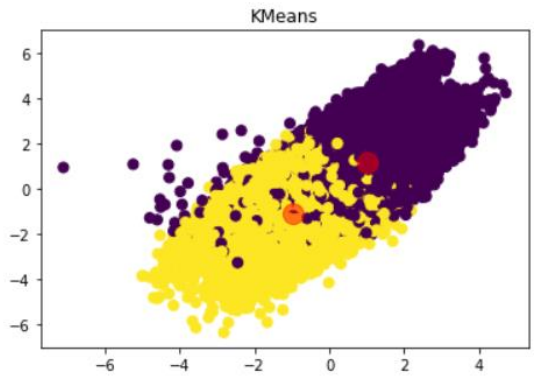

```
[[4787 8480]
 [1679 1980]]
0.39979912560557723

Text(0.5, 1.0, 'Expectation Maximization')
```



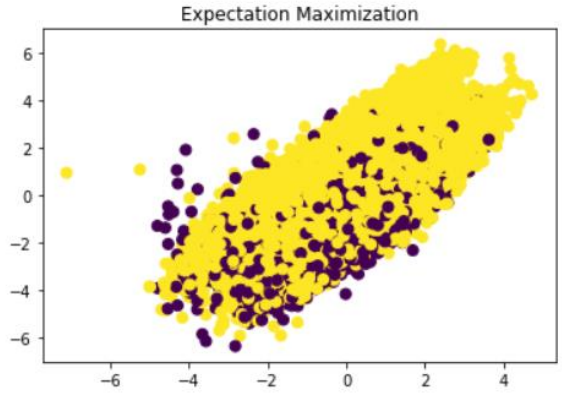
RCA:

The below graph shows the cluster group created by the Features from RCA

```
0.47300011816140847

Text(0.5, 1.0, 'KMeans')
```




```
[[6987 6280]
 [1664 1995]]
0.5306628855015951

Text(0.5, 1.0, 'Expectation Maximization')
```



The red dots represent the centers of the respective clusters.

From the graphs, we can clearly see that PCA gives a compact cluster whereas ICA gives a very dispersed cluster. RCA gives optimum cluster of mildly dispersed and compact at the same time. All the clusters from K-means clearly distinguishes the class labels whereas the clusters from Expectation maximization is well dispersed and both the class labels merge into each other causing more errors.

The accuracy of features after performing PCA is 64.90% for K-Means and 59.50% for Expectation Maximization.

The accuracy of features after performing ICA is 50.07% and 39.97% for Expectation Maximization.

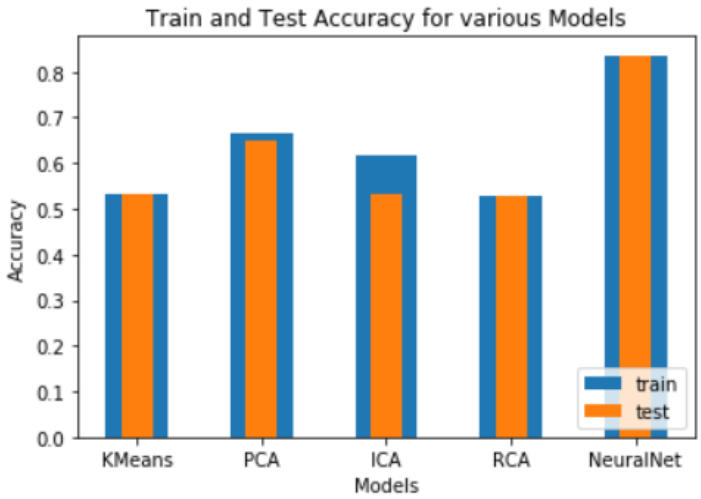
The accuracy of features after performing RCA is 47.30% and 53.06% for expectation Maximization.

K-means performs better on the dataset.

PCA performs the best from the feature transformation algorithm. PCA is the best algorithm which retains the most information from the feature set in most of the cases.

Task 4:

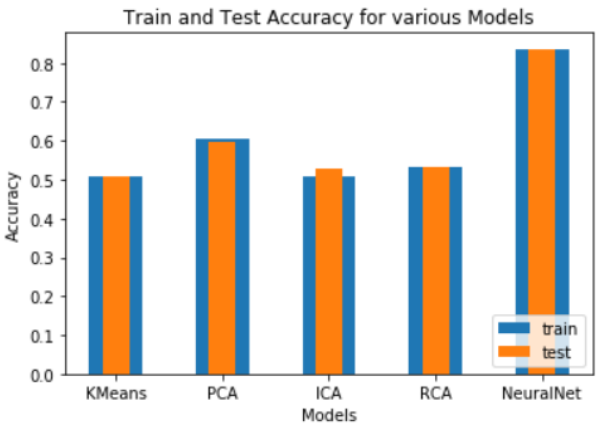
We perform Artificial Neural Networks on the transformed features based on the best model hyper parameters we decided from assignment 3. Since PCA retains the most information, we will use the features from RCA for the neural networks and compare the same with the K-means. The below graph gives the Test and Train accuracy for all the algorithms including Neural Networks.



From the graph, we can clearly see that the transformed features perform way better in Neural Networks than in K-means or Expectation Maximization. The Test accuracy for Neural network is 83.39% and train accuracy is 83.76%. When we perform Neural Networks, we see an astonishing jump in inaccuracy by 20% from the second-best model K-means with PCA. One of the major reasons neural network performs better is due to its ability to form much complex decision boundaries as hypothesis. This ability enables it to classify more data points correctly. The neural network ran comparatively faster than running on the features without dimensionality reduction. Hence dimensionality reduction enables us to increase the speed of few algorithms.

Task 5:

We performed neural network analysis with features from algorithm K-means and expectation maximization from Task 1. The train and test accuracy of the algorithms are 77.97% and 77.79%. The model performed decently when compared with other clustering algorithms.



Data in Transformed Domains:

In Rain in Australia dataset, both PCA and RCA fluctuated hugely from +10 to -10 whereas ICA ranged mostly between -0.5 to +0.5. This may due to the shortage of individual elements available in the dataset. Apparently, ICA performed the poor of all the 3 feature transformation algorithms.

Conclusion:

In clustering algorithms, K-means with PCA performs the best on the dataset. Overall Neural Networks with PCA features performs exceptionally well giving a significant boost to the accuracy and enabling us to form complicated hypothesis functions.