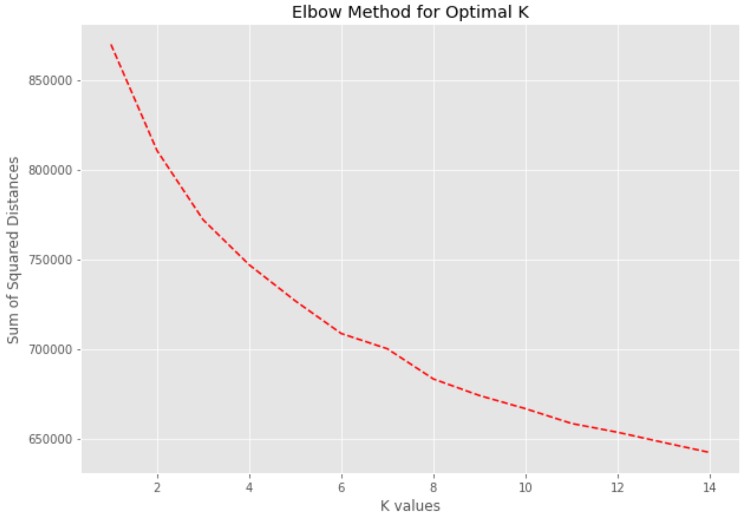
## Assignment 4: K-Means, Expectation maximization clustering algorithms ICA, PCA, Randomized projection feature selections algorithm

**Dataset – Average GPU runtime:**

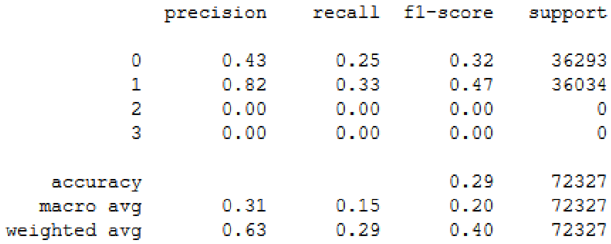
The dataset (SGEMM GPU kernel performance) dataset can be downloaded at:
https://archive.ics.uci.edu/ml/datasets/SGEMM+GPU+kernel+performance#
There are 14 parameters. The first 4 are ordinal and the last four variables are binary. The dataset has total 241600 data entries and 18 features with the last four being the runtime measurement.

**Task 1:**

Applied K-means algorithm on the dataset. Plotted elbow curve graph to obtain optimal number of clusters.



Based on the above graph the slope of the line stops decreasing dramatically after k value = 4. So for this dataset we will use no. of clusters = 4.



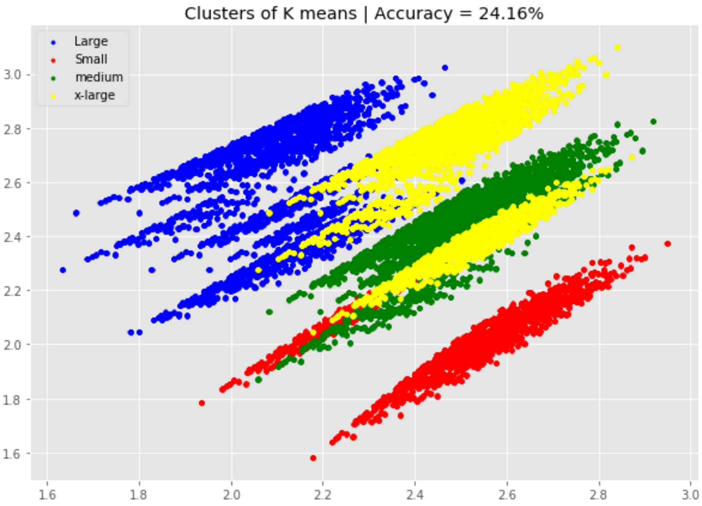
```
              precision    recall  f1-score   support

           0       0.43      0.25      0.32     36293
           1       0.82      0.33      0.47     36034
           2       0.00      0.00      0.00         0
           3       0.00      0.00      0.00         0

    accuracy                           0.29     72327
   macro avg       0.31      0.15      0.20     72327
weighted avg       0.63      0.29      0.40     72327
```

Train accuracy: 24.163

Test accuracy : 29.1537

The test dataset show a moderate data accuracy with average precesion of just 0.63.K-means performs better in the test dataset with average precision of about 0.63 which is better than the training dataset.

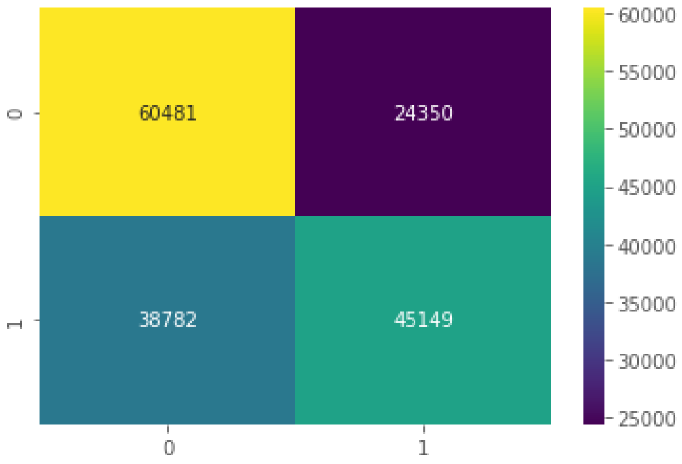
- The clusters when plotted are not very distinguisable but some of the clusturs are slightly seperable visually. The algorithm worked but not efficiently enough.
- They do not line up with the labels.
- Points are not compact and spread across the graph.
- The main reason behind these types of clusters would be the clustering algorithm. Expectation maximization performs better in forming the clusters and has better accuracy.



**Expectation maximization :**

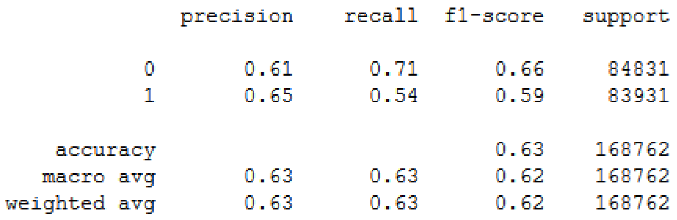
Train accuracy : 62.59

Test accuracy : 62.41



Expectation maximization performs lot better than K means getting a trian accuracy of 62.59 and test accuracy of 62.41.

The average precision achieved is moderate with value around 0.63 which tells that it is able to clustur the dataset well.

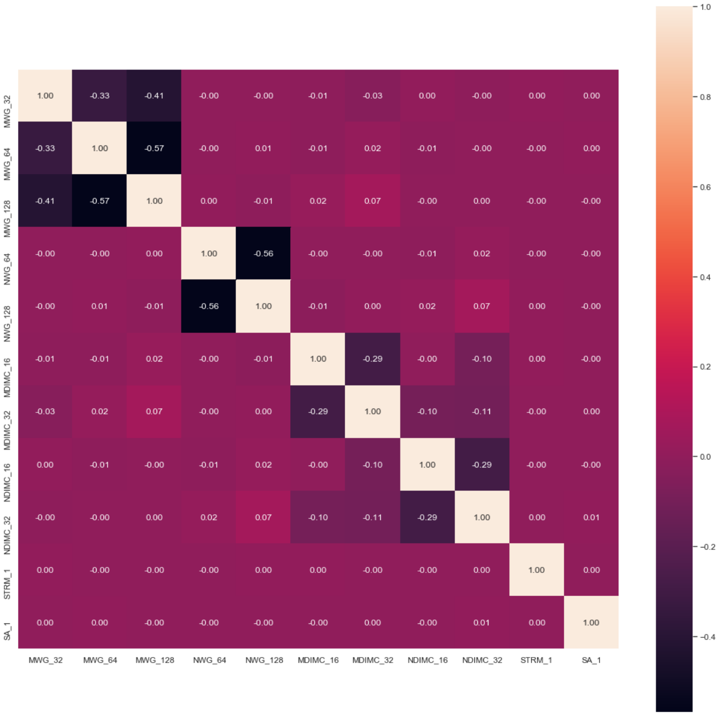


```
              precision    recall  f1-score   support

           0       0.61      0.71      0.66     84831
           1       0.65      0.54      0.59     83931

    accuracy                           0.63    168762
   macro avg       0.63      0.63      0.62    168762
weighted avg       0.63      0.63      0.62    168762
```

| **Clustering algorithm** | **Train accuracy** | **Test accuracy** | **Precision** | **Recall** | **f1-score** |
|---|---:|---:|---:|---:|---:|
| K-means | 24.163 | 29.1537 | 0.63 | 0.29 | 0.4 |
| Expectation maximization | 62.59 | 62.41 | 0.63 | 0.62 | 0.62 |

**Task 2:**
**Dimensionality reduction**

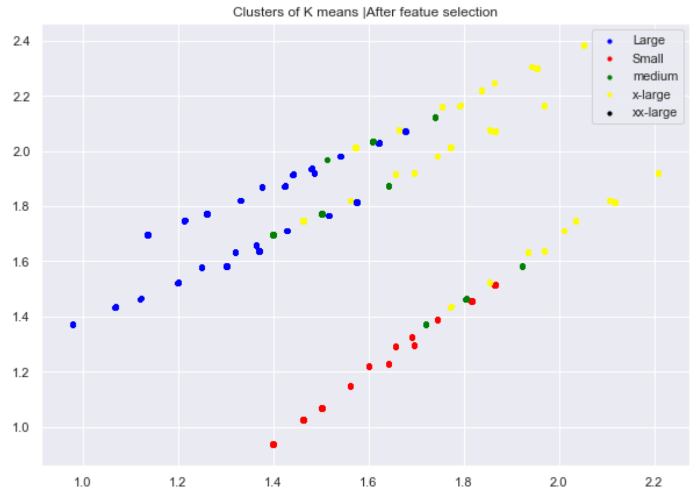
Before: there were huge numbers of negative correlation within the data.



Features selected : 'MWG_32', 'MWG_128', 'NWG_64', WG_128', 'MDIMC_16', 'MDIMC_32', 'NDIMC_16', 'NDIMC_32', 'STRM_1', 'SA_1'

- Number of features dropped as a consequence of dimensionality reduction. Therefor we now have all positive co-related values.

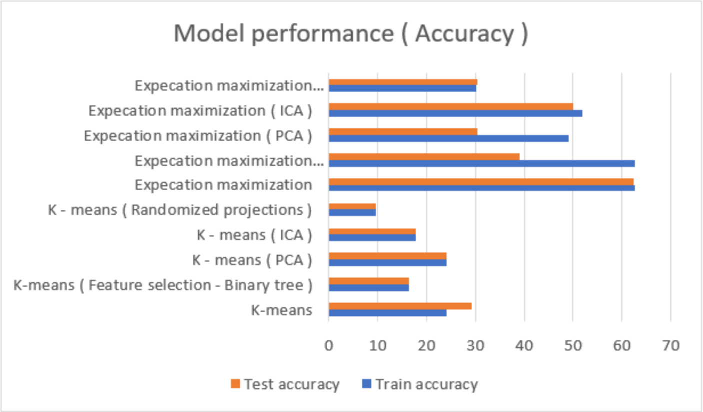
After applying feature selection, we plotted the dataset. The plots appear more separated and individually distinguishable. The clusters do not overlap that much. But since the k-means does not perform that good even after feature selection so the clusters are not that separable.

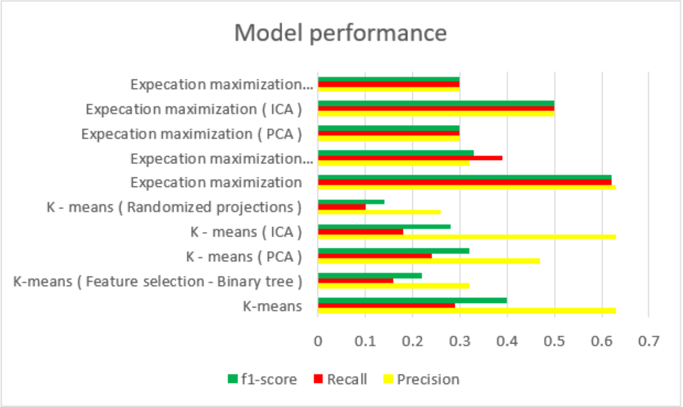


**Task 3:**

**Feature selected using binary tree:** 'MWG_32', 'MWG_128', 'NWG_64',  WG_128', 'MDIMC_16', 'MDIMC_32', 'NDIMC_16', 'NDIMC_32', 'STRM_1', 'SA_1'

| Clustering algorithm | Train accuracy | Test accuracy | Precision | Recall | f1-score |
|---|---|---|---|---|---|
| K-means | 24.163 | 29.1537 | 0.63 | 0.29 | 0.4 |
| K-means (Feature selection - Binary tree) | 16.4521 | 16.453 | 0.32 | 0.16 | 0.22 |
| K - means (PCA) | 23.9917 | 24.0145 | 0.47 | 0.24 | 0.32 |
| K - means (ICA) | 17.7836 | 17.884 | 0.63 | 0.18 | 0.28 |
| K - means (Randomized projections) | 9.601 | 9.5206 | 0.26 | 0.1 | 0.14 |
| Expectation maximization | 62.59 | 62.41 | 0.63 | 0.62 | 0.62 |
| Expectation maximization (Feature Selection - Binary tree) | 62.59 | 39 | 0.32 | 0.39 | 0.33 |
| Expectation maximization (PCA) | 49.18 | 30.32 | 0.3 | 0.3 | 0.3 |
| Expectation maximization (ICA) | 51.91 | 49.94 | 0.5 | 0.5 | 0.5 |
| Expectation maximization (Randomized projections) | 30.1 | 30.32 | 0.3 | 0.3 | 0.3 |





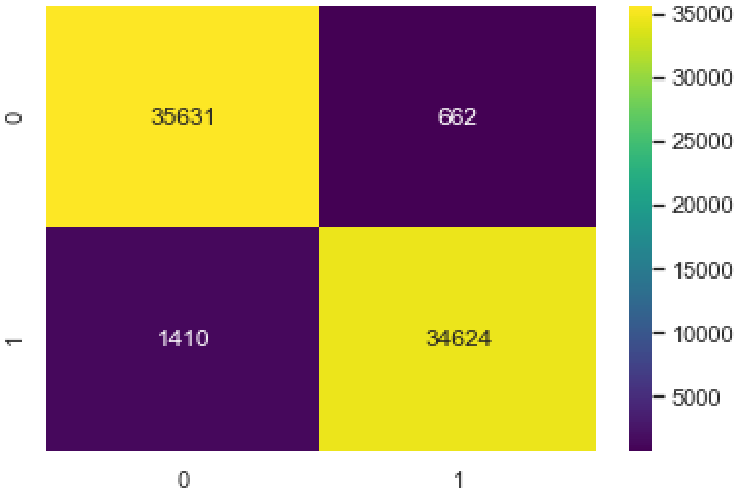
- Applyed the clusturing algorithm after applying feature dimensionality reduction algorithm. Expectation maximization gives the best train set accuracy.
- Model train accuracy is best for expectation maximization ( ICA ) and expectation maximization.
- K – means performance really poorly with no dimensionality reduction algorithm, but its performance changes after applying PCA, ICA, Randomized projections and feature selection (Bineary tree)

- K – means even with the feature reduction performs really bad in comparison to expectation maximization.
- K-means (randomized projection) pricision is really low as compared to other models.
- We can observe by the above data that the clusters are formed differently for each clustering algorithm of PCA, ICA, feature selections and randomized projections. Clustering algorithm without feature selection gives the best accuracy till now for this dataset.
- ICA is performing satisfactory with expectation maximization, but does not perform well with k – means algorithm.

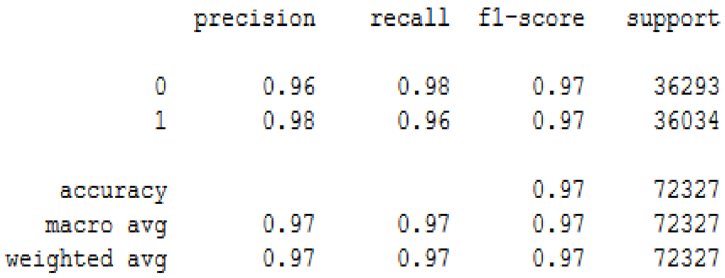
**Task 4:**

**Neural network :**

Train accuracy : 97.1658
Test accuracy : 97.1352

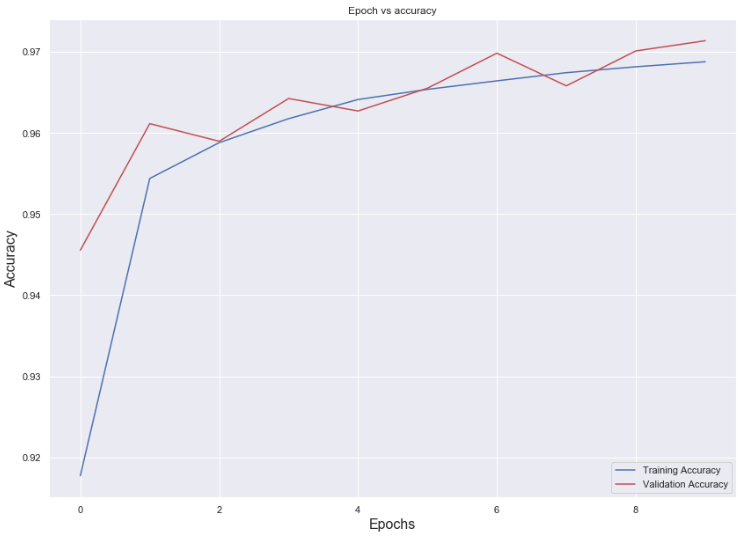


- As you can see neural network of relu, tanh, tanh, sigmoid gave the best result in the previous assignment. In this dataset it gives around 97.1352% accuracy which is really good and better than just performing neural network model.



```
              precision    recall  f1-score   support

           0       0.96      0.98      0.97     36293
           1       0.98      0.96      0.97     36034

    accuracy                           0.97     72327
   macro avg       0.97      0.97      0.97     72327
weighted avg       0.97      0.97      0.97     72327
```

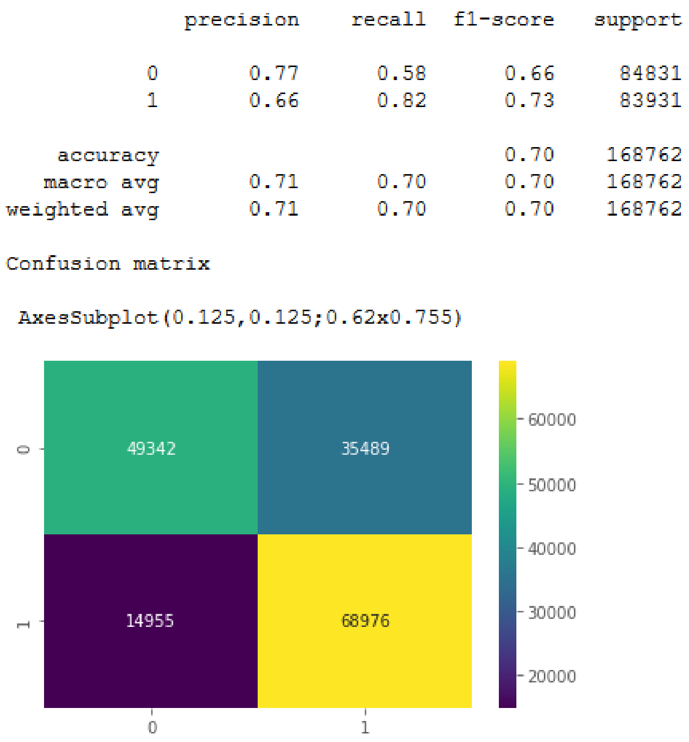
We can see how the train and test data set's accuracy increases as the number of epochs increases. The model performs good in comparison to all the previous experiment.



**Task 5 :**

Test accuracy: 70.1093

- Considering the dataset of the 1st task as input for the neural network, the accuracy achieved for this dataset with output as its labels is good with the value of 70.1.
- The model is not performing as good as in the previous scenario but having an accuracy of 70.1093 is good to consider that the model is able to predict clusters properly.



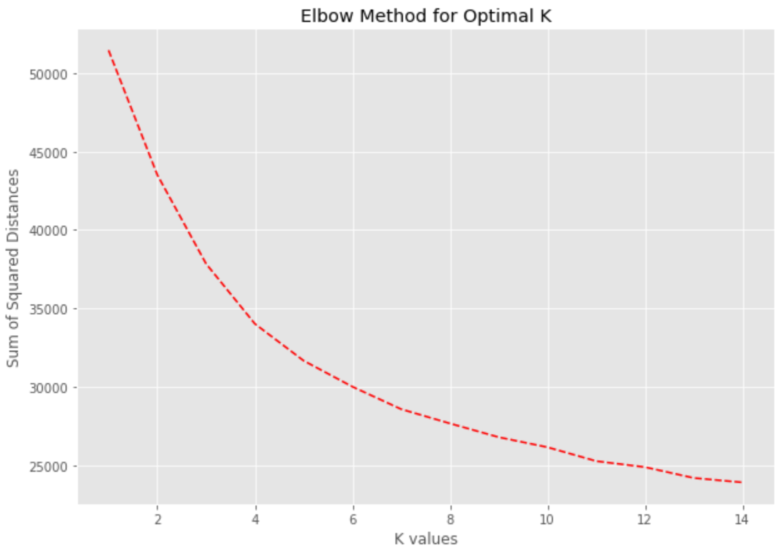
```
              precision    recall  f1-score   support

           0       0.77      0.58      0.66     84831
           1       0.66      0.82      0.73     83931

    accuracy                           0.70    168762
   macro avg       0.71      0.70      0.70    168762
weighted avg       0.71      0.70      0.70    168762

Confusion matrix

 AxesSubplot(0.125,0.125;0.62x0.755)
```

**Dataset – Mushroom-classification:**

The dataset is obtained from Kaggle. The following is the link to the dataset. https://www.kaggle.com/uciml/mushroom-classification. This dataset contains hypothetical samples of gilled mushrooms in the Agarics and Lepiota family. There are 23 features such as cap-shape, cap-size, cap-color, bruises etc.

**Task 1:**

Applied K-means algorithm on the dataset. Plotted elbow curve graph to obtain optimal number of clusters.



Based on the above graph the slope of the line stops decreasing dramatically after k value = 4. So for this dataset we will use no. of clusters = 4.

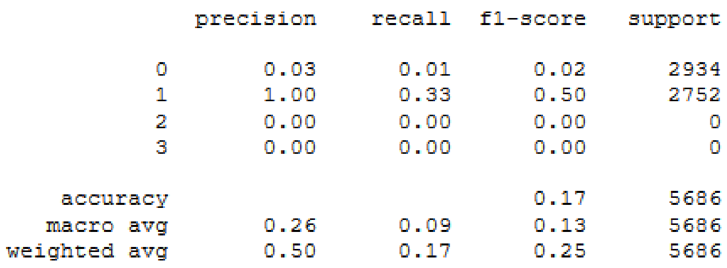
Train accuracy: 10.787

Test accuracy : 16.6549

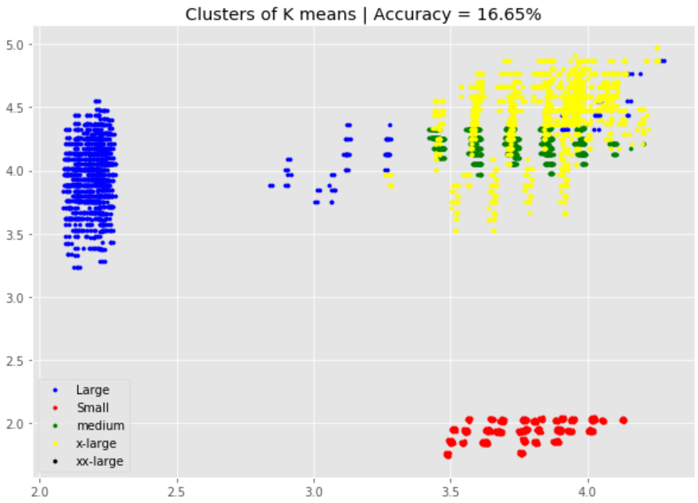
The test dataset show a very low data accuracy with average precesion of just 0.13.

K-means performs better in the test dataset with average precision of about 0.5 which is better than the training dataset.

- The clusters when plotted are not very distinguisable but some of the clusturs are slightly seperable visually. The algorithm worked but not efficiently enough.
- They do not line up with the labels.
- Majority of the points are compact but some of them are spread out in different clusters.
- The main reason behind these types of clusters would be the information of these variables are not sufficient to form these clusters. We need more variables to test out different scenerios.



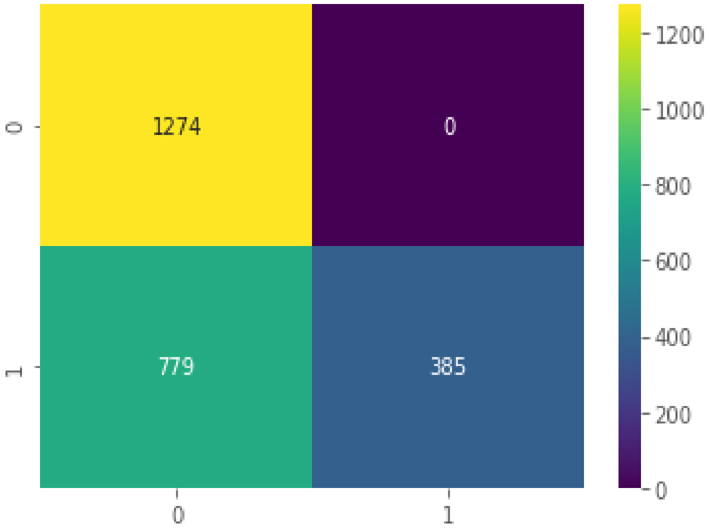
```
              precision    recall  f1-score   support

           0       0.03      0.01      0.02      2934
           1       1.00      0.33      0.50      2752
           2       0.00      0.00      0.00         0
           3       0.00      0.00      0.00         0

    accuracy                           0.17      5686
   macro avg       0.26      0.09      0.13      5686
weighted avg       0.50      0.17      0.25      5686
```



**Expectation maximization :**

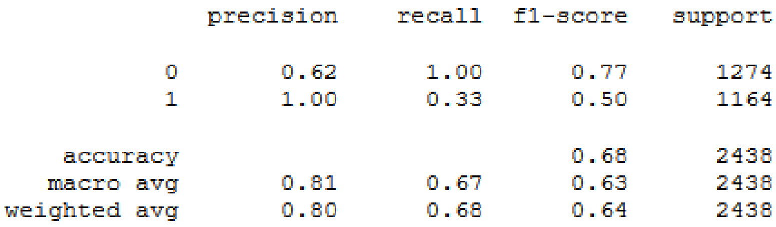
Train accuracy : 67.64

Test accuracy : 68.05



Expectation maximization performs lot better than K means getting a trian accuracy 0f 67.64 and test accuracy of 68.05.

The average precision achieved is good with the value around 0.80 which tells that it is able to clustur the dataset well.



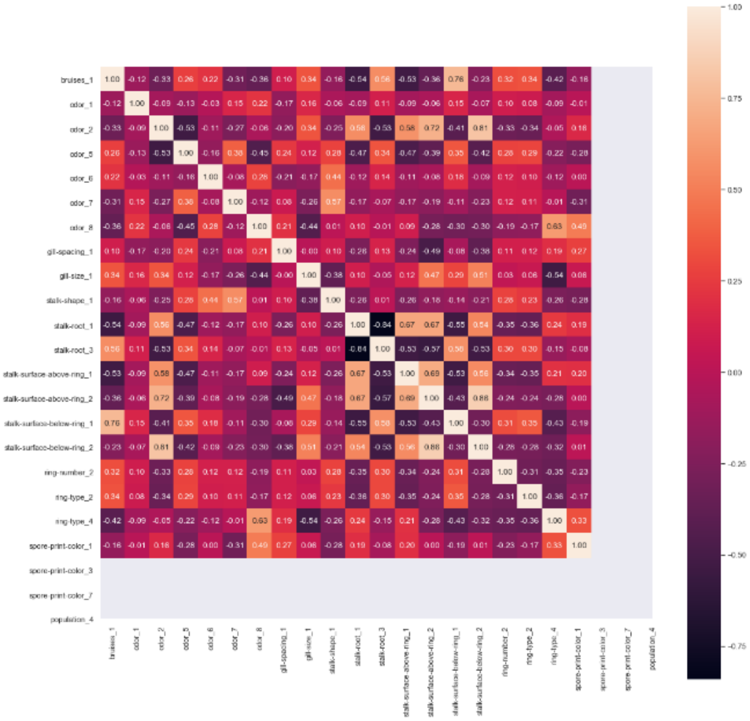
```
              precision    recall  f1-score   support

           0       0.62      1.00      0.77      1274
           1       1.00      0.33      0.50      1164

    accuracy                           0.68      2438
   macro avg       0.81      0.67      0.63      2438
weighted avg       0.80      0.68      0.64      2438
```

| Clustering algorithm | Train accuracy | Test accuracy | Precision | Recall | f1-score |
|---|---|---|---|---|---|
| K-means | 10.787 | 16.6549 | 0.5 | 0.17 | 0.25 |
| Expectation maximization | 67.64 | 68.05 | 0.8 | 0.68 | 0.64 |

**Task 2 :**

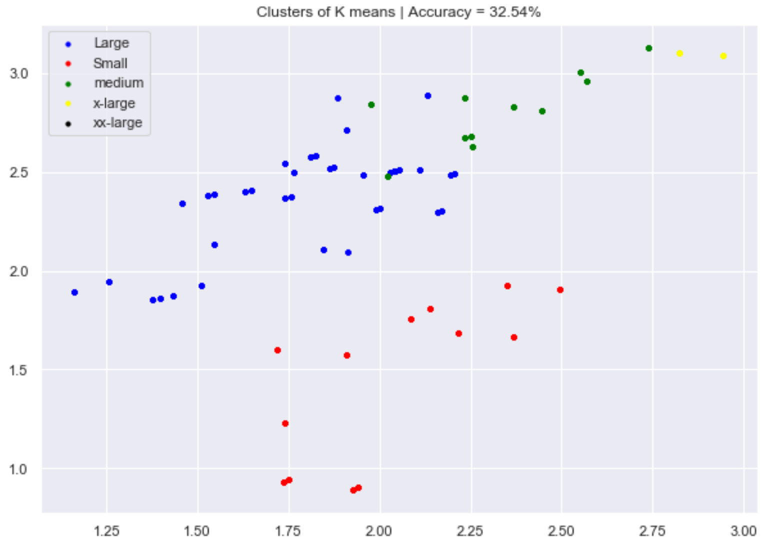
**Dimensionality reduction**

Before: there were huge numbers of negative correlation within the data. The columns had a lot of negative co-relation which was not required for the analysis.



Feature selected : 'bruises_1', 'odor_1', 'odor_2', 'odor_5', 'odor_6', 'gill-spacing_1', 'gill-size_1', 'stalk-shape_1', 'stalk-root_1', 'stalk-root_3', 'stalk-surface-above-ring_1', 'stalk-surface-above-ring_2', 'stalk-surface-below-ring_1', 'ring-type_2', 'ring-type_4', 'spore-print-color_1', 'spore-print-color_2', 'spore-print-color_3', 'spore-print-color_7', 'population_4'

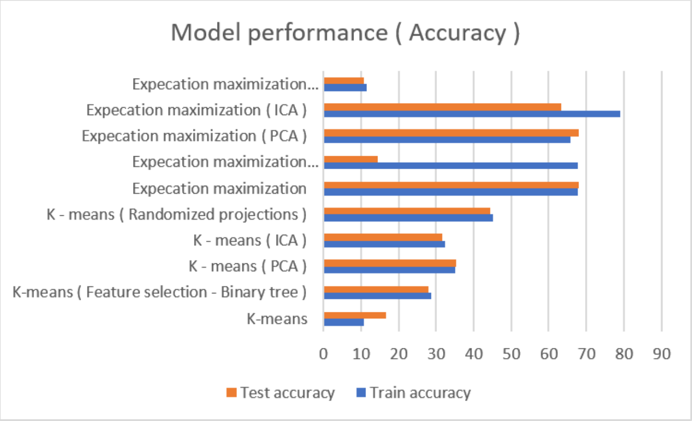
After applying feature selection we plot the scatter plot again. The plot now seems to be more spread out than before. This is mostly because the variables that were removed were less contributor to the final output and were just over crowding the plot.

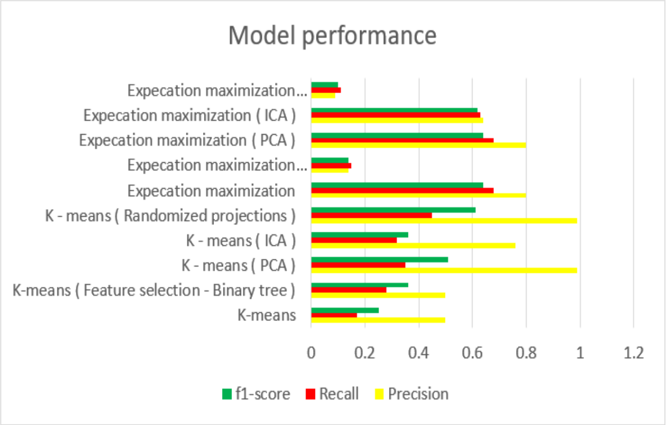


**Task 3 :**

**Feature selected using binary tree:** 'bruises_1', 'odor_1', 'odor_2', 'odor_5', 'odor_6', 'gill-spacing_1', 'gill-size_1', 'stalk-shape_1', 'stalk-root_1', 'stalk-root_3', 'stalk-surface-above-ring_1', 'stalk-surface-above-ring_2', 'stalk-surface-below-ring_1', 'ring-type_2', 'ring-type_4', 'spore-print-color_1', 'spore-print-color_2', 'spore-print-color_3', 'spore-print-color_7', 'population_4

| **Clustering algorithm** | **Train accuracy** | **Test accuracy** | **Precision** | **Recall** | **f1-score** |
|---|---|---|---|---|---|
| K-means | 10.787 | 16.6549 | 0.5 | 0.17 | 0.25 |
| K-means (Feature selection - Binary tree) | 28.7724 | 28.0557 | 0.5 | 0.28 | 0.36 |
| K - means (PCA) | 35.03 | 35.43 | 0.99 | 0.35 | 0.51 |
| K - means (ICA) | 32.48 | 31.54 | 0.76 | 0.32 | 0.36 |
| K - means (Randomized projections) | 45.28 | 44.5 | 0.99 | 0.45 | 0.61 |
| Expectation maximization | 67.64 | 68.05 | 0.8 | 0.68 | 0.64 |
| Expectation maximization<br>( Feature Selection - Binary tree) | 67.64 | 14.56 | 0.14 | 0.15 | 0.14 |
| Expectation maximization (PCA) | 65.74 | 68.05 | 0.8 | 0.68 | 0.64 |
| Expectation maximization (ICA) | 78.98 | 63.21 | 0.64 | 0.63 | 0.62 |
| Expectation maximization<br>( Randomized projections) | 11.54 | 10.83 | 0.09 | 0.11 | 0.1 |



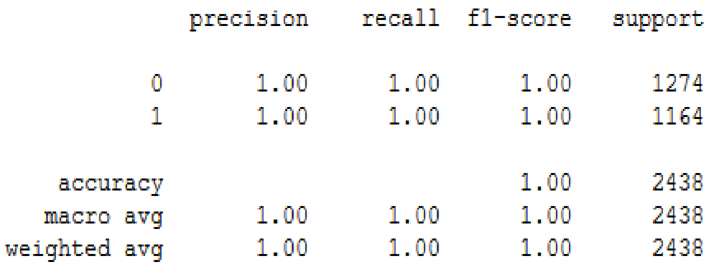


- Applying the clusturing algorithm after applying feature dimensionality reduction algorithm. Expectation maximization ( ICA ) gives the best train set accuracy.
- Model train accuracy is best for expectation maximization ( PCA ) and expectation maximization.
- K – means performance really poorly with no dimensionality reduction algorithm, but its performance changes after applying PCA, ICA, Randomized projections and feature selection (Bineary tree)
- K – means ( randomized projection ) gives the best precesion.
- Expectation maximization pricision is really low as compared to other models.
- We can observe by the above data that the clusters are formed differently for each clustering algorithm of PCA, ICA, feature selections and randomized projections. Clustering algorithm with PCA feature selection gives the best accuracy till now for this dataset.
- ICA is performing satisfactory with expectation maximization.

**Task 4:**

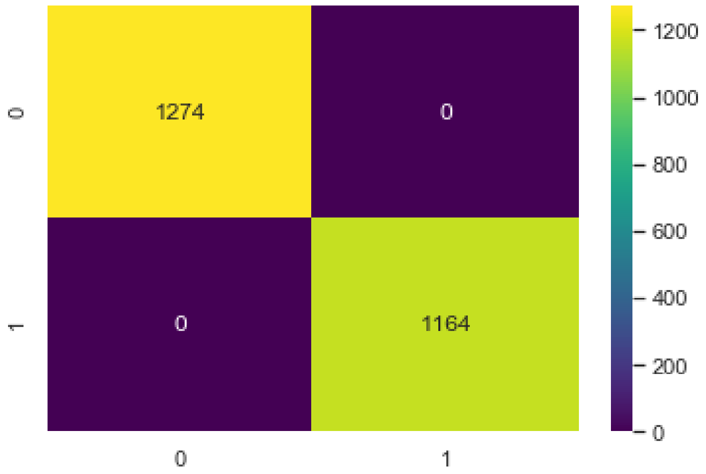
**Neural network:**

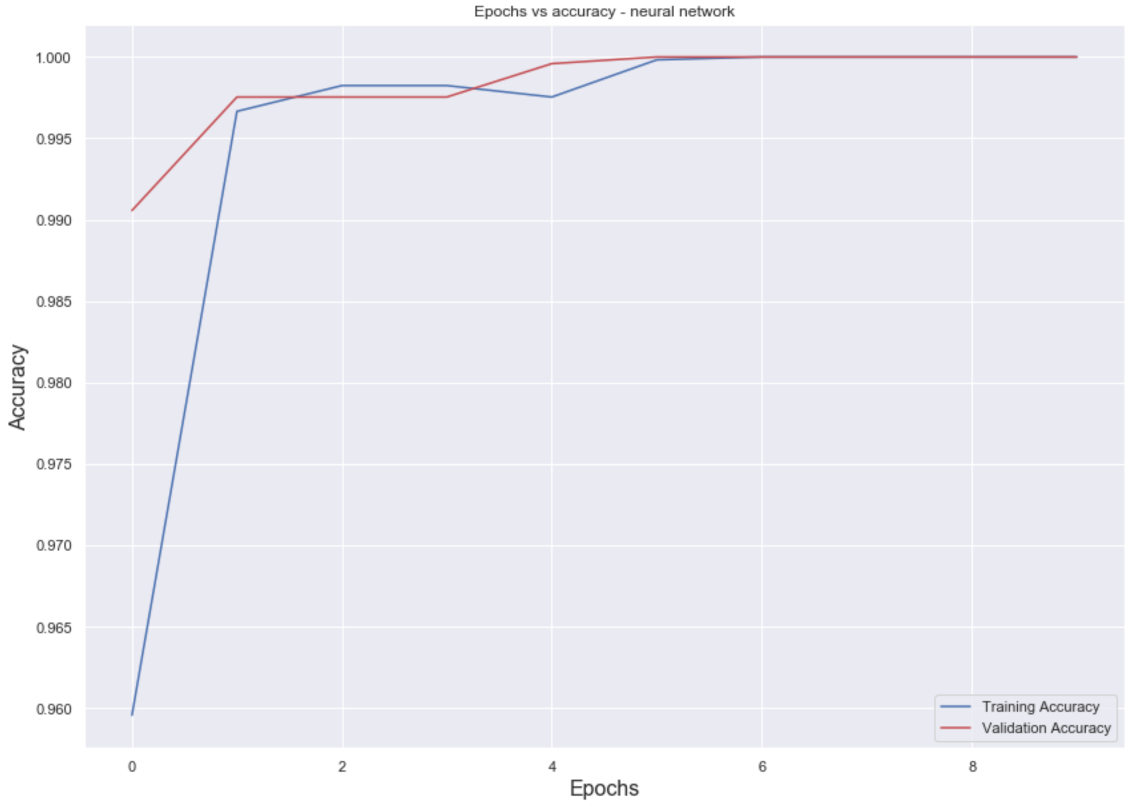
Train accuracy : 100
Test accuracy : 100



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1274
           1       1.00      1.00      1.00      1164

    accuracy                           1.00      2438
   macro avg       1.00      1.00      1.00      2438
weighted avg       1.00      1.00      1.00      2438
```

- As you can see neural network of relu, tanh, tanh, sigmoid gave the best reault in the previous assignment. But on this dataset it gives 100% accuracy which shows high bias for this model after dimension selection.
- In this scenerio neural network does not perform better than the previous model.
- We need to experiment with the number of layers and the functions again on this to get a better result.

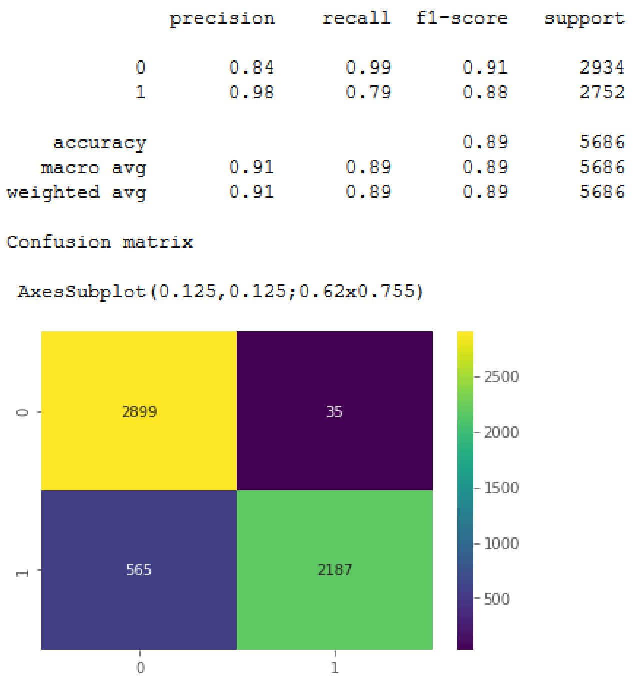




- We can see how the train and test data set's accuracy increases as the number of epochs reaches 5.
- The accuracy reaches 100% very quickly which is a concern that signals to reconfigure the model to get more realistic values.

**Task 5:**

Test accuracy: 89.44



```
              precision    recall  f1-score   support

           0       0.84      0.99      0.91      2934
           1       0.98      0.79      0.88      2752

    accuracy                           0.89      5686
   macro avg       0.91      0.89      0.89      5686
weighted avg       0.91      0.89      0.89      5686

Confusion matrix

 AxesSubplot(0.125,0.125;0.62x0.755)
```

- Considering the dataset of the 1st task as input for the neural network, the accuracy achieved for this dataset with output as its labels is really good with the value of 89.44.
- The data is working fairly better than after using the transformed domain through the model. It does not show high bias as it previously did if we ran the feature selected data through the neural network model.