## Assignment 4

Evan Jones Boddu, ejb180003@utdallas.edu

In this report, I have implemented two of the clustering algorithms, K-Means and Expectation Maximization on two different datasets. The main goal is to find the patterns using both the datasets and understand the patterns. The K-Means and Expectation Maximization algorithms are performed with the help of the libraries that are available in the python language ( on Jupyter Notebook). Keras library is used (TensorFlow Backend) for implementing Neural Network.

### Dataset 1: Appliances Energy Prediction Dataset

The dataset that has been used in this assignment is taken from the previous assignment (which has been used in assignment 1, assignment 2 as well as assignment 3) and the link to the dataset can be found here. Since the dataset is familiar, I will go ahead and show the implementations and the plots that have been performed using this dataset. The dataset is not a classification dataset, it has a continuous variable as its target variable, hence the following preliminary steps have been performed on the dataset before building and running the models:

- Converted the target variable ( "Appliances"), which is a continuous variable into a binary classification problem with values 0 and 1.
- I have put a threshold for the feature Appliances to convert it into a binary classification model. If Appliances is greater than 85, then it is considered as 1 or else it is considered as 0.
- Split the dataset into 70:30 ratio.
- Trained the dataset on different learning algorithms using different optimizers and picked the best optimizer.

Furthermore, I have selected all the features from the dataset so that the data will as raw as possible and to get the best results when the dataset has been tweaked very little. I have assigned the target to 'y' and all the other independent variables to 'x'. The dataset consists of 19735 observations and 28 features.

### Task 1: Run the Clustering Algorithms on your datasets and describe your observations (with plots):

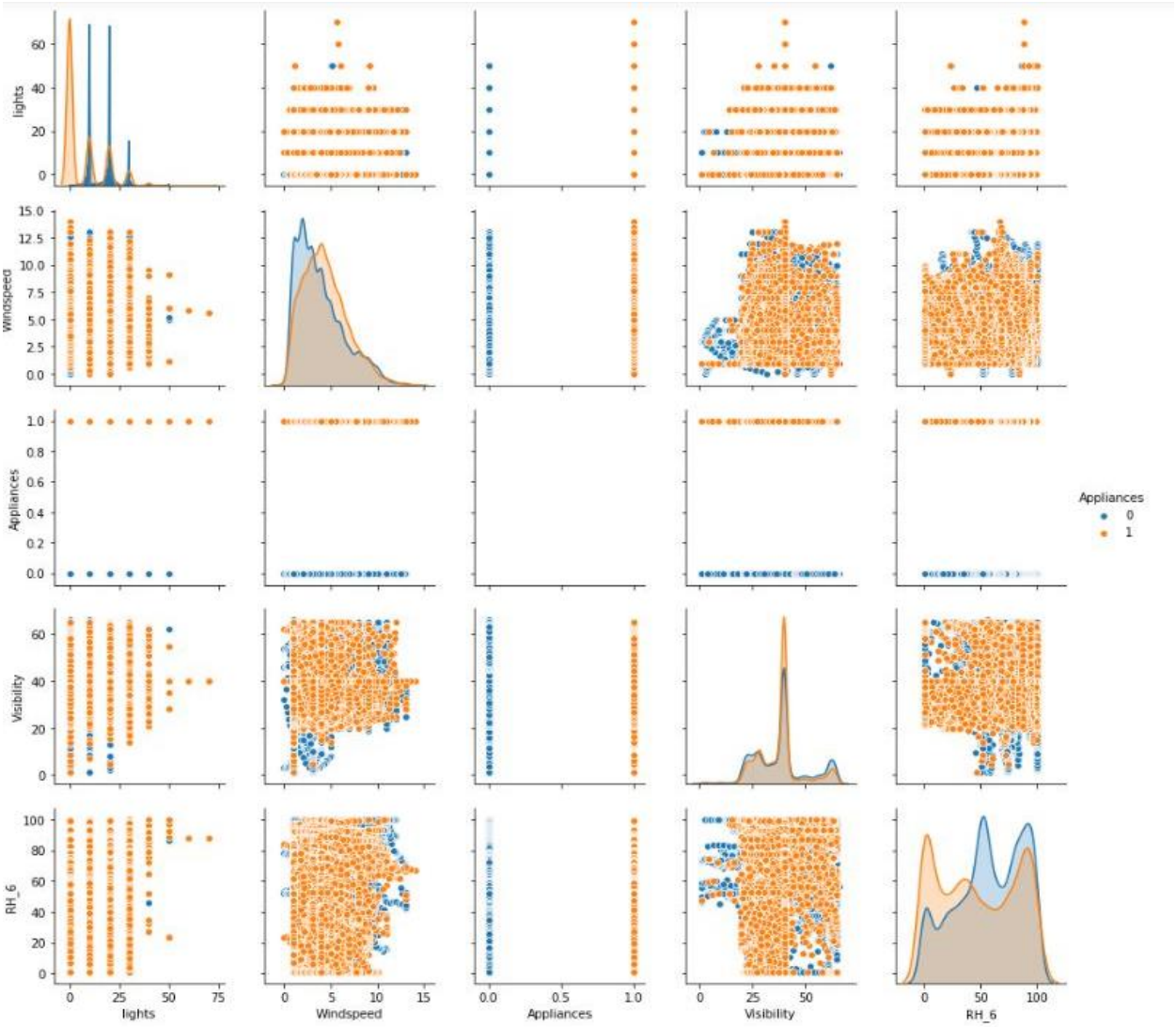
I have trained the whole data on both the algorithms which are K-Means and Expectation Maximization without using the feature selection method in this step.

### K-Means:

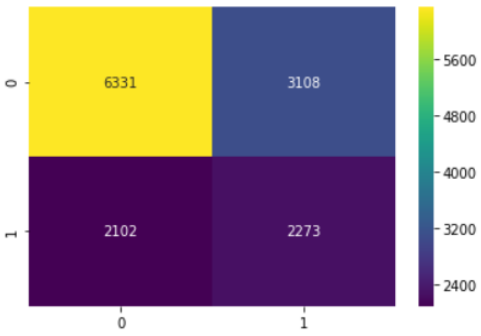
To understand the data visually, I have plotted various plots with significant features. Looking at the figure below, I am unable to observe the cluster patterns that could mean something.

The choice which is K in K-Means is taken as 2 because we have already made the output as a Binary Classification output which is 0 and 1. An alternative way to find the value of K is to train the model and to validate it for different values of K. This method is visually represented and is known as the "Elbow Plot". The figure below shows the elbow plot for the Energy Prediction Dataset. On the x-axis there are different values for K (which is the cluster value) and on the y-axis we have SSE which is a distance metrics used to calculate the errors. From the graph, we can see that for K = 6 is where the graph bends where the SSE is also less. But since we have already defined our problem into two classes, we will stick to K = 2.

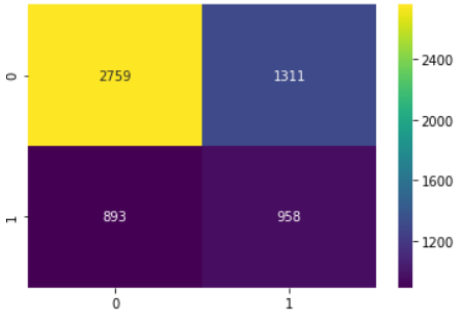
After running the model on the entire dataset with K = 2, the accuracy that is observed is 62.28% on train set and 62.77% on the test set. You can see the confusion matrix for the both in the figure below. There is no major difference between the validation accuracy and the test accuracy of the model.



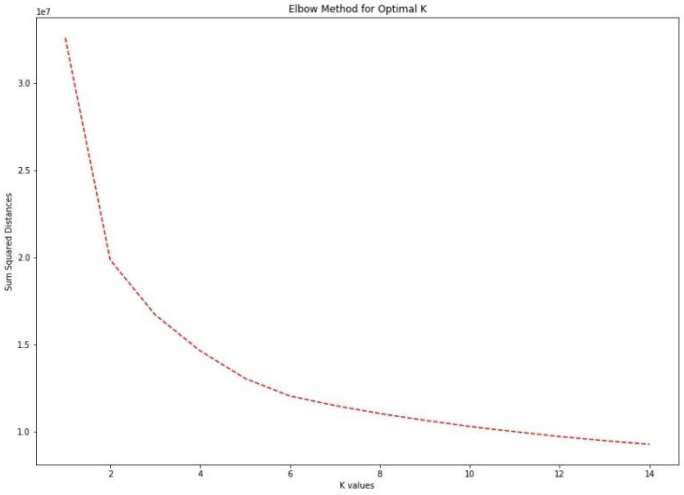

Various plots with Significant Features




Confusion Matrix for The Train Set



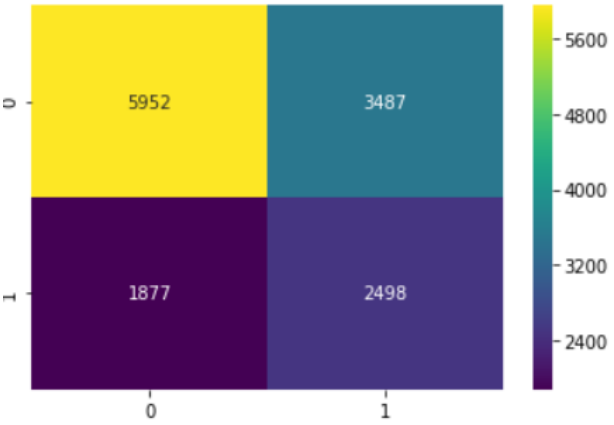

Confusion Matrix for The Test Set



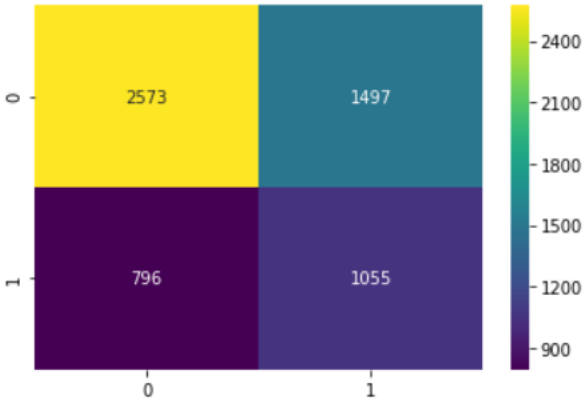

## Expectation Maximization:

After running EM on the whole dataset, we get an overall validation accuracy of 61.17% and a test accuracy of 61.27%. Expectation Maximization will perform soft clustering on all the points that are facing the tie situation and would give each node probabilities. I have used 2 gaussian curves to classify the data. The confusion matrix for that is shown below along.

Overall, the models are struggling to classify all the data points into two clusters since there are so many features which might either inflate the model or just be redundant and possibly when we reduce the features using dimensionality reduction techniques on the data set, we might see better accuracy than what we are getting here. There are 4 different types of techniques that we are targeting to use namely, Feature Selection (Decision Tree, Forward Selection, Backward Elimination), Principal Component Analysis, Independent Analysis, Random Component Analysis.




EM Confusion Matrix for Train Set




EM Confusion Matrix for Test Set

## Task 2: Apply the dimensionality reduction algorithms on your datasets and describe your observations:

| Dimension Reduction Techniques | Accuracy | | Confusion Matrix | |
|---|---|---|---|---|
| | K-Means | GMM | K-Means | GMM |
| Feature Selection | 62.41% | 50.48% | [[6357 3082] [2111 2264]] | [[4506 4933] [1908 2467]] |

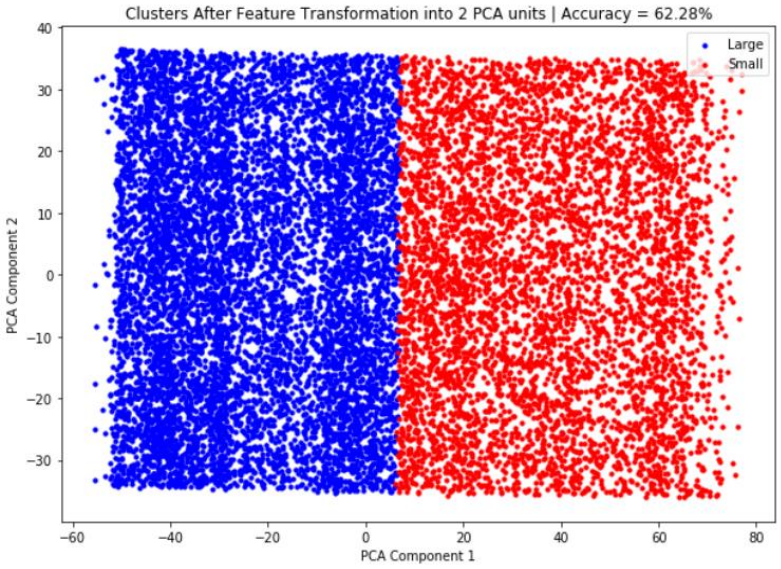
| | | | | |
|---|---|---|---|---|
| Principal Component Analysis (PCA) | 62.05% | 62.75% | [[6289 3150]<br>[2092 2283]] | [[6414 3025]<br>[2121 2254]] |
| Independent Component Analysis (ICA) | 51.93% | 49.93% | [[5410 4029]<br>[2611 1764]] | [[4676 4763]<br>[2153 2222]] |
| Random Component Analysis (RCA) | 63.80% | 59.45% | [[6731 2708]<br>[2292 2083]] | [[5863 3576]<br>[2025 2350]] |

## Task 3: Run the clustering algorithms again, this time after applying dimensionality reduction. Describe the difference compared to the previous experimentation:

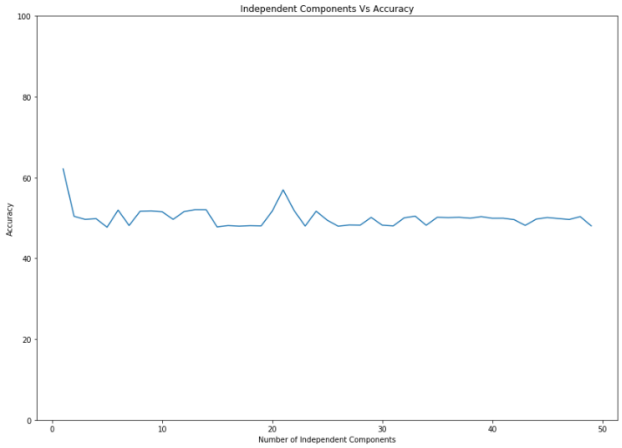
The above table shows the performance comparison with different dimensionality reduction techniques

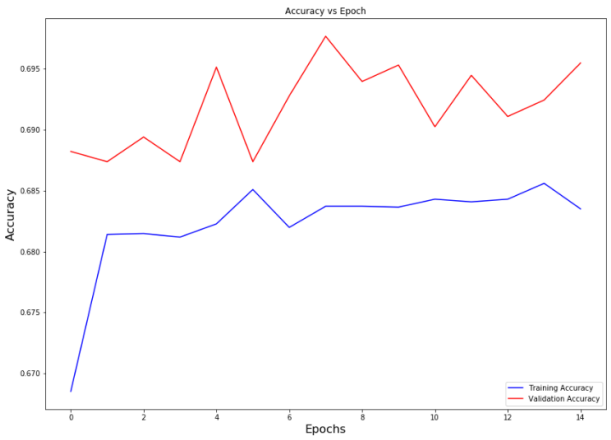
Feature Selection using Decision Tree: Decision Trees are used for feature selection as they are used for a classification problem and faster decision making in terms of computation of the data this big. The output columns given are **'lights', 'RH_1', 'T2', 'T5', 'RH_5', 'T6', 'RH_6', 'RH_7', 'T8', 'RH_8', 'RH_9', 'T_out', 'Press_mm_hg', 'RH_out'**. These features are then used to train both the clustering algorithms and measure the performance.

PCA: After applying PCA, we can see that most of the variance is achieved from the principal component 1, it amounts to the 99% of the total variance in the data. Hence, I have used the first two principal components for training models and for showing the cluster separation too. It was observed that GMM performed much better than the other in predicting the classes.




The figure to the left, shows the cluster separation that has been achieved using PCA (components =2) using the K-Means Algorithm. The plot shows a very sharp separation between the two classes. However, after computing the accuracies we see that misclassifications have happened (if you look at the table above) due to which the accuracy of the model is not correct. On the other hand, using GMM has worked very well as it performed best on the soft clustering and handled the equal probability cases.



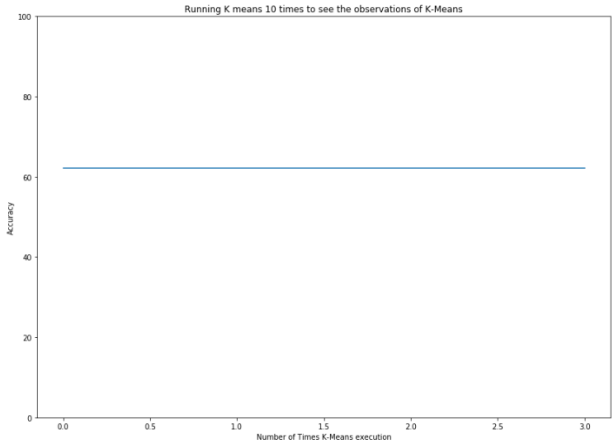



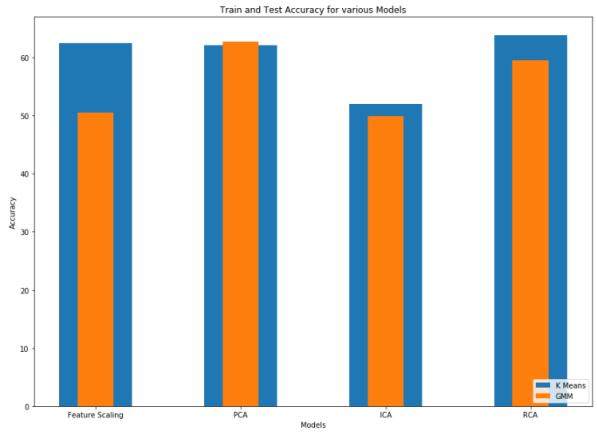

Variation of ICA with components

Accuracy vs Epoch

ICA: Similarly, as you can see in the above left figure, as we increase the number of independent components, there is some variation in the accuracy of the model. When number of the independent components is 6, the model is giving an accuracy of 51.93.

RCA: Using RCA for different number of components will not change anything about the formation of the clusters. The accuracy will always remain the same for different number of components. As you can see in the figure below.




The figure above to the right, shows the performance of different dimensionality reduction techniques on using K-Means and GMM. As we can see from the above figure, RCA gives the best results.

## Task 4: Run your Neural Network learner from assignment 3 on the data after dimensionality reduction (from task 2). Explain and plot your observations:

After performing the dimensionality reduction techniques and using PCA, the components required to describe the total variance comes down to 2. I have used the first two components for training the neural network using ReLu activation functions at each layer except input layer and output layer. Since this is a classification problem which has a binary output, I have used the sigmoid activation function. Since the dimensions were reduced to 2, it has greatly increased the speed of execution and reduced the processing speed. The figure above shows the accuracy v/s epochs plot where we can see that with every increase in

the value of epoch, the accuracy keeps getting better and better for the training set and after a drop at a certain point, the test curve also gets better. The overall accuracy achieved is 69.55% with ([[3736, 334],[1469, 382]]).

## Task 5: Use the clustering results from task 1 as the new features and apply neural network learner on this new data consisting of only clustering results as features and class label as the output. Again, Plot and explain your results:

For this, I have used the clustered groups from K-Means using entire dataset and the probabilities from Expectation Maximization algorithm along with the actual dependent variable. Together, I trained them using the Neural net. The accuracy is 68.33% with a confusion matrix of ([[9439, 0],[4375, 0]]).

## Dataset 2: Heart Disease Dataset

This dataset consists of 76 attributes, but all published experiments refer to using a subset of 14 features. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The goal field refers to the presence of heart disease in the patient. You can find the link to this dataset here.

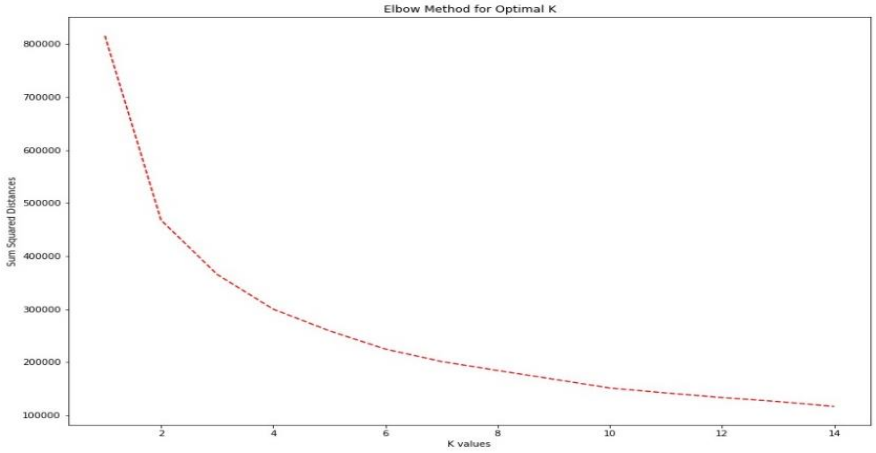
## K-Means:

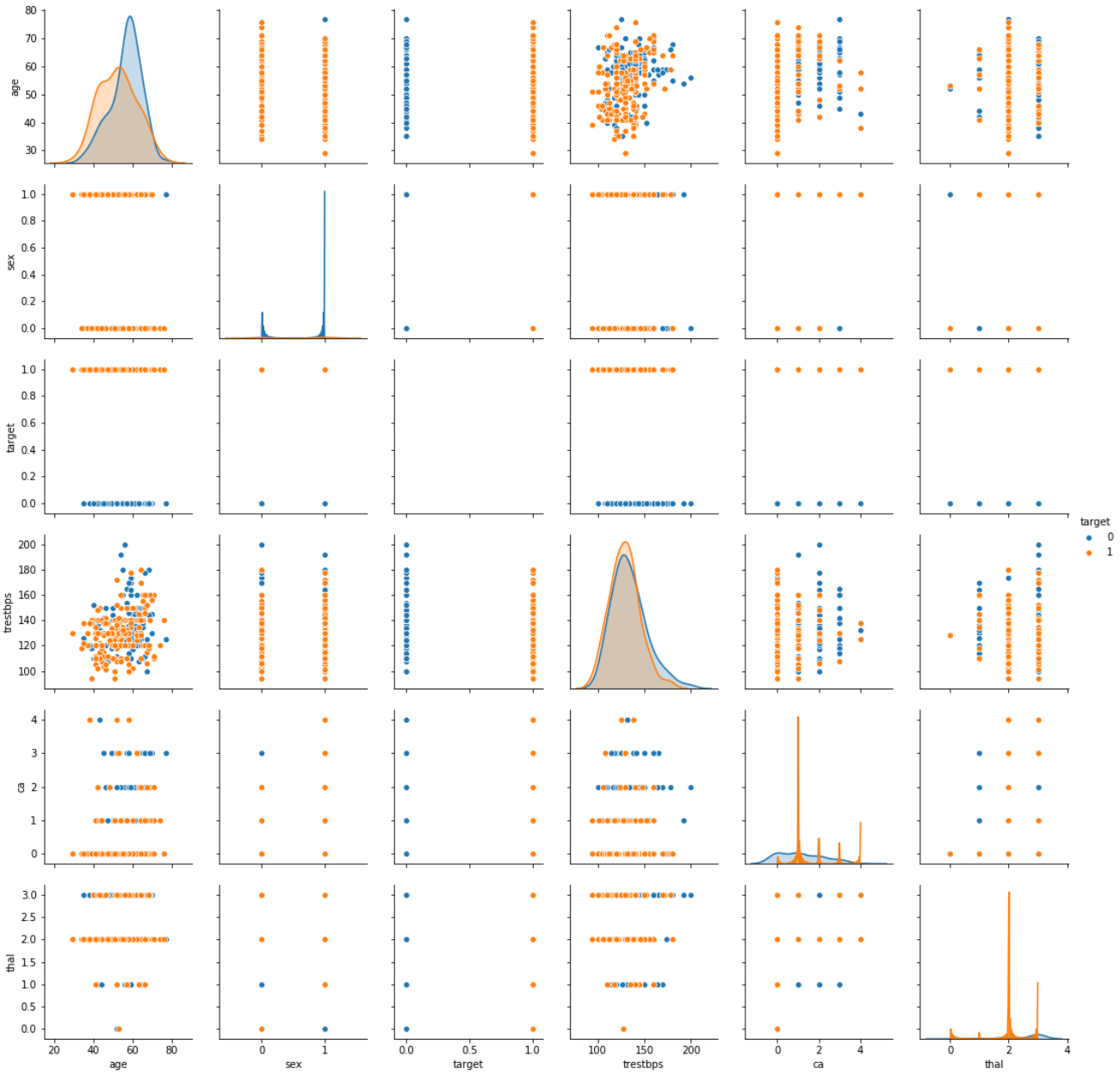
In order to understand the data visually, I have plotted a graph with significant features. From the figure below , I am not able to observe a particular pattern that shows a good separation.

The elbow chart below represents different values of L and their corresponding error plot (figure below). The error metric decreases with increase in the value of K. but the data has only two classes and hence the choice for K is 2 because the output has only two classes 0 and 1. Looking at the plot below, we may pick 10 as the value of K but then the whole purpose of classification would not serve. After running the K-Means on the whole data, we get an accuracy of 58.02% with a confusion matrix of [[46 54][35 77]].

## Expectation Maximization:

After performing E.M on the whole dataset, we get an overall validation accuracy of 64.15% with a confusion matrix of [[89 11][65 47]] and test accuracy of 54.95% with a confusion matrix of [[25 13][28 25]]. E.M will perform soft clustering on all the data points that are facing the tie situation and would give probabilities for every node.






Task 2: Apply the dimensionality reduction algorithms on your datasets and describe your observations:

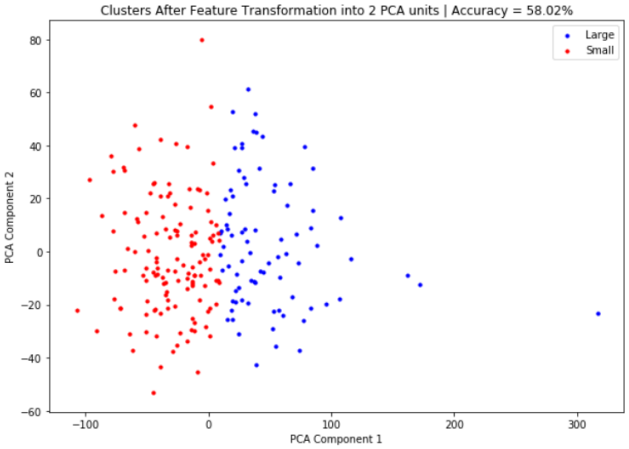
| Dimension Reduction Techniques | Accuracy | | Confusion Matrix | |
|---|---|---|---|---|
| | K-Means | GMM | K-Means | GMM |
| Feature Selection | 72.64% | 27.83% | [[66 34]<br>[24 88]] | [[46 54]<br>[99 13]] |

| | | | | |
|---|---|---|---|---|
| Principal Component Analysis (PCA) | 58.49% | 24.53% | [[47 53]<br>[35 77]] | [[56 44]<br>[94 18]] |
| Independent Component Analysis (ICA) | 79.72% | 30.84% | [[73 27]<br>[16 96]] | [[27 73]<br>[87 25]] |
| Random Component Analysis (RCA) | 61.79% | 28.30% | [[54 46]<br>[35 77]] | [[34 66]<br>[86 26]] |

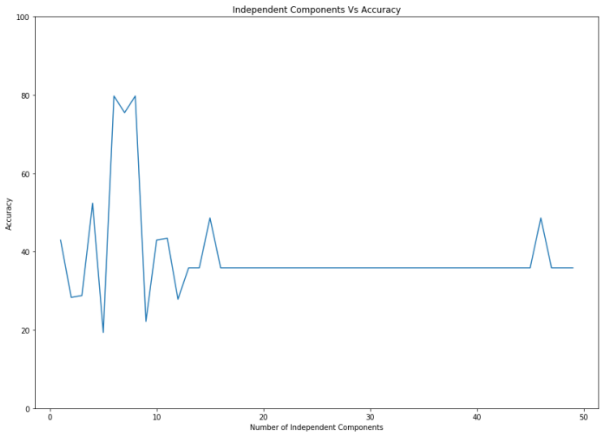
## Task 3: Run the clustering algorithms again, this time after applying dimensionality reduction. Describe the difference compared to the previous experimentation:

The above table shows the performance comparison with different dimensionality reduction techniques. Decision Trees is used as a Search algorithm for feature selection as they are used for a classification problem and fast decision making. The output features that have been selected are 'cp', 'thalach', 'exang', 'oldpeak', 'slope', 'ca'. These features are then used to train both of the clustering algorithms and measure their performances. The features obtained resulted in very low accuracy while using GMM but increased the accuracy when K-Means has been implemented.

PCA: After applying PCA, I used 2 components to cover the maximum variance. Both the clustering methods were performed on the reduced features and the performance scores are shown in the table above. The figure below shows the separation that have been achieved using the two principal components for visualization purpose.

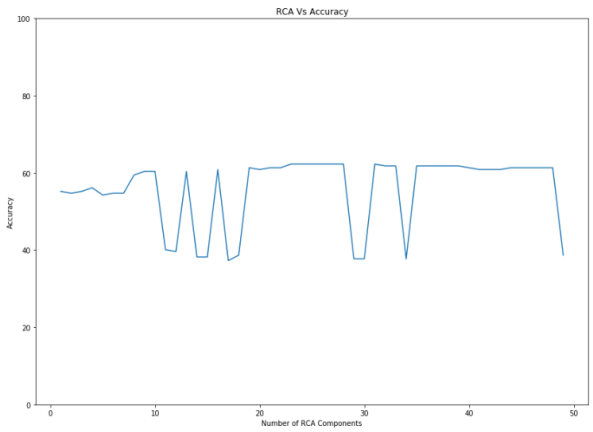



K-Means Cluster Separation

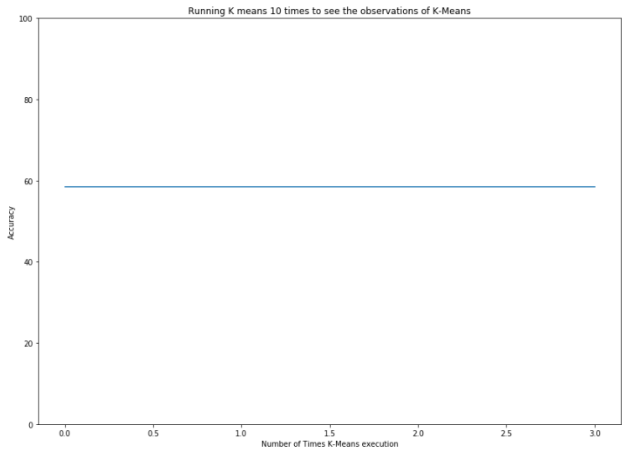



Variation of ICA with components

ICA: As you can see the figure below, as the number of components increase, there is a variation in the accuracy of the model. When the number of components is 8, the model produces 79.72% accuracy with a confusion matrix of [[73 27][16 96]].




Variation of RCA with components

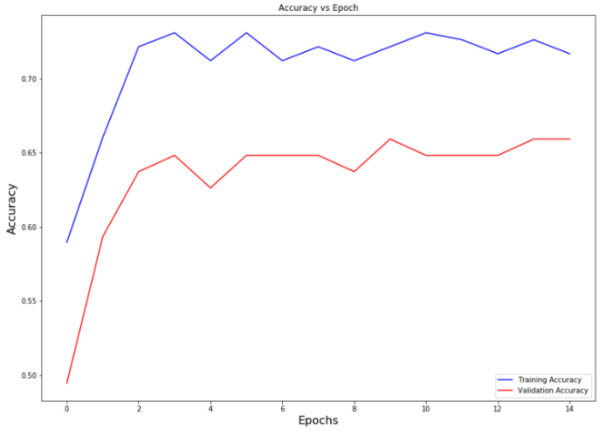



Variation of Components

The above figure shows that the accuracy of the model doesn't change for different number of components.

RCA: Using RCA for different number of components does not change the wat in which the clusters are formed but affects the accuracy of the model and can be verified from the figure above. For different components, sometimes you get high accuracy and sometimes low.

## Task 4: Run your Neural Network learner from assignment 3 on the data after dimensionality reduction (from task 2). Explain and plot your observations:




I have trained the neural networks using ReLu activation function at each layer except for the input layer and output layer. Since this a classification problem, I have used the sigmoid activation function since the dimensions were already reduced to 2, it has reduced the execution time.

The figure to the left shoes the accuracy v/s epoch plot where we can see, as the number of iterations increase, the accuracy also increases and becomes better. The overall accuracy that has been achieved is 65.93% with a confusion matrix of ([[24, 14],[17, 36]]).

## Task 5: Use the clustering results from task 1 as the new features and apply neural network learner on this new data consisting of only clustering results as features and class label as the output. Again, Plot and explain your results:

For this, I have clustered the groups from K-Means using the whole dataset and the probabilities from the E.M algorithm along with the actual target variable. I have trained them using the Neural Network and the resulting accuracy was 63.68% with a confusion matrix of ([[38, 62],[15, 97]]).

## QUESTIONS:

What type of clusters did you get? Did they line up with the class labels? If not, did they line up naturally? Were they compact or not? Why do you think you got these types of clusters? How did you pick the features generated by ICA and RP?

- After looking at the cluster separation plots, there is a proper separation of classes and maybe very few may overlap due to the shape of the cluster. The number of components or features were picked according to the plots that have been shown above which shows the variation of ICA and RP components.

When you reproduced your clustering experiments on the datasets projected onto the new spaces created by ICA, PCA and RP, did you get the same clusters as before? Different clusters? Why? Why not? Compare and contrast the different algorithms?

- The clusters remain the same shape. The shape of the cluster is not affected because I am not using any agglomerative distance metrics such as the different types of linkage.

When you re-ran your neural network algorithms were there any differences in performance? Speed? Anything at all?

- With less features, the computational time as well as power reduces a lot because it does not have to store all the data but only the features that have been used. Similarly, I reduced the data to 2 components and hence the speed of the execution has increased.