Shalini Singh- sxs180240

Project Report

Executive Summary

In this assignment we are implementing following algorithms:

a. K-Means Clustering: It is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. It is an unsupervised algorithm.

b. Expectation Maximization: It is similar to k-means but in this instead of assigning examples to clusters to maximize the differences in means for continuous variables, the EM clustering algorithm computes probabilities of cluster memberships based on one or more probability distributions.

c. We will be using various dimensionality reductions methods as well such as PCA, ICA and random projections (RCA).

Datasets

Audit data

The second dataset I have chosen is an Audit dataset from Kaggle. The Audit risk dataset consists of data of various firms and their risk factors. They belong to a multitude of sectors ranging from Irrigation, Public Health, Animal Husbandry to Fisheries, Tourism, Science and Technology. This dataset is developed by a 3rd party Audit company that wishes to calculate and assess 'Risk' by analysing the present and historical risk factors thereby facilitating the audit-planning process. The dataset has over

18 columns, with one 'target' variable – 'Risk' .

The primary reason I found this dataset interesting is because I have a strong interest in fraud detection. Initial EDA is performed and columns 'TOTAL', 'Score' and 'LOCATION_ID' are dropped because of redundancy in information and 'Money_Value' feature is imputed with the mean value as it has an unusual number of zeroes. The problem statement is to predict if a firm is under 'Risk' or not.

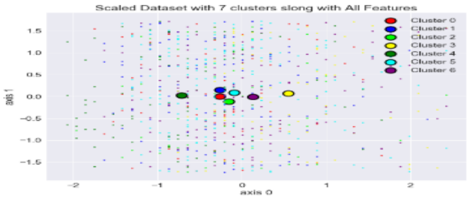
DATASET FEATURE SCALING

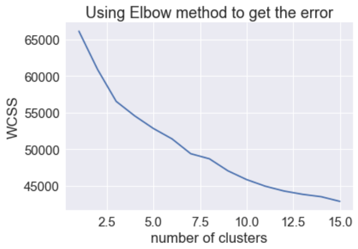
- Following the split into 70/30, I have scaled my features. Scaling the features to make sure they are all on the same scale not only helps with computation time but also allows regularization to work properly

IMPLEMENTING K-MEANS CLUSTERING

Using all the Features:

a. To select optimum k, performed a grid search between 1 and 15. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 6.
b. The clusters are not well defined as all of them are overlapping without getting proper shapes or even segregating and are not even being naturally aligned. We may need to use more random initializations to get better clusters to avoid getting stuck in local optimal values



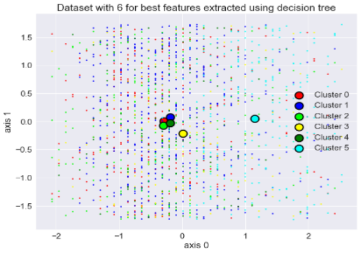


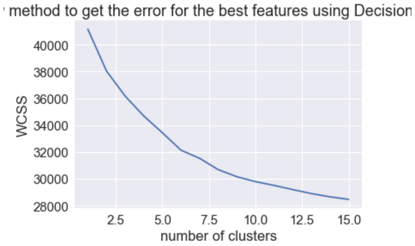
Using Important Features:

a. Used optimal decision tree from assignment 3 which had an optimal depth of 7 to help in selecting the features. We choose all features which had an importance of > 0 using entropy as the splitting criterion which resulted in reducing our features from 14 to 9

b. To select optimum k, performed a grid search between 1 and 15. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 6.

c. The cluters again line up randomly with no proper clusters. Also, the density of data points distributed is uniform throughout the graph. The centroids seem to overlap. We may need to use more random initializations to get better clusters to avoid getting stuck in local optimal values. Although the cluster value was same a in previous step the centroids line up differently due to the different conditions used.



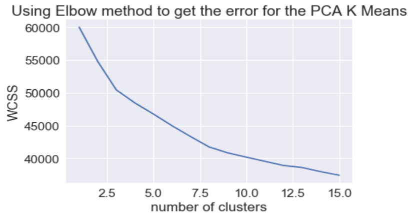


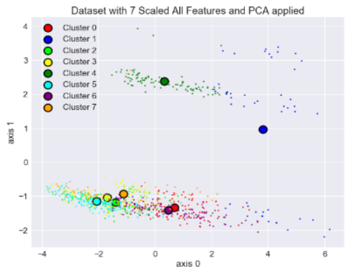
Using PCA:

Using 90% of the variance explained from PCA as the number of features. This reduces our features from 14 to 6.

a. To select optimum k, performed a grid search between 1 and 15. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 8.

b. The clusters are defined much better as compared to earlier graphs. We have cluster that look compact and have outliers as well. The data here seems to be mutually orthogonal forcing the data to be grouped into clusters as well. There is some overlap between the clusters here as well.



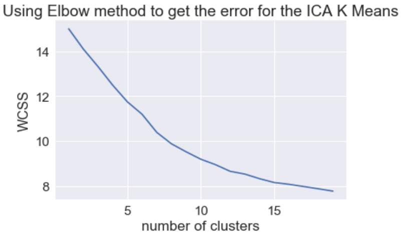


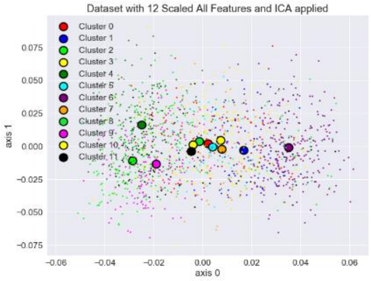
Using ICA

ICA tries to parse out the underlying signals from the data, i.e. maximizing independence by keeping mutual information as high as possible with original features and keeping the new features independent of each other.

a. To select optimum k, performed a grid search between 1 and 20. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 12.

b. Since ICA is about independence, we can see how this projects our data into rather distinct clusters. We see various groupings of data here. We here want our clusters to pick up these groupings and line up naturally, but it is not. The reasoning, these algorithms, at times, have a hard time finding these clusters and can get stuck. When this happens, it sticks multiple groups together which is happening in our case as well when it should be its own group.



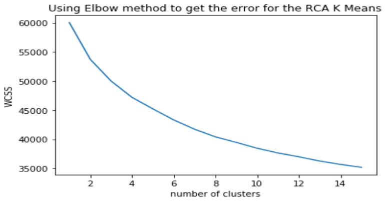


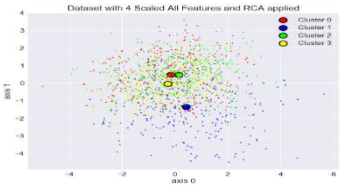
Using RCA:

Finding the correct number of components for RCA can be tough because it’s a bit arbitrary. Random projections take random directions with projecting the data onto these random directions. The benefit of this RP being able to still pick up on correlation between features along with increased speed compared to previous projections. RCA takes higher number of features as compared to ICA & PCA.

a. To select optimum k, performed a grid search between 1 and 20. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 4.

b. We can see different projections compared to the earlier times here. The 4 clusters which we selected as optimal by the elbow method show data points in clusters overlapping each other and clusters being bunched up with each other. As is mentioned in the definition of RCA data is being projected rather randomly than following any projections.

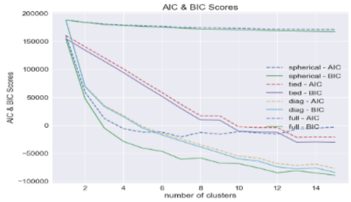


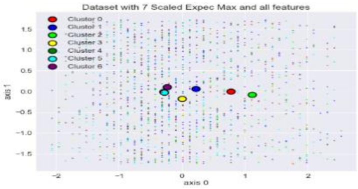


IMPLEMENTING EXPECTATION MAXIMIZATION

Using all the Features:

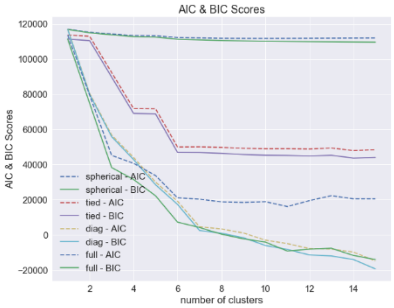
a. To select optimum k, performed a grid search between 1 and 15 over 4 different types of covariances spherical, tied, diagonal and full. We looked at the AIC & BIC values of each covariance type which are plotted against cluster number. This helps us in looking for the “elbow” with the lowest AIC/BIC score. This returned an optimal value of 7. Here we are using full covariance it allows each cluster its own general covariance matrix.
b. The clusters are not well defined as all of them are overlapping without getting proper shapes or even segregating and are not even being naturally aligned. We may need to use more random initializations to get better clusters to avoid getting stuck in local optimal values

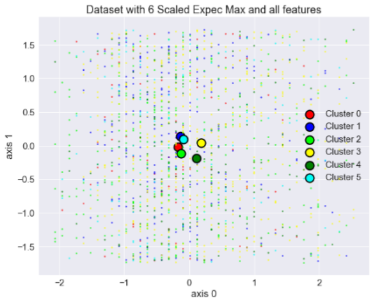




Using Important Features: Did the same feature selection as we did in K-Means Clustering

a. To select optimum k, performed a grid search between 1 and 15 over 4 different types of covariances spherical, tied, diagonal and full. We looked at the AIC & BIC values of each covariance type which are plotted against cluster number. This helps us in looking for the “elbow” with the lowest AIC/BIC score. This returned an optimal value of 6. Using full covariance here as well.
b. The clusters formed here are somewhat like what we got in k-means clustering, only here the clusters are overlapping more. The clusters again line up randomly with no proper grouping. Also, the density of data points distributed is uniform throughout the graph. We might need to use more random initializations to get better clusters to avoid getting stuck in local optimal values.

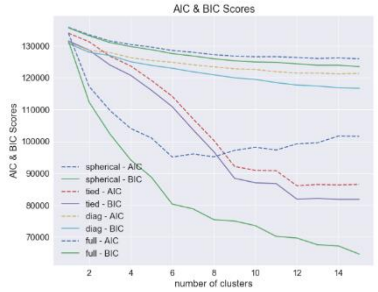


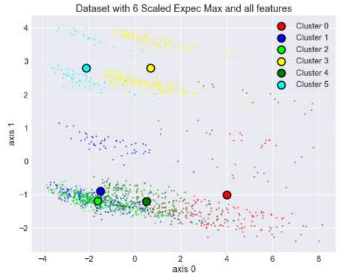


Using PCA: Using the similar dimensions as in K-means

a. To select optimum k, performed a grid search between 1 and 15 over 4 different types of covariances spherical, tied, diagonal and full. We looked at the AIC & BIC values of each covariance type which are plotted against cluster number. Again, this helps us in looking for the "elbow" with the lowest AIC/BIC score. This returned an optimal value of 6. Using full covariance here as well.

b. We are getting very different clusters as compared to previous cases of full scaled and features selected data. This could be thought because the data is being forced to be mutually orthogonal. The clusters formed here are better than the one's done through k-means as they are overlapping less, thus have good space between them. The outlier blob can be seen in the upper corners of the figure.

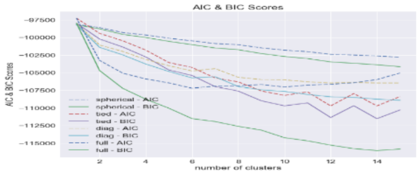


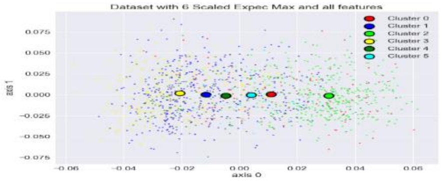


Using ICA: Using the similar dimensions as in K-means

a. To select optimum k, performed a grid search between 1 and 15 over 4 different types of covariances spherical, tied, diagonal and full. We looked at the AIC & BIC values of each covariance type which are plotted against cluster number. Again, this helps us in looking for the "elbow" with the lowest AIC/BIC score. This returned an optimal value of 6. Using full covariance here as well.

b. ICA is about independence, where it wants to projects its data into rather distinct clusters, which cannot be seen here. It looks like the centroids line up but are not able to capture perfect clusters maybe due to lack of data or it needs more random starts. We expect our clusters to pick up the groupings and line up naturally, but it is not. Given the spread of our data of our data projection are all getting stuck in the middle and near each other. In this situation, we see that expectation maximization and k-means perform similar for ICA

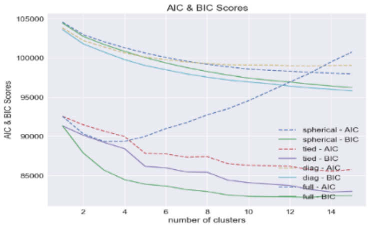


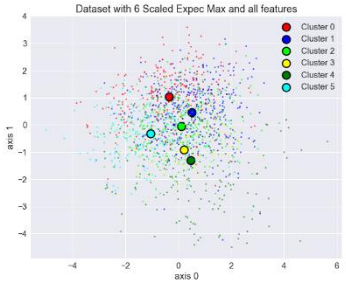


Using RCA: Using the similar dimensions as in K-means

a. To select optimum k, performed a grid search between 1 and 15 over 4 different types of covariances spherical, tied, diagonal and full. We looked at the AIC & BIC values of each covariance type which are plotted against cluster number. Again, this helps us in looking for the "elbow" with the lowest AIC/BIC score. This returned an optimal value of 6. Using full covariance here as well.

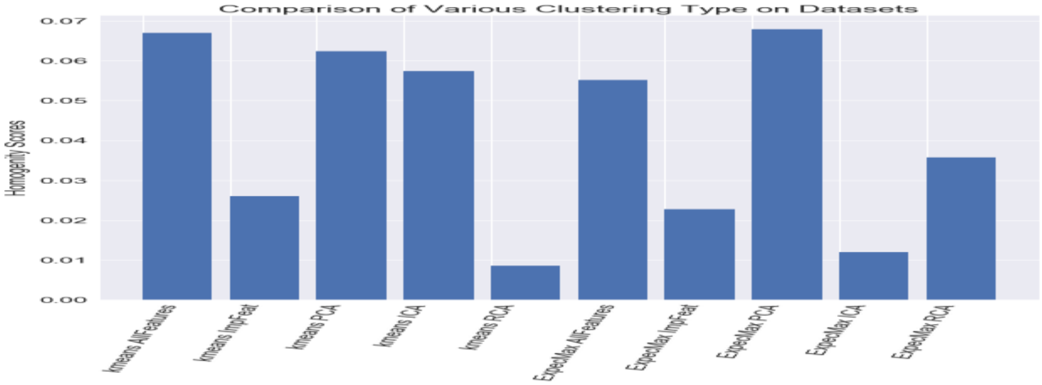
b. We can see different projections compared to the earlier times here as was the case for k-means as well. The 6 clusters which we selected as optimal by the elbow method show data points in clusters overlapping each other and clusters being bunched up with each other. As is mentioned in the definition of RCA data is being projected rather randomly than following any projections. RCA happens to catch the covariances in these random projections. The cluster are also similarly aligned in the center with not much overlap. Moreover, the clusters don't seem to be compact.





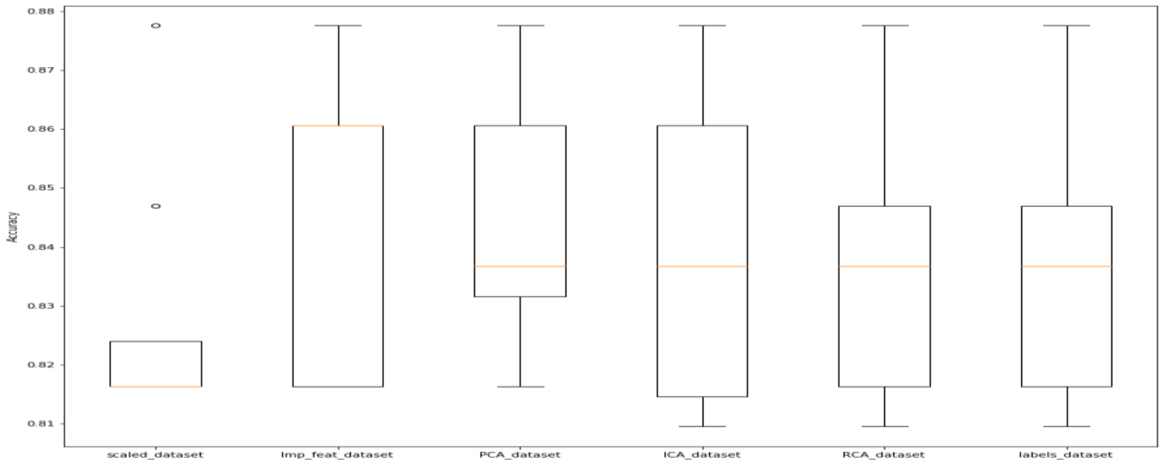
Comparing and Contrasting K-Means & Expectation Maximization

1. When we compare the graphs for the various k-means clustering algorithm, we see that the clusters formed are very different in all the algorithms. PCA, ICA and RCA all appear to cluster our data very well. It's finding the data densities and centering based on those densities. The all features and selected feature dataset is finding it hard to find these densities and hence the clusters. For PCA data is rather orthogonally projected, ICA tries to find independent components while RCA appears to be finding the random gaussians underneath but clusters appear to be overlapping each other and don't appear to be separated that well. This could be due to only plotting in two dimensions. Across the cluters we are getting different no. of clusters and cluster numbers except for first two k-means. Different projections are putting data in different projection space.
2. For Expectation Maximization also we are getting different clusters and cluster numbers. The behavior of the different situations appear to be different. Here too, PCA, ICA & RCA perform better than the other two cases. Here too the similar conditions or performances can be seen as in k-means only that it takes more time to execute than k-means.
3. One of the various methods that I could find online to compare the performances of these models is using the homogeneity score comparison. A cluster is homogenous if all class labels in the cluster are the same therefore we want our score to be as high as possible. When comparing the homogeneity score of the models the one with the best homogeneity score is Expectation Maximization with PCA followed by K-means with all the features. PCA might have discarded the unimportant features away when throwing away the low variance variables.



IMPLEMENTING NEURAL NETWORK

In the last assignment I had implemented PCA on the dataset. Therefore, the PCA dataset performance is taken as the one for last assignment. When comparing the test accuracy scores for all the datasets all of them most get similar accuracy as in the last assignment too I had implemented PCA. Here the best one comes out to be the one where we had selected features using the greedy algorithm i.e. Decision Tree to split our data based on class labels. Decision tree method selected the best features. Next the worst performing dataset is the original scaled dataset where we had not selected any important features nor projected the data in a different projection space.



Sgemm_data

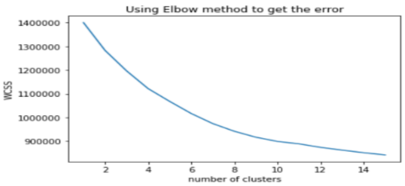
- Dataset consists of 241600 observations on 15 variables.
- The description of the variables can be found out by going to the dataset link.
- The dependent variable for the linear regression model is Avgcnt(y): Average GPU run time.
- None of the column contains any missing value, so no missing value imputation is required.

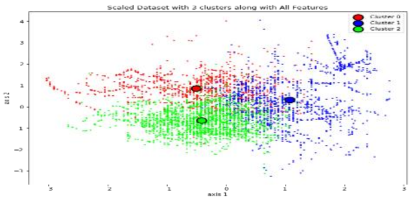
The features – 'Run 1', 'Run 2', and 'Run 3' and 'Run 4 are dropped as the average of these four parameters are calculated and used as target variable

IMPLEMENTING K-MEANS CLUSTERING

Using all the Features:

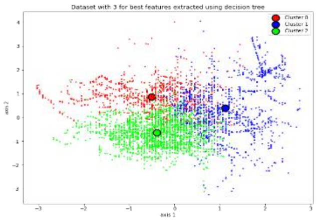
a. To select optimum k, performed a grid search between 1 and 14. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 3.
b. The clusters are well defined as distinct grouping can be seen with clusters aligned with the centroids. The density of data is more in the center with little overlapping. Here data line up naturally according to the clusters.

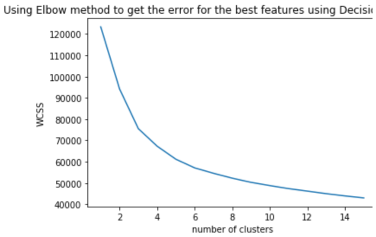




Using Important Features:

a. Used optimal decision tree from assignment 3 which had an optimal depth of 7 to help in selecting the features. We choose all features which had an importance of > 0 using entropy as the splitting criterion which resulted in reducing our features from 15 to 9
b. To select optimum k, performed a grid search between 1 and 14. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 3.
c. The clusters here gain are well defined as distinct grouping can be seen with clusters aligned with the centroids. The density of data is more in the center with little overlapping. Here data line up naturally according to the clusters. The clusters are also compact.

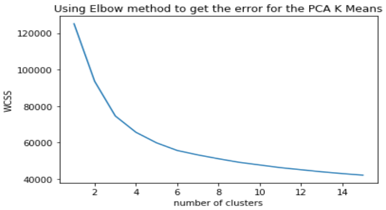


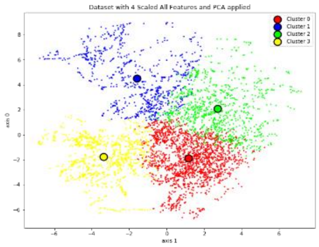


Using PCA: Using 90% of the variance explained from PCA as the number of features. This reduces our features from 14 to 12

To select optimum k, performed a grid search between 1 and 15. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 4.

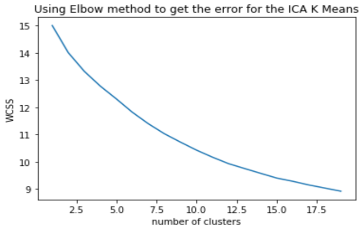
The clusters are defined much better as compared to earlier graphs as they are more compact. The data here seems to be mutually orthogonal forcing the data to be grouped into clusters as well. There is some overlap between the clusters here as well. . The density of data is more in the center with little overlapping. As the variability increases our clusters loose compactness.

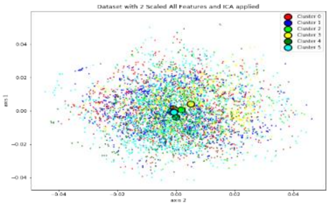




Using ICA

ICA tries to parse out the underlying signals from the data, i.e. maximizing independence by keeping mutual information as high as possible with original features and keeping the new features independent of each other.
a. To select optimum k, performed a grid search between 1 and 20. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 6.
b. Since ICA is about independence, we can see how this is trying to project our data into rather distinct clusters but fails. Therefore, we see random groupings of data here. We here want our clusters to pick up these groupings and line up naturally, but it is not. The reasoning, these algorithms, at times, have a hard time finding these clusters and can get stuck. When this happens, it sticks multiple groups together which is also not happening in our case as well when it should be its own group. Maybe random restarts should be tried even more.

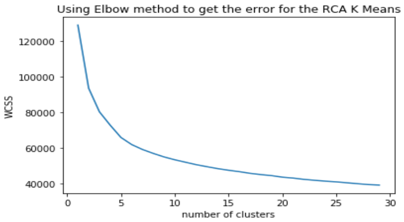


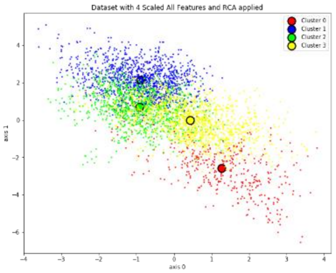


Using RCA: Finding the correct number of components for RCA can be tough because it's a bit arbitrary. Random projections take random directions with projecting the data onto these random directions. The benefit of this RP being able to still pick up on correlation between features along with increased speed compared to previous projections. RCA takes higher number of features as compared to ICA & PCA.

a. To select optimum k, performed a grid search between 1 and 20. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 4.

b. We can see different projections compared to the earlier times here. The 4 clusters which we selected as optimal by the elbow method show data points in clusters overlapping each other and clusters being bunched up with each other. As is mentioned in the definition of RCA data is being projected rather randomly than following any projections but luckily is able to catch the variances in the dataset. In RCA we are projecting data into a multivariate gaussian distribution and the clusters that we are getting appear to be Gaussian as our data projection is Gaussian due to the package used for implementing RCA.

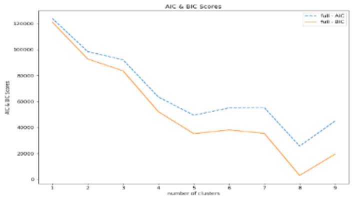


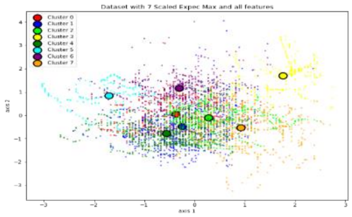


IMPLEMENTING EXPECTATION MAXIMIZATION

1. **Using all the Features:**
   a. To select optimum k, performed a grid search between 1 and 15 over just one type of covariance i.e. full as we wanted to capture all the covariances. We looked at the AIC & BIC values of each covariance type which are plotted against cluster number. This helps us in looking for the "elbow" with the lowest AIC/BIC score. This returned an optimal value of 8. Here we are using full covariance it allows each cluster its own general covariance matrix.

b. The clusters are not well defined as all of them are overlapping without getting proper shapes or even segregating and are not even being naturally aligned. We may need to use more random initializations to get better clusters to avoid getting stuck in local optimal values. Here the clusters we get are worse than the one's we got for k-means where even with smaller no. of clusters they were well segregated.



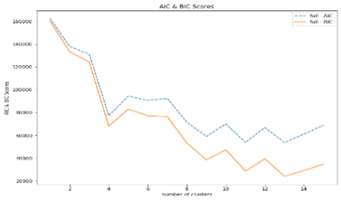


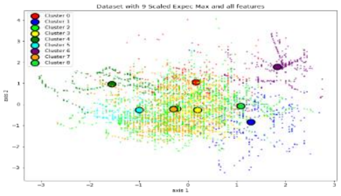
Using Important Features:

a. Used optimal decision tree from assignment 3 which had an optimal depth of 7 to help in selecting the features. We choose all features which had an importance of > 0 using entropy as the splitting criterion which resulted in reducing our features from 15 to 9

b. To select optimum k, performed a grid search between 1 and 15. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 9.

c. The clusters here again are not well defined as no distinct grouping can be seen. The density of data is more in the center with little overlapping. Some data lining up naturally can be seen. Here the clusters we get are worse than the one's we got for k-means where even with smaller no. of clusters they were well segregated.



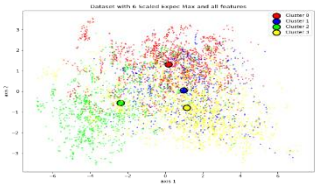


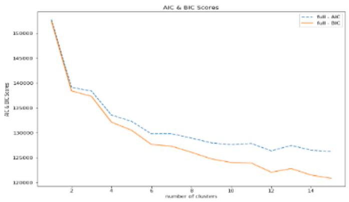
Using PCA:

Using 90% of the variance explained from PCA as the number of features. This reduces our features from 14 to 10

a. To select optimum k, performed a grid search between 1 and 15. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 6.

b. The clusters are formed here are not at all good as PCA does perform well. Maybe adding more data could help. It performs worse than K-means. Here the number of clusters is more as Expec. Maxm needed more clusters stating the fact that the data needs more clusters to be well segregated but turns out to be wrong in this case. The data appears to be somewhat orthogonal but not properly clustered.



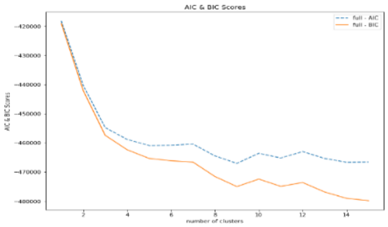


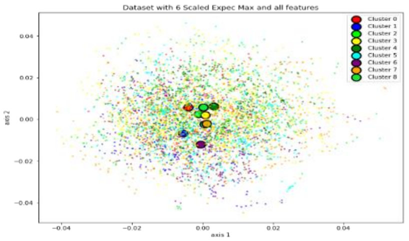
Using ICA

ICA tries to parse out the underlying signals from the data, i.e. maximizing independence by keeping mutual information as high as possible with original features and keeping the new features independent of each other.

a. To select optimum k, performed a grid search between 1 and 20. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 9.

b. Since ICA is about independence, we can see how this is trying to project our data into rather distinct clusters but fails. Therefore, we see random groupings of data here. Here too ICA performs badly on the dataset. Centroids are all stuck in the center with density of the data being maximum there, but still ICA is unable to get independent components as Independent components are not always present.

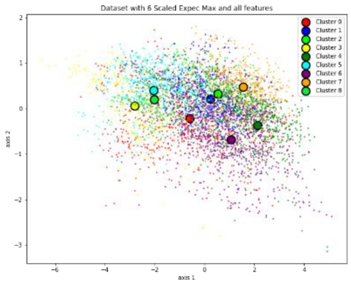


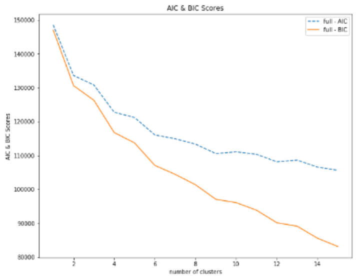


Using RCA: Finding the correct number of components for RCA can be tough because it's a bit arbitrary. Random projections take random directions with projecting the data onto these random directions. The benefit of this RP being able to still pick up on correlation between features along with increased speed compared to previous projections. RCA takes higher number of features as compared to ICA & PCA.

a. To select optimum k, performed a grid search between 1 and 20. We looked at the sum of squared error within the clusters(WCSS) using the elbow method and plotted that against cluster number. This returned an optimal value of 9.

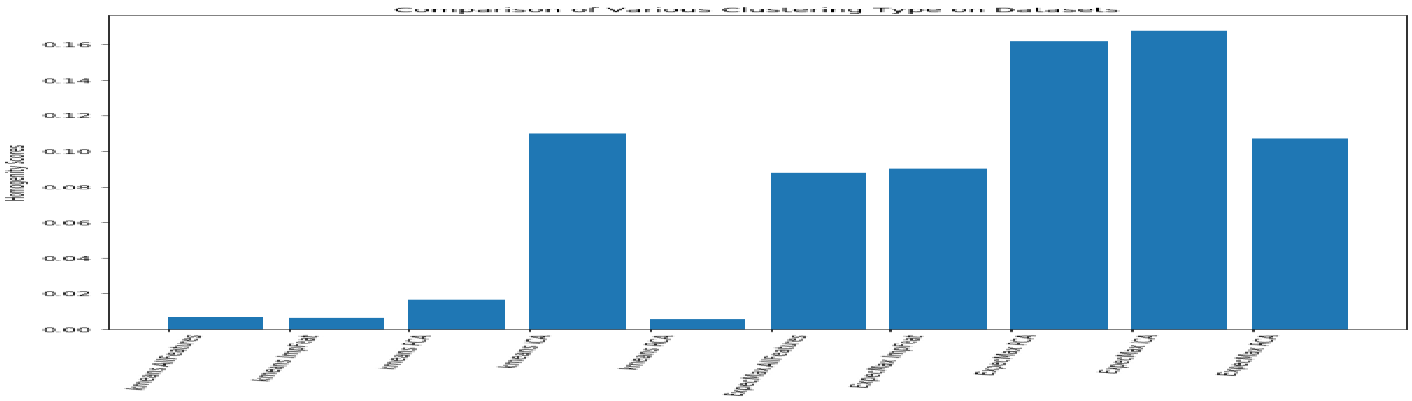
b. We can see different projections compared to the earlier times here. The 9 clusters which we selected as optimal by the elbow method show data points in clusters overlapping each other and clusters being bunched up with each other. As is mentioned in the definition of RCA data is being projected rather randomly than following any projections but unluckily here RCA is unable to give good clusters unlike in the case of k-means. In RCA we are projecting data into a multivariate gaussian distribution and the clusters that we are getting appear to be Gaussian as our data projection is Gaussian due to the package used for implementing RCA.





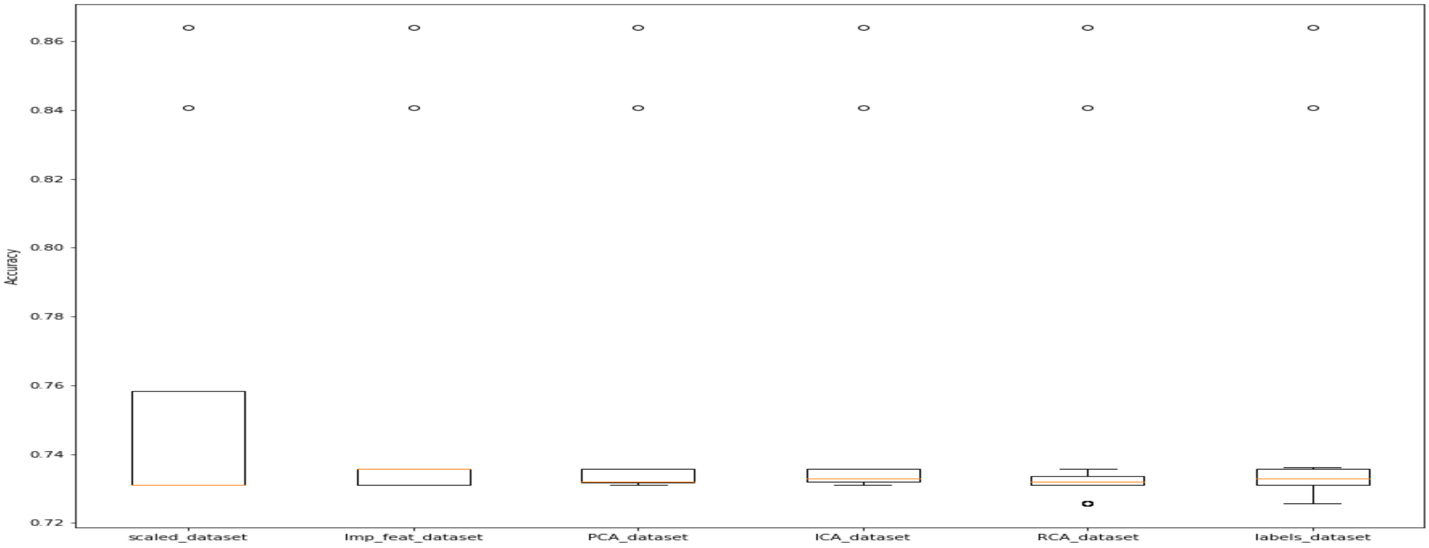
Comparing and Contrasting K-Means & Expectation Maximization

1. When we compare the graphs for the various k-means clustering algorithm, we see that the clusters formed are very different in all the algorithms. PCA, ICA and RCA all appear to cluster our data well. It's finding the data densities and centering based on those densities. The all features and selected feature dataset also group data well. For PCA data is rather orthogonally projected, ICA tries to find independent components while RCA appears to be finding the random gaussians underneath but clusters appear to be overlapping each other and don't appear to be separated that well. This could be due to only plotting in two dimensions. Across the projections we are getting different no. of clusters and cluster numbers except for first two k-means. Different projections are putting data in different projection space. Even with simple projections are dataset is performing fairly.
2. One of the various methods that I could find online to compare the performances of these models is using the homogeneity score comparison. A cluster is homogenous if all class labels in the cluster are the same therefore we want our score to be as high as possible. When comparing the homogeneity score of the models the one with the best homogeneity score is Expectation Maximization with ICA followed by is Expectation Maximization with ICA which is surprising but maybe we had missed the homogenous factor earlier. PCA might have discarded the unimportant features away when throwing away the low variance variables. ICA would have been able to find Independent Components here.



IMPLEMENTING NEURAL NETWORK

In the last assignment I had implemented PCA on the dataset. Therefore, the PCA dataset performance is taken as the one for last assignment. When comparing the test accuracy scores for all the datasets all of them most get similar accuracy as in the last assignment. Here the best one comes out to be the one where we had selected features using the greedy algorithm i.e. Decision Tree to split our data based on class labels. Decision tree method selected the best features. It is followed by ICA dataset which could be because of the reason that ICA would have been able to find independent components. Next the worst performing dataset is the original scaled dataset where we had not selected any important features nor projected the data in a different projection space.


Accuracy
scaled_dataset
Imp_feat_dataset
PCA_dataset
ICA_dataset
RCA_dataset
labels_dataset