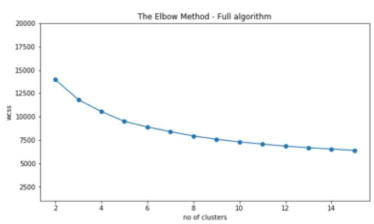
To implement the K-Means and Expectation maximization algorithms in the following datasets.

- Dataset 1 : Power consumption of a house : Build a classification based on various environmental and household factors such as pressure, temperature and so on.
- Dataset 2 : Counter strike game : Classify data based on various counter strike games played.

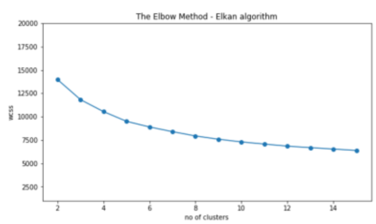
K – Means Clustering : It is an example of unsupervised classification techniques where K-clusters are picked, and each center claims its closest center and we pick the cluster according to elbow rule.

1. *complete dataset* : Running the cluster on the dataset yields the following Within-Cluster-Sum-of-Squares(WCSS) plotted on y-axis and number of clusters on x-axis.

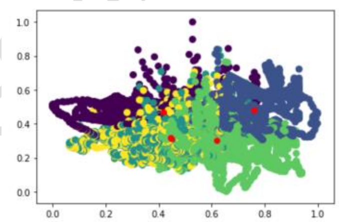
For dataset 1, after the cluster size of 5, there seems to be not much decrease in wcss and so we can say that cluster of 5 is good. The full and Elkan algorithm seems to make no difference in WCSS as shown in 2 graphs and so we can use either of them. (A &B). A sample plot is shown between 2 columns – RH1 and TH1 columns Clustering in (C) and 5 red points indicate centroids for all 5 different colored clusters. The clusters are close packed.



A

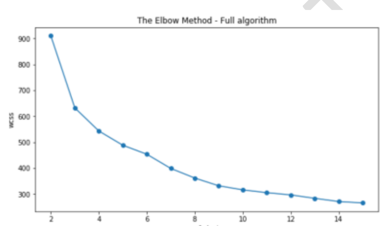


B

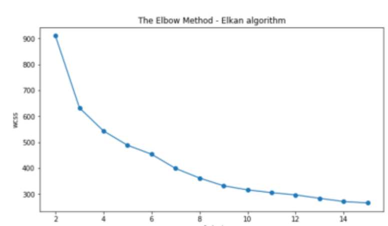


C

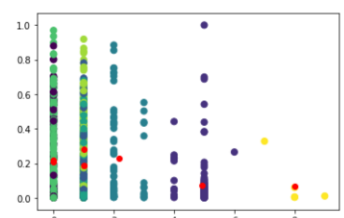
For dataset 2, after the cluster size of 8 there seems to be not much decrease in wcss and so we can have 8 as our cluster size and determine our model. Full and elkan seems to make no difference and so we can use any of them to build our cluster model. (A&B). A sample plot is shown between 2 columns – map and wait time columns Clustering in (C) and 8 red points indicate centroids for all 8 different colored clusters. The clusters are far packed.



A



B



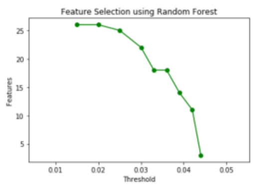
C

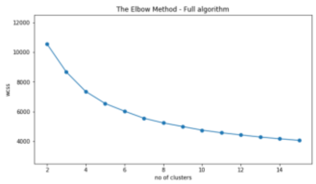
2&3

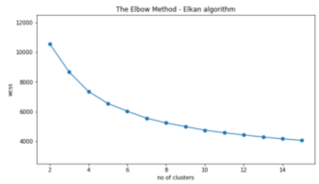
Feature selection :

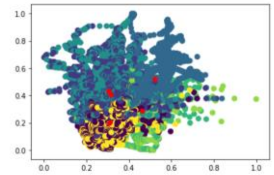
Dataset 1: I have used Random forest to build feature selection for the thresholds : 0.015, 0.02, 0.025, 0.030 ,0.033,0.036,0.039,0.042,0.044 and 100 estimators.

For dataset 1, the threshold of 0.033 is giving us 15 features and so I have kept the threshold to be 0.033 to estimate the model. (A) The clusters are close packed.







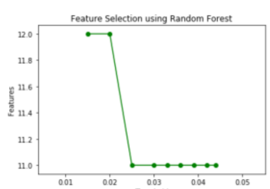


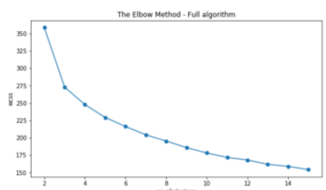
A **B** **C** **D**

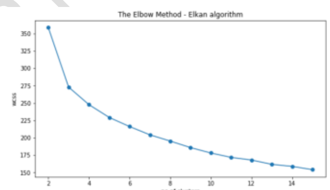
These features are used to run our model : RH_1, T2, RH_2, RH_3, RH_4, RH_5, T6, RH_6, RH_7,T8, RH_8, RH_9, T_out, Press_mm_hg, RH_out, Tdewpoint, rv1, rv2. By elbow method, we can run up to 7 clusters this time and build our clusters and full, elkan doesn't make any difference. (B&C). The sample 7 clusters are represented in D with centroids of each cluster in red color for the first 2 columns.

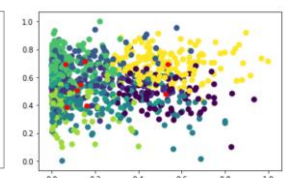
For dataset 2, I have used Random forest to build feature selection for the thresholds : 0.015, 0.02, 0.025, 0.030 ,0.033,0.036,0.039,0.042,0.044 and 100 estimators. There is a constant number of features that are getting returned after 0.025 threshold and so we can have that to choose the features for feature selection using 0.025. (A).

Features I used : 'Wait Time(s)', ' Match Time(s)', 'Team A Rounds', 'Team B Rounds', 'Ping', 'Kills', 'Assists', 'Deaths', "Mvp's", "HS%", 'Points'







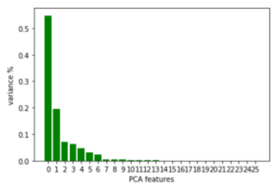


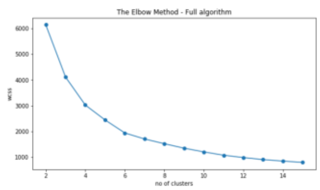
A **B** **C** **D**

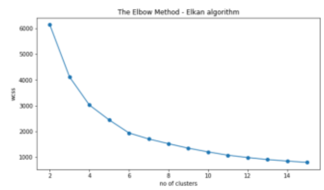
The clusters and WCSS are plotted on B&C, it is observed that it is same for both algorithms and after the elbow of 8 clusters, we observe decrease in wcss and so we stop with 8. The corresponding sample cluster diagram is plotted in D where each color represents a cluster having center as red color respectively for first 2 columns. The clusters are close packed.

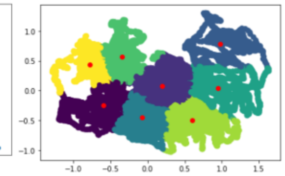
PCA :

For the dataset 1, Feature transformation is done using PCA to build a model which has maximum variance. There is a sharp drop in variation after 1st and 2nd feature – all variables were given as inputs. I have chosen 2 features since after that it becomes less than 0.1 (A). The clusters can be 8 and both algorithms makes no much difference after 8(B&C). Clustering with red points are centroid is plotted for those features. (D) The clusters are far packed.







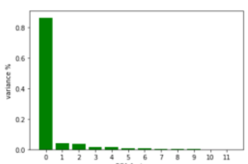


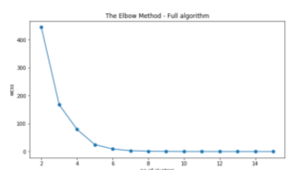
A B C D

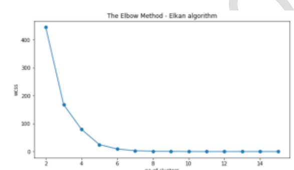
For the dataset 2, more than 80% of the variation is explained by the first component and so we can have 1&2 component for building our model as after that it becomes less.

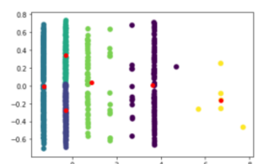
After using this to build the elbow graph, the 2 algorithms – full and elkan are not having any difference and it is seen that with 6 clusters we can achieve good results since after that there is no much improvement in WCSS. (B&C)

The clusters are build for the the cluster size 6 and each cluster is plotted in each color with centeroids as red color. (In diagram D) The clusters are far packed.



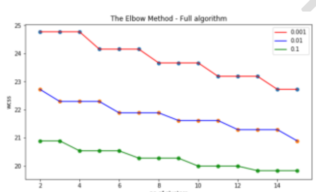


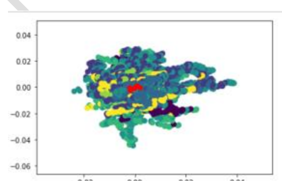




A B C D

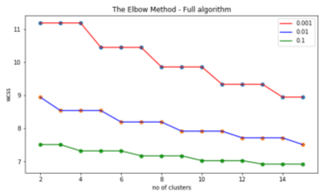
ICA : Feature transformation using ICA involves clustering by mutual independence. I have used all components to build ICA with K-Means are built based on this transformed set. I have used full algorithm and have implemented tolerance 0.001, 0.01 and 0.1 to see WCSS changes in each case. WCSS is less for 0.1 tolerance and C represents the 7 clusters and red point is centroid for each of the clusters. The clusters are close packed.

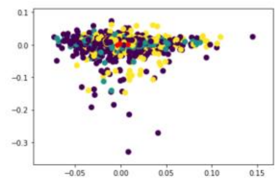




A B

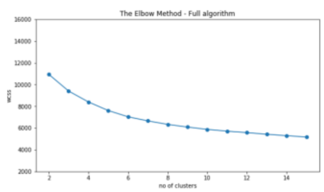
In the second dataset, I have used all components to build ICA with K-Means are built based on this transformed set. I have used full algorithm and have implemented tolerance 0.001, 0.01 and 0.1 to see WCSS changes in each case. WCSS is less for 0.1 tolerance and C represents the 3 clusters and red point is centroid for each of the clusters. The clusters are close packed.

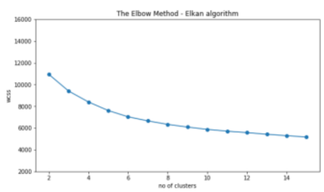


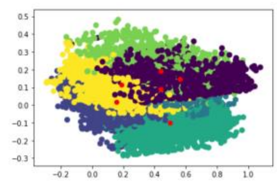


A B

Randomized projection : For the first dataset, all components are used to build randomized projection on K-Means and it is observed that clusters after 6 doesn't make much sense and so we can stop with 6 (A&B). The 6 clusters are plotted with centroid in red color (C). The clusters are far packed.



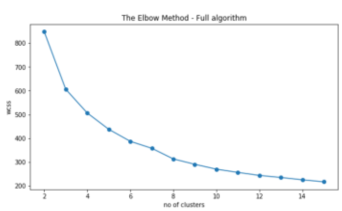


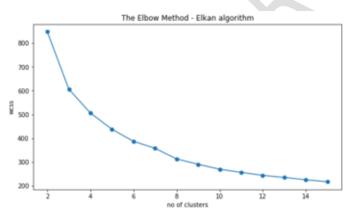


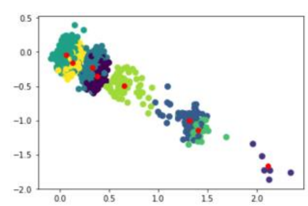
A B C

For the second dataset, all components are used in to build the model. After applying the model to the K-means it is seen that we can stop by 8 clusters because there is no much difference after that (A&B).

The cluster is plotted for 8 and it is represented in the diagram C below. The clusters are far packed.





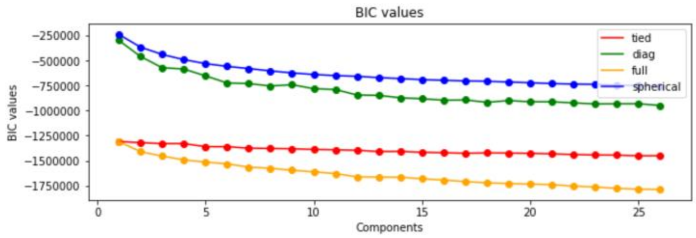


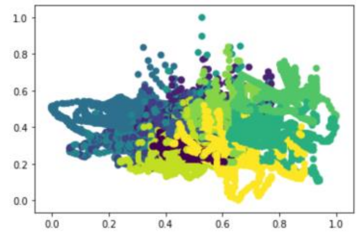
A B C

Expectation Maximization : This is an unsupervised classification technique where maximum likelihood is used in constructing clusters. Experiments are carried out for various components : tied, diag , full, spherical. The one with lowest BIC is used to build our model.

We obtained the data selected and transformed model by PCA, ICA, Selection and randomized projects; the same dataset is used for building expectation maximization models.

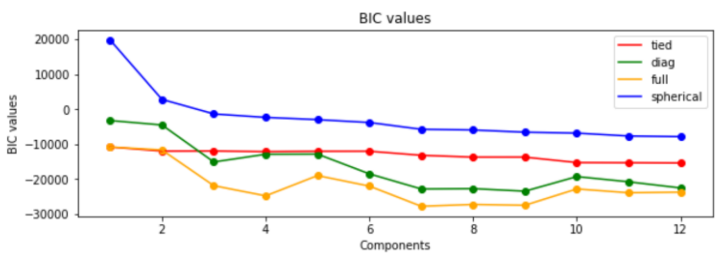
Complete dataset : For dataset 1, Running the cluster on the dataset yields the following Within-Cluster-Sum-of-Squares(WCSS) plotted on y-axis and number of clusters on x-axis, BIC is minimum for full covariance and components are making much difference after one point and so I did 12 components. (A) The clusters are plotted in B with each cluster in each color. The clusters are far packed.

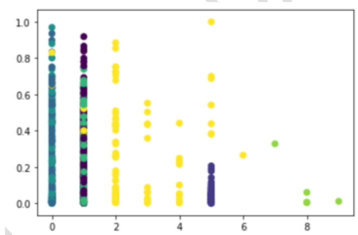




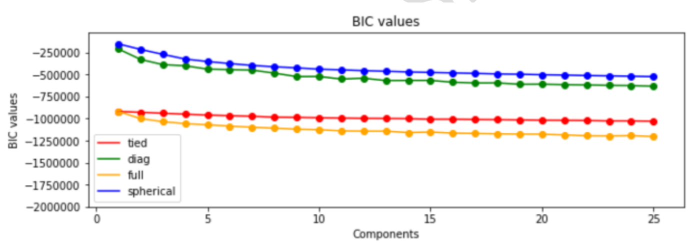
A **B**

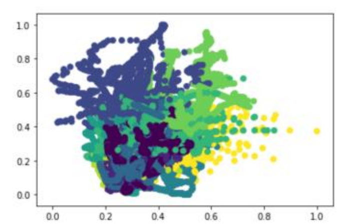
For dataset 2, Running the cluster on the dataset yields the following Within-Cluster-Sum-of-Squares(WCSS) plotted on y-axis and number of clusters on x-axis. BIC is minimum for full covariance and components are making much difference after one point and so I did 7 components. (A) The clusters are plotted in B with each cluster in each color.(B) The clusters are far packed.





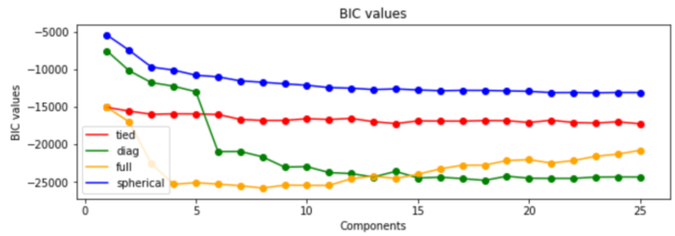
Feature selection : For the first dataset, I obtained the features already - RH_1, T2, RH_2, RH_3, RH_4, RH_5, T6, RH_6, RH_7,T8, RH_8, RH_9, T_out, Press_mm_hg, RH_out, Tdewpoint, rv1, rv2. Full covariance is giving us the lowest bcc and we can have 10 components since it doesn't make much difference after that (A). The clusters are close packed. The various clusters below are represented using various colors. Using this in building expectation maximization (B) for the first 2 columns:-

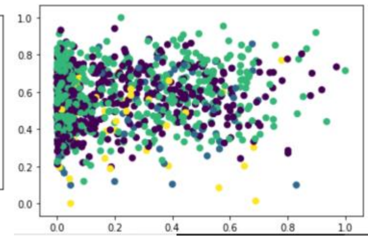




A **B**

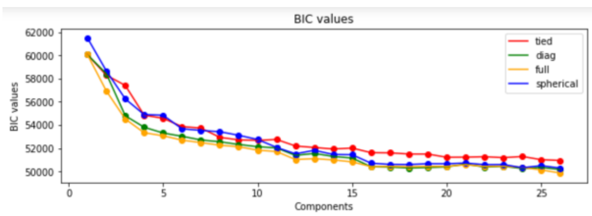
For the second dataset, we obtained the features 'Wait Time(s)', ' Match Time(s)', 'Team A Rounds', 'Team B Rounds', 'Ping', 'Kills', 'Assists', 'Deaths', "Mvp's", "HS%", 'Points'. The lowest BIC is obtained with 4 components and clusters are built from it as shown in diagram B for the first 2 columns. The clusters are close packed.

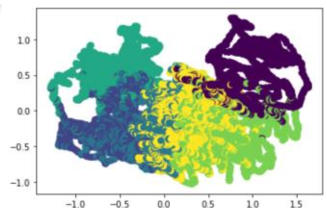




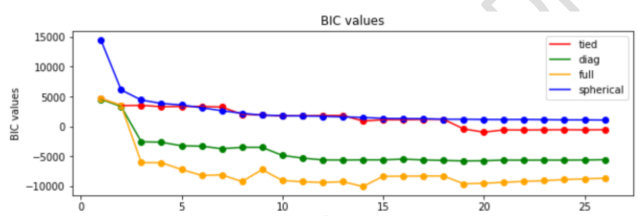
A **B**

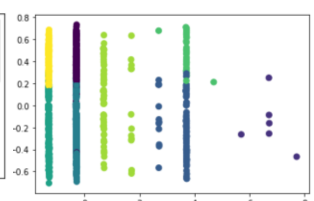
PCA : For dataset 1, We obtained that 2 components of transformed model gives us the maximum variation. So, building the model based on it for expectation maximization. Full covariance with 10 components is better since after that it becomes linear i.e. not much change. The corresponding clustering is plotted, and it is displayed below with each color representing a cluster. The clusters are far packed.



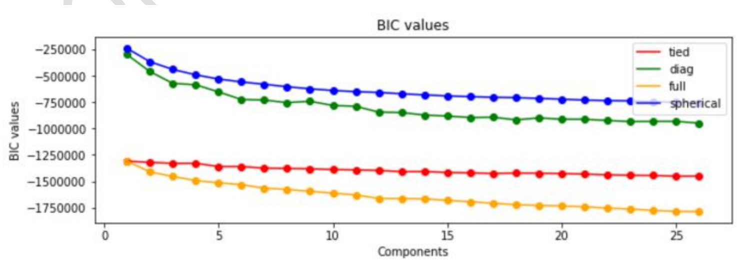


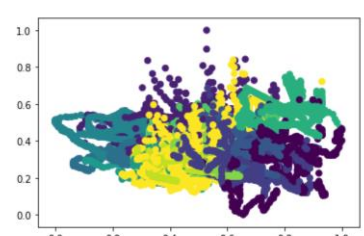
For dataset 2, I obtained that 6 components of transformed model gave us the maximum variation. So, building the model based on it for expectation maximization. Full covariance with 8 components is better since after that it becomes linear i.e. not much change. The corresponding clustering is plotted, and it is displayed below with each color representing a cluster. The clusters are far packed.



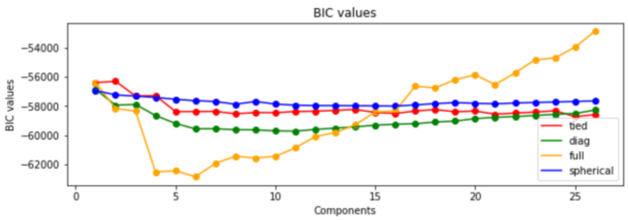


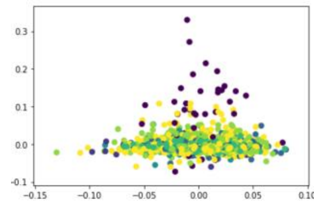
ICA: The 26 components of ICA is applied over the randomized projections to see which component for Gaussian can be applied to obtain minimum BIC value. After 12 components for full covariance there is not much change and so we can stop with 12 (LHS diagram). The final cluster obtained is shown in RHS diagram below. The clusters are close packed.





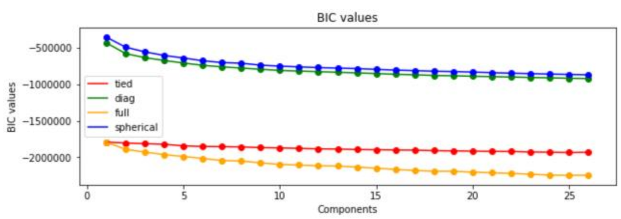
After 6 components for full covariance there is not much change and so we can stop with 6 (LHS diagram). The final cluster obtained is shown in RHS diagram below with each cluster in a color. The clusters are close packed.

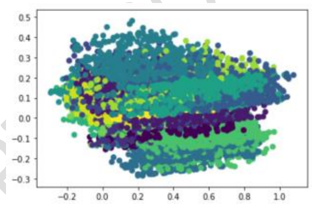




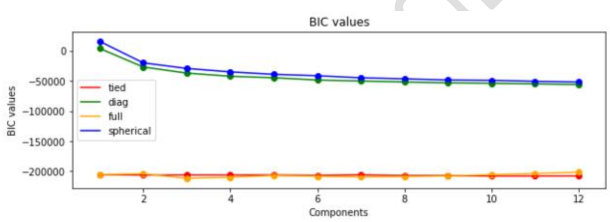
Randomized projects: All Randomized projections components are applied in the dataset to build Gaussian projection. For Gaussian mixture, experiment is done to determine the components required.

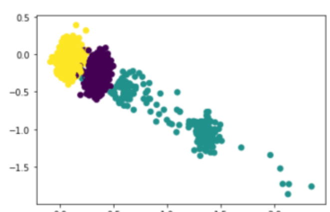
Dataset 1 : For Full covariance with 15 components would be better to build model as it has the minimum BIC. After 15 components there is not much change and so we can stop with 15. The 12 clustered expectation maximization is built, and it is shown in the diagram RHS and each color represents each cluster. The clusters are close packed.





Dataset 2: The BIC values for 12 components are seen and it is seen that full covariance with 3 components are giving better output. With 3 components, we are building the Expectation maximization clustering and the it is plotted below in B. The clusters are far packed.





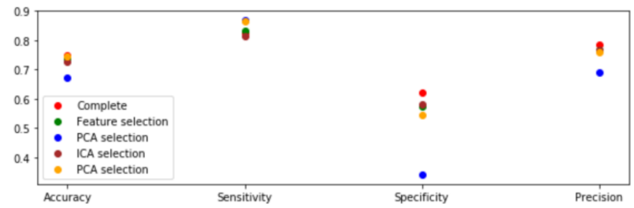
A **B**

4 & 5 *Neural networks:* Based on assignment 3, it was observed for this dataset 2 hidden layers having (5,3) nodes on each of these layers respectively and 1000 iterations provided us the best accuracy. These results will be applied on complete dataset, after feature selection and data obtained after 3 of the feature transformation algorithms.

<u>Dataset 1 : Power consumption</u>

Predicted -> Actual	**0**	**1**
0	TN	FP

1	FN	TP

Dataset / Algorithm	Run Time - seconds	Confusion matrix	Accuracy	Sensitivity	Specificity	Precision
Complete	13.36	[[1376 835] [657 3053]]				
Feature selection	12.23	[[1270 941] [623 3087]]				
PCA	1.12	[[754 1457] [494 3216]]				
ICA	2.76	[[1284 927] [685 3025]]				
Randomized Projections	8.39	[[1200 1011] [504 3206]]				

It has been seen that the accuracy, specificity and precision is better when the entire dataset is run. The time taken to run each of the algorithms is given and it is seen that it is not same. PCA is getting us the least run time and complete it taking too long to run.

4(i) Complete dataset : The first row in the table above represents the run time, confusion matrix, sensitivity and specificity and precision of neural networks applied on complete dataset and it has taken 13.36 seconds to run.

4 (ii) PCA: Variation is becoming less than 0.05 after 3 components and so we will apply this modified model to construct neural networks. The model has taken 1.12 seconds to run and the results are tabulated above.

4 (iii) ICA : All 26 components are fed into the model and the data obtained is used to build neural net. The model has taken 2.76 seconds to run and the results are tabulated above.

4 (iv) Randomized projections : It is an example of unsupervised classification techniques where the maximum likelihood of various input variables will be found and clustered accordingly. The model has taken 8.39 seconds to run and the results are tabulated above.

5 (i) Feature selection : These features are used to build the model RH_1, T2, RH_2, RH_3, RH_4, RH_5, T6, RH_6, RH_7,T8, RH_8, RH_9, T_out, Press_mm_hg, RH_out, Tdewpoint, rv1, rv2. The model has taken 12.23 seconds to run and the results are tabulated above.

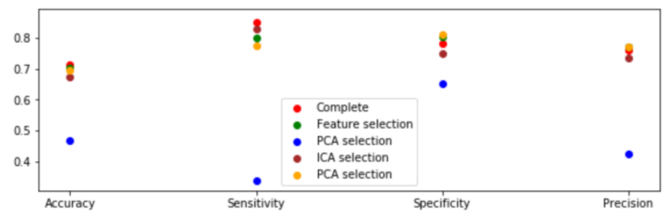
<u>Dataset 2 : Counter strike</u>

4 & 5 *Neural networks:* Based on assignment 3, it was observed for this dataset 2 hidden layers having (4,3) nodes on each of these layers respectively and 3000 iterations provided us the best accuracy. These results will be applied on complete dataset, after feature selection and data obtained after 3 of the feature transformation algorithms.

The confusion matrix is in this format. The tie is ignored from the prediction. :-

Predicted -> Actual	**0**	**1**	**2**

0	TN	FP	-
1	FN	TP	-
2	-	-	-

Dataset / Algorithm	**Run Time - seconds**	**Confusion matrix**	**Accuracy – Sensitivity – Specificity - Precision**
Complete	4.15	[[129 36 12] [20 113 6] [7 0 17]]	
Feature selection	2.44	[[131 32 14] [27 108 4] [8 2 14]]	
PCA	0.50	[[114 61 2] [89 45 5] [3 16 5]]	
ICA	2.63	[[119 40 18] [23 110 6] [2 3 19]]	
Randomized Projections	2.56	[[132 31 14] [30 104 5] [8 1 15]]	

It has been seen that the accuracy, sensitivity is better when the entire dataset is run. The time taken to run each of the algorithms is given and it is seen that it is not same. PCA is getting us the least run time and complete it taking too long to run.

4(i) Complete dataset : The first row in the table above represents the run time, confusion matrix, sensitivity, specificity and precision of neural networks applied on complete dataset with 13.36 runtime.

4 (ii) PCA: Variation is becoming less after 3 components and so we will apply this modified model of 2 components to construct neural networks. The model has taken 0.50 seconds to run.

4 (iii) ICA : All 12 components are fed into the model and the data obtained is used to build neural net. The model has taken 2.63 seconds to run and the results are tabulated above.

4 (iv) Randomized projections : It is an example of unsupervised classification techniques where the maximum likelihood of various input variables will be found and clustered accordingly. The model has taken 2.56 seconds to run and the results are tabulated above.

5 (i) Feature selection : After using the features obtained from previous experiment, the model has taken 2.44 seconds to run and the results are tabulated above.

Task 5 : The K-means outputs are fed as inputs to neural networks for both datasets with class label as output.

<u>Dataset 1: K-Means</u>

```
[[ 458 1753]
 [ 508 3202]]
Accuracy 0.6181388279006924
Sensitivity 0.8630727762803234
Specificity 0.20714608774310267
Precision 0.6462159434914228
```

<u>Dataset 1: Expectation maximization</u>

```
[[ 899 1312]
 [ 651 3059]]
Accuracy 0.6684681641614593
Sensitivity 0.8245283018867925
Specificity 0.40660334690185435
Precision 0.6998398535804163
```

Dataset 2: K-Means

```
[[134  43   0]
 [106  33   0]
 [ 10  14   0]]
Accuracy 0.5284810126582279
Sensitivity 0.23741007194244604
Specificity 0.7570621468926554
Precision 0.4342105263157895
```

Dataset 2: Expectation maximization

```
[[151  26   0]
 [114  25   0]
 [  7   8   9]]
Accuracy 0.5569620253164557
Sensitivity 0.17985611510791366
Specificity 0.8531073446327684
Precision 0.49019607843137253
```

Accuracy, Specificity, Sensitivity and precision is less when K-means, Expectation maximization outputs are fed as input in both datasets.

Findings

The k was chosen after performing a lot of experiments and within cluster distance was used in choosing the optimal cluster using elbow rule. The experiments done for 2 datasets resulted in 2 different cluster optimal sizes. The centroid method was used throughout this where center would be picked up first and cluster would be built based on that. The clusters packing was given for each clusters and some clusters were close packed, and others were far. After the feature transformation and feature selections were done, the clustering was well separated in most cases than which was done in complete dataset.

The features were picked by tolerance experiment in ICA and Randomized projections. When the clusters were built from the dataset which were feature transformed and feature selected, clustering was different than those when done as a complete dataset for both cases. This is because when the features are selected or transformed, the variables that contribute much to the prediction based on variation explanation were only chosen or modify and so based on this, clustering changes.

When these datasets were applied for the neural networks with the hidden layers chosen on previous assignment, the run time of neural networks varied for each of those. In both cases it was seen that the PCA had lower run time than those of others. The complete dataset had long run because of the huge volume. Time parameter, which I used to determine the speed and in the previous page had tabulated for both datasets on how neural networks behaved in each case, also accuracy parameters were plotted.

In case of data selection, there were few columns lower than that of the original dataset because some unimportant columns were eliminated from the model. In case of data transformation, a new transformed model was obtained in ICA, PCA and randomized projection which did not look like the original dataset.

In both cases, when I tried neural networks, the accuracy were high for the complete dataset. When we are having huge dataset, it takes long time to run and if we are having more than million rows, it is going to take too long to run and at that time we can use these feature transformation and feature selection algorithms. Also, the curse of dimensionality problem is somewhat handled when we reduce the number of variables in the model. In case of PCA, ICA and randomized projection if we get components which are going to explain most the variation then we can simply use that instead of wasting time implementing the entire dataset which has explanatory variables which do not explain much variation

Since it is not possible to implement the cluster diagram for all dimensions I chose to just represent with 2 columns for illustration purpose. The experiments were done for tolerance, components, various algorithms in each of the feature selection and transformation models. ICA had a lower intra cluster distance amongst all other in both datasets.