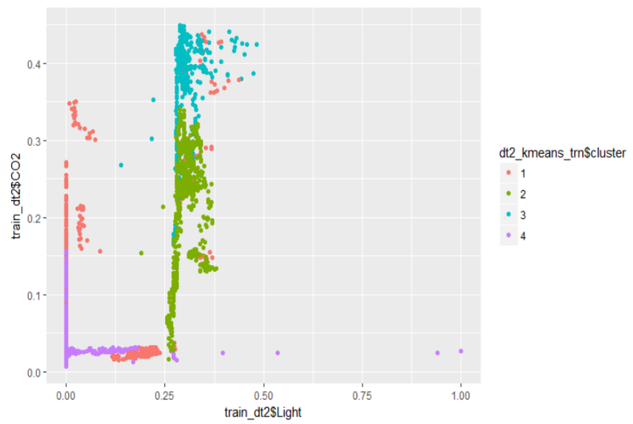
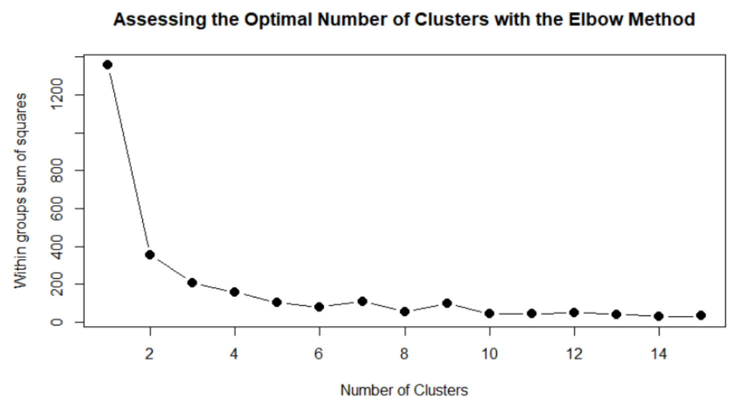
Project 4 Report

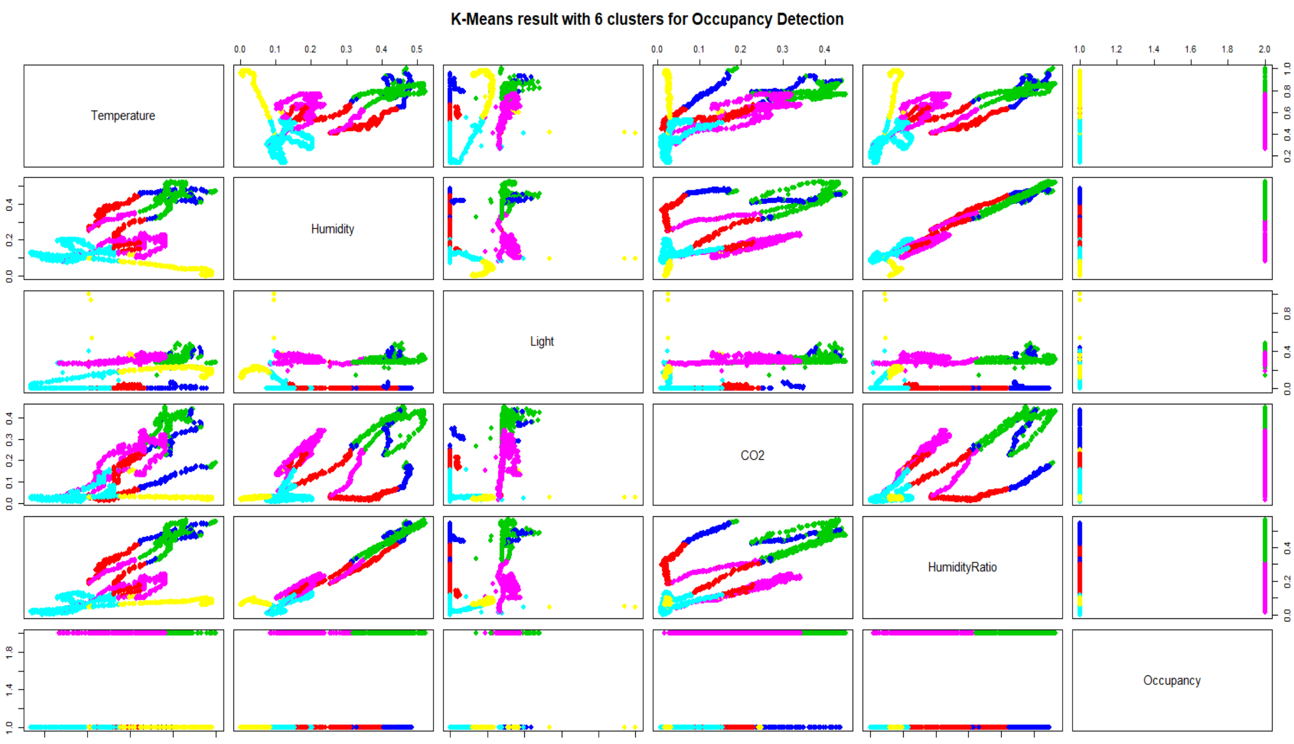
K-means Clustering:

Occupancy Dataset: Initial trial configuration with 4 clusters. The clusters look a bit well separated with some points distracted.





Choosing the optimum clusters as 6 from the Elbow method as we can observe the dip in the graph. Clusters comparatively looks better as compared to the above image of clusters



For Occupancy Detection: Sum of Square value before dimensionality reduction

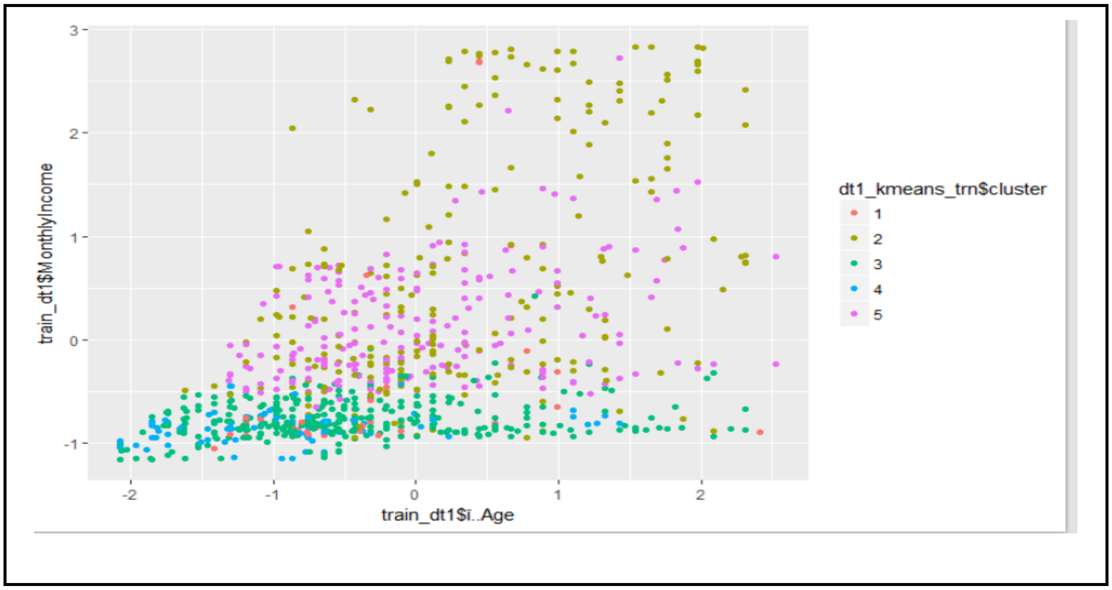
```
Within cluster sum of squares by cluster:
[1] 100.033564  16.496998   6.300081  29.385794
 (between_SS / total_SS =  88.8 %)
```

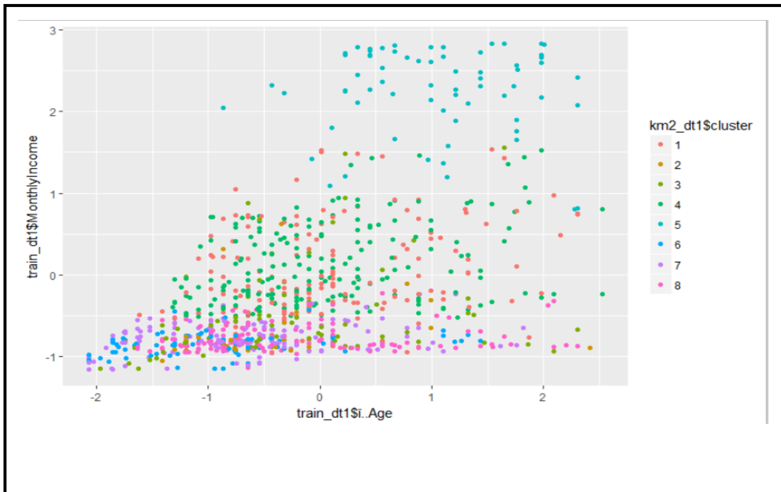
This value (88%) will help us to compare the results post dimensionality reduction.

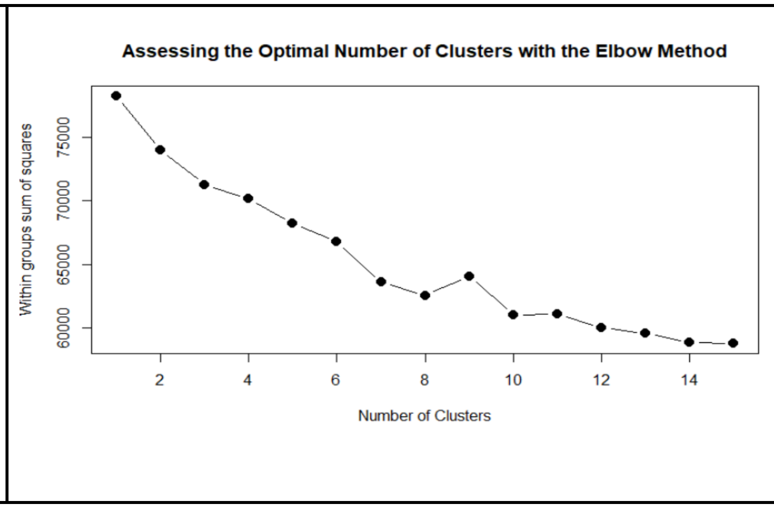
K Means Implementation for IBM Attrition Dataset:

```
Within cluster sum of squares by cluster:
[1] 14491.271  7931.264 20073.277 16662.663  7877.581
 (between_SS / total_SS =  14.3 %)
```

This value (14%) will help us to compare the results post dimensionality reduction.



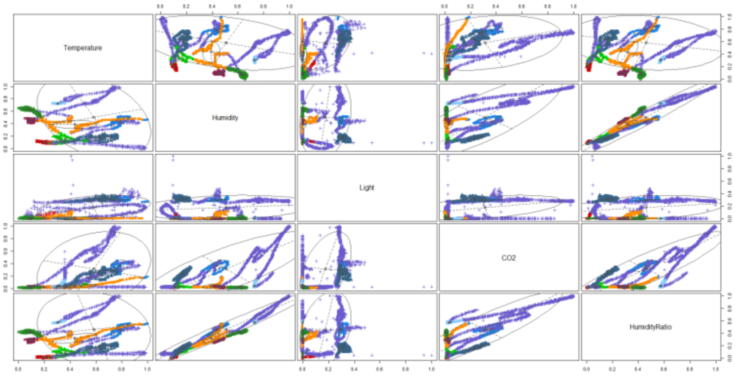


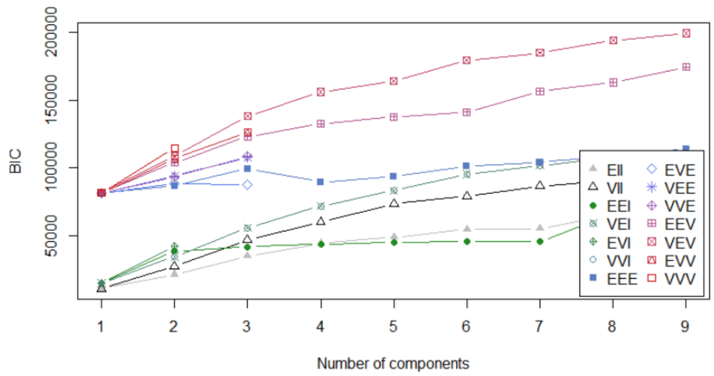


Choosing the optimum clusters as 8 from the Elbow method as we can observe the dip from the graph. The clusters are plotted on two features i.e., the Monthly income of the employee and age. The observations in cluster 5 seem to be scattered i.e. most of them have high income and are above the median age. None of the clusters seem to compact.

Expectation Maximization with Occupancy Detection Dataset

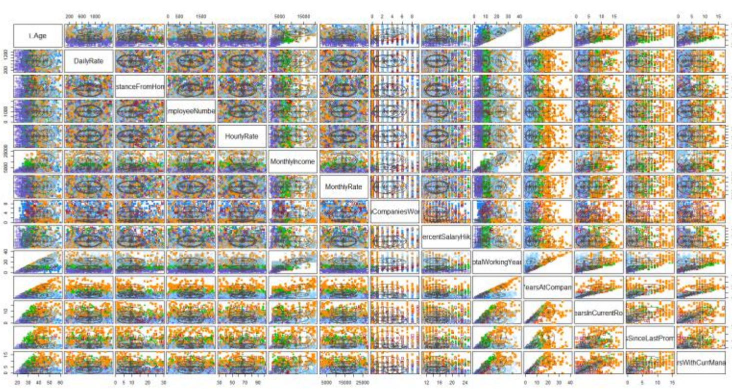
```
> summary(fit_occ)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VEV (ellipsoidal, equal shape) model with 9 components:

 log.likelihood    n  df      BIC      ICL
       100360.3 8143 156 199315.9 199218.1

Clustering table:
   1    2    3    4    5    6    7    8    9
 460  707 1056 1703 1286  633  920  799  579
```

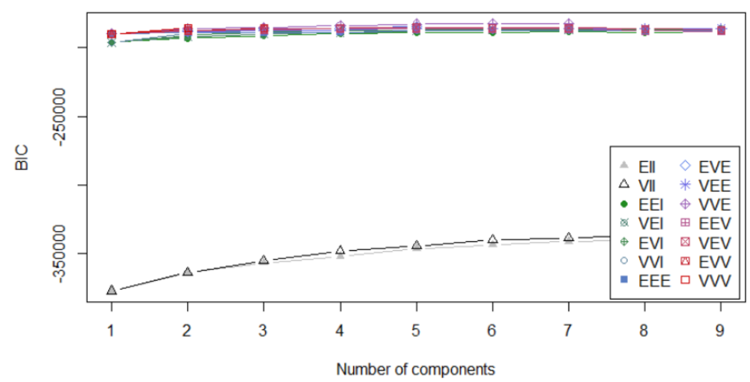




Expectation Maximization generates 9 soft clusters based on expected probability.

Expectation Maximization with IBM Attrition Dataset

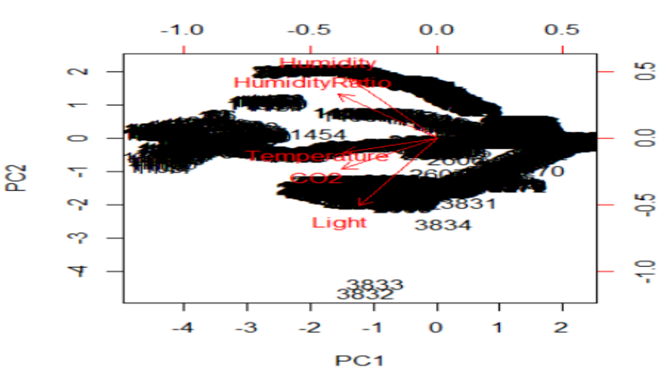





```
> summary(fit_attr)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VVE (ellipsoidal, equal orientation) model with 6 components:

 log.likelihood    n  df       BIC     ICL
      -90360.42 1470 264 -182646.2 -182877

Clustering table:
  1   2   3   4   5   6
392 230 218 297 144 189
```

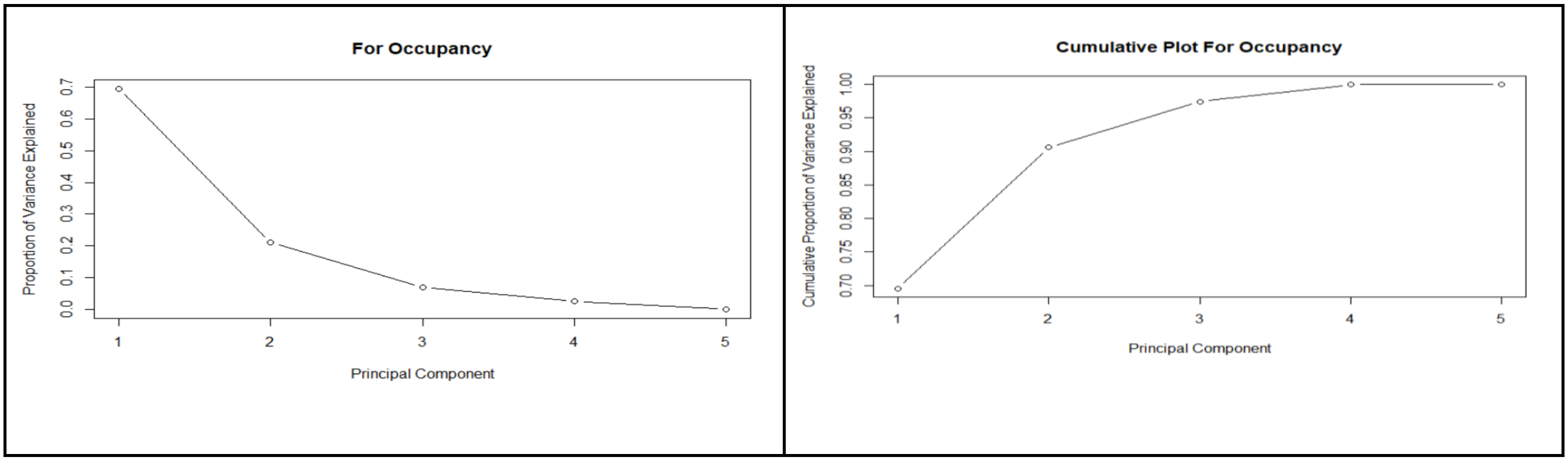
Expectation Maximization generates 6 soft clusters based on expected probability.

Principal Component Analysis for Occupancy Detection Dataset:



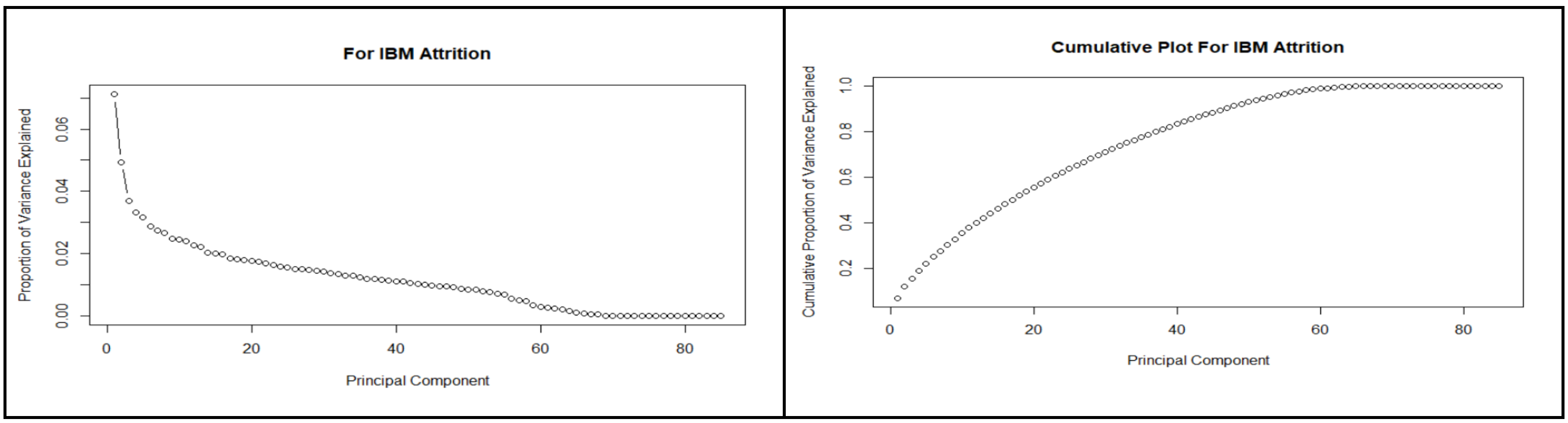
The image shows (on next page) the each of those features is plotted across two principal components.

new spaces you created with the various algorithms



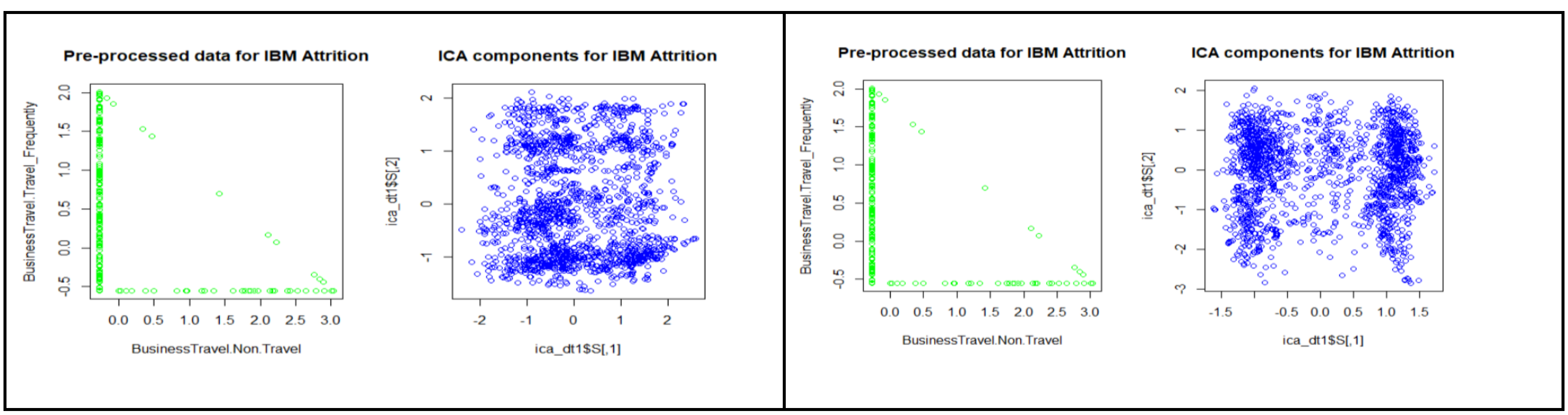
98% of variance explained by first 3 principal components. Principal components 4 and 5 are not much contributing to the to the variance and can be ignored.

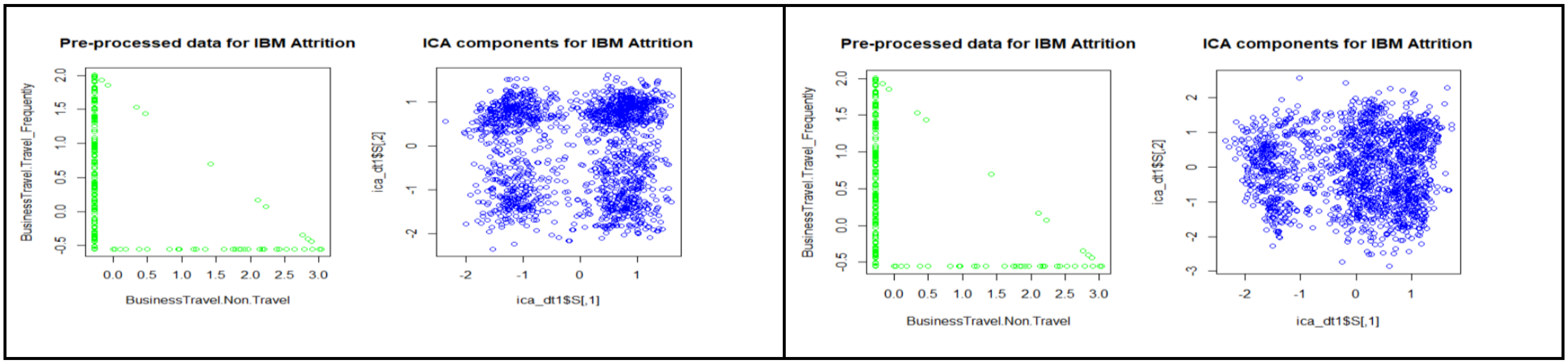
<u>Principal Component Analysis for IBM Attrition Dataset</u>

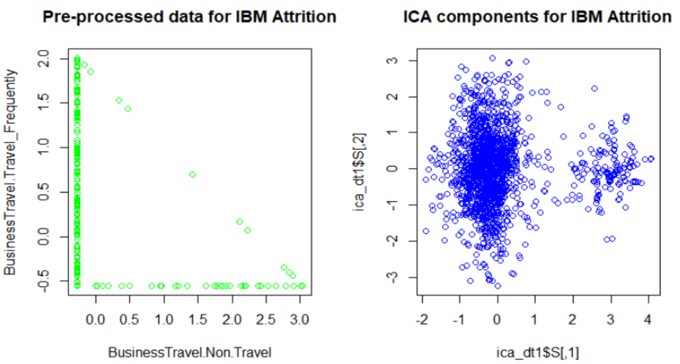


Post 98%, Principal components are not showing much variance. So, we can choose only 60 components for IBM dataset.Rest of the components can be ignored.

<u>Independent Component Analysis for Attrition</u>

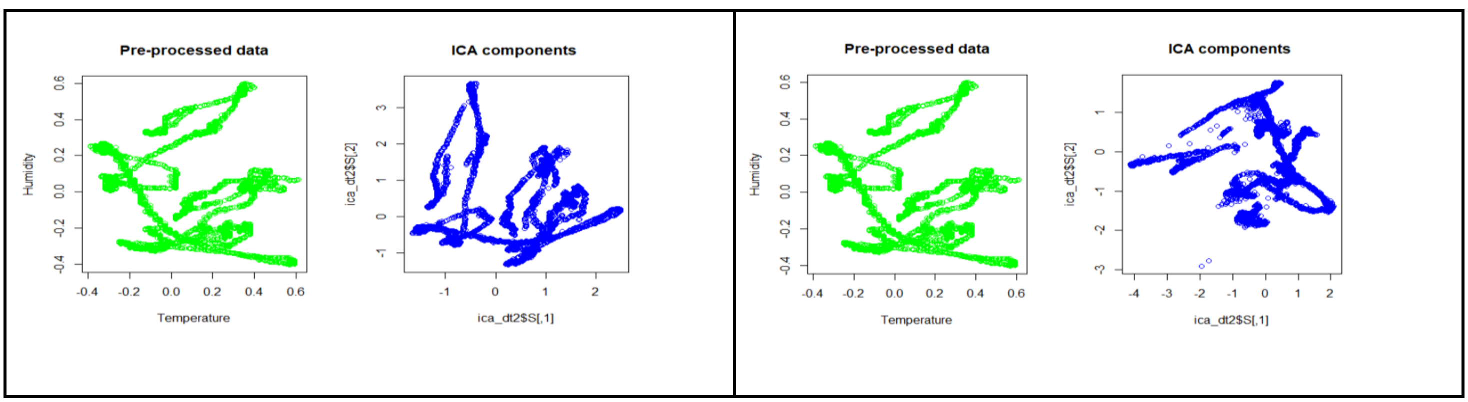


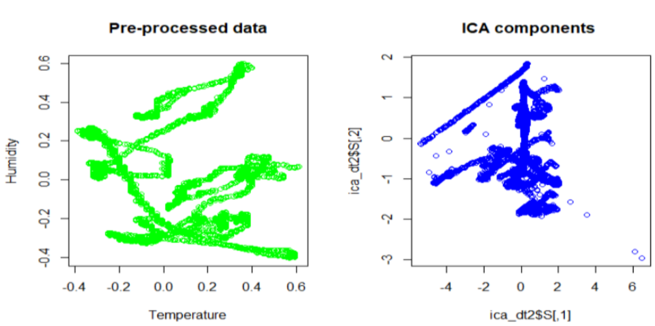




For independent component analysis, we tried a different number of features (5,10,15,20,25) to be extracted from the original dataset. The above chart represents the how data is separated into the ICA components. We can observe that when the 15 components are extracted, the data is distinguishable in the chart. So, 15 features from ICA is the optimum number which can be extracted.

Independent Component Analysis for Occupancy





Since we have just 5 features available for reduction, we tried a different number of features (2,3,4) to be extracted from the original dataset. The above chart represents the how data is separated into the ICA components. We can observe that when the 3 components are extracted, the data is distinguishable in the chart. So, 3 features from ICA is the optimum number which can be extracted.

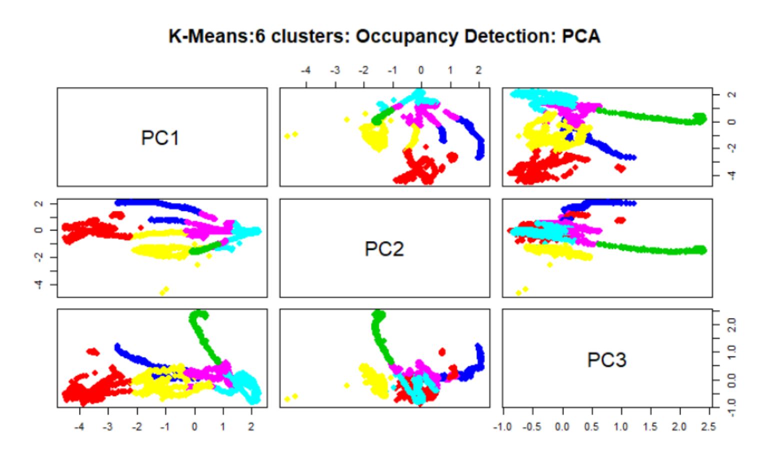
Clustering after dimensionality reduction

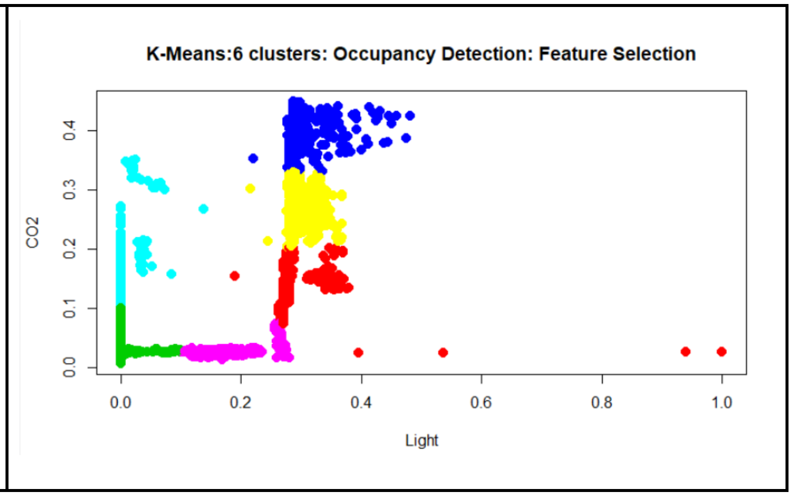
K-Means on Occupancy detection after PCA and Feature selection

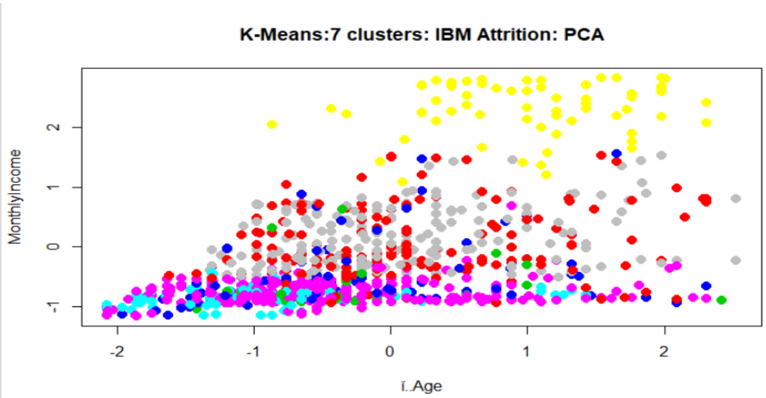
```
Within cluster sum of squares by cluster:
[1] 318.54947  75.43616 448.92015 261.29531 316.97194 440.84035
 (between_SS / total_SS =  90.6 %)
```

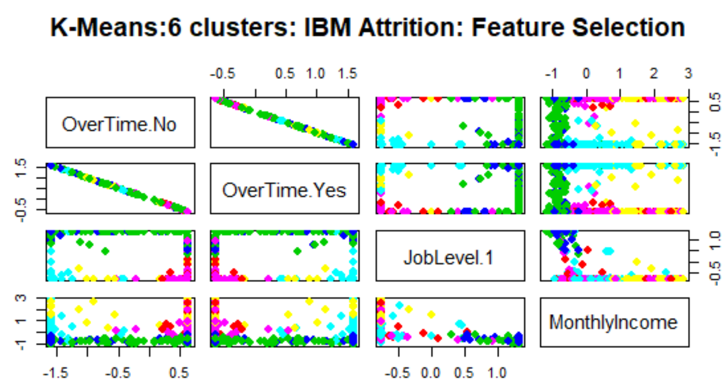
```
Within cluster sum of squares by cluster:
[1] 1.4576339 0.8115126 0.7906111 1.0971247 0.5622270 0.7764616
 (between_SS / total_SS =  96.4 %)
```

The value is increased from 88 to 90% for PCA and 96% for feature selection, so it's not compact. Choosing PCA for further operations based on these results.





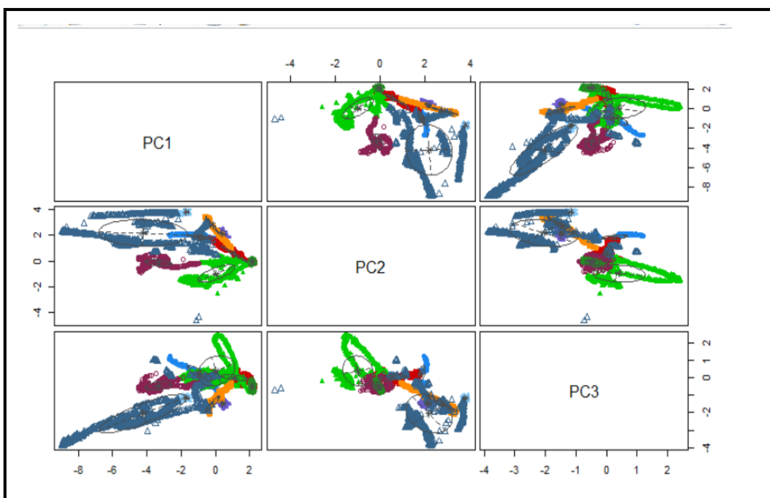


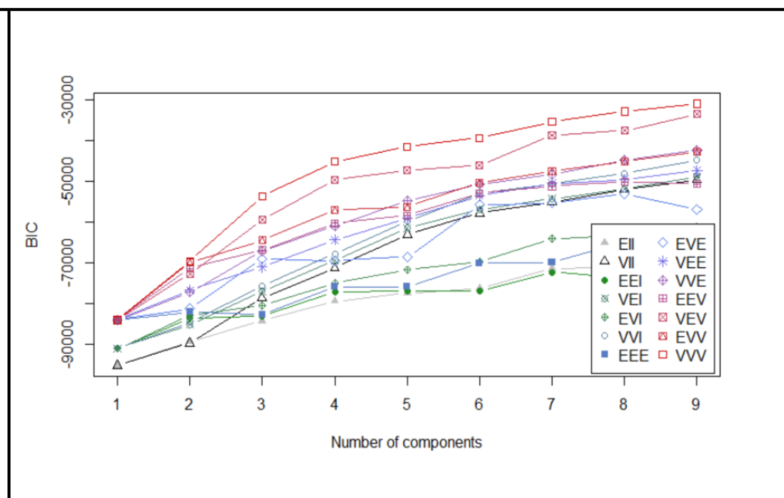



```
within cluster sum of squares by cluster:
[1] 13983.802 14672.897  4315.977 17187.394  7417.707  8818.930  2581.686
 (between_SS / total_SS =  18.4 %)
```

(for PCA) Within Cluster Sum of square value is increased from 14% to 18%. So, it's not compact.

<u>Expectation Maximization on Occupancy detection after PCA</u>

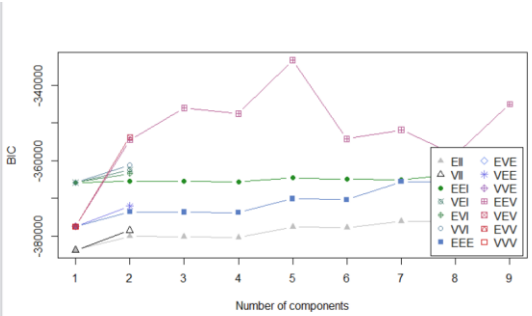




After doing PCA on occupancy detection the Expectation Maximization created 9 soft clusters which were more compact.

```
> summary(fit_occ1)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 9 components:

 log.likelihood    n df       BIC       ICL
      -15106.02 8143 89 -31013.48 -31426.84

Clustering table:
   1    2    3    4    5    6    7    8    9
 451 1293 1418  968 1106  618  570  624 1095
```

Expectation Maximization on IBM Attrition after PCA

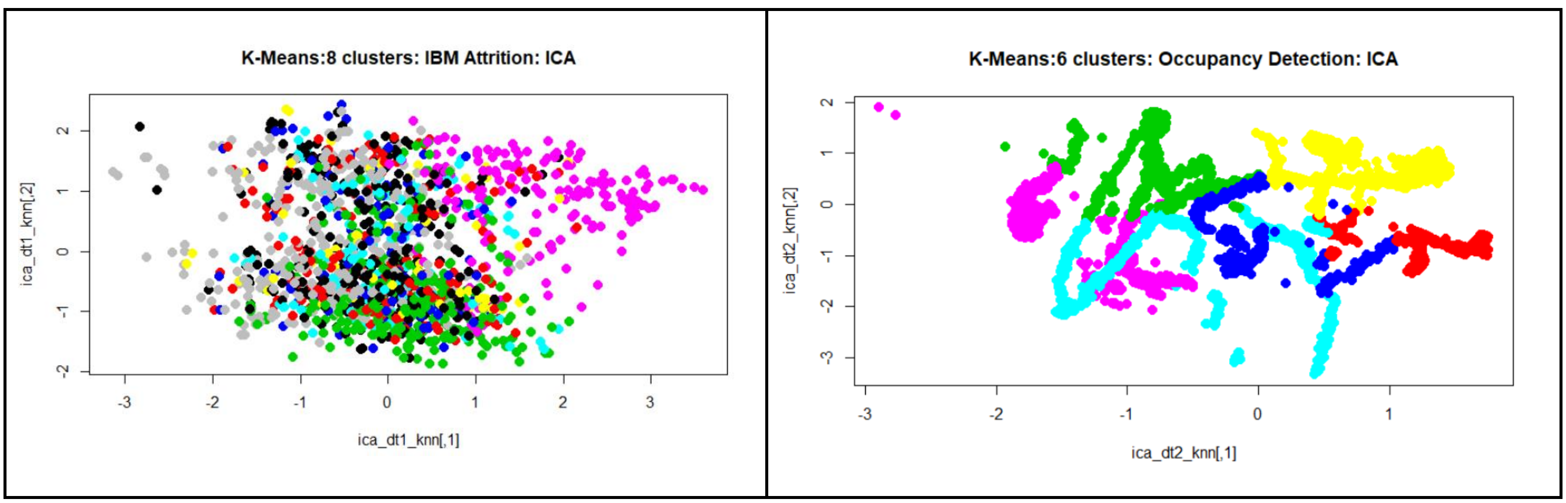



```
> summary(fit_attr1)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust EEV (ellipsoidal, equal volume and shape) model with 5 components:

 log.likelihood    n   df       BIC       ICL
        -131600 2014 9214 -333299.1 -333299.6

Clustering table:
   1    2   3   4   5
 132 1277 227 138 240
```

After doing PCA on IBM Attrition the Expectation Maximization created 6 soft clusters which were more compact.

Clustering after Independent component analysis

K-means on IBM Attrition and Occupancy detection after ICA

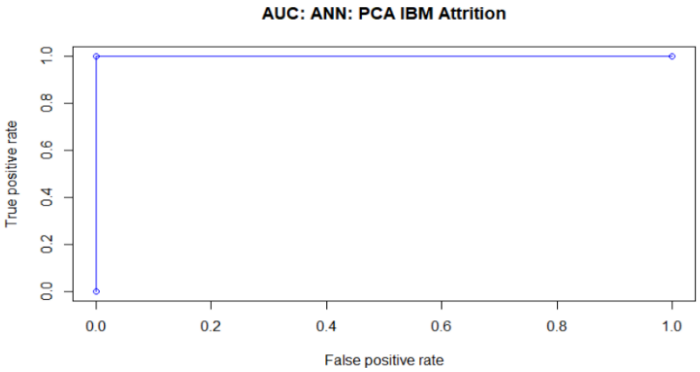


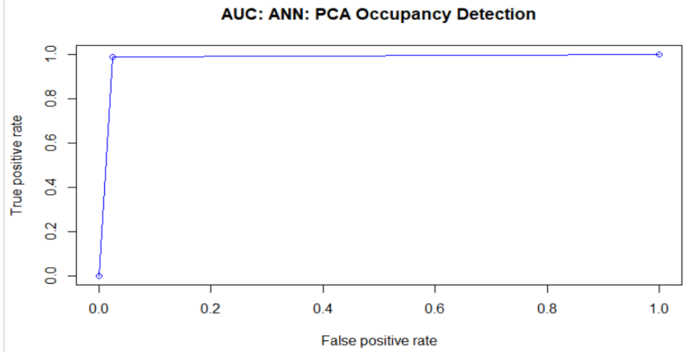
Clustering on the independent components which we extracted in the above steps. The above graph shows 8 clusters for the IBM Attrition dataset, the clustering is done on the 15 independent components after ICA. The graph on the right shows output of clustering on the 3 independent components after ICA

<u>Neural network learner after dimensionality reduction:</u>

"AUC of ANN Model on Train Occupancy dataset: 0.98289", "AUC of ANN Model on Train IBM Attrition dataset: 0.987 "

The values and curves as shown in the image below indicate that maximum area is covered under the curve which highlights model performance is good. Confusion Matrix for IBM Attrition (left side) post with PCA using a neural network. Model sensitivity, specificity is also good.






```
> print(conf1)
Confusion Matrix and Statistics

          Reference
Prediction  no yes
       no  533   0
       yes   0 474

               Accuracy : 1
                 95% CI : (0.9963, 1)
    No Information Rate : 0.5293
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.5293
         Detection Rate : 0.5293
   Detection Prevalence : 0.5293
      Balanced Accuracy : 1.0000

       'Positive' Class : no
```

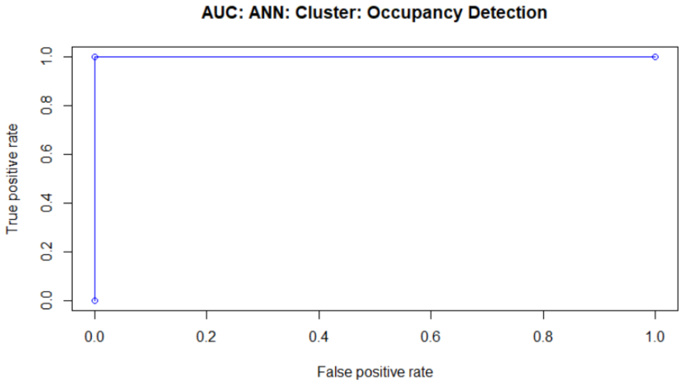
```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  2858   11
       yes   72 1130

               Accuracy : 0.9796
                 95% CI : (0.9748, 0.9837)
    No Information Rate : 0.7197
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9503
 Mcnemar's Test P-Value : 4.523e-11

            Sensitivity : 0.9754
            Specificity : 0.9904
         Pos Pred Value : 0.9962
         Neg Pred Value : 0.9401
             Prevalence : 0.7197
         Detection Rate : 0.7020
   Detection Prevalence : 0.7047
      Balanced Accuracy : 0.9829

       'Positive' Class : no
```

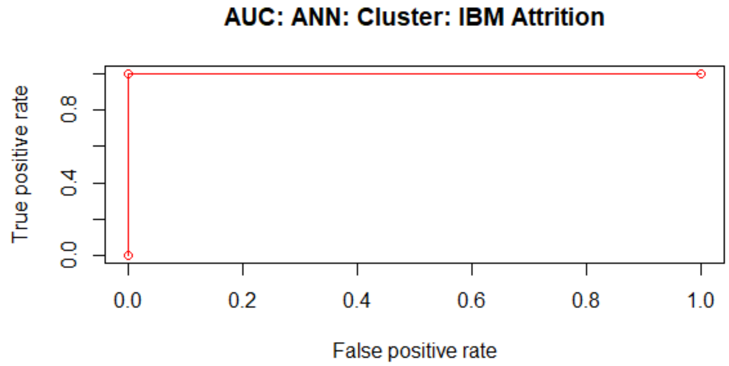
Confusion Matrix for IBM Attrition (right side) post with PCA using a neural network. This model also gave better results as compared to the dataset without dimension transformation.

<u>Clustering and Neural Network:</u> Based on the clustering output obtained from the task 1, We applied the neural network learner on this new data consisting of only clustering results as features and class label as the output for both Occupancy detection and IBM attrition dataset. The left section of the below diagrams is of Occupancy Detection and right sections are of IBM Attrition which contains confusion Matrix and AUC curve. The Accuracy and AUC values for both datasets is better.

```
> print(conf3)
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  2930    0
       yes    0 1141

               Accuracy : 1
                 95% CI : (0.9991, 1)
    No Information Rate : 0.7197
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.7197
         Detection Rate : 0.7197
   Detection Prevalence : 0.7197
      Balanced Accuracy : 1.0000

       'Positive' Class : no
```

```
For Train Dataset
Confusion Matrix and Statistics

          Reference
Prediction  no yes
       no  533   0
       yes   0 474

               Accuracy : 1
                 95% CI : (0.9963, 1)
    No Information Rate : 0.5293
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.5293
         Detection Rate : 0.5293
   Detection Prevalence : 0.5293
      Balanced Accuracy : 1.0000

       'Positive' Class : no

[1] "AUC of ANN Model on Train Occupany dataset: 1"
```





ANN output (from Project 3) without clustering output

Model	**Accuracy IBM**	**AUC IBM**	**Accuracy Occupancy**	**AUC Occupancy**
ANN	89.37%	0.8964721	91.23%	0.94877

If we compare these values with output we received after dimensionality reduction and clustering, We can say we get much better results.

References:

https://rpubs.com/FelipeRego/K-Means-Clustering

https://www.r-bloggers.com/k-means-clustering-in-r/

https://uc-r.github.io/kmeans_clustering

http://miningthedetails.com/blog/r/fselector/

https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/