**Subject: Data Preparation, Clustering and Classification on**

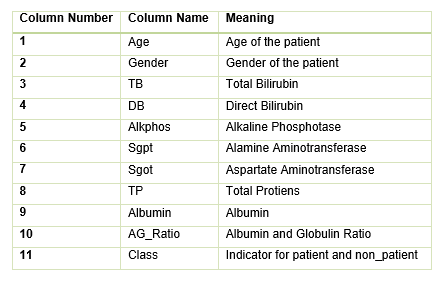
**“ILPD (Indian Liver Patient Dataset)”.**

**Objective:** The goal of this project is to do Clustering and Classification on “ILPD (Indian Liver Patient Dataset” dataset. This data has 583 observations(rows) and 11 attributes(columns).The attributes of the columns contains different parameters such as Age, Gender, total Bilirubin, Albumin etc. while the last column is a binary class variable with value 1 for “Patient” and 2 for “Non-Patient”. I will cover the total report by three main tasks such as Data Preparation, Clustering and Classification.

**Task1-Data Preparation:**

1.1: At first I have import the required packages by library command which will be needed for the whole project. These packages are**:"ggplot2"** for visualization,**"plotly"** for Interactive data visualizations, “**psych”** for correlation visualizations, **“caret"** for Machine learning, "**party**" for decision Tree and finally **"class”.** Then I have extracted the data into an R dataframe by **”read.CSV”** function and named it as “ilpd”.

1.2: I’ve assigned the following names to the 11 different columns such as:



1.3: There are some missing values in the column AG\_Ratio. I’ve filled them with the median of the column. To find the Median at first I’ve omit the NA value of the column by **omit.na** function and then calculate the median of the rest values. After finding the Median I’ve replaced the NA values with that value by **is.na** function.

1.4: I’ve replaced all “2” in the “class” column with “0” to indicate “Non-Patient”. Now “0” represents Non-patient and “1” represents “Patient”.

1.5: Since I’ve changed the value of class column, R might define the Class column as Integer. That’s why I have changed its type from integer to factor by **as.factor** function.

1.6. Finally I have saved the dataframe as ‘ilpd\_processed.Rda”file by **saveRDS()** function.

**Task 2 – Clustering(K-Means & Hierarchical):**

2.1: At first I’ve load the preprocessed data file from Task1 into a data frame by **readRDS** function and saved it in the name of “ilpd\_df”. I also excluded Age, Gender, AG\_Ratio and Class attributes in this task by **c()** function and saved the file as”ilpd7col”name.

2.2: I rescaled the values of every column to the range of (0,1) by **FUN = function(X) (X - min(X))/diff(range(X))** and converted the datatype from vector to dataframe by **data.frame** function

2.3: I’ve clustered the data into 2 Clusters (i, e. k=2) using K-Means clustering with the default parameters by the function **kmeans()** and plotted the result of the clusters as 2D (X=Alkphos, Y=TP) by **ggplot()** function. The resulted plot is as follows:

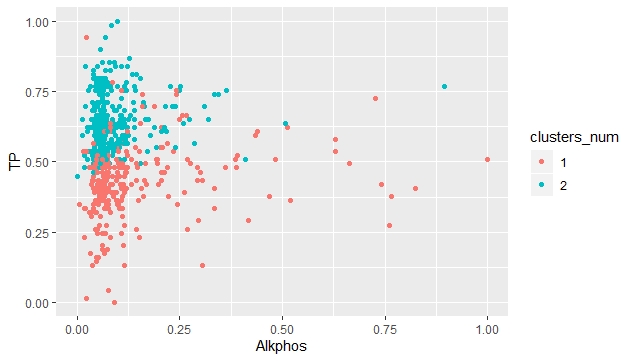


Fig:2.3- K-means Clustering with k=2 & default parameter.

2.4:I’ve plotted another 2D plot with the same dimension above but colored the points according to class column and found the resulted plot as follows:

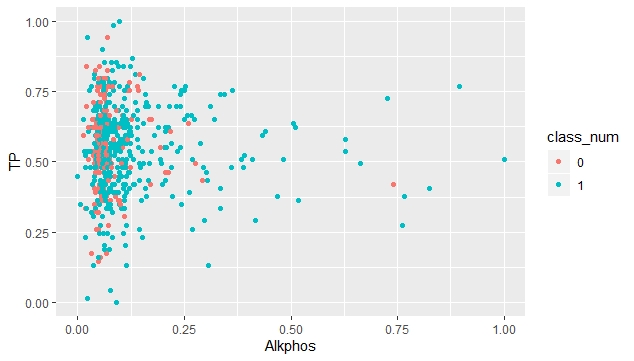


Fig: 2.4a- K-means Clustering with k=2, default parameter & Class Column Consider.

I’ve also plotted another plot to cluster the data into 2 clusters( k=2) by K-Means clustering considering cluster number with changed hyper parameters and found follows figure:

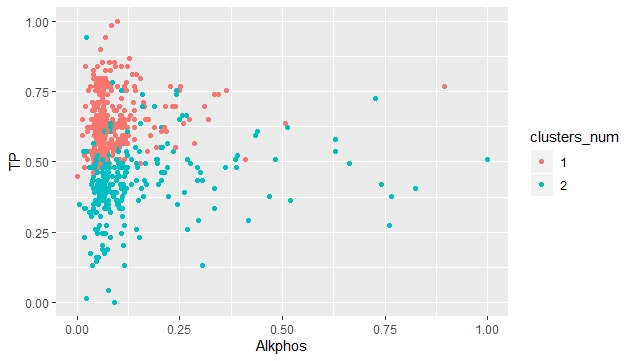


Fig:2.4b- K-means Clustering with k=2 , changed hyper parameter & clusters number Consider.

2.5: As a comparison of the two above plots (task 2.3 & Task 2.4a) we can see the clusters visually represent the patient vs non\_patient classes.

2.6:I have clustered the data into more than 2 clusters(i.e, k=3,4,5) using K-means clustering and plotted all the results as follows:

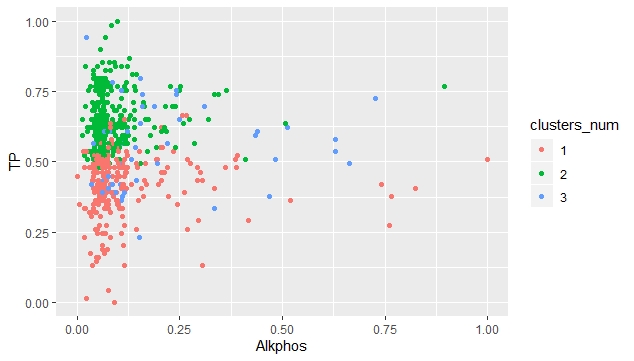


Fig:2.6a:K=3 clustering

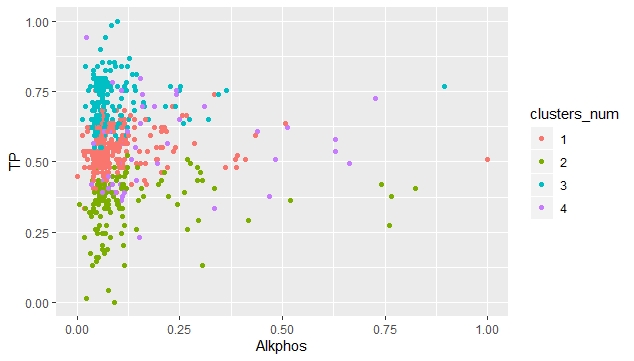


Fig:2.6b:K=4 clustering

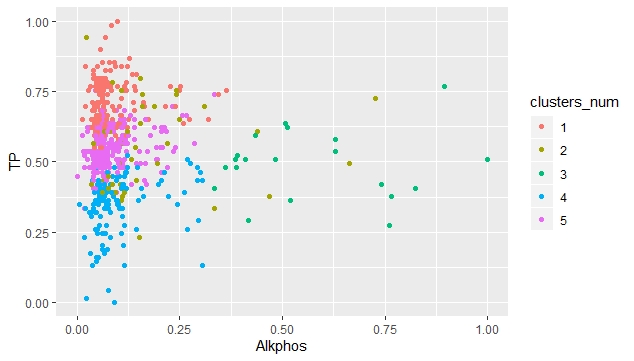


Fig:2.6c:K=5 clustering

2.7: As a comparison of the plots and Sum Squared Error (SSE) obtained in the previous tasks my observations are as follows:

K-means clustering with 2 clusters of sizes 313, 269 (default parameter)

Within cluster sum of squares by cluster:

[1] 11.09091 30.38465

(between\_SS / total\_SS = 31.9 %)

K-means clustering with 2 clusters of sizes 313, 269(Changed hyper parameters)

Within cluster sum of squares by cluster:

[1] 11.09091 30.38465

(between\_SS / total\_SS = 31.9 %)

K-means clustering with 3 clusters of sizes 295, 237, 50

Within cluster sum of squares by cluster:

[1] 9.956856 12.061168 8.105771

(between\_SS / total\_SS = 50.5 %)

K-means clustering with 4 clusters of sizes 228, 137, 168, 49

Within cluster sum of squares by cluster:

[1] 5.962593 6.958464 4.528318 7.930842

(between\_SS / total\_SS = 58.3 %)

K-means clustering with 5 clusters of sizes 166, 45, 20, 132, 219

Within cluster sum of squares by cluster:

[1] 3.845438 6.711155 1.930572 4.808752 4.270082

(between\_SS / total\_SS = 64.6 %)

Since for the K=2 the SSE is minimum which is 31.9% and it remains same after tuning the hyper parameter hence 2 cluster(k=2) is the best quality clustering among all of these.

2.8: I’ve applied Hierarchical clustering to data using the **“hclust”** function with default parameters and plotted the corresponding dendrogram. Also clusters them into 2,3,4 and 5 clusters. The resulted plots are as follows:

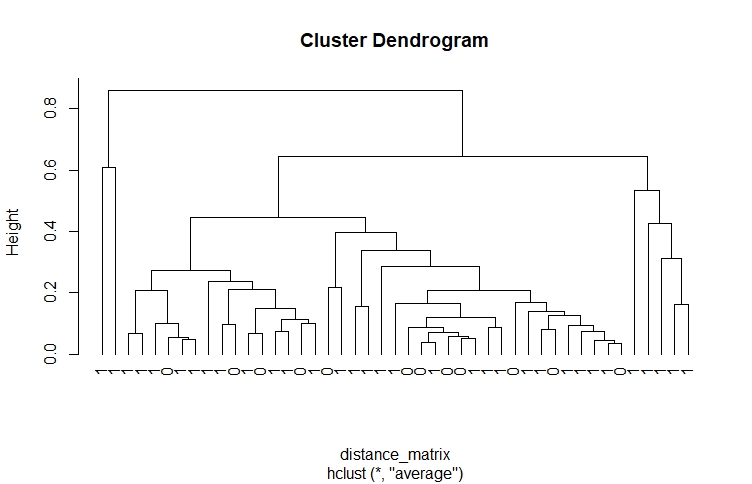


Fig:2.8a: Hierarchical clustering with default parameters

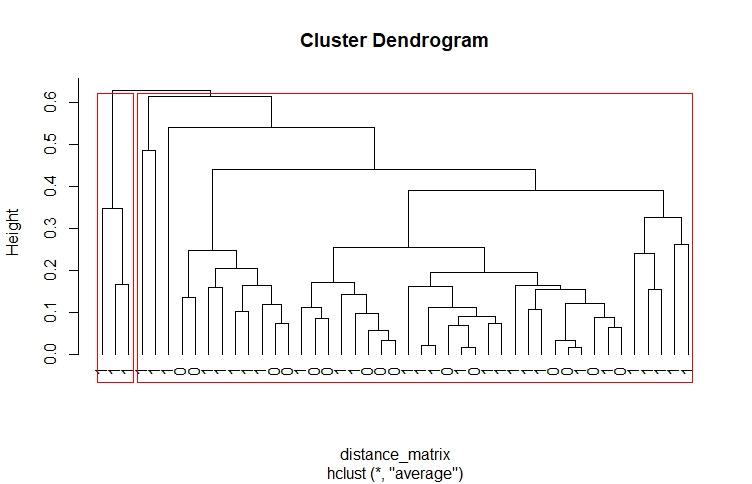


Fig:2.8b: Clustering the Dendogram into 2 Clusters

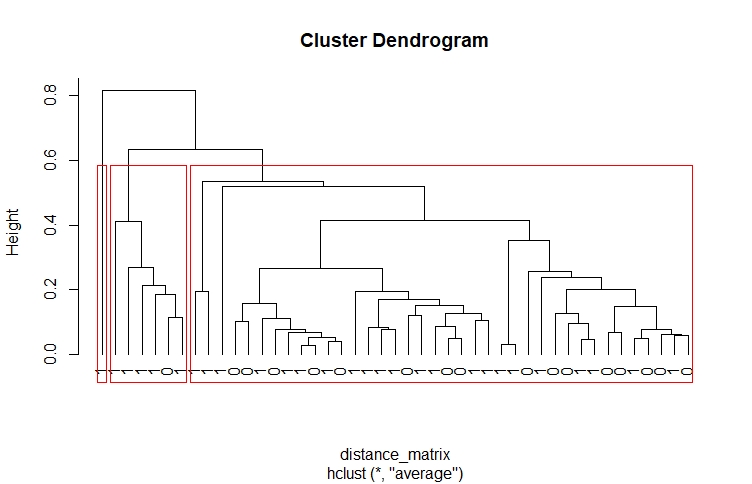


Fig:2.8c: Clustering the Dendogram into 3 Clusters

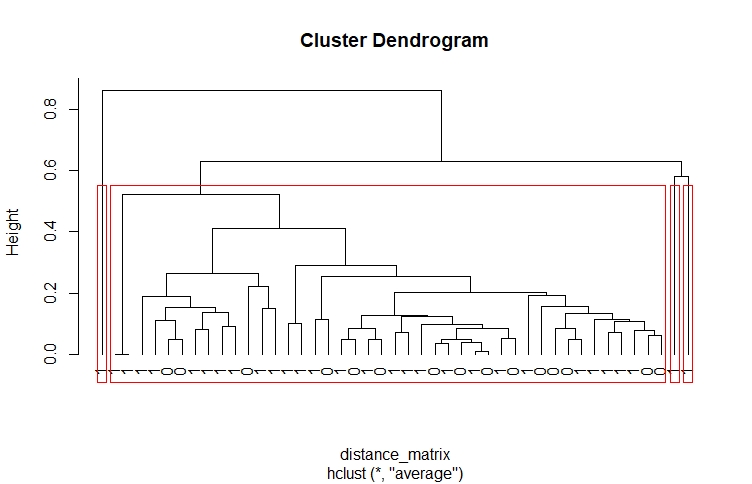


Fig:2.8d: Clustering the Dendogram into 4 Clusters

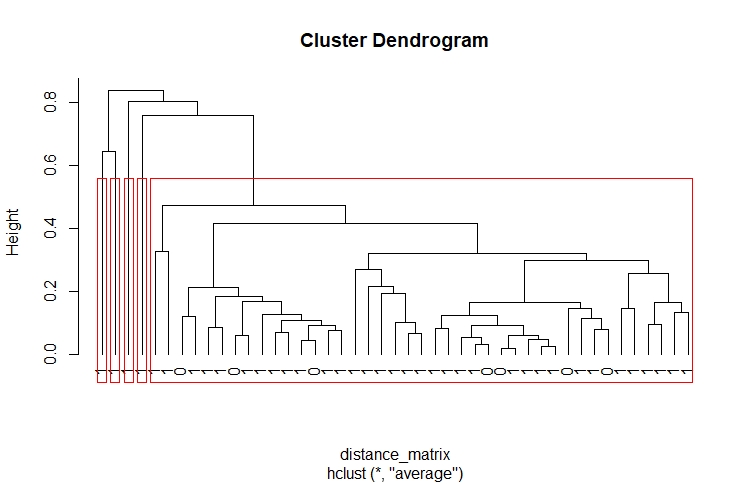


Fig:2.8e: Clustering the Dendogram into 5 Clusters

2.9: After comparing the plots of 2.3,2.4,2.6 and 2.8,my observation is as I can see more than 3 distinguish clusters in some plotting, there should have a new subtype diseases.

2.10:I have tried different agglomeration methods(I,e. “MIN”,”MAX” and ”AVERAGE”) clustering by the methods of **“single”,”complete**” and “**average**” and found the below dendrograms:

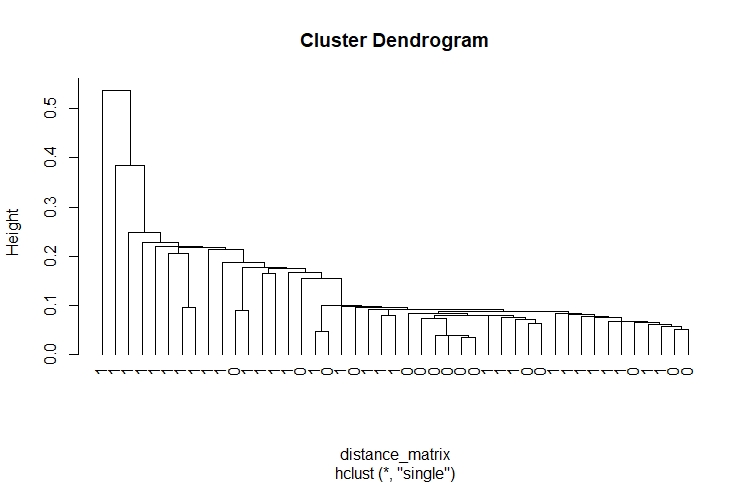


Fig:2.10a- MIN agglomeration hierarchical clustering

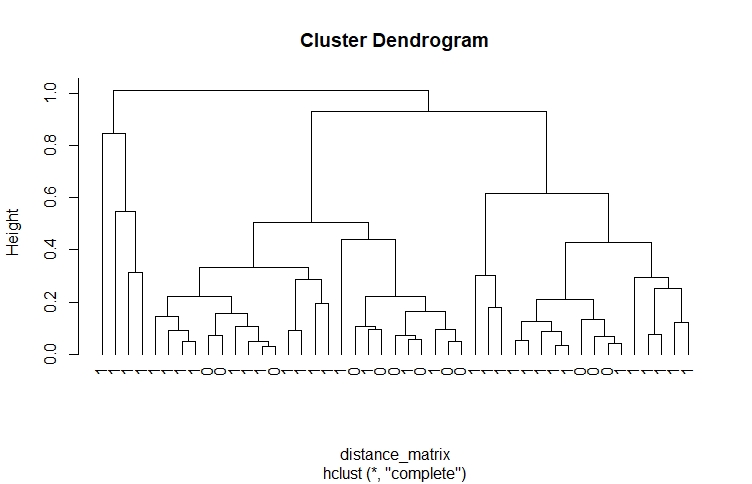


Fig:2.10b- MAX agglomeration hierarchical clustering

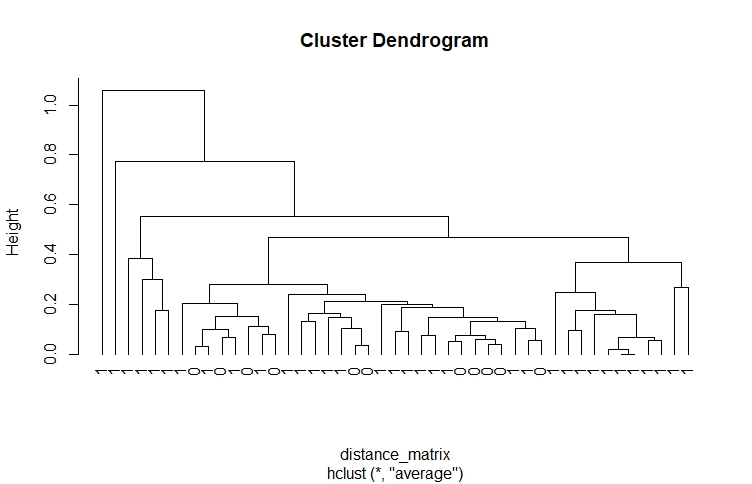


Fig:2.10b- MAX agglomeration hierarchical clustering

From the above three plots we can see the data is sensitive to the used agglomeration method. I can also see that the default agglomeration method used in task 2.8 is the “AVERAGE” method.

**Task 3 -Classification (Binary Classification using Decision Tree & K-NN Techniques )**

3.1: I have loaded the preprocessed data file from Task 1 into data frame and saved as ilpd\_df by **readRDS** function. Since knn() function would not accept the string labels for gender,I have changed the Gender labels from factor to numbers by factor(Gender, labels = c(1, 2) function.

Then I have set seed(45), because seed is a Pseudo random generator which generates same sequence every time. After that I have divided the dataset into "Training"(70%) & "Test"(30%) by **prob = c(0.7, 0.3)** function.

3.2:I have learnt a classification tree from the training data using the default parameters of the **ctree()** function from the **“Party”** library. I have plotted the classification tree and found the below figure:

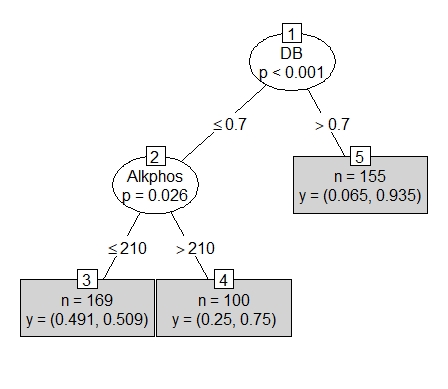


Fig:3.2:-Classification using Decision Tree(Default PM)

From this tree we can see **(1)**the “DB” has been selected as parent or decision nodes and splitted to check the condition whether <= 0.7 or >0.7. **(2)**On Basis of that <= 0.7 wing creates new decision node as Alkhphos and again run through two conditions as <= 210 or >210. This results to reach out two new leaf nodes as **(3)**n=169 and **(4)**n= 100.

On the other hand the condition >0.7 reached a leaf node as **(5)**n=155

We can say “DB” and “Alkphos” are important variables.

From the learned tree I have predicted the class labels of the “Test Data”by **predict() function.** Also calculated the accuracy, precision and recall by confusion matrix. The result is as follows :

prediction\_dt 1 2

1 0 0

2 49 109

3.3:I have built the classification tree again via the **ctree()** function but using **ctree\_control()**parameters to get a better accuracy by tuning. I have installed **“partykit”** **package** and **“partykit”library**,as it contains much improved re-implementations of **ctree().**

I used **ctree\_control(teststat = c("quadratic", "maximum"), splitstat = c("quadratic", "maximum"), mincriterion = 0.9999)** hyperparameters to tune and got the new plot as follows.

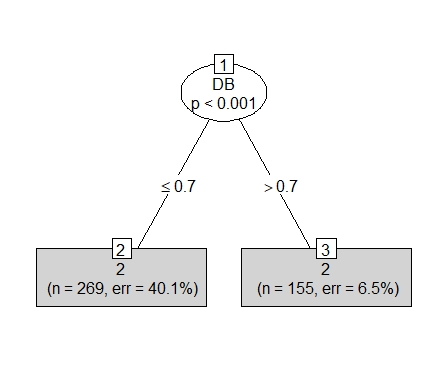


Fig:3.3a:-Classification using Decision Tree-(with Ctree Control Parameter)

I can see the tree configuration has been changed with this ctree\_control hyperparameter tuning.So by tuning some parameters we can achieve more meaningful representation.

From this new learned tree I have predicted the class labels of the “Test Data”by **predict() function.** Also calculated the accuracy, precision and recall by confusion matrix. The result is as follows :

prediction\_dt 1 2

1 0 0

2 49 109

Now, I have changed one of the Ctree\_control hyperparameters **mincriterion = 0.6000** and found the below plot:

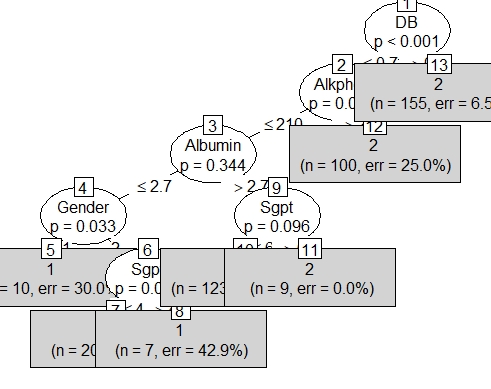


Fig 3.3b: Classification using Decision Tree-(with Ctree Control Parameter mincriterion = 0.6000)

I can see the tree configuration has been changed a lot with this ctree\_control hyperparameter tuning.So by tuning some parameters we can achieve more meaningful representation as well as better accuracy.

From this new learned tree I have predicted the class labels of the “Test Data”by **predict() function.** Also calculated the accuracy, precision and recall by confusion matrix. The result is as follows :

prediction\_dt 1 2

1 24 37

2 25 72

3.4:I have applied K-NN classification to predict the labels in the test subset with **knn()** function and calculated the accuracy,precision and recall.I have tried different values of K(i.e. K= 1,2,3,4,5) and found the below results:

**K=1**

table(prediction\_knn1, test\_data$Class)

prediction\_knn1 1 2

1 20 20

2 29 89

Accuracy: 0.6899 ACC=(TP+TN)/TP+TN+FP+FN

Precision:0.5000 PPV=TP/TP+FP

Recall:0.4082 TPR=TP/TP+FN

F1 Score:0.4494 F1=2TP/(2TP+FP+FN)

**K=2**

table(prediction\_knn2, test\_data$Class)

prediction\_knn2 1 2

1 16 23

2 33 86

Accuracy: 0.6456

Precision:0.4103

Recall:0.3265

F1 Score:0.3636

**K=3**

table(prediction\_knn3, test\_data$Class)

prediction\_knn3 1 2

1 19 19

2 30 90

Accuracy: 0.6899

Precision:0.5000

Recall:0.3878

F1 Score:0.4368

**K=4**

table(prediction\_knn4, test\_data$Class)

prediction\_knn4 1 2

1 21 19

2 28 90

Accuracy: 0.7025

Precision:0.5250

Recall:0.4286

F1 Score:0.4719

**K=5**

table(prediction\_knn5, test\_data$Class)

prediction\_knn5 1 2

1 15 14

2 34 95

Accuracy: 0.6962

Precision:0.5172

Recall:0.3061

F1 Score:0.3846

From the above result if we do comparison then we can find k=4 has the highest value of F1 ,

which is 0.4719. So **K=4** will give us better classification.

**Conclusion:**

The results of K-means Clustering ,Hierarchical clustering, Decision Tree binary Classification, K-NN binary classification were very significant and

meaningful to draw some precise observations.

**Reference:**

1. J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann,2001.(Chap:1,2,3)
2. Lecture & Tutorial Notes week 2,3,4,5,6. INFS-7203-Data Mining, The University of Queensland.
3. <https://stackoverflow.com>
4. www.rdocumentation.org /packages/partykit.