**20BDS0146**

**VENNELA G**

**PROGRAMMING FOR DATA SCIENCE**

**LAB ASSESSMENT-4**

1. Read mtcars dataset from R and perform following model fitting techniques (Any other datasets also applicable)

LOGISTIC REGRESSION

install.packages("caTools")

install.packages("ROCR")

library(caTools)

library(ROCR)

data(mtcars)

x=mtcars

split<-sample.split(mtcars,SplitRatio = 0.8)

TrainData<-subset(mtcars,split=="TRUE")

TestData<-subset(mtcars,split=="FALSE")

logistic\_model<- glm(vs~wt+disp, data = TrainData, family = "binomial")

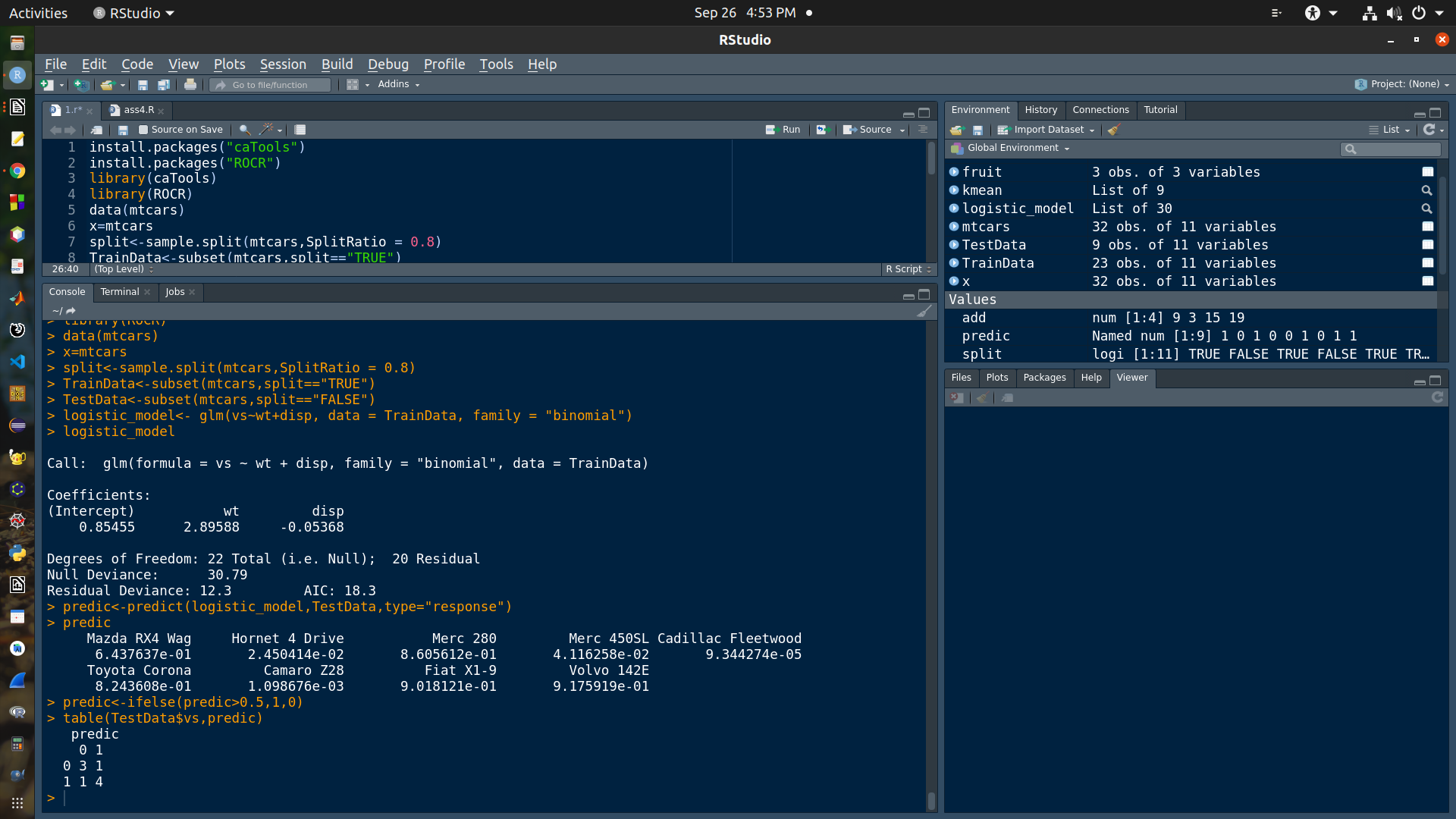
logistic\_model

predic<-predict(logistic\_model,TestData,type="response")

predic

predic<-ifelse(predic>0.5,1,0)

table(TestData$vs,predic)



DECISION TREE

install.packages("caTools")

install.packages("ROCR")

install.packages("partykit")

library(caTools)

library(ROCR)

library(partykit)

data(mtcars)

x=mtcars

split<-sample.split(mtcars,SplitRatio = 0.8)

TrainData<-subset(mtcars,split=="TRUE")

TestData<-subset(mtcars,split=="FALSE")

decision\_tree<-ctree(vs~.+disp,TrainData)

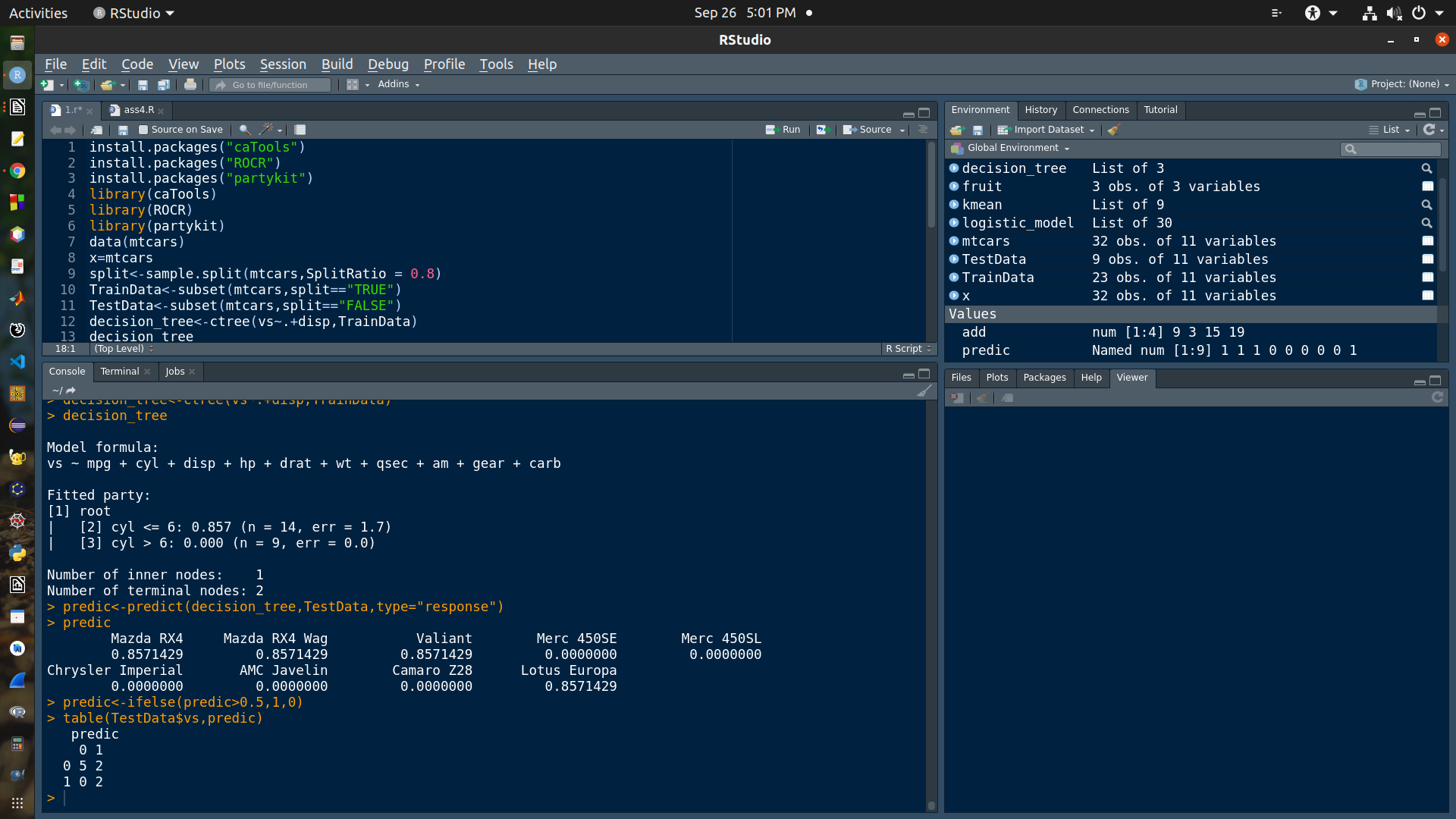
decision\_tree

predic<-predict(decision\_tree,TestData,type="response")

predic

predic<-ifelse(predic>0.5,1,0)

table(TestData$vs,predic)



NAIVE BAYES

install.packages("caTools")

install.packages("ROCR")

install.packages("e1071")

install.packages("caret")

library(caTools)

library(ROCR)

library(e1071)

library(caret)

data(mtcars)

x=mtcars

split<-sample.split(mtcars,SplitRatio = 0.8)

TrainData<-subset(mtcars,split=="TRUE")

TestData<-subset(mtcars,split=="FALSE")

naive\_bayes<-naiveBayes(vs~.+disp, data = TrainData)

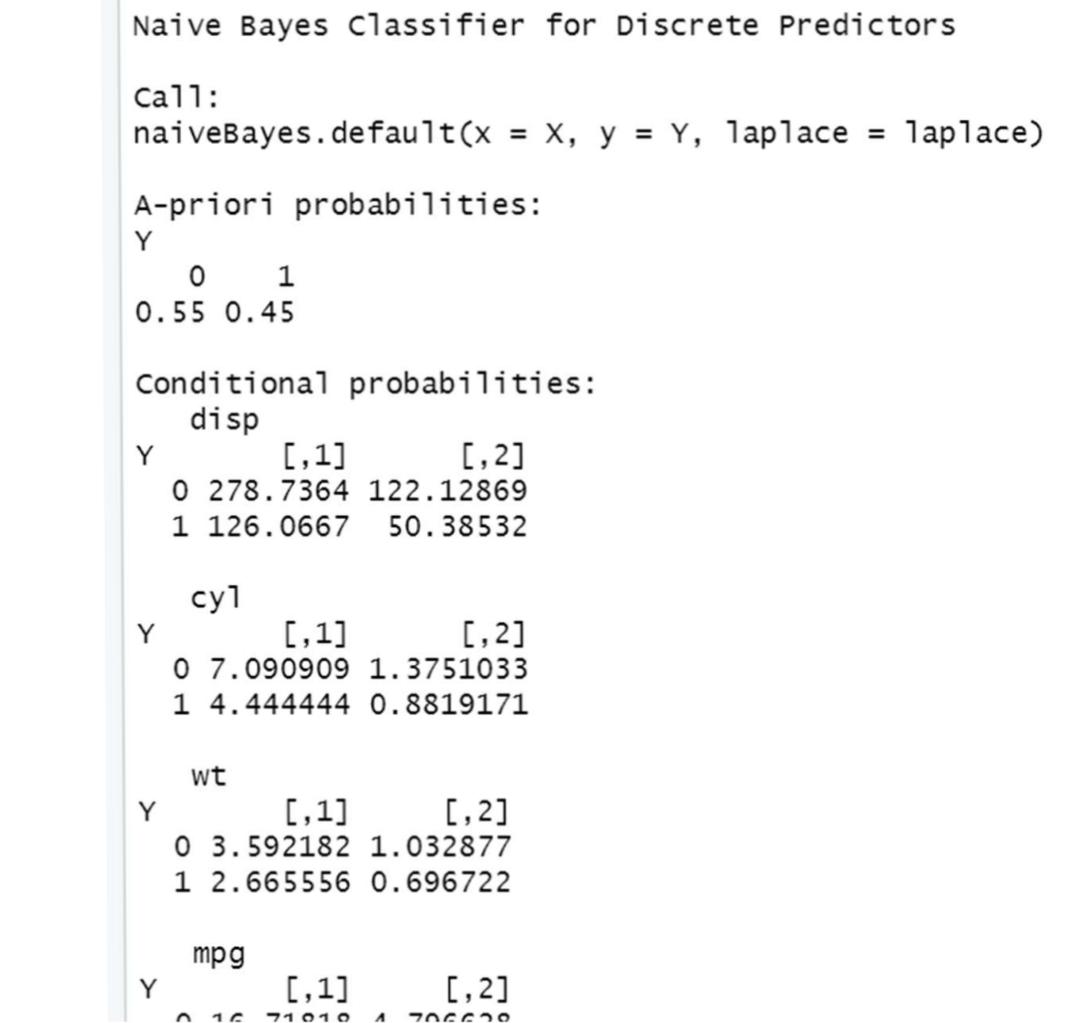
naive\_bayes

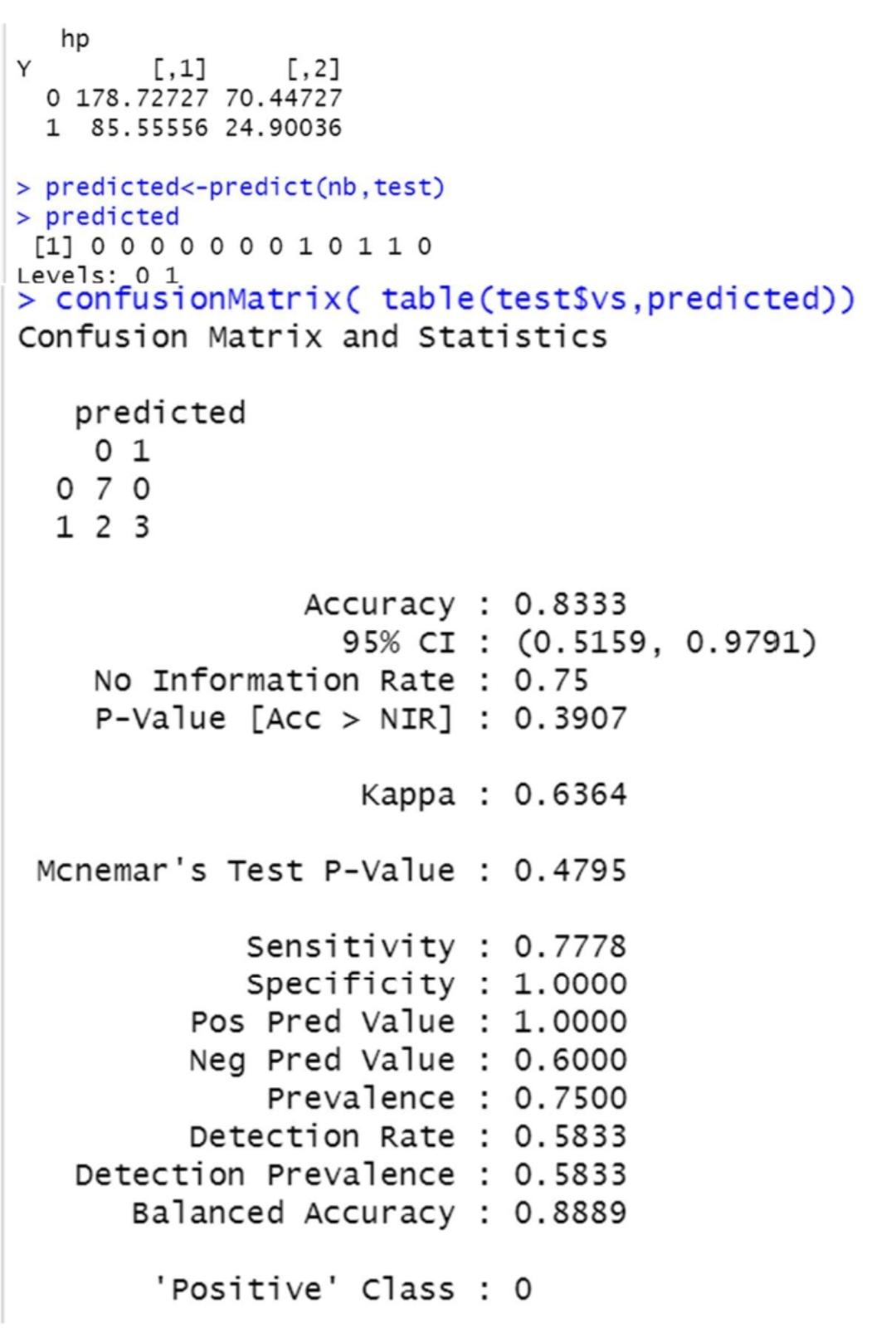
predic<-predict(naive\_bayes,TestData,type="response")

predic

predic<-ifelse(predic>0.5,1,0)

table(TestData$vs,predic)





SVM

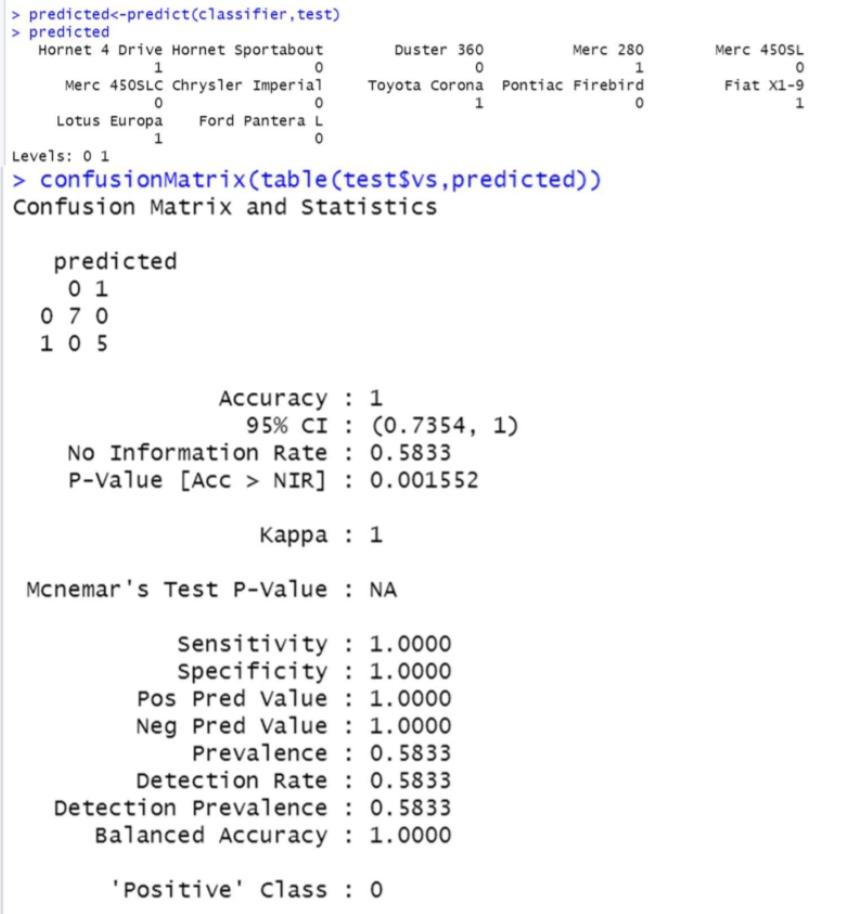
classifier = svm(formula =vs~disp+cyl+wt+mpg+hp, data = train, type = 'C-classification', kernel = 'linear')

classifier

predicted<-predict(classifier,test)

predicted

confusionMatrix(table(test$vs,predicted))



RANDOM FOREST

RF = randomForest(x = train[-8], y = train$vs, ntree = 500)

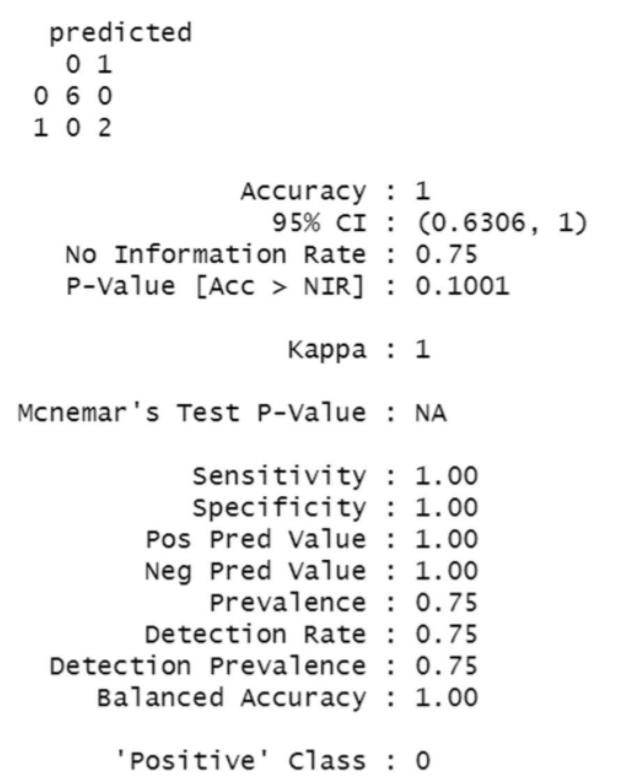
RF

predicted = predict(RF, newdata = test[-8])

predicted

predicted<-ifelse(predicted>0.5,1,0)

confusionMatrix(table(test[8],predicted))



1. Compare each of the above models using the following parameters

a. Accuracy

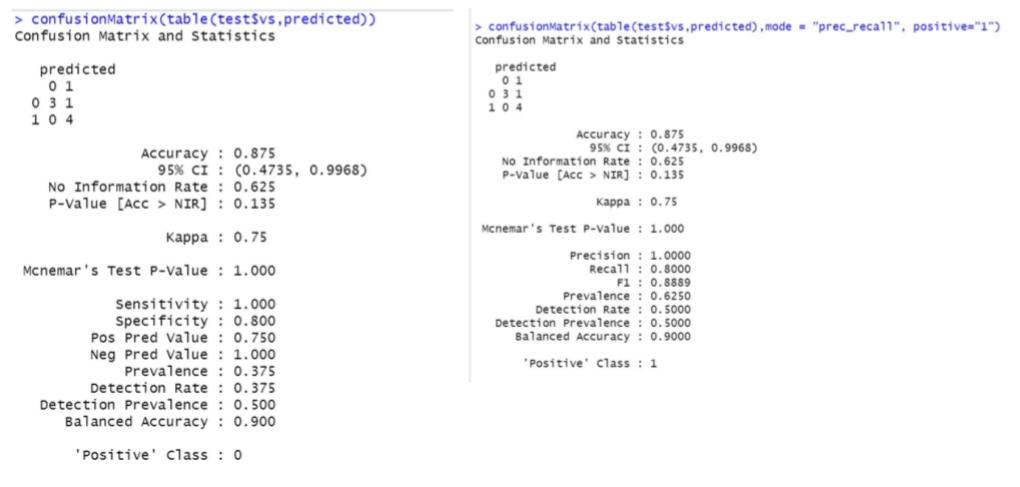
b. Precision

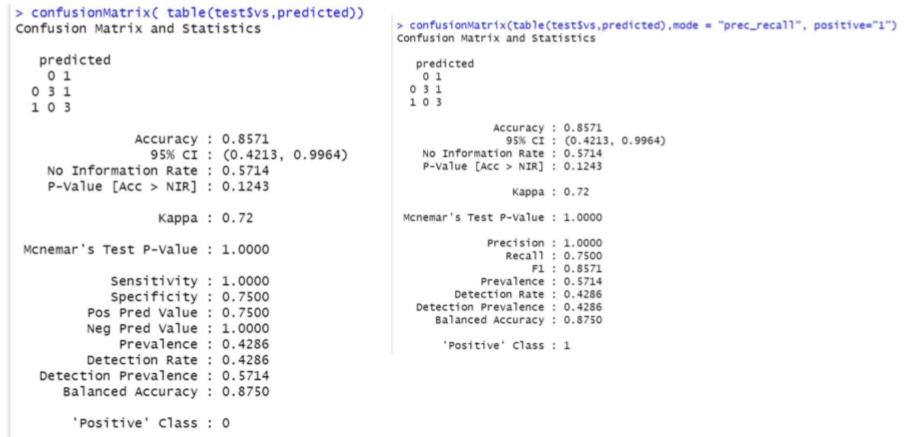
c. Recall

d. Sensitivity

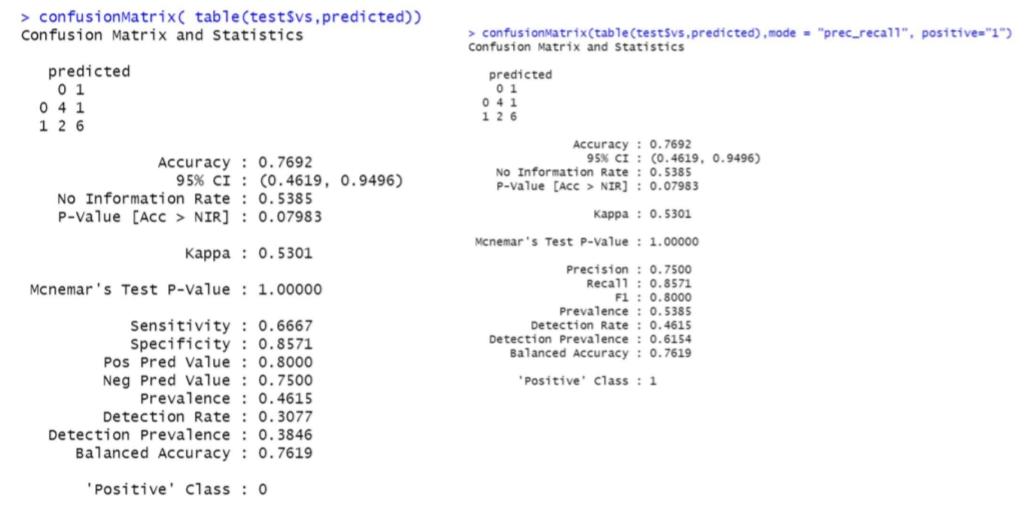
e. Specificity

LOGISTIC REGRESSION

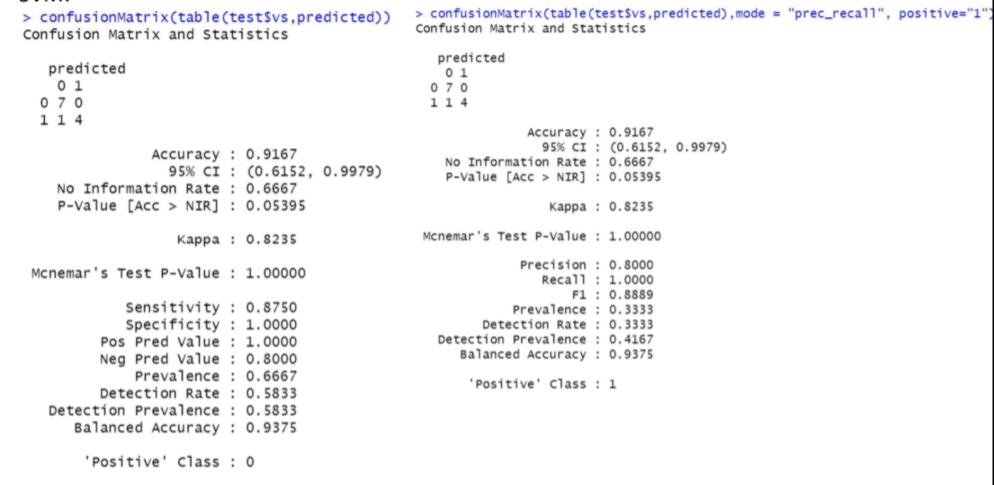


DECISION TREE

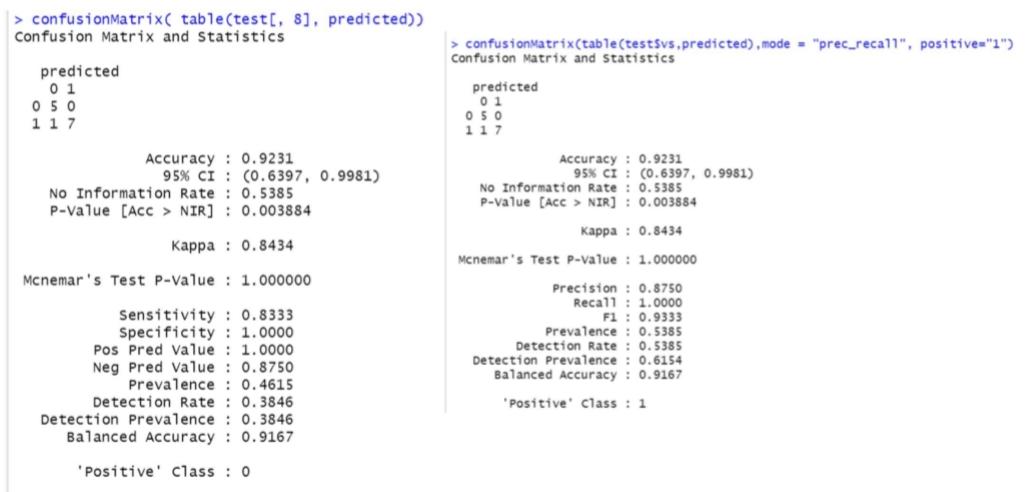
NAIVE BAYES



SVM



RANDOM FOREST



3.Perform k-means clustering in IRIS dataset

iris\_1 <- iris[, -5]

set.seed(240)

kmeans.re <- kmeans(iris\_1, centers = 3, nstart = 20)

kmeans.re

kmeans.re$cluster

# Confusion Matrix

cm <- table(iris$Species, kmeans.re$cluster)

cm

confusionMatrix(table(iris$Species, kmeans.re$cluster))

# Model Evaluation and visualization

plot(iris\_1[c("Sepal.Length", "Sepal.Width")])

plot(iris\_1[c("Sepal.Length", "Sepal.Width")],

col = kmeans.re$cluster)

plot(iris\_1[c("Sepal.Length", "Sepal.Width")],

col = kmeans.re$cluster,

main = "K-means with 3 clusters")

## Plotiing cluster centers

kmeans.re$centers

kmeans.re$centers[, c("Sepal.Length", "Sepal.Width")]

# cex is font size, pch is symbol

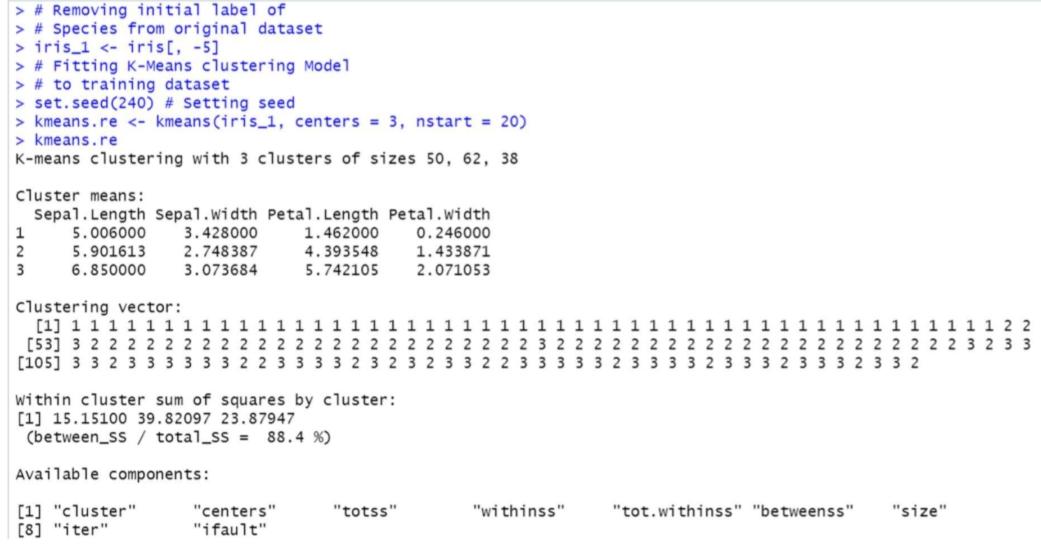
points(kmeans.re$centers[, c("Sepal.Length", "Sepal.Width")],

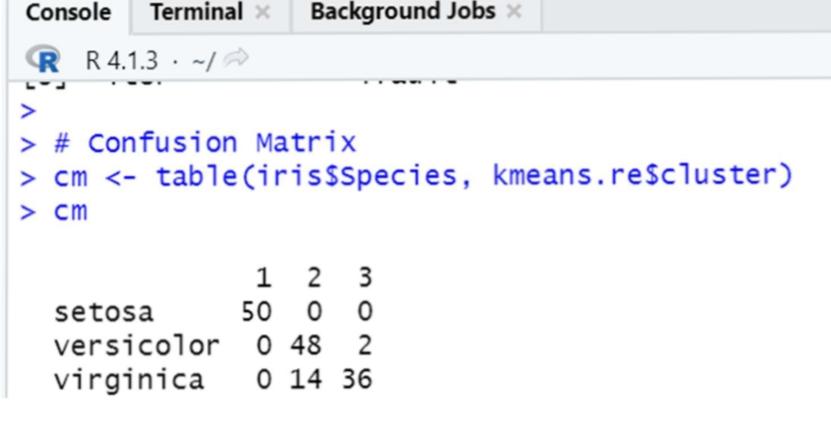
col = 1:3, pch = 8, cex = 3)

## Visualizing clusters

y\_kmeans <- kmeans.re$cluster

clusplot(iris\_1[, c("Sepal.Length", "Sepal.Width")], y\_kmeans, lines = 0, shade = TRUE, color = TRUE, labels = 2, plotchar = FALSE, span = TRUE, main = paste("Cluster iris"),xlab = 'Sepal.Length', ylab = 'Sepal.Width')





4.Perform Hierarchical clustering in mtcars dataset

distance\_mat <- dist(mtcars, method = 'euclidean')

distance\_mat

set.seed(240) # Setting seed

Hierar\_cl <- hclust(distance\_mat, method = "average")

Hierar\_cl

# Plotting dendrogram

plot(Hierar\_cl)

# Choosing no. of clusters

# Cutting tree by height

abline(h = 110, col = "green")

# Cutting tree by no. of clusters

fit <- cutree(Hierar\_cl, k = 3 )

fit

table(fit)

rect.hclust(Hierar\_cl, k = 3, border = "green")

