cat("\f") # Clear the console

rm(list = ls()) # Clear the working environment

############################ Load the required packages ########################################

library(ISLR)

library(e1071)

library(pROC)

library(dplyr)

########################## Load the data #######################################################

iris <- iris

glimpse(iris) # Metadata

str(iris)

head(iris) # See the first 6 values of the dataset

sum(is.na(iris)) # Check for NA's

attach(iris) #Optional Step

######################## Subset the data ####################################################

Random.seed <- c('Mersenne-Twister', 490)

set.seed(490) # Set seed for replication

index <- sample(1:nrow(iris),0.5 \* nrow(iris), replace = FALSE) # Create an index to split the data

# iris\_train <- iris[index,]

# iris\_test <- iris[-index,]

iris\_train\_x <- iris[index, 1:4]

iris\_test\_x <- iris[-index, 1:4]

iris\_train\_y <- iris[index, 5]

iris\_test\_y <- iris[-index, 5]

**#### \*\*\*\* For the TA: I need to understand why we chose to subset the dataset like we did. I have never**

**#### subsetted data like this and am struggling to see the necessity or requirement to do the split**

**#### like it is done in the lines. Is it done to adhere to the arguments for naiveBayes() only? \*\*\*\***

# table(Species)

model <- naiveBayes(iris\_train\_x, iris\_train\_y) # Apply the Naive Bayes Estimator

## Predictions for Training set

pred\_train <- predict(model,iris\_train\_x, type = "class")

cfm\_train <- table(pred\_train, iris\_train\_y)

cfm\_train

## Predictions for Testing set

pred\_test <- predict(model,iris\_test\_x, type = "class")

cfm\_test <- table(pred\_test, iris\_test\_y)

cfm\_test

## Predictions for the entire set

pred\_all <- predict(model, iris[1:4], type = "class")

cfm\_all <- table(pred\_all, Species)

cfm\_all

accuracy\_train <- sum(diag(cfm\_train))/sum(cfm\_train) # Training Accuracy

accuracy\_train

error\_train <- 1 - accuracy\_train # Training Error

error\_train

accuracy\_test <- sum(diag(cfm\_test))/sum(cfm\_test) # Testing Accuracy

accuracy\_test

error\_test <- 1 - accuracy\_test # Testing Error

error\_test

accuracy\_all <- sum(diag(cfm\_all))/sum(cfm\_all) # Overall Error

accuracy\_all

error\_all <- 1 - accuracy\_all # Overall Accuracy

error\_all

########################### Calculate Sensitivity, Specificity, PPV & NPV ##############################

########################################## TRAINING SET #################################################################

############################################# SETOSA ####################################################################

macro\_spec <- 0

macro\_sens <- 0

train\_sens <- cfm\_train["setosa","setosa"]/sum(cfm\_train[,'setosa'])

train\_sens

train\_spec <- sum(diag(cfm\_train[2:3,2:3]))/(sum(diag(cfm\_train[2:3,2:3])) + sum(cfm\_train['setosa',2:3]))

train\_spec

ppv\_train <- cfm\_train['setosa','setosa']/sum(cfm\_train['setosa',])

ppv\_train

npv\_train <- sum(diag(cfm\_train[2:3,2:3]))/(sum(diag(cfm\_train[2:3,2:3])) + sum(cfm\_train[2:3,'setosa']))

npv\_train

macro\_sens <- macro\_sens + train\_sens

macro\_spec <- macro\_spec + train\_spec

############################################# VERSICOLOR ###################################################################

train\_sens <- cfm\_train["versicolor","versicolor"]/sum(cfm\_train[,'versicolor'])

train\_sens

train\_spec <- sum(diag(cfm\_train[c(1,3),c(1,3)]))/(sum(diag(cfm\_train[c(1,3),c(1,3)])) + sum(cfm\_train['versicolor',c(1,3)]))

train\_spec

ppv\_train <- cfm\_train['versicolor','versicolor']/sum(cfm\_train['versicolor',])

ppv\_train

npv\_train <- sum(diag(cfm\_train[c(1,3),c(1,3)]))/(sum(diag(cfm\_train[c(1,3),c(1,3)])) + sum(cfm\_train[c(1,3),'versicolor']))

npv\_train

macro\_sens <- macro\_sens + train\_sens

macro\_spec <- macro\_spec + train\_spec

############################################# VIRGINICA ###################################################################

train\_sens <- cfm\_train["virginica","virginica"]/sum(cfm\_train[,'virginica'])

train\_sens

train\_spec <- sum(diag(cfm\_train[1:2,1:2]))/(sum(diag(cfm\_train[1:2,1:2])) + sum(cfm\_train['virginica',1:2]))

train\_spec

ppv\_train <- cfm\_train['virginica','virginica']/sum(cfm\_train['virginica',])

ppv\_train

npv\_train <- sum(diag(cfm\_train[1:2,1:2]))/(sum(diag(cfm\_train[1:2,1:2])) + sum(cfm\_train[1:2,'virginica']))

npv\_train

macro\_sens <- macro\_sens + train\_sens

macro\_spec <- macro\_spec + train\_spec

macro\_sens <- macro\_sens/3

macro\_spec <- macro\_spec/3

macro\_sens

macro\_spec

########################################## TESTING SET #################################################################

############################################# SETOSA ####################################################################

macro\_spec <- 0

macro\_sens <- 0

test\_sens <- cfm\_test["setosa","setosa"]/sum(cfm\_test[,'setosa'])

test\_sens

test\_spec <- sum(diag(cfm\_test[2:3,2:3]))/(sum(diag(cfm\_test[2:3,2:3])) + sum(cfm\_test['setosa',2:3]))

test\_spec

ppv\_test <- cfm\_test['setosa','setosa']/sum(cfm\_test['setosa',])

ppv\_test

npv\_test <- sum(diag(cfm\_test[2:3,2:3]))/(sum(diag(cfm\_test[2:3,2:3])) + sum(cfm\_test[2:3,'setosa']))

npv\_test

macro\_sens <- macro\_sens + test\_sens

macro\_spec <- macro\_spec + test\_spec

############################################# VERSICOLOR ###################################################################

test\_sens <- cfm\_test["versicolor","versicolor"]/sum(cfm\_test[,'versicolor'])

test\_sens

test\_spec <- sum(diag(cfm\_test[c(1,3),c(1,3)]))/(sum(diag(cfm\_test[c(1,3),c(1,3)])) + sum(cfm\_test['versicolor',c(1,3)]))

test\_spec

ppv\_test <- cfm\_test['versicolor','versicolor']/sum(cfm\_test['versicolor',])

ppv\_test

npv\_test <- sum(diag(cfm\_test[c(1,3),c(1,3)]))/(sum(diag(cfm\_test[c(1,3),c(1,3)])) + sum(cfm\_test[c(1,3),'versicolor']))

npv\_test

macro\_sens <- macro\_sens + train\_sens

macro\_spec <- macro\_spec + train\_spec

############################################# VIRGINICA ###################################################################

test\_sens <- cfm\_test["virginica","virginica"]/sum(cfm\_test[,'virginica'])

test\_sens

test\_spec <- sum(diag(cfm\_test[1:2,1:2]))/(sum(diag(cfm\_test[1:2,1:2])) + sum(cfm\_test['virginica',1:2]))

test\_spec

ppv\_test <- cfm\_test['virginica','virginica']/sum(cfm\_test['virginica',])

ppv\_test

npv\_test <- sum(diag(cfm\_test[1:2,1:2]))/(sum(diag(cfm\_test[1:2,1:2])) + sum(cfm\_test[1:2,'virginica']))

npv\_test

macro\_sens <- macro\_sens + train\_sens

macro\_spec <- macro\_spec + train\_spec

macro\_sens <- macro\_sens/3

macro\_spec <- macro\_spec/3

macro\_sens

macro\_spec

########################################## FULL DATA SET #################################################################

############################################# SETOSA ####################################################################

macro\_spec <- 0

macro\_sens <- 0

sens <- cfm\_all["setosa","setosa"]/sum(cfm\_all[,'setosa'])

sens

spec <- sum(diag(cfm\_all[2:3,2:3]))/(sum(diag(cfm\_all[2:3,2:3])) + sum(cfm\_all['setosa',2:3]))

spec

ppv <- cfm\_all['setosa','setosa']/sum(cfm\_all['setosa',])

ppv

npv <- sum(diag(cfm\_all[2:3,2:3]))/(sum(diag(cfm\_all[2:3,2:3])) + sum(cfm\_all[2:3,'setosa']))

npv

macro\_sens <- macro\_sens + sens

macro\_spec <- macro\_spec + spec

############################################# VERSICOLOR ###################################################################

sens <- cfm\_all["versicolor","versicolor"]/sum(cfm\_all[,'versicolor'])

sens

spec <- sum(diag(cfm\_all[c(1,3),c(1,3)]))/(sum(diag(cfm\_all[c(1,3),c(1,3)])) + sum(cfm\_all['versicolor',c(1,3)]))

spec

ppv <- cfm\_all['versicolor','versicolor']/sum(cfm\_all['versicolor',])

ppv

npv <- sum(diag(cfm\_all[c(1,3),c(1,3)]))/(sum(diag(cfm\_all[c(1,3),c(1,3)])) + sum(cfm\_all[c(1,3),'versicolor']))

npv

macro\_sens <- macro\_sens + sens

macro\_spec <- macro\_spec + spec

############################################# VIRGINICA ###################################################################

sens <- cfm\_all["virginica","virginica"]/sum(cfm\_all[,'virginica'])

sens

spec <- sum(diag(cfm\_all[1:2,1:2]))/(sum(diag(cfm\_all[1:2,1:2])) + sum(cfm\_all['virginica',1:2]))

spec

ppv <- cfm\_all['virginica','virginica']/sum(cfm\_all['virginica',])

ppv

npv <- sum(diag(cfm\_all[1:2,1:2]))/(sum(diag(cfm\_all[1:2,1:2])) + sum(cfm\_all[1:2,'virginica']))

npv

macro\_sens <- macro\_sens + sens

macro\_spec <- macro\_spec + spec

macro\_sens <- macro\_sens/3

macro\_spec <- macro\_spec/3

macro\_sens

macro\_spec

##################################### ROC CURVE ###########################################################################

prob <- predict(model,iris\_test\_x, type = "raw")

setosa\_labels <- rep(0, length(iris\_test\_y))

versicolor\_labels <- rep(0, length(iris\_test\_y))

virginica\_labels <- rep(0, length(iris\_test\_y))

for(f in 1:length(iris\_test\_y)){

if(iris\_test\_y[f] == "setosa"){

setosa\_labels[f] <- 1

} else if(iris\_test\_y[f] == "versicolor"){

versicolor\_labels[f] <- 1

} else if(iris\_test\_y[f] == "virginica"){

virginica\_labels[f] <- 1

}

}

setosa\_roc <- roc(setosa\_labels, prob[,'setosa'], auc.polygon = TRUE, max.auc.polygon = TRUE, print.auc = TRUE,show.thres = TRUE)

setosa\_smooth\_roc <- smooth(setosa\_roc, method = "density")

plot(setosa\_smooth\_roc, col = 'red', xaxt='n', xlab="False Positive Rate (1 - Specificity)", ylab = "True Positive Rate (Sensitivity)")

par(new = TRUE)

virginica\_roc <- roc(virginica\_labels, prob[,'virginica'], auc.polygon = TRUE, max.auc.polygon = TRUE, print.auc = TRUE,show.thres = TRUE)

virginica\_smooth\_roc <- smooth(virginica\_roc, method = "density")

plot(virginica\_smooth\_roc, col = 'green', xaxt='n', xlab="", ylab = "")

par(new = TRUE)

versicolor\_roc <- roc(versicolor\_labels, prob[,'versicolor'], auc.polygon = TRUE, max.auc.polygon = TRUE, print.auc = TRUE,show.thres = TRUE)

versicolor\_smooth\_roc <- smooth(versicolor\_roc, method = "density")

plot(versicolor\_smooth\_roc, col = 'blue', xaxt='n', xlab="", ylab = "")

y\_labels <- c(setosa\_labels,versicolor\_labels,virginica\_labels)

y\_probs <- c(prob[,"setosa"],prob[,"versicolor"],prob[,"virginica"])

par(new = TRUE)

micro\_roc <- roc(y\_labels,y\_probs,auc.polygon = TRUE, max.auc.polygon = TRUE, print.auc = TRUE, show.thres = TRUE)

micro\_smooth\_roc <- smooth(micro\_roc, method = "density")

plot(micro\_smooth\_roc, col = 'black', lty = 'dotdash', xaxt='n', xlab="", ylab = "")

par(new=TRUE)

macro\_sensitivity <- (setosa\_smooth\_roc$sensitivities + versicolor\_smooth\_roc$sensitivities + virginica\_smooth\_roc$sensitivities) / 3

macro\_specificity <- (setosa\_smooth\_roc$specificities + versicolor\_smooth\_roc$specificities + virginica\_smooth\_roc$specificities) / 3

lines(macro\_specificity, macro\_sensitivity, type='l', xlim = rev(range(macro\_specificity)), col='magenta', lty=4)

axis(1, at=(5:0) \* 0.2, labels=(0:5) \* 0.2, pos=c(-0.04,0))

legend(0.4, 0.5, legend = c('setosa','versicolor', 'virginica', 'micro-avg', 'macro-avg'), col = c('red', 'blue', 'green', 'black', 'magenta'), lty = c(1,1,1,4,4))

Chart

Description automatically generated

############################################ AUC ###################################################################################

probs\_train <- predict(model, iris\_train\_x, type = "raw") # Training AUC

multiclass.roc(iris\_train\_y, probs\_train)

multiclass.roc(iris\_test\_y, prob) # Testing AUC

probs\_all <- predict(model, iris[1:4], type = "raw") # Overall AUC

multiclass.roc(Species, probs\_all)

**Results** -> The model identifies and classifies the species accordingly with a high rate of certainty, especially for Setosa.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Train | Test | All |
| Accuracy | | 0.946 | 0.946 | 0.946 |
| AUC | | 0.99 | 0.99 | 0.99 |
| Macro Sensitivity | | 0.95 | 0.92 | 0.94 |
| Macro Specificity | | 0.97 | 0.98 | 0.97 |
| Sensitivity | Setosa | 1.00 | 1.00 | 1.00 |
| Versicolor | 0.95 | 0.84 | 0.90 |
| Virginica | 0.89 | 1.00 | 0.94 |
| Specificity | Setosa | 1.00 | 1.00 | 1.00 |
| Versicolor | 0.94 | 1.00 | 0.97 |
| Virginica | 0.97 | 0.92 | 0.95 |
| PPV | Setosa | 1.00 | 1.00 | 1.00 |
| Versicolor | 0.88 | 1.00 | 0.93 |
| Virginica | 0.96 | 0.84 | 0.90 |
| NPV | Setosa | 1.00 | 1.00 | 1.00 |
| Versicolor | 0.97 | 0.92 | 0.95 |
| Virginica | 0.93 | 1.00 | 0.96 |