NAME: Hammad Ansari **ROLL NO: 2018450002**

Experiment No. 8

Title: To Implement Support Vector Machine in R

Problem:

SVM on Heart Disease Dataset

Steps are as follows:

Code:

1) Importing libraries

```
require(ggplot2)

require(pROC) #to plot the ROC curves
```

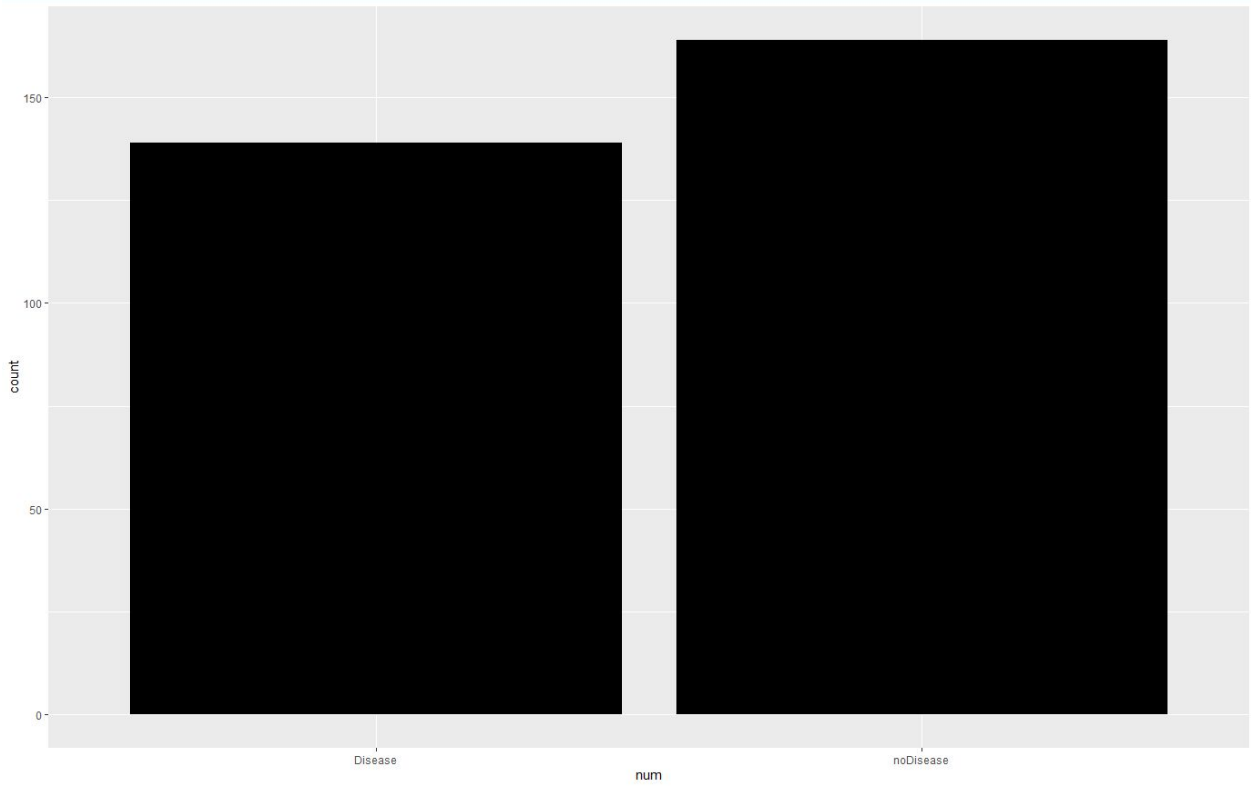
2) Loading dataset

```
heartdf <-
read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data", header = FALSE, sep = ",", na.strings = '?')

names(heartdf) <- c("age", "sex", "cp", "trestbps", "chol", "fbs", "restecg",
                    "thalach", "exang", "oldpeak", "slope", "ca", "thal", "num")

attach(heartdf)

head(heartdf, 3)

dim(heartdf) # dimensions of the data set
```

```
> head(heartdf, 3)
  age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal num
1  63   1  1      145  233   1       2     150     0     2.3     3  0    6   0
2  67   1  4      160  286   0       2     108     1     1.5     2  3    3   2
3  67   1  4      120  229   0       2     129     1     2.6     2  2    7   1

> dim(heartdf) # dimensions of the data set
[1] 303  14
```

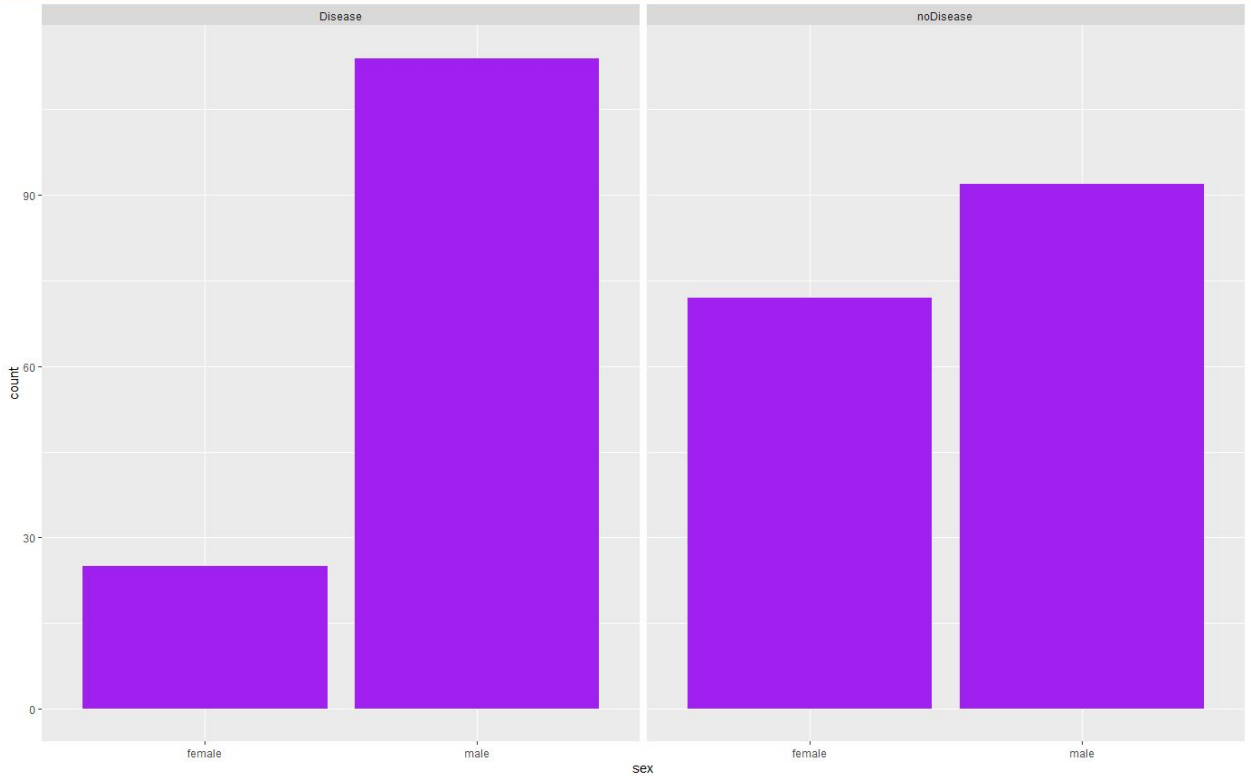
3) Conversion of variables

```
#converting the NUM variable to binary class variable

heartdf$num <- ifelse(heartdf$num > 0, "Disease", "No Disease")

table(heartdf$num)

#distribution of the target variable

ggplot(heartdf, aes(x = num)) +

 geom_bar(fill = "black")
```




```
#converting to factor variable

heartdf$sex <- ifelse(heartdf$sex == 0, "female", "male")

table(heartdf$sex)

table(sex = heartdf$sex, disease = heartdf$num)

ggplot(heartdf, aes(x = sex)) +
```

```
 geom_bar(fill = "purple") +
 facet_wrap(~num)
```




```
#heart disease and age
#making a box plot to understand the statistical distribution
by(heartdf$age, heartdf$num, summary)

ggplot(heartdf, aes(x = num, y = age)) +
  geom_boxplot()

cor.test(age, chol) #very low correlation
```

4) Confusion matrix of chest pain and heart disease

table(cp, num)

```
> #confusion matrix of chest pain and heart disease
> table(cp, num)
   num
cp   0  1  2  3  4
  1 16  5  1  0  1
  2 41  6  1  2  0
  3 68  9  4  4  1
  4 39 35 30 29 11

> #confusuon matrix of exersice induced asthama and heart disease
> table(exang, num)
     num
exang   0   1   2   3   4
    0 141  30  14  12   7
    1  23  25  22  23   6

> cor.test(age, thalach)

        Pearson's product-moment correlation

data:  age and thalach
t = -7.4329, df = 301, p-value = 1.109e-12
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.4849644 -0.2941816
sample estimates:
       cor
-0.3938058
```

#confusuon matrix of exersice induced asthama and heart disease

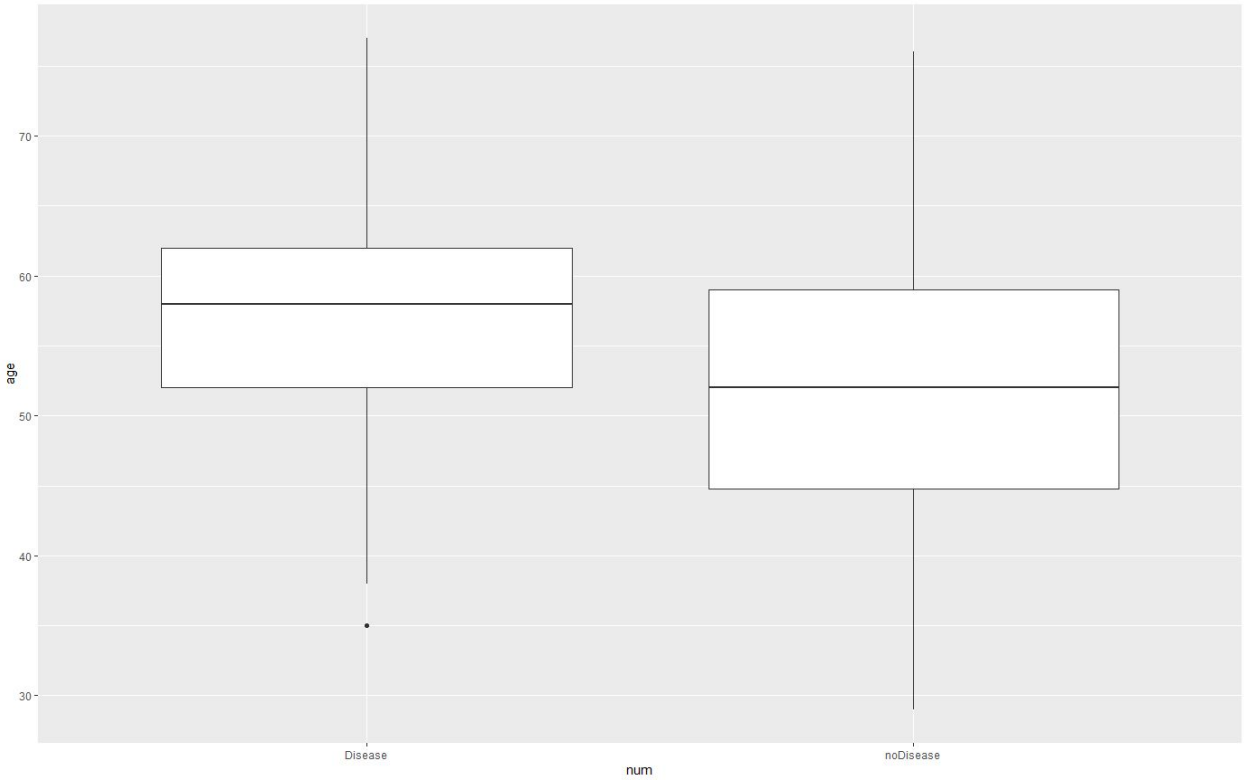
table(exang, num)

cor.test(age, thalach)

ggplot(heartdf, aes(x = age, y = thalach)) +

geom_point() +

geom_smooth()



5) Creating training and testing dataset

```
library(caret)

set.seed(20)

inTrainRows <- createDataPartition(heartdf$num, p = 0.7, list = FALSE)

trainData <- heartdf[inTrainRows,]

testData <- heartdf[-inTrainRows,]

nrow(trainData) / (nrow(testData) + nrow(trainData)) #checking whether really 70% -> OK


# for this to work add names to all levels (numbers not allowed)

feature.names = names(heartdf)

for (f in feature.names) {
```

```
  if (class(heartdf[[f]]) == "factor") {
    levels <- unique(c(heartdf[[f]]))
    heartdf[[f]] <- factor(heartdf[[f]],
                           labels = make.names(levels))
  }
}
```

```
#converting to factor variable with 2 levels
heartdf$num <- as.factor(heartdf$num)
levels(heartdf$num) <- c("Notdisease", "Disease")
table(heartdf$num)
```

```
> nrow(trainData) / (nrow(testData) + nrow(trainData)) #checking whether really 70% -> OK
[1] 0.7029703

> # for this to work add names to all levels (numbers not allowed)
> feature.names = names(heartdf)

> for (f in feature.names) {
+   if (class(heartdf[[f]]) == "factor") {
+     levels <- unique(c(heartdf[[f]]))
+     heartdf[[f]] <- factor(heartdf[[ .... [TRUNCATED]

> #converting to factor variable with 2 levels
> heartdf$num <- as.factor(heartdf$num)

> levels(heartdf$num) <- c("Notdisease", "Disease")

> table(heartdf$num)

Notdisease    Disease
       139        164
```

```
set.seed(10)
inTrainRows <- createDataPartition(heartdf$num, p = 0.7, list = FALSE)
trainData2 <- heartdf[inTrainRows,]
testData2 <- heartdf[-inTrainRows,]
```

6) Creating SVM Model

```
#cross validation
fitControl <- trainControl(method = "repeatedcv",
                           number = 10,
                           repeats = 10,
## Estimate class probabilities
                           classProbs = TRUE,
## Evaluate performance using
## the following function
                           summaryFunction = twoClassSummary)


svmModel <- train(num ~ ., data = na.omit(trainData2),
                  method = "svmRadial",
                  trControl = fitControl,
                  preProcess = c("center", "scale"),
                  tuneLength = 8,
                  metric = "ROC")


svmModel
```

```
> svmModel
Support Vector Machines with Radial Basis Function Kernel

208 samples
 13 predictor
  2 classes: 'Notdisease', 'Disease'

Pre-processing: centered (13), scaled (13)
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 186, 187, 186, 188, 187, 187, ...
Resampling results across tuning parameters:

  C      ROC        Sens       Spec
   0.25  0.9060093  0.7917778  0.8785606
   0.50  0.9056490  0.7936667  0.8669697
   1.00  0.8982500  0.7894444  0.8544697
   2.00  0.8965909  0.7858889  0.8340909
   4.00  0.8876911  0.7772222  0.8295455
   8.00  0.8681313  0.7664444  0.8153788
  16.00  0.8479596  0.7347778  0.7888636
  32.00  0.8297997  0.7396667  0.7746970

Tuning parameter 'sigma' was held constant at a value of 0.05200353
ROC was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.05200353 and C = 0.25.
```

7) Prediction on test data-class labels

svmPrediction <- predict(svmModel, testData2)

#probability of no heart disease-finding probabilities value

svmPredictionprob <- predict(svmModel, testData2, type = 'prob')[2]

```
> svmPredictionprob
      Disease
1  0.51991705
2  0.04393355
3  0.01200108
4  0.46507916
5  0.96314802
6  0.06644721
7  0.20348437
8  0.68682237
9  0.89725195
10 0.86429672
11 0.94654034
12 0.88678906
13 0.15005658
14 0.93303157
15 0.63116135
16 0.82486273
17 0.13848293
18 0.92465390
19 0.10442217
20 0.17792405
21 0.51490174
22 0.77838358
23 0.69962112
24 0.57976342
25 0.96444390
26 0.84687615
```

8) Generating a confusion matrix

ConfMatrixPrediction <- confusionMatrix(svmPrediction, na.omit(testData2)$num)

ConfMatrixPrediction$table

```
> #prediction on test data-class labels
> svmPrediction <- predict(svmModel, testData2)

> #probability of no heart disease-finding probabilities value
> svmPredictionprob <- predict(svmModel, testData2, type = 'prob')[2]

> #generating a confusion matrix
> ConfMatrixPrediction <- confusionMatrix(svmPrediction, na.omit(testData2)$num)

> ConfMatrixPrediction$table
            Reference
Prediction   Notdisease Disease
  Notdisease         33       7
  Disease             8      41
```

9) Recursive Object Characteristic and Area Under the Curve

ROC and AUC value

AUC <- roc(na.omit(testData2)$num,
as.numeric(as.matrix((svmPredictionprob))))$auc

Accuracy <- ConfMatrixPrediction$overall['Accuracy']

svmPerformance <- cbind(AUC, Accuracy)

svmPerformance

auc_roc <- roc(na.omit(testData2)$num,
as.numeric(as.matrix((svmPredictionprob))))

plot(auc_roc)

NAME: Hammad Ansari **ROLL NO: 2018450002**