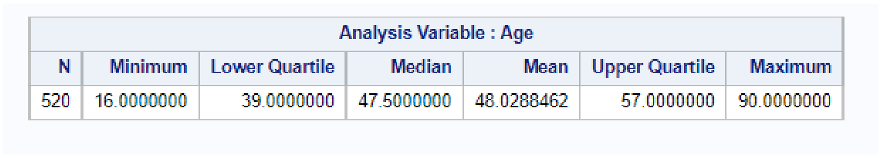
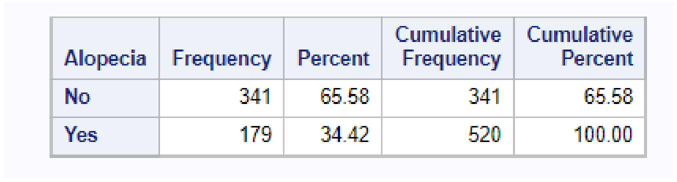
APPENDICES:

Appendix A: Exploratory Data Analysis



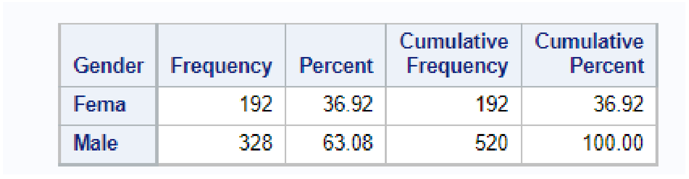
Analysis Variable : Age						
N	Minimum	Lower Quartile	Median	Mean	Upper Quartile	Maximum
520	16.0000000	39.0000000	47.5000000	48.0288462	57.0000000	90.0000000

***Table A-1**: This table includes the minimum, maximum, 1st Quartile, median, mean, and 3rd Quartile values for the numerical variable, age of patients.*



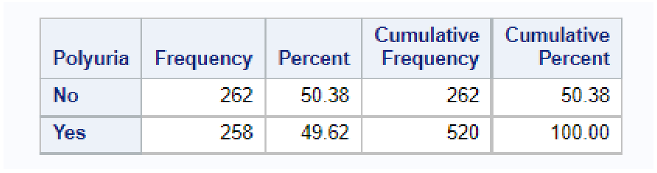
Alopecia	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	341	65.58	341	65.58
Yes	179	34.42	520	100.00

***Table A-2**: This table displays the counts of patients that experience Alopecia symptom (loss of hair).*



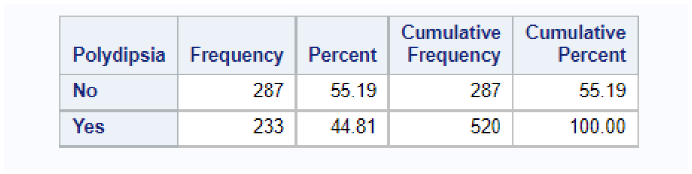
Gender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Fema	192	36.92	192	36.92
Male	328	63.08	520	100.00

***Table A-3**: This table displays the counts of the gender of patients.*



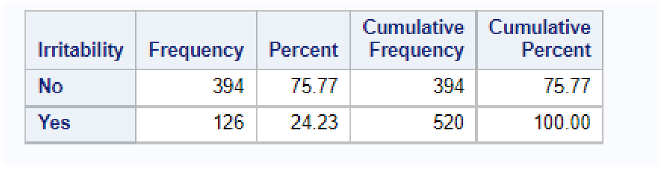
Polyuria	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	262	50.38	262	50.38
Yes	258	49.62	520	100.00

***Table A-4**: This table displays the counts of patients who have polyuria (excessive/frequent urination).*



Polydipsia	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	287	55.19	287	55.19
Yes	233	44.81	520	100.00

***Table A-5**: This table displays the counts of patients that experience polydipsia (excessive/increased thirst).*



Irritability	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	394	75.77	394	75.77
Yes	126	24.23	520	100.00

***Table A-6**: This table displays the counts of patients that experience irritability and having mood swings.*

Itching	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	267	51.35	267	51.35
Yes	253	48.65	520	100.00

Table A-7*: This table displays the counts of patients that have itchy skin.*

Obesity	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	432	83.08	432	83.08
Yes	88	16.92	520	100.00

Table A-8*: This table displays the counts of patients that are obese or overweight.*

Polyphagia	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	283	54.42	283	54.42
Yes	237	45.58	520	100.00

Table A-9*: This table displays the counts of patients that experience polyphagia (extreme hunger).*

weakness	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	215	41.35	215	41.35
Yes	305	58.65	520	100.00

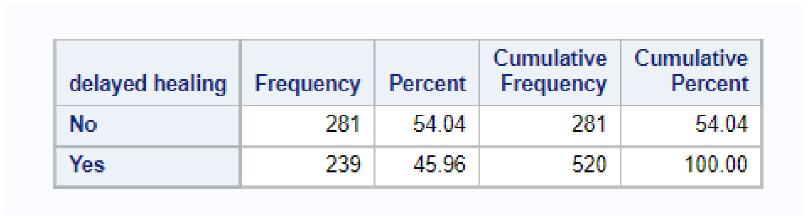
Table A-10*: This table displays the counts of patients that experience fatigue, weak, tired feeling.*

sudden weight loss	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	303	58.27	303	58.27
Yes	217	41.73	520	100.00

Table A-11*: This table displays the counts of patients that have unexplained weight loss.*

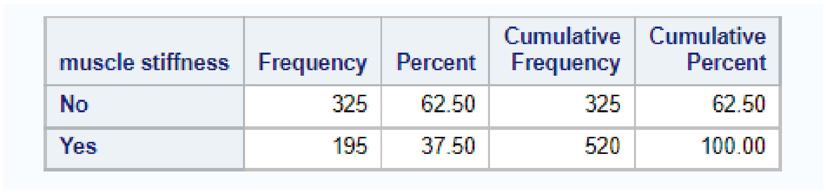
Genital thrush	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	404	77.69	404	77.69
Yes	116	22.31	520	100.00

Table A-12*: This table displays the counts of patients that have genital thrush.*



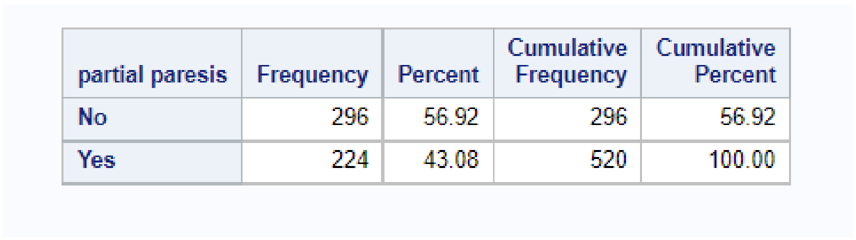
delayed healing	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	281	54.04	281	54.04
Yes	239	45.96	520	100.00

Table A-13: *This table displays the counts of patients that experience delayed wound healing.*



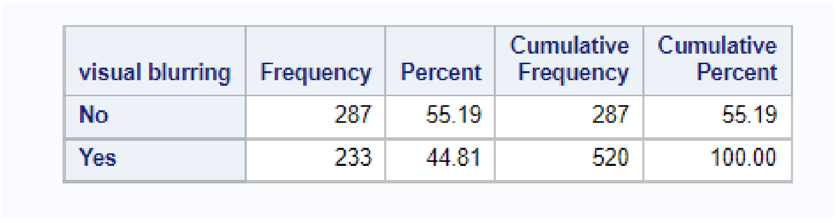
muscle stiffness	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	325	62.50	325	62.50
Yes	195	37.50	520	100.00

Table A-14: *This table displays the counts of patients who have muscle stiffness.*



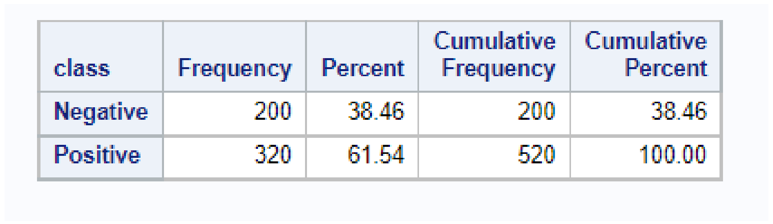
partial paresis	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	296	56.92	296	56.92
Yes	224	43.08	520	100.00

Table A-15: *This table displays the counts of patients who have partial paresis.*



visual blurring	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No	287	55.19	287	55.19
Yes	233	44.81	520	100.00

Table A-16: *This table displays the counts of patients that experience blurred vision.*



class	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Negative	200	38.46	200	38.46
Positive	320	61.54	520	100.00

Table A-17: *This table displays the counts of the response variable, class (Negative = Patients will be at negative risk for diabetes; Positive = Patients will be at Positive risk for diabetes)*

Appendix B: Statistical analysis / Modeling

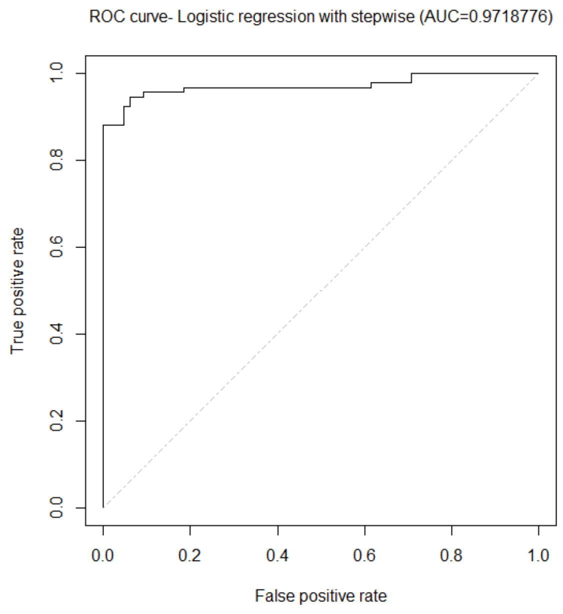


Figure B-1: *This Receiver Operating Characteristics (ROC) curve for the testing set fitted with conventional logistic regression after stepwise selection has an area under the curve of 0.9718776 in which 0.5 is used as the classification cutoff.*

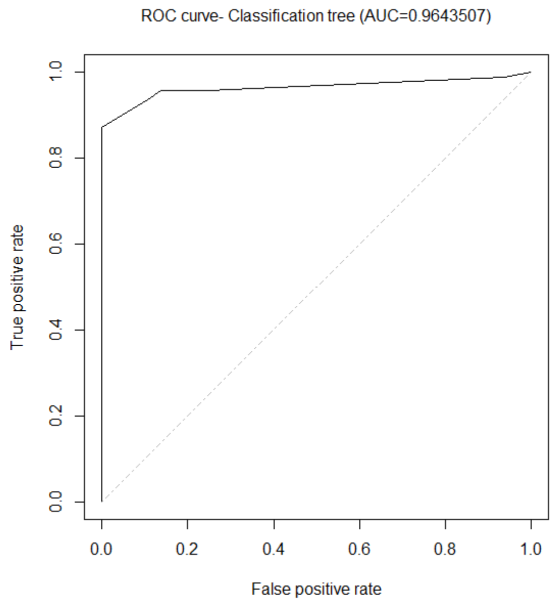


Figure B-2: *This Receiver Operating Characteristics (ROC) curve for the testing set fitted with classification tree has an area under the curve of 0.9643507.*

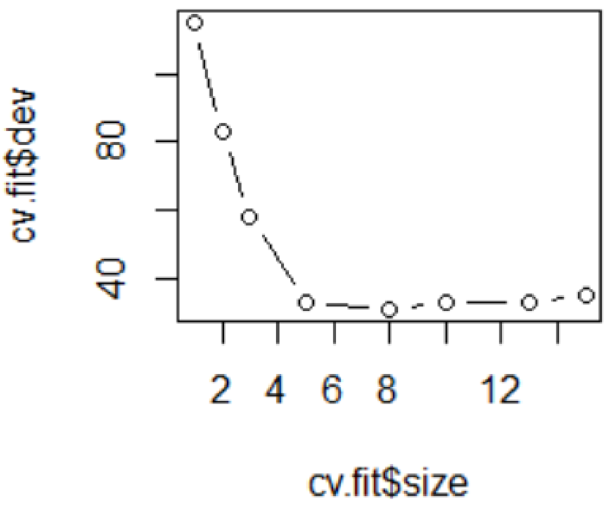


Figure B-3: *This is the plot of tree sizes versus their corresponding deviances in the cross validation process in order to select the optimal tree for tree pruning. The best subtree size is 8 since it has the smallest cross validation deviance.*

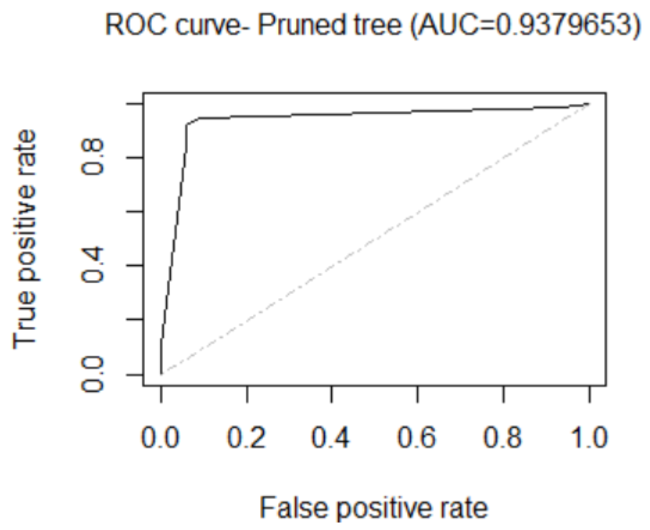


Figure B-4: *This Receiver Operating Characteristics (ROC) curve for the testing set fitted with classification tree after pruning has an area under the curve of 0.9379653.*

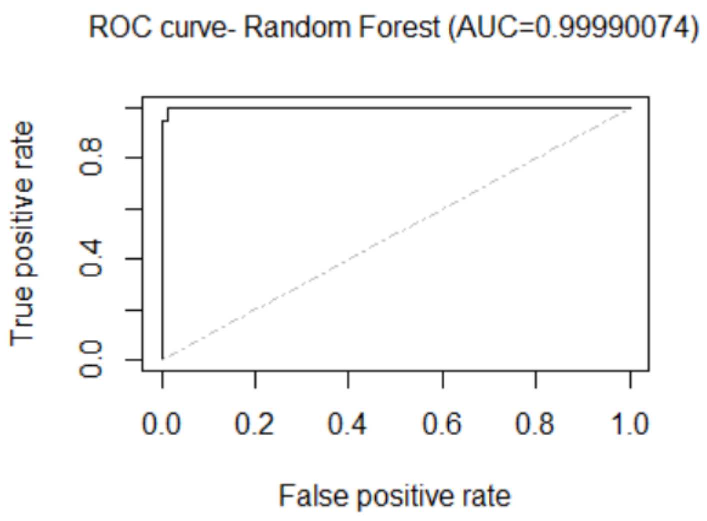


Figure B-5*: This Receiver Operating Characteristics (ROC) curve for the testing set fitted with random forest has an area under the curve of 0.999990074 which is close to 1.*

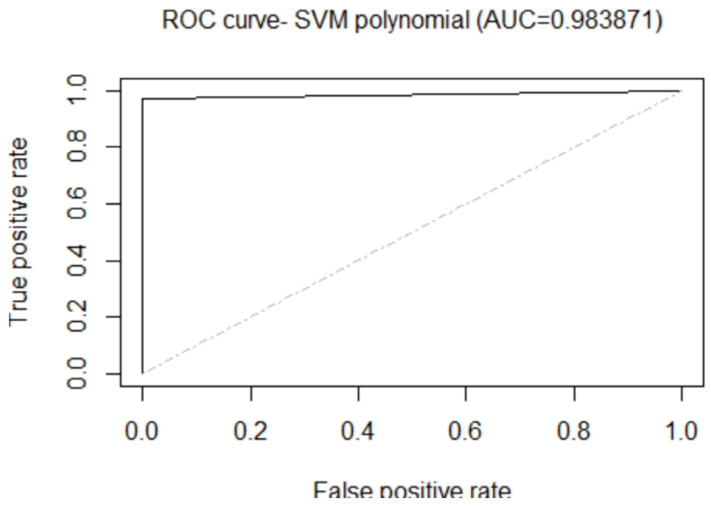


Figure B-6*: This Receiver Operating Characteristics (ROC) curve for the testing set fitted with the support vector machine using the polynomial kernel has an area under the curve of 0.983871.*

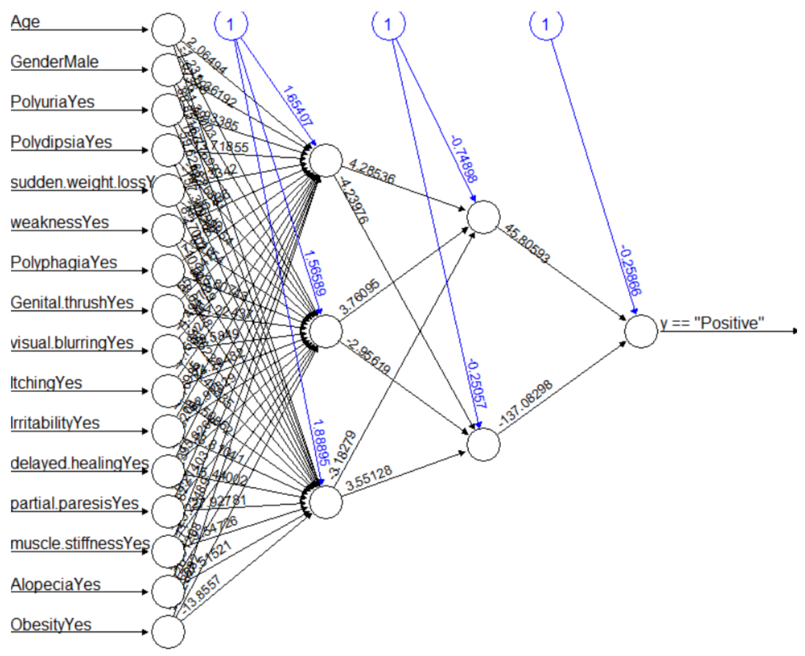


Figure B-7: This is a neural network plot that consists of the input layer, 2 hidden layers of which the first hidden layer has 3 nodes while the second hidden layer has 2 nodes, as well as a final output layer which is the predicted class.

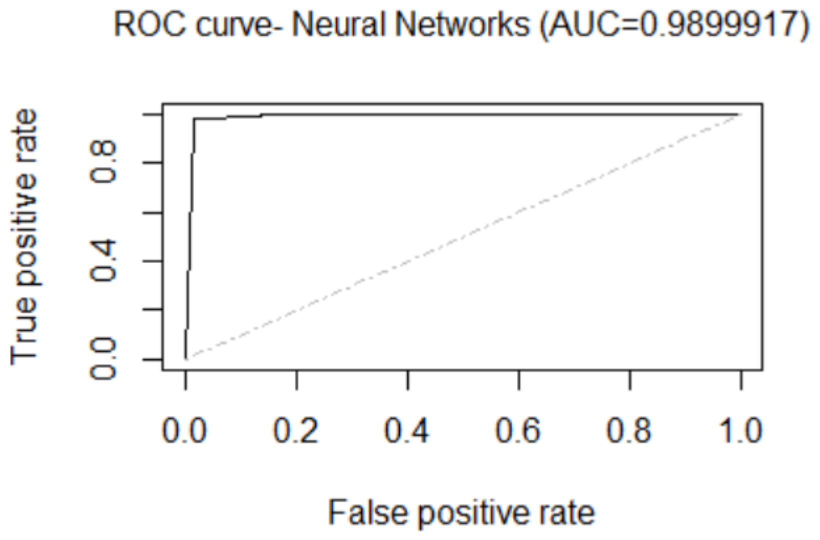


Figure B-8: This Receiver Operating Characteristics (ROC) curve for the testing set fitted with the neural networks with 2 hidden layers has an area under the curve of 0.9899917.

Appendix C: R codes

```
diabetes = read.csv("diabetes_data_upload.csv", header=TRUE)
head(diabetes)
str(diabetes)
dim(diabetes)
summary(diabetes)
attach(diabetes)

library(plyr)
count(diabetes, "class") #40% Negative, 60% Positive



diabetes$Gender = as.factor(diabetes$Gender)
diabetes$Polyuria = as.factor(diabetes$Polyuria)
diabetes$Polydipsia = as.factor(diabetes$Polydipsia)
diabetes$sudden.weight.loss = as.factor(diabetes$sudden.weight.loss)
diabetes$weakness = as.factor(diabetes$weakness)
diabetes$Polyphagia = as.factor(diabetes$Polyphagia)
diabetes$Genital.thrush = as.factor(diabetes$Genital.thrush)
diabetes$visual.blurring = as.factor(diabetes$visual.blurring)
diabetes$Itching = as.factor(diabetes$Itching)
diabetes$Irritability = as.factor(diabetes$Irritability)
diabetes$delayed.healing = as.factor(diabetes$delayed.healing)
diabetes$partial.paresis = as.factor(diabetes$partial.paresis)
diabetes$muscle.stiffness = as.factor(diabetes$muscle.stiffness)
diabetes$Alopecia = as.factor(diabetes$Alopecia)
diabetes$Obesity = as.factor(diabetes$Obesity)
diabetes$class = as.factor(diabetes$class)

str(diabetes)

#Check for Missing Values
missing = diabetes[!complete.cases(diabetes),] #no missing values


library(ggplot2)
install.packages('ggpubr')
library(ggpubr)


a<-ggplot(diabetes, aes(fill=class, x=Gender))+geom_bar()
b<-ggplot(diabetes, aes(fill=class,x=Polyuria))+geom_bar()
```

```
c<-ggplot(diabetes, aes(fill=class,x=Polydipsia))+geom_bar()
d<-ggplot(diabetes, aes(fill=class,x=sudden.weight.loss))+geom_bar()
e<-ggplot(diabetes, aes(fill=class,x=weakness))+geom_bar()
f<-ggplot(diabetes, aes(fill=class,x=Polyphagia))+geom_bar()
g<-ggplot(diabetes, aes(fill=class,x=Genital.thrush))+geom_bar()
h<-ggplot(diabetes, aes(fill=class,x=visual.blurring))+geom_bar()
i<-ggplot(diabetes, aes(fill=class,x=Itching))+geom_bar()
j<-ggplot(diabetes, aes(fill=class,x=Irritability))+geom_bar()
k<-ggplot(diabetes, aes(fill=class,x=delayed.healing))+geom_bar()
l<-ggplot(diabetes, aes(fill=class,x=partial.paresis))+geom_bar()
m<-ggplot(diabetes, aes(fill=class,x=muscle.stiffness))+geom_bar()
n<-ggplot(diabetes, aes(fill=class,x=Alopecia))+geom_bar()
o<-ggplot(diabetes, aes(fill=class,x=Obesity))+geom_bar()
cols <- c("#F76D5E", "#FFFFBF")

p<-ggplot(diabetes, aes(x = Age, fill = class)) +
  geom_density(alpha=0.8)



figure1 <- ggarrange(a,b,c,d,
                    ncol = 2, nrow = 2)
figure1

figure2 <- ggarrange(e,f,g,h,
                    ncol = 2, nrow = 2)
figure2

figure3 <- ggarrange(i,j,k,l,
                    ncol = 2, nrow = 2)
figure3

figure4 <- ggarrange(m,n,o,p,
                    ncol = 2, nrow = 2)
figure4

set.seed(202112)
n1 = dim(diabetes)[1]
train = which(runif(n1)<= .7)

x = model.matrix(class ~ Age + Gender + Polyuria + Polydipsia + sudden.weight.loss +
                weakness + Polyphagia + Genital.thrush + visual.blurring + Itching +
                Irritability + delayed.healing + partial.paresis + muscle.stiffness + Alopecia +
                Obesity,  diabetes)[,-1]
```

```
y = diabetes$class

data = data.frame(y,x)
dim(data)
data.train = data[train,]
data.test = data[-train,]


####################################################
#Discriminant analysis

library(biotools)
boxM(data.train[,-1], data.train[,1]) #p-value < 2.2e-16 reject null use QDA

#normality assumption cannot be tested since most of the predictors are categorical

#######################################################
#Conventional logistic regression

fit.glm = glm(y~., data=data.train, family="binomial")
summary(fit.glm)

model=step(fit.glm) #stepwise selection
summary(model)

coef(model)
exp(coef(model))
par(mfrow=c(2,2))
plot(model)

#Homser and Lemeshow GOF test
install.packages("ResourceSelection")
library(ResourceSelection)
hoslem.test(x = fit.glm$y, y = fitted(fit.glm), g = 10) #p-value = 0.527 (accept null, no lack of fit)

pred2 = predict(fit.glm, newdata=data.test, type = "response")
ypred2 = ifelse(pred2>.5, "Positive", "Negative")
length(which(ypred2==data.test$y))/(n1-length(train)) #compute correct classification rate/test
set prediction result=0.9177215
length(which(ypred2!=data.test$y))/(n1-length(train)) #misclassification rate = 0.08227848

##############if use the conventional logistic (stepwise selection)

install.packages("ResourceSelection")
```

```
library(ResourceSelection)
hoslem.test(x = model$y, y = fitted(model), g = 10) #p-value = 0.9039

pred2 = predict(model, newdata=data.test, type = "response")
ypred2 = ifelse(pred2>.5, "Positive", "Negative")
length(which(ypred2==data.test$y))/(n1-length(train)) #compute correct classification rate/test
set prediction result=0.9303797
length(which(ypred2!=data.test$y))/(n1-length(train)) #misclassification rate = 0.06962025

# check the area under ROC for the test set and plot
library(ROCR)
pred2.roc = ROCR::prediction(pred2, data.test$y) #transform the input data into a standardized
format
performance(pred2.roc,"auc")@y.values[[1]] #get AUC of the ROC for the test set
(AUC=0.9718776)
perf = performance(pred2.roc,"tpr","fpr")
par(mfrow=c(1,1))
plot(perf,colorize=FALSE, col="black") # plot ROC curve
lines(c(0,1),c(0,1),col = "gray", lty = 4 )
title(main = "ROC curve- Logistic regression with stepwise (AUC=0.9718776)",  cex.main = 1,
font.main= 1)

################################################################################
#Classification tree

set.seed(202112)
n1 = dim(diabetes)[1]
train = which(runif(n1)<= .7)
diabetes.train = diabetes[train,]
diabetes.test = diabetes[-train,]

library(tree)
fit.tree = tree(class ~., diabetes.train)
summary(fit.tree) #15 terminal nodes
plot(fit.tree)
text(fit.tree, pretty=0)

pred.tree = predict(fit.tree, diabetes.test, type="class") #the argument type="class" instructs R to
return the  class prediction
table(pred.tree, diabetes.test$class)

APER.tree = 12/nrow(diabetes.test)
APER.tree #misclassification rate = 0.07594937 (before pruning)
```

```
# check the area under ROC for the test set and plot
library(ROCR)

pred2 = predict(fit.tree, newdata=diabetes.test, type = "vector")
tree.roc = prediction(pred2[,2], diabetes.test$class) #transform the input data into a
standardized format
performance(tree.roc,"auc")@y.values[[1]] #get AUC of the ROC for the test set
(AUC=0.9643507)
perf = performance(tree.roc,"tpr","fpr")
par(mfrow=c(1,1))
plot(perf,colorize=FALSE, col="black") # plot ROC curve
lines(c(0,1),c(0,1),col = "gray", lty = 4 )
title(main = "ROC curve- Classification tree (AUC=0.9643507)",  cex.main = 1,   font.main= 1)


################################################################################
#prune tree
set.seed(202112)
cv.fit = cv.tree(fit.tree, FUN=prune.misclass)
cv.fit #smallest deviance is 31 and its corresponding subtree size is 8

par(mfrow =c(1,1))
plot(cv.fit$size ,cv.fit$dev ,type="b")  #plot tree sizes versus their correspongding deviations to
select the optimal tree (smallest dev)
plot(cv.fit$k ,cv.fit$dev ,type="b") #plot different alpha values versus the correspongding
deviations of the trees built based the alpha values, to select the optimal tree (smallest dev)


prune.fit = prune.misclass(fit.tree, best=8)  #build the optimal tree select by CV; best=8 means
that the number of the terminal nodes of the optimal tree is 8.
summary(prune.fit)
par(mfrow =c(1,1))
plot(prune.fit)
text(prune.fit, pretty=0)

#predict testing set
prune.pred = predict(prune.fit, diabetes.test, type="class")
table(prune.pred, diabetes.test$class)

APER.prune = (4+7) /nrow(diabetes.test)
APER.prune #misclassification rate = 0.06962025 (after pruning)
```

```
# check the area under ROC for the test set and plot
library(ROCR)

pred2 = predict(prune.fit, newdata=diabetes.test, type = "vector")
prune.roc = prediction(pred2[,2], diabetes.test$class) #transform the input data into a
standardized format
performance(prune.roc,"auc")@y.values[[1]] #get AUC of the ROC for the test set
(AUC=0.9379653)
perf = performance(prune.roc,"tpr","fpr")
par(mfrow=c(1,1))
plot(perf,colorize=FALSE, col="black") # plot ROC curve
lines(c(0,1),c(0,1),col = "gray", lty = 4)
title(main = "ROC curve- Pruned tree (AUC=0.9379653)",  cex.main = 1,   font.main= 1)

##############################################################################
#Random Forest
library(randomForest)
set.seed(202112)

rf.fit = randomForest(class~.,data=diabetes, subset=train,
mtry=4,importance=TRUE)#m=sqrt(16)=4 subset predictors to make the split
yhat.rf = predict(rf.fit, newdata=diabetes.test, type='class')
table(yhat.rf, diabetes.test$class)

APER.rf = (3) /nrow(diabetes.test)
APER.rf #misclassification rate = 0.01898734 (random forest)
importance(rf.fit)
par(mfrow=c(1,1))
varImpPlot(rf.fit)

# check the area under ROC for the test set and plot

library(ROCR)

pred2 = predict(rf.fit, newdata=diabetes.test, type = "prob")
rf.roc = prediction(pred2[,2], diabetes.test$class) #transform the input data into a standardized
format
performance(rf.roc,"auc")@y.values[[1]] #get AUC of the ROC for the test set (AUC=0.9990074)
perf = performance(rf.roc,"tpr","fpr")
par(mfrow=c(1,1))
plot(perf,colorize=FALSE, col="black") # plot ROC curve
lines(c(0,1),c(0,1),col = "gray", lty = 4)
title(main = "ROC curve- Random Forest (AUC=0.99990074)",  cex.main = 1,   font.main= 1)
```

```
##############################################################################
############
#SVM

install.packages("e1071")
library(e1071)

set.seed(202112)
tune.out = tune(svm, y~., data=data.train, kernel="linear",ranges=list(cost=c(0.001, 0.01, 0.1,
1,5, 10, 100)))
summary(tune.out) #cost=100 with performance error 0.06869369


bestmod = tune.out$best.model #71 support vectors
summary(bestmod)

yhat.svm = predict(bestmod, data.test)
table(predict=yhat.svm, truth=data.test$y)

APER.svm = (4+7) /nrow(data.test)
APER.svm #misclassification rate = 0.06962025 (same misclassification rate after prunning the
tree)


set.seed(202112)
tune.out=tune(svm, y~., data=data.train, kernel="radial",
ranges=list(cost=c(0.1,1,10,100,1000),gamma=c(0.5,1,2,3,4)))
summary(tune.out) #cost=100, gamma=0.5, performance error = 0.06081081

bestmod.radial = tune.out$best.model
summary(bestmod.radial) #197 support vectors

yhat.svm = predict(bestmod.radial, data.test)
table(predict=yhat.svm, truth=data.test$y)

APER.svm = 6 /nrow(data.test)
APER.svm #misclassification rate = 0.03797468
######################################## use polynomial ##################

set.seed(202112)
tune.out=tune(svm, y~., data=data.train, kernel="polynomial", degree = 3,
          ranges=list(cost=c(0.001, 0.01, 0.1, 1,5, 10, 100),gamma=c(0.5,1,2,3,4)))
summary(tune.out) #cost=0.001, gamma=2, performance error = 0.04954955
```

```
bestmod.poly = tune.out$best.model
summary(bestmod.poly) #114

yhat.svm = predict(bestmod.poly, data.test)
table(predict=yhat.svm, truth=data.test$y)

APER.svm = 3 /nrow(data.test)
APER.svm #misclassification rate = 0.01898734 (same as random forest)


# check the area under ROC for the test set and plot
library(ROCR)

svmfit.opt = svm(y~., data=data.train, kernel="polynomial", gamma=2, cost=0.001, degree=3)
pred2 = predict(svmfit.opt, newdata=data.test)
rf.roc = prediction(as.numeric(pred2), as.numeric(data.test$y))
performance(rf.roc,"auc")@y.values[[1]] #get AUC of the ROC for the test set (AUC=0.983871)
perf = performance(rf.roc,"tpr","fpr")
par(mfrow=c(1,1))
plot(perf,colorize=FALSE, col="black") # plot ROC curve
lines(c(0,1),c(0,1),col = "gray", lty = 4)
title(main = "ROC curve- SVM polynomial (AUC=0.983871)",  cex.main = 1,   font.main= 1)


##################alternative way of getting svm roc plot
library(pROC)
svmfit.opt = svm(y~., data=data.train, kernel="polynomial", gamma=2, cost=0.001, degree=3)
pred2 = predict(svmfit.opt, newdata=data.test)
roc_svm_test <- roc(response = as.numeric(data.test$y), predictor =as.numeric(pred2))
plot(roc_svm_test, print.auc=TRUE, print.auc.x = 0.5, print.auc.y = 0.3)
legend(0.3, 0.2, legend = c("test-svm"), lty = c(1), col = c("blue"))

#####################################################
#neural network

library(neuralnet)
#Normalize Age data
scaletrain = data.train
scaletrain$Age = scale(scaletrain$Age)
scaletest = data.test
scaletest$Age = scale(scaletest$Age)
```

```
set.seed(202112)
nn = neuralnet(y == "Positive"~., data=scaletrain, hidden = c(3,2), linear.output = FALSE)
nn.results <- compute(nn, scaletest[,2:17])
results <- data.frame(actual = scaletest$y, prediction = nn.results$net.result)

plot(nn)

prednn <- predict(nn, scaletest)
table(scaletest$y == "Positive", prednn[, 1] > 0.5)
APER.net = (3)/nrow(scaletest)
APER.net #0.01898734 (same as polynomial)


# check the area under ROC for the test set and plot
library(ROCR)

pred2 = predict(nn, newdata=scaletest)
rf.roc = ROCR::prediction(pred2, scaletest$y) #transform the input data into a standardized
format
performance(rf.roc,"auc")@y.values[[1]] #get AUC of the ROC for the test set (AUC=0.9899917)
perf = performance(rf.roc,"tpr","fpr")
par(mfrow=c(1,1))
plot(perf,colorize=FALSE, col="black") # plot ROC curve
lines(c(0,1),c(0,1),col = "gray", lty = 4)
title(main = "ROC curve- Neural Networks (AUC=0.9899917)",  cex.main = 1,   font.main= 1)
```