**R CODE for the project**

setwd("/Users/sree/Desktop/Data Analytics/Final Project")

getwd()

data <- read.csv("diabetes.csv",header=T)

data

summary(data)

str(data)

cor(data)

#to chnage outcoe data into factor column

data$Outcome <- factor(data$Outcome)

data$Outcome

names(data)[7] <- "dpf"

names(data) <- tolower(names(data))

names(data)

#to see the number of datas and rows read

print(paste0("number of observations = ", dim(data)[1]))

print(paste0("number of predictors = ", dim(data)[2]))

par(mfrow = c(1, 2))

#to see the frequency of variables used in the data

hist(data$pregnancies)

hist(data$age)

hist(data$glucose)

hist(data$bmi)

hist(data$insulin)

# boxplot between insulin level and the outcome variable

with(data, boxplot(insulin ~ outcome,

ylab = "insulin",

xlab = "Presence of Diabetes",

main = "Figure A",

outline = FALSE))

#relation between different varibales used in the data

pairs(data)

# correlation matrix

library(reshape2)

cor\_melt <- melt(cor(data[, 1:8]))

cor\_melt <- cor\_melt[which(cor\_melt$value > 0.3 & cor\_melt$value != 1), ]

cor\_melt <- cor\_melt[1:5, ]

cor\_melt

#splitting the data into train and test data for prediction using stratified sampling

library(caret)

library(e1071)

set.seed(300)

inTrain <- createDataPartition(y = data[,"outcome"], p = 0.8, list = FALSE)

list(inTrain)

train\_pima <- data[inTrain,]

test\_pima <- data[-inTrain,]

train\_pima

test\_pima

names(data)

#building the logistic model for the data read

model <- glm(outcome ~pregnancies+glucose +bloodpressure+skinthickness+insulin+bmi+dpf+age,family=binomial(link='logit'),data=train\_pima)

summary(model)

#confint(model)

#exp(coef(model))

anova(model, test="Chisq")

#residuals(model,type="deviance")

#to find mcfadden R2 of the model

library(pscl)

pR2(model)

#predicting the model for the frst model

fitted.results <- predict(model,newdata=subset(test\_pima,select=c(1,2,3,4,5,6,7,8)),type='response')

fitted.results <- ifelse(fitted.results > 0.5,1,0)

fitted.results

#finding the accruacy

misClasificError <- mean(fitted.results != test\_pima$outcome)

print(paste('Accuracy',1-misClasificError))

#backwardmodel

step(model,direction="backward",trace=T)

backmodel<-glm(formula = outcome ~ pregnancies + glucose + bloodpressure +

bmi + dpf, family = binomial(link = "logit"), data = train\_pima)

summary(backmodel)

fitted.results.backmodel <- predict(backmodel,newdata=subset(test\_pima,select=c(1,2,3,4,5,6,7,8)),type='response')

fitted.results.backmodel <- ifelse(fitted.results.backmodel > 0.5,1,0)

fitted.results.backmodel

misClasificError <- mean(fitted.results.backmodel != test\_pima$outcome)

print(paste('Accuracy',1-misClasificError))

anova(backmodel, test="Chisq")

mea1=influence.measures(backmodel)

summary(mea1)

nullR2 =glm(outcome ~ 1, family = binomial(link = "logit"), data = train\_pima)

fit2R = 1-logLik(backmodel)/ logLik(nullR2)

fit2R

#forwardmodel

base=glm(outcome ~pregnancies, family = binomial("logit"), data = train\_pima)

step(base, scope=list(upper=model,lower=~1 ), direction="forward",trace=F)

model1=glm(formula = outcome ~ pregnancies + glucose + bmi + bloodpressure +

dpf, family = binomial, data = train\_pima)

summary(model1)

fitted.results.forwmodel <- predict(model1,newdata=subset(test\_pima,select=c(1,2,3,4,5,6,7,8)),type='response')

fitted.results.forwmodel <- ifelse(fitted.results.forwmodel > 0.5,1,0)

fitted.results.forwmodel

misClasificError <- mean(fitted.results.forwmodel != test\_pima$outcome)

print(paste('Accuracy',1-misClasificError))

anova(model1, test="Chisq")

mea2=influence.measures(model1)

summary(mea2)

nullR3 =glm(outcome ~ 1, family = binomial(link = "logit"), data = train\_pima)

fit3R = 1-logLik(model1)/ logLik(nullR3)

fit3R

#stepwiseregression

step(base, scope=list(lower=~1, upper=model), data=train\_pima, direction="both")

summary(stepwise)

step\_model=glm(formula = outcome ~ pregnancies + glucose + bmi + bloodpressure +

dpf, family = binomial, data = train\_pima)

summary(step\_model)

fitted.results.stepmoodel <- predict(step\_model,newdata=subset(test\_pima,select=c(1,2,3,4,5,6,7,8)),type='response')

fitted.results.stepmoodel <- ifelse(fitted.results.stepmoodel > 0.5,1,0)

fitted.results.stepmoodel

misClasificError <- mean(fitted.results.stepmoodel != test\_pima$outcome)

print(paste('Accuracy',1-misClasificError))

anova(step\_model, test="Chisq")

mea2=influence.measures(step\_model)

summary(mea2)

nullR4 =glm(outcome ~ 1, family = binomial(link = "logit"), data = train\_pima)

fit4R = 1-logLik(model1)/ logLik(nullR4)

fit4R

#copying the output of the final result into a csv file

write.csv(fitted.results.stepmoodel,file="dataoutput.csv")

#plotting the residual for the build model

par(mfrow = c(2, 2))

plot(model)

library(InformationValue)

optCutOff <- optimalCutoff(test\_pima$outcome, fitted.results.stepmoodel)[1]

optCutOff

sensitivity<-sensitivity(test\_pima$outcome, fitted.results.stepmoodel, threshold = optCutOff)

sensitivity

#confustion matril for the build model

confusionMatrix(test\_pima$outcome, fitted.results.stepmoodel, threshold = optCutOff)

#plotting the ROC curve

plotROC(test\_pima$outcome, fitted.results.stepmoodel)