# import the data

setwd("C:\\Users\\admin\\Desktop\\project")

data <- read.csv("Dataset\_spine.csv")

summary(data)

library(corrplot)

# making dummy variable for status

data$Status <- ifelse(data$Status=="Abnormal",1,0)

# correlation plot

cor <- cor(data)

corrplot(cor, method = "circle", type = "upper")

# outlier capping

attach(data)

bx1 <- boxplot(data$pelvic\_incidence)

bx1$stats

quantile(data$pelvic\_incidence, seq(0,1,0.02))

data$pelvic\_incidence <- ifelse(data$pelvic\_incidence>95,90,data$pelvic\_incidence)

boxplot(data$pelvic\_incidence)

bx2 <- boxplot(data$pelvic\_tilt)

bx2$stats

quantile(data$pelvic\_tilt, seq(0,1,0.02))

data$pelvic\_tilt <- ifelse(data$pelvic\_tilt>38,36, data$pelvic\_tilt)

bx3 <- boxplot(data$lumbar\_lordosis\_angle)

bx3$stats

quantile(data$lumbar\_lordosis\_angle, seq(0,1,0.02))

data$lumbar\_lordosis\_angle <- ifelse(data$lumbar\_lordosis\_angle>100,93, data$lumbar\_lordosis\_angle)

bx4 <- boxplot(data$sacral\_slope)

bx4$stats

quantile(data$sacral\_slope, seq(0,1,0.02))

data$sacral\_slope <- ifelse(data$sacral\_slope>79,69, data$sacral\_slope)

bx5 <- boxplot(data$pelvic\_radius)

bx5$stats

quantile(data$pelvic\_radius, seq(0,1,0.02))

data$pelvic\_radius <- ifelse(data$pelvic\_radius<89,90, data$pelvic\_radius)

data$pelvic\_radius <- ifelse(data$pelvic\_radius>146,145, data$pelvic\_radius)

bx6 <- boxplot(data$degree\_spondylolisthesis)

bx6$stats

quantile(data$degree\_spondylolisthesis, seq(0,1,0.02))

data$degree\_spondylolisthesis <- ifelse(data$degree\_spondylolisthesis>100,90, data$degree\_spondylolisthesis)

# divide the data into train and test

split <- sample(1:nrow(data), 0.8\*nrow(data))

train <- data[split,]

test <- data[-split,]

table(train$Status)

table(test$Status)

# fit the model on train data

library(car)

model <- glm(Status~pelvic\_incidence + sacral\_slope + pelvic\_radius +

degree\_spondylolisthesis, data = train, family = "binomial")

summary(model)

vif(model)

# prediction on model

pred <- predict(model, type = "response")

prediction <- ifelse(pred>0.5,1,0)

# check the result using confusion matrix

library(caret)

confusionMatrix(train$Status,prediction)

# check the RMSE of the model

library(Metrics)

rmse(train$Status,prediction)

# check the model on test data

predTest <- predict(model,test)

predictionT <- ifelse(predTest>0.5,1,0)

confusionMatrix(test$Status, predictionT)

# ROC plot

library(InformationValue)

plotROC(actuals = train$Status, predictedScores = as.numeric(fitted(model)))

# Random Forest Model #

library(randomForest)

modelrf <- randomForest(as.factor(Status) ~ ., data = train, do.trace=T)

summary(modelrf)

importance(modelrf)

varImpPlot(modelrf)

# prediction using rf model

preRF\_train <- predict(modelrf, train)

confusionMatrix(preRF\_train, train$Status)

preRF\_test <- predict(modelrf, test)

confusionMatrix(preRF\_test, test$Status)