R CODE FOR TERM PROJECT

#IMPORTING DATASET

library(ranger)

library(caret)

library(data.table)

creditcard\_data <- read.csv("C:/Users/Saurav N. Thakor/Desktop/credit.csv")

print(creditcard\_data) A close up of a paper

Description automatically generated

**#DATA EXPLORATION**

table(creditcard\_data$class)

summary(creditcard\_data$Amount)

names(creditcard\_data)

sd(creditcard\_data$Amount)

nrow(creditcard\_data)

dim(creditcard\_data)

View(creditcard\_data)

summary(creditcard\_data)

A group of numbers and symbols

Description automatically generated

A screenshot of a computer

Description automatically generated

dim(creditcard\_data)

head(creditcard\_data,6)

tail(creditcard\_data,6)

A screenshot of a computer

Description automatically generated

table(creditcard\_data$Class)

summary(creditcard\_data$Amount)

names(creditcard\_data)

var(creditcard\_data$Amount)

A screenshot of a computer

Description automatically generated

**# DATA VISUALIZATION**

# Separate data for fraud and non-fraud transactions

fraud\_data <- subset(creditcard\_data, Class == 1)

non\_fraud\_data <- subset(creditcard\_data, Class == 0)

# Create histograms for fraud and non-fraud transactions

par(mfrow = c(1, 2)) # Set up a 1x2 grid for side-by-side plots

hist(fraud\_data$Amount, breaks = 30, col = "red", main = "Fraudulent Transactions", xlab = "Amount")

hist(non\_fraud\_data$Amount, breaks = 30, col = "green", main = "Non-Fraudulent Transactions", xlab = "Amount")

par(mfrow = c(1, 1)) # Reset the plotting layout to default

A red line with green lines

Description automatically generated

# CORRELATION

cor\_matrix = cor(creditcard\_data)

print(cor\_matrix)

cor\_test <- cor.test(creditcard\_data$Class,creditcard\_data$Amount)

print(cor\_test)

A close up of a newspaper

Description automatically generated

install.packages("corrplot")

library(corrplot)

num\_values <- creditcard\_data[,2:31]

cormat <- cor(num\_values)

corrplot(cormat, method = "ellipse")

A graph with numbers and lines

Description automatically generated

#DATA MANIPULATION

head(creditcard\_data) A screenshot of a computer

Description automatically generated

creditcard\_data$Amount=scale(creditcard\_data$Amount)

NewData=creditcard\_data[,-c(1)]

head(NewData)

A screenshot of a computer

Description automatically generated

#DATA MODELING

#SPLITTING DATA IN TO TRAINING AND TESTING SETS

library(caTools)

set.seed(123)

data\_sample = sample.split(NewData$Class,SplitRatio=0.80)

train\_data = subset(NewData,data\_sample==TRUE)

test\_data = subset(NewData,data\_sample==FALSE)

dim(train\_data)

dim(test\_data)

A computer screen with text

Description automatically generated

# Fit a linear regression model

linear\_model <- lm(Amount ~ V1 + V2 + V3, data =creditcard\_data)

plot(creditcard\_data$V1,creditcard\_data$Amount, main = "Multiple Linear Regression", xlab ="V1" , ylab= "Amount")

intercept <- coef(linear\_model)[1]

slope <- coef(linear\_model)[2]

abline(intercept,b = slope,col ="red")

A graph showing a number of bubbles

Description automatically generated

# Print the summary of the regression model

summary(linear\_model)

A screenshot of a computer program

Description automatically generated

#LOGISTIC REGRESSION

Logistic\_Model=glm(Class~.,test\_data,family=binomial())

summary(Logistic\_Model)

A screenshot of a computer

Description automatically generated

plot(Logistic\_Model)

A graph with a line

Description automatically generatedA graph with numbers and lines

Description automatically generated

A graph of a chain

Description automatically generatedA graph of a number of lines

Description automatically generated with medium confidence

# Check the mean values by class

aggregate(Amount ~ Class, data = creditcard\_data, FUN = mean)

A close-up of a credit card

Description automatically generated

# Perform t-test

t\_test\_result <- t.test(creditcard\_data$Amount ~ creditcard\_data$Class)

# Display the t-test result

print(t\_test\_result)

**A screenshot of a computer code

Description automatically generated**

# KNN (k- NEASREST NEIGHBORS)setting

install.packages("class")

library(class)

# Set the number of neighbors (k)

k <- 5

# Train the KNN model

knn\_model <- knn(train\_data[, -ncol(train\_data)], test\_data[, -ncol(test\_data)], train\_data$Class, k = k)

print(knn\_model)

A screen shot of a number

Description automatically generated

summary(knn\_model)

