**Student performance prediction**

**1.IMPORTING LIBRARIES**

**CODE:**

**library(dplyr) # for data manipulation**

**library(stringr) # for data manipulation**

**library(caret) # for sampling**

**library(caTools) # for train/test split**

**library(ggplot2) # for data visualization**

**OUTPUT:**

Text

Description automatically generated with low confidence

**2.DATA CLEANING**

**CODE:**

**df<-read.csv("C:/Users/bhuva/Documents/StudentsPerformance.csv")**

**head(df)**

**class(df)**

**dim(df)**

**str(df)**

**summary(df)**

**OUTPUT:**

A screenshot of a computer

Description automatically generated with medium confidence

**3. DATA MANIPULATION**

**CODE:**

**df$test.preparation.course<-toupper(df$test.preparation.course)**

**head(df$test.preparation.course)**

**head(unique(df))**

**head(duplicated(df))**

**df %>% distinct()**

**OUTPUT:**

Table

Description automatically generated

**CODE:**

**colSums(is.na(df))**

**df[is.na(df)] = 0**

**head(df)**

**colSums(is.na(df))**

**colnames(df)**

**namesOfColumns <- c("Gender","Race","Parent\_Education","Lunch","Test\_Prep","Math\_Score","Reading\_Score","Writing\_Score")**

**colnames(df) <- namesOfColumns**

**colnames(df)**

**OUTPUT:**

A picture containing text

Description automatically generated

**CODE:**

**print(head(df))**

**print(tail(df))**

**print(mean(df$math.score))**

**print(sd(df$math.score))**

**print(table(df$Gender))**

**OUTPUT:**

Table

Description automatically generated

**CODE:**

**print(table(df$Gender))**

**table(df$Race)**

**table(df$Parent\_Education)**

**table(df$Lunch)**

**table(df$Test\_Prep)**

**table(df$Math\_Score)**

**# Categories are uniform throughout the dataset**

**OUTPUT:**

A picture containing table

Description automatically generated

**4.DATA VISUALIZATION**

**CODE:**

**### Data Visualization**

**## Plot Exam scores by Gender to determine if there is a different score tendency for each gender.**

**#Barplot Math scores by Gender plot**

**a1 <- ggplot(df, aes(Math\_Score)) + geom\_histogram(binwidth=5, color="gray", aes(fill=Gender))**

**a1 <- a1 + xlab("Math Scores") + ylab("Gender") + ggtitle("Math Scores by Gender")**

**a1**

**OUTPUT:**

Chart, histogram

Description automatically generated

**CODE:**

**#Barplot Reading scores by Gender plot**

**a2 <- ggplot(df, aes(Reading\_Score)) + geom\_histogram(binwidth=5, color="gray", aes(fill=Gender))**

**a2 <- a2 + xlab("Reading Scores") + ylab("Gender") + ggtitle("Reading Scores by Gender")**

**a2**

**OUTPUT:**

Chart, histogram

Description automatically generated

**CODE:**

**#Barplot Writing scores by Gender plot**

**a3 <- ggplot(df, aes(Writing\_Score)) + geom\_histogram(binwidth=5, color="gray", aes(fill=Gender))**

**a3 <- a3 + xlab("Writing Scores") + ylab("Gender") + ggtitle("Writing Scores by Gender")**

**a3**

**OUTPUT:**

Chart, histogram

Description automatically generated

**CODE:**

# Math scores by Gender plot

p <- ggplot(df, aes(Math\_Score)) + geom\_histogram(binwidth=5, color="gray", aes(fill=Gender))

p <- p + xlab("Math Scores") + ylab("Gender") + ggtitle("Math Scores by Gender")

p

**OUTPUT:**

Chart, histogram

Description automatically generated

**CODE:**

# Reading scores by Gender plot

p1 <- ggplot(df, aes(Reading\_Score)) + geom\_histogram(binwidth=5, color="gray", aes(fill=Gender))

p1 <- p1 + xlab("Reading Scores") + ylab("Gender") + ggtitle("Reading Scores by Gender")

p1

**OUTPUT:**

Chart, histogram

Description automatically generated

**CODE:**

# Writing scores by Gender plot

p2 <- ggplot(df, aes(Writing\_Score)) + geom\_histogram(binwidth=5, color="gray", aes(fill=Gender))

p2 <- p2 + xlab("Writing Scores") + ylab("Gender") + ggtitle("Writing Scores by Gender")

p2

**OUTPUT:**

Chart, histogram

Description automatically generated

CODE:

# Math scores by race

p3 <- ggplot(df, aes(Math\_Score)) + geom\_histogram(binwidth=5, color="blue", aes(fill=Race))

p3 <- p3 + xlab("Math Scores") + ylab("Race") + ggtitle("Math Scores by Race")

p3

OUTPUT:

Chart, histogram

Description automatically generated

**5.BOXPLOT**

**CODE:**

# Boxplot of scores and Test Prep by Gender

b <- ggplot(df, aes(Gender, Writing\_Score, color = Test\_Prep))

b <- b + geom\_boxplot()

b <- b + ggtitle("Writing scores by Gender Boxplot")

b <- b + xlab("Gender") + ylab("Writing Scores")

b

**OUTPUT:**

Chart, box and whisker chart

Description automatically generated

**CODE:**

b1 <- ggplot(df, aes(Gender, Math\_Score, color = Test\_Prep))

b1 <- b1 + geom\_boxplot()

b1 <- b1 + ggtitle("Math scores by Gender Boxplot")

b1 <- b1 + xlab("Gender") + ylab("Math Scores")

b1

**OUTPUT:**

Chart, box and whisker chart

Description automatically generated

**CODE:**

b2 <- ggplot(df, aes(Gender, Reading\_Score, color = Test\_Prep))

b2 <- b2 + geom\_boxplot()

b2 <- b2 + ggtitle("Reading scores by Gender Boxplot")

b2 <- b2 + xlab("Gender") + ylab("Reading Scores")

b2

**OUTPUT:**

Chart, box and whisker chart

Description automatically generated

**6.HEAT WAVE GRAPH**

**CODE:**

# Parents education and Scores

e <- ggplot(df) +

geom\_bin2d(aes(x=Writing\_Score, y=Parent\_Education)) +

xlab("Writing Scores") + ylab("Parents Education")

e

**OUTPUT:**

Chart, bar chart

Description automatically generated

**CODE:**

e1 <- ggplot(df) +

geom\_bin2d(aes(x=Reading\_Score, y=Parent\_Education)) +

xlab("Reading Scores") + ylab("Parents Education")

e1

**OUTPUT:**

Chart, bar chart

Description automatically generated

**CODE:**

e2 <- ggplot(df) +

geom\_bin2d(aes(x=Math\_Score, y=Parent\_Education)) +

xlab("Math Scores") + ylab("Parents Education")

e2

**OUTPUT:**

Chart, bar chart

Description automatically generated

**7.PIE CHART**

**CODE:**

a=table(df$Gender)

pct=round(a/sum(a)\*100)

lbs=paste(c("Female","Male")," ",pct,"%",sep=" ")

library(plotrix)

pie3D(a,labels=lbs,

main="Pie Chart Depicting Ratio of Female and Male")

**OUTPUT:**

Chart, pie chart

Description automatically generated

CODE:

# Equivalent with a formula

pairs(~Math\_Score + Reading\_Score + Writing\_Score, data = df)

# Equivalent but using the plot function

plot(data)

OUTPUT:

A picture containing diagram

Description automatically generated

**8.DATA ENCODING/SPLITTING**

**CODE:**

### Prediction model

randIndex <- sample(1:dim(df)[1])

# # In order to split data, create a 2/3 cutpoint and round the number

cutpoint2\_3 <- floor(2\*dim(df)[1]/3)

# create train data set, which contains the first 2/3 of overall data

trainData <- df[randIndex[1:cutpoint2\_3],]

dim(trainData)

head(trainData)

# create test data, which contains the left 1/3 of the overall data

testData <- df[randIndex[(cutpoint2\_3+1):dim(df)[1]],]

dim(testData) # check test data set

head(testData)

**OUTPUT:**

A picture containing text

Description automatically generated

**9.APPLYING LINEAR REGRESSION MODEL**

**CODE:**

#------------------------------------------------------lm model

model <- lm(Math\_Score ~ Writing\_Score + Gender + Race + Lunch + Parent\_Education + Test\_Prep,data=trainData)

summary(model)

lmPred <- predict(model,testData,interval = "prediction", level=0.95)

summary(lmPred)

head(lmPred)

**OUTPUT:**

Text

Description automatically generated

**10.ADDING PREDICTIONS**

**CODE:**

# 1. Add predictions

mydata1 <- cbind(testData, lmPred)

head(mydata1)

# 2. Regression line + confidence intervals

p <- ggplot(mydata1, aes( fit, Math\_Score)) +

geom\_point() +

stat\_smooth(method = lm)

# 3. Add prediction intervals

p + geom\_line(aes(y = lwr), color = "red", linetype = "dashed")+

geom\_line(aes(y = upr), color = "red", linetype = "dashed") +

xlab("Predicted Scores") + ylab("Test Scores")

**OUTPUT:**

Table

Description automatically generated

**Chart, scatter chart

Description automatically generated**

**DECISION TREE:**

**CODE:**

**library(rpart)**

**library(rpart.plot)**

**decisionTree\_model <- rpart(Math\_Score ~ . , df, method = 'class')**

**predicted\_val <- predict(decisionTree\_model, df, type = 'class')**

**probability <- predict(decisionTree\_model, df, type = 'prob')**

**rpart.plot(decisionTree\_model)**

**OUTPUT:**

Diagram

Description automatically generated