install.packages("ggplot2")

install.packages("readxl")

install.packages("gridExtra")

install.packages("corrplot")

install.packages("caret")

install.packages("rpart")

install.packages("rpart.plot")

install.packages("randomForest")

install.packages("lattice")

install.packages("Metrics")

library(readxl)

library(ggplot2)

library(gridExtra)

library(corrplot)

library(caret)

library(rpart)

library(rpart.plot)

library(randomForest)

library(lattice)

library(Metrics)

#data loaded

library(ggplot2)

CarPrediction <- read\_excel("C:\\Users\\hp\\Documents\\CarPrediction.xlsx")

View(CarPrediction)

#data cleaning

library(ggplot2)

library(gridExtra)

library(lattice)

# Converted to factors

CarPrediction$Make <- as.factor(CarPrediction$Make)

CarPrediction$Model <- as.factor(CarPrediction$Model)

CarPrediction$Condition <- as.factor(CarPrediction$Condition)

#No outliers

ggplot(data=CarPrediction) + geom\_boxplot(mapping = aes(x = Price))

ggplot(data = CarPrediction)+geom\_boxplot(mapping = aes(x= Mileage))

ggplot(data = CarPrediction)+geom\_boxplot(mapping = aes(x= Year))

#No null values

#no NA values

sum(is.na(CarPrediction$Price))

#No zeroes

sum(CarPrediction == 0)

# Scatter mileage vs. price

scatter1<-ggplot(data=CarPrediction, aes(x = Mileage, y = Price)) +

geom\_point() +

labs(title = "Scatter of Mileage vs Price")+theme\_minimal()

# Scatter year vs. price

scatter2<-ggplot(data=CarPrediction, aes(x = Year, y = Price)) +

geom\_point() +

labs(title = "Scatter of Year vs Price")+theme\_minimal()

grid.arrange(scatter1,scatter2,ncol=2)

#finding correlation

library(ggplot2)

library(corrplot)

variables\_numeric <- sapply(CarPrediction, is.numeric)

data\_numeric <- CarPrediction[, variables\_numeric]

data\_numeric <- cbind(data\_numeric, price = CarPrediction$Price)

# correlation matrix

matrix\_correlation <- cor(data\_numeric)

# plot

corrplot(matrix\_correlation, method = "color")

#Averaging price with categorical variables

average\_price\_Make <- aggregate(Price ~ Make, data = CarPrediction, mean)

average\_price\_Make

average\_price\_Model <- aggregate(Price ~ Model, data = CarPrediction, mean)

average\_price\_Model

average\_price\_Condition <- aggregate(Price ~ Condition, data = CarPrediction, mean)

average\_price\_Condition

#create coloumn chart for categorical variables

library(ggplot2)

library(gridExtra)

columnchart\_make1<- ggplot(average\_price\_Make, aes(x = Make, y = Price)) +

geom\_col() +

labs(title = "Average range of Price of Make", x = "Make", y = "Average Price") +

theme\_minimal()

columnchart\_model2<- ggplot(average\_price\_Model, aes(x = Model, y = Price)) +

geom\_col() +

labs(title = "Average range of Price of Model", x = "Model", y = "Average Price") +

theme\_minimal()

columnchart\_condition3<- ggplot(average\_price\_Condition, aes(x = Condition, y = Price)) +

geom\_col() +

labs(title = "Average range of Price of Condition", x = "Condition", y = "Average Price") +

theme\_minimal()

grid.arrange(columnchart\_make1,columnchart\_model2,columnchart\_condition3,ncol=2)

#data partitioning

library(caret)

set.seed(123)

trainedgroup <- createDataPartition(CarPrediction$Price, p = 0.7, list = FALSE)

train\_data <- CarPrediction[trainedgroup, ]

test\_data <- CarPrediction[-trainedgroup, ]

#modelling

# Build MLR

mlr\_model <- lm(Price ~ Mileage + Year + Model + Condition + Make, data = train\_data)

summary(mlr\_model)

#predictions

mlr\_prediction <- predict(mlr\_model , newdata = test\_data)

head(mlr\_prediction)

#Evaluation

MSE1 <- mean((mlr\_prediction - test\_data$Price)^2)

MSE1

# Root Mean Squared Error (RMSE)

RMSE1 <- sqrt(MSE1)

RMSE1

#MAE

MAE1 <- mean(abs(mlr\_prediction - test\_data$Price))

MAE1

# R squared

R\_squared1 <- 1 - sum((test\_data$Price - mlr\_prediction)^2) / sum((test\_data$Price - mean(test\_data$Price))^2)

R\_squared1

RMSE1

MAE1

R\_squared1

#create relation

library(ggplot2)

prediction\_dataframe <- data.frame(Actual = test\_data$Price, Predicted = mlr\_prediction)

ggplot(prediction\_dataframe, aes(x = Actual , y = Predicted )) +

geom\_point() +

labs(title = "Actuals and predicted values of price",

x = "Actual value of Price",

y = "Predicted value of Price") +

theme\_minimal()

#decision tree

library(rpart.plot)

library(rpart)

#Build

tree\_model <- rpart(Price ~ Year + Mileage + Condition + Make + Model, data = train\_data, control = rpart.control(minsplit = 5, minbucket = 5))

tree\_model

rpart.plot(tree\_model, extra = 101, type = 0)

#prediction

decisiontree\_prediction <- predict(tree\_model, newdata = test\_data, type = "vector")

head(decisiontree\_prediction)

#evaluation

# MAE

MAE2 <- mean(abs(decisiontree\_prediction - test\_data$Price))

MAE2

# MSE

MSE2 <- mean((decisiontree\_prediction - test\_data$Price)^2)

MSE2

# RMSE

RMSE2 <- sqrt(MSE2)

RMSE2

#R-Squared

R\_squared2<- 1 - sum((test\_data$Price - decisiontree\_prediction)^2) / sum((test\_data$Price - mean(test\_data$Price))^2)

R\_squared2

MAE2

RMSE2

R\_squared2

#create relation

tree\_prediction\_dataframe <- data.frame(Actual = test\_data$Price, Predicted = decisiontree\_prediction)

ggplot(tree\_prediction\_dataframe, aes(x = Actual, y = Predicted)) +

geom\_point() +

labs(title = "Actuals and Predicted values of Prices",

x = "Actual value of Price",

y = "Predicted value of Price") +

theme\_minimal()

#random forest

#Build

library(randomForest)

rf\_model <- randomForest(Price ~ Year + Mileage + Condition + Make + Model, data = train\_data)

rf\_model

random\_forest\_plot<-plot(randomForest(Price ~ Year + Mileage + Condition + Make + Model, data = CarPrediction))

random\_forest\_plot

varImpPlot(rf\_model, main = "plot varimp")

importance(rf\_model)

#predictions

rf\_predictions <- predict(rf\_model, newdata = test\_data)

head(rf\_predictions)

##evaluation

# MAE

MAE3 <- mean(abs(rf\_predictions - test\_data$Price))

MAE3

#MSE

MSE3 <- mean((rf\_predictions - test\_data$Price)^2)

MSE3

# RMSE

RMSE3<- sqrt(MSE3)

RMSE3

#R squared

r\_Squared3 <- 1 - sum((test\_data$Price - rf\_predictions)^2) / sum((test\_data$Price - mean(test\_data$Price))^2)

r\_Squared3

MAE3

RMSE3

r\_Squared3

#create plot

rf\_prediction\_dataframe <- data.frame(Actual = test\_data$Price, Predicted = rf\_predictions)

ggplot(rf\_prediction\_dataframe, aes(x = Actual, y = Predicted)) +

geom\_point() +

labs(title = "Actuals and Predicted values of Price",

x = "Actual value of Price",

y = "Predicted value of Price") +

theme\_minimal()

# MAE plot

MAE\_plot<-ggplot() +geom\_col(data = data.frame(Model = c("mlr\_model", "tree\_model", "rf\_model"),

MAE = c(MAE1, MAE2, MAE3)),

aes(x = Model, y = MAE, fill = Model), position = "dodge") +

labs(title = "Comparison of MAE",

x = "My Models",

y = "MAE") +

theme\_minimal()

# RMSE plot

RMSE\_plot<-ggplot() +geom\_col(data = data.frame(Model = c("mlr\_model", "tree\_model", "rf\_model"),

RMSE = c(RMSE1, RMSE2, RMSE3)),

aes(x = Model, y = RMSE, fill = Model),position = "dodge") +

labs(title = "Comparison of RMSE",

x = "My Models",

y = "RMSE") +

theme\_minimal()

# R-squared

Rsquared\_plot<-ggplot(data = data.frame(Model = c("mlr\_model", "tree\_model", "rf\_model"),

R\_squared = c(R\_squared1, R\_squared2, r\_Squared3))) +

geom\_col(aes(x = Model, y = R\_squared, fill = Model), position = "dodge") +

labs(title = "Comparison R-squared values",

x = " My Models",

y = "R-squared") +

theme\_minimal()

grid.arrange(MAE\_plot,RMSE\_plot,Rsquared\_plot,ncol=2)

# MAPE of my models

mape\_mlrvalues <- 1 - mean(abs(mlr\_prediction - test\_data$Price)) / mean(test\_data$Price)

mape\_decisiontreevalues <- 1 - mean(abs(decisiontree\_prediction - test\_data$Price)) / mean(test\_data$Price)

mape\_randomforestvalues <- 1 - mean(abs(rf\_predictions - test\_data$Price)) / mean(test\_data$Price)

data\_numberaccuracy <- data.frame(Model = c("mlr\_model", "tree\_model", "rf\_model"),

Accuracy = c(mape\_mlrvalues, mape\_decisiontreevalues, mape\_randomforestvalues),

R\_squared = c(R\_squared1, R\_squared2, r\_Squared3))

ggplot(data = data\_numberaccuracy, aes(x = Model)) +

geom\_col(aes(y = Accuracy, fill = "Accuracy"), position = "dodge") +

geom\_col(aes(y = R\_squared, fill = "R\_squared"), position = "dodge") +

labs(title = "Compare Accuracy and R-squared",

x = "My Model",

y = "The Value") +

scale\_fill\_manual(values = c(Accuracy = "pink", R\_squared = "purple")) +

theme\_minimal()