setwd("C:/Users/awaldert/Desktop/Data\_Science")

# Installing needed pqackages

library(dplyr)

library(rpart)

library(ggplot2)

library(caTools)

# Learning how ML works using the diamionds Dataset

View(diamonds)

str(diamonds)

dim(diamonds)

length(diamonds$price)

# Splitting the data into train and test sets

sample.split(diamonds$price, SplitRatio = 0.65) -> Split\_Values

subset(diamonds, Split\_Values==T) -> Train\_Set

subset(diamonds, Split\_Values==F) -> Test\_Set

head(Train\_Set)

dim(Train\_Set)

dim(Test\_Set)

# Building a linear model on top of the training dataset

lm(price~., Train\_Set) -> mod\_regress #Lets use stepwise backward regression to optimize our model and compare to the RMSE obtained before

predict(mod\_regress, Test\_Set) -> result\_regress

cbind(Actual = Test\_Set$price, Predicted = result\_regress) -> Final\_Data

as.data.frame(Final\_Data)-> Final\_Data

head(Final\_Data)

# Finding theb RMSE (Root Mean Squared Error)

(Final\_Data$Actual- Final\_Data$Predicted) -> error1

cbind(Final\_Data, error) -> Final\_Data

Initial\_RMSE <- sqrt(mean(Final\_Data$error^2))

Initial\_RMSE

########################################################################Optional########################################################################################

# susing stepwise backward regression to optimize our model and compare to the RMSE obtained before

reduced.mod.regress <- step(mod\_regress, direction = "backward")

predict(reduced.mod.regress, Test\_Set) -> red.result\_regress

cbind(Actual = Test\_Set$price, Predicted = red.result\_regress) -> Final\_Data.red.mod

as.data.frame(Final\_Data.red.mod)-> Final\_Data.red.mod

head(Final\_Data.red.mod)

setwd("C:/Users/awaldert/Desktop/Data\_Science")

# Installing needed pqackages

library(dplyr)

library(rpart)

library(ggplot2)

library(caTools)

# Learning how ML works using the diamionds Dataset

View(diamonds)

str(diamonds)

dim(diamonds)

length(diamonds$price)

# Splitting the data into train and test sets

sample.split(diamonds$price, SplitRatio = 0.65) -> Split\_Values

subset(diamonds, Split\_Values==T) -> Train\_Set

subset(diamonds, Split\_Values==F) -> Test\_Set

head(Train\_Set)

dim(Train\_Set)

dim(Test\_Set)

# Building a linear model on top of the training dataset

lm(price~., Train\_Set) -> mod\_regress #Lets use stepwise backward regression to optimize our model and compare to the RMSE obtained before

predict(mod\_regress, Test\_Set) -> result\_regress

cbind(Actual = Test\_Set$price, Predicted = result\_regress) -> Final\_Data

as.data.frame(Final\_Data)-> Final\_Data

head(Final\_Data)

# Finding theb RMSE (Root Mean Squared Error)

(Final\_Data$Actual- Final\_Data$Predicted) -> error1

cbind(Final\_Data, error) -> Final\_Data

Initial\_RMSE <- sqrt(mean(Final\_Data$error^2))

Initial\_RMSE

########################################################################Optional########################################################################################

# susing stepwise backward regression to optimize our model and compare to the RMSE obtained before

reduced.mod.regress <- step(mod\_regress, direction = "backward")

predict(reduced.mod.regress, Test\_Set) -> red.result\_regress

cbind(Actual = Test\_Set$price, Predicted = red.result\_regress) -> Final\_Data.red.mod

as.data.frame(Final\_Data.red.mod)-> Final\_Data.red.mod

head(Final\_Data.red.mod)

plot(Final\_Data.red.mod)

# Finding theb RMSE (Root Mean Squared Error) of the reduced model

(Final\_Data.red.mod$Actual- Final\_Data.red.mod$Predicted) -> error2

cbind(Final\_Data.red.mod, error2) -> Final\_Data\_Reduced\_Model

Reduced\_RMSE <- sqrt(mean(Final\_Data\_Reduced\_Model$error2^2))

Reduced\_RMSE

Initial\_RMSE

summary(reduced.mod.regress)

summary(mod\_regress)

# Comparing RSME based on differenc combinations and trying stepwise forward regression to optimize the model

# Possible Option to eliminate the effect of outliers by curring bottom and top 10% of the dataset

# Finding theb RMSE (Root Mean Squared Error) of the reduced model

(Final\_Data.red.mod$Actual- Final\_Data.red.mod$Predicted) -> error2

cbind(Final\_Data.red.mod, error2) -> Final\_Data\_Reduced\_Model

Reduced\_RMSE <- sqrt(mean(Final\_Data\_Reduced\_Model$error2^2))

Reduced\_RMSE

Initial\_RMSE

summary(reduced.mod.regress)

summary(mod\_regress)

plot(Final\_Data\_Reduced\_Model)

par(mfrow =c(1, 2))

# Comparing RSME based on differenc combinations and trying stepwise forward regression to optimize the model

# Possible Option to eliminate the effect of outliers by curring bottom and top 10% of the dataset