```r
rm(list=ls())

setwd("C:/Users/Debayan Chakraborty/Documents/Edwisor Cab project R")

getwd()

load_lib = c("ggplot2", "corrgram", "DMwR", "usdm", "randomForest", "plyr",
"dplyr", "DataCombine", "inTrees", "rpart", "rpart.plot", "geosphere",
"DataCombine", "MASS", "miscTools","stats", "caret")

lapply(load_lib, install.packages)

lapply(load_lib, require, character.only = TRUE)

#In the above codes, fistly we have cleaned the R environment, secondly we set
our working directory and finally installed and loaded the required libraries.
Now we will be extracting the required csv file and perform exploratory data
analysis on it#

Rtrain = read.csv("train_cab.csv", header = T, sep = ",", na.strings = c("", " ",
"NA"))
Rtest = read.csv("test.csv", header = T, sep = ",")

#Exploratory data analysis#

View(Rtrain)

dim(Rtrain)

View(Rtest)
dim(Rtest)

str(Rtrain)

##Feature Engineering##

#Data type conversion#

Rtrain$passenger_count = as.factor(Rtrain$passenger_count)

Rtrain$fare_amount = as.numeric(as.character(Rtrain$fare_amount))

Rtrain$pickup_datetime <- gsub('\\ UTC','',Rtrain$pickup_datetime)

Rtrain$pickup_datetime = as.Date(Rtrain$pickup_datetime)
Rtrain$Year = substr(as.character(Rtrain$pickup_datetime),1,4)
Rtrain$Month = substr(as.character(Rtrain$pickup_datetime),6,7)
Rtrain$Date = substr(as.character(Rtrain$pickup_datetime),9,10)
Rtrain$Hour = substr(as.character(Rtrain$pickup_datetime),12,13)
Rtrain$Minute = substr(as.character(Rtrain$pickup_datetime),15,16)

#Replicating the same thing to test

Rtest$pickup_datetime <- gsub('\\ UTC','',Rtest$pickup_datetime)

Rtest$pickup_datetime = as.Date(Rtest$pickup_datetime)
```

```
plong = Rtrain['pickup_longitude']
dlong = Rtrain['dropoff_longitude']

rangeR = function(plong, plat, dlong, dlat) {
  R = 6371.145
  del_long = (dlong - plong)
  del_lat = (dlat - plat)
  a = sin(del_lat/2)^2 + cos(plat) * cos(dlat) * sin(del_long/2)^2
  c = 2 * atan2(sqrt(a),sqrt(1-a))
  rangeR = R * c
  return(rangeR)

}

for (i in 1:nrow(Rtrain))
{
  Rtrain$rangeR[i]= rangeR(Rtrain$pickup_longitude[i], Rtrain$pickup_latitude[i],
Rtrain$dropoff_longitude[i],
                         Rtrain$dropoff_latitude[i])
}

for (i in 1:nrow(Rtest))
{
  Rtest$rangeR[i]= rangeR(Rtest$pickup_longitude[i], Rtest$pickup_latitude[i],
Rtest$dropoff_longitude[i],
                             Rtest$dropoff_latitude[i])
}

#Outlier analysis#
#While coding in python that outliers are not removed properly by applying the
IQR formula
#Hence here also we will manually remove the outliers varaible wise#


Rtrain$fare_amount[Rtrain$fare_amount<1] = NA
Rtrain$fare_amount[Rtrain$fare_amount>453] = NA

sum(is.na(Rtrain))

Rtrain = DropNA(Rtrain)

sum(is.na(Rtrain))

#Outlier removal of passenger count#

Rtrain$passenger_count[Rtrain$passenger_count<1] = NA
Rtrain$passenger_count[Rtrain$passenger_count>6] = NA

sum(is.na(Rtrain))
Rtrain = DropNA(Rtrain)

sum(is.na(Rtrain))

#Outlier analysis of
Rtrain$rangeR[Rtrain$rangeR<0.1] = NA
Rtrain$rangeR[Rtrain$rangeR>150] = NA
```

```
#fare amount Vs date#

ggplot(data = Rtrain, aes(x = Date, y = fare_amount))+
  geom_bar(stat = "identity")+
  labs(title = "Fare Amount Vs. date", x = "Date", y = "Fare")+
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))+
  theme(axis.text.x = element_text( color="blue", size=6, angle=45))

#fare amount vs Hour

ggplot(data = Rtrain, aes(x = Hour, y = fare_amount))+
  geom_bar(stat = "identity")+
  labs(title = "Fare Amount Vs. hour", x = "Hour", y = "Fare")+
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))+
  theme(axis.text.x = element_text( color="blue", size=6, angle=45))

#fare amount vs Month

ggplot(data = Rtrain, aes(x = Month, y = fare_amount))+
  geom_bar(stat = "identity")+
  labs(title = "Fare Amount Vs. month", x = "Month", y = "Fare")+
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))+
  theme(axis.text.x = element_text( color="blue", size=6, angle=45))


#fare amount vs passenger count

ggplot(data = Rtrain, aes(x = passenger_count, y = fare_amount))+
  geom_bar(stat = "identity")+
  labs(title = "Fare Amount Vs. passenger count", x = "passenger count", y =
"Fare")+
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))+
  theme(axis.text.x = element_text( color="blue", size=6, angle=45))


##Feature selection##

numeric_index = sapply(Rtrain,is.numeric)

numeric_data = Rtrain[,numeric_index]

cnames = colnames(numeric_data)

#Correlation analysis for numeric variables

corrgram(Rtrain[,numeric_index],upper.panel=panel.pie, main = "Correlation Plot")



##removing the unnecessary variables#

Rtrain = subset(Rtrain, select = -
c(pickup_datetime,pickup_latitude,dropoff_latitude,pickup_longitude,dropoff_longi

Rtest = subset(Rtest, select = -
c(pickup_datetime,pickup_latitude,dropoff_latitude,pickup_longitude,dropoff_longi
str(Rtrain)

#feature scaling#

hist(Rtrain$rangeR)

for(i in rangeR){

  print(i)
  Rtrain[,i] = (Rtrain[,i] - min(Rtrain[,i]))/(max(Rtrain[,i])-min(Rtrain[,i]))

 }

hist(Rtrain$rangeR)

#Modelling#

#Train test split#

set.seed(123)
split_set = createDataPartition(Rtrain$fare_amount, p = 0.8, list = FALSE)
trainset = Rtrain[split_set,]
testset  = Rtrain[-split_set,]

#Liner regression#
lrgmodel = lm(fare_amount ~.,data = trainset)

summary(lrgmodel)

#predict for test#

predlrg_test = predict(lrgmodel, testset)

predlrg_test

regr.eval(trues = testset, predlrg_test, stats = c("mae","mse", "rmse", "mape"))

#Decision Tree#

dtreemodel = rpart(cnt~.,trainset, method = "anova")
dtreepreds = predict(dtreemodel, testset)
dtreepreds

regr.eval(trues = testset, dtreepreds, stats = c("mae","mse","rmse","mape"))

rpart.plot(dtreemodel,type = 3, digits = 3, fallen.leaves = TRUE)

#Random Forest#

rforestmodel = randomForest(cnt~., trainset)
rforestpreds = predict(rforestmodel, testset)
rforestpreds
regr.eval(trues = testset,rforestpreds, stats = c("mae", "mse", "rmse","mape"))
```