DMPM Assignment 4

Name: Rushikesh Jyoti

Division: A

Roll no: 27

SRN: 201901139

Question

- Read the dataset **"FlightDelays.csv"** that is provided to you.
- Build a suitable logistic regression model using R and predict the status of the flights in your test data set.
- Draw the ROCR and Lift curves of your model and comment on the effectiveness of your model.
- Develop some metrics to determine the accuracy of your classification model (Error % using confusion matrix)

Code

```
library(dplyr)

library(caret)

library(reshape2)

library(pROC)

library(corrplot)

library(caTools)
```

```
flight = read.csv("FlightDelays.csv")
head(flight)
summary(flight)
str(flight)

table(flight$delay)
flight = flight %>% mutate(delay = ifelse(delay == "ontime",0,1))



# We need to convert categorical data to numeric data aka encoding the data
encode_category = function(x, order = unique(x)) {
   as.numeric(factor(x, levels = order, exclude = NULL))
}



flight[["tailnu"]]  = encode_category(flight[["tailnu"]])
flight[["dest"]]    = encode_category(flight[["dest"]])
flight[["origin"]]  = encode_category(flight[["origin"]])
flight[["carrier"]] = encode_category(flight[["carrier"]])

# We don't really need date
flight = select(flight, -date)
```

```
head(flight)



# Split the data for training and testing sets
set.seed(101)
sample = sample.split(flight$delay, SplitRatio = .70)
train  = subset(flight, sample == TRUE)
test   = subset(flight, sample == FALSE)

head(test)

# Plot the correlation heat map
corrplot(cor(train), tl.col="black")



# Build the model
logreg = glm(delay ~ ., family = binomial(link = 'logit'),
             data = train)
summary(logreg)

# Predict the values using the model
prob = logreg %>% predict(test_new, type = "response")
```

```
test$prob = prob
threshold = 0.3

# If prediction is less than threshold then put 0 otherwise 1
test = test %>% mutate(predicted = ifelse(prob < threshold,0,1))
head(test_new)

# The confusion matrix
mat = table(test$delay, test$predicted)
mat

# Metrics to check efficiency of model
accuracy   = (mat[1] + mat[4]) / (sum(mat))
error_rate = 1 - accuracy
precision  = mat[1] / (mat[1] + mat[3])
recall     = mat[1] / (mat[1] + mat[2])

cat("Accuracy: ", accuracy * 100,
    "%\nError Rate:", error_rate * 100,
    "%\nPrecision: ",precision * 100,
    "%\nRecall:",recall * 100,"%")

# ROCR curve
```

roc = roc(test$delay ~ prob, plot = TRUE, print.auc = TRUE)

Output

Taking a look at the dataset

```
> flight = read.csv("FlightDelays.csv")
> head(flight)
  schedtime carrier deptime dest distance     date flightnumber origin weather
1      1455      OH    1455  JFK      184 1/1/2004         5935    BWI       0
2      1640      DH    1640  JFK      213 1/1/2004         6155    DCA       0
3      1245      DH    1245  LGA      229 1/1/2004         7208    IAD       0
4      1715      DH    1709  LGA      229 1/1/2004         7215    IAD       0
5      1039      DH    1035  LGA      229 1/1/2004         7792    IAD       0
6       840      DH     839  JFK      228 1/1/2004         7800    IAD       0
  dayweek daymonth tailnu  delay
1       4        1 N940CA ontime
2       4        1 N405FJ ontime
3       4        1 N695BR ontime
4       4        1 N662BR ontime
5       4        1 N698BR ontime
6       4        1 N687BR ontime
```

We need to predict if flights are delayed or not, so let's look at delays in the dataset

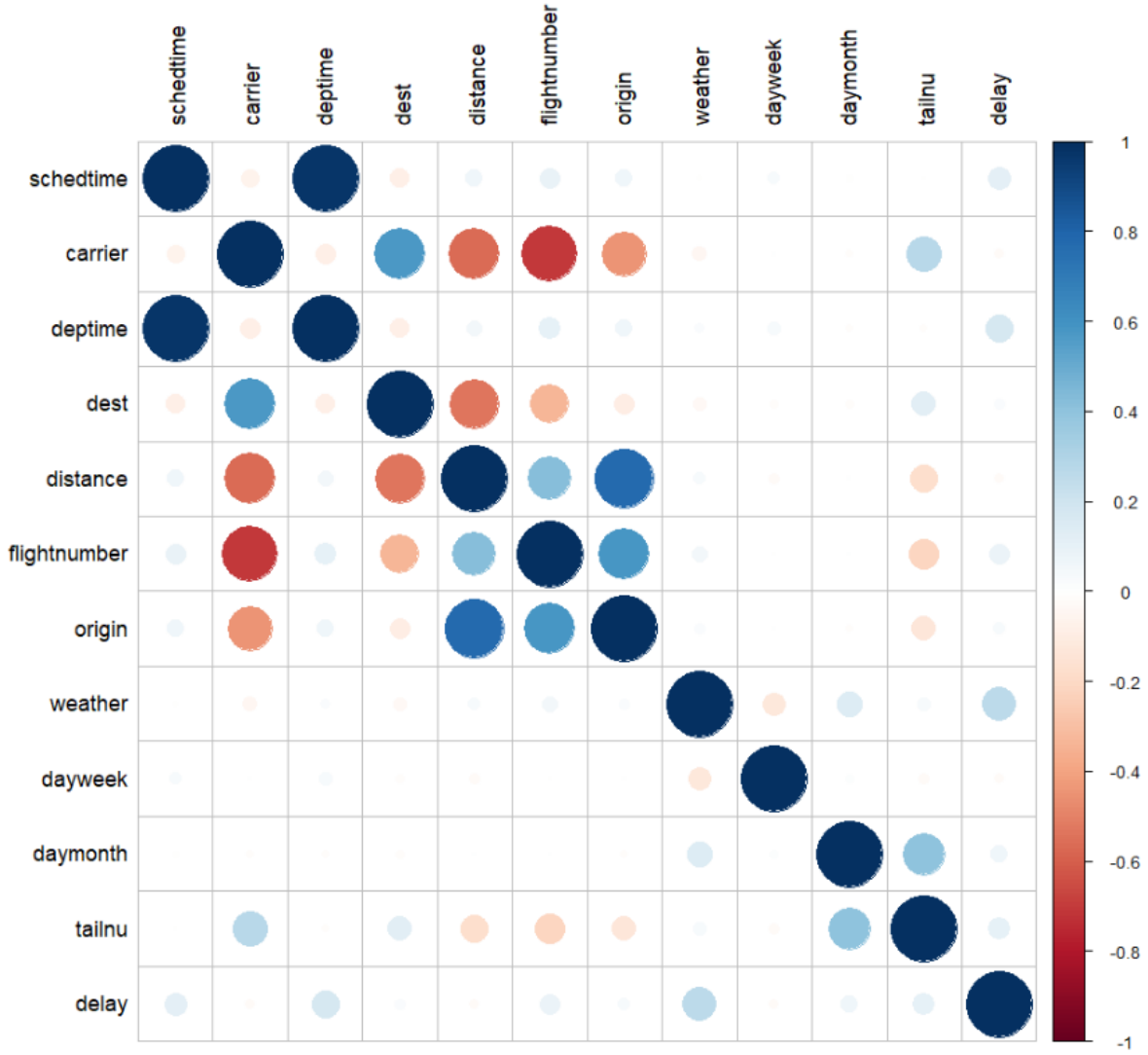
```
> table(flight$delay)

delayed  ontime
    428    1773
```

After encoding the categorical variables,

```
> head(flight)
  schedtime carrier deptime dest distance flightnumber origin weather dayweek
1      1455       1    1455    1      184         5935      1       0       4
2      1640       2    1640    1      213         6155      2       0       4
3      1245       2    1245    2      229         7208      3       0       4
4      1715       2    1709    2      229         7215      3       0       4
5      1039       2    1035    2      229         7792      3       0       4
6       840       2     839    1      228         7800      3       0       4
  daymonth tailnu delay
1        1      1     0
2        1      2     0
3        1      3     0
4        1      4     0
5        1      5     0
6        1      6     0
```

Taking a look at correlation between variables



Split the dataset into 70-30

```
> sample = sample.split(flight$delay, SplitRatio = .70)
> train  = subset(flight, sample == TRUE)
> test   = subset(flight, sample == FALSE)
> count(train)
     n
1 1541
> count(test)
    n
1 660
```

Building the model

```
> logreg = glm(delay ~ ., family = binomial(link = 'logit'),
+              data = train)
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> summary(logreg)

Call:
glm(formula = delay ~ ., family = binomial(link = "logit"), data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1718  -0.5393  -0.4399  -0.3273   8.4904

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.032e+01  3.118e+00  -3.312 0.000927 ***
schedtime    -2.297e-02  1.947e-03 -11.797  < 2e-16 ***
carrier       2.097e-01  6.739e-02   3.112 0.001859 **
deptime       2.349e-02  1.930e-03  12.173  < 2e-16 ***
dest          2.766e-01  1.953e-01   1.416 0.156714
distance      3.134e-02  1.554e-02   2.017 0.043721 *
flightnumber  1.612e-04  5.882e-05   2.740 0.006143 **
origin       -5.895e-01  3.686e-01  -1.599 0.109762
weather       1.691e+01  3.941e+02   0.043 0.965771
dayweek      -1.837e-02  4.045e-02  -0.454 0.649723
daymonth      6.893e-03  9.907e-03   0.696 0.486588
tailnu        2.189e-03  5.890e-04   3.716 0.000202 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1519.2  on 1540  degrees of freedom
Residual deviance: 1092.0  on 1529  degrees of freedom
AIC: 1116

Number of Fisher Scoring iterations: 15
```

Predicting the values

```
> prob = logreg %>% predict(test, type = "response")
> head(select(test, prob))
         prob
2  0.08711047
12 0.04673596
13 0.05515797
19 0.02414693
21 0.03662941
25 0.06056722
```

The confusion matrix

```
> # The confusion matrix
> mat = table(test$delay, test$predicted)
> mat

     0   1
  0 520  12
  1  42  86
```

Metrics of the model

```
Accuracy:  91.81818 %
Error Rate: 8.181818 %
Precision:  97.74436 %
Recall: 92.52669 %
```

Sidenote: woho! Those are some good numbers!

ROC Curve