R Project Classification

Reading data

```
df <- read.csv("fedex.csv")
#making data frame smaller since original is 3.6 million rows (selecting 50,000 rows randomly)
df1 <- df[sample(nrow(df), 50000), ]

#https://www.kaggle.com/manishtripathi86/fedex-data
#We are predicing wheter the shipment was delayed or not (0 being not delayed, 1 being delayed)

#summary(df)
```

Data Exploration: We see that shipment delay has a large impact on whether it was delayed. The residuals are not as random as we would like and have a flat pattern.

```
#sapply(df1, function(x) sum(is.na(x)==TRUE))
#dataframe for all nas omitted, omitted them instead of using median/mean because the target(delivery s
dfOmit <- na.omit(df1)

dfOmit$Delivery_Status <- factor(dfOmit$Delivery_Status)
dfOmit$Carrier_Name <- factor(dfOmit$Carrier_Name)
dfOmit$Source <- factor(dfOmit$Source)
dfOmit$Destination <- factor(dfOmit$Destination)
#get rid of year column because all are in 2008
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
dfOmit <- dfOmit %>%
mutate(Year = NULL)

dfOmit$Carrier_Name <- as.integer(as.factor(dfOmit$Carrier_Name))
dfOmit$Source <- as.integer(as.factor(dfOmit$Source))
dfOmit$Destination <- as.integer(as.factor(dfOmit$Destination))
```

```
set.seed(1234)
#i <- sample(1:nrow(dfOmit)*0.75,replace=FALSE)
#train <- dfOmit[i,]
#test <- dfOmit[-i,]

ind <- sample(2,nrow(dfOmit),replace=TRUE,prob =c(.75,.25))
train <- dfOmit[ind==1,1:14]
test <- dfOmit[ind==2, 1:14]
trainLabels <- dfOmit[ind==1,14]
testLabels <- dfOmit[ind==2,14]


#sapply(lapply(dfOmit, unique), length)
```

Logistic Regression

```
glm1 <- glm(Delivery_Status~Carrier_Name+Carrier_Num+Shipment_Delay+Month+DayofMonth+DayOfWeek+Shipment
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm1)
```

```
##
## Call:
## glm(formula = Delivery_Status ~ Carrier_Name + Carrier_Num +
##     Shipment_Delay + Month + DayofMonth + DayOfWeek + Shipment_Delay,
##     family = "binomial", data = train)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -5.049e-04  -2.000e-08  -2.000e-08  -2.000e-08   5.631e-04
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.965e+02  2.413e+03  -0.206    0.837
## Carrier_Name   -6.612e-03  1.317e+01  -0.001    1.000
## Carrier_Num    -5.237e-05  4.318e-02  -0.001    0.999
## Shipment_Delay  3.205e+01  1.546e+02   0.207    0.836
## Month          -2.038e-02  4.562e+01   0.000    1.000
## DayofMonth     -6.329e-04  8.756e+00   0.000    1.000
## DayOfWeek      -7.630e-03  3.930e+01   0.000    1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3.7274e+04  on 36728  degrees of freedom
## Residual deviance: 1.2334e-04  on 36722  degrees of freedom
## AIC: 14
##
## Number of Fisher Scoring iterations: 25
```

```
library(ROCR)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
pred <- predict(glm1, newdata=test, type = "response")

pr <- ifelse(pred > 1.5, 2, 1)
prMatrix <- ifelse(pred > 1.5,1,0)#prediction for matrix
acc1 <- mean(pr==as.integer(test$Delivery_Status))
print(paste("glm1 accuracy = ",acc1))
```

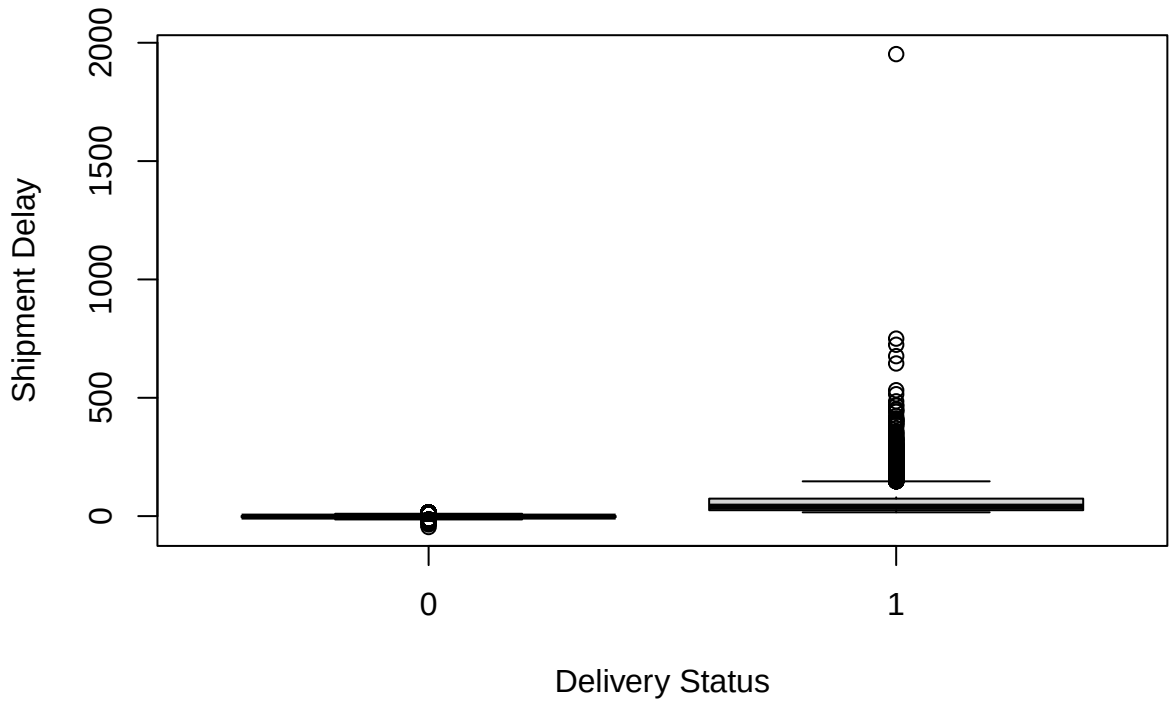
```
## [1] "glm1 accuracy =  0.798383105098169"
```

```
confusionMatrix(as.factor(prMatrix),test$Delivery_Status,positive="0")
```

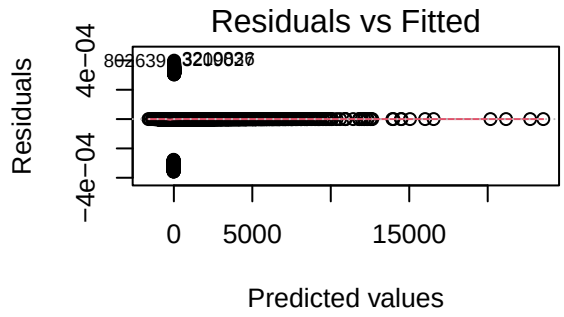
```
## Warning in confusionMatrix.default(as.factor(prMatrix), test$Delivery_Status, :
## Levels are not in the same order for reference and data. Refactoring data to
## match.
```

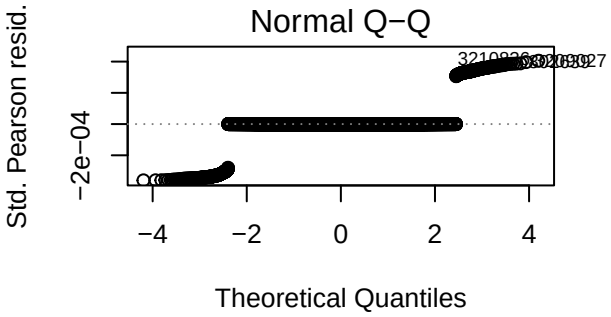
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 9678 2444
##          1    0    0
##
##                Accuracy : 0.7984
##                  95% CI : (0.7911, 0.8055)
##     No Information Rate : 0.7984
##     P-Value [Acc > NIR] : 0.5054
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.7984
##          Neg Pred Value :    NaN
##              Prevalence : 0.7984
##          Detection Rate : 0.7984
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

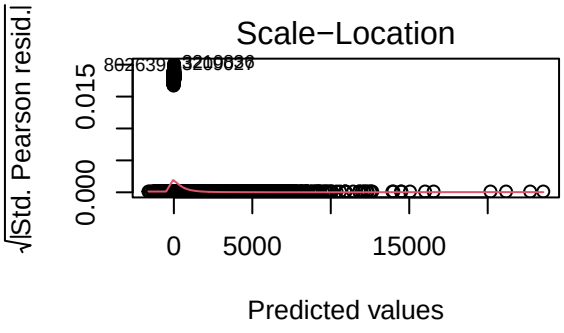
```
plot(dfOmit$Shipment_Delay~dfOmit$Delivery_Status,xlab = "Delivery Status", ylab = "Shipment Delay" )
```

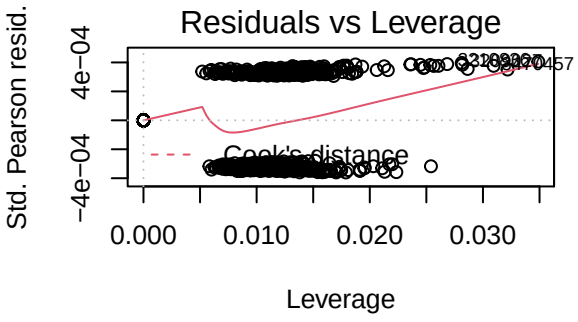



```
par(mfrow=c(2,2))
plot(glm1)
```









kNN

```
library(class)

#trainLabels <- as.integer(trainLabels)
#testLabels <- as.integer(testLabels)


predKnn <- knn(train = train, test = test, cl = trainLabels, k = 5)
results1 <- predKnn == testLabels
acc2 <- length(which(results1==TRUE)) / length(results1)
cat("The accuracy was ", acc2)
```

```
## The accuracy was  0.839053
```

SVM

```
library(e1071)

svm1 <- svm(Delivery_Status~Carrier_Name+Carrier_Num+Shipment_Delay+Month+DayofMonth+DayOfWeek+Shipment_
cost=10, scale=TRUE)
pred2 <- predict(svm1, newdata=test)
table(pred2, test$Delivery_Status)
```

```
##
```

```
## pred2    0    1
##     0 9678    0
##     1    0 2444
```

```
mean(pred2==test$Delivery_Status)
```

```
## [1] 1
```

Result Analysis: The best performing algorithm was the SVM, then kNN, and lastly logistic regression. k = 5 gave the best accuracy for the kNN model. From the big picture we can see that Shipment_Delay was one of the best predictors for predicting whether a shipment would be delayed or not.