Predicting Flight Status

Shilpi Jaiswal

June 2, 2016

* Introduction

Perhaps the most frustrating and costly event that an airplane traveler may face is not purchasing a plane ticket, but having to frantically scramble and rearrange plans when they arrive at the airport, only to find that their scheduled flight has been delayed, or cancelled altogether. Stories are often told of airplane travelers waiting multiple days for delayed flights, in which they are forced to rebook their flight and find nearby shelter in the meantime. However, many airlines themselves are the ones that finance these accommodations. Other times, when the length of delay isn't planned to be more than a day or so, these travelers end up spending their wait time in the airport itself. In these instances, it isn't just the traveler that is faced with having to spend valuable time and money rearranging, but the airport and the airline companies. This makes knowing of a flight?s status ahead of time a valuable tool for all parties involved.

Whatever the case may be, we believe that there are instances when delays and cancellations are predictable and thus some financial damage can be avoided, especially with the massive amount of data collected by airports. With that being said, effectively analyzing the data requires in-depth technical skills, and if done properly, can have significant business implications. Since flight delays and cancellations tend to be the result of poor weather conditions and technical issues surrounding the planes and airports, and since much of this data is generally publicly available, it is important for someone in the airline business to gather and analyze the data effectively so as to predict future occurrences and lead to better decision making overall.

Our intent of this study is to do just that; we have a single city dataset for all flights leaving New York City airports in 2013, and we hope to build a trustworthy model from it that can be used to better plan airplane travels. In this project, we thoroughly analyze the data to find the key contributors to flight cancellations and delays in order to evaluate and predict the probability that future flights will be cancelled and/or delayed. Also, we recognize the important business potential that an accurate predictive algorithm for this type of data holds. If we were able to build a precise model, then it has potential for widespread usability for any airline customer, possibly in the form of a smartphone application (app). An app of this sort may be able to link live weather updates and other crucial sources of data to help travelers plan their flights accordingly. For instance, if the data is plugged into the algorithm and it is found that a certain flight has a significantly high probability of cancellation, then the traveler can see this and arrange ahead of time, rather than having to scramble to change plans at the last minute, when in fact it may already be too late to do so.

* Data, Methodology and Results

Here are the packages used.

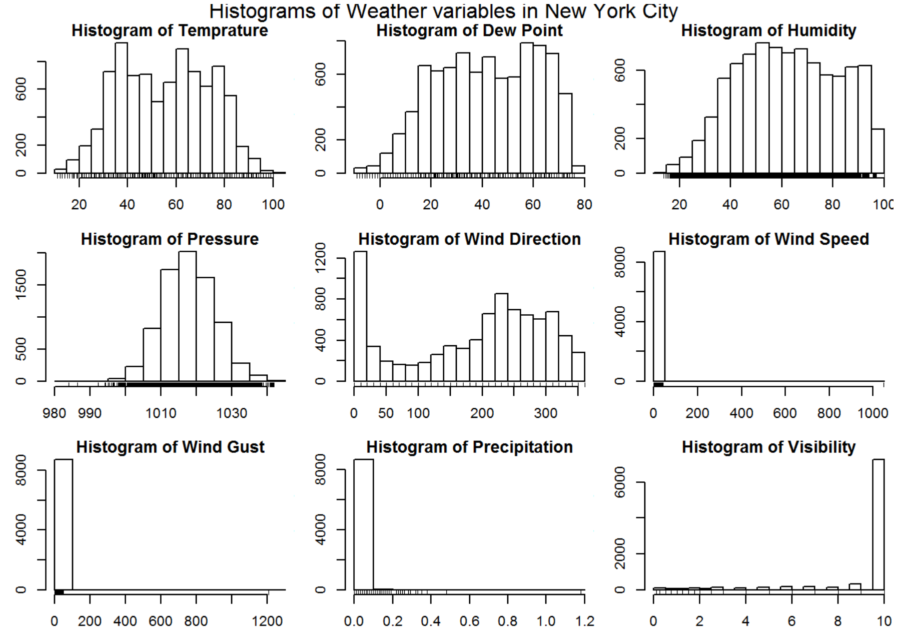
```r
library(RSQLite)
library(dplyr)
library(nycflights13)
library(mfx)
library(ggplot2)
library(scales)
library(MASS)
library(ISLR)
library(class)
library(boot)
library(stargazer)
library(tree)
library(randomForest)
library(leaps)
library(glmnet)
library(gbm)
```

The Data used is the nycflight13, which contains data regarding all flight originating from NYC including weather readings in 2013. "cancelation" is defined as 1 if "air time" is N/A, which indicates the flight never took off. Flight is considered delayed if "departure delay" is greater than 15 minutes, which is the industry standard. These two are categorical variables.

```
db<-nycflights13_sqlite()
flights_sqlite<- tbl(db, "flights")
flights_sqlite<-flights_sqlite %>%
  mutate(canceled = if(is.na(arr_time)){1}else{0},
         takeoffdelay = if(dep_delay>15){1}else{0})
weather_sqlite <- tbl(db, "weather")
planes_sqlite <- tbl(db, "planes")
planes_sqlite <- planes_sqlite %>% dplyr::rename(manuyear=year)
flights <- flights_sqlite %>% collect()
planes <- planes_sqlite %>% collect()
weather <- weather_sqlite %>% collect()
```

---

* Exploring raw data



Clearly from above histograms we can see there are outliers or some anomalies in wind speed, wind gust, precip and visib

```
ordered_weather <- arrange(weather, desc(wind_speed), wind_gust, precip, visib)
head(ordered_weather, 5)
```

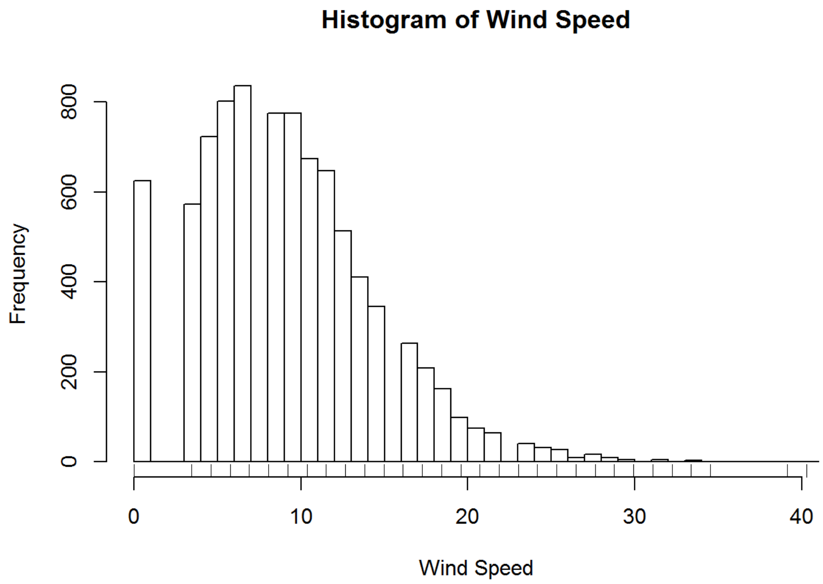
```
## Source: local data frame [5 x 14]
## 
##   origin  year month   day  hour  temp  dewp humid wind_dir wind_speed
##    (chr) (dbl) (dbl) (int) (int) (dbl) (dbl) (dbl)    (dbl)      (dbl)
## 1    EWR  2013     2    12     8 39.02 26.96 61.63      260 1048.36058
## 2    EWR  2013     1    31     9 60.80 59.00 93.79      230   40.27730
## 3    EWR  2013     1    31    13 46.04 30.02 53.33      270   39.12652
## 4    EWR  2013     6    25    20 89.60 66.20 46.14      270   34.52340
## 5    EWR  2013     1    31    15 44.96 21.02 38.23      260   33.37262
## Variables not shown: wind_gust (dbl), precip (dbl), pressure (dbl), visib
##   (dbl)
```

We can see there is one observation with an anomolous wind_speed so we proceed to remove it along with the precip and visib column which have similar values for all rows.

```
weather <- subset(ordered_weather, wind_speed<1000, select = -c(visib, precip))
```

Observe wind_speed after removal of outlier

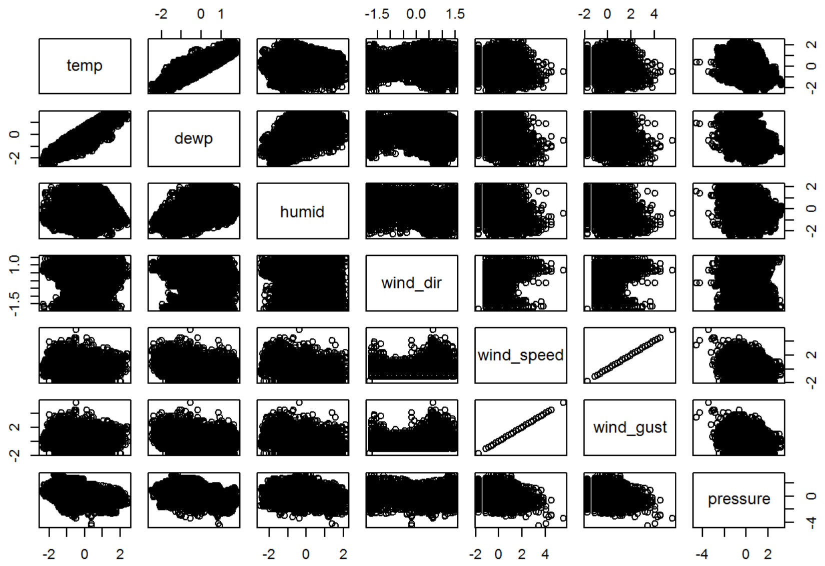
```
hist(weather$wind_speed, breaks = 50, main = "Histogram of Wind Speed", xlab = "Wind Speed")
rug(weather$wind_speed)
```



Now lets check for corrlelation between weather variables

```
##                   temp       dewp       humid    wind_dir  wind_speed
## temp        1.00000000  0.8985373  0.05595643 -0.08279157 -0.09403104
## dewp        0.89853726  1.0000000  0.48037637 -0.21505249 -0.25621428
## humid       0.05595643  0.4803764  1.00000000 -0.33822055 -0.39998386
## wind_dir   -0.08279157 -0.2150525 -0.33822055  1.00000000  0.44463185
## wind_speed -0.09403104 -0.2562143 -0.39998386  0.44463185  1.00000000
## wind_gust  -0.09403104 -0.2562143 -0.39998386  0.44463185  1.00000000
## pressure   -0.26826760 -0.3051828 -0.17789221 -0.19710396 -0.21852985
##              wind_gust   pressure
## temp       -0.09403104 -0.2682676
## dewp       -0.25621428 -0.3051828
## humid      -0.39998386 -0.1778922
## wind_dir    0.44463185 -0.1971040
## wind_speed  1.00000000 -0.2185298
## wind_gust   1.00000000 -0.2185298
## pressure   -0.21852985  1.0000000
```

Looks like temp amd dewp are highly correlated and wind speed and wind gust have a correlation coefficiemt of 1. Made more clear by below plots. The rest of the variables have no clear relationship.



* Data Cleaning

```
### Now lets proceed to join the flights, plane and weather data sets.
subtotal <- inner_join(flights, planes, by = "tailnum")

total <- inner_join(subtotal, weather, by=c( "origin", "year","month","day","hour"))
total <- subset(total, select=-c(hour, minute, dep_delay,arr_time,arr_delay,air_time, carrier, tailnum, flight, distance, ty
pe, manufacturer, model, engines, speed, dewp, wind_gust))

### Removing NA values
completefun <- function(data, desiredcols){
  completeVec <- complete.cases(data[ ,desiredcols])
  return(data[completeVec, ])
}

total <- completefun(total)
```

We can see that the table has rows for both cancelled and non cancelled flights

```
## Source: local data frame [10 x 16]
##
##     year month   day dep_time origin  dest canceled takeoffdelay manuyear
##    (dbl) (dbl) (int)    (int)  (chr) (chr)    (dbl)        (dbl)    (int)
## 1   2013     1     1     2016    EWR   OKC        1            1     2004
## 2   2013     1     2     2145    EWR   RSW        1            1     1998
## 3   2013     1    11     1344    EWR   MSN        1            0     2000
## 4   2013     1    13     2239    EWR   BWI        1            1     2005
## 5   2013     1    25     2010    EWR   GSO        1            1     2001
## 6   2013    10    31      625    EWR   HOU        1            0     2006
## 7   2013    11     1      855    EWR   LAX        1            0     2012
## 8   2013    11    14     1811    EWR   SAN        1            1     2008
## 9   2013    11    18     1310    EWR   CLT        1            0     2009
## 10  2013    11    19     1904    EWR   FLL        1            0     1998
## Variables not shown: seats (int), engine (chr), temp (dbl), humid (dbl),
##   wind_dir (dbl), wind_speed (dbl), pressure (dbl)
```

```
## Source: local data frame [10 x 16]
##
##     year month   day dep_time origin  dest canceled takeoffdelay manuyear
##    (dbl) (dbl) (int)    (int)  (chr) (chr)    (dbl)        (dbl)    (int)
## 1   2013     9    30     2114    EWR   MSP        0            1     2005
## 2   2013     9    30     2116    EWR   SDF        0            0     2001
## 3   2013     9    30     2119    EWR   MCI        0            1     2000
## 4   2013     9    30     2122    EWR   DCA        0            0     2001
## 5   2013     9    30     2127    EWR   CLT        0            0     2002
## 6   2013     9    30     2142    EWR   PWM        0            0     1998
## 7   2013     9    30     2149    EWR   BOS        0            0     1998
## 8   2013     9    30     2150    EWR   MHT        0            0     2002
## 9   2013     9    30     2211    EWR   STL        0            1     2003
## 10  2013     9    30     2233    EWR   SFO        0            1     1993
## Variables not shown: seats (int), engine (chr), temp (dbl), humid (dbl),
##   wind_dir (dbl), wind_speed (dbl), pressure (dbl)
```

Scaling data

```
total <- total %>% mutate( ID = c(1:nrow(total)))
weather.var <- subset(total, select = c(temp, humid, wind_dir, wind_speed, pressure))
flight.var <- subset(total, select = -c(temp, humid, wind_dir, wind_speed, pressure))
scale.matrix <- scale(weather.var)
scale.df <- data.frame(scale.matrix)

scale.id  <- scale.df %>% mutate(ID = c(1:nrow(scale.df)))

final.data <- left_join(flight.var, scale.id, by = "ID")

final.data <- subset(final.data, select = -ID)
```

Split data to Training and Validation set

```
test.ind <- sample(nrow(final.data ), 30000)
test <- final.data [test.ind,]
train <- final.data [-test.ind,]
```

Now that the main data set is ready, let’s remove unwanted tables from workspace

```
rm(weather_corr)
rm(subtotal)
rm(ordered_weather)
rm(planes)
rm(flights)
rm(weather)
rm(scale.id)
rm(scale.matrix)
rm(total)
```

* Data Visualization and Feature Selection

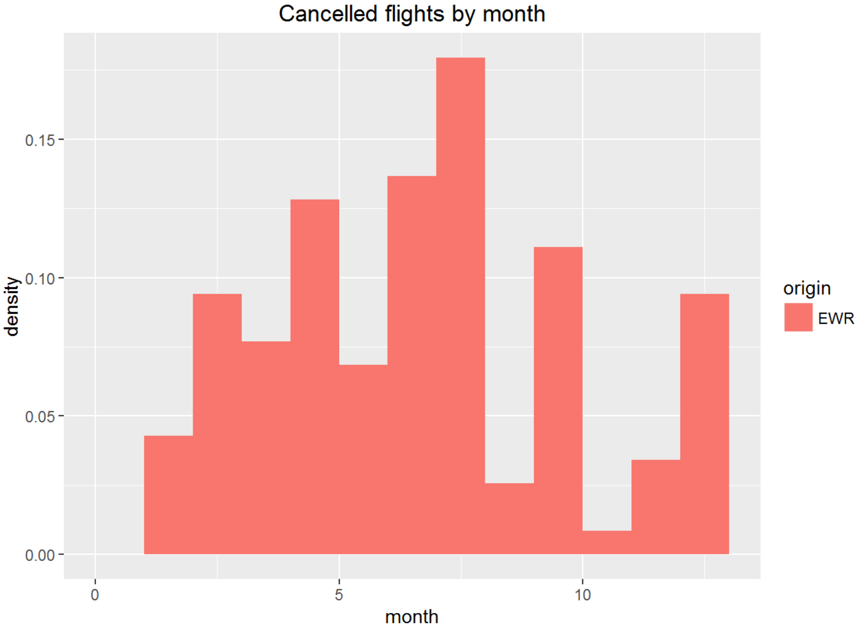
Here we check if time of the year has any effect on cancelation or delay, since it may correlate to weather conditions across seasons.

Histogram of Cancelation by Month.

```
Cancelled <- final.data[(final.data$canceled == 1),]

cancelmonthplot <- ggplot(
  data = Cancelled,
  aes(x = month,
      y = ..density.., fill = origin)) +
  geom_histogram(binwidth = 1) +
  ggtitle("Cancelled flights by month")

cancelmonthplot
```



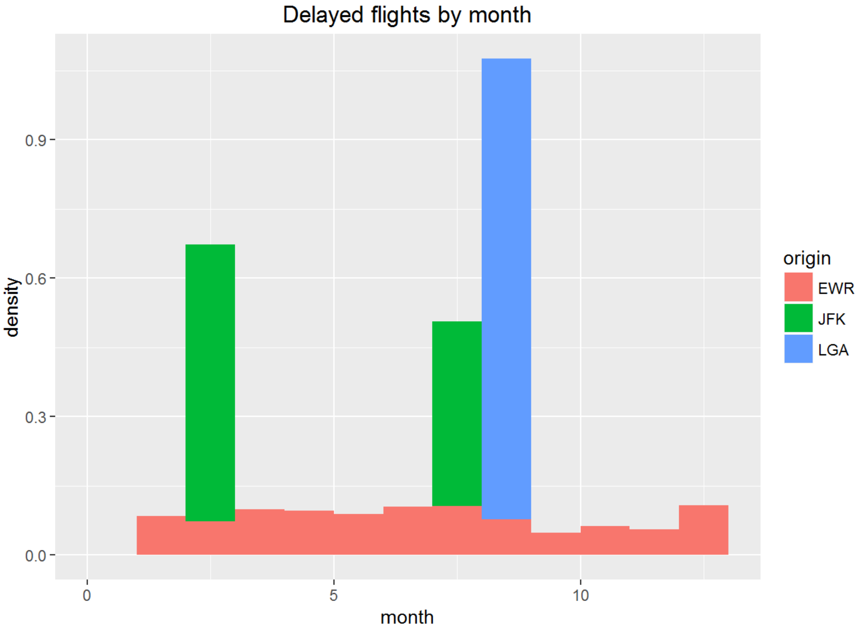
June and July has higher cancellation rates. From this histogram we can see that cancelation is concentrated in June and July. This maybe connected to the seasonal climate in NYC that during summer there are more extreme weather conditions like storms or tornados.

Histogram of delay by Month.

```
Delayed <- final.data[(final.data$takeoffdelay==1),]

delaymonthplot<- ggplot(
  data = Delayed,
  aes(x=month,
      y=..density.., fill = origin)) +
  geom_histogram(binwidth = 1) +
  ggtitle("Delayed flights by month")

delaymonthplot
```

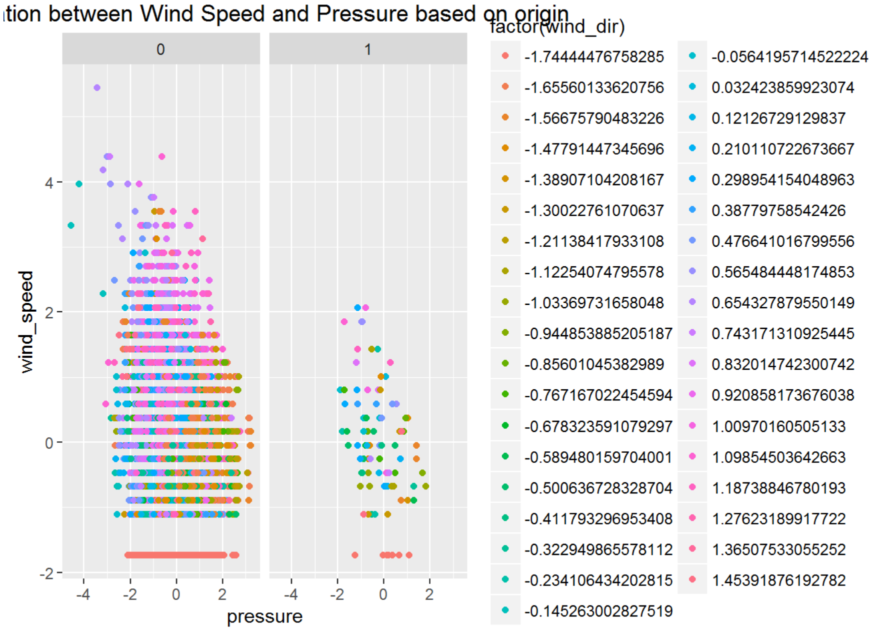


June, July December has higher delay rates. The histogram for delay is consistant with cancelation, but more delay occour during December in addition to June and July. This is likely due to the weather condition during winter like heavy snow or forzen runways.

Visualization of relation between wind speed and pressure.

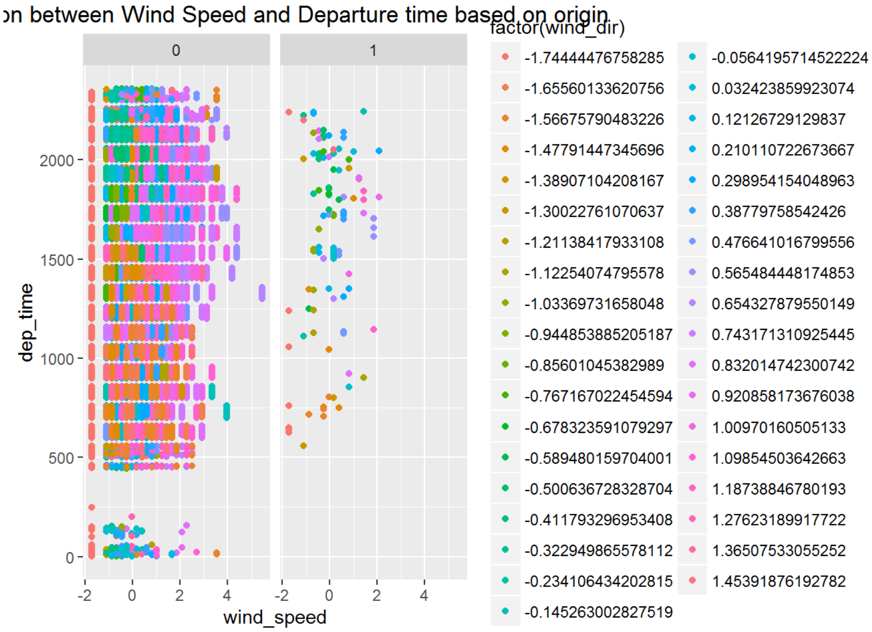
```
Visualization3 <- ggplot(data = final.data,
  aes(x = pressure,
      y = wind_speed)) +
      geom_point(aes(colour = factor(wind_dir))) +
      facet_grid(.~canceled) +
  ggtitle("Relation between Wind Speed and Pressure based on origin")

Visualization3
```



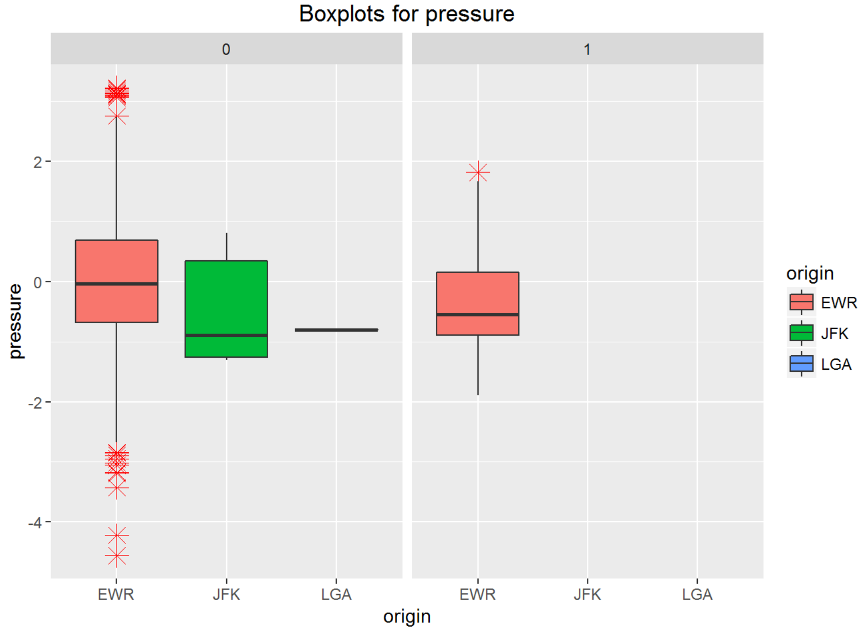
```
Visualization4 <- ggplot(data = final.data,
      aes(x = wind_speed, y = dep_time)) +
      geom_point(aes(colour = factor(wind_dir))) +
      facet_grid(.~canceled) +
      ggtitle("Relation between Wind Speed and Departure time based on origin")

Visualization4
```

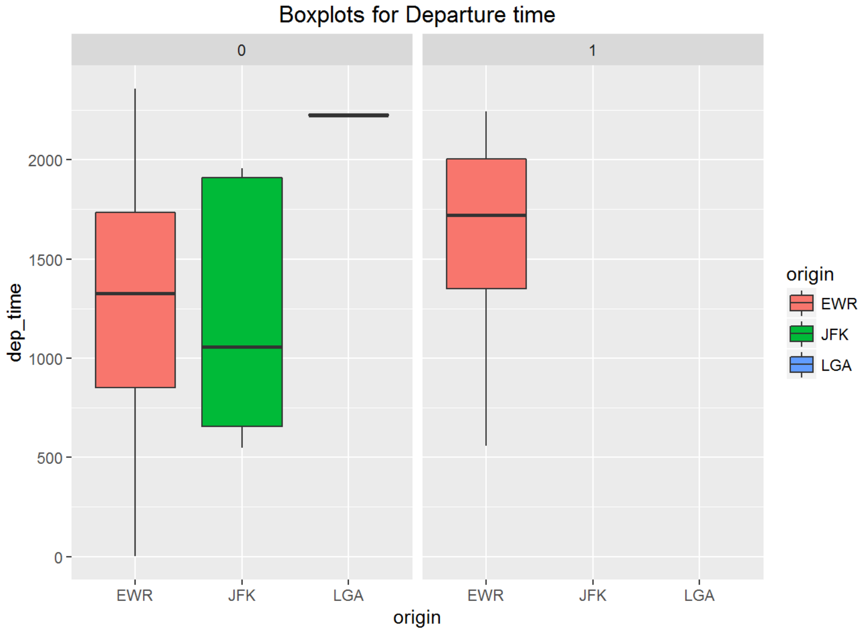


Boxplots for pressure, wind speed, departure time by origin

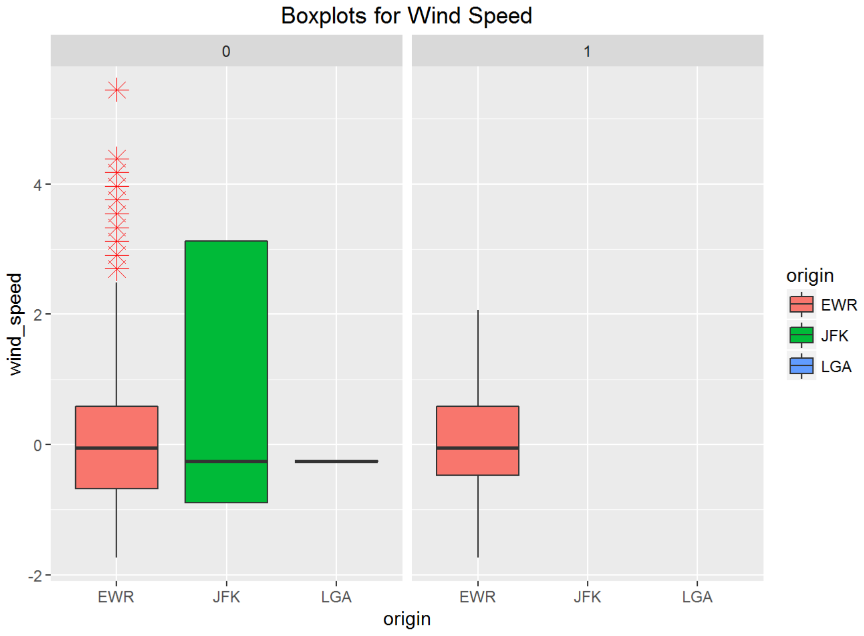
```
ggplot(final.data, aes(x=origin, y=pressure, fill = origin)) +
geom_boxplot(outlier.colour="red", outlier.shape=8, outlier.size=4)+
    facet_grid(.~canceled) +
ggtitle("Boxplots for pressure")
```



```
ggplot(final.data, aes(x=origin, y=dep_time, fill = origin)) +
 geom_boxplot(outlier.colour="red", outlier.shape=8, outlier.size=4)+
     facet_grid(.~canceled) +
 ggtitle("Boxplots for Departure time")
```



```
ggplot(final.data, aes(x=origin, y=wind_speed, fill = origin)) +
geom_boxplot(outlier.colour="red", outlier.shape=8, outlier.size=4)+
    facet_grid(.~canceled) +
ggtitle("Boxplots for Wind Speed")
```



---

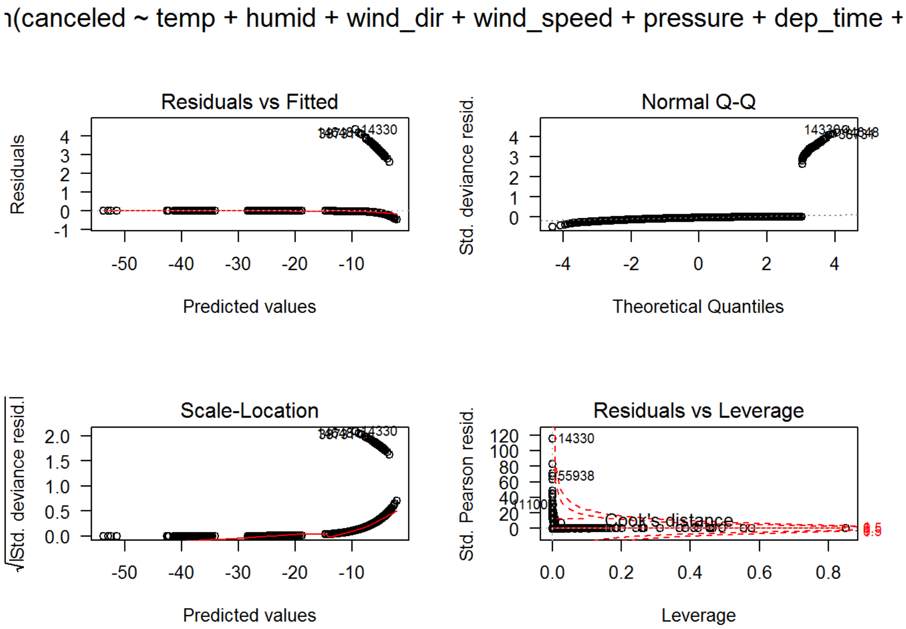
* MODELS

From here we start to construct different models with differnt methods and compare the Cross Validation results.

Logistic Regression

Predicting Flight Cancellation Status

```
logistic.fit <- glm(canceled ~ temp+humid+wind_dir+wind_speed+pressure+dep_time+as.factor(month)+as.factor(origin)+seats+as.
factor(engine)+as.factor(manuyear), family=binomial, data=train, na.action=na.omit)
```

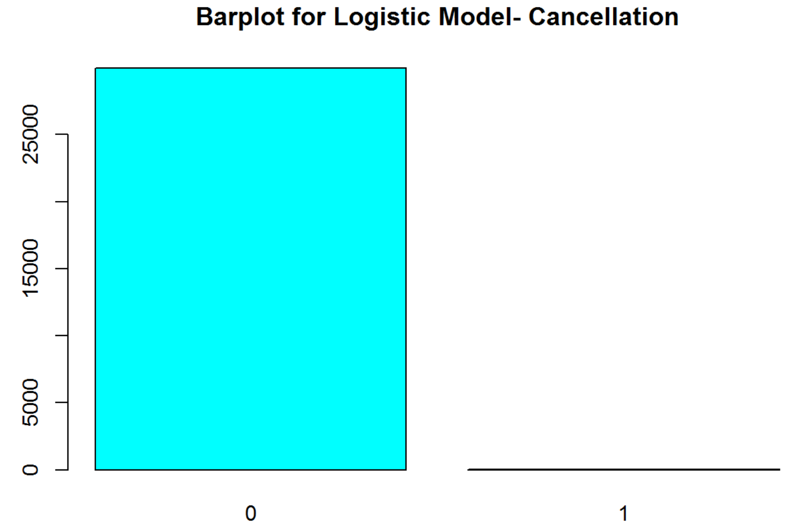


Test the model’s prediction power

```
logistic.prob <- predict(logistic.fit, test, type = "response")
logistic.prediction <- ifelse(logistic.prob > 0.8, 1, 0)
```

Confusion Matrix and Barplot for Logistic Model- Cancellation

```
##
## logistic.prediction     0     1
##                   0 29961    39
```



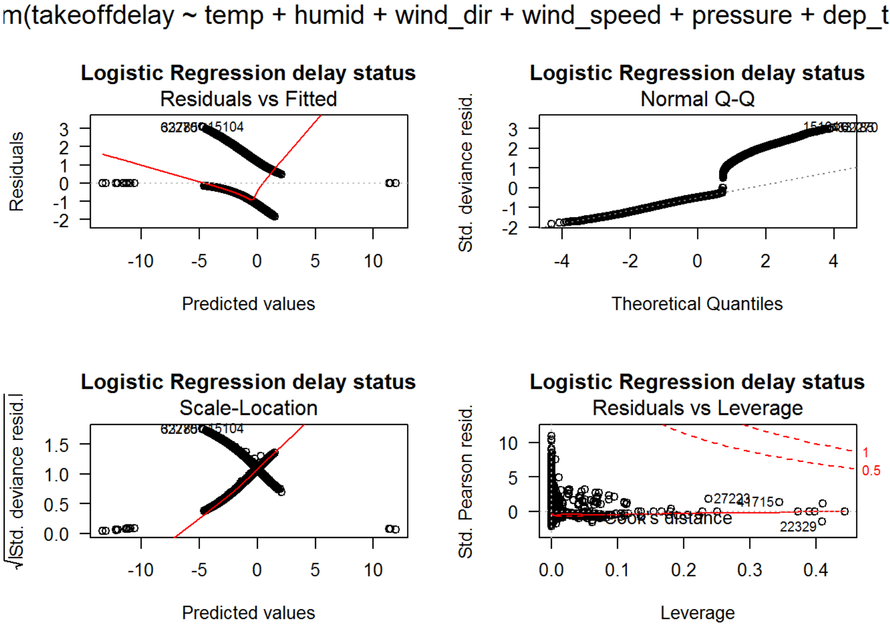
Prediction Rate

```
## [1] 0.9987
```

From a simplle logistic regression, we manage to predict cancelation with 99.883% accuracy. This is very promising, considering the data we fit into the model is actual historical data.

Predicting Flight Delay Status

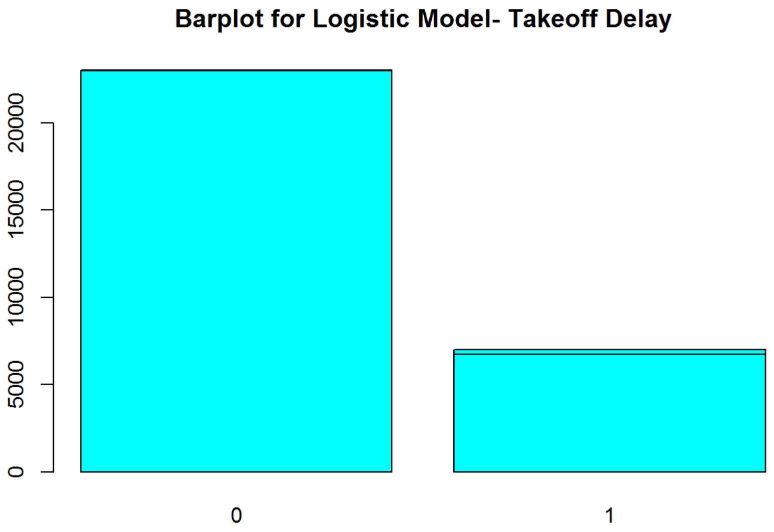
```
takeoffdelay.log.fit <- glm(takeoffdelay ~ temp+humid+wind_dir+wind_speed+pressure+dep_time+as.factor(month)+as.factor(origi
n)+seats+as.factor(engine)+as.factor(manuyear), family=binomial, data=train, na.action=na.omit)
```



Test the model’s prediction power

Confusion Matrix and Barplot for Logistic Model- Takeoff Delay.

```
##
## takeoffdelay.log.prediction     0     1
##                           0 22981  6747
##                           1    32   240
```



Prediction Rate

```
## [1] 0.7740333
```

Prediction for delay is slightly lower in accuracy, but still able to reach about 77%. Due to many more variables that can cause delay, this prediction is still valid.

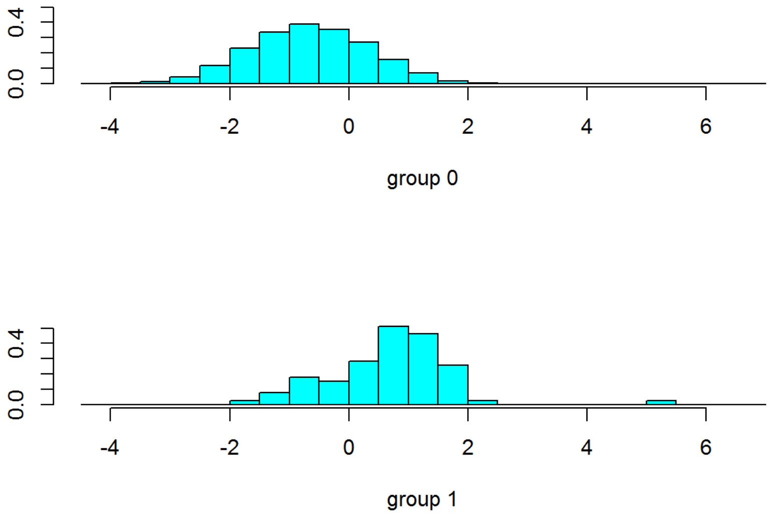
———————————————————————————————————-

Linear Discriminant Analysis Model

Predicting Flight Cancellation Status

```
cancel.lda.fit<-lda(canceled ~ temp+humid+wind_dir+wind_speed+pressure+dep_time+as.factor(month)+as.factor(origin)+seats+as.
factor(engine)+as.factor(manuyear), data=train)
```

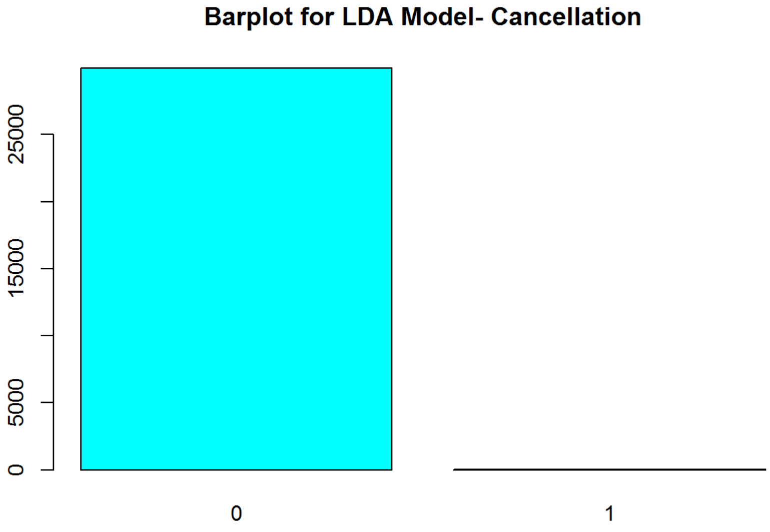
Plots and Summary of LDA regresion for predicting Cancellation Status



```
##         Length Class  Mode     
## prior     2    -none- numeric  
## counts    2    -none- numeric  
## means   128    -none- numeric  
## scaling  64    -none- numeric  
## lev       2    -none- character
## svd       1    -none- numeric  
## N         1    -none- numeric  
## call      3    -none- call     
## terms     3    terms  call     
## xlevels   4    -none- list     
```

Confusion Matrix

```
##                     
## cancel.lda.prediction     0     1
##                     0 29956    39
##                     1     5     0
```



Prediction Rate

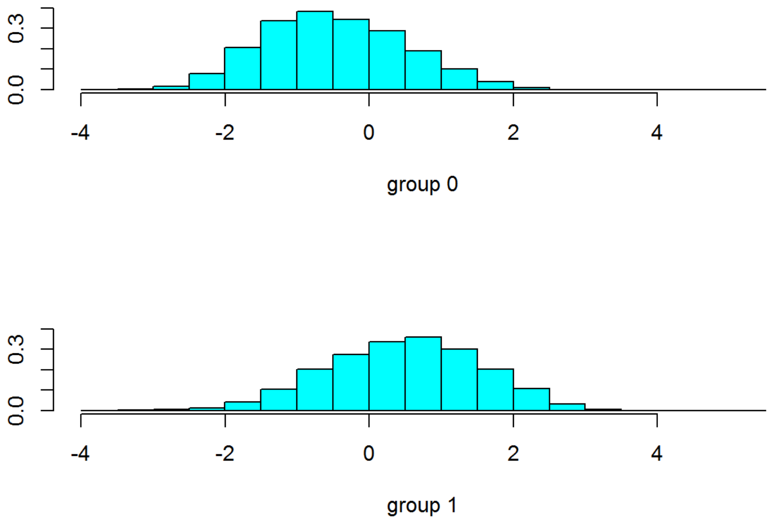
```
## [1] 0.9985333
```

Predicting Flight Delay Status

```
takeoffdelay.lda.fit<-lda(takeoffdelay ~ temp+humid+wind_dir+wind_speed+pressure+dep_time+as.factor(month)+as.factor(origin)
+seats+as.factor(engine)+as.factor(manuyear), data=train)
```

**

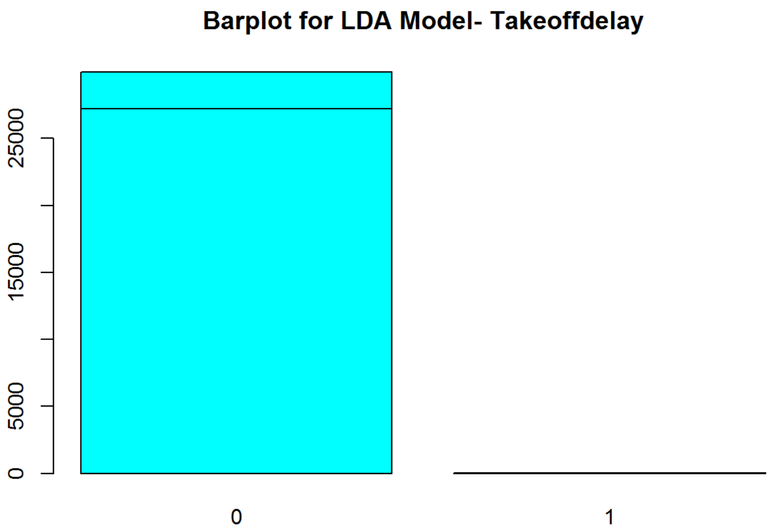
Plots and Summary of LDA regresion for predicting Delay Status



```
##         Length Class  Mode     
## prior     2    -none- numeric  
## counts    2    -none- numeric  
## means   128    -none- numeric  
## scaling  64    -none- numeric  
## lev       2    -none- character
## svd       1    -none- numeric  
## N         1    -none- numeric  
## call      3    -none- call     
## terms     3    terms  call     
## xlevels   4    -none- list     
```

Confusion Matrix and Barplot for LDA Model

```
##                          
## takeoffdelay.lda.prediction     0     1
##                           0 27223    27
##                           1  2738    12
```



Prediction Rate

```
## [1] 0.7909
```

Our lda regression predicting outcome of the flight status(cancelled, delayed or normal) produced a result consistant with what we found in logistic regressions. Since it predicts cancellation and delay within the same regression, the 79% accuracy is more than acceptable.

K Nearest Neighbour Model, with K=10 to reduce computation time.

Predicting Flight Cancellation Status

```
#Knn
cancel.KNN <- knn(train[ , 12:16], test[ , 12:16], train$canceled, k=10)
mean(cancel.KNN == test$canceled)
```

```
## [1] 0.9987
```

Predicting Flight Delay Status

```
takeoffdelay.KNN <- knn(train[ , 12:16], test[ , 12:16], train$takeoffdelay, k=10)
mean(takeoffdelay.KNN == test$takeoffdelay)
```

```
## [1] 0.7948333
```

Quadratic Discriminant Analysis

```
#Qda, doesn't work
flightstate.qda.fit <- qda(canceled ~ temp+humid+wind_dir+wind_speed+pressure+dep_time+as.factor(month)+ as.factor(origin) +
 seats+as.factor(engine)+as.factor(manuyear), family=binomial, data=train, na.action=na.omit)

flightstate.qda.fit

flightstate.qda.test<-predict(flightstate.qda.fit)$class
mean(flightstate.qda.test==total.test$flightstate)
```

Qda regression, unfortunately did not fit into our data at all at this point. An error of “rank deficiency in group Canceled” is reported. Which might mean the decision boundry is linear.

Decision Tree

```
#tree, doesn't work
tree.cancel<-tree(canceled~temp+dewp+wind_speed+precip+humid+pressure+visib+carrier+type+manuyear+engines+seats+engine+origi
n.x+distance, data=total.train)
summary(tree.cancel)
```

The model ends up with is a single node tree with zero variables used to construct branches. This infact confirms the high accuracy in glm prediction and is a tell tale sign that the decision boundry is highly linear.

Random forest and boosted tree model.

```
#bootsing tree, doesn't work
pows <- seq(-10, -0.2, by = 0.1)
lambdas <- 10^pows
train.err <- rep(NA, length(lambdas))
for (i in 1:length(lambdas)) {
  boost.fit <- gbm(canceled~temp+dewp+wind_speed+precip+humid+pressure+visib+manuyear+seats+engines+distance+month+day+hour,
 data = total.train, distribution = "gaussian", n.trees = 1000, shrinkage = lambdas[i])
  pred.train <- predict(boost.fit, total.train, n.trees = 1000)
  train.err[i] <- mean(pred.train==total.train$canceled)
}
plot(lambdas, train.err, type = "b", xlab = "Shrinkage values", ylab = "Training MSE")
```

```
#random forest
typeof(total$carrier)
cancel.rf<-randomForest(canceled~ temp+dewp+wind_speed+precip+humid+pressure+visib+distance, data = total.train, mtry = 3, n
tree = 500, importance = TRUE)
cancel.rf
```

Again with no success.

Cross Validation

In order to perform cross validation, we fit our glm models into the whole dataset.

```
# 10 fold cross validation
total.cancel.log.fit<-glm(canceled~temp+dewp+wind_speed+precip+humid+pressure+visib+carrier+type+manuyear+engines+seats+engi
ne+origin.x+distance, family=binomial, data=total, na.action=na.omit)
summary(total.cancel.log.fit)
```

```
# 10 fold cross validation
total.takeoffdelay.log.fit<-glm(takeoffdelay~temp+dewp+wind_speed+precip+humid+pressure+visib+carrier+type+manuyear+engines+
seats+engine+origin.x+distance, family=binomial, data=total, na.action=na.omit)
summary(total.takeoffdelay.log.fit)
```

Then we performed a cross validation, 10 fold to reduce computation time.

```
# 10 fold cross validation
cancel.cv.err<-cv.glm(total, total.cancel.log.fit, K=10)
cancel.cv.err$delta
takeoffdelay.cv.err<-cv.glm(total, total.takeoffdelay.log.fit, K=10)
takeoffdelay.cv.err$delta
```

The result it produced is consistent with our previous finding. The prediction accuracy for delay improved slightly, about 85%.

The result is again similar to our findings in glm, lda and SVM.

* Conclusion

These results support our initial claim that weather is a very significant determinant of flight delay and cancellation. The fact that logistics, LDA and KNN models are giving us almost 99% accuracy for flight cancellation and almost 80% accuracy for flight delay, we would choose to use these models for cancellation and delay predictions in any prediction application we built.

In contrast the results of tree, random forest, and gradient boosting end up not being the best models for flight delay and cancellation predictions, which gives us an insight into the nature of the prediction boundry which most prabably is linear.