

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

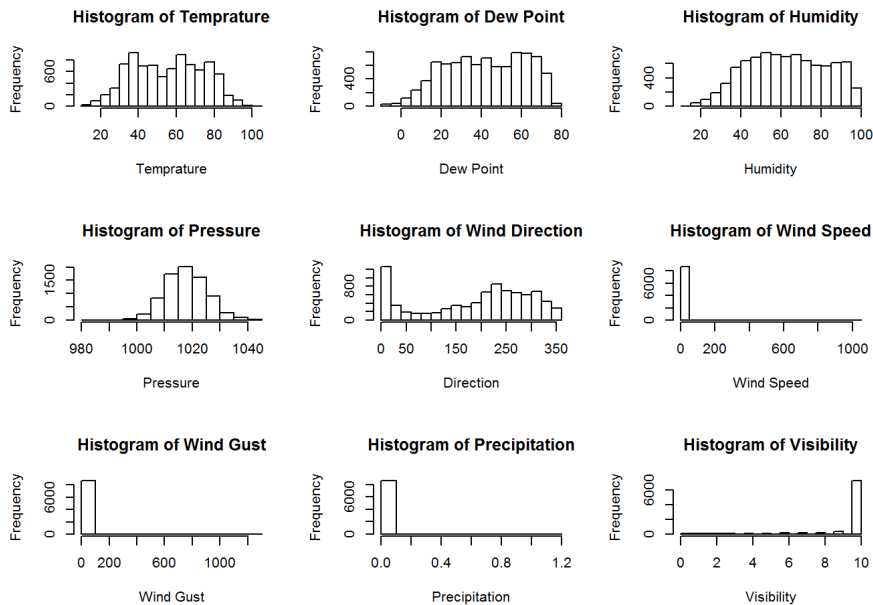
Here are the packages used.

```
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. "cancellation" 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)
```

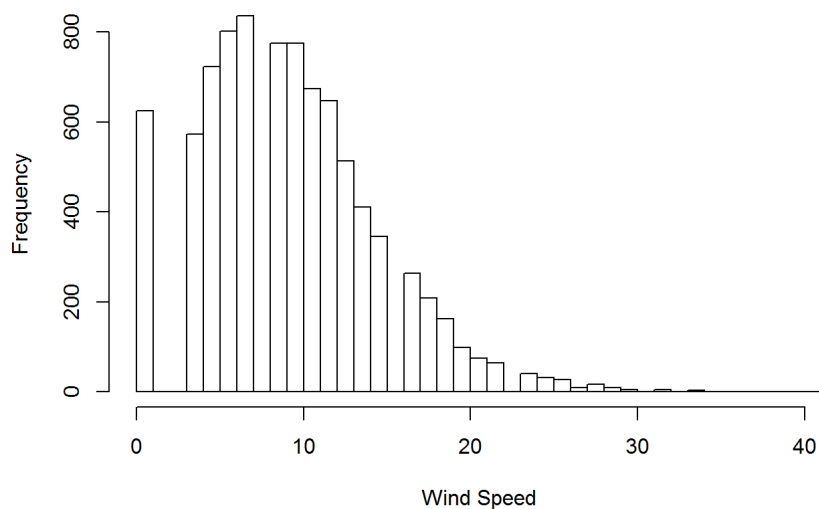
We can see there is one observation with an anomalous wind_speed so we proceed to remove it along with the precip and visib column which have similar values for all rows.

```
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")
```

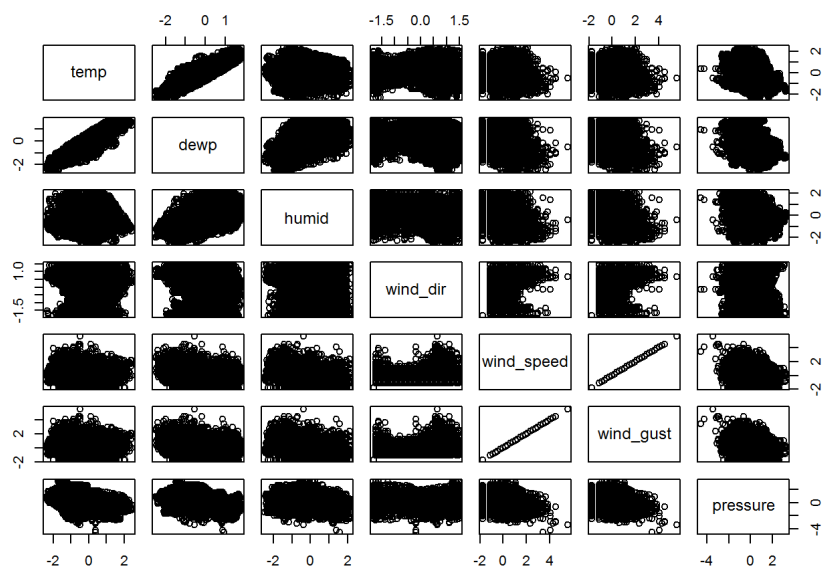
Histogram of Wind Speed



Now lets check for correlation between weather variables

```
##          temp      dewp      humid  wind_dir  wind_speed
## temp      1.0000000  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 coefficient 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, type, 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

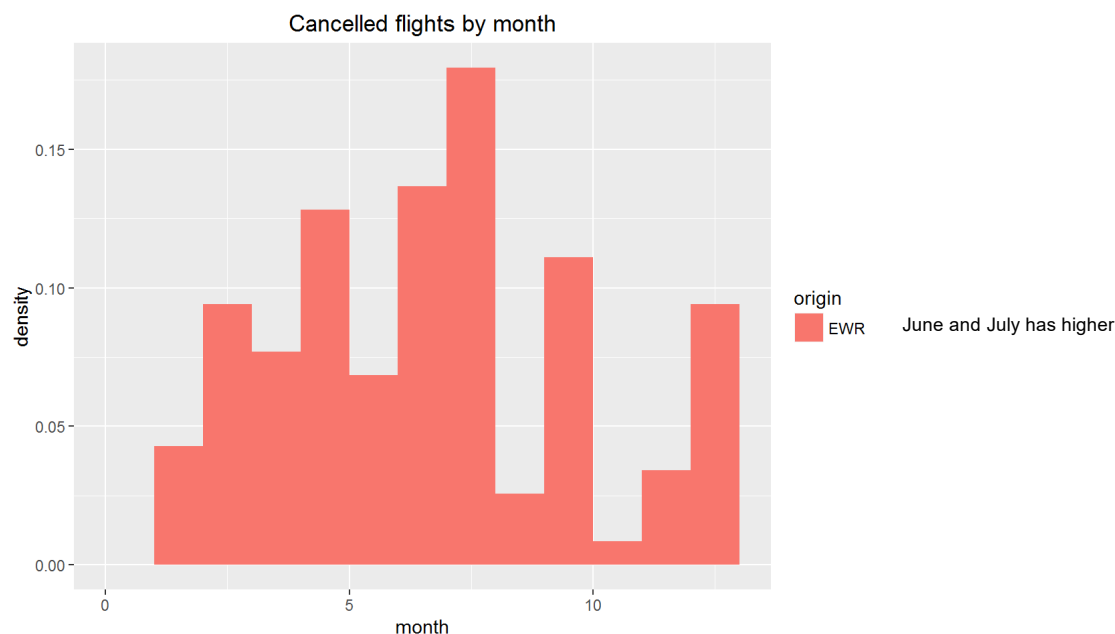
Here we check if time of the year has any effect on cancelation or delay, since it may correlate to weather conditions across seasons.

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
```



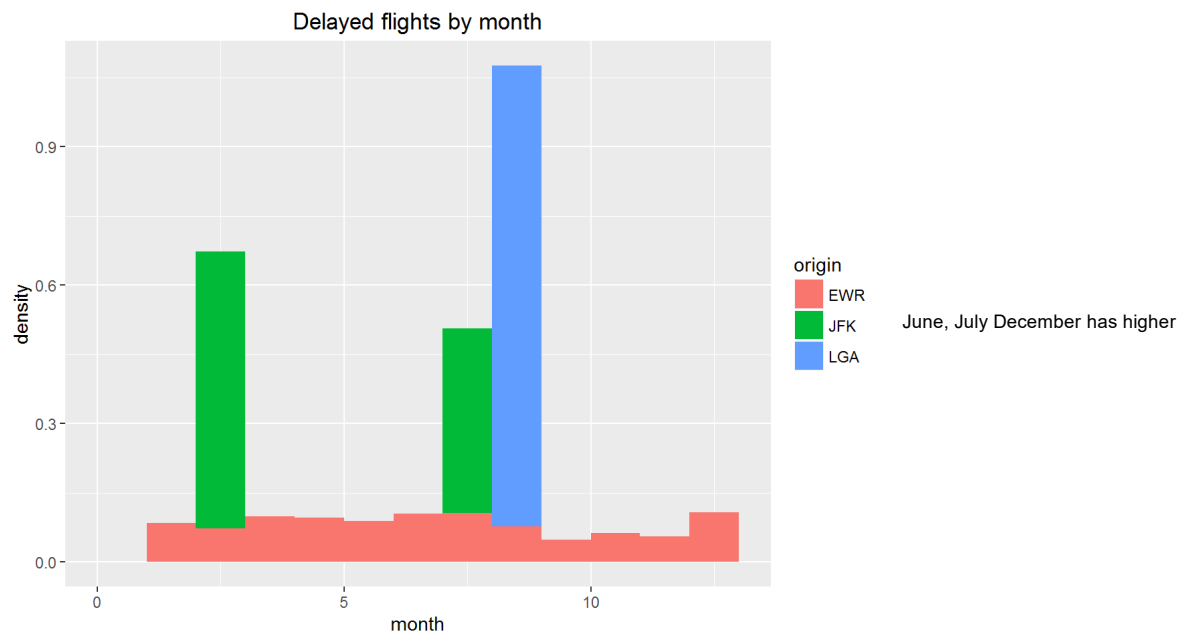
cancellation rates. From this histogram we can see that cancelation is concentrated in June and July. This maybe connected to the seasonal climate in NYC that during summer there are more extreme weather conditions like storms or tornados.

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
```



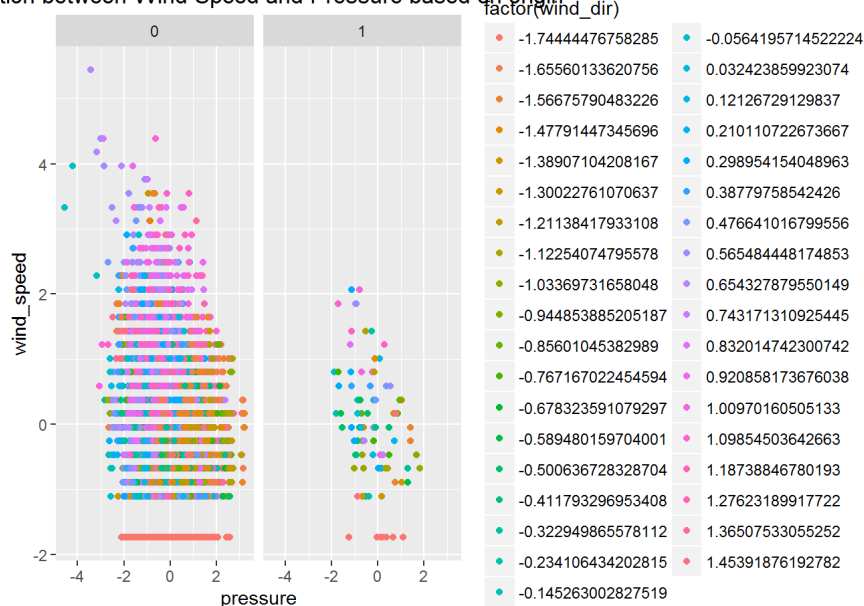
delay rates. The histogram for delay is constant with cancellation, but more delay occur during December in addition to June and July. This is likely due to the weather condition during winter like heavy snow or frozen runways.

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

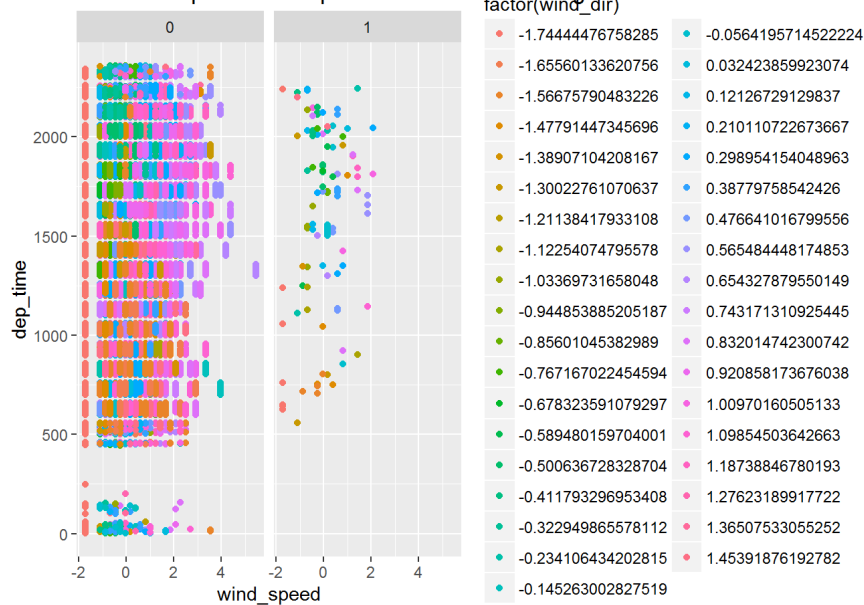
tion between Wind Speed and Pressure based on origin



```
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")
```

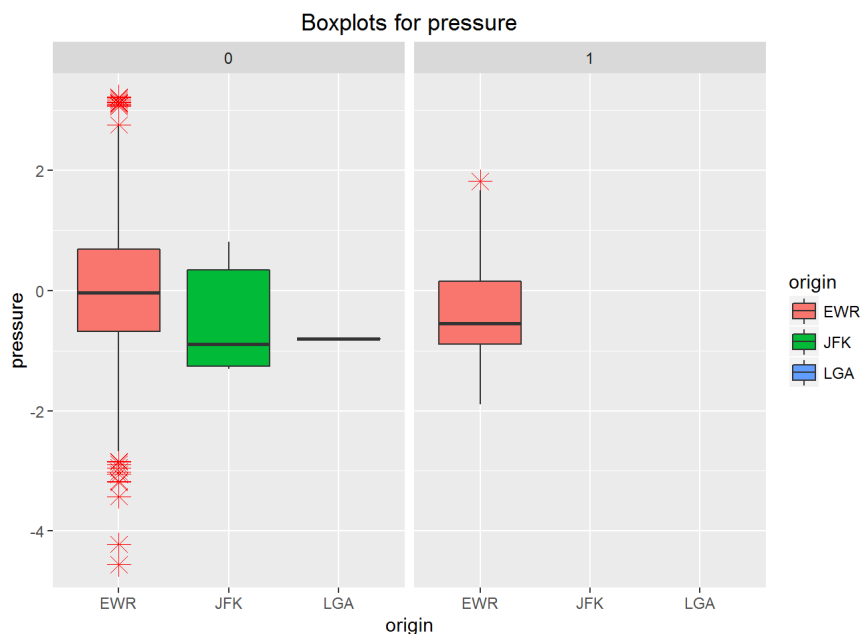
Visualization4

on between Wind Speed and Departure time based on origin

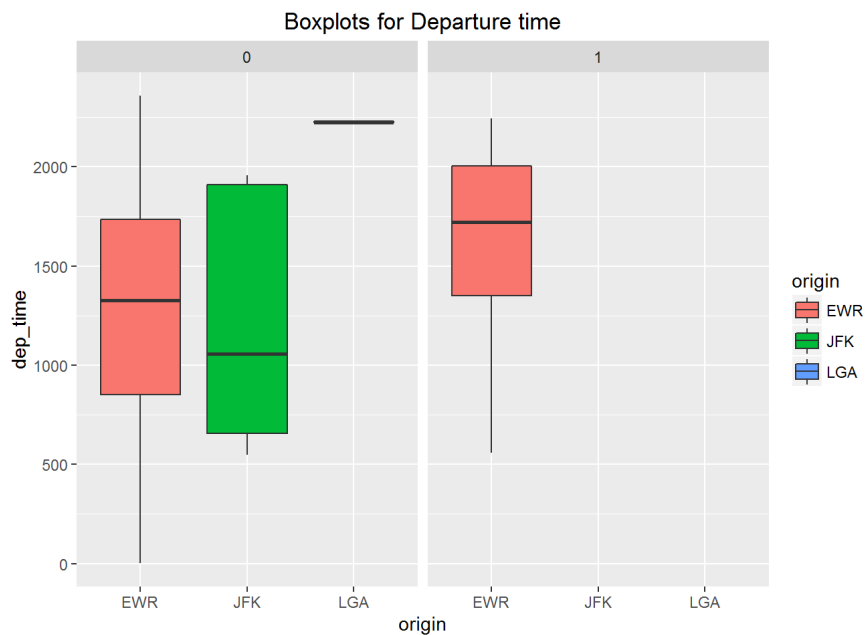


Boxplots for pressure, wind speed, departure time by origin

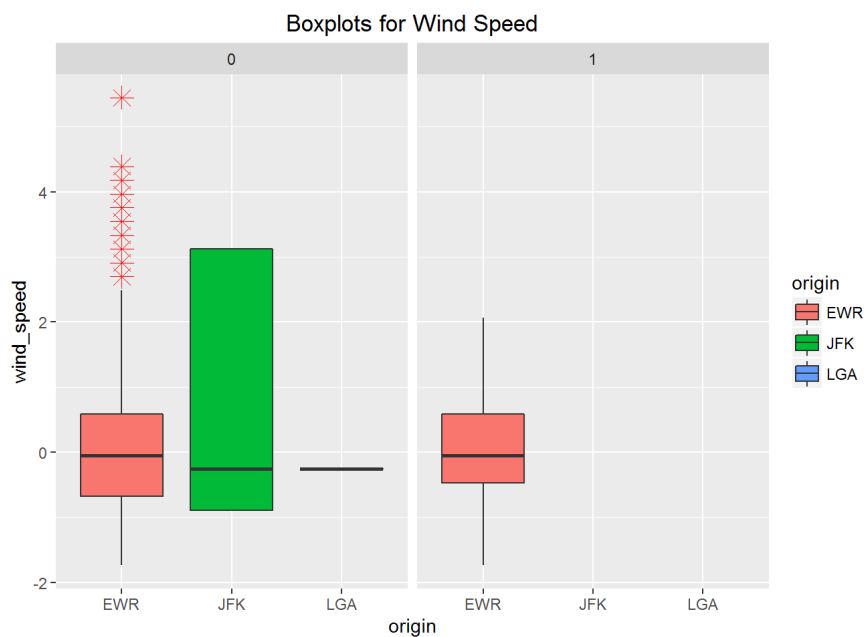
```
ggplot(final.data, aes(x=origin, y=pressure, fill = origin)) +
  geom_boxplot(outlier.colour="red", outlier.shape=8, outlier.size=4)+
  facet_grid(.~canceled) +
  ggtitle("Boxplots for pressure")
```



```
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 different methods and compare the Cross Validation results.

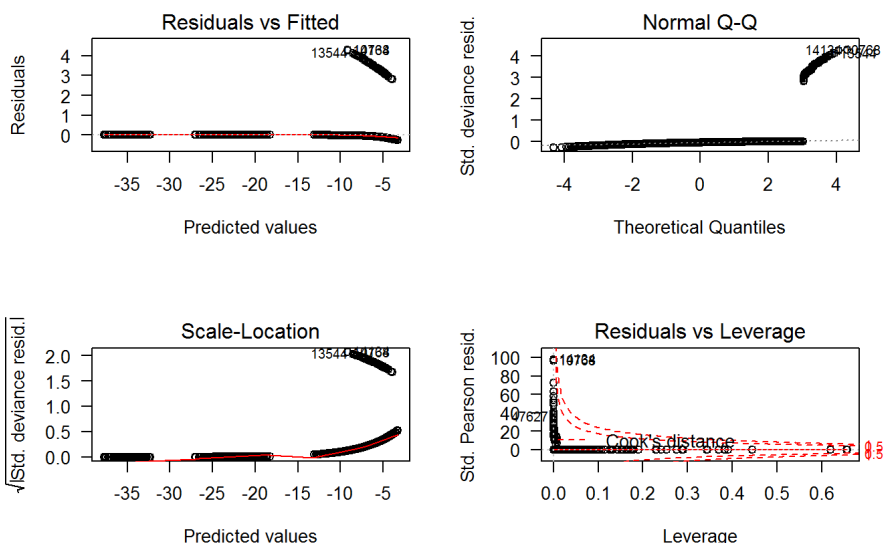
Logistic Regression

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)
```



```
lm(canceled ~ temp + humid + wind_dir + wind_speed + pressure + dep_time +
```



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

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

Prediction Rate

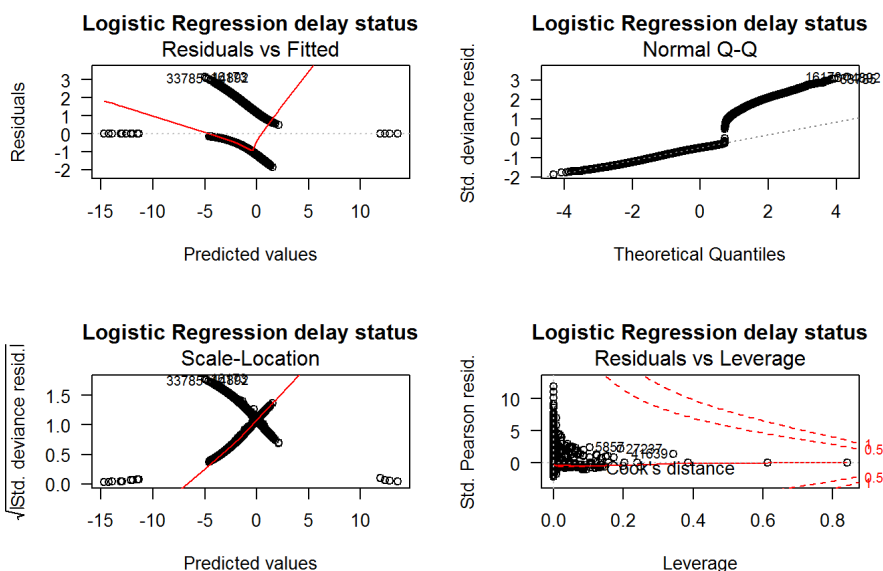
```
## [1] 0.9987
```

From a simple logistic regression, we manage to predict cancellation 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(origin)+seats+as.factor(engine)+as.factor(manueyear), family=binomial, data=train, na.action=na.omit)
```

```
lm(takeoffdelay ~ temp + humid + wind_dir + wind_speed + pressure + dep_ti
```



Test the model's prediction power

Confusion Matrix

```
##
## takeoffdelay.log.prediction      0      1
##                                0 23001 6736
##                                1   39  224
```

Prediction Rate

```
## [1] 0.7741667
```

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.

Finally table for the two regressions

```
##
## Logistic regression Model to predict Cancellation & Delay
## =====
##                                canceled      takeoffdelay
##                                (1)          (2)
## -----
## temp                1.040***      -0.054**
##                    (0.272)        (0.023)
##
## humid                0.253*        0.269***
##                    (0.140)        (0.013)
##
## wind_dir            -0.572***      -0.037***
##                    (0.147)        (0.013)
##
## wind_speed           0.114          0.151***
##                    (0.138)        (0.013)
##
## pressure             0.007          -0.146***
##                    (0.137)        (0.013)
##
## dep_time             0.001***      0.002***
##                    (0.0003)       (0.00003)
##
## as.factor(origin)JFK  -13.268       -0.825
##                    (4,073.271)    (0.559)
##
## as.factor(origin)LGA  -14.338       12.074
##                    (20,555.790)   (228.370)
##
## seats               -0.006***      -0.003***
##                    (0.002)       (0.0002)
##
## Constant             -8.322        10.800
##                    (22,658.590)   (214.158)
##
## N                    64,239        64,239
## Log Likelihood        -545.214     -30,198.300
## Akaike Inf. Crit.     1,218.429     60,524.600
## =====
## Notes:                ***Significant at the 1 percent level.
##                        **Significant at the 5 percent level.
##                        *Significant at the 10 percent level.
```

Linear Discriminant Analysis Model

Predicting Flight Cancellation Status

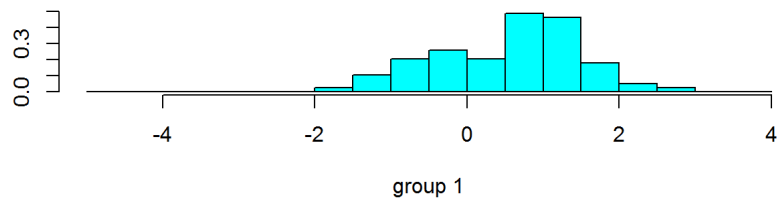
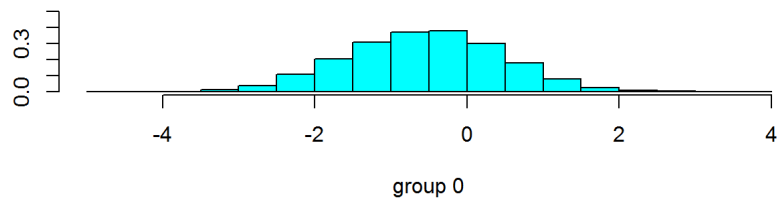
```
cancel.lda.fit<-lda(canceled ~ temp+humid+wind_dir+wind_speed+pressure+dep_time+as.factor(month)+as.factor(origin)+seats+as.
factor(engine)+as.factor(manuyear), data=train)
```

```
## Call:
## 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)
##
## Prior probabilities of groups:
##      0      1
## 0.998785784 0.001214216
##
## Group means:
##      temp      humid      wind_dir      wind_speed      pressure dep_time
## 0 0.0002347948 0.002940957 -0.003046493 -0.002264813 0.002148362 1328.107
## 1 0.3895340510 0.101064838 -0.384456857 -0.004171360 -0.234045006 1579.551
## as.factor(month)2 as.factor(month)3 as.factor(month)4 as.factor(month)5
## 0 0.07359611 0.08804414 0.08957155 0.08151369
## 1 0.10256410 0.10256410 0.10256410 0.08974359
## as.factor(month)6 as.factor(month)7 as.factor(month)8 as.factor(month)9
## 0 0.08165396 0.08427238 0.08572186 0.08154486
## 1 0.12820513 0.16666667 0.02564103 0.08974359
## as.factor(month)10 as.factor(month)11 as.factor(month)12
## 0 0.08791945 0.08519194 0.07912907
## 1 0.01282051 0.03846154 0.07692308
## as.factor(origin)JFK as.factor(origin)LGA seats
## 0 0.0006546033 3.117158e-05 127.1430
## 1 0.0000000000 0.000000e+00 103.7308
## as.factor(engine)Reciprocating as.factor(engine)Turbo-fan
## 0 0.002041739 0.8256729
## 1 0.000000000 0.8333333
## as.factor(engine)Turbo-jet as.factor(engine)Turbo-prop
## 0 0.1716931 1.558579e-05
## 1 0.1666667 0.000000e+00
## as.factor(engine)Turbo-shaft as.factor(manuyear)1963
## 0 0.0005455027 0.0001870295
## 1 0.0000000000 0.0000000000
## as.factor(manuyear)1967 as.factor(manuyear)1968 as.factor(manuyear)1975
## 0 1.558579e-05 0.0001246863 0.0002026153
## 1 0.000000e+00 0.0000000000 0.0000000000
## as.factor(manuyear)1976 as.factor(manuyear)1977 as.factor(manuyear)1978
## 0 0.0001714437 0.0001714437 7.792896e-05
## 1 0.0000000000 0.0000000000 0.000000e+00
## as.factor(manuyear)1979 as.factor(manuyear)1980 as.factor(manuyear)1983
## 0 0.0003428874 0.0003896448 0.000374059
## 1 0.0000000000 0.0000000000 0.0000000000
## as.factor(manuyear)1985 as.factor(manuyear)1986 as.factor(manuyear)1987
## 0 0.0004052306 0.0003273016 0.00166768
## 1 0.0000000000 0.0000000000 0.0000000000
## as.factor(manuyear)1988 as.factor(manuyear)1989 as.factor(manuyear)1990
## 0 0.002696342 0.002618413 0.009242375
## 1 0.0000000000 0.0000000000 0.0000000000
## as.factor(manuyear)1991 as.factor(manuyear)1992 as.factor(manuyear)1993
## 0 0.01182962 0.01482209 0.01039572
## 1 0.03846154 0.02564103 0.00000000
## as.factor(manuyear)1994 as.factor(manuyear)1995 as.factor(manuyear)1996
## 0 0.01999657 0.008400742 0.008276056
## 1 0.02564103 0.000000000 0.012820513
## as.factor(manuyear)1997 as.factor(manuyear)1998 as.factor(manuyear)1999
## 0 0.01340378 0.09619551 0.1025233
## 1 0.00000000 0.10256410 0.1025641
## as.factor(manuyear)2000 as.factor(manuyear)2001 as.factor(manuyear)2002
## 0 0.08690638 0.1295491 0.1259020
## 1 0.06410256 0.1666667 0.1282051
## as.factor(manuyear)2003 as.factor(manuyear)2004 as.factor(manuyear)2005
## 0 0.05832203 0.05663877 0.04820685
## 1 0.06410256 0.07692308 0.08974359
## as.factor(manuyear)2006 as.factor(manuyear)2007 as.factor(manuyear)2008
## 0 0.03170150 0.01775222 0.04270507
## 1 0.03846154 0.01282051 0.02564103
## as.factor(manuyear)2009 as.factor(manuyear)2010 as.factor(manuyear)2011
## 0 0.02595034 0.01488443 0.01176727
## 1 0.01282051 0.00000000 0.00000000
## as.factor(manuyear)2012 as.factor(manuyear)2013
## 0 0.02694783 0.01747167
## 1 0.01282051 0.00000000
##
## Coefficients of linear discriminants:
##
## LD1
## temp 0.6532602102
## humid 0.2448394989
## wind_dir -0.4104479386
## wind_speed 0.0664073137
```

```

## pressure -0.0552579471
## dep_time 0.0008488122
## as.factor(month)2 0.4207638457
## as.factor(month)3 0.1611699979
## as.factor(month)4 -0.4207549528
## as.factor(month)5 -0.9011766320
## as.factor(month)6 -0.8543883397
## as.factor(month)7 -0.7272153142
## as.factor(month)8 -1.9462883260
## as.factor(month)9 -0.9567677152
## as.factor(month)10 -1.6981094085
## as.factor(month)11 -0.5922480698
## as.factor(month)12 -0.0790112444
## as.factor(origin)JFK -0.1390089418
## as.factor(origin)LGA -1.2643400671
## seats -0.0046071950
## as.factor(engine)Reciprocating -1.1057557574
## as.factor(engine)Turbo-fan 0.1396300698
## as.factor(engine)Turbo-jet 0.2180030507
## as.factor(engine)Turbo-prop -0.5251319325
## as.factor(engine)Turbo-shaft -0.9492151778
## as.factor(manuyear)1963 -0.3009962028
## as.factor(manuyear)1967 -0.5251319325
## as.factor(manuyear)1968 -0.1525321625
## as.factor(manuyear)1975 -0.4656114666
## as.factor(manuyear)1976 -0.2035270692
## as.factor(manuyear)1977 -0.4876493546
## as.factor(manuyear)1978 -0.5343173716
## as.factor(manuyear)1979 -0.8675551409
## as.factor(manuyear)1980 -0.2571492779
## as.factor(manuyear)1983 -0.2135846245
## as.factor(manuyear)1985 -0.1794348279
## as.factor(manuyear)1986 -0.4625090088
## as.factor(manuyear)1987 -0.4116792623
## as.factor(manuyear)1988 -0.5095021418
## as.factor(manuyear)1989 -0.6337771501
## as.factor(manuyear)1990 -0.5532897550
## as.factor(manuyear)1991 2.3032511482
## as.factor(manuyear)1992 0.9949252505
## as.factor(manuyear)1993 -0.5084600931
## as.factor(manuyear)1994 0.6349083495
## as.factor(manuyear)1995 -0.3962053419
## as.factor(manuyear)1996 0.8866905392
## as.factor(manuyear)1997 -0.4901197730
## as.factor(manuyear)1998 0.1877577441
## as.factor(manuyear)1999 0.0103945769
## as.factor(manuyear)2000 -0.1517646761
## as.factor(manuyear)2001 0.3324129141
## as.factor(manuyear)2002 0.0115051065
## as.factor(manuyear)2003 0.0548467885
## as.factor(manuyear)2004 0.3260504021
## as.factor(manuyear)2005 0.7346958242
## as.factor(manuyear)2006 0.2401096030
## as.factor(manuyear)2007 0.0121534044
## as.factor(manuyear)2008 -0.1024165480
## as.factor(manuyear)2009 -0.0857099900
## as.factor(manuyear)2010 -0.7159833352
## as.factor(manuyear)2011 -0.5541830162
## as.factor(manuyear)2012 0.0596878220
## as.factor(manuyear)2013 -0.3864929374

```



Confusion Matrix

```
##
## cancel.lda.prediction    0    1
##                0 29961   39
##                1     0    0
```

Prediction Rate

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

Predicting Flight Delay Status

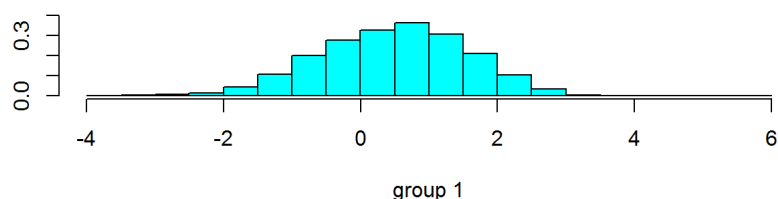
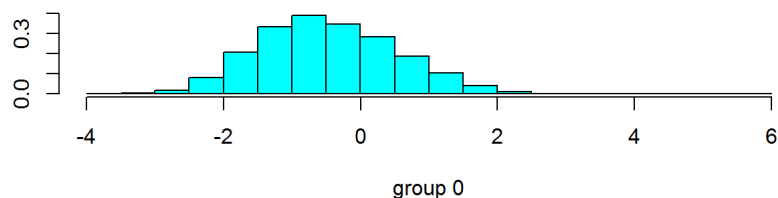
```
takeoffdelay.lda.fit<-lda(takeoffdelay ~ temp+humid+wind_dir+wind_speed+pressure+dep_time+as.factor(month)+as.factor(origin)
+seats+as.factor(engine)+as.factor(manuyear), data=train)
```

```
## Call:
## 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)
##
## Prior probabilities of groups:
##      0      1
## 0.7671664 0.2328336
##
## Group means:
##      temp      humid      wind_dir      wind_speed      pressure      dep_time
## 0 -0.04197098  0.04142953 -0.02373019 -0.07477239  0.06032755 1242.465
## 1  0.14132929 -0.12336381  0.06311544  0.23663135 -0.19077868 1611.598
## as.factor(month)2 as.factor(month)3 as.factor(month)4 as.factor(month)5
## 0      0.07382006      0.08522381      0.08751674      0.07968427
## 1      0.07300929      0.09741258      0.09640971      0.08758441
## as.factor(month)6 as.factor(month)7 as.factor(month)8 as.factor(month)9
## 0      0.07477375      0.07777688      0.08796315      0.09196055
## 1      0.10456642      0.10610417      0.07802367      0.04726884
## as.factor(month)10 as.factor(month)11 as.factor(month)12
## 0      0.09585650      0.09435494      0.06998498
## 1      0.06137594      0.05475697      0.10924651
## as.factor(origin)JFK as.factor(origin)LGA      seats
## 0      0.0007710726      0.0000000000 130.4179
## 1      0.0002674333      0.0001337167 116.2307
## as.factor(engine)Reciprocating as.factor(engine)Turbo-fan
## 0      0.002110304      0.8189400
## 1      0.001805175      0.8478973
## as.factor(engine)Turbo-jet as.factor(engine)Turbo-prop
## 0      0.1783004      2.029138e-05
## 1      0.1498964      0.000000e+00
## as.factor(engine)Turbo-shaft as.factor(manuyear)1963
## 0      0.0006290329      0.0002434966
## 1      0.0002674333      0.0000000000
## as.factor(manuyear)1967 as.factor(manuyear)1968 as.factor(manuyear)1975
## 0      2.029138e-05      0.0001014569      0.0002029138
## 1      0.000000e+00      0.0002005750      0.0002005750
## as.factor(manuyear)1976 as.factor(manuyear)1977 as.factor(manuyear)1978
## 0      0.0001826225      0.0001623311      8.116554e-05
## 1      0.0001337167      0.0002005750      6.685833e-05
## as.factor(manuyear)1979 as.factor(manuyear)1980 as.factor(manuyear)1983
## 0      0.0003855363      0.0004261191      0.0004261191
## 1      0.0002005750      0.0002674333      0.0002005750
## as.factor(manuyear)1985 as.factor(manuyear)1986 as.factor(manuyear)1987
## 0      0.0004261191      0.0003449535      0.0019682643
## 1      0.0003342916      0.0002674333      0.0006685833
## as.factor(manuyear)1988 as.factor(manuyear)1989 as.factor(manuyear)1990
## 0      0.002901668      0.002556714      0.009354328
## 1      0.002005750      0.002808050      0.008825299
## as.factor(manuyear)1991 as.factor(manuyear)1992 as.factor(manuyear)1993
## 0      0.01207337      0.01483300      0.009902196
## 1      0.01116534      0.01484255      0.011967641
## as.factor(manuyear)1994 as.factor(manuyear)1995 as.factor(manuyear)1996
## 0      0.02075809      0.008522381      0.008725295
## 1      0.01751688      0.007956141      0.006819549
## as.factor(manuyear)1997 as.factor(manuyear)1998 as.factor(manuyear)1999
## 0      0.01416339      0.0972566      0.09993507
## 1      0.01083105      0.0927325      0.11105168
## as.factor(manuyear)2000 as.factor(manuyear)2001 as.factor(manuyear)2002
## 0      0.08579197      0.1276125      0.1228440
## 1      0.09045932      0.1361236      0.1359898
## as.factor(manuyear)2003 as.factor(manuyear)2004 as.factor(manuyear)2005
## 0      0.05620713      0.05572014      0.04774563
## 1      0.06532059      0.05977134      0.04994317
## as.factor(manuyear)2006 as.factor(manuyear)2007 as.factor(manuyear)2008
## 0      0.03175602      0.01864778      0.04490483
## 1      0.03155713      0.01477569      0.03536806
## as.factor(manuyear)2009 as.factor(manuyear)2010 as.factor(manuyear)2011
## 0      0.02741366      0.01560407      0.012560367
## 1      0.02106037      0.01243565      0.009092732
## as.factor(manuyear)2012 as.factor(manuyear)2013
## 0      0.02830648      0.01848545
## 1      0.02239754      0.01404025
##
## Coefficients of linear discriminants:
##      LD1
## temp      -0.120960796
## humid      0.325716288
## wind_dir    -0.047443884
## wind_speed  0.173843334
```

```

## pressure -0.176675477
## dep_time 0.002080933
## as.factor(month)2 -0.189877764
## as.factor(month)3 -0.003620355
## as.factor(month)4 0.262121190
## as.factor(month)5 0.233830189
## as.factor(month)6 0.430183666
## as.factor(month)7 0.495862289
## as.factor(month)8 0.024848992
## as.factor(month)9 -0.532833601
## as.factor(month)10 -0.413783898
## as.factor(month)11 -0.430117497
## as.factor(month)12 0.467602410
## as.factor(origin)JFK -0.460294787
## as.factor(origin)LGA 3.116254258
## seats -0.003036722
## as.factor(engine)Reciprocating -5.743120549
## as.factor(engine)Turbo-fan -5.436145911
## as.factor(engine)Turbo-jet -5.602726969
## as.factor(engine)Turbo-prop -2.738956825
## as.factor(engine)Turbo-shaft -6.368097121
## as.factor(manuyear)1963 -1.595402626
## as.factor(manuyear)1967 -2.738956825
## as.factor(manuyear)1968 0.905880537
## as.factor(manuyear)1975 -0.265253463
## as.factor(manuyear)1976 0.333896740
## as.factor(manuyear)1977 0.827840303
## as.factor(manuyear)1978 0.250972795
## as.factor(manuyear)1979 -0.124257296
## as.factor(manuyear)1980 -0.272600009
## as.factor(manuyear)1983 -0.717602420
## as.factor(manuyear)1985 0.385975249
## as.factor(manuyear)1986 0.069464013
## as.factor(manuyear)1987 -0.517702551
## as.factor(manuyear)1988 -0.056679320
## as.factor(manuyear)1989 0.465365173
## as.factor(manuyear)1990 0.428861832
## as.factor(manuyear)1991 0.370733357
## as.factor(manuyear)1992 0.578015511
## as.factor(manuyear)1993 0.703266855
## as.factor(manuyear)1994 0.214196986
## as.factor(manuyear)1995 0.383902171
## as.factor(manuyear)1996 0.132214945
## as.factor(manuyear)1997 0.185329639
## as.factor(manuyear)1998 0.150572393
## as.factor(manuyear)1999 0.209026629
## as.factor(manuyear)2000 0.180978808
## as.factor(manuyear)2001 0.185394528
## as.factor(manuyear)2002 0.189858823
## as.factor(manuyear)2003 0.237326410
## as.factor(manuyear)2004 0.116146122
## as.factor(manuyear)2005 0.093342803
## as.factor(manuyear)2006 0.074432237
## as.factor(manuyear)2007 -0.001915069
## as.factor(manuyear)2008 -0.031372406
## as.factor(manuyear)2009 0.111402520
## as.factor(manuyear)2010 -0.136312891
## as.factor(manuyear)2011 0.003673269
## as.factor(manuyear)2012 0.192184252
## as.factor(manuyear)2013 0.116314288

```



Confusion Matrix

```
##
## takeoffdelay.lda.prediction    0    1
##                               0 27315  26
##                               1  2646  13
```

Prediction Rate

```
## [1] 0.7912333
```

Our lda regression predicting outcome of the flight status(cancelled, delayed or normal) produced a result consistent 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] 0.7927667
```

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(manyear), 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 boundary is linear.

Decision Tree

```
#tree, doesn't work
tree.cancel<-tree(canceled~temp+dewp+wind_speed+precip+humid+pressure+visib+carrier+type+manuear+engines+seats+engine+origin.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+manuear+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+manuear+engines+seats+engine+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+manuear+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 prably is linear.

