TIM 209 Final report

YuanWu, Shilpi Jaiswal, Brhanu Berhea

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 isnt 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 isnt 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 flights 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 packages we utilized.

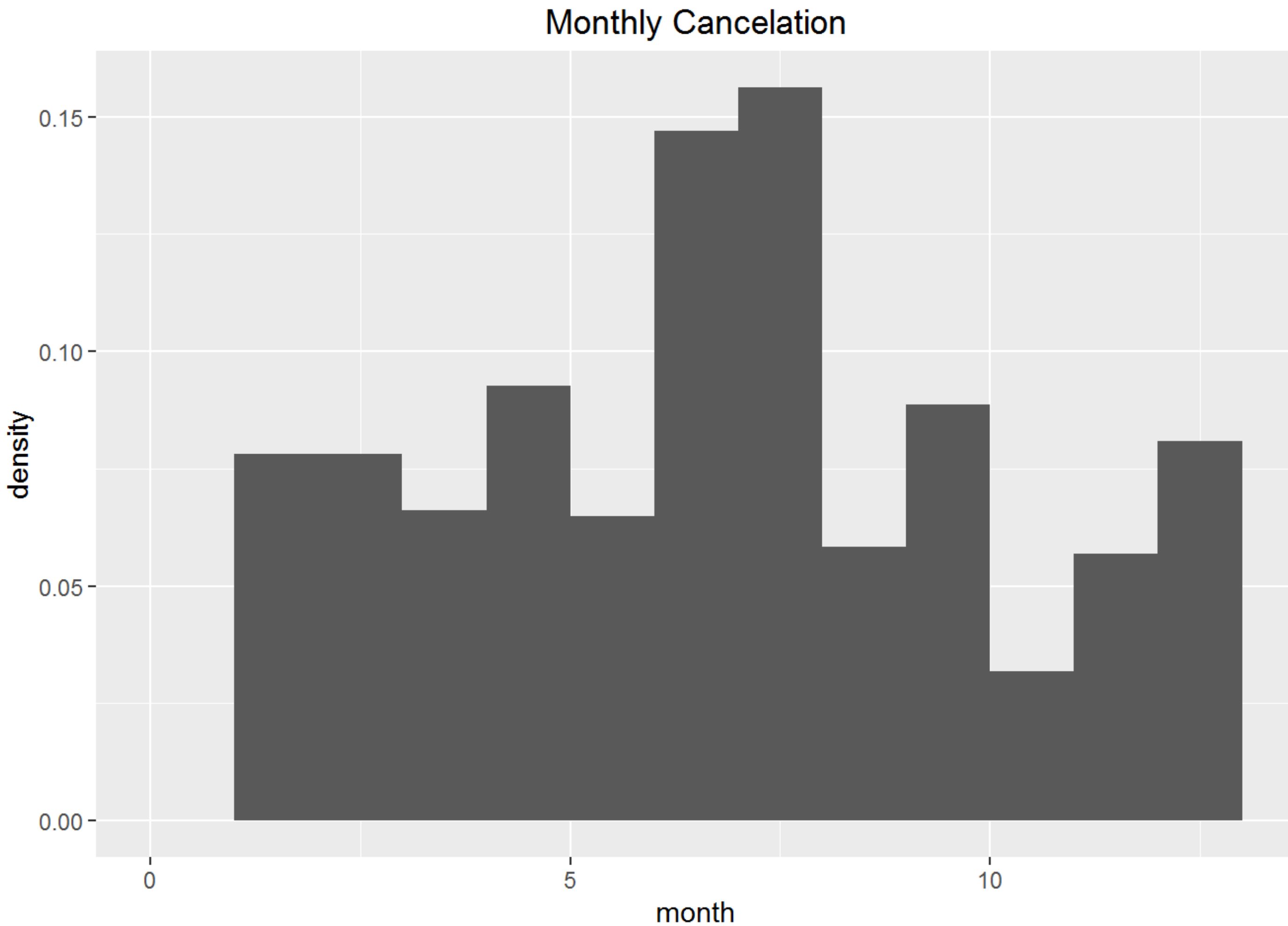
```
library(RSQLite)
library(dplyr)
library(nycflights13)
library(mfx)
library(ggplot2)
library(scales)
library(MASS)
library(ISLR)
library(class)
library(boot)
library(stargazer)
library(tree)
library(randomForest)
library(leaps)
library(glmnet)
library(gbm)
```

The Data we used is the nycflight13, which contains data regarding all flight information and weather readings in 2013 at New York City. We joined all subsets together to fit into our models.

We define “cancelation” as 1 if “air time” is N/A, which indicates the flight never took off. Delay is defined if “departure delay” is greater than 15 minutes, which is the industry standard. These two are dummy variables.

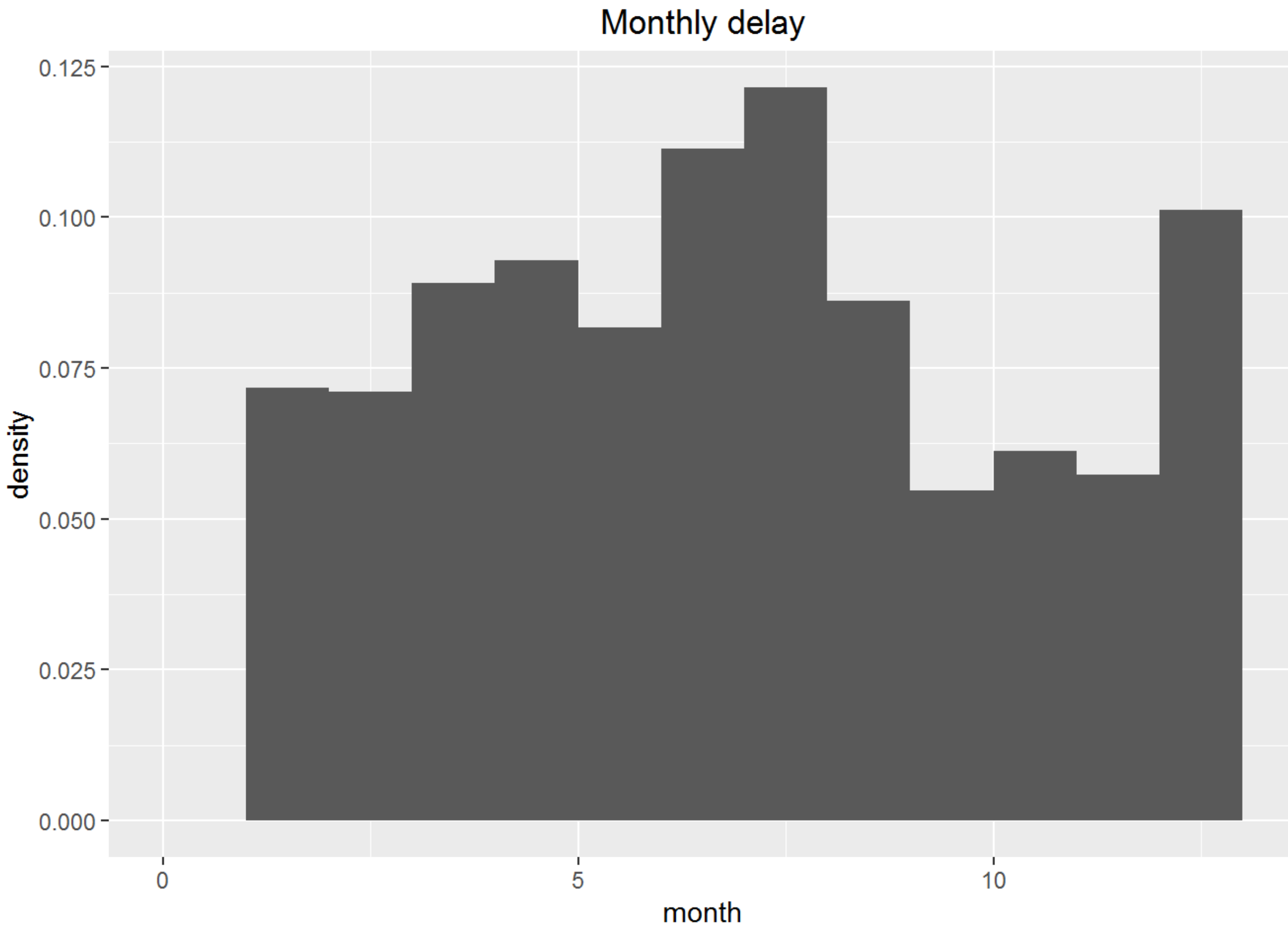
Here we check if time of the year has any effect on cancelation or delay, since it may correlate to weather conditions across seasons.

```
#histogram of montly cancelation. June and July has higher cancel rates
totalcanceled<-total[(total$canceled==1),]
cancelmonthplot<- ggplot(
  data = totalcanceled,
  aes(x=month,
      y=..density..)) +
  geom_histogram(binwidth = 1) +
  ggtitle("Monthly Cancelation")
cancelmonthplot
```



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 monthly delay. June, July December has higher delay rates
totaldelayed<-total[(total$takeoffdelay==1),]
delaymonthplot<- ggplot(
  data = totaldelayed,
  aes(x=month,
      y=..density..)) +
  geom_histogram(binwidth = 1) +
  ggtitle("Monthly delay")
delaymonthplot
```



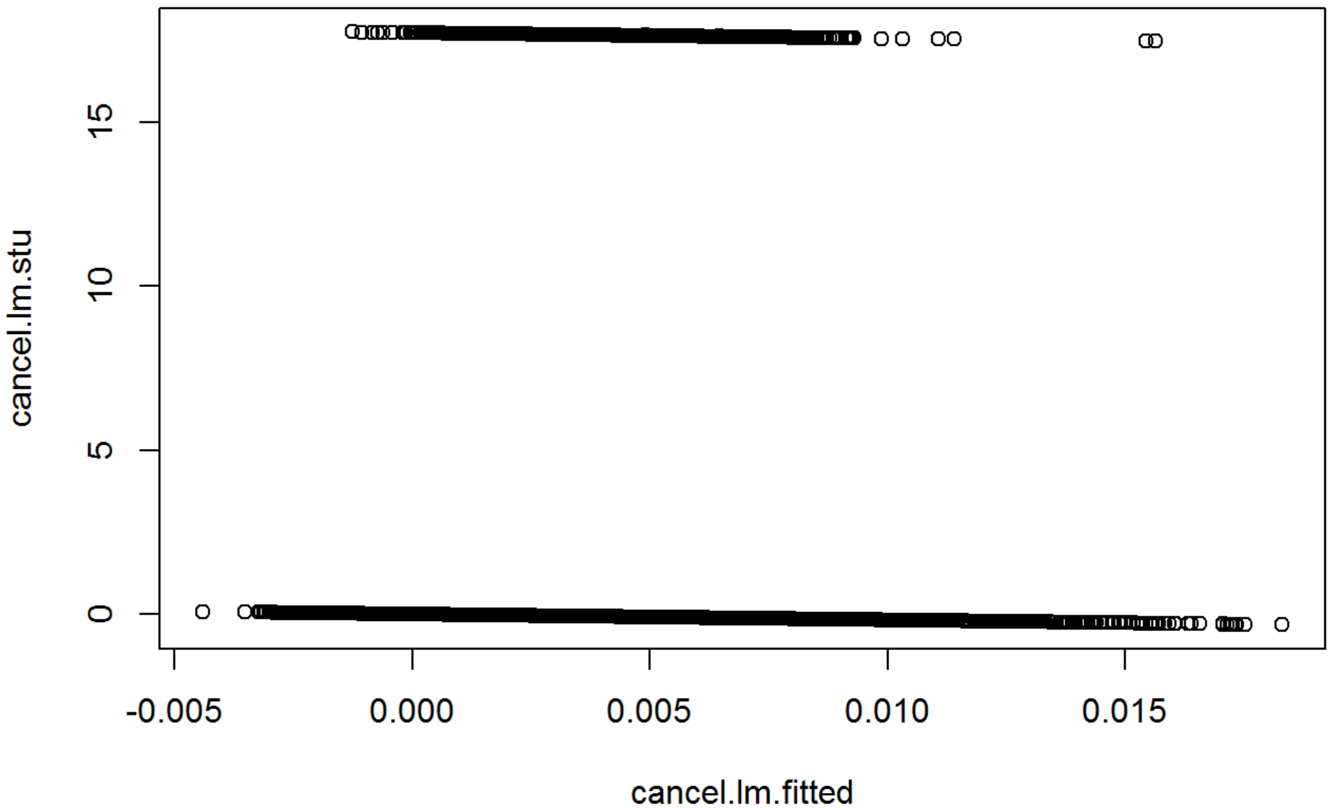
The histogram for delay is consistant with cancelation, but more delay occour during December in addition to June and July. This is likely due to the weather condition during winter like heavy snow or forzen runways, though can be alliviated, will cause delays.

Now we check if outliers exist within our data that can cause error in our regressions.

```
#Checking outlier
cancel.lm.fit<-lm(canceled~temp+dewp+wind_speed+precip+humid+pressure+visib+carrier+dista
nce, data=total, na.action=na.omit)
summary(cancel.lm.fit)
```

```
##
## Call:
## lm(formula = canceled ~ temp + dewp + wind_speed + precip + humid +
##     pressure + visib + carrier + distance, data = total, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.01830 -0.00457 -0.00307 -0.00167  1.00126
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.219e-02  1.753e-02   2.977  0.00291 **
## temp        -2.760e-04  6.894e-05  -4.003 6.26e-05 ***
## dewp         3.509e-04  7.407e-05   4.737 2.17e-06 ***
## wind_speed   3.501e-07  6.490e-06   0.054  0.95698
## precip       2.356e-02  8.302e-03   2.838  0.00454 **
## humid       -1.655e-04  3.623e-05  -4.567 4.94e-06 ***
## pressure    -3.069e-05  1.662e-05  -1.846  0.06489 .
## visib       -4.632e-04  9.752e-05  -4.750 2.04e-06 ***
## carrierAA   -2.702e-03  8.026e-04  -3.367  0.00076 ***
## carrierAS   -3.743e-03  2.345e-03  -1.596  0.11042
## carrierB6   -4.926e-03  5.428e-04  -9.075  < 2e-16 ***
## carrierDL   -5.049e-03  5.580e-04  -9.048  < 2e-16 ***
## carrierEV   -1.507e-03  5.364e-04  -2.809  0.00497 **
## carrierF9   -5.750e-03  2.511e-03  -2.290  0.02202 *
## carrierFL   -3.167e-03  1.221e-03  -2.595  0.00946 **
## carrierHA   -1.163e-02  3.553e-03  -3.274  0.00106 **
## carrierMQ   -7.154e-05  2.030e-03  -0.035  0.97189
## carrierOO   -6.881e-03  1.154e-02  -0.597  0.55083
## carrierUA   -4.489e-03  5.620e-04  -7.989 1.37e-15 ***
## carrierUS   -4.015e-03  6.392e-04  -6.281 3.37e-10 ***
## carrierVX   -5.987e-03  1.045e-03  -5.728 1.01e-08 ***
## carrierWN   -4.041e-03  7.336e-04  -5.508 3.63e-08 ***
## carrierYV   -6.286e-03  2.690e-03  -2.337  0.01946 *
## distance     1.279e-06  1.901e-07   6.728 1.72e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05646 on 235748 degrees of freedom
## Multiple R-squared:  0.00128,    Adjusted R-squared:  0.001183
## F-statistic: 13.14 on 23 and 235748 DF,  p-value: < 2.2e-16
```

```
cancel.lm.stu<-studres(cancel.lm.fit)
cancel.lm.fitted<-fitted(cancel.lm.fit)
plot(cancel.lm.stu~cancel.lm.fitted)
```



According to the result, we have a group of data that have studentized error that exceed |3|, but they are consistent within the group, indicating that they are not outliers but observations with different scales.

Now we construct our training and test subsets. We split our dataset into two roughly equal size by a random draw, but make the test set smaller. The result is 100000 observations in test set and 135772 observations in training set.

```
test.ind<-sample(nrow(total), 100000)
total.test<-total[test.ind,]
total.train<-total[-test.ind,]
```

From here we start to construct different models with differnt methods covered in class to compare the prediction accuracies.

```
cancel.log.fit<-glm(canceled~temp+dewp+wind_speed+precip+humid+pressure+visib+carrier+typ
e+manuyear+engines+seats+engine+origin.x+distance, family=binomial, data=total.train, na.
action=na.omit)
summary(cancel.log.fit)
```

```
##
## Call:
## glm(formula = canceled ~ temp + dewp + wind_speed + precip +
##     humid + pressure + visib + carrier + type + manuyear + engines +
##     seats + engine + origin.x + distance, family = binomial,
##     data = total.train, na.action = na.omit)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.4230  -0.0895  -0.0704  -0.0550   3.9104
##
## Coefficients: (1 not defined because of singularities)
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   9.376e+01  1.142e+03   0.082  0.93457
## temp                         -5.591e-02  3.323e-02  -1.682  0.09251 .
## dewp                          7.706e-02  3.549e-02   2.171  0.02990 *
## wind_speed                    1.999e-04  2.624e-03   0.076  0.93927
## precip                        4.108e+00  1.947e+00   2.110  0.03488 *
## humid                        -3.439e-02  1.712e-02  -2.009  0.04453 *
## pressure                     -1.157e-02  7.255e-03  -1.595  0.11070
## visib                        -1.000e-01  3.591e-02  -2.786  0.00534 **
## carrierAA                    -1.594e+00  3.998e-01  -3.987 6.70e-05 ***
## carrierAS                    -1.836e+00  1.038e+00  -1.768  0.07710 .
## carrierB6                    -1.381e+00  2.220e-01  -6.221 4.95e-10 ***
## carrierDL                    -1.864e+00  2.720e-01  -6.854 7.18e-12 ***
## carrierEV                    -5.997e-01  2.174e-01  -2.759  0.00580 **
## carrierF9                    -1.359e+00  1.032e+00  -1.317  0.18770
## carrierFL                    -8.478e-01  4.553e-01  -1.862  0.06261 .
## carrierHA                    -1.544e+01  5.090e+02  -0.030  0.97580
## carrierMQ                    -1.243e+00  8.405e-01  -1.479  0.13909
## carrierOO                    -1.384e+01  1.680e+03  -0.008  0.99343
## carrierUA                    -1.677e+00  2.887e-01  -5.810 6.25e-09 ***
## carrierUS                    -1.232e+00  3.049e-01  -4.042 5.30e-05 ***
## carrierVX                    -2.318e+00  5.496e-01  -4.217 2.47e-05 ***
## carrierWN                    -1.287e+00  3.183e-01  -4.042 5.31e-05 ***
## carrierYV                    -1.376e+01  3.920e+02  -0.035  0.97200
## typeFixed wing single engine  6.302e-01  8.967e+02   0.001  0.99944
## typeRotorcraft               -4.143e-01  8.967e+02   0.000  0.99963
## manuyear                     -3.525e-02  1.252e-02  -2.815  0.00487 **
## engines                      -1.224e+01  4.080e+02  -0.030  0.97608
## seats                        -9.352e-04  1.290e-03  -0.725  0.46829
## engineReciprocating          -2.332e+00  1.287e+00  -1.812  0.07005 .
## engineTurbo-fan               1.093e+01  7.985e+02   0.014  0.98908
## engineTurbo-jet               1.073e+01  7.985e+02   0.013  0.98928
## engineTurbo-prop             -2.735e+00  1.683e+03  -0.002  0.99870
## engineTurbo-shaft                    NA         NA      NA       NA
## origin.xJFK                  -3.307e-01  1.728e-01  -1.914  0.05564 .
## origin.xLGA                   1.133e-01  1.506e-01   0.753  0.45173
## distance                      6.053e-04  9.408e-05   6.434 1.25e-10 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5692.6  on 135771  degrees of freedom
## Residual deviance: 5504.5  on 135737  degrees of freedom
## AIC: 5574.5
##
## Number of Fisher Scoring iterations: 17
```

```
cancel.log.prob<-predict(cancel.log.fit,total.test,type="response")
cancel.log.fit.test<-ifelse(cancel.log.prob>0.5,1,0)
cancel.log.fit.test <- cancel.log.fit.test
mean(cancel.log.fit.test==total.test$canceled, na.rm=T)
```

```
## [1] 0.99665
```

From a simplle logistic regression, we manage to predict cancelation with 99.675% accuracy. This is very promising, considering the data we fit into the model is actual historical data.

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

```
##
## Call:
## glm(formula = takeoffdelay ~ temp + dewp + wind_speed + precip +
##     humid + pressure + visib + carrier + type + manuyear + engines +
##     seats + engine + origin.x + distance, family = binomial,
##     data = total.train, na.action = na.omit)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8285  -0.7182  -0.5810  -0.4117   2.6650
##
## Coefficients: (1 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  4.268e+01  3.640e+00  11.726  < 2e-16 ***
## temp                        -1.853e-02  4.002e-03  -4.631 3.64e-06 ***
## dewp                         3.358e-02  4.299e-03   7.811 5.69e-15 ***
## wind_speed                   1.492e-03  2.896e-04   5.151 2.59e-07 ***
## precip                       3.150e+00  4.409e-01   7.146 8.91e-13 ***
## humid                       -2.990e-02  2.157e-03 -13.859  < 2e-16 ***
## pressure                    -3.026e-02  1.004e-03 -30.143  < 2e-16 ***
## visib                       -1.720e-01  5.425e-03 -31.701  < 2e-16 ***
## carrierAA                   -6.953e-01  6.150e-02 -11.306  < 2e-16 ***
## carrierAS                   -9.079e-01  1.668e-01  -5.442 5.27e-08 ***
## carrierB6                   -1.584e-01  3.150e-02  -5.030 4.91e-07 ***
## carrierDL                   -6.722e-01  3.892e-02 -17.269  < 2e-16 ***
## carrierEV                    2.522e-01  3.603e-02   6.999 2.58e-12 ***
## carrierF9                    1.060e-01  1.391e-01   0.762  0.44592
## carrierFL                    8.322e-02  7.066e-02   1.178  0.23891
## carrierHA                   -1.363e+00  3.226e-01  -4.225 2.39e-05 ***
## carrierMQ                   -3.926e-01  1.493e-01  -2.629  0.00856 **
## carrierOO                   -4.738e-01  6.486e-01  -0.731  0.46505
## carrierUA                   -3.323e-01  4.167e-02  -7.975 1.52e-15 ***
## carrierUS                   -9.615e-01  4.725e-02 -20.349  < 2e-16 ***
## carrierVX                   -4.241e-01  6.483e-02  -6.541 6.12e-11 ***
## carrierWN                    1.197e-01  4.500e-02   2.659  0.00784 **
## carrierYV                    2.732e-02  1.456e-01   0.188  0.85111
## typeFixed wing single engine  1.157e+00  5.442e-01   2.126  0.03353 *
## typeRotorcraft               4.643e-01  7.115e-01   0.653  0.51396
## manuyear                    -5.216e-03  1.714e-03  -3.043  0.00234 **
## engines                      2.601e-01  1.530e-01   1.700  0.08918 .
## seats                        6.690e-04  1.532e-04   4.368 1.26e-05 ***
## engineReciprocating         -9.174e-01  4.553e-01  -2.015  0.04390 *
## engineTurbo-fan             -1.006e-01  6.653e-01  -0.151  0.87979
## engineTurbo-jet             -1.312e-01  6.654e-01  -0.197  0.84372
## engineTurbo-prop             6.860e-01  8.446e-01   0.812  0.41667
## engineTurbo-shaft                   NA         NA      NA       NA
## origin.xJFK                 -5.890e-02  2.427e-02  -2.426  0.01525 *
## origin.xLGA                 -1.263e-01  2.150e-02  -5.877 4.19e-09 ***
## distance                    -4.232e-06  1.316e-05  -0.321  0.74783
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 137720  on 135771  degrees of freedom
## Residual deviance: 131067  on 135737  degrees of freedom
## AIC: 131137
##
## Number of Fisher Scoring iterations: 4
```

```
takeoffdelay.log.prob<-predict(takeoffdelay.log.fit,total.test,type="response")
takeoffdelay.log.fit.test<-ifelse(takeoffdelay.log.prob>0.5,1,0)
takeoffdelay.log.fit.test <- takeoffdelay.log.fit.test
mean(takeoffdelay.log.fit.test==total.test$takeoffdelay, na.rm=T)
```

```
## [1] 0.79552
```

Prediction for delay is slightly lower in accuracy, but still able to reach about 80%. Due to many more variables that can cause delay, this prediction is still valid.

```
#lda, works
flightstate.lda.fit<-lda(flightstate~temp+dewp+wind_speed+precip+humid+pressure+visib+car
rier+type+manuyear+engines+seats+engine+origin.x+distance, data=total.train)
flightstate.lda.fit
```

```
## Call:
## lda(flightstate ~ temp + dewp + wind_speed + precip + humid +
##     pressure + visib + carrier + type + manuyear + engines +
##     seats + engine + origin.x + distance, data = total.train)
##
## Prior probabilities of groups:
##    Canceled     Delayed      Normal
## 0.003093421 0.203679698 0.793226880
##
## Group means:
##              temp     dewp wind_speed      precip    humid pressure
## Canceled 61.08929 46.68586   9.735051 0.004380952 61.88088 1016.433
## Delayed  59.31259 43.13141  10.882307 0.002968106 58.11496 1016.246
## Normal   54.61887 39.85181   9.580576 0.001372820 60.20460 1018.144
##             visib  carrierAA   carrierAS carrierB6 carrierDL carrierEV
## Canceled 9.314286 0.05238095 0.002380952 0.1261905 0.1119048 0.2833333
## Delayed  9.380730 0.02632531 0.001627251 0.1996818 0.1213929 0.2538873
## Normal   9.611131 0.03775372 0.002915560 0.1903935 0.1826032 0.1629092
##            carrierF9  carrierFL    carrierHA   carrierMQ    carrierOO
## Canceled 0.002380952 0.01428571 0.0000000000 0.004761905 0.0000000000
## Delayed  0.002820568 0.01258407 0.0003977725 0.003182180 0.0001084834
## Normal   0.002042749 0.01025089 0.0013927835 0.003361251 0.0001114227
##          carrierUA  carrierUS  carrierVX  carrierWN   carrierYV
## Canceled 0.1857143 0.05000000 0.00952381 0.04047619 0.000000000
## Delayed  0.1961742 0.03641426 0.01551313 0.05409706 0.002495118
## Normal   0.2061691 0.07853442 0.01929469 0.04028858 0.001875615
##          typeFixed wing single engine typeRotorcraft manuyear  engines
## Canceled                  0.007142857    0.002380952 2000.779 1.990476
## Delayed                   0.003833080    0.001627251 2001.792 1.996022
## Normal                    0.004466193    0.001439210 2001.338 1.995348
##             seats engineReciprocating engineTurbo-fan engineTurbo-jet
## Canceled 127.4905         0.004761905       0.8666667       0.1238095
## Delayed  129.0802         0.003724597       0.8669270       0.1272510
## Normal   140.5988         0.004828316       0.8409720       0.1524727
##          engineTurbo-prop engineTurbo-shaft origin.xJFK origin.xLGA
## Canceled     0.0000000000       0.002380952   0.2952381   0.2476190
## Delayed      0.0001808057       0.001627251   0.3231359   0.2226079
## Normal       0.0001299931       0.001439210   0.3413062   0.2733106
##          distance
## Canceled 1155.836
## Delayed  1036.025
## Normal   1089.013
##
## Coefficients of linear discriminants:
##                                          LD1           LD2
## temp                            2.539164e-02  -0.087517641
## dewp                           -5.202615e-02   0.106476875
## wind_speed                     -3.501364e-03  -0.002323602
## precip                         -8.241616e+00  12.086350629
## humid                           4.919843e-02  -0.029565520
```

```
## pressure                       5.222296e-02   0.019774784
## visib                          3.191822e-01   0.035145269
## carrierAA                      1.222745e+00  -1.897638846
## carrierAS                      1.507500e+00  -2.613814992
## carrierB6                      3.257085e-01  -2.427140020
## carrierDL                      1.176021e+00  -2.583972365
## carrierEV                     -5.188836e-01  -1.703332197
## carrierF9                     -1.825900e-01  -2.972663406
## carrierFL                     -1.181321e-01  -1.911433550
## carrierHA                      1.807634e+00  -5.570697748
## carrierMQ                      7.227852e-01  -1.522807306
## carrierOO                      9.938862e-01  -3.580850931
## carrierUA                      6.450823e-01  -2.747539739
## carrierUS                      1.475529e+00  -1.345587960
## carrierVX                      8.000915e-01  -3.662857240
## carrierWN                     -2.179590e-01  -2.839862402
## carrierYV                     -2.739435e-02  -4.078827746
## typeFixed wing single engine -1.530093e+00  -0.445389818
## typeRotorcraft                 9.600063e-02  -8.725596847
## manuyear                       1.024991e-02  -0.044139564
## engines                       -4.101087e-01  -1.920705413
## seats                         -9.934845e-04  -0.002756616
## engineReciprocating            2.187584e+00 -18.923881491
## engineTurbo-fan                1.145967e+00 -16.674232043
## engineTurbo-jet                1.215270e+00 -16.858505930
## engineTurbo-prop              -1.680695e-01 -21.202433849
## engineTurbo-shaft              9.600063e-02  -8.725596847
## origin.xJFK                    1.223639e-01  -0.511381036
## origin.xLGA                    2.159031e-01   0.277306059
## distance                      -9.813273e-06   0.001034037
##
## Proportion of trace:
##    LD1    LD2
## 0.9819 0.0181
```

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

```
## [1] 0.7904575
```

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.

```
#Qda, doesn't work
flightstate.qda.fit<-qda(flightstate~temp+dewp+wind_speed+precip+humid+pressure+visib+car
rier+distance, data=total.train)
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.

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+carri
er+type+manuyear+engines+seats+engine+origin.x+distance, family=binomial, data=total, na.
action=na.omit)
summary(total.cancel.log.fit)
```

```
##
## Call:
## glm(formula = canceled ~ 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)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.3564  -0.0901  -0.0718  -0.0573   3.8383
##
## Coefficients: (1 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  4.980e+01  8.946e+02   0.056 0.955611
## temp                        -9.349e-02  2.607e-02  -3.586 0.000336 ***
## dewp                         1.175e-01  2.785e-02   4.217 2.47e-05 ***
## wind_speed                   2.087e-04  2.016e-03   0.104 0.917544
## precip                       3.312e+00  1.534e+00   2.159 0.030825 *
## humid                       -5.480e-02  1.337e-02  -4.097 4.19e-05 ***
## pressure                    -1.048e-02  5.406e-03  -1.939 0.052535 .
## visib                       -1.310e-01  2.677e-02  -4.895 9.81e-07 ***
## carrierAA                   -1.042e+00  2.879e-01  -3.621 0.000294 ***
## carrierAS                   -1.348e+00  6.133e-01  -2.197 0.027995 *
## carrierB6                   -1.395e+00  1.619e-01  -8.619  < 2e-16 ***
## carrierDL                   -1.623e+00  1.967e-01  -8.254  < 2e-16 ***
## carrierEV                   -6.534e-01  1.605e-01  -4.072 4.66e-05 ***
## carrierF9                   -1.951e+00  1.017e+00  -1.917 0.055194 .
## carrierFL                   -1.067e+00  3.834e-01  -2.782 0.005404 **
## carrierHA                   -1.523e+01  3.913e+02  -0.039 0.968948
## carrierMQ                   -6.824e-01  5.586e-01  -1.222 0.221837
## carrierOO                   -1.394e+01  1.327e+03  -0.011 0.991619
## carrierUA                   -1.530e+00  2.095e-01  -7.302 2.84e-13 ***
## carrierUS                   -1.214e+00  2.250e-01  -5.394 6.89e-08 ***
## carrierVX                   -1.756e+00  3.343e-01  -5.252 1.50e-07 ***
## carrierWN                   -1.330e+00  2.415e-01  -5.505 3.69e-08 ***
## carrierYV                   -1.380e+01  3.028e+02  -0.046 0.963653
## typeFixed wing single engine  1.569e+00  6.798e+02   0.002 0.998158
## typeRotorcraft               1.807e-01  6.798e+02   0.000 0.999788
## manuyear                    -1.363e-02  9.170e-03  -1.486 0.137266
## engines                     -1.188e+01  3.356e+02  -0.035 0.971768
## seats                       -1.299e-03  9.388e-04  -1.383 0.166586
## engineReciprocating         -1.517e+00  1.120e+00  -1.354 0.175627
## engineTurbo-fan              1.186e+01  5.911e+02   0.020 0.983994
## engineTurbo-jet              1.177e+01  5.911e+02   0.020 0.984112
## engineTurbo-prop            -1.522e+00  1.256e+03  -0.001 0.999033
## engineTurbo-shaft                   NA         NA      NA       NA
## origin.xJFK                 -3.917e-01  1.282e-01  -3.054 0.002255 **
## origin.xLGA                  3.415e-02  1.132e-01   0.302 0.762832
## distance                     5.455e-04  6.972e-05   7.825 5.07e-15 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10180.9  on 235771  degrees of freedom
## Residual deviance:  9864.8  on 235737  degrees of freedom
## AIC: 9934.8
##
## Number of Fisher Scoring iterations: 17
```

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

```
##
## Call:
## glm(formula = 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)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8024  -0.7180  -0.5824  -0.4146   2.6741
##
## Coefficients: (1 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  4.496e+01  2.761e+00  16.281  < 2e-16 ***
## temp                        -2.228e-02  3.052e-03  -7.302 2.84e-13 ***
## dewp                         3.730e-02  3.279e-03  11.374  < 2e-16 ***
## wind_speed                   1.548e-03  2.379e-04   6.507 7.67e-11 ***
## precip                       2.790e+00  3.319e-01   8.407  < 2e-16 ***
## humid                       -3.167e-02  1.644e-03 -19.263  < 2e-16 ***
## pressure                    -2.965e-02  7.611e-04 -38.957  < 2e-16 ***
## visib                       -1.769e-01  4.127e-03 -42.853  < 2e-16 ***
## carrierAA                   -6.816e-01  4.655e-02 -14.640  < 2e-16 ***
## carrierAS                   -8.747e-01  1.274e-01  -6.868 6.49e-12 ***
## carrierB6                   -1.506e-01  2.381e-02  -6.325 2.53e-10 ***
## carrierDL                   -6.646e-01  2.943e-02 -22.581  < 2e-16 ***
## carrierEV                    2.455e-01  2.723e-02   9.016  < 2e-16 ***
## carrierF9                    6.209e-02  1.069e-01   0.581  0.56120
## carrierFL                    3.328e-02  5.411e-02   0.615  0.53856
## carrierHA                   -1.351e+00  2.506e-01  -5.391 7.02e-08 ***
## carrierMQ                   -5.057e-01  1.128e-01  -4.484 7.31e-06 ***
## carrierOO                   -4.327e-01  5.057e-01  -0.856  0.39217
## carrierUA                   -3.164e-01  3.155e-02 -10.030  < 2e-16 ***
## carrierUS                   -9.479e-01  3.580e-02 -26.481  < 2e-16 ***
## carrierVX                   -4.526e-01  4.977e-02  -9.093  < 2e-16 ***
## carrierWN                    8.597e-02  3.430e-02   2.507  0.01219 *
## carrierYV                    1.005e-01  1.101e-01   0.913  0.36110
## typeFixed wing single engine  5.137e-01  3.431e-01   1.497  0.13429
## typeRotorcraft               9.493e-02  5.174e-01   0.183  0.85445
## manuyear                    -6.362e-03  1.299e-03  -4.898 9.67e-07 ***
## engines                      1.809e-01  1.267e-01   1.428  0.15329
## seats                        5.059e-04  1.164e-04   4.347 1.38e-05 ***
## engineReciprocating         -5.540e-01  3.836e-01  -1.444  0.14869
## engineTurbo-fan             -3.160e-01  4.781e-01  -0.661  0.50861
## engineTurbo-jet             -3.503e-01  4.782e-01  -0.733  0.46384
## engineTurbo-prop            -1.061e-01  6.562e-01  -0.162  0.87151
## engineTurbo-shaft                   NA         NA      NA       NA
## origin.xJFK                 -5.278e-02  1.842e-02  -2.865  0.00417 **
## origin.xLGA                 -1.262e-01  1.633e-02  -7.731 1.07e-14 ***
## distance                    -2.680e-06  9.993e-06  -0.268  0.78853
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239000  on 235771  degrees of freedom
## Residual deviance: 227710  on 235737  degrees of freedom
## AIC: 227780
##
## Number of Fisher Scoring iterations: 4
```

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
```

```
## [1] 0.003187523 0.003187459
```

```
takeoffdelay.cv.err<-cv.glm(total, total.takeoffdelay.log.fit, K=10)
takeoffdelay.cv.err$delta
```

```
## [1] 0.1548751 0.1548723
```

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

We attempted to fit our data into a tree model, with little success.

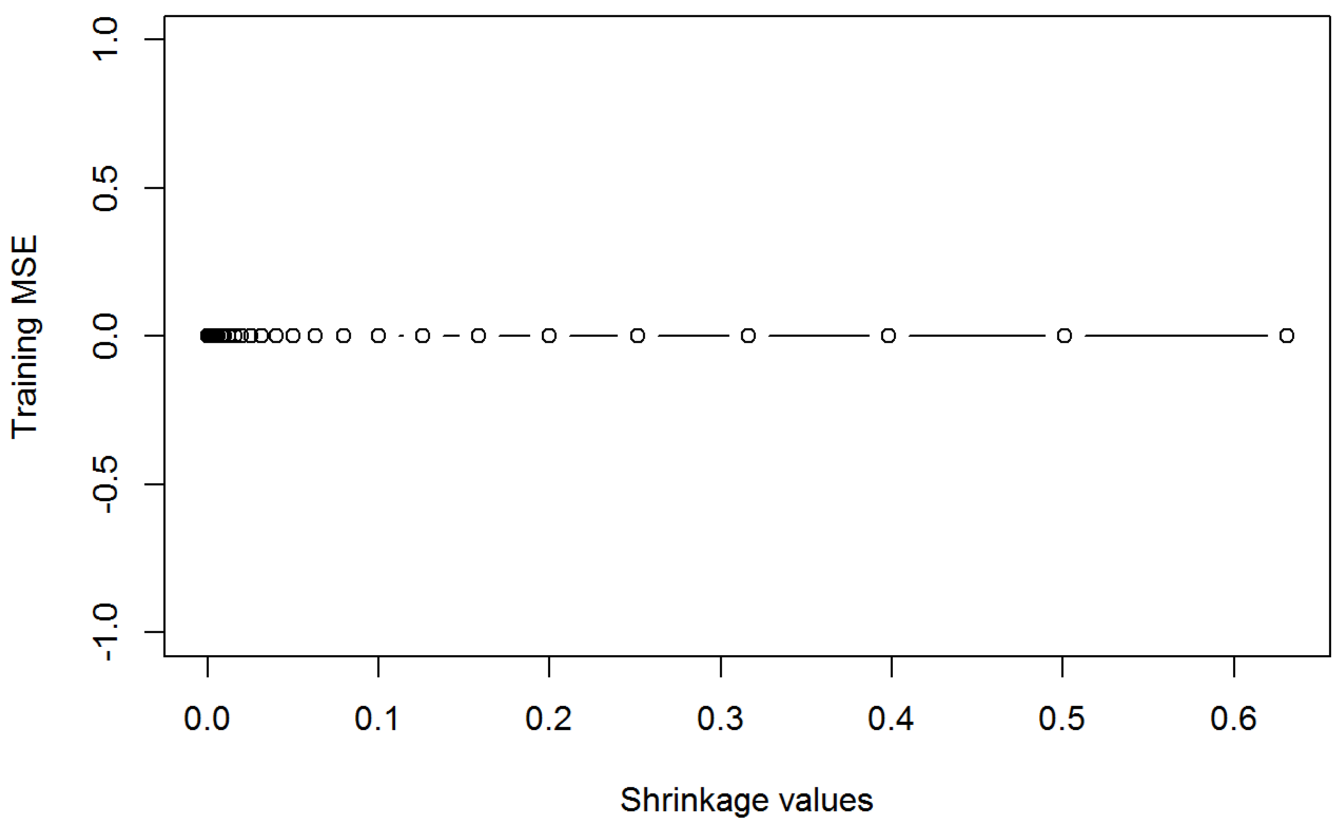
```
#tree, doesn't work
tree.cancel<-tree(canceled~temp+dewp+wind_speed+precip+humid+pressure+visib+carrier+type+
manuyear+engines+seats+engine+origin.x+distance, data=total.train)
summary(tree.cancel)
```

```
##
## Regression tree:
## tree(formula = canceled ~ temp + dewp + wind_speed + precip +
##     humid + pressure + visib + carrier + type + manuyear + engines +
##     seats + engine + origin.x + distance, data = total.train)
## Variables actually used in tree construction:
## character(0)
## Number of terminal nodes:  1
## Residual mean deviance:  0.003084 = 418.7 / 135800
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.003093 -0.003093 -0.003093  0.000000 -0.003093  0.996900
```

What we ends up with is a single nod tree with zero variables used to construct branches. This infact confirmed our high accuracy in our glm prediction and is a tell tale sign that the broundry is highly linear.

We tried a final effort by fitting a 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+sea
ts+engines+distance+month+day+hour, data = total.train, distribution = "gaussian", n.tree
s = 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)
```

```
## [1] "character"
```

```
cancel.rf<-randomForest(canceled~ temp+dewp+wind_speed+precip+humid+pressure+visib+distan
ce, data = total.train, mtry = 3, ntree = 500, importance = TRUE)
cancel.rf
```

```
##
## Call:
##  randomForest(formula = canceled ~ temp + dewp + wind_speed +      precip + humid + pr
essure + visib + distance, data = total.train,      mtry = 3, ntree = 500, importance = T
RUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 0.003191118
##                     % Var explained: -3.48
```

Again with no success.

We tested our model with K nearest neighber method, again K=10 to reduce computation time.

```
#Knn
cancel.KNN.test1<-knn(total.train[,23:31], total.test[,23:31], total.train$canceled, k=1
0)
mean(cancel.KNN.test1==total.test$canceled)
```

```
## [1] 0.99665
```

```
takeoffdelay.KNN.test1<-knn(total.train[,23:31], total.test[,23:31], total.train$takeoffd
elay, k=10)
mean(takeoffdelay.KNN.test1==total.test$takeoffdelay)
```

```
## [1] 0.81946
```

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

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.