

# Capstone Project

## Flight Delays Analysis

## Contents

Background and Motivation .....	3
Client Benefits .....	3
Goals of Capstone Project.....	4
Data .....	5
Data at a glance – 2008 flights .....	5
Data Munging & Data Exploration .....	7
Predictive Models for Flight Delays .....	18
Approach:.....	19
Iteration #1 - Data Preparation.....	20
Models and Analysis .....	21
Iteration #2 - Input and Process Weather Data .....	32
Data Wrangling – Steps.....	33
Models and Analysis (Weather Data combined) .....	34
Iteration #3 – Running models with full dataset .....	43
Findings .....	45

## Background and Motivation

Every year approximately 20% of airline flights are delayed or cancelled, resulting in significant costs to both travelers and airlines. As a frequent traveler, we are always apprehensive about my flight getting delayed and we keep wondering if only we knew what causes the delay and predict all the factors with reasonable confidence, then not only the airlines will have more control over their planning and mitigate all the frustrations caused to the traveler. Historically,

Flight delays cost the airline industry \$8 billion a year due to increased spending on crews, fuel and maintenance. (Source: FAA 2010)

Delay cost passengers nearly \$17 billion. (Source FAA 2010)

## Client Benefits

Accurately predicting airline delay will allow

- Client Airline to proactively identify potential causes and find ways to alleviate such causes
- Passengers to be mentally prepared and reduce stress and anxiety due to uncertainty of delay

## Goals of Capstone Project

- Build a supervised learning model that predicts airline delay from historical flight information.
- Follow a two pronged approach to draw inferences.

### **Explicative inference**

Explore some of the potential causes of delays.

- Do some airports systematically generate delays independently of the company?
- Are some airlines just less timely than other airlines?
- What is the influence of the time of day?
- Are some airports more prone to weather delays than others?

### **Predictive inference**

My aim is to classify delay and non-delays. Inputs are the airline carrier, the departure airport, the destination airport, the time of the day, weather, and other variables deemed relevant by a statistical analysis. Output is the classification estimates in terms of binary measures – delays and non-delays.

## Data

We used four different data sources

1. The official flight database for every domestic flight in the US
2. Historical weather data
3. Airport information with geodata and names (e.g. for visualization and interpretation of results)
4. Information about aircraft models

### Data Sources:

- The 2008 departure and arrival data is from the American Statistical Association
  - <http://stat-computing.org/dataexpo/2008/the-data.html>
- Historical weather and flight demand data for 2008 is from the FAA Aviation Systems Performance Metrics (ASPM) (<https://aspm.faa.gov/>)
  - Data was downloaded in two files (one for the 1st half of 2008 and the other for the 2nd half of 2008)

For this exercise, we decided to use 2008 dataset not for any special reason but purely arbitrary for academic exercise. Despite the dataset being too large, we were still able to connect this dataset with external weather data.

### Data at a glance – 2008 flights

Total flights in 2008 – 7 million

Average daily flights – 19,204

Average flight time – 2 hours 7 minutes

Average flight distance – 726 miles

Airline with highest daily operations – 3252 flights (SouthWest)

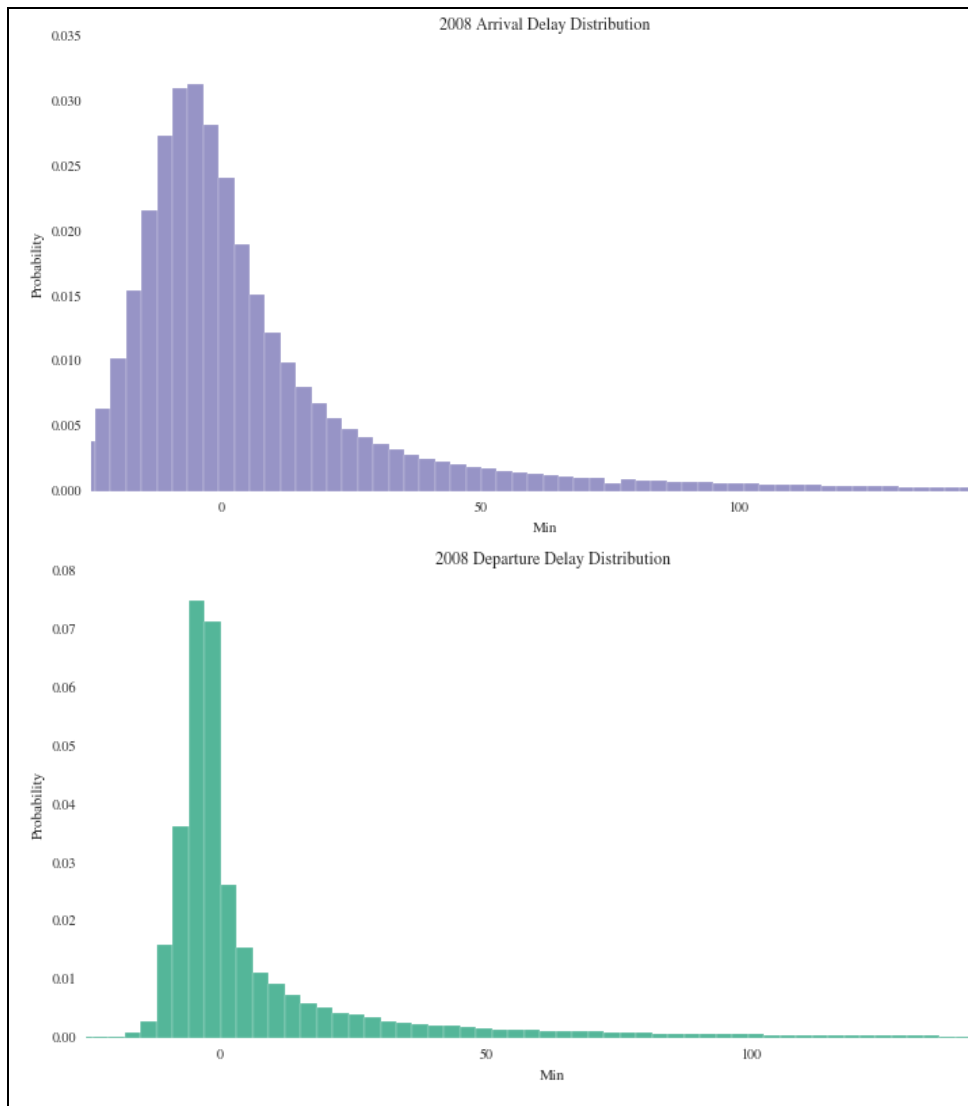
Airline with highest departure delay – 14.1 minutes (United)

Airline with highest arrival delay – 12.6 minutes (American)

## Data Munging & Data Exploration

We check the structure of the dataset and see that there are several columns that correspond to 'Delays' and also 'Cancelled' flights. We would eventually want to exclude these, since cancelled flights have no delay attributes and restricting the analysis to delayed flights means that we would miss non-delayed flights.

We need to generate departure and arrival hour attributes from the departure time variables, in order to look at the time of day effect on delays.



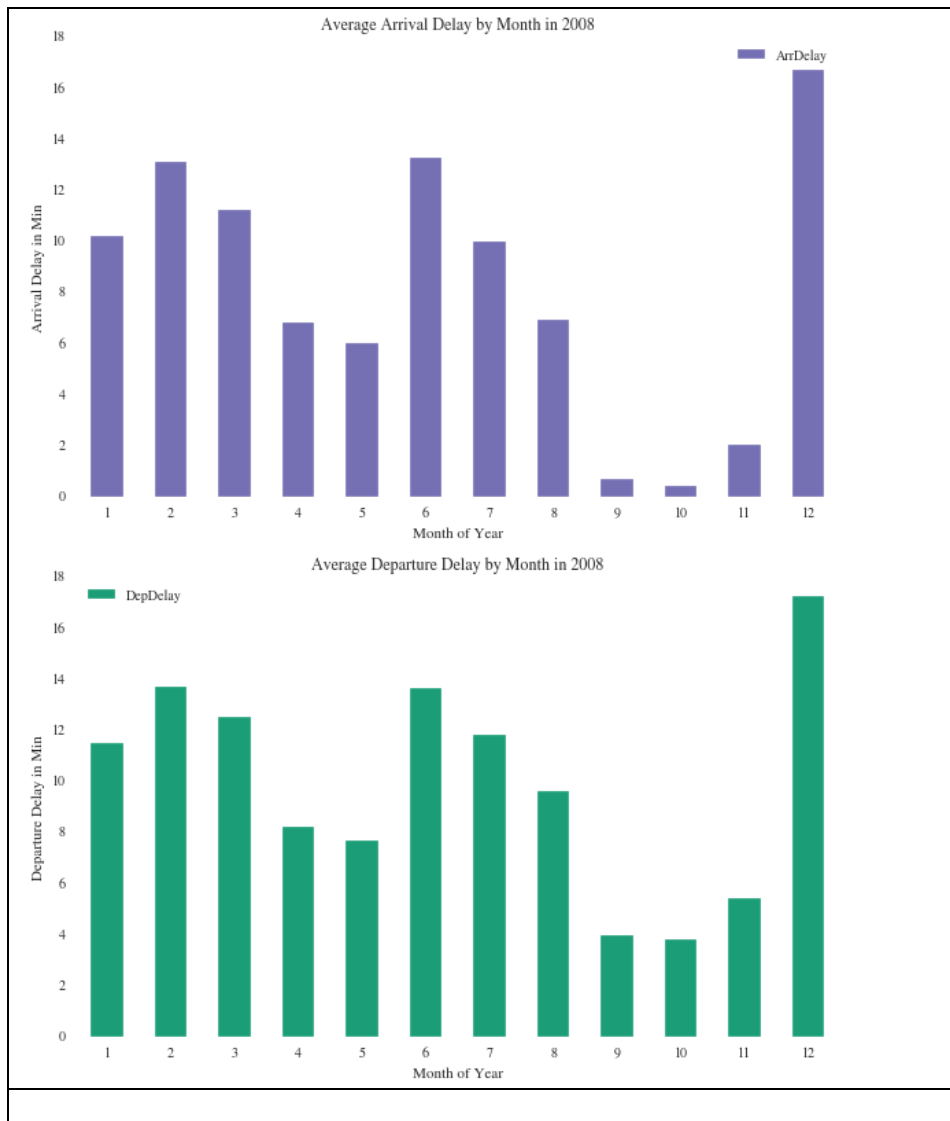
## 2008 Departure Delay and Arrival Delay Distribution

We notice a much higher probability of short delays - actually negative, so advances - for departure delays and a wider distribution (in minutes) for arrivals. Notice the long right-hand tails. Some flights are delayed for very long times, over two hours. On the other hand, the delays are centered on just below zero. In both cases, the mode of the distribution is less than zero, meaning most of the flights leave from gate and arrive at gate even before the published schedule time of departure and arrival. As we will show below, the longer delays cancel out the shorter negative delays (advances), leading to average delays that are above zero.

The x-axis for the two plots are to scale. As a result, we can see that the arrival delay distribution, compared with the departure delay distribution, leans toward left. A flight delay is defined by the schedule time of an event compared against the actual time of the event. Airlines usually put extra buffer time in a flight to ensure on-time arrival. Therefore, the departure delay and arrival delay distributions difference indicates that some departure delays are recovered during the flights due to the extra amount of time embedded in the flight time between two airports.

**Next**, we consider the impact of month on the delays. We would expect that winter months have the longest delay. A column chart with departure and arrival delay in minutes plotted by month is the most effective way to see the potential effects of month.

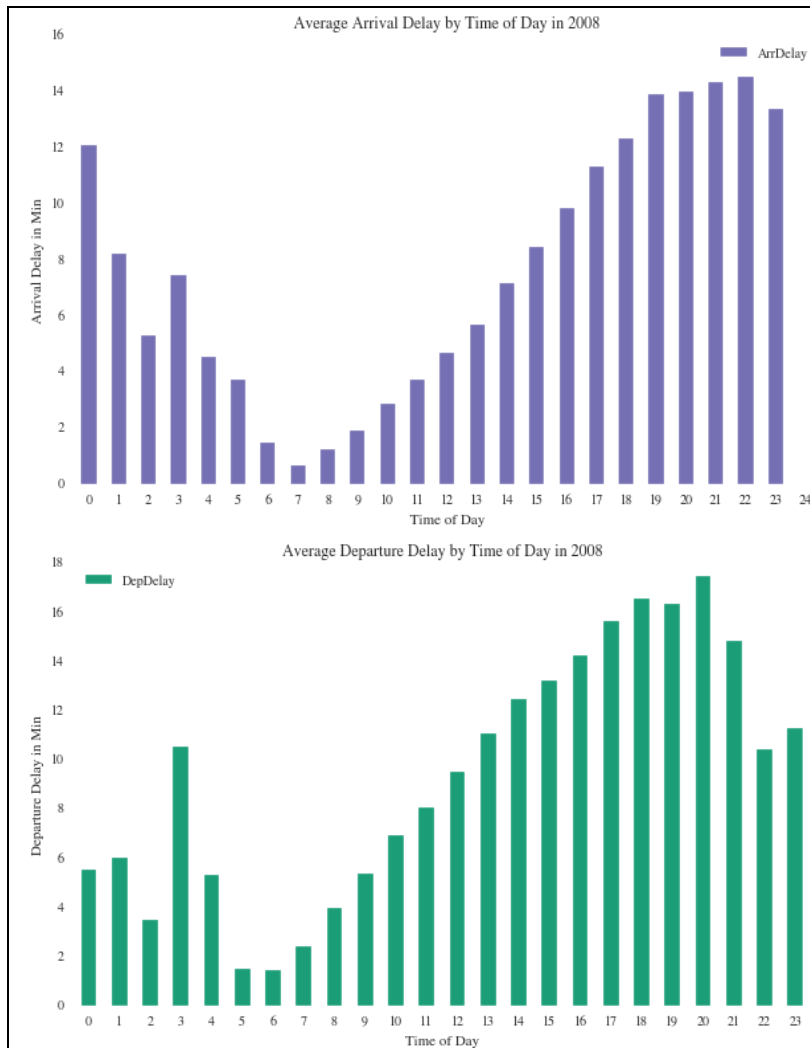




### Delays by Month

For both departures and arrivals, the impact of December is clear - the highest delays are in that month. On the other hand, September, October and November are the months with the least amount of delay. For the summer, June and July are marked by higher delays. Also, February posts high delay values as well. The reason for winter's high delay values is probably because of snowstorms in the northeast of the US. Also, in summer, thunderstorms in Dallas Forth Worth (DFW) and Chicago areas can cause high delay impact to the rest of country. A snowstorm/storm may only affect operations at an airport or two. However, delay propagation, which marks as the major contributor for flight delay, can cause ripple effects on delay to downstream flight operations.

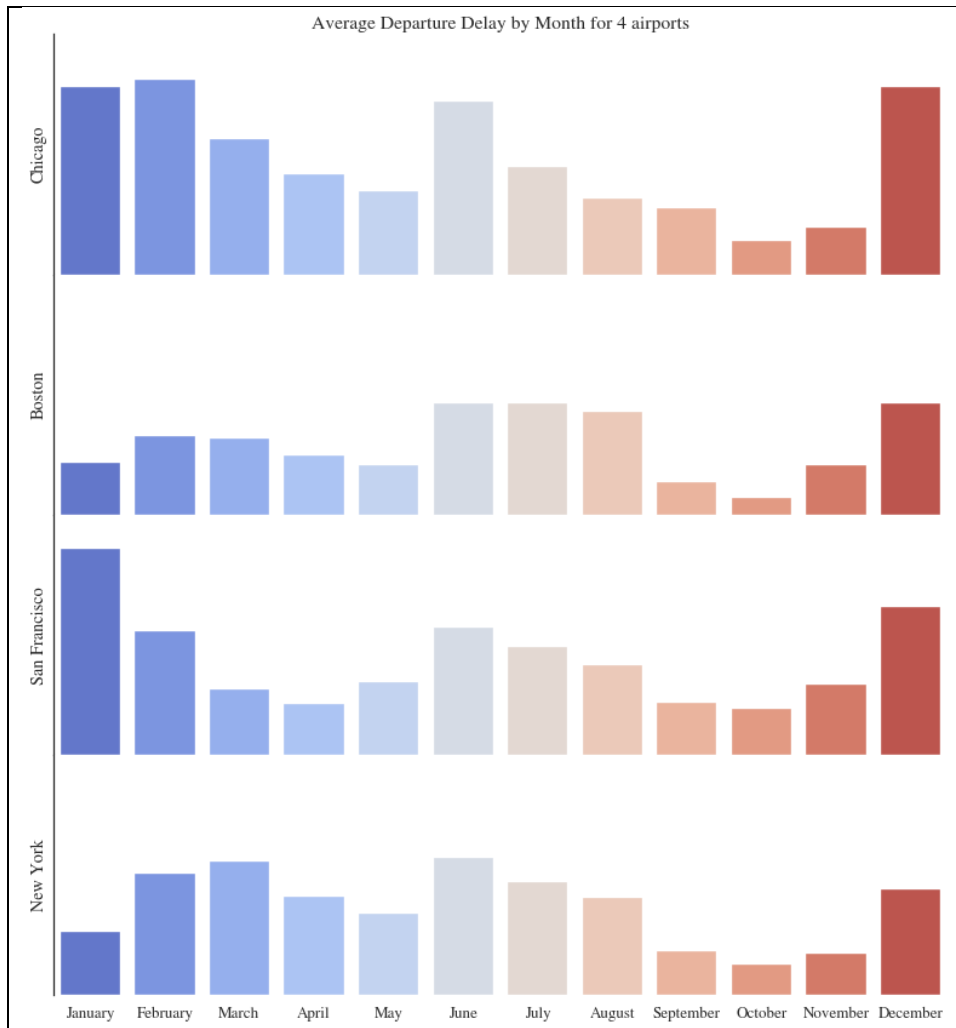
**Next**, we also think that the time of day should have an impact. Normally, flight delays cumulate throughout the day through a knock-on effect, where delayed flights provoke other delays because of tight schedules and runway congestion. We plot the mean delay by hour of day in a column chart.



## Delays by Hour

We see a marked "V" shaped decline in delay with the lowest delays in early morning hours. Both departure and arrival delays accumulate from earlier morning reaching their peaks in the evening hours. For departure, the highest mean delay is during prime-time of 18:00 to 21:00, and for arrivals, it is slightly later (the average flight duration is a few hours) and peaks at around 22:00. The increasing of flight delay by the hours of the day is mainly caused by flight delay propagation. Although a flight is built with scheduled buffer time for unforeseeable flight delay during the flight operations, it is not sufficient to cover all types of delay. As a result, if a flight is delayed, the next flight has to wait for the late arrival flight to be ready before it can be operated. Hence, flight delays for both departure and arrival flights do increase over time.

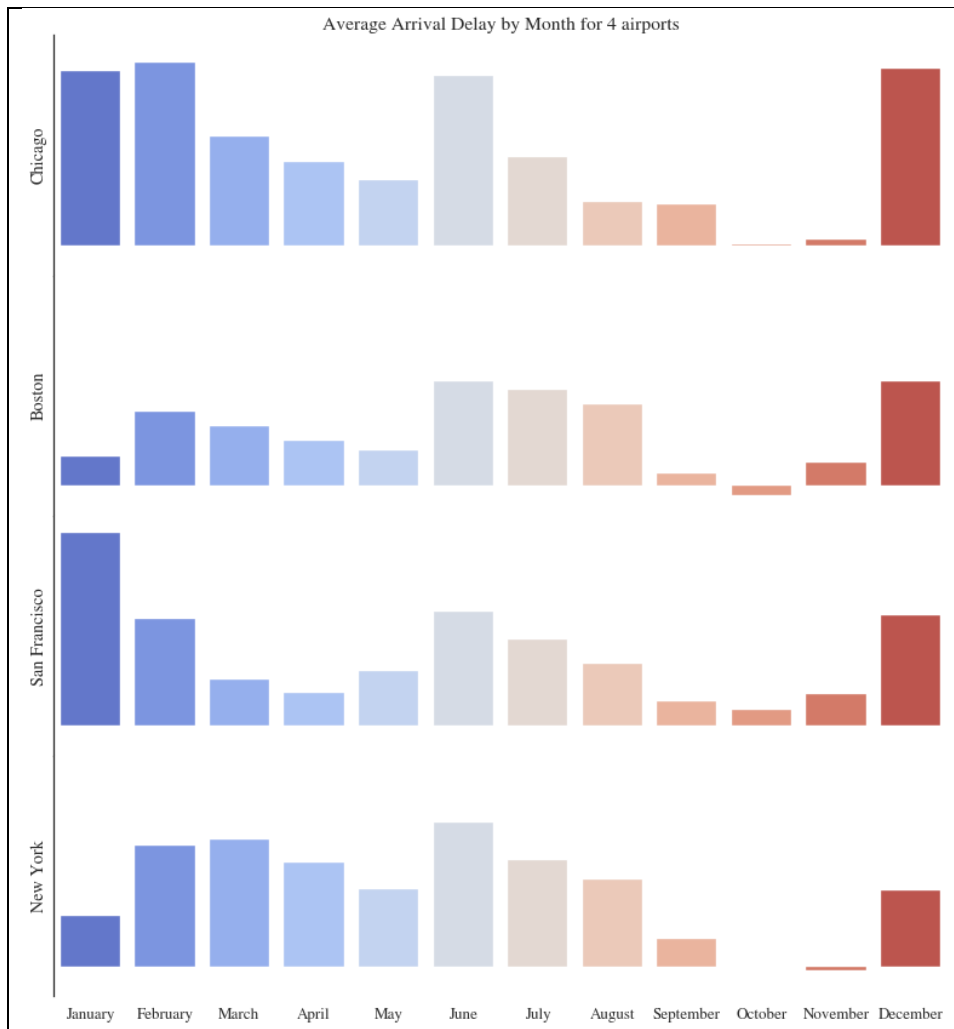
**The variation in mean delay by hour that we see implies that Hour of Day should be a good predictor of flight delay.**



### Delays by Month for select Airports

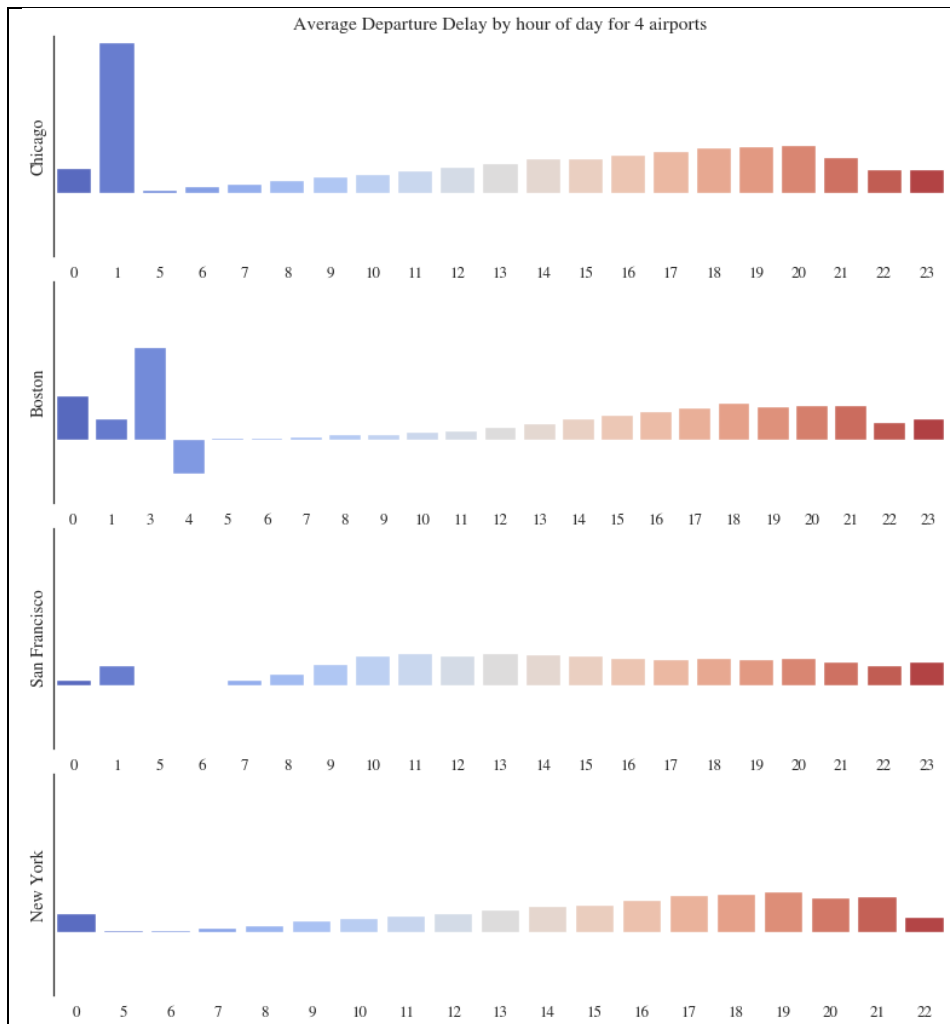
We see differences between airports - Chicago O'Hare and San Francisco are similar to the overall profile for mean delay at all airports, with higher delays in December and January and a midsummer bump. On the other hand, Boston Logan shows lower mean delays in the beginning of the year and more delays in all 3 summer holiday months. New York LaGuardia has more delay in the springtime, with February, March and June with a higher mean delay than in December. In all locations, December is a month with higher than normal delays.

Given that Northeast Corridor do receives significant amount of rain/snow, further analysis is required for examining the cause and distributions (temporal and spatial relationship) of the flight departure delay.



For arrival delays, we see four distinct peak months for Chicago O'Hare. December, January, February and June. The latter may be because of holiday travel. The former three may be entirely weather related since these three months have the worst climatic conditions. San Francisco is similar to Chicago again. The peak delay is in January possibly weather related (bad winter conditions in January 2008). Again the springtime peak in New York shows the longest delays from February to June. In all locations, the end of year only has minimal delays in October and November, with delays rising back up in December.

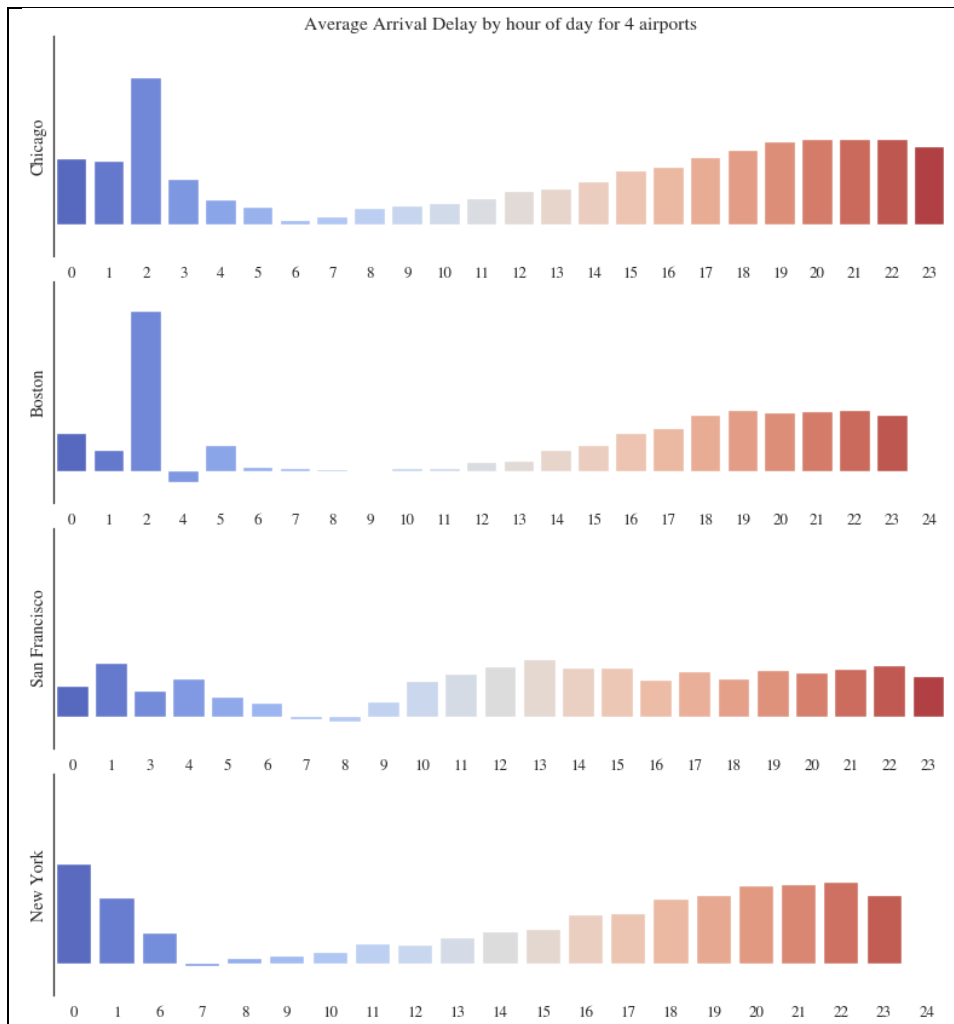
Next we would like to consider the impact of Hour of Day across the airports. We repeat the analysis for departure and arrival delay by hour.



### Delays by Hour for select Airports

It is apparent that the early morning (late night) hours are linked with longer departure delays for Chicago and Boston. Midnight and 1:00 am have large delays at ORD, and for BOS, the delay gets worse until 3:00 am and then reverses.

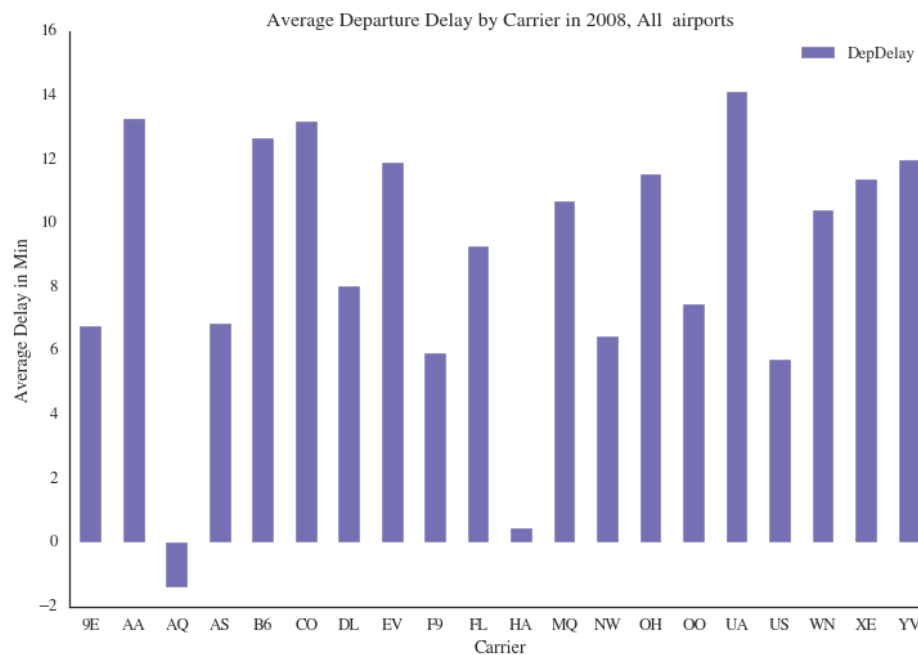
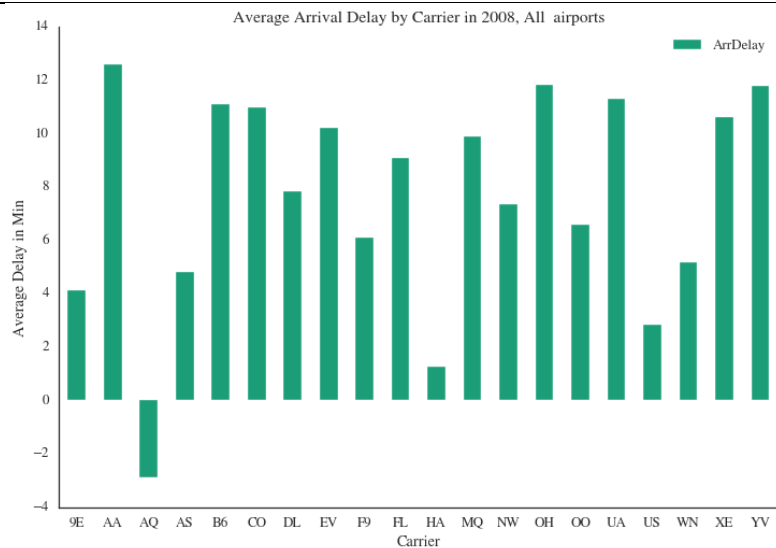
Generally, in all four airports, we see the delays accumulating throughout the day until 19:00 or 20:00 and then dropping off. This profile is most apparent for Chicago and New York. On the other hand, in San Francisco, the delays start mid-morning and remain stable throughout the afternoon.



The early morning hours have peak delays again for Chicago and Boston. In both cases, this occurs at 2:00am, with delayed flights arriving beyond schedule. Generally, as the day wears on, arrival delays increase up until 22:00. It is unclear whether schedule constraints cause the early morning delays (less staff in the early morning) since delays drop off before peaking again. Only San Francisco has a distinct profile with more stable delays throughout the day, with a morning dip from 5:00 am to 9:00am.

It is obviously that the departure and arrival delay profiles for an airport follow similar shapes. However, the arrival delay distribution shifts to the right for around 2 hours from the departure delay. The average US scheduled flight duration, from 2008 data, is 128 minutes with an average flight distance of 726 miles. The 128 minutes flight duration explains the reason why arrival delay distribution is roughly 2 hours beyond the departure distribution.

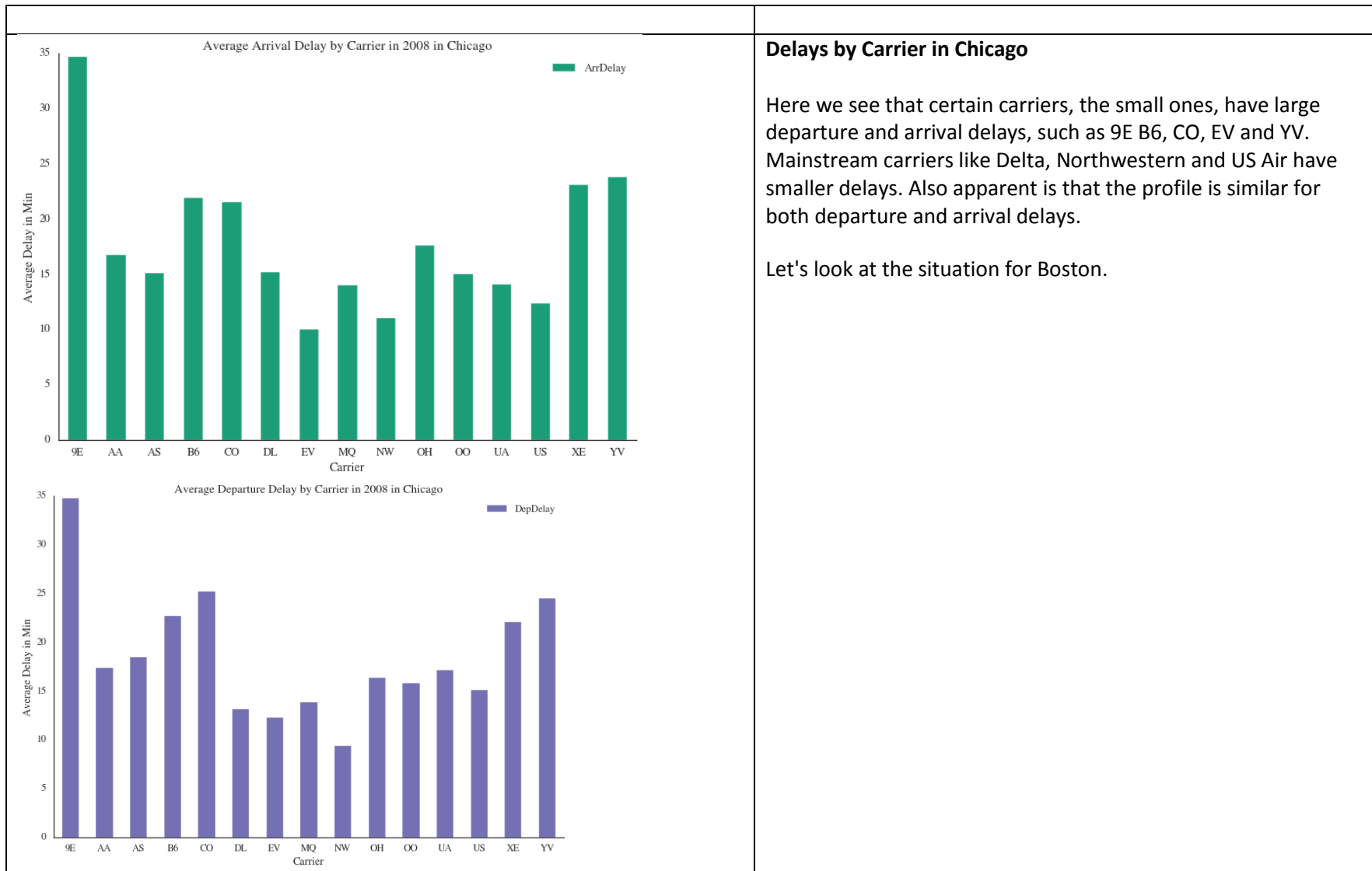
Next we look generally at delays linked to carriers, for all airports.



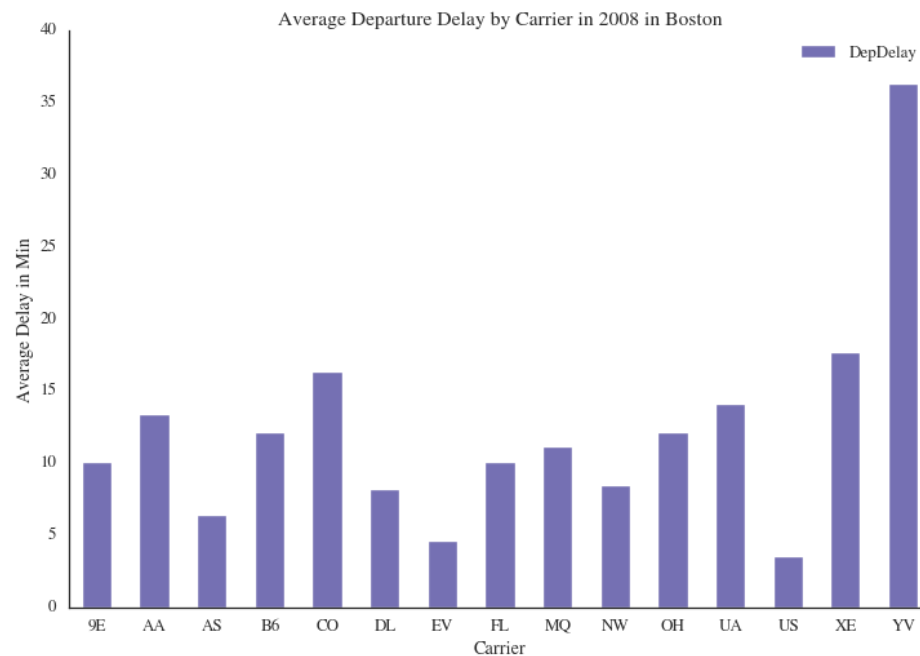
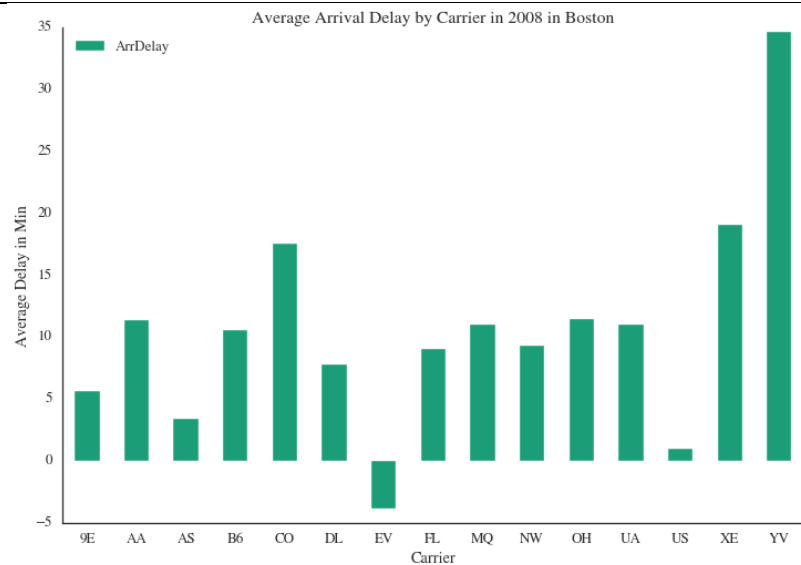
## Delays by Carrier

For flights with one of 20 unique carriers, average flight delays vary considerably. The analysis is affected by the number of flights for each carrier. Some, with a small number of flights, like Aloha Airlines (AQ) or Hawaiian Airlines (HA), have lowest mean delays. Both AQ and HA mostly operate flight between Hawaii and Continental United States. Since the flights are mostly 5-6 hours, these flights are usually exempt from Ground Delay Programs (GDPs) when an airport encounter inclement weather conditions. This means that no delay is imposed by air traffic management system. Therefore, the average delays for these two carriers can be quite low.

What if we look at just one airport - Chicago? Does carrier lead to better prediction in this case?







## Delays by Carrier in Boston

The departure and arrival delay for XE and YV is much longer. With the threshold of 15 minutes mean delay, you can expect a delayed flight if you are travelling with CO, XE or YV. In contrast, travelling with EV or US means you can expect to be on time.

However, data exploration may only reveal partial fact at different airports. However, the true causality still requires additional analysis and evaluation using domain judgement.

## Predictive Models for Flight Delays

We will be creating several machine learning models to predict flight departure delay delays using 2008 data for flight departures from Chicago O'Hare International Airport (ORD).

We will attempt to build both regression and classification models to predict flight departure delay. In this project, instead of predicting the time measure of flight departure delay, we will opt to predict if a flight is delayed or flight is not delayed.

1. A flight is on-time if the departure delay is within 15-min of the scheduled departure time (CRSDepTime).
2. A flight is delayed if the departure delay is more than 15-min late from the scheduled departure time (CRSDepTime).

For flight departure delay prediction, the following features are potential candidates for the model:

1. Month
2. Day of Week (weekday vs. weekend)
3. Departure Hours (convert from CRSDepTime)
4. Arrival Hours (convert from CRSArrTime)
5. Departure Airport
6. Arrival Airport
7. CRSElapsedTime (total time for a flight)
8. Flight Distance
9. Carrier Name

In the following analysis, we will use different models to determine if these features are major contributors to predict flight departure delay.

After this, we will add weather factors at departure and arrival such as:

1. Total Flight Demand
2. Airport Meteorological Conditions  
V stands for VMC, and  
I stands for IMC
3. Airport Temperature

## Approach:

### Iteration #1:

**Dataset:** Flight delay records originating from Chicago airport (335330 records)

We trained the following models with a random subset of 10k records and tested its performance on other subset of 10k records.

1. PCA Analysis (first experiment to see if there is a clear distinction of delays and non-delays)
2. Linear Regression – to examine the significance of features)
3. Logistic Regression Classification
4. Random Forest Classification
5. Random Forest classification performed with Cross Validation with 10 folds
6. SVM Classification

Please note that the first two experiments are not the appropriate models for classifying the given dataset, nevertheless they were run to see if we can find clues supporting the Logistic Classification and Random Forest Classification models.

### Iteration #2

**Dataset:** Weather data was combined with Flight Delays. Feature set was expanded with weather dimensions.

All the above models were repeated with the new dataset again with train-test split of 20k records

### Iteration #3

**Full Dataset** of 335,330 records was used and the classification models were run to validate the findings.

## Iteration #1 - Data Preparation

We would like to build an analysis dataset by choosing the threshold of 15 minutes, beyond which we consider the class change to "Delayed" flight. This is a standard threshold in the aviation industry, with indicators on delayed flights commonly based on 15 minutes of delay.

In order to reduce the scope of the prediction problem, we chose the following criteria:

1. Threshold of 15 minutes beyond which the classification changes to Delayed Flight
2. Restrict the flights to Chicago ORD as the origin
3. Considered Departure Delay as measured by DepDelay

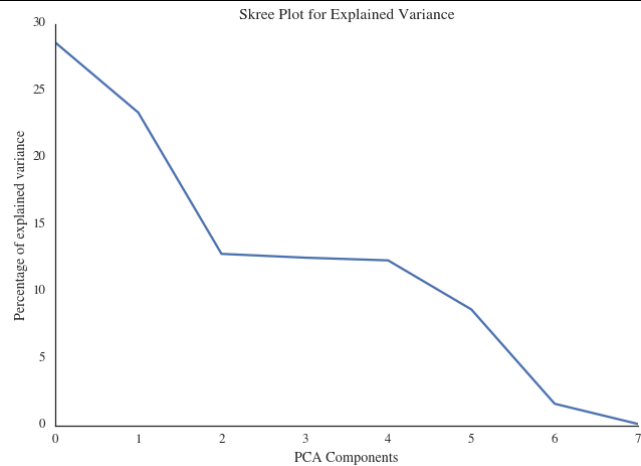
We have dropped the following features from our training and test data for simplicity sake.

- CRSArrTime, TaxiIn and TaxiOut, TailNum

We have retained Flight Distance, Destination, Carrier Airlines as predicting features. We needed to factorize the qualitative variables so that the categories were represented by numbers in order to model. We did this for the Carrier and the Airport name categories

Since the dataset is so large - we further constrained it so that Python could perform the analysis on our computers in a reasonable amount of time and without crashing. To do so, the dataset was sampled randomly for 20k rows. Further, these observations were randomly split into training and test sets, so that 10k observations were used for analysis. Also, this training set was scaled so that techniques like PCA and SVM will work. For these purposes, the SciKitLearn Standard Scaler was used. It applies the fit transform from the training set to transform the test set so that the prediction will be coherent. Also, we want to check if there are any remaining null values in the dataset before proceeding.

## Models and Analysis



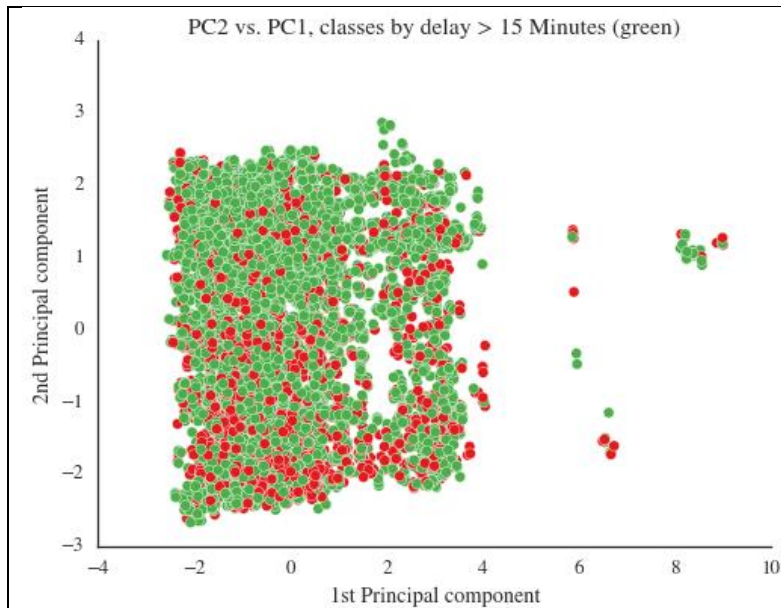
### PCA Analysis

PCA Analysis is generally performed when we have high number of dimensions; for example in face recognition. This model reduces the high dimensionality to lower dimensions space. Airline data set is not a good example to build this model for drawing conclusions. Nevertheless we have attempted PCA Analysis as a mere academic exercise.

As the first step, we want to see if data can be discriminated using linear methods. The Skree plot on the left shows the variation explained by the principal components.

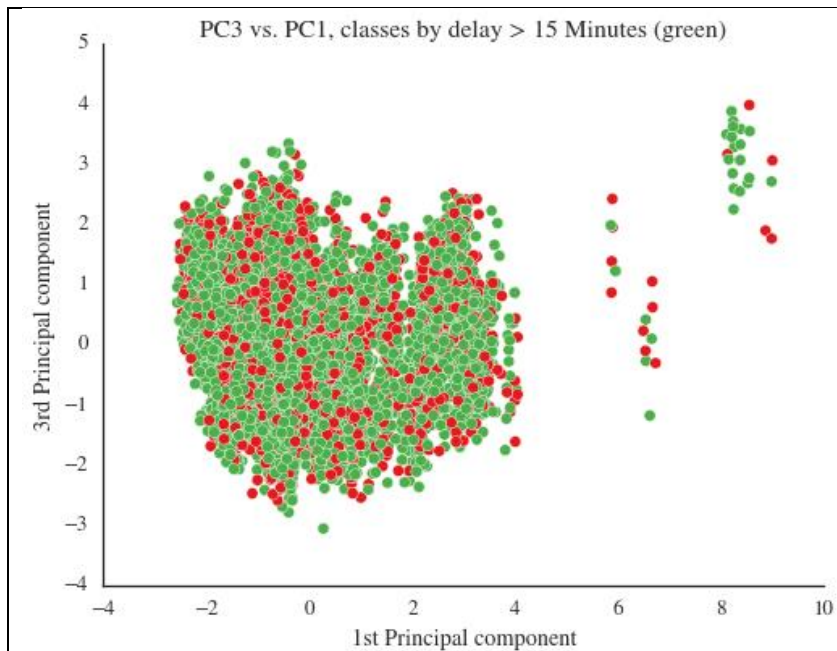
- The 1st Principal Component explains 28 % of the variance
- The 1st and 2nd Principal Components explain 51 % of the variance
- The 1st, 2nd and 3rd Principal Components explain 64 % of the variance

The first two PC explain over 50% of the variation. The blue Skree plot line shows that the Skree is basically from the second PC and is substantial to about PC 5. This means that we can use just 2 components, but that the characterization will not be complete since it only will show the about 50% of the variation. We need to go into higher orders to have a better characterization. Going to 3 PCs will improve, reaching 64% of the variance explained.



As the second step, we plot out the points according to coefficients of the first two PCs and color by the class of the points to see some class boundaries.

The scatterplot shows that green (>15 minute delay) are perhaps higher on the 2nd component, with this component separating only marginally. The first component separates two small clusters at coefficients of 6 and 8. There are only a few points in each. This is likely because PC1 is mostly flight distance and duration and these are the longest flights. (seen from eigen values)



As the third step, we plot the 3rd against 1st PC to see if there is better separation on classes.

We can clearly imagine a hyperplane separating the small clusters here. We can again see the PC1 separating the clusters at coefficients 6 and 8. The separation of red and green color has not improved any further. Clearly these PC may be used to separate the classes, but it is not clear if they do so effectively based on the arbitrary cut-off value of 15 minutes.

#### Conclusion from PCA Analysis:

Evidence from the principal components is not telling us clearly what features are causing the delay. As we stated earlier, PCA Analysis for this dataset may not be an algorithm of choice because of limited number of features.

#### OLS Regression Results

```
=====
Dep. Variable:          DepDelay    R-squared:                0.118
Model:                  OLS         Adj. R-squared:           0.118
Method:                 Least Squares  F-statistic:             929.9
Date:                  Fri, 09 Sep 2016  Prob (F-statistic):       0.00
Time:                  19:24:00      Log-Likelihood:          -5.6905e+05
No. Observations:      111611       AIC:                    1.138e+06
Df Residuals:          111594       BIC:                    1.138e+06
Df Model:               16
Covariance Type:       nonrobust
=====
```

	coef	std err	t	P> t	[95.0% Conf. Int.]
const	17.8201	0.119	150.215	0.000	17.588 18.053
Month	-3.1992	0.124	-25.762	0.000	-3.443 -2.956
DayOfWeek	-1.3933	0.120	-11.616	0.000	-1.628 -1.158
DepHr	8.4637	0.211	40.112	0.000	8.050 8.877

#### Linear Regression Analysis

Next we performed Least Squares Linear regression model using two methods:

1. Stats Model OLS class
2. Scikit Learn Linear Regression class

To the left, you see StatsModel OLS results.

In this model, it looks like Destination has the least

ArrHr	3.3556	0.207	16.180	0.000	2.949	3.762	explanatory power, followed by the DayofWeek and the UniqueCarrier.
UniqueCarrier	-0.1200	0.128	-0.939	0.348	-0.370	0.130	
Dest	-0.4237	0.155	-2.741	0.006	-0.727	-0.121	
CRSElapsedTime	8.9988	0.980	9.187	0.000	7.079	10.919	
Distance	-7.7868	0.983	-7.924	0.000	-9.713	-5.861	
Origin_Demand	-6.5798	0.139	-47.273	0.000	-6.853	-6.307	
MC_DEP	1.8091	0.171	10.583	0.000	1.474	2.144	
VISIBLE_DEP	-4.5980	0.174	-26.436	0.000	-4.939	-4.257	
TEMP_DEP	-0.3291	0.185	-1.776	0.076	-0.692	0.034	
Dest_Demand	-0.2110	0.162	-1.306	0.191	-0.528	0.106	
MC_ARR	0.8902	0.149	5.958	0.000	0.597	1.183	
VISIBLE_ARR	-1.9260	0.148	-12.973	0.000	-2.217	-1.635	
TEMP_ARR	1.4741	0.189	7.783	0.000	1.103	1.845	
=====							
Omnibus:	92775.291	Durbin-Watson:	2.001				
Prob(Omnibus):	0.000	Jarque-Bera (JB):	4452788.851				
Skew:	3.720	Prob(JB):	0.00				
Kurtosis:	33.035	Cond. No.	19.2				
=====							

Estimated intercept coefficient: [ 16.9949]		
Adjusted R^2 of the regression: 0.0602483862767		

	Features	Estimated Coefficients
1	Month	-4.010132
2	DayOfWeek	-0.962251
3	DepHr	7.396517
4	ArrHr	1.437795
5	UniqueCarrier	-1.086182
6	Dest	-0.910527
7	CRSElapsedTime	15.193715
8	Distance	-12.568533

</

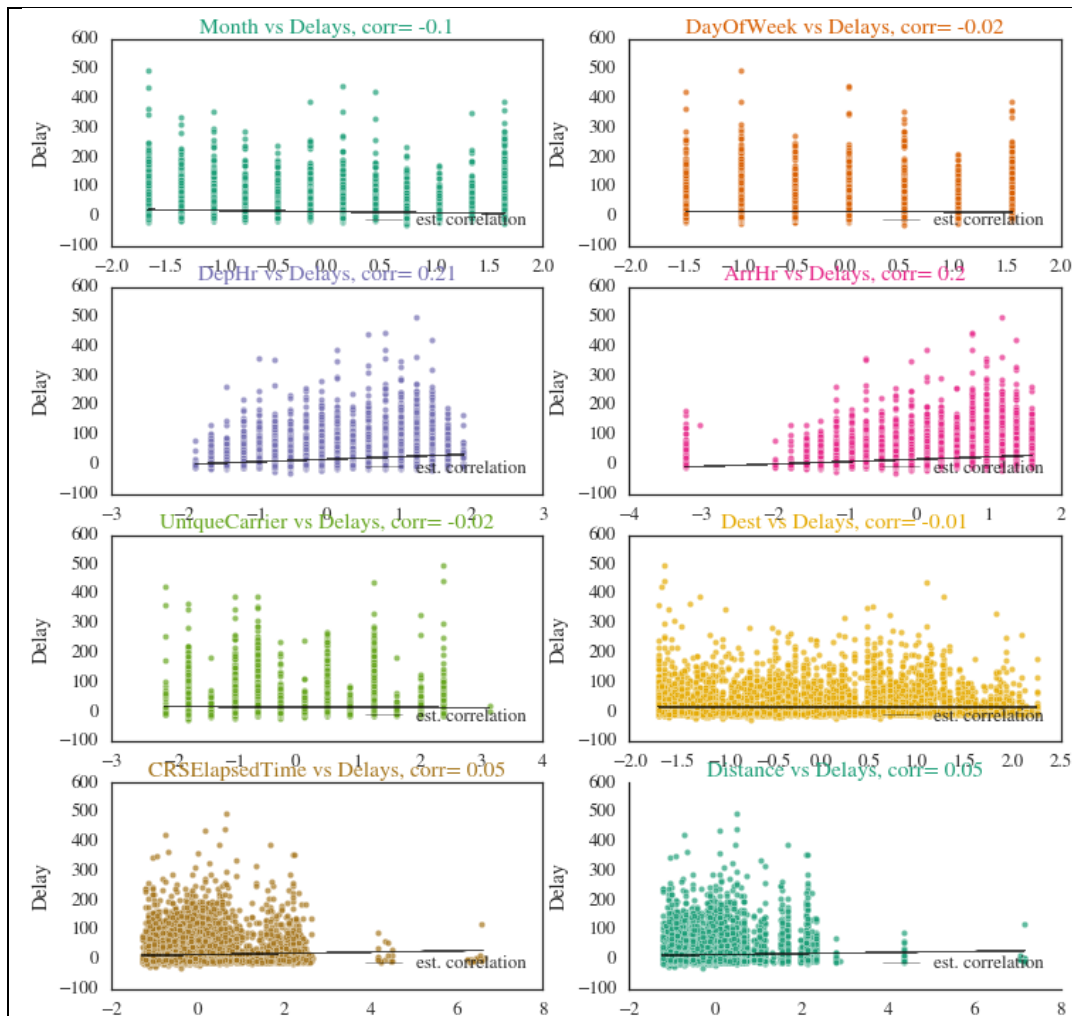


Estimated intercept coefficient: [ 16.9949]  
Adjusted R^2 of the regression: 0.0584066725982

	Features	Estimated Coefficients
1	Month	-3.996851
2	DepHr	7.383429
3	ArrHr	1.409180
4	CRSElapsedTime	15.628053
5	Distance	-13.519708

Since we found **DayofWeek**, **Destination** and **Carrier** having no significant impact, we tried re-running the model dropping the above features

The adjusted R2 decreased, showing that the alternative model also does not explain as much variation.



## Correlations:

Here we look a little more closely at the attributes themselves and how they are related to the classes. We will use scatterlots and plot each variable against delay. Then, since we have a large number of attributes, the best would be a parallel coordinates plot of the attributes, coloring by class.

## Results:

The scatterplots does not show a clear correlation. However they need close reading.

- The highest correlation coefficient is for **ArrivalHour**, because of the time of day effect that was noted earlier. This is compounded by the early morning delays - which cause a spike at the lower end in the scatterplot.
- Carrier** and **Month** show some correlation which confirms what we say in the column plots above on for the full dataset.
- The bottom two plots reveal the seperated clumps that we say above in the PC scatterplots. It is clear with these scatterplots that PC1 is capturing the **Duration** and **Distance** characteristics mainly. These may be correlated to delay, but in fact negatively - because with longer distance there is less delay since the flights are prioritized for take-off and the longer duration allow them to make up tardy departures. So we see that long-haul are less likely to be delayed.

## Classification Models

Next, we study Classification models.

The first model is a logistic regression model with L2 penalty function. We fit the training model that we randomly split 50-50 above. It is a fairly 'vanilla' classification that we shall use as a benchmark.

Given the OLS estimation above, we are not sure that the linear regression model is the best choice for our problem but it is good to evaluate against in terms of accuracy.

**Results:** We ran the logistic classification model with our training set and tested its performance on the test set. Below are the results of the confusion matrix metrics.

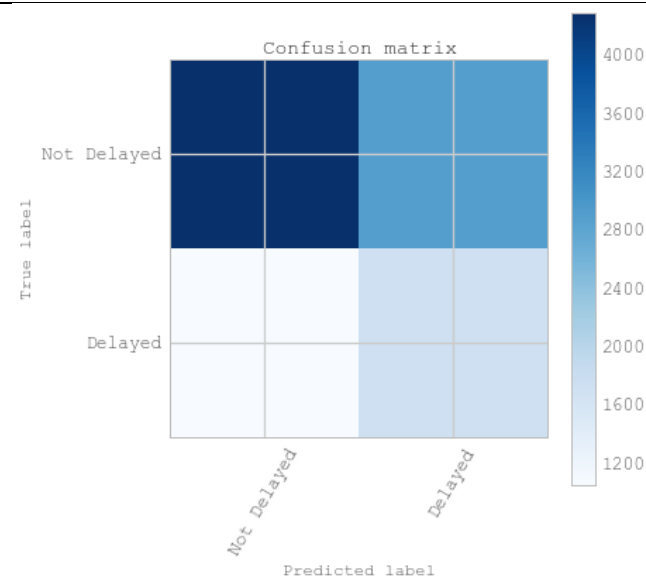
The accuracy score of 60% is already encouraging, meaning that even this first classification model guesses right 60% of the time, better than a coin flip. We know that the positive case of a delay of more than 15 minutes is less frequent than no delay (around 2800 delayed cases to more than 7000 for no delay), so we may prefer to look at the F1 measure, which is 47% here.

Confusion matrix

	0	1
0	4303	2906
1	1060	1731

**precision = 0.37, recall = 0.62, F1 = 0.47, accuracy = 0.60**

	0	1
0	0.596893	1.041204
1	0.147038	0.620208



### **Random forest Classification**

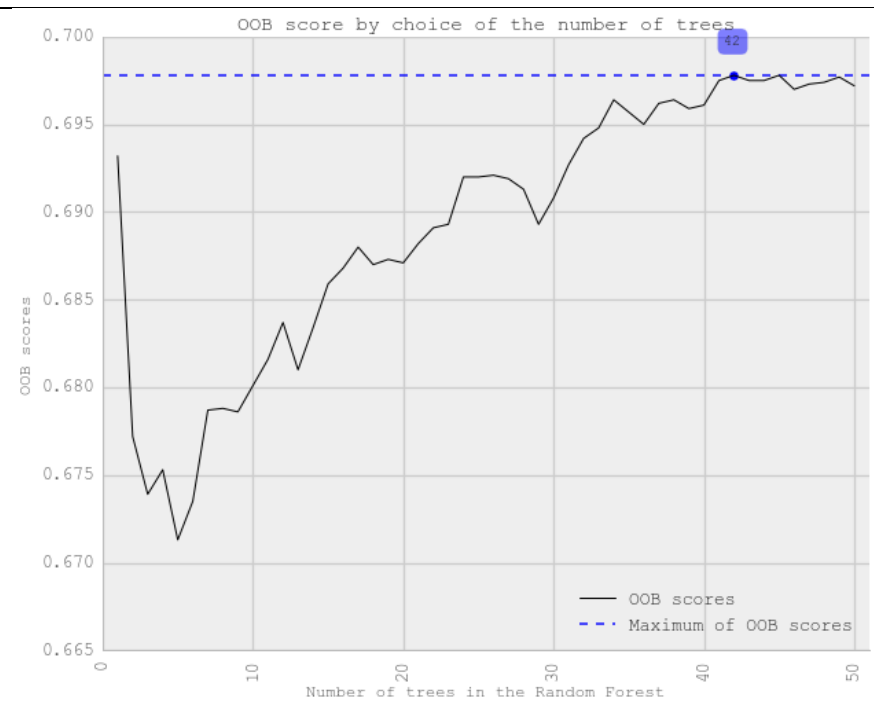
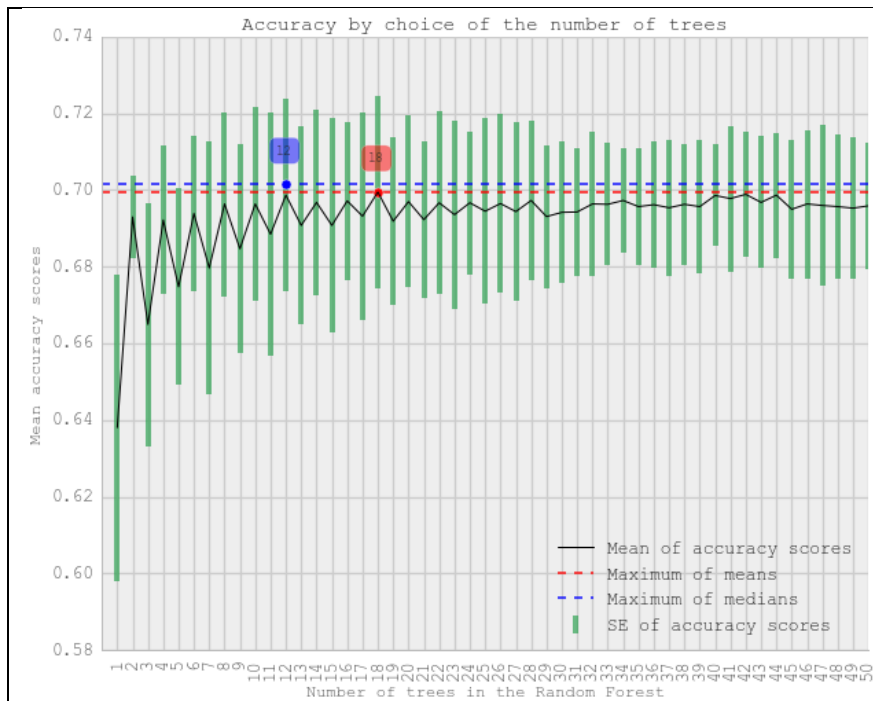
Next we performed Random Forest Model with Training Data and evaluate the model with Confusion Matrix metrics. In order to choose the optimal number of trees, we classify using 1 to 50 trees and look at the mean, median and dispersion of the resulting accuracy measures.

We have plotted Seaborn Boxplots to see Accuracy scores versus the number of trees. Based on the optimal number of 48 to 50 trees, we ran the random forest classification again and save the confusion matrix information.

So based on the optimal choice of the number of trees, we classify again, saving the confusion matrix information.

#### **Results:**

We see that the accuracy improves compared to the LR model, rising to 71%. While we are better classifying the flights that are not delayed, we are also classifying more delayed flights as not-delayed (our false positive rate went up). The F1 score goes down. The left-hand side of the matrix is darker than the right, showing that the RF classifier guesses 'not delayed' more often than delayed. We are happy with the improvement in accuracy, but wondering how to improve the precision and the F1 score.

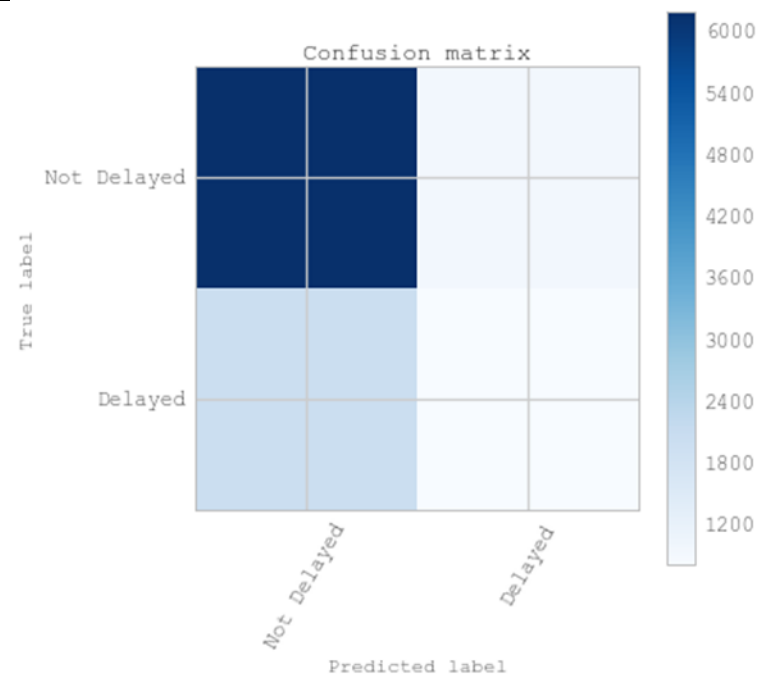


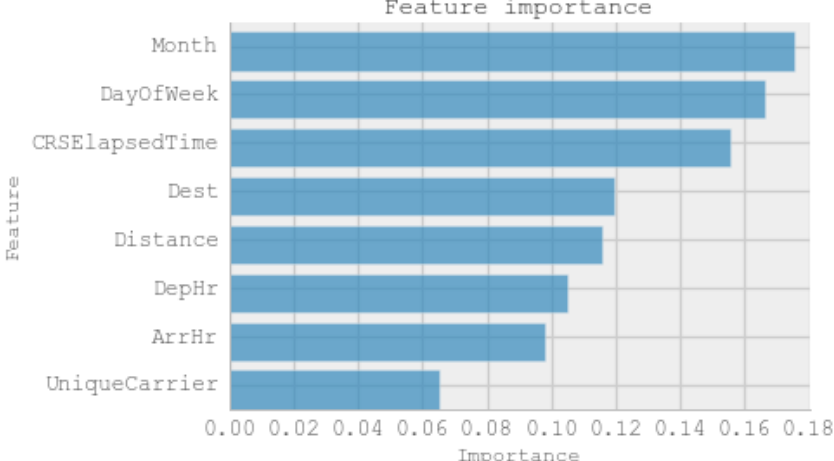
Confusion matrix

	0	1
0	6213	960
1	2010	817

precision = 0.46, recall = 0.29, F1 = 0.35, accuracy = 0.70

	0	1
0	0.866165	0.339583
1	0.280217	0.288999



 <table border="1"> <caption>Feature Importance Data</caption> <thead> <tr> <th>Feature</th> <th>Importance</th> </tr> </thead> <tbody> <tr> <td>Month</td> <td>0.17</td> </tr> <tr> <td>DayOfWeek</td> <td>0.16</td> </tr> <tr> <td>CRSElapsedTime</td> <td>0.15</td> </tr> <tr> <td>Dest</td> <td>0.11</td> </tr> <tr> <td>Distance</td> <td>0.10</td> </tr> <tr> <td>DepHr</td> <td>0.09</td> </tr> <tr> <td>ArrHr</td> <td>0.08</td> </tr> <tr> <td>UniqueCarrier</td> <td>0.06</td> </tr> </tbody> </table>	Feature	Importance	Month	0.17	DayOfWeek	0.16	CRSElapsedTime	0.15	Dest	0.11	Distance	0.10	DepHr	0.09	ArrHr	0.08	UniqueCarrier	0.06	<p><b>Next we get the Feature Importance measures from the Random Forest Model.</b></p> <p>From the feature importance measures, we see that <b>Month</b> and <b>Dayofthe week</b> have high importance, followed by the <b>Duration</b> and <b>Distance</b>. Even though we have shown that <b>ArrHr</b> is closely linked to delay, it is not at the top of the importance chart.</p>
Feature	Importance																		
Month	0.17																		
DayOfWeek	0.16																		
CRSElapsedTime	0.15																		
Dest	0.11																		
Distance	0.10																		
DepHr	0.09																		
ArrHr	0.08																		
UniqueCarrier	0.06																		
<pre>array([ 0.69130869, 0.71528472, 0.683    , 0.688    , 0.696    ,         0.718    , 0.684    , 0.693    , 0.68068068, 0.69369369])</pre> <pre>RF_scores.min(), RF_scores.mean(), RF_scores.max() (0.68068068068068066, 0.69429677809677803, 0.71799999999999997)</pre>	<p><b>Random Forest Classification with Cross-validation with 10 folds</b></p> <p>Next, we look at cross-validation to get an indication of the variation in accuracy scores for the choice of number of trees. We cross-validate with 10 random folds on the training data and check the mean score as well as the minimum and maximum scores. (see left column)</p> <p>We can expect a best accuracy of 71.79% with this model and choice of number of trees.</p>																		
<pre>array([ 0.71328671, 0.71328671, 0.714    , 0.714    , 0.714    ,         0.714    , 0.714    , 0.714    , 0.71371371, 0.71371371])</pre> <pre>SVC_scores.min(), SVC_scores.mean(), SVC_scores.max() (0.71328671328671334, 0.71380008540008544, 0.71399999999999997)</pre>	<p><b>SVM Classifier</b></p> <p>Next we would like to consider a SVM classifier - first taking what might be considered a 'vanilla' choice of tuning parameters and RBF kernel, where C=100 and gamma= 0.0001.</p> <p>The SVM Classifier does not seem to improve the performance by too much, with scores close to the RF classifier. The SVM requires a much longer time to compute. Given the small performance gain we try to tune the SVM parameters, concentrating on the gamma parameter.</p>																		

## Iteration #2 - Input and Process Weather Data

In addition to using flight data to predict flight departure delay, additional database is incorporated in the following section hoping for better predictability.

### ***FAA Airline Operations Performance Data***

The Aviation System Performance Metrics (ASPM) data from the FAA operations performance data (<https://aspm.faa.gov/>) website contains airport specific weather, flight demand and airport capacity information. Detailed ASPM data description can be found here ([http://aspmhelp.faa.gov/index.php/Aviation\\_Performance\\_Metrics\\_%28APM%29](http://aspmhelp.faa.gov/index.php/Aviation_Performance_Metrics_%28APM%29)).

The Aviation System Performance Metrics (ASPM) online access system provides data on flights to and from the ASPM airports (currently 77) in the United States; and all flights by the ASPM carriers, including flights by those carriers to international and domestic non-ASPM airports. All IFR traffic, and some Visual Flight Rules (VFRs) traffic for these carriers and airports is included.

The data used in this project is from 2008. It contains hourly weather and operational data for 77 US airports. Data fields are classified as below:

- Date/Period Information
- Flight identification
- Departure Information
  - Departure Gate Data
  - Taxi out Data
  - Wheels off Data
  - Departure Delay Data
- Enroute Information
- Arrival Information
  - Wheels on
  - Taxi in
  - Arrival Gate Data
  - Arrival Delay Data



## Data Wrangling – Steps

1. We exclude many variables from the dataset that we don't need. The Taxi in/out variables and all the delay variables are dropped.
2. We are interested in the weather variables so we decide to select only them from the dataset.
3. The new dataset contains variables for the scheduled operation at the airport - more or less and indicator of the demand per hour in the airport. This indicator is broken down into departures operations (flights per hour) and arrivals operations - both would count in determining the demand in any given hour at the airport, so we combine them into a sum of airport 'demand'.
4. We check the resulting dataset to see if we have the right columns and then duplicate the dataset since we need information from both the departure and the arrival airports.
5. We check the types of resulting dataset. Some of the variables are numeric values stored as text. We convert them to numeric in order to use in the analysis.
6. To quickly investigate the variation in the dataset, we used the 'describe' method and checked the quartiles and ranges. We also saw that we had almost 5 million records.
7. Then we converted the objects to numeric for visibility, ceiling and temperature.

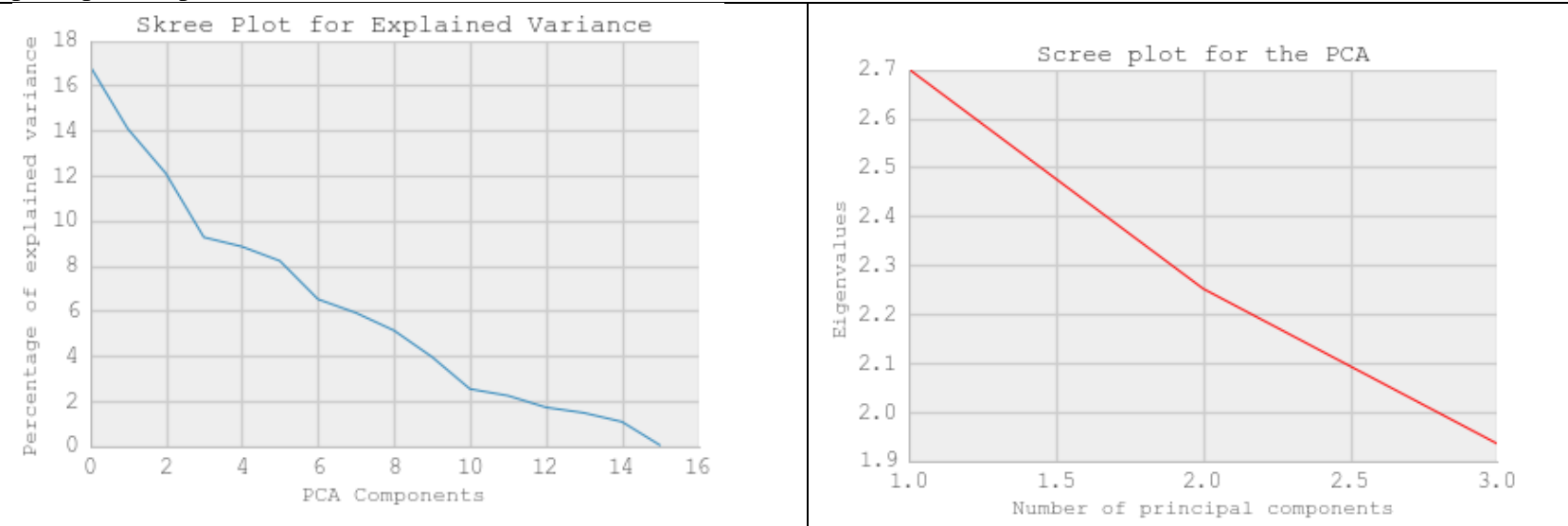
### Impact of weather on Flight delays

The whole process is repeated as we did in the case of pure airline delays without merging weather data. We repeat the selection of the random sample of the full dataset, in order to conduct the exploratory analysis. This time, we focus on the new weather variables in particular. The question is: do we expect to see some improvement with the new information.

Once again, we select Chicago and look at departure delays. We use the first exploration above as a point of reference. We factorized the categorical variables. Then we have the additional 'MC' variables to factorize. We used Standard Scaler to normalize the variables.

## Models and Analysis (Weather Data combined)

Again, the first step was to visualize a PCA plot to get a general impression of how the data spreads based on the most important principal components.

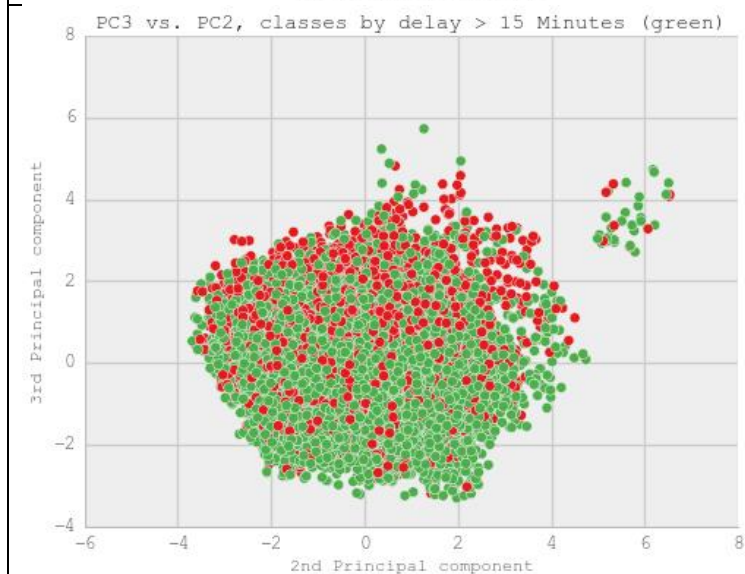
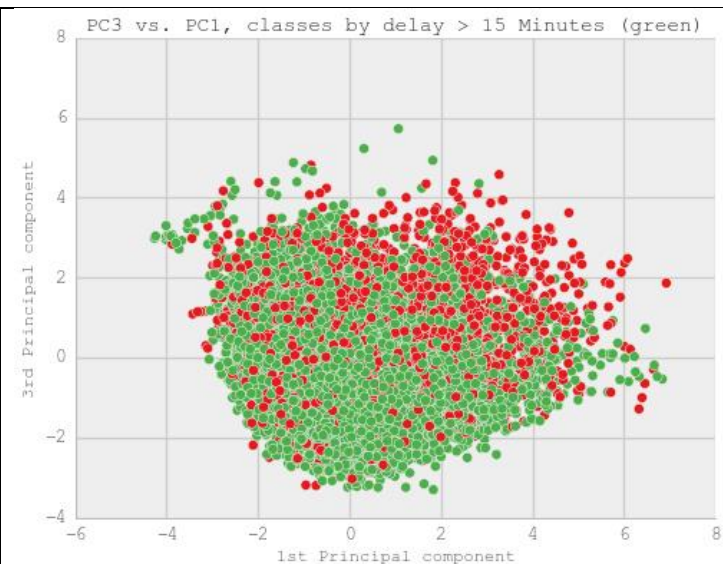
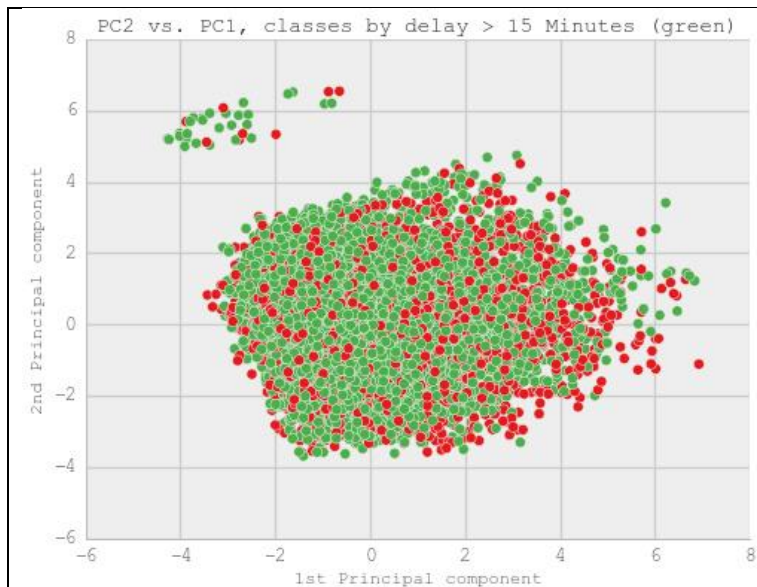


### PCA Analysis Repeated

First plot above shows that the skree has been flattened. No longer is there a bump. We need more components in order to better characterize the data. In fact, the first three do not explain much less than 50% - It appears that we need to include closer to 10 to get most of the variance explained.

- The 1st Principal Component explains 15 % of the variance
- The 1st and 2nd Principal Components explain 27 % of the variance
- The 1st, 2nd and 3rd Principal Components explain 36 % of the variance

The plot of eigenvalues against the first 3 components shows that only 36% of the variance is explained. The shape of the plot is almost linear meaning that the components each explain close to the same amount of variance - i.e. similar effects.



Study all the plots above.

It appears that the third principal component seems to separate the classes from top to bottom quite well, with the red (non delayed) dots having more positive coefficients, and the green (delayed) dots with negative coefficients.

The separation in the plots is better than before.

The separated cluster in the left is driving some of the variability - likely these are the features for distance and elapsed time that are most correlated with the second and third principal components. The dots in the clump have the highest second and third principal component coefficients.

Nonetheless, PCA Analysis doesn't give any indication significantly of the features impacting the delay.

### OLS Regression Results

Dep. Variable:	DepDelay	R-squared:	0.129
Model:	OLS	Adj. R-squared:	0.128
Method:	Least Squares	F-statistic:	92.52
Date:	Wed, 21 Sep 2016	Prob (F-statistic):	1.75e-284
Time:	22:39:40	Log-Likelihood:	-50543.
No. Observations:	10000	AIC:	1.011e+05
Df Residuals:	9983	BIC:	1.012e+05
Df Model:	16		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[95.0% Conf. Int.]
const	17.7412	0.379	46.752	0.000	16.997 18.485
Month	-3.7502	0.397	-9.448	0.000	-4.528 -2.972
DayOfWeek	-1.1542	0.384	-3.005	0.003	-1.907 -0.401
DepHr	8.6314	0.676	12.777	0.000	7.307 9.956
ArrHr	2.5271	0.664	3.805	0.000	1.225 3.829
UniqueCarrier	-0.4676	0.409	-1.145	0.252	-1.268 0.333
Dest	0.3196	0.497	0.643	0.520	-0.655 1.294
CRSElapsedTime	8.8788	3.165	2.805	0.005	2.675 15.083
Distance	-7.8536	3.175	-2.473	0.013	-14.078 -1.629
Origin_Demand	-6.3941	0.442	-14.462	0.000	-7.261 -5.527
MC_DEP	1.6574	0.550	3.013	0.003	0.579 2.736
VISIBLE_DEP	-4.8070	0.563	-8.533	0.000	-5.911 -3.703
TEMP_DEP	-0.9153	0.594	-1.542	0.123	-2.079 0.249
Dest_Demand	0.2651	0.517	0.513	0.608	-0.748 1.278
MC_ARR	0.7750	0.477	1.623	0.105	-0.161 1.711
VISIBLE_ARR	-3.0989	0.474	-6.537	0.000	-4.028 -2.170
TEMP_ARR	2.2347	0.606	3.686	0.000	1.046 3.423

Omnibus:	6185.618	Durbin-Watson:	1.958
Prob(Omnibus):	0.000	Jarque-Bera (JB):	70193.087
Skew:	2.845	Prob(JB):	0.00
Kurtosis:	14.666	Cond. No.	19.4

2088.6265910949801

### OLS Regression (repeated) with StatsModel OLS algorithm

With the addition of weather data, we have doubled the number of explanatory variables in the model.

The adjusted  $R^2$  shows a marked improvement, rising to almost 13%. On the other hand, the MSE of the prediction on the test set rises to 2088. The model has several coefficients that are not significantly different from zero - notably Carrier, Destination, Departure Airport Temperature, Demand at Destination, Arrival hour

Estimated intercept coefficient: [ 17.7412]  
Adjusted R^2 of the regression: 0.129131402697

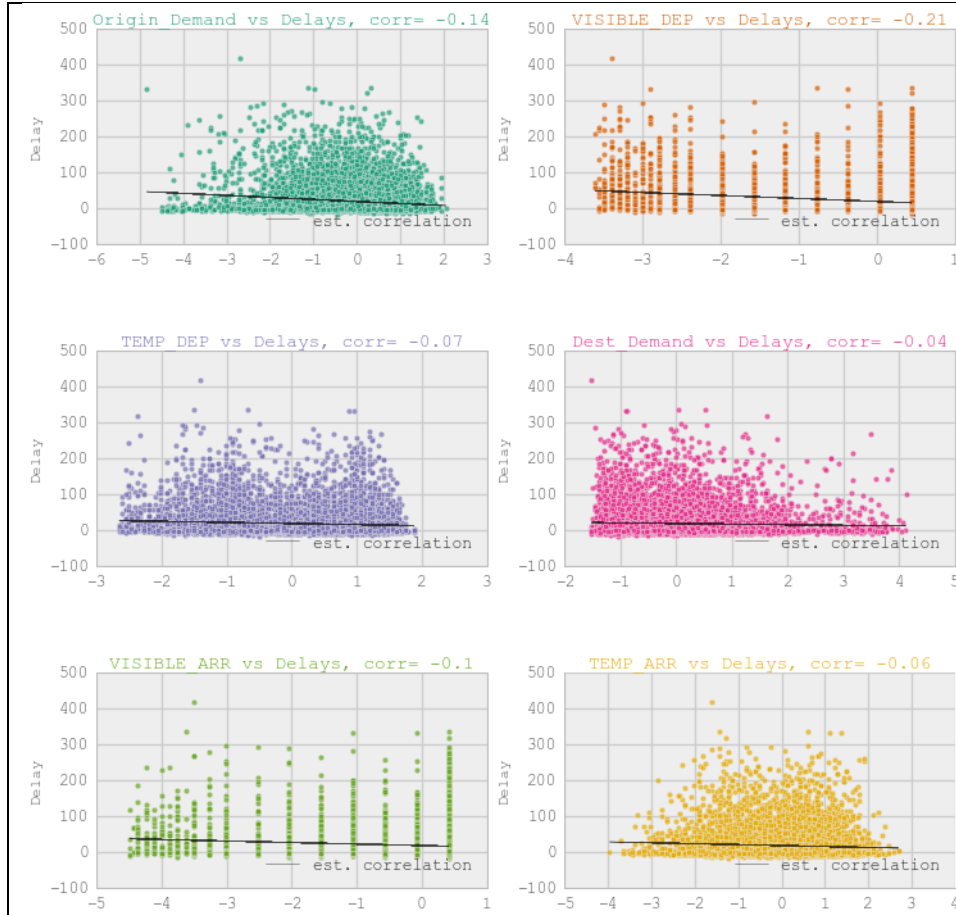
Features	Estimated Coefficients
Month	-3.750163
DayOfWeek	-1.154214
DepHr	8.631354
ArrHr	2.527110
UniqueCarrier	-0.467587
Dest	0.319591
CRSElapsedTime	8.878835
Distance	-7.853562
Origin_Demand	-6.394145
MC_DEP	1.657399
VISIBLE_DEP	-4.807043
TEMP_DEP	-0.915289
Dest_Demand	0.265107
MC_ARR	0.774963
VISIBLE_ARR	-3.098913
TEMP_ARR	2.234718

### Linear Regression Algorithm performed with Scikit Learn (Repeated)

We see also that there is a negative correlation between demand at the origin, visibility at departure and origin airport, distance and delay. When the demand is higher, delay is lower. This is perhaps counter-intuitive.

We see that ElapsedTime and Distance, despite having the most non-zero coefficients, are not the ones that would be kept in the feature selection - but rather DepHr, ArrHr, ETMS\_DEP, MC\_DEP and VISIBLE\_DEP.

We confirm this by looking at the scatterplots and correlations.



With the scatterplots of the new variables, we see that delay is correlated negatively. One explanation is that the more **demand destination airports** are those for which flights are prioritized.

Also **Visibility** is negatively correlated - which makes sense, since as visibility worsens, more flights are typically delayed.

As expected, **temperature at both the departure and arrival airports** is negatively correlated with departure delays. A parallel coordinate plot shows more relationships with attributes.

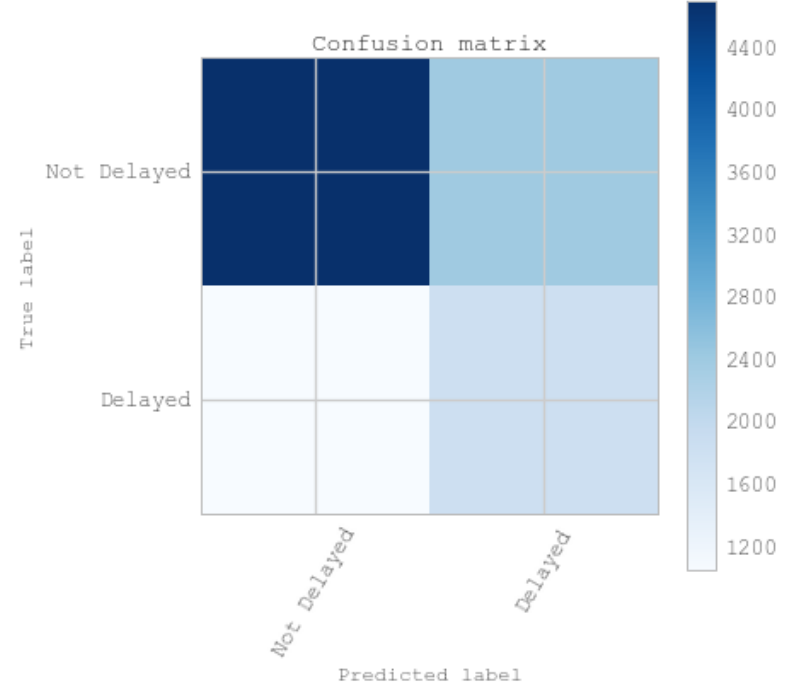
## Logistic regression Model

Confusion matrix

	0	1
0	4713	2398
1	1054	1835

precision = 0.43, recall = 0.64, F1 = 0.52, accuracy = 0.65

	0	1
0	0.662776	0.830045
1	0.148221	0.635168



Logistic regression model (Repeated with Weather Data)

The accuracy for the LR classification, has improved from 60% to 65%. The F1 score is 52% leading to conclude that the additional features help to obtain better prediction performance.

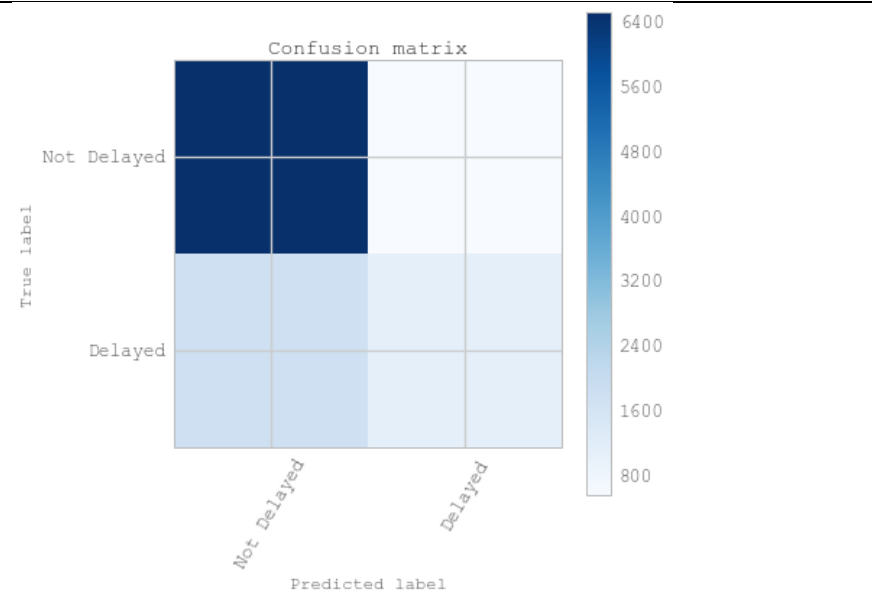
## Random Forest Classification

Confusion matrix

	0	1
0	6546	565
1	1807	1082

precision = 0.66, recall = 0.37, F1 = 0.48, accuracy = 0.76

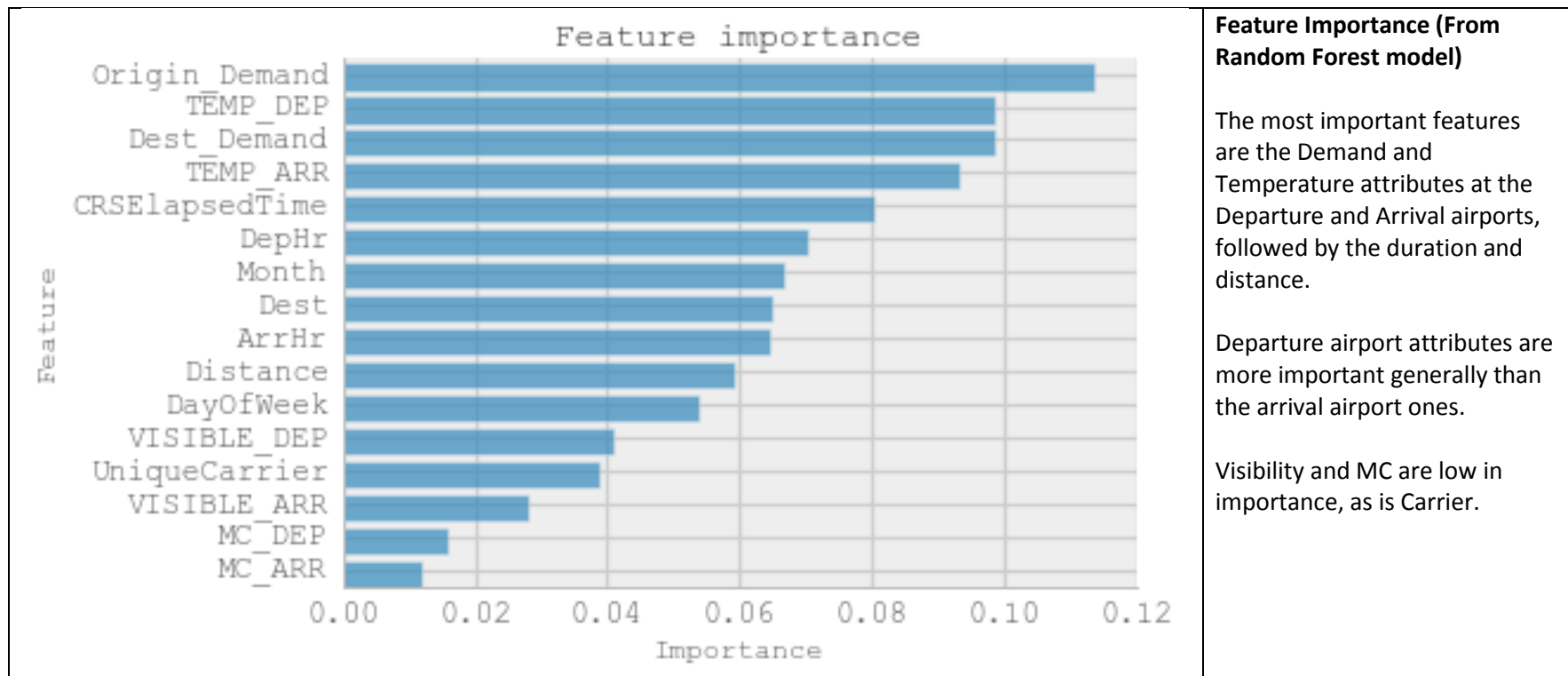
	0	1
0	0.920546	0.195569
1	0.254113	0.374524



## Random Forest Algorithm Analysis

Accuracy now has improved to 76%, which is a good gain. Precision has also improved to 66% from 53% before and the F1 has also jumped to 48%. The new model seems to perform better.





<pre>RF_scores.min(), RF_scores.mean(), RF_scores.max() 0.74025974026 0.756101551102 0.771</pre>	<p><b>Cross-Validation Scores</b></p> <p>Let's look at cross-validation scores, using 10-fold CV and 50 trees. The model seems robust, and the previous score of 77% is among the highest scores</p>
<pre>SVC_scores.min(), SVC_scores.mean(), SVC_scores.max() 0.716 0.730501039401 0.74574574574</pre>	<p><b>SVM Classification</b></p> <p>The scores from SVM are lower. This could be from the choice of</p>

tuning parameters, but it does not seem likely that more tuning would bring us above 77%.

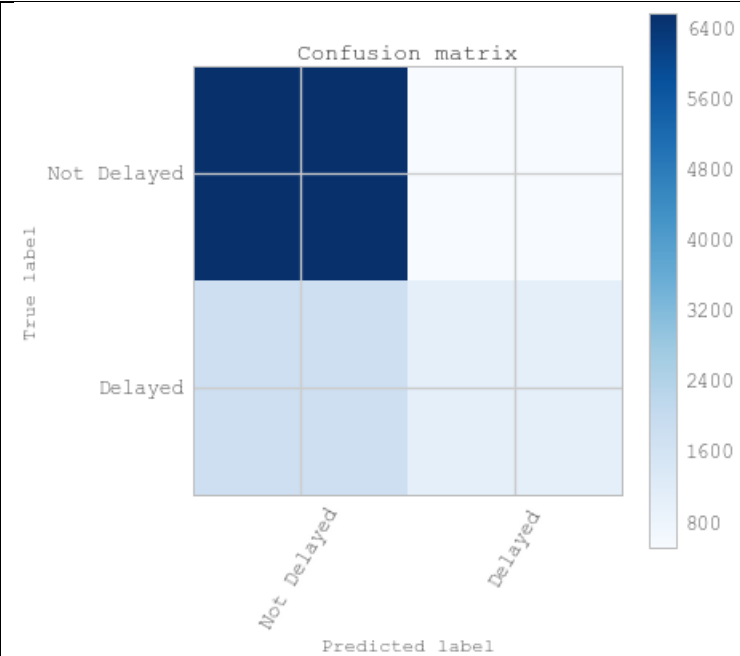
Let's try improving accuracy with dummy encoding for the categorical variables.

Confusion matrix

	0	1
0	6590	521
1	1817	1072

precision = 0.67, recall = 0.37, F1 = 0.48, accuracy = 0.77

	0	1
0	0.926733	0.180339
1	0.255520	0.371063



### With Onehotencoder

We increase precision marginally, but accuracy stays the same. It does not seem to be that important to dummy encode the features.

### Iteration #3 – Running models with full dataset

Finally, as a last step, we go back to the full dataset and perform classification using the RF with 50 trees and the new weather variables. We need to split into train and test datasets randomly and then scale them (actually we don't really need to scale for RF, but we do it anyway).

```
RF_scores.min(), RF_scores.mean(), RF_scores.max()
0.76713556133 0.768902711934 0.772242630589
```

#### Cross validation scores

The cross validation scores look as good as with the random sampled subset, with close to 77% accuracy.

Next we use the training set and then the test data to see how well the model performs.

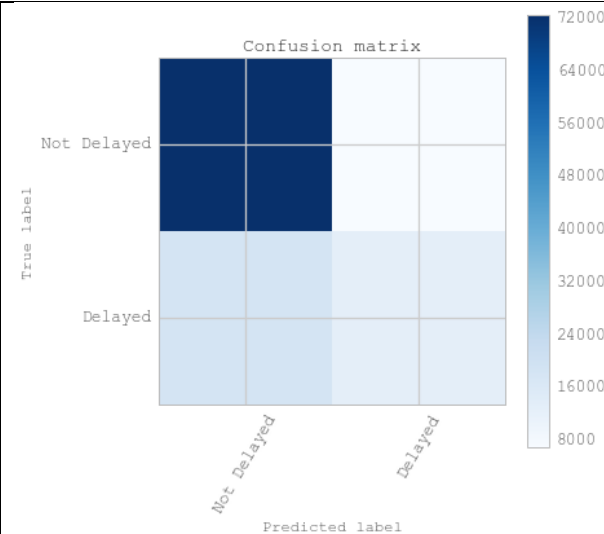
#### Random Forest Classification Scores

Confusion matrix

	0	1
0	72650	6930
1	18685	13347

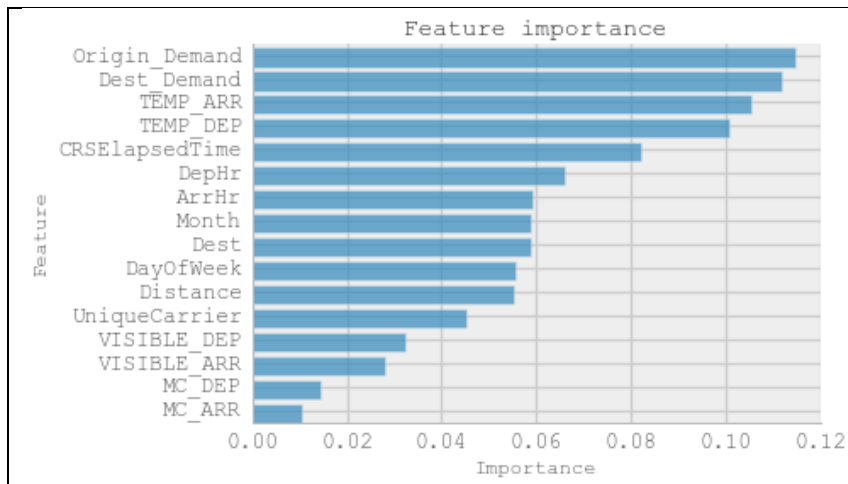
precision = 0.66, recall = 0.42, F1 = 0.51, accuracy = 0.77

	0	1
0	0.912918	0.216346
1	0.234795	0.416677



We get precision at 77%, and the F1 score is above 50%. Precision is at 66%. The selected model is good at predicting departure delays of more than 15 minutes.

Let's look at the feature importance ranking.



### Feature Importance (Full Dataset)

On the full dataset, origin and destination demand are the most important features, followed by temperature at the destination and origin.

Then duration and departure hour are the most important. Again, the MC and Visibility variables are the least important.

### Next Step

As a last step, we try dummy encoding on the full dataset to see if there is an improvement.

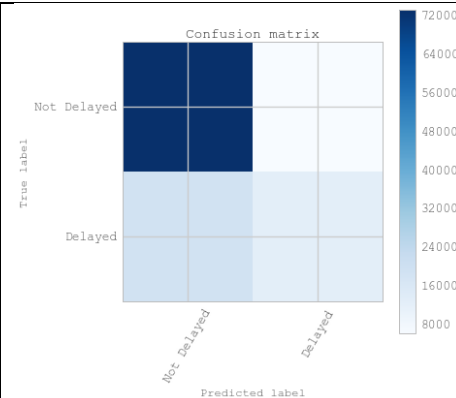
### With Onehotencoder (Full dataset)

Confusion matrix

	0	1
0	73315	6265
1	19176	12856

precision = 0.67, recall = 0.40, F1 = 0.50, accuracy = 0.77

	0	1
0	0.921274	0.195586
1	0.240965	0.401349



There is not really an improvement with the dummy encoding on the full dataset. We stick with the original factored variables, since they are easier to interpret. The analysis has shown that we can arrive at good prediction with the combination of flight origin and destination information such as time of day and the month. Adding weather and airport demand information improves the predication.

**The Random Forest classifier seems to be the best for this flight delay problem.**

## Findings

**Objective:** To predict flight departure delay. Instead of quantifying the amount of delay, we try to categorize if a flight is on-time (with a departure delay less than 15-min) or delayed (with a departure delay more than 15-min).

In the analysis, we used 2008 Chicago O'Hare International (ORD) departure traffic as an example for our flight departure delay analysis.

Several machine learning techniques are used for predicting if a flight has departure delay. We did this in three iterations as stated earlier.

First, we used 1) a regression model to examine the significance of each features and 2) a feature selection approach to examine the impact of feature combinations. These two techniques determined the features to retain in the model. Instead of using the whole set, we first sample only 10,000 records to run through different machine learning models.

Two machine learning models were used: a random forest classifier and a SVM classifier with cross validation scores using 10 folds.

Also, we applied an approach called One-Hot-Encoder to create a variant of the model for evaluating potential prediction performance as well. Finally, once we determine an appropriate model variant, we use the full ORD data set to examine the model performance.

Our result shows that the Random Forest method yields the best performance compared against the SVM model. The SVM model is very time consuming (computer intensive) but did not necessarily yield better results.

### Prediction

At the end, our model correctly predicted 91% of the non-delayed flights. However, the delayed flights predicted correctly only 41% of the time. A lot of the time, when we predict a flight that is not delayed, it is actually delayed. It could be that additional features related to the causes of flight delay are not yet discovered using our existing data sources. We can only assume that additional data, perhaps be included to improve the model:

1. Tail Number – indicating the age of the flight
2. Airport Air Traffic Control Event

### Features Importance

The following features were found to be critical in predicting delays

1. Demand – Traffic volumes at airport
2. Temperatures
3. Flight duration

For the interest of time, we only use Chicago (ORD) as an example. In addition to the ideas we proposed in the previous section to improve the accuracy of the model for delayed flights. This model can be extended to other airports easily. In addition, given the amount of data that is available, we can further create sub- models for specific airlines at an airport provided that the airlines have fair share of the operations at an airport.