Machine Learning Engineer Nano-degree
Capstone Project
Predicting arrival delays for flights in the US.

Project Overview

Airline travel is increasingly becoming a primary way of travel for people over medium and long distances. The increased air travel and congested plane slots at the airport result in sometimes significant delays. This is particularly true during the high travel season, not to mention weather related delays. Is it possible to discern a pattern in airline arrival delays and build a model that can learn and predict delays based on historical data?

In this project we attempt to forecast and attribute flight delays in the domestic US market. In general, flights may be delayed for various reasons. Flights may get diverted or canceled due adverse weather, may get delayed due to aircraft maintenance issues, or labor unrest or just plain congestion in the air or at the airports. In many instances planes may depart from their origin later than the publicized departure time but airlines usually build in a guard band in the schedule and a late departure may not pre-ordain a late arrival. Also delayed departures can be made up during the actual flight time by flying the aircraft faster (burning more fuel etc.) There are many parts in this puzzle and we will try to pinpoint the main factors that can explain the later arrivals. We will then build a model to train and see if it can accurately predict on a held out set. This is the Machine Learning part. Arguably some airlines are better at managing delays than others and we will take a look at the data to see which airlines manage delays better and which routes in particular are delay prone. This is the Data Analysis part.

Getting the data

ın this project, we will evaluate the performance and predictive power of various models that have been trained and tested on data collected by the Bureau of Transportation Statistics (BTS.) A model trained on this data that is seen as a *good fit* could then be used to make certain predictions about a flight delays.
All the data that we have taken can be found at the following web address Government website. The guide for each of the fields is as below.

Field	Description
`FlightDate`	Flight Date (yyyymmdd)
`UniqueCarrier`	Unique Carrier Code. When the same code has been used by multiple carriers, a numeric suffix is used for earlier users, for example, PA, PA(1), PA(2). Use this field for analysis across a range of years.
`FlightNum`	Flight Number

OriginAirportID	Origin Airport, Airport ID. An identification number assigned by US DOT to identify a unique airport. Use this field for airport analysis across a range of years because an airport can change its airport code and airport codes can be reused.
OriginCityMarketID	Origin Airport, City Market ID. City Market ID is an identification number assigned by US DOT to identify a city market. Use this field to consolidate airports serving the same city market.
OriginCityName	Origin Airport, City Name
DestAirportID	Destination Airport, Airport ID. An identification number assigned by US DOT to identify a unique airport. Use this field for airport analysis across a range of years because an airport can change its airport code and airport codes can be reused.
DestCityMarketID	Destination Airport, City Market ID. City Market ID is an identification number assigned by US DOT to identify a city market. Use this field to consolidate airports serving the same city market.
DestCityName	Destination Airport, City Name
DepTime	Actual Departure Time (local time: hhmm)
DepDelay	Difference in minutes between scheduled and actual departure time. Early departures show negative numbers.
ArrTime	Actual Arrival Time (local time: hhmm)
ArrDelay	Difference in minutes between scheduled and actual arrival time. Early arrivals show negative numbers.
AirTime	Flight Time, in Minutes
Flights	Number of Flights
Distance	Distance between airports (miles)
CarrierDelay	Carrier Delay, in Minutes
WeatherDelay	Weather Delay, in Minutes
NASDelay	National Air System Delay, in Minutes
SecurityDelay	Security Delay, in Minutes
LateAircraftDelay	Late Aircraft Delay, in Minutes

A quick check on the dates of the data shows that these data are for the month of January/2016. We will analyze the data in more details later to make sure it is not dominated by outliers or disproportionately scaled features.

Description of the data

```
'This is the information about the data types in our flight data
that we read in (un) pre-processed'

FL_DATE                   object
UNIQUE_CARRIER            object
FL_NUM                     int64
ORIGIN_AIRPORT_ID          int64
ORIGIN_CITY_MARKET_ID      int64
ORIGIN_CITY_NAME          object
DEST_AIRPORT_ID            int64
DEST_CITY_MARKET_ID        int64
DEST_CITY_NAME            object
DEP_TIME                 float64
DEP_DELAY                float64
ARR_TIME                 float64
ARR_DELAY                float64
AIR_TIME                 float64
FLIGHTS                  float64
DISTANCE                 float64
CARRIER_DELAY            float64
WEATHER_DELAY            float64
NAS_DELAY                float64
SECURITY_DELAY           float64
LATE_AIRCRAFT_DELAY      float64
Unnamed: 21              float64
dtype: object
```

Problem Statement

A cursory investigation about the Flight delay data shows that we need to separate the dataset into **features** and the **target variable**. The **target variable**, `'ARR_DELAY'`, will be the variable we seek to predict. All the columns other than the target variable end up as the **features**. Some of the features give us quantitative information and some give us qualitative information about each data point. These are stored in `features_set`.

Metrics

We carry out two kinds of analysis on our data. The first analysis is regression and the second analysis is classification. For the regression we evaluate the goodness of fit using the regression score built into the model. For the classification we valuate the goodness of classification using the F-Score, which is also built into the model.

Data Exploration

Next we carry out some data exploration. For detailed analysis please look @ the iPython notebook.

```
'Descriptive Statistics for Arr/Dep delays'
```

	ARR_DELAY	DEP_DELAY	DISTANCE	AIR_TIME
count	433298.000000	433298.000000	433298.000000	433298.000000
mean	1.533654	7.755997	843.752214	116.395818
std	39.096842	36.730931	609.423956	72.981191
min	-79.000000	-47.000000	31.000000	8.000000
25%	-15.000000	-5.000000	391.000000	62.000000
50%	-7.000000	-2.000000	679.000000	98.000000
75%	5.000000	5.000000	1089.000000	148.000000
max	1659.000000	1663.000000	4983.000000	698.000000

```
Flight Delay dataset has 445827 data points with 22 variables each.
```

Exploratory Visualization

We find out the longest and shortest flights.

Longest:
http://weekendblitz.com/top-7-longest-domestic-us-flights

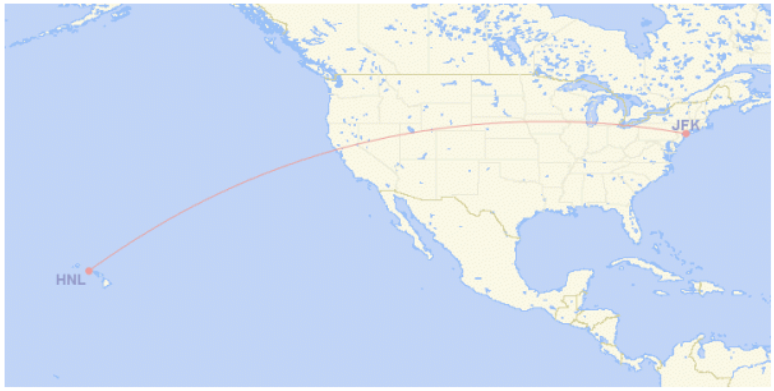
Shortest:
Google Maps (google maps doesn't show a flight exists. What does google know? ;)

```
'Display the Longest distance flight in our Database'
```

	FL_DATE	UNIQUE_CARRIER	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN_CITY_MARKET_ID	ORIGIN_CITY_NAME	DEST_AIRPORT_ID	DEST_CITY_MARKET_ID	DEST_CITY_NAME	DEP_TIME	DEP_DELAY	ARR_TIME	ARR_DELAY	AIR_TIME	FLIGHTS	DISTANCE	CARRIER_DELAY	WEATHER_DELAY	NAS_DELAY	SECURITY_DELAY	LATE_AIRCRAFT_DELAY
	1/23/16	HA	50	12173	32134	Honolulu, HI	12478	31703	New York, NY	2024	244	1125	270	551	1	4983	26	244	0	0	0

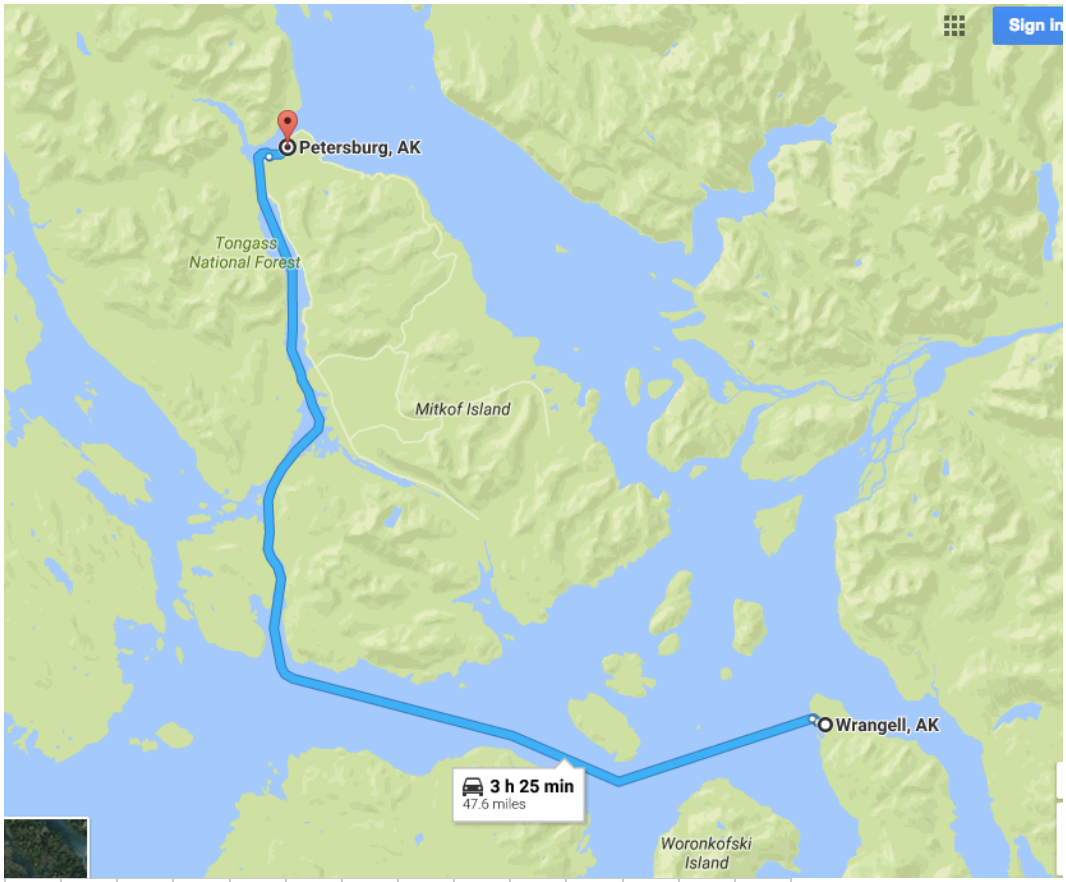
Longest non-stop flight operated in all 50 states:

Winner: New York (JFK) to Honolulu (HNL) at **4,983 miles** in 11 hours and 40 minutes. Operated by Hawaiian Airlines.



'Display the Shortest distance flight in our Database'

	FL_DATE	UNIQUE_CARRIER	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN_CITY_MARKET_ID	ORIGIN_CITY_NAME	DEST_AIRPORT_ID	DEST_CITY_MARKET_ID	DEST_CITY_NAME	DEP_TIME	DEP_DELAY	ARR_TIME	ARR_DELAY	AIR_TIME	FLIGHTS	DISTANCE	CARRIER_DELAY	WEATHER_DELAY	NAS_DELAY	SECURITY_DELAY	LATE_AIRCRAFT_DELAY
	1/24/16	AS	65	15841	35841	Wrangell, AK	14256	34256	Petersburg, AK	1057	-9	1140	10	15	1	31	NaN	NaN	NaN	NaN	NaN

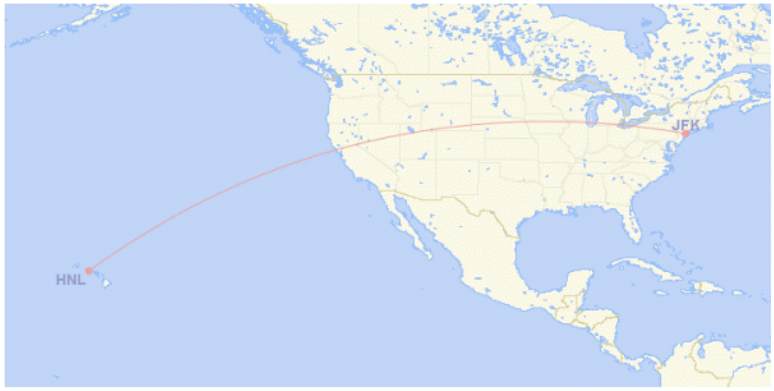


'Display the Longest flying time flight in our Database'

	FL_DATE	UNIQUE_CARRIER	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN_CITY_MARKET_ID	ORIGIN_CITY_NAME	DEST_AIRPORT_ID	DEST_CITY_MARKET_ID	DEST_CITY_NAME	DEP_TIME	DEP_DELAY	ARR_TIME	ARR_DELAY	AIR_TIME	FLIGHTS	DISTANCE	CARRIER_DELAY	WEATHER_DELAY	NAS_DELAY	SECURITY_DELAY	LATE_AIRCRAFT_DELAY
	1/12/16	HA	51	12478	31703	New York, NY	12173	32134	Honolulu, HI	955	-5	1655	30	698	1	4983	30	0	0	0	0

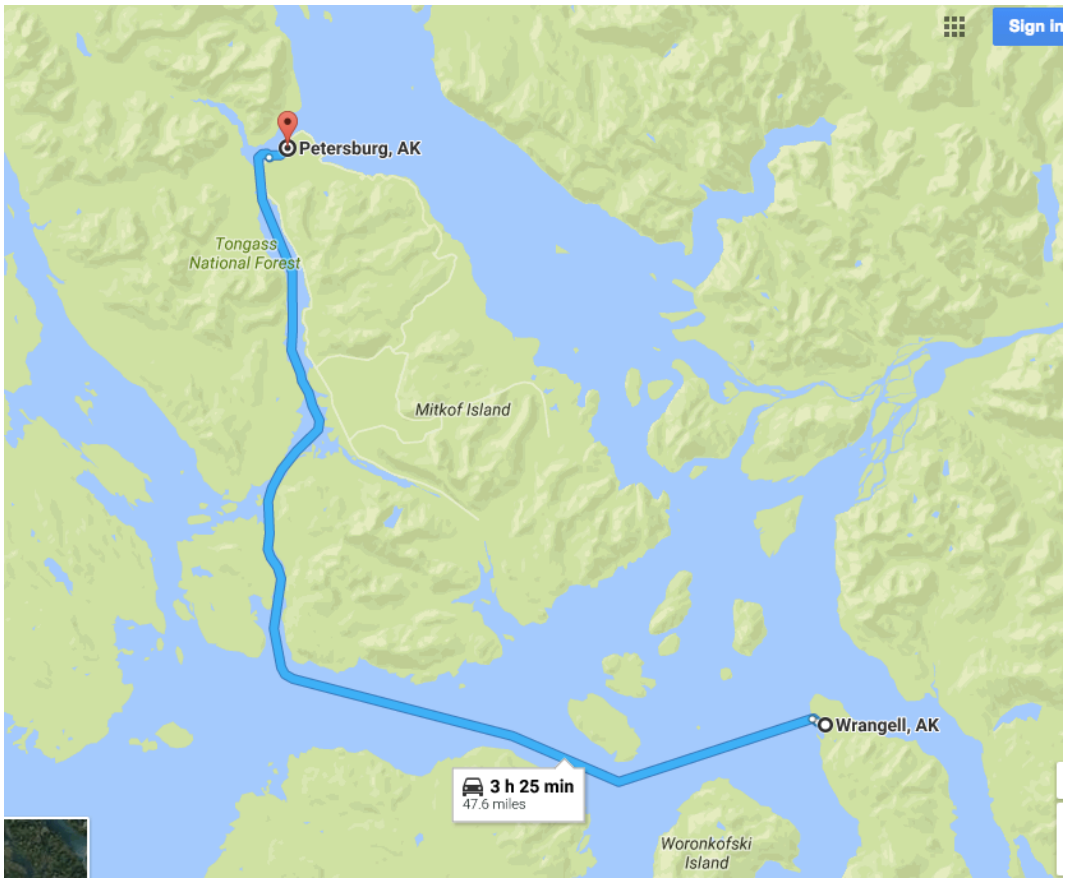
Longest non-stop flight operated in all 50 states:

Winner: New York (JFK) to Honolulu (HNL) at **4,983 miles** in 11 hours and 40 minutes. Operated by Hawaiian Airlines.



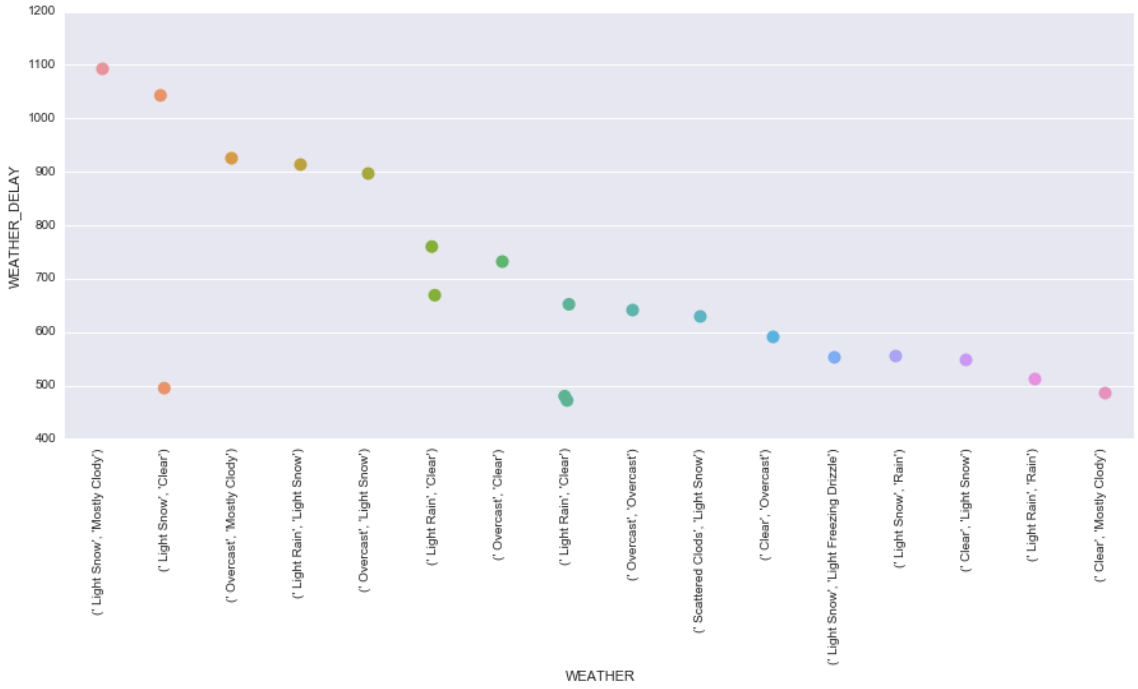
'Display the Shortest flying time flight in our Database'

	FL_DATE	UNIQUE_CARRIER	FL_NUM	ORIGIN_AIRPORT_ID	ORIGIN_CITY_MARKET_ID	ORIGIN_CITY_NAME	DEST_AIRPORT_ID	DEST_CITY_MARKET_ID	DEST_CITY_NAME	DEP_TIME	DEP_DELAY	ARR_TIME	ARR_DELAY	AIR_TIME	FLIGHTS	DISTANCE	CARRIER_DELAY	WEATHER_DELAY	NAS_DELAY	SECURITY_DELAY	LATE_AIRCRAFT_DELAY
	1/7/16	AS	65	15841	35841	Wrangell, AK	14256	34256	Petersburg, AK	1046	-20	1106	-21	8	1	31	NaN	NaN	NaN	NaN	NaN

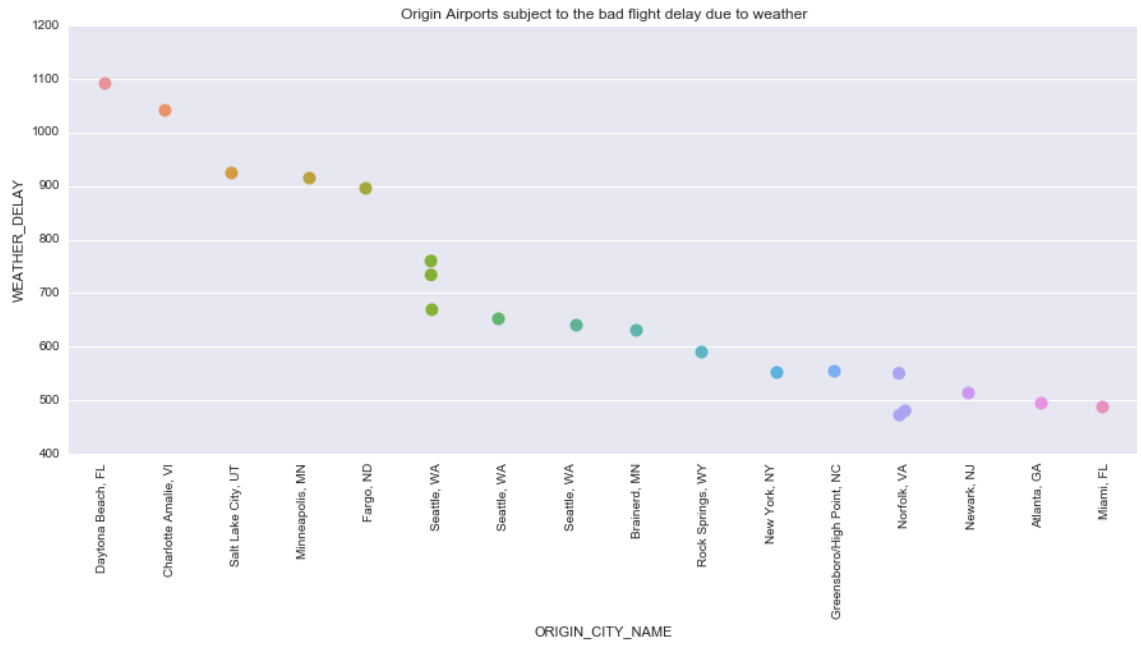


The data we have gathered is rich and lets see if we can try to glean some insights from it. Out of curiosity we will answer the following questions:

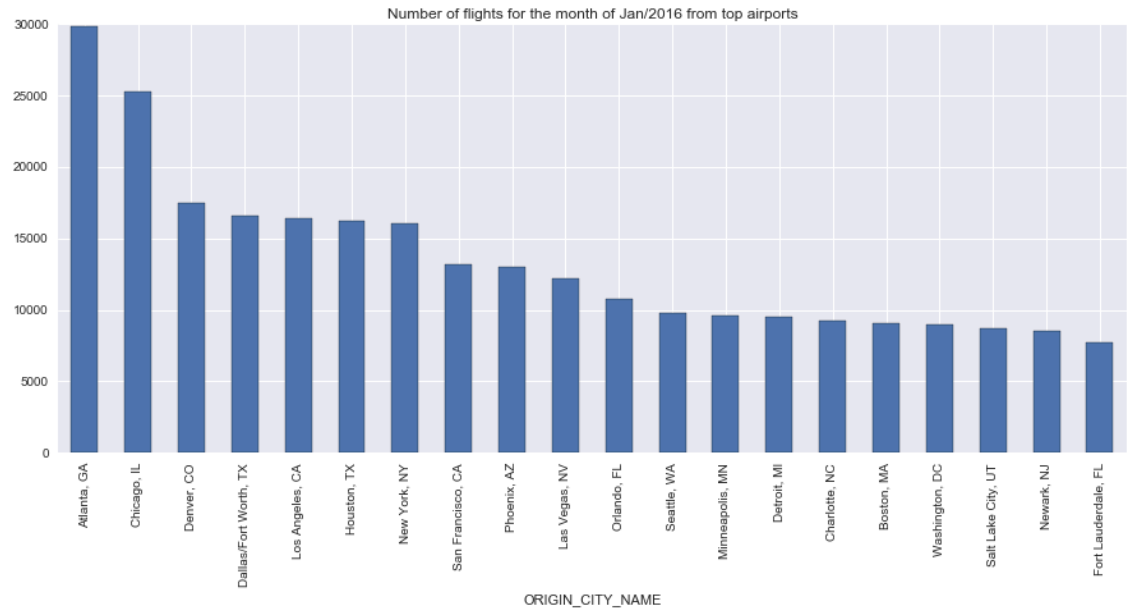
1) For the largest arrival delays which were related to weather and what was the weather like at these airports/cities on the day of flight.



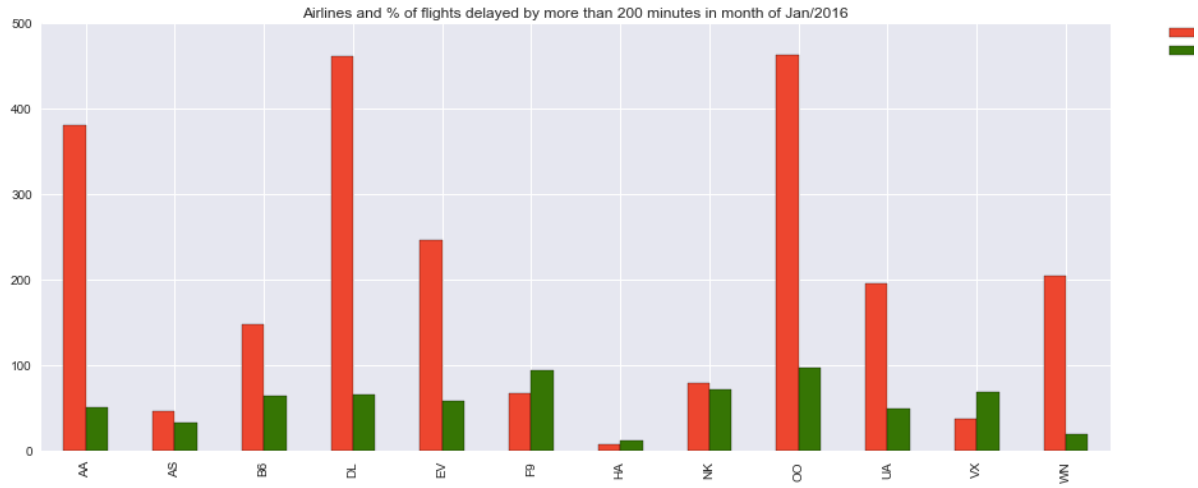
2) Which airport experienced these large weather related delays



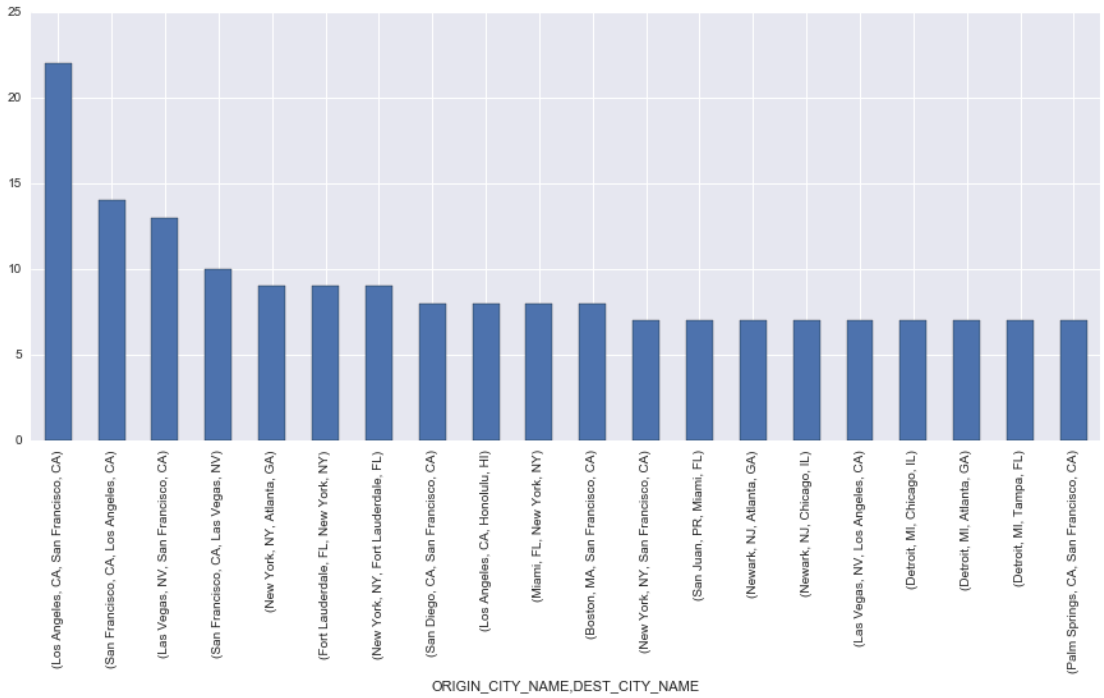
3) Finally we will plot out the airports which had the most flights for the month of January.



4) What airlines had the most number of flights in this month? What % of these flights were delayed by more than 200 minutes?



5) Which city pairs experienced the most delayed flights beyond 200 minutes?



Note that our data does not have any weather related information for the days of the flights (much less for the days of the significant flight delays). In order to correlate our flight delay data with the weather we will have to query historical data from http://www.wunderground.com.

Also note that since querying the website is very expensive we try to limit the data we query and hence we restrict it to only flights that were delayed more than 200 minutes and this delay was mostly weather related. In order to save time we have already saved data once queried as a csv file and this file is read by default. The behavior can be controlled by a variable (callApi) by changing its value from False to True.

Algorithms and Techniques

We use multiple regressors and classifiers in our analysis to work on our data

Regressors:
regressor1 = DecisionTreeRegressor(random_state=0)
regressor2 = RandomForestRegressor(random_state=0, n_estimators=20)

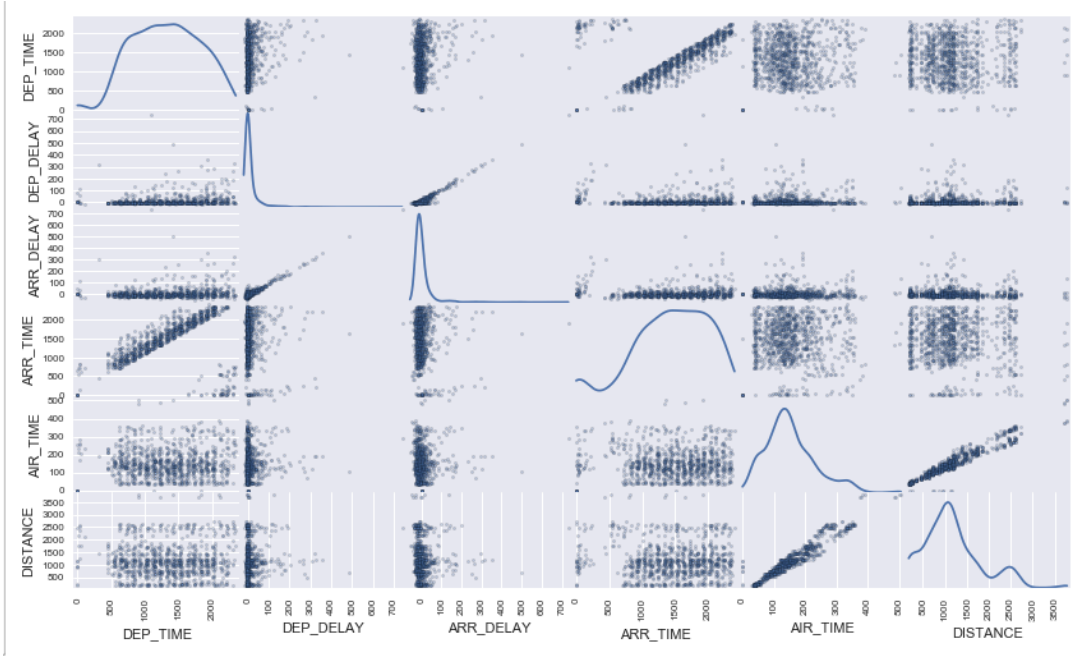
Classifiers:
clf_A = DecisionTreeClassifier(random_state=0)
clf_B = KNeighborsClassifier()
clf_C = GaussianNB()
clf_D = RandomForestClassifier(n_estimators=20)

Note that all of these classifiers and regressor are non-linear i.e they can work on a feature set even if there is no linear relationship between a feature and the predicted variable. However in order to predict a linear relationship between a feature and the predicted we define a R^2.

We find a string linear relationship between the DEP_DELAY and ARR_DELAY

```
Model has a coefficient of determination, R^2, of 0.859.
```

No other feature shows a string linear relationship. A scatter plot shows this visually



We run the regression and classification on two datasets. The first dataset is a reduced dataset including only the numerical continuous features. The second dataset is the full feature dataset including the categorical features that we mapped to numerical to allow for scikit-learn algorithms to run (more on this in the **Data Preprocessing** section)

Benchmark

We compare the predicted variable with the actual variable value to figure out the goodness of the algorithm. Under this scenario the benchmark (or the best prediction returns a value of 1)

Data Preprocessing

We carry out the following data preprocessing (in some cases upfront for all the models and sometimes specifically just before a model is used)

1) We make the model month and year agnostic. This means the model can be used for a date in the future for any month and year. Moreover this allows us to easily change a string/date object to an integer. More on that in the next point…….

2) Scikit-learn classification does not allow for categorical variables. This seems like a severe limitation of Python scikit-learn. In any case we either using mapping (recommended) or dummies (very expensive) to change categorical variables to numerical to be able to be used in regression/Classification.

3) Finally the predicted variable cannot be continuos for classification. Therefore we break it into ranges for classification problem. Classification on a continuos numerical variable (as is in our case for ARR_DELAY) is hard because it is hard to get the correct values of the ranges and the classifier cannot understand the ordinal nature of the data series once a continuos numerical variable is broken into ranges.

Implementation and Refinement

1) Having `DEP_DELAY` in the regression/classification feature set is a problem. `DEP_DELAY` is not known until the flight has departed. Thus any model that relies on `DEP_DELAY` to make predictions is useless by being dependent on a variable that cannot be known ahead of time. Thus we drop `DEP_DELAY` from our model. We also drop the following columns 'CARRIER_DELAY', 'WEATHER_DELAY', 'NAS_DELAY', 'SECURITY_DELAY', 'LATE_AIRCRAFT_DELAY' when we run the model on the full feature set because these delays are strongly related to `DEP_DELAY`.

2) We had to refine the ranges of the predicted variable in the classification to make the algorithm result more realistic. Because of the ordinal nature of the data and large range of the predicted variable the ranges that python comes up with automatically are so broad that any weak algorithm would also work fine. We refined the delay ranges with more precision and broke up the delays as

[-79, 0), = "early"
[0, 60), = "late",
[60 260), = "very late"

Model Evaluation and Validation

The following are the final prediction scores for our models.

Regression:

Regressor	Reduced Feature Set	Reduced Feature Set (Drop DEP_DELAY)	Full Feature Set
DecisionTreeRegressor	test set: 0.999827880785	test set: 0.999601775045	test set: 0.999601775045
RandomForestRegressor	test set: 0.999969064736	test set: 0.999984687763	test set: 0.999984687763

Classification:

Classifier	Reduced Feature Set	Full Feature Set
DecisionTreeClassifier	F1 score for training set: 0.9078	F1 score for training set: 1.0000
	F1 score for test set: 0.6693	F1 score for test set: 0.7189
KNeighborsClassifier	F1 score for training set: 0.7459	F1 score for training set: 0.8474
	F1 score for test set: 0.6534	F1 score for test set: 0.7745
GaussianNB	F1 score for training set: 0.5651	F1 score for training set: 0.5361
	F1 score for test set: 0.5642	F1 score for test set: 0.5391
RandomForestClassifier	F1 score for training set: 0.9042.	F1 score for training set: 0.9966
	F1 score for test set: 0.6706.	F1 score for test set: 0.7593

Justification

Out of these models we chose the RandomForestRegressor as the best model to optimize because of great predictive power. Further since the predicted variable is a continuous variable we prefer to choose a regressor. Running the grid search on this regressor returns an optimal max_depth of 20. We need to optimize the number of estimators as well.

```
Beginning optimization.......
Parameter 'max_depth' is 20 for the optimal model.

'Regression Score for regression prediction using a RandomForestRegressor with DEP_DELAY NOT part of the FULL feature
set'

0.999987715468
```

Reflection

Matplotlib is seriously deficient. It is not very intuitive to use and results may not be what you expect them to be.

Jupyter notebook also has some issues. It is hard to figure out if it is running or just hung up. Sometimes the kernel will be up but the notebook will not respond (even to simple commands.) Output of commands are not in the same order as they are executed. For example print statements will be output before any plots are created.

Improvement

We have run PCA on the dataset and have good results but are not sure how to interpret the results yet and if PCA is useful.

We can run clustering on this dataset as well, however the number of data points may be too large for a lazy classifier.