**Predicting Upcoming Flight Delays**

University of Maryland, Baltimore County

DATA 606 – Capstone in Data Science

Shalini Jain

Spring 2020

**Project Overview (motivation, problem definition) (from Delivery-1)**

The goal of my project is to predict upcoming flight delays based on previous delays via Machine Learning. My project can help airports and airlines prevent future delays by knowing what is causing the delay. As a result, they will be able to increase productivity. Additionally, it can help customers know which airlines/airports to use and/or avoid. I plan to use a dataset found on Kaggle, *“2015 Flight Delays and Cancellations.”*

The data was recorded by the U.S. Department of Transportations (DOT) Bureau of Transportation Statistics which tracks the on-time performance of domestic flights operated by large air carriers. The dataset includes information on a number of on-time, delayed, canceled, and diverted flights that were published in DOT’s monthly Air Travel Consumer Report. I plan to use the CSV files within the dataset. The first CSV, “airlines” is 359 Bytes and has two columns. The second CSV, “airports” is 24 KB and contains seven columns. Lastly, the third CSV, “flights” is 592.4 MB and contains 31 columns.

**Literature Review**

Many projects started off by constructing a model with a binary classifier which means that they were predicting that either the flight was delayed or not delayed. However, I want to use a multi-class model that will predict the magnitude of the delay. Therefore, I will not be using the binary classifier model. As mentioned before, I am mostly focusing on weather induced delays. Other projects focus on various types of delays such as, aircraft delays, security delays, etc. Additionally, other projects only focus on American Airlines, while my project will analyze fourteen airlines. It also only focuses on five airports, while mine will focus on 322 airports. I am hoping this will give me a lot more data to work with in order to predict more accurate delays.

**Data Preparation, Data Cleaning, and Exploratory Data Analysis (EDA)**

After receiving feedback from my peers, further data exploration, and other research I decided to make a few changes in order to improve my project overall. For one, I became more interested in weather induced delays.

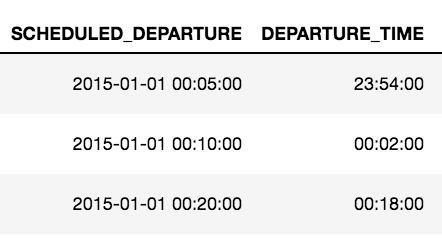
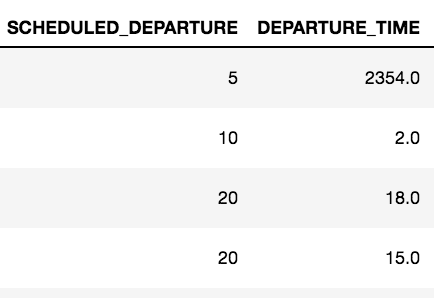
While conducting my own research on my dataset, I decided to do weather induced delays because it seemed appealing and about 18.56% of the delays in the dataset were due to weather related issues. As a result, I decided to use a weather Application Programming Interface (API) that would match my dataset. I first attempted to use DarkSky Application Programming Interface (API), I soon switched to World Weather Online API. I found that World Weather Online API seemed to be much better suited for my project. This API allowed me to pull historical weather data from January 1, 2015 to December 31, 2015 which matches the data in original flights data-frame. The API obtains many columns that will be helpful, including data in centimeters of snow, wind gust, cloud cover, wind direction degree, wind speed, etc.

Before I could complete the next stages of my project, I needed to perform cleaning and transformations. For example, there were many null values or unnecessary columns. Once my dataset is good to go, I plan to use machine learning algorithms to test my data and find predicted flight delays. Although the analysis and testing aspects of my project are essential, I also wanted to ensure that my visualizations represented my results in an appropriate manner. I played around with various visualization techniques to see which one best represents my data as well as the results.

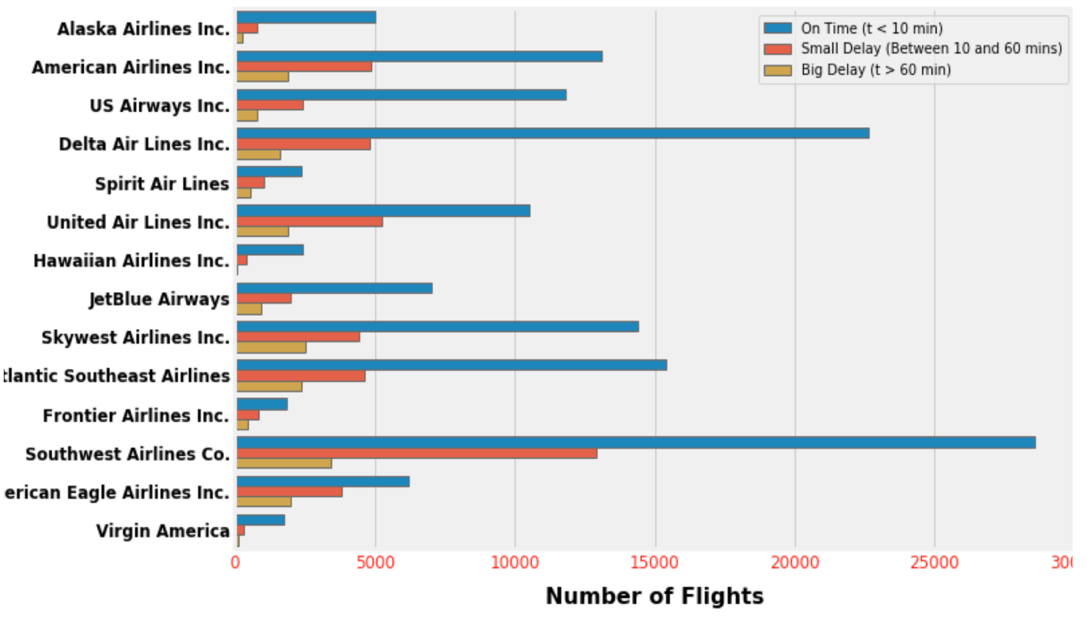
I cleaned, transformed, and performed EDA on the flight’s dataset. Luckily, my dataset was already in pretty good shape beforehand. Because I am only focusing on weather induced delays, I deleted all rows that stated that flights were cancelled. I also deleted other delay types (i.e. security delay) because I was more focused on weather induced delays. The flight dataset originally had 5,819,079 rows. 89,884 of those rows were cancelled flights, which left me with 5,729,195 rows after I cleaned the dataset. I then also deleted the “reason of cancellation” column because it became irrelevant.

As mentioned earlier, I am lucky because my dataset was already in pretty good shape. However, there was still some more data cleansing and Exploratory Data Analysis (EDA) to perform. For one, the date was split into three different columns (year, month, and day) which I then combined into “DATE.”. I also converted the “DATE” to a datetime.time. As a result, I also had to convert the date in the historical weather data-frame so that the two data-frames could match. I then joined the two data-framed based on the “DATE” column.

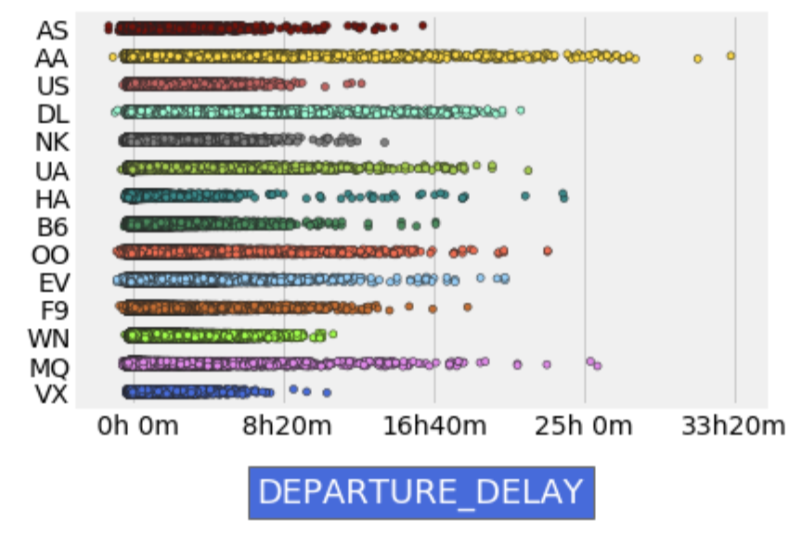
From the historical weather data-frame, I dropped many columns that did not seem necessary, such as: sunHour, uvIndex, uvIndex1, moonrise, moonset, moon illumination, sunset, sunrise, temperature, heat index, and feels like.

****

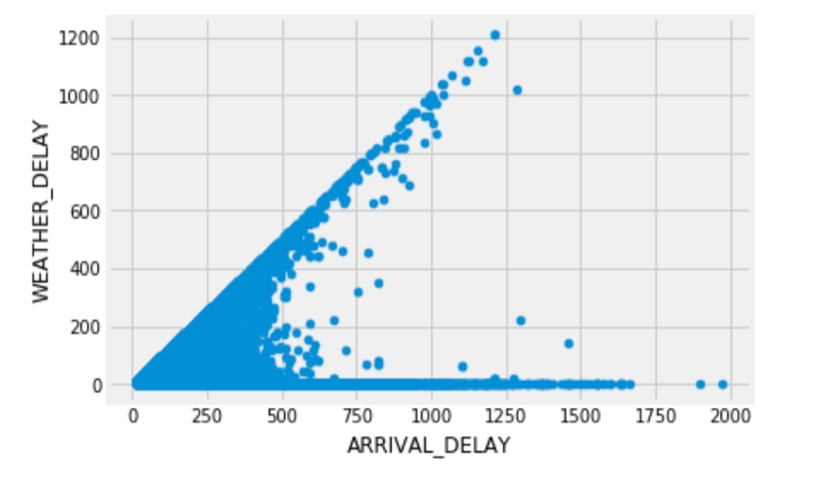
***Fig 1****: Above is an example of how the times were formatted before (left) to how they are now formatted*



***Fig 2:*** *Shows the magnitude of delays based on each airline. On-time flights include delays less than 10 mins while big-delayed flights begin at 60 minutes*



**Fig 3:** *Shows departure delays based on each of the airports. For example, American Airlines (AA) has a lot of delays while Virgin Airlines does not*

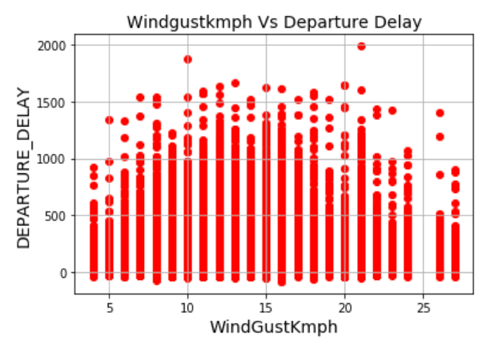
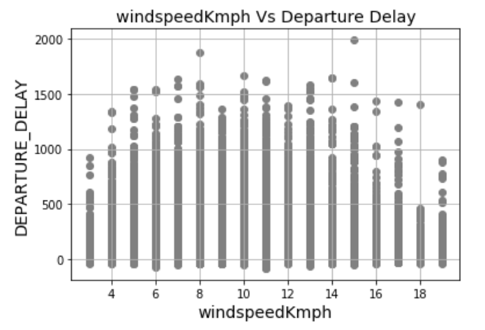


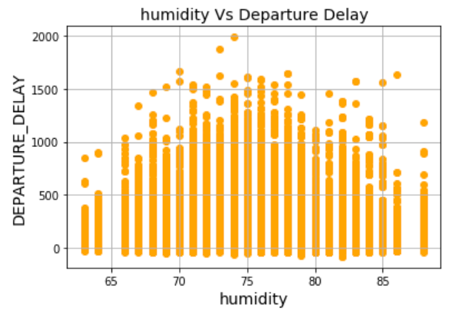
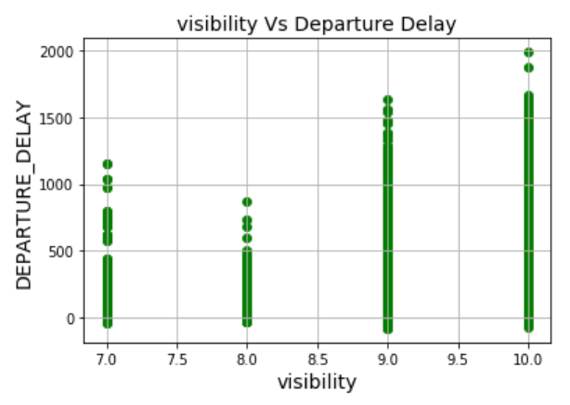
**Fig 4:** *Shows departure delays based on each of the airports. For example, American Airlines (AA) has a lot of delays while Virgin Airlines does not*

**Model Construction and Implementation**

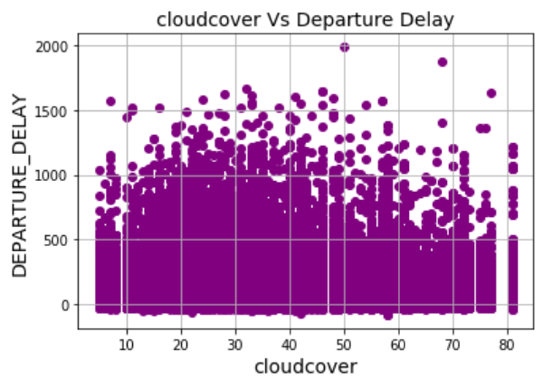
I plan to focus on Machine Learning (ML) algorithms. Originally, I wanted to use various machine learning algorithms and then compare the accuracy. Because I wanted to use certain variables for my model(s), I used label encoder. This allowed me to convert categorical variables into a form that can be used for my Machine Learning predications. I also created a matrix to then allow me to implement linear regression and then completed a linear regression however; I was not satisfied. I used Scikit-Learning which changes categorical/text data into numbers. My input value was departure minutes and my Mean-Square-Error (MSE) resulted as 74.8 which I obviously was not pleased by.

After more research and thought, I decided to solely focus on Multiple Linear Regression models. My goal of doing so is to explore the relationships of different variables from the dataset and to also compare the accuracy of each model. Before constructing my models, I wanted to look at the relationship between the variables I wanted to use and Departure Delay. Below are a few image examples that show the relationship between a variable (i.e. wind gust) and departure delay.





**Fig 5:** *Examples of graphs that I used to check the linearity between a variable and departure delay*



As you can see in the images above, there is clearly a relationship between departure delay and these variables. So, for each of my models, I included two independent variables while departure delay was always my dependent variable. The formula I used for my multiple linear regression is as follows:

**Departure Delay(Y) = (Intercept) + (Ind. Variable 1 coef)\*X1 + (Ind. variable 2 coef)\*X2**

The two independent variables in the formula above changed depending on the model. For example, in one of the models I used arrival delay and visibility as my independent variables and in another model, I used humidity and maximum temperature. For X1 and X2, I used the averages of the respected independent variable. To check the accuracy of my models, I looked at the R-squared value and the Standard Error. Next, I actually implemented the model and compared results.

**Results**

***Model 1:*** I used Arrival Delay and Wind Gust as my independent variables and Departure Delay as my dependent variable

* **The predicted departure delay:** 10 minutes
* **The R-squared value:** 0.887
* **Standard Error:** 0.017

***Model 2:*** I used Arrival Delay and Visibility as my independent variables and Departure Delay as my dependent variable

* **The predicted departure delay:** 9.5 minutes
* **The R-squared value:** 0.887
* **Standard Error:** 0.098

***Model 3:*** I used Cloud Cover and Visibility as my independent variables and Departure Delay as my dependent variable

* **The predicted departure delay:** 9.5 minutes
* **The R-squared value:** 0.00
* **Standard Error:** 0.321

***Model 4:*** I used Maximum Temperature and Humidity as my independent variables and Departure Delay as my dependent variable

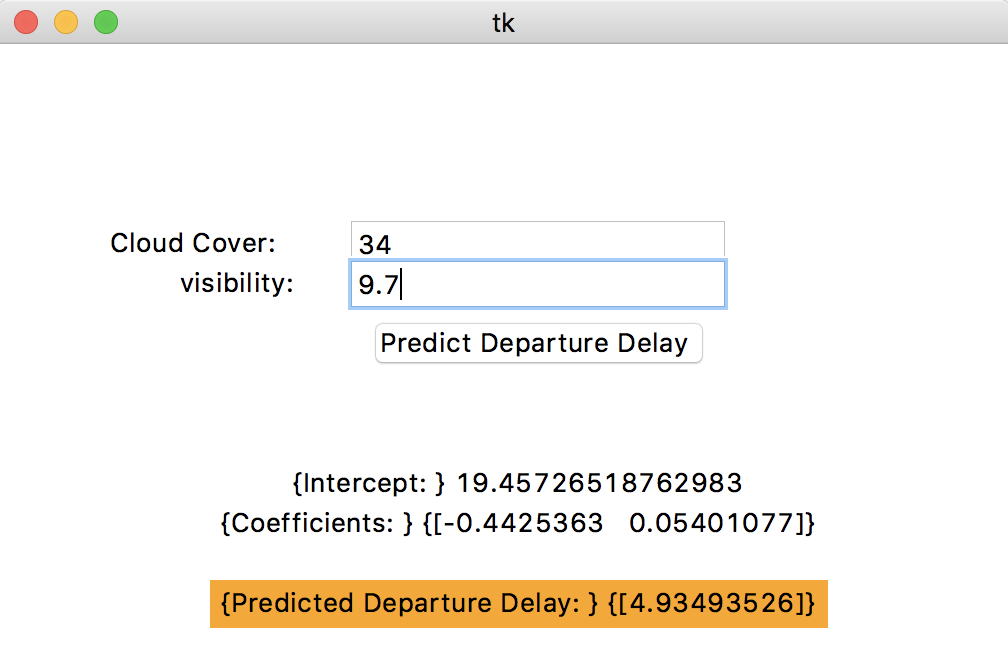
* **The predicted departure delay:** 5 minutes
* **The R-squared value:** 0.00
* **Standard Error:** 0.771

***Model 5:*** I used Wind Gust and Wind Speed as my independent variables and Departure Delay as my dependent variable

* **The predicted departure delay:** 9.6 Minutes
* **The R-squared value:** 0.002
* **Standard Error:** 0.052

The results on my models were not as great as I had wished. Overall, Model 1 and Model 2 seemed to have the most accuracy and best results. They had the highest R-squared value and a low Standard Error. Both of these models include Arrival Delay which is most likely the reason for its high accuracy. Additionally, it seems that Wind Gust and Visibility also play a large part in departure delays. The R-squared value for Models 1, 2, and 3 are either very low or are zero. While I am disappointed to see the lower R-squared values, I plan to create models with a better fit in the future.

**Graphical User Interface**



**Fig 6:** *After creating the various models, I realized I wanted to create a sample interface where a user can input data and see a predicted departure delay. In this particular image, the user can input Cloud Cover and Visibility*

**Setbacks and Future Endeavors**

A major setback for me during this project, was that I was unable to obtain data on snow that matched my Flights dataset. While I used the World Weather API to collect all data for 2015, it did not have any data on snow. I even used a search engine (google) to confirm that it did snow in 2015. In the future, I plan to find historical weather data that includes snow data by using a different weather API. Moreover, I may attempt to find a more up-to-date dataset that already include weather data. Furthermore, I plan to improve my current models and create new models to increase accuracy. I also would like to create a Random Forest model(s). Lastly, I plan to play more with the GUI I created. I would like to include more variables for users to input (i.e. Min/max temperature or inches of snow).

References

Brownlee, J. 2016, March 25. Machine Learning Mastery. *Linear Regression for Machine Learning.* Retrieved from, <https://machinelearningmastery.com/linear-regression-for-machine-learning/>

Cassidy, I. 2018, October 1st. Upside. *Applying Predictive Analytics to Flight Delays. Retrieved from,* <https://engineering.upside.com/applying-predictive-analytics-to-flight-delays-85413ca4939f>

Chakrabarty, N. N.a. Arxiv. *A Data Mining Approach to Flight Arrival Delay Prediction for American Airlines.* Retrieved from, <https://arxiv.org/pdf/1903.06740.pdf>

N.a. N.a. World Weather Online. Retrieved from, <https://www.worldweatheronline.com/developer/api/>

N.a. N.a. 2017, February 09. Kaggle. *2015 Flight Delays and Cancellations.* Retrieved from, <https://www.kaggle.com/usdot/flight-delays#airlines.csv>

N.a. N.a. Yale Stats. Multiple Linear Regression. *Retrieved from,* <http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm>