# Big Data Analytics and Text Mining

# Flight Delay Analysis

Final Project Report

Ramakanth Vemula
167004695/rv356

Krishna Anantha Padmanabhan
167007637/ka478

Department of Computer Science
Rutgers University, New Brunswick, NJ

May 4, 2017

# 1 Project Goal

The aim of our project is to perform an analysis of Flight delays in the US and build a model to try and predict future delays for a flight path.
Flight data contained within datasets such as the one from Bureau of Transportation Statistics [1] are pretty comprehensive. Aircraft carriers and the Airport Authorities can use this data to improve and streamline their services. Generally, medium and large airports typically service hundreds to thousands of flights per day which sometimes means that delay in one flight (whatever the cause) might have a cascading effect leading to a delay in subsequent flights. Using the data, we gain an understanding into previous delays and factors affecting these. Airline services can then use this information to appropriately move resources around to prevent and mitigate delays, thereby improving customer satisfaction.
The delays experienced can be looked at from two different perspectives. One, from the customer perspective where only the arrival delay matters to him. The other is through the Airport Authority/Carrier point of view where the other delays such as Weather and security also matter. We perform an analysis with respect to both and present results. We have also implemented machine learning algorithms to gain a deeper insight into the data and predict flight delays.

# 2 Information on Data

We are using DataBricks community edition in order to collaborate efficiently on the cloud.

## 2.1 Data Source

We have obtained the data from the Bureau of Transportation Statistics [1] that provides extensive aviation information for flights in the United States of America. We have chosen to work with 3 years' data, that is, 2014, 2015 and 2016. This collection of specific data is done from the website itself.

## 2.2 Data Format

The data is in the form of multiple csv files containing information for each month. Thus we have a total of 36 files with 17,256,548 records. We now filter the data to keep only the fields that are required.

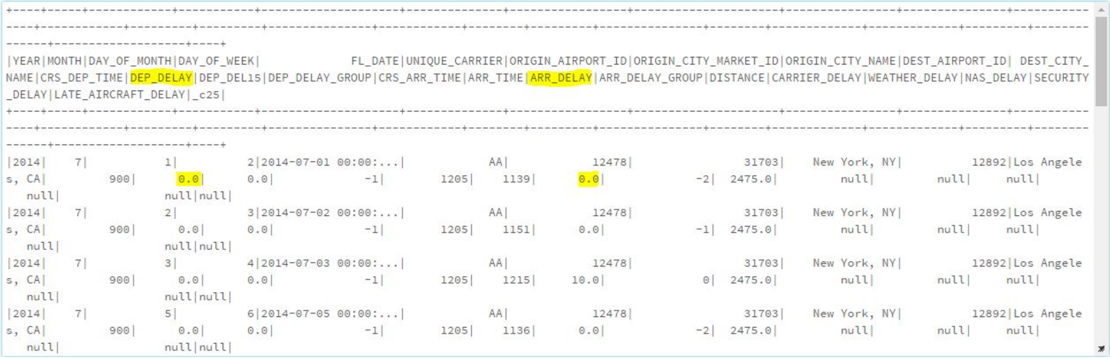
**Fields Used**

- YEAR: Integer field that contains the year of flight departure.
- MONTH: Integer field that contains the month in which the flight departed.
- DAY_OF_MONTH: Integer field that contains the day of the month that the flight departs.
- DAY_OF_WEEK: Integer field that contains the day of the week that the flight departs. It starts with 1 for Monday.
- FL_DATE: A timestamp that contains the date of flight departure.
- UNIQUE_CARRIER: String field that contains the 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).
- ORIGIN_AIRPORT_ID: Integer field that contains the identification number assigned by US DOT to identify a unique airport.
- ORIGIN_CITY_MARKET_ID: Integer field that contains the identification number assigned by US DOT to identify a unique city.
- ORIGIN_CITY_NAME: String field that contains the Origin Airport, City Name.
- DEST_AIRPORT_ID: Integer field that contains the Destination Airport, Airport ID. An identification number assigned by US DOT to identify a unique airport.
- DEST_CITY_NAME: String field that contains the Destination Airport, City Name.
- CRS_DEP_TIME: Integer field that contains the CRS Departure Time (local time: hhmm).
- DEP_DELAY: Double field that contains difference in minutes between scheduled and actual departure time. Early departures show negative numbers.
- DEP_DEL15: Double field that contains Departure Delay Indicator, 15 Minutes or More (1=Yes).

- DEP_DELAY_GROUP: Integer field that contains Departure Delay Group. It groups delays such that 15-40 minutes is group 1, 40-70 is group 2 and so on.
- CRS_ARR_TIME: Integer field that contains the CRS Arrival Time (local time: hhmm).
- ARR_TIME: Integer field that contains the Actual Arrival Time (local time: hhmm).
- ARR_DELAY: Double field that contains difference in minutes between scheduled and actual arrival time. Early arrivals show negative numbers.
- ARR_DELAY_GROUP: Integer field that contains Arrival Delay Group. It groups delays such that 15-40 minutes is group 1, 40-70 is group 2 and so on.
- DISTANCE: Double field that contains the distance between airports (miles).
- CARRIER_DELAY: Double field that contains the Carrier Delay, in minutes.
- WEATHER_DELAY: Double field that contains the Weather Delay, in minutes.
- NAS_DELAY: Double field that contains the National Air System Delay, in minutes.
- SECURITY_DELAY: Double field that contains the Security Delay, in minutes.
- LATE_AIRCRAFT_DELAY: Double field that contains the Late Aircraft Delay, in minutes.

## 2.3 Data Preprocessing

Our data contains two fields that need to be preprocessed, namely DEP_DELAY and ARR_DELAY. These fields contain negative values that depart or arrive early respectively. We have changed all the negative numbers to 0(zero) since we are only concerned about delayed flights. An example of the work done is in figure 1 with the modified data highlighted. We then remove the records with incomplete data and then use one hot encoder, string indexer, and vector assembler on specific attributes in order to group the data into a set of features along with the label.



```
|YEAR|MONTH|DAY_OF_MONTH|DAY_OF_WEEK|            FL_DATE|UNIQUE_CARRIER|ORIGIN_AIRPORT_ID|ORIGIN_CITY_MARKET_ID|ORIGIN_CITY_NAME|DEST_AIRPORT_ID| DEST_CITY_
NAME|CRS_DEP_TIME|DEP_DELAY|DEP_DEL15|DEP_DELAY_GROUP|CRS_ARR_TIME|ARR_TIME|ARR_DELAY|ARR_DELAY_GROUP|DISTANCE|CARRIER_DELAY|WEATHER_DELAY|NAS_DELAY|SECURITY
_DELAY|LATE_AIRCRAFT_DELAY|_c25|
|2014|    7|           1|          2|2014-07-01 00:00:...|            AA|            12478|                31703|   New York, NY|          12892|Los Angele
s, CA|         900|      0.0|      0.0|             -1|        1205|    1139|      0.0|             -2|  2475.0|         null|         null|     null|
   null|               null|null|
|2014|    7|           2|          3|2014-07-02 00:00:...|            AA|            12478|                31703|   New York, NY|          12892|Los Angele
s, CA|         900|      0.0|      0.0|             -1|        1205|    1151|      0.0|             -1|  2475.0|         null|         null|     null|
   null|               null|null|
|2014|    7|           3|          4|2014-07-03 00:00:...|            AA|            12478|                31703|   New York, NY|          12892|Los Angele
s, CA|         900|      0.0|      0.0|             -1|        1205|    1215|     10.0|              0|  2475.0|         null|         null|     null|
   null|               null|null|
|2014|    7|           5|          6|2014-07-05 00:00:...|            AA|            12478|                31703|   New York, NY|          12892|Los Angele
s, CA|         900|      0.0|      0.0|             -1|        1205|    1136|      0.0|             -2|  2475.0|         null|         null|     null|
   null|               null|null|
```

Figure 1: Data Preprocessing

# 3 Analysis of Data

The delays experienced can be looked at from two different perspectives. One, from the customer perspective where only the arrival delay matters to him. The other is through the Airport Authority/Carrier point of view where the other delays such as Weather and security also matter. We perform an analysis with respect to both and present a few results.

## 3.1 Cause of Most Delay

There are primarily 5 factors affecting delays :

(a) Carrier Delay

(b) Weather Delay

(c) National Air System Delay

(d) Security Delay

(e) Late Aircraft Delay

We take a count of all the delays which appear.
By looking at the pie chart in Fig **??**, we see that the primary cause of delay is Late Aircrafts i.e Aircrafts which take a longer time to cover the distance they are supposed to. The carrier delay comes at a close second.

| Carrier | Weather | NAS | Security | Late Aircraft |
|---|---|---|---|---|
| 60965702 | 8694985 | 45025928 | 228260 | 77904313 |

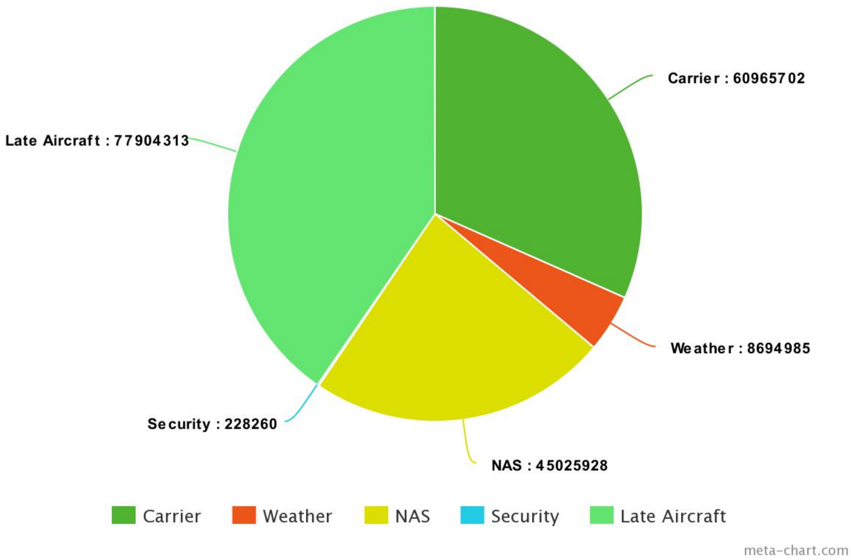



Figure 2: Pie Chart Showing Distribution of Delay Causes

## 3.2 Airports with the Most Delay

Using the data we can view the numbers of delayed flights in an airport. Figure **??** shows the cities sorted based on the decreasing order of number of flights. As expected, the most Flight delays arise from bigger airports in bigger cities.
Also, it an be noted that though there are more flights getting delayed from Atlanta, GA as compared to Los Angeles, CA, the percentage of flights delayed as compared to the total number of flights is smaller.

| ORIGIN_CITY_NAME | FlightDelay | FlightTotal | Percentage |
|---|---|---|---|
| Chicago, IL | 260746 | 1109467 | 23.50191578478675 |
| Atlanta, GA | 191264 | 1133578 | 16.87259279908396 |
| Dallas/Fort Worth... | 153095 | 734953 | 20.830583724401425 |
| Denver, CO | 141128 | 664532 | 21.237201519264687 |
| Los Angeles, CA | 131642 | 647316 | 20.33658985719494 |
| Houston, TX | 129979 | 641604 | 20.258446019663218 |
| New York, NY | 119879 | 612155 | 19.58311212029633 |
| San Francisco, CA | 104602 | 501429 | 20.860779891071317 |
| Las Vegas, NV | 89818 | 433610 | 20.714005673300893 |
| Phoenix, AZ | 84784 | 479337 | 17.68776455812925 |
| Newark, NJ | 75053 | 337533 | 22.23575176353127 |
| Orlando, FL | 69718 | 356161 | 19.574855191893555 |
| Baltimore, MD | 61545 | 281170 | 21.888892840630223 |
| Washington, DC | 59086 | 360778 | 16.377384430314486 |
| Boston, MA | 58427 | 349907 | 16.697865432815004 |
| Charlotte, NC | 53501 | 332815 | 16.075297086970238 |
| Detroit, MI | 53497 | 348364 | 15.356638458623738 |
| Seattle, WA | 51304 | 362626 | 14.14790996784566 |

Figure 3: Table Showing Total number of Flights Delayed, Total Flights and % of Flights Delayed for an Airport

### 3.3 Carriers with the Most Delay

Similar to the above section, we can view the total number of flights delayed by each carrier. This information can help carriers see their position w.r.t others and also help customers in making a better decision while booking the flight.
For instance from Figure **??**, the carrier HA (Hawaiian Airlines) might be more reliable thought it might operate lesser number of flights as compared to the larger ones like WN.

| UNIQUE_CARRIER | FlightDelay | FlightTotal | Percentage |
|---|---|---|---|
| WN | 809971 | 3735932 | 21.68056056694822 |
| AA | 374146 | 2178176 | 17.177032526297232 |
| DL | 351809 | 2599002 | 13.536311245624281 |
| UA | 326210 | 1554318 | 20.987339785037552 |
| EV | 324367 | 1748988 | 18.545982019316313 |
| OO | 286797 | 1807316 | 15.868669341719986 |
| B6 | 168615 | 799214 | 21.097603395335916 |
| MQ | 135195 | 687333 | 19.669505174347805 |
| US | 82851 | 613380 | 13.507287488995402 |
| NK | 61000 | 255578 | 23.867469030980757 |
| F9 | 57661 | 271431 | 21.243336243833607 |
| AS | 49971 | 510058 | 9.797121111716706 |
| VX | 33783 | 188534 | 17.918783879830695 |
| FL | 13639 | 79495 | 17.15705390276118 |
| HA | 13167 | 227793 | 5.780247856606656 |

Figure 4: Table Showing Total number of Flights Delayed, Total Flights and % of Flights Delayed for a Carrier

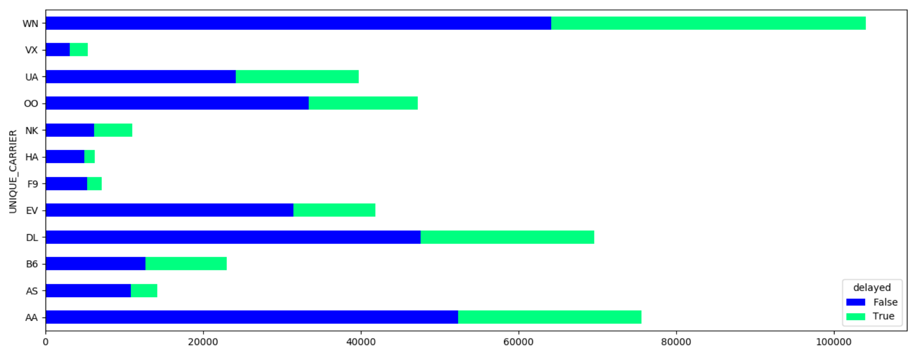



Figure 5: Carrier Flights On Time and Delayed vs. Total Flights

## 3.4 Routes with the Most Delay

Every country has popular routes which generally contain more flights. We can analyze which of these paths are the most prone to delays.

From Figure 6 we can see that the routed from Los Angeles to San Francisco and vice versa is a pretty popular route, and it also has the most Delays.

| ORIGIN_CITY_NAME | DEST_CITY_NAME | FlightDelay | FlightTotal | Percentage |
|---|---|---|---|---|
| Los Angeles, CA | San Francisco, CA | 11777 | 45400 | 25.940528634361232 |
| Chicago, IL | New York, NY | 10880 | 41550 | 26.18531889290012 |
| San Francisco, CA | Los Angeles, CA | 9953 | 46097 | 21.591426773976615 |
| New York, NY | Chicago, IL | 8328 | 41803 | 19.92201516637562 |
| Chicago, IL | Los Angeles, CA | 8216 | 31616 | 25.986842105263158 |
| Chicago, IL | San Francisco, CA | 8205 | 26760 | 30.661434977578477 |
| Chicago, IL | Washington, DC | 7556 | 33698 | 22.42269570894415 |
| Fort Lauderdale, FL | New York, NY | 7396 | 28949 | 25.548378182320633 |
| Los Angeles, CA | Las Vegas, NV | 7395 | 33481 | 22.087153908186732 |
| Miami, FL | New York, NY | 7303 | 30904 | 23.631245146259385 |
| Orlando, FL | New York, NY | 7120 | 29316 | 24.287078728339473 |
| Atlanta, GA | New York, NY | 7103 | 32029 | 22.176777295575885 |
| Chicago, IL | Minneapolis, MN | 7041 | 28829 | 24.423323736515314 |
| Las Vegas, NV | San Francisco, CA | 6988 | 25579 | 27.31928535126471 |
| Los Angeles, CA | New York, NY | 6986 | 38223 | 18.276953666640505 |
| Chicago, IL | Atlanta, GA | 6911 | 34538 | 20.00984422954427 |
| Chicago, IL | Denver, CO | 6888 | 26932 | 25.575523540769346 |
| Los Angeles, CA | Chicago, IL | 6833 | 32592 | 20.965267550319098 |

Figure 6: Table Showing Total number of Flights Delayed, Total Flights and % of Flights Delayed for Different Routes

## 3.5 Days with Delayed Flights

We tried to find out if specific days of the week had more delayed flights than other days. The results for that are in Figure 8. DAY_OF_WEEK has numbers from 1 to 7 with 1 being Monday and other days following it. Almost all days have 15 - 18 percent of delayed flights but a key piece of information that we can observe is that during the middle of the week (Tuesdays and Wednesdays), the number of flights is lesser and the percentage of flights delayed is less as compared to weekend flights (Sunday, Monday).

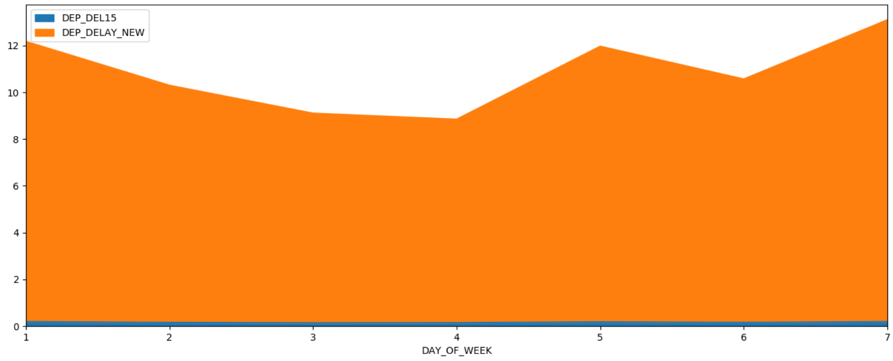



Figure 7: Graph Showing Delays on the Days in the Week

| DAY_OF_WEEK | FlightsDelayed | FlightsOnDay | Percentage |
|---:|---:|---:|---:|
| 1 | 474703 | 2561802 | 18.53004252475406 |
| 2 | 428057 | 2504060 | 17.09451850195283 |
| 3 | 437641 | 2543920 | 17.203410484606433 |
| 4 | 497393 | 2567913 | 19.369542503971125 |
| 5 | 494534 | 2575163 | 19.20398825239412 |
| 6 | 331267 | 2086821 | 15.874241250207852 |
| 7 | 425587 | 2416869 | 17.609022251516322 |

Figure 8: Delays on Specific Days of the Week

## 3.6 Months with Delayed Flights

Once we analyzed the days of a week, we moved on to finding out the delays in specific months. The results for the year of 2016 are in Figure 10. 1 represents the month of January and so on. We can observe from the data that the summer months of June, July and August have the highest volume of flights and the delay is also higher in these months. Another important information is that the month of December also has a high number of delayed flights since it contains the Christmas holidays when people travel a lot and it is also during the winter when weather might play a role in flight delays.


Figure 9: Graph Showing Delays based on Month

| MONTH | FlightsDelayed | FlightsInMonth | Percentage |
|---|---|---|---|
| 1 | 270385 | 1387744 | 19.483780870246964 |
| 2 | 242694 | 1283682 | 18.906084217119194 |
| 3 | 265640 | 1487192 | 17.86184971409206 |
| 4 | 225976 | 1430280 | 15.799423889028722 |
| 5 | 258722 | 1475629 | 17.532997792805645 |
| 6 | 330022 | 1494151 | 22.087593556474545 |
| 7 | 327110 | 1544055 | 21.185126177500155 |
| 8 | 288433 | 1516374 | 19.021230910052534 |
| 9 | 185306 | 1389313 | 13.337959120802871 |
| 10 | 200208 | 1449802 | 13.80933396422408 |
| 11 | 200034 | 1380964 | 14.48509881503066 |
| 12 | 294652 | 1417362 | 20.78876109279069 |

Figure 10: Delays on Specific Months

# 4 Machine Learning Experiments

There are widespread applications of the data depending upon the needs of the user. As stated earlier, customers might need to know about arrival delays while Aviation experts might look at wide ranging data. If we talk in terms of a consumer, a machine learning model could inform the user before booking a flight if that flight would be delayed or not. On the other hand, aviation experts could find out the implications of weather delay, security delay, etc., on overall delay.

For the next phase of the project, we ran different machine learning algorithms to predict if a flight would be delayed or not. Their results are as follows:

| **Algorithm** | **Accuracy Percentage** |
|---|---|
| Logistic Regression | 83.145 % |
| Naive Bayes | 81.119 % |
| Decision Tree | 84.131 % |
| Random Forest | 80.099 % |

Table 1: Comparison of Machine Learning Algorithms

# 5 Additional Work

We have developed a front end user interface that lets users select flights and displays delays. It determines the user type first and on that basis displays the options available to them. For example, if a traveler starts using the application, it displays a list of all flights between different destinations and then displays the predicted delay for that flight plan. Similarly, for an air traffic controller it will display options to select the analysis that has to be done on different parameters such as carrier, weather, etc. This implementation is not in working stage yet since a Java wrapper has to be built around a working Scala class. Even though the Scala part of the work is complete, we are facing issues with integrating the Scala and Java parts of the code. we believe that we will be able to complete it given another month and the entire application will be hosted on an open source platform such as GitHub. Figures Figures 11 and  12 will give a better understanding of the application.


Figure 11: Main Screen of the Application

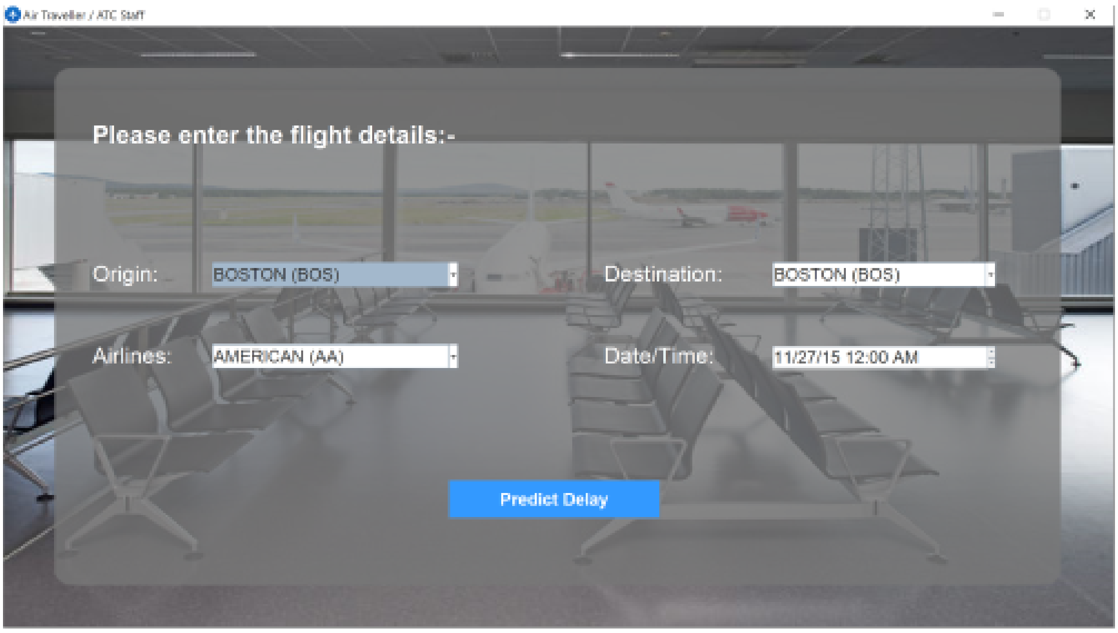



Figure 12: Second Screen of the Application

# 6 Conclusion

This was our first time working on a big data processing engine. Spark is a really powerful tool to learn and perform data operations while at the same time it simplifies the coding constructs.
The distinction between RDDs, Dataframes and Datasets is not obvious initially. Our project has given us a better understanding about these concepts. The Machine Learning algorithms are inbuilt into spark and templates can easily be found. This made it easy for us to quickly build our code and perform checks as to which fields/attributes carried the most weight in performing our classification.

# 7 Acknowledgements

We have primarily used Databricks to implement the Scala code with Spark engine. We have also used the MLlib library available on spark along with Java for coding the front end part. Python was used to do exploratory data analysis. We would like to thank the professor for motivating us to work on such an interesting project through which we gained valuable insights in Scala and Spark.

# References

[1] "https://www.bts.gov/," January 2017.

[2] "https://spark.apache.org/docs/1.6.0/sql-programming-guide.html," January 2017.

[3] "https://www.meta-chart.com/pie#/data," March 2017.

[4] "http://spark.apache.org/docs/latest/ml-guide.html."