Flight Delay Prediction
Project Report

Anna Martignano & Muhammad Hamza Malik
Data-Intensive Computing H19-P1

October 27, 2019

1 Dataset

The dataset used to create a prediction model is based on the information made available by the Bureau of Transportations Statistics to analyze past flights records. The column available in the dataset are the following:

- DepTime - actual departure time (local time: hhmm)
- CRSDepTime - CRS departure time (local time: hhmm)
- ArrTime - actual arrival time (localtime: hhmm)
- CRSArrTime - CRS arrival time (localtime: hhmm)
- UniqueCarrier - identification number assigned by US DOT to identify a unique airline
- FlightNum - flight identifier
- TailNum - aircraft registration number
- ActualElapsedTime - actual elapsed time of flight (minutes)
- CRSElapsedTime - CRS elapsed time of flight (minutes)
- AirTime - elapsed time of flight (minutes)
- ArrDelay - difference between scheduled and actual arrival time (minutes), early arrivals show negative numbers
- DepDelay - difference between scheduled and actual departure time (minutes), early departures show negative numbers
- Origin - origin airport
- Dest - destination airport

- Distance - distance between airports (miles)
- TaxiIn - taxi out time (minutes)
- TaxiOut - taxi in time (minutes)
- Cancelled - cancelled flight indicator (binary, 1=yes)
- CancellationCode - specifies the reason for cancellation
- Diverted - diverted flight indicator (binary, 1= yes)
- CarrierDelay—WeatherDelay—NASDelay—SecurityDelay—LateAircraftDelay - delay classification (all of them are expressed in minutes)

2 Method

A detailed description of all the steps undertaken is provided in the following section.

2.1 CLEANING

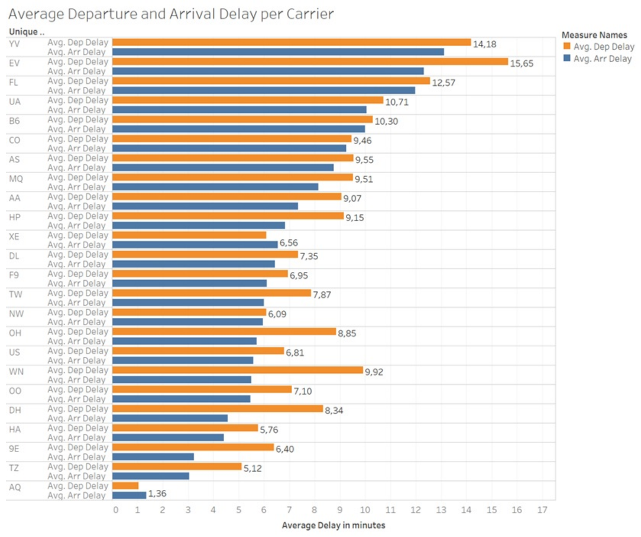
The original dataset contained lots of columns with garbage data, i.e. null values and not alphanumeric characters, so we performed data cleaning operations in R studio. Below the operations performed are explained:

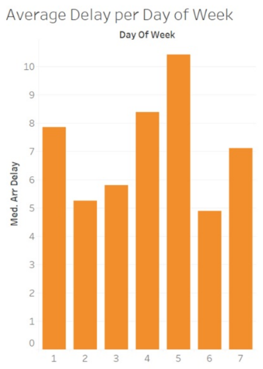
- Replacement of null values with 0's since for some columns we have to process data which is of datatype Int.
- Pruning of entries which contained not alphanumeric characters.
- Replacement of special characters or strings with correct ones, e.g. not available field (NA) replaced with 0's, etc.

After performing the above-mentioned steps, we wrote back the cleaned data frame into a csv file. So, we used this new csv file as input for processing and building up the model.

2.2 PREPROCESSING

The csv was loaded into a Cassandra table since our application establishes a connection with Cassandra and retrieve the information from it. The advantage of using a database is due to the fact that the schema is already defined in Cassandra, then the importing of the rows is expedited. In this way, the dataset is ready to be processed. We have used as support for the following steps the visualization of the cleaned dataset both in SparkSQL and Tableau. Here there are reported two examples of the view extracted by the Dataset. From these two charts we can already notice that the average arrival delay per day of week (1="Monday") significantly change, and in the second charts we can notice a trend in which usually the Arrival Delay decreases respect to its corresponding Delay.



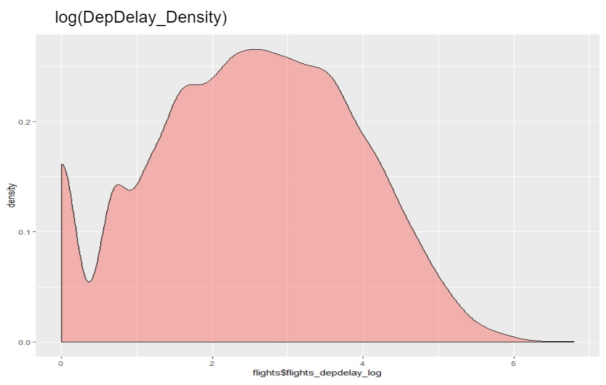


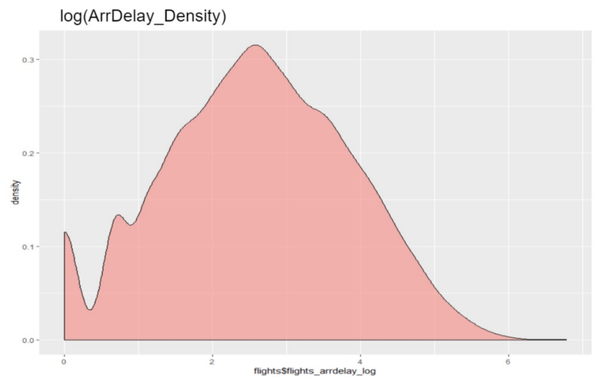
2.3 VARIABLES SELECTION

In order to determine which variables are the best one to explain the arrival delay time, a Goodman Kruskal correlation matrix among the quantitative variables was considered inside the R studio code snippet. In addition, we performed a Chi-square test to get the p values for different variable against the target variable. The outcome of these tests indicates that DepDelay and DayofWeek are the most related variables. Indeed, they have the lowest p-values, i.e. less than 0.05, and the highest correlation values. Besides these two main variables, also the following were added to the model to improve the performance prediction: Month, Distance, CRSElapsedTime, TaxiOut, and of course ArrivalDelay since it is used as target variable.

2.4 TRANSFORMATION

In any linear regression model, it is important to study variables distribution, or an approximation of them, it is desired in order to define a good model. Taking this into account, the distributions of the target variable (ArrDelay) and one of the predicting variables (DepDelay), which was chosen since its correlation to the target is maximum, were plotted and their distributions examined. When we plotted density graphs of both variables, we found right skewness, indicating that most of the values oscillate close to values around to 0, but there are some high values that, while not frequent, are important to take into account. The best thing to fix this problem is to apply a logarithmic transformation to the variables. In this way, values that are too big will become smaller and get closer to the most frequent ones. Below are the results after applying log transformation.
Looking at the above plots, it can be seen that this transformation improves





the right skewness of the variables.

2.5 MACHINE LEARNING MODEL

The machine learning model selected is linear regression. Linear regression can be described as an approach for modelling the relationship between a scalar dependent variable, in this case arrival delay time, and one or more explanatory variables. The case of one explanatory variable (independent variable) is called simple linear regression. For more than one explanatory variables (independent variable), the process is called Multiple Linear Regression, which will be the one to use considering we have many variables that could be useful to predict the arrival delay time.

3 Results

3.1 MODEL EVALUATION IN SPARK

In order to evaluate our model we considered the following measures:

- **RMSE** - Also known as Root Mean Square Error, is a measure of the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed, which means that the closer this result is to 0, the better. RMSE in our case is 1.6512 which is a little higher. So, it means we probably need more variables for prediction that are more significant and more related to the targeted variable.

- **R2** - Which stands for R2, is a measure that, when the square root is applied to it, indicates the total proportion of variance explained by the model. R2 in our case is 0.5770, which implies that $\sqrt{R2}$ = R = 0.758 and that approximately 76% of the variance present in the data is being explained by the model.

The dataset chosen in order to build our model is around 3.2 MB which includes around 30,000 flights. We believe that this dataset is good enough to build our model in our local machine. Later on, once the development of the model was completed, we also performed the model evaluation on a cluster in Databricks. Therefore, exploiting the full potential of spark with a much bigger dataset of 700 MB. For processing this big amount of data, we first cleaned the file using StringReplacer.c instead of CleaningAndAnalysis.R, since the performance of C programming language permits to replace the content of very large files very fast. Then we loaded the cleaned data into Cassandra in our local machine. On the local machine, the dataset retrieved by Cassandra was truncated by Jupyter Notebook to 200,000 records from 7 million. However, size is still much bigger than previous dataset, i.e 22.4 MB. Instead, databricks cluster was able to handle all the dataset and slowest operations was the model fitting which required almost 2 minutes. Below are the results of model evaluation obtained by processing the entire big dataset on Databricks:

- **RMSE** - 1.6187 which is slightly better than the previous result.

- **R2** - 0.6111 which is a slightly better too than the previous result.

Taking into account the results obtained, the first thing to say about the model is that it is not the best model to predict the arrival delay time, since the RMSE returned, while closer to 0, it is still a little bit high to be considered a great one. By taking into account that we only have two significant attributes in dataset the obtained results are definitely quite good and make the chosen model a good one. However, this model still can give better results in terms of RMSE and R2 if we have more significant attributes in our datasets that can predict target attribute.

3.2 MODEL VISUALIZATION

It is very interesting to notice from the comparison of the two models that in both datasets the pattern is the same. The x-axis is represented by the predicted

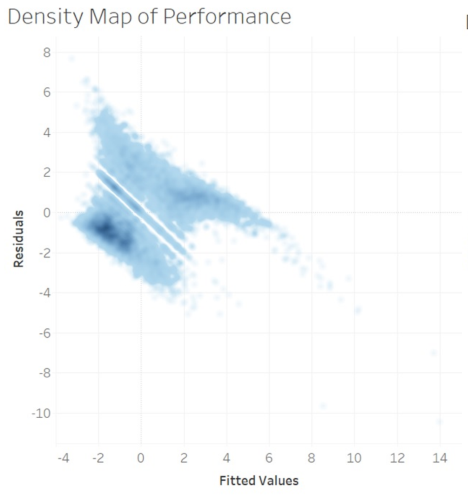


Figure 1: 3.2 MB Dataset

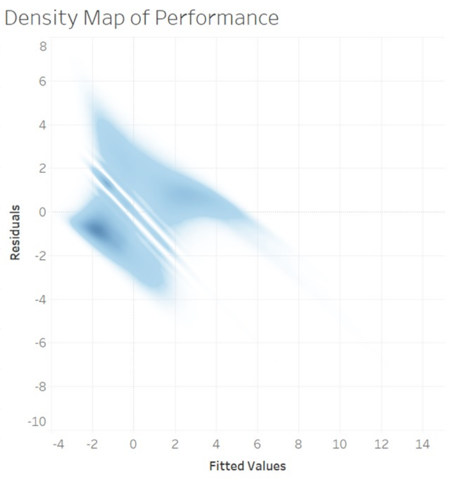


Figure 2: 0.7 GB Dataset

values obtained by the modelling of the datasets and the y-axis is the difference between predicted values and the real ones. The points are concentrated in the area of -2 to 6 as predicted values with a residual which does not exceed in general an error of ± 2. This confirms our hypothesis that even the first dataset selected is still enough populated to perform the prediction.

4 How to run the code

4.1 IMPORT DATA SOURCE - CASSANDRA

First of all, it is necessary to upload the dataset of the flights information into a Cassandra table since the application is going to retrieve the flight information from there for creating the pipeline. To run the followin instructions, a previous installation and configuration of Cassandra is assumed.
In a terminal window we are going to run the Cassandra server, to let the Spark open a connection with Cassandra.

```
$CASSANDRA_HOME/bin/cassandra -f
```

In an other terminal window, open the Cassandra bash to create the keyspace and the table where the information are going to be inserted.

```
$CASSANDRA_HOME/bin/cqlsh
```

Before creating the *flightspace* key_space and the *flights* table, it is essential to check that the csv file *flights.csv* is in the same location of the current path specified in the terminal window.

```
create keyspace flightspace with replication = {'class': '
    ↪ SimpleStrategy', 'replication_factor': 1};
use flightspace;
create table flights (Year int, Month int, DayofMonth int,
    ↪ DayofWeek int, DepTime int, CRSDepTime int, ArrTime int,
    ↪ CRSArrTime int, UniqueCarrier varchar, FlightNum int,
    ↪ TailNum varchar, ActualElapsedTime int, CRSElapsedTime int
    ↪ , AirTime int, ArrDelay int, DepDelay int, Origin varchar,
    ↪  Dest varchar, Distance int, TaxiIn int, TaxiOut int,
    ↪ Cancelled int, CancellationCode int, Diverted int,
    ↪ CarrierDelay int, WeatherDelay int, NASDelay int,
    ↪ SecurityDelay int, LateAircraftDelay int, PRIMARY KEY(
    ↪ FlightNum));
copy flightspace.flights (Year, Month, DayofMonth, DayofWeek,
    ↪ DepTime, CRSDepTime, ArrTime, CRSArrTime, UniqueCarrier,
    ↪ FlightNum, TailNum, ActualElapsedTime, CRSElapsedTime,
    ↪ AirTime, ArrDelay, DepDelay, Origin, Dest, Distance,
    ↪ TaxiIn, TaxiOut, Cancelled, CancellationCode, Diverted,
    ↪ CarrierDelay, WeatherDelay, NASDelay, SecurityDelay,
    ↪ LateAircraftDelay) from 'flights.csv' with delimiter=','
    ↪ and header=true;
select * from flights limit 100;
```

4.2 CASSANDRA AND JUPYTER NOTEBOOKS

To run the notebook on the local machine, on Jupyter Notebook, the following line must be appended into spark-defaults.conf file in the $SPARK_HOME/conf path. In this way, it is possible to create a connector among Spark and Cassandra.

```
spark.jars.packages com.datastax.spark:spark-cassandra-
    ↪ connector_2.11:2.3.2
#please note that you eventually need to specify different spark/
    ↪ cassandra version numbers
```

4.3 CASSANDRA AND DATABRICKS

To set up the connection among Cassandra and Databricks, it is simply necessary to have the Cassandra server operative and managing the cluster configuration on Databricks. In particular the security configuration must be changed, cluster should be able to listen to port 9042 as well, since this port is used by Cassandra to send the information. Then it must be as well installed on the used cluster the Maven library with the latest spark-cassandra connector. Then, it is possible to run both the Scala notebooks.

4.4 TABLEAU AND DATABRICKS

To perform the data analysis we have used as well Tableau, since it is a powerful visualization tool. The set up the connection we have used as a reference the guideline provided in Databricks documentation page, installing as well ODBC Simba. For such visualization the FlightsDatasetAnalysis notebook has been used to import the data. A temporary view of the dataframe has been created to perform the queries, and as well the dataframe has been saved as a table to be accessed by Tableau.

5 Final Conclusion and Personal Remarks

This deliverable has been useful to get some of the first impressions when working with large amounts of data and technologies that allow it. When trying to run with datasets of heavy weight in the local machine, one could see the limitations in terms of speed and processing. After trying the code out by using databricks spark cluster, a great difference could be noticed when running the code. Also we figured out that data cleaning and preprocessing are one of the most critical processes when dealing with big datasets before the execution of model and we had an opportunity to learn different techniques to clean the data in this project.