Commercial flights delay prediction

Máster Universitario en Ingeniería Informática
Universidad Politécnica de Madrid

José Domínguez Pérez
Ismael Muñoz Aztout
Jonatan Ruedas Mora

December 2020

Contents

1   Introduction

For this analysis we have developed two projects.

- The first one has been developed in the R programming language. In this project we perform an exploratory data analysis to investigate which variables are related to our target variable. This project contains the source code in which we carry out the processing and visualization of the dataset and a folder called plots where the plots of the analysis are saved.

- The second one is the Spark project that is requested in this practical work. It is written in Scala and is the project where we load and process the data to create a machine learning model that can predict the delays of commercial flights. The project follows the structure of a Maven project.

2   Analysis

Once we have explained how our practical work is structured, we will explain how we carry out the exploratory analysis of the data to select the useful variables.

To do this, we cleaned several values of the dataset like the forbidden variables and the variables that we have considered useless (*Cancelled* and *CancellationCode*). We also have created a new column called *Date* that merges the *Year*, *Month* and *DayofMonth* variables. Later, we have factorized the categorical values in order to prepare them to generate the correlation matrix. Apart from that, instead of omitting the NA values we gave a value of 0 to all the missing values because in some datasets (i.e., 1987) there were some related columns that were full of them.

When we were sure that the data was correctly cleaned we proceeded to generate the correlation matrix in which we could see which variables were strongly correlated with our target variable. As a result, we decided to use only the 2 most correlated variables because the other correlation values were quite low.
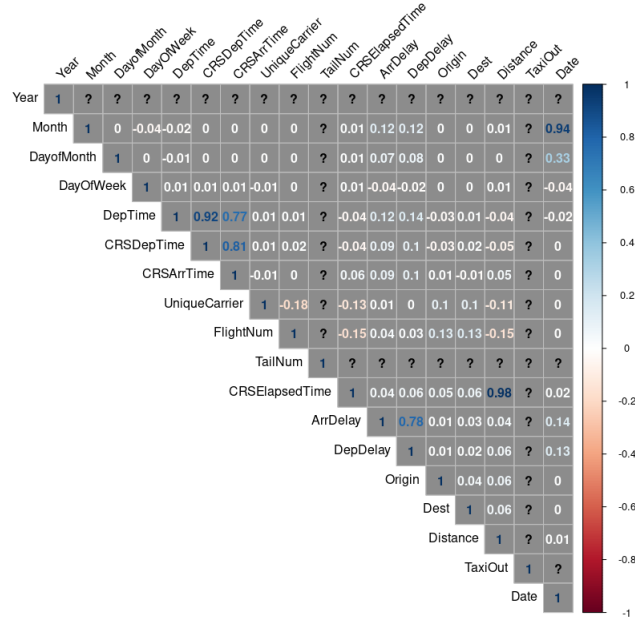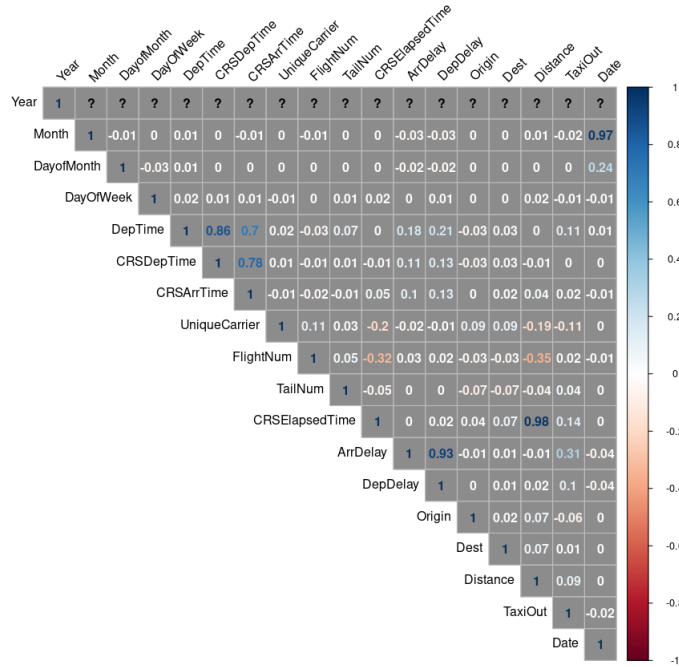
Figure 1: 1987 correlation matrix

Figure 2: 2008 correlation matrix

After that, we generated a scatterplot for the 2 selected variables to see how they correlate with the target variable. We realized that in some datasets, like 1987, the

TaxiOut variable was full of NA values but when it does have value it influences the ArrDelay so we decided to keep that variable and use it to build the model. The figure 4 shows that graphic because we used 0 to replace NA and the column only had missing values.
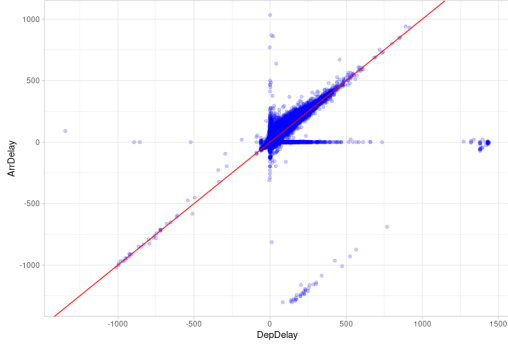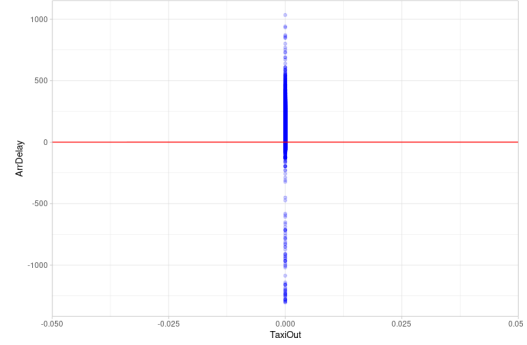

Figure 3: 1987 ArrDelay vs DepDelay

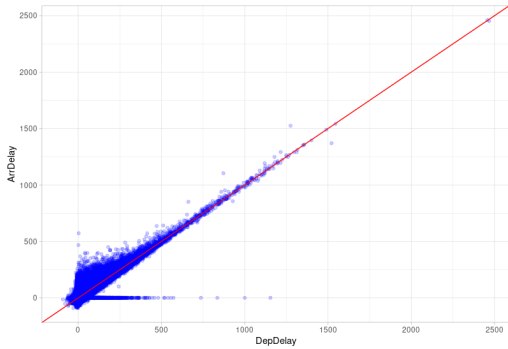Figure 4: 1987 ArrDelay vs TaxiOut

Figure 5: 2008 ArrDelay vs DepDelay
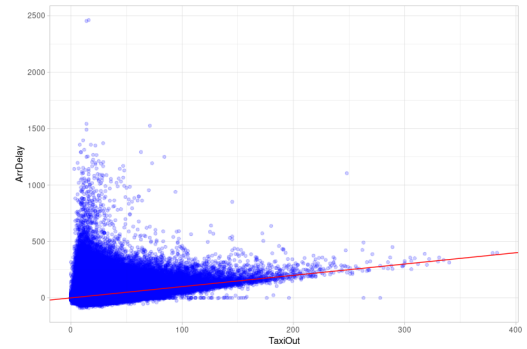
Figure 6: 2008 ArrDelay vs TaxiOut

2.1   Insights

After carrying out different tests with different datasets, we decided to use the DepDelay and TaxiOut variables in order to make our prediction for the aforementioned reasons, discarding the rest of the variables as they have a very low correlation index or they were not correlated at all.

3   Variable selection

3.1   Selected variables

We have selected the following variables:

- *DepDelay*

- *TaxiOut*

3.2   Excluded variables

Forbidden variables:

- *Arrtime*

- *ActualElapsedtime*

- *AirTime*

- *TaxiIn*

- *Diverted*

- *CarrierDelay*

- *WeatherDelay*

- *NASDelay*

- *SecurityDelay*

- *LateAircraftDelay*

Categorical variables have been factorized in order to have numerical variables. These variables are:

- *UniqueCarrier*

- *TailNum*

- *Origin*

- *Dest*

The *Cancelled* and *CancellationCode* variables has been excluded because when a flight is cancelled, it will not arrive to the destination and it will be useless to predict arrival delay.

When cleaning the dataset in Scala, we remove all of the forbidden variables and what we call *useless* variables that are the ones that have a weak level of correlation (under 0.25 more or less) with the target variable.

3.3   Transformed variables

We have transformed the categorical variables in our dataset into numbers by factorizing them. This way, we can assign them a numerical value in order to perform calculations such as correlation.

3.4   Created variables

We have created a new column called *Date* that merges the *Year*, *Month* and *DayofMonth* variables but we discarded later because it was not correlated to the target variable.

4   Machine learning model

4.1   Selected techniques

First of all, our target variable is continuous and we have its actual value (supervised learning), so we have selected some regression algorithms. These algorithms are as follows:

- **Linear Regression**, which is a basic and commonly used type of predictive analysis and the main reasons why we have chosen this algorithm are:

  - Provides good predictive results of the target variable with respect to the set of predictor variables.
  - It is one of the best known algorithms and we are familiar with it after having used it in various subjects.

- **Gradient Boosted Tree**, which is one of the most powerful techniques for building predictive models. BTs iteratively train decision trees in order to minimize a loss function. Like decision trees, GBTs handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and are able to capture non-linearities and feature interactions. The main reason why we have chosen this algorithm is:

  - To be able to compare the results of both algorithms, where it can be verified that the linear regression algorithm is much better than GBT.

4.2   Validation process

The process of validation we have used in order to measure the accuracy of the selected machine learning algorithm is as follows:

Firstly, we have divided our dataset into two parts:

- The **training set**: used to train our data.

- The **test set**: used to assess the actual performance of the chosen algorithm. When training the dataset, we are provided with different metrics that can be used to validate our model. However, these metrics are calculated based on the training data and we need to ensure that this algorithm is also accurate with respect to independent data that has not been used in the training step. This is why we keep a part of the dataset that has not been used for training.

Secondly, apart from splitting out the dataset into the training and test dataset, we also divide the training part into different validation sets in order to conduct cross-validation. This validation process is used to find the best parameters for our machine learning algorithm, which is also called hyper-parameter tuning. This way, we can select the best parameters for our model before assessing its performance with the test data. More specifically, we are going to make use of the K-fold cross-validation, where training data is randomly partitioned into K equal sized subsets that are used to validate the remaining training data. Next, we will show an illustration to clarify the whole process of validation we have implemented. In the figure 7, a K-fold cross-validation of K=5 has been used. We have used the mininum recommended value for that parameter, K=3
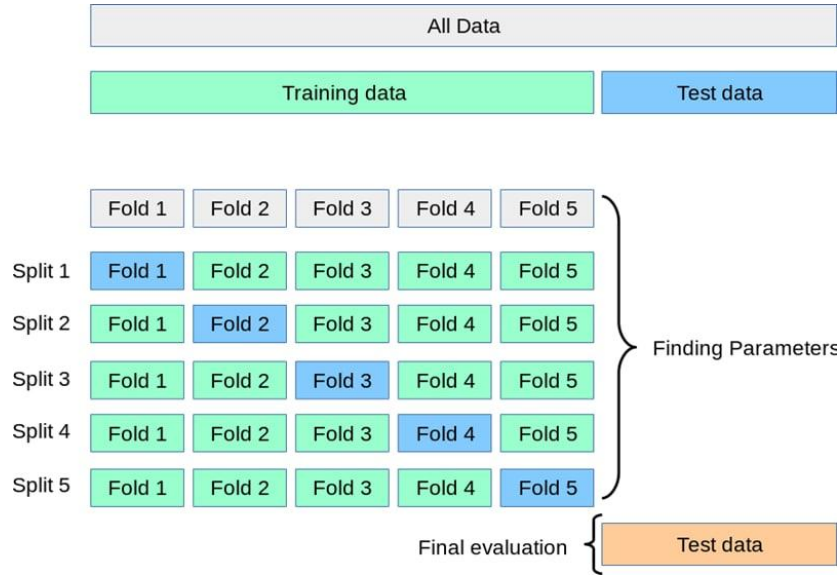

Figure 7: Validation process

Once we have found the parameters that best fit the model, we can use these parameters to assess the actual accuracy of our model by evaluating it with the test set. Next, we will discuss the metrics we have used to assess the accuracy of the models. As we have implemented regression algorithms in order to predict the target variable (since this variable is continuous), we have used the following regression metrics:

- Mean Squared Error (**MSE**): this metric measures the average squared difference between the estimated and actual values.

- Root Mean Squared Error (**RMSE**): this metric measures the average difference between the estimated and actual values.

- Mean Absolute Error (**MAE**): this metric measures the average difference between the estimated and actual values, without considering their directions.

- Coefficient of determination ($\mathbf{R^2}$): this metric measures "how well" our predictor variables explain the target variable. The values of this metric ranges from 0 (when predictor variables cannot explain the target variable) to 1 (when predictor variables explain the target variable at its best).

Finally, we are going to discuss what regression metrics we have chosen and why.

Firstly, the coefficient of determination is a very useful metric when assessing the accuracy of a model due to the fact that the value of this coefficient always ranges from 0 to 1, regardless the type of algorithm. However, it does not provide us with an idea of the **magnitude** of the error. This is why we have discarded this metric.

Then, we have the first 3 metric errors, which are very similar metrics that tell us the magnitude of the error. The difference between these metrics is that the MSE provide us with the average squared error instead of the average error, so we can also discard it. Now we have to choose between the RMSE and the MAE. The main difference between these last two metrics is that the RMSE gives a higher weight to large errors, which are commonly undesirable. So this is why we have chosen the **RMSE** metric.

4.3   Model evaluation

The results that we have obtained using the dataset of the year 2008 are these.

Figure 8: Linear Regression result

Figure 9: Gradient Boosted Tree result

As we can see, the linear regression algorithm is much more precise than the GBT algorithm due to the fact that the RMSE for the linear regression is almost half of the GBT. Also, we can see that all of the others metrics are better for the linear regression model.

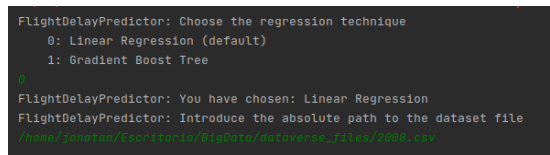Therefore, we can conclude that it would be much more optimal to use the linear regression to deduce our target

5   Execution instructions

5.1   Input data

You have to provide the full path where your dataset is stored or the folder with all the datasets.

5.2   ML technique selection

We have 2 possible techniques to perform the model construction, Linear Regression and Gradient Boosted Tree. You have to select one of those techniques using the keyboard numbers 0 or 1. By default Linear Regression will be used.


```
FlightDelayPredictor: Choose the regression technique
    0: Linear Regression (default)
    1: Gradient Boost Tree
0
FlightDelayPredictor: You have chosen: Linear Regression
FlightDelayPredictor: Introduce the absolute path to the dataset file
/home/jonatan/Escritorio/BigData/dataverse_files/2008.csv
```

Figure 10: App start parameters

5.3   Error handling

We check some errors like empty folders, not *.csv* file or wrong technique selection, those errors are managed through exceptions and terminate the execution of the program.

6   Final conclusions

We found this project very interesting, we really enjoyed the Scala programming and Spark documentation was really well written so it was easy to find out how to solve the problems. The main problem was the statistical aspect because we do not have advanced background about this field, even though, this project has been educative and interesting.