Assignment on Big Data, Spark Programming
Students: Filip Finfando, Michail Gongolidis
Date: 2019-01-25

Problem characterization

The objective of this report is to present the way we created the model that predicts the arrival delay time of a commercial flight, given a set of parameters known at time of the take-off. We tried to solve this prediction problem by analyzing the flight data set provided by the US Department of Transportation. The report is divided into the following parts:

1. Data: short explanation of our exploration,cleaning and formatting of the data set.In this section we explain which variables were excluded and included in the model.
2. Model: description of the machine learning technique used with the evaluation metrics and results
3. Instructions: instructions on how to easily use and test the application
4. Summary: conclusions

Data

All the data exploration and analysis was done in the Rstudio environment.We loaded parts of the dataset to test them, plot them and detect any outliers or corellations.

Removed variables and observations

Firstly, flights that were cancelled or diverted were removed from the dataset (Cancelled = 1 or Diverted=1). Such flights are irreleveant for the task of predicting the delay of the particular flight. Variables related to Divertion or Cancellation were removed from the dataset, as they do not carry any useful information for our task.

Secondly, columns containing data available only after the departure of the flight were dropped. Those are known as "forbidden columns" and they are not available at the moment of prediction.

Thirdly, columns "TailNum", "FlightNum", "Year", "DayOfMonth", "Origin", "Dest", "CRSElapsedTime", "CRSArrTime" were dropped as they contain too many categories or do not provide useful information with respect to "ArrDelay" variable.

Next the dataset was checked for outliers. In order to improve predictions it was decided to remove top 1% and bottom 1% of extreme observations with respect to ArrDelay variable from training data set.

In the end all rows with null values are dropped. However during testing no such rows were identified after preprocessing.

Exploration

We wanted to visually explore the data in order to find out which variables could be good predictors of delay on arrival (ArrDelay). Out of all variables the one describing delay at departure (DepDelay) seems to describe the arrival in the best way. The relation between variable is very linear. This is in line with our intuition as the duration of the flight rarely changes and most delays happen either at the origin airport or because of the bad weather conditions in the destination. Figure 1 shows the relation between DepDelay and ArrDelay variables.

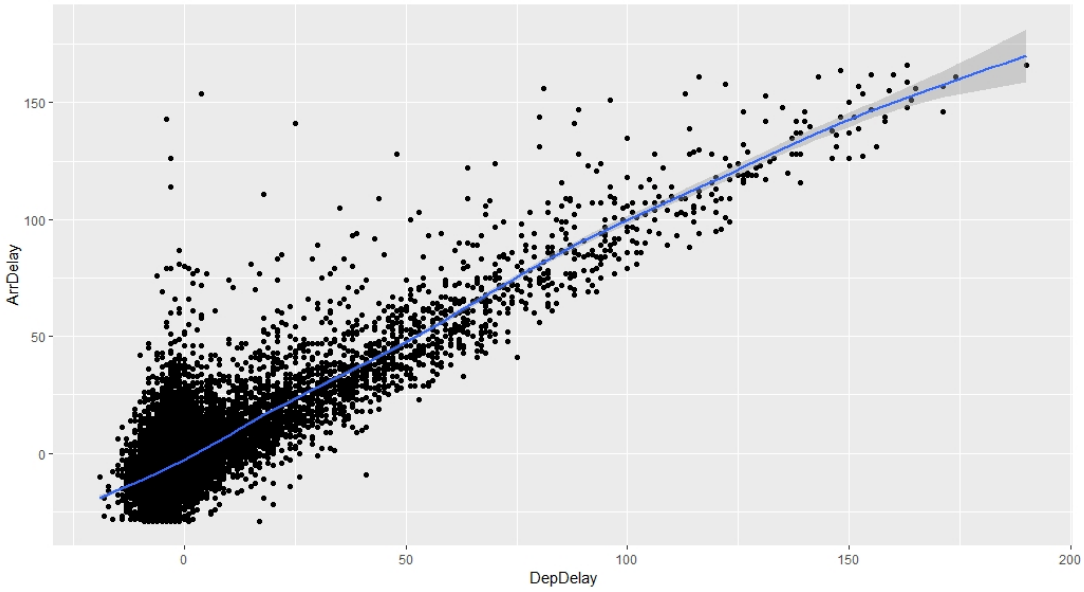

Figure 1: Scatter plot of DepDelay and ArrDelay variables

We expected also significant difference in delays between flights scheduled during peak hours and off-peak hours. We extracted hour from scheduled departure time and visualized it using boxplot presented on figure 2. We can see that the delays of the flights

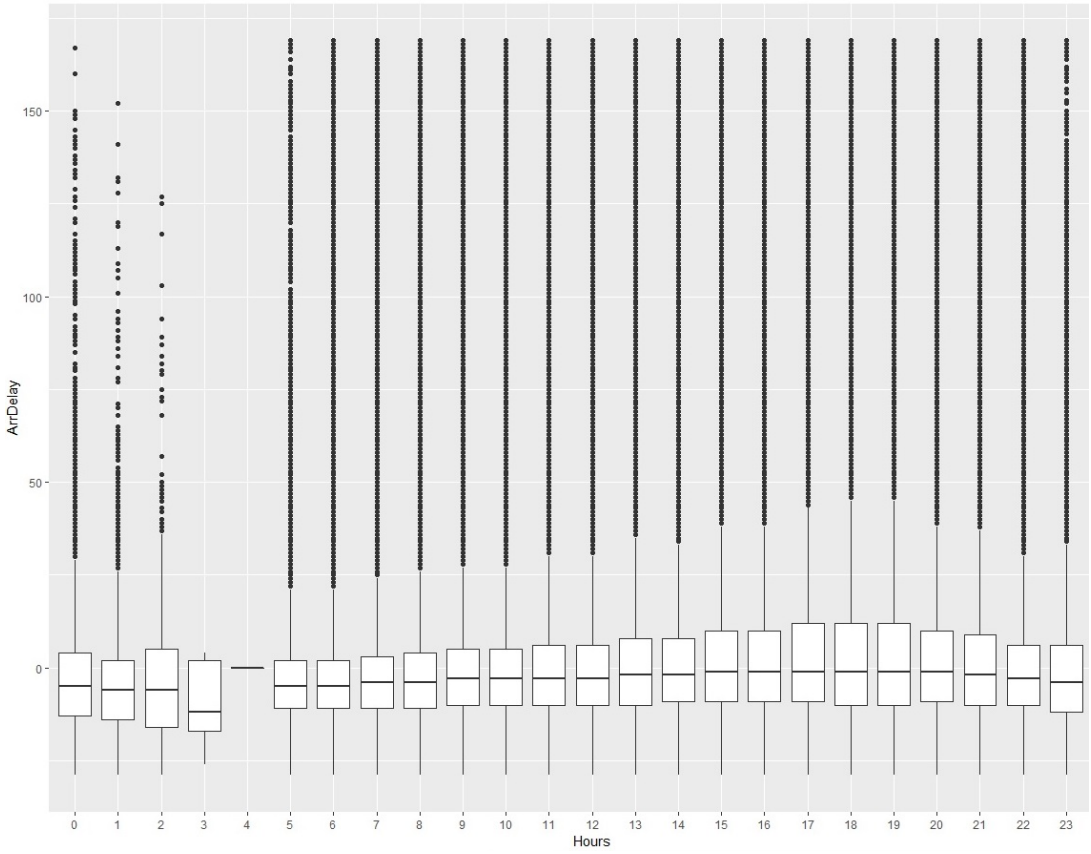

Figure 2: Boxplot of ArrDelay every Hour through the day

scheduled late in the night (or early in the morning) are different to those in the date time or early in the night. We have to state that in the above plot the outliers are already filtered out.

Selected Variables

In this subsection we state the variables we selected for the model and the reason for that decision.

Transformations and new creations

During the exploration phase we found out, that there is a significant difference of average arrival delay between flights at night and during the day. This is probably due to airports being less crowded and lower number of flights scheduled at night. In order to capture this information, a new variable "Hour" was created which extracted the hour from scheduled departure time of the plane. Also a binary variable "NightFlight" was created which is a binary variable active for flights departing between 11pm and 4am.

Variables "DepDelay", "Distance", "TaxiOut" and "ArrDelay" were casted from String to Double format as their nature is continuous.

Variables "Month", "DayOfWeek", "UniqueCarrier" were transformed into categorical variables using Indexer and OneHotEncoder.

Summary

During testing phase we found out that many of the variables are not good predictors of ArrDelay and do not improve the RMSE metric. In the end the following variables were included in the model:

- MonthVec - OneHotEncoded variable of Month
- DepDelay - Double
- CompanyVec - OneHotEncoded variable for Airline Companies
- TaxiOut - Double
- NightFlight - Binary

We used the Intellij enviroment and we implemented scala code to do the transformations in the model presented in the next section.

Model

For the purpose of the creation of the predictive model we decided to implement and compare 3 different machine learning algorithms included in the Apache MLib library: Linear Regression Model, the Gradient Boosted Tree Regression and the Random Forest Regression.

Evaluation and comparison

In order to evaluate and compare the performance of the models, the dataset was divided into training and testing set (70% and 30% of the observations respectively). After the division, we remove the 2% of the outliers of the total dataset only from the training set as mentioned before. The models were fitted on the training set and later evaluated on the testing set using multiple metrics. It was decided, that RMSE (Root Mean Squared Error) is going to be our key metric, as it penalizes very large errors and is indifferent to positive and negative errorrs.

Model selection

After careful analysis of obtained results models' parameteres were adjusted and the process of training was repeated until satisfying result was obtained. The final model chosen was: Linear regression with following parameters: setMaxIter(10), setElasticNet-Param(0.5). The RMSE on test predictions for the chosen model trained on data from year 2008 was: 10.71. It is slightly higher than RMSE on train data set, which means the model was fitted to the data well.

Instructions

In order to run the application one needs to

- install Apache Maven
- install Apache Spark
- have access to unzipped flights data in local file system

Go into app folder

```
cd path\to\app
```

Use Maven to build the JAR file

```
mvn package
```

Run the application using spark-submit

```
spark-submit --master local --class eit_group.App
.\target\eit_artifact-1.0-SNAPSHOT.jar
\path\to\input\data\file \path\to\output\folder
```

Summary

Overall, carrying out the project with Spark on Scala programming language was clear and fast. Spark has a great Machine Learning library and very good and understandable documentation. Scala is simple and easy to use. Finally, the assignment was clearly presented and explained and we did not stumbled upon any difficulty.

Attachments

Source code containing spark application is attached.

References

[1] https://spark.apache.org/docs/2.2.0/index.html