Big Data:
Spark Practical Work
First Semester
2024/2025

Table of Contents

Introduction

The objective of this work is to help students to put into practice the concepts learnt during the theory lessons, and to get proficiency in the use of Spark. In this exercise the students, **ORGANIZED IN GROUPS OF 3 PEOPLE**, are required to develop a machine learning model for a real-world problem, using real-world data they must predict the arrival delay of commercial flights.

The Problem

The basic problem of this exercise is to create a model capable of predicting the arrival delay time of a commercial flight, given a set of parameters known at time of take-off. To do that, students will use publicly available data from commercial USA domestic flights. The main result of this work will be a **Notebook** detailing the process followed to develop a ML model and a **Spark application** programmed to load the model and test its performance on unseen data.

The Data

For this exercise, students will use data published by the US Department of Transportation. This data can be downloaded from the following URL:

https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/HG7NV7

The dataset is divided into several independent files, to make download easier. You do not need to download and use the entire dataset. A small piece should be sufficient, one that fits in your development environment and does not take too long to process. The Spark application you develop, however, should be able to work with any subset of this dataset, and not be limited to a specific piece.

Forbidden variables

The dataset consists of a single table with 29 columns. Some of these columns must not be used, and therefore need to be filtered at the beginning of the analysis. These are:

- ArrTime
- ActualElapsedTime
- AirTime
- TaxiIn
- Diverted
- CarrierDelay
- WeatherDelay
- NASDelay
- SecurityDelay
- LateAircraftDelay

These variables contain information that is unknown at the time the plane takes off and, therefore, cannot be used in the prediction model.

Allowed variables

Any other variable present in the dataset, and not included in the previous list, can be used for the model.

Target variable

The target variable for the prediction model will be **ArrDelay**.

Exercise Requirements

Divided in **groups of three people**, students have to develop a Spark application capable of loading and processing the input data, and loading and validating the prediction model. Each one of these steps are described in detail in the following sections.

Exploring, analyzing and processing the data (notebook)

For this steps you should create a jupyter notebook to perform the following operations on data available in a directory called ../training_data:

- Selecting the variables that are going to be used.
- Transforming the variables that cannot be used in its raw form.
- Creating new, derived variables that could be helpful to create a better model.

To perform these operations, the students need to analyze and understand each of the variables involved. The dataset public documentation provides a lot of useful information that can help to design these steps. Some situations that may be encountered include:

- Several variables may not contain useful information or are forbidden.
- Several variables may contain information in a way that is difficult to understand/process. These need to be transformed into something meaningful.
- Several variables may provide better information when combined with others. In these cases, new variables could be derived from them.

All these decisions are left to the good judgment of the students. The target variable, ArrDelay, must not be modified during this process.

Creating the model (notebook)

After generating the definitive set of variables, the prediction model must be trained using ArrDelay as the target variable. The students can select any machine learning technique provided by the Spark API they wish to create this model. This selection must be properly justified in the report delivered.

Validating the model (notebook)

The model created must be validated, and some measure of its accuracy should be provided. The validation procedure and accuracy metric used is to be decided by the students. As in the previous case, the selection of the evaluation technique and accuracy measure must be sufficiently justified in the report. This validation should be done with the data available in the ../training_data directory.

Storing the model (notebook)

Finally, the model created must be stored in a file called best_model

Testing the model with unseen data (application)

Once you trained, validated and stored your best_model you should generate a Spark application that will load test data from an unknown location, this means the location of the test data will be passed as a parameter to the spark-submit script. The application should:

1. Load the test data from the location passed
2. Load your best_model
3. Perform some predictions on the test data
4. Perform a complete performance evaluation on the test data

Deliverables

As a result of this work, students are required to deliver:

- A jupyter notebook
- A Spark application with its dependencies
- A report

These three deliverables are now described.

Jupyter Notebook

The notebook should contain all the code used to:

- Load the training dataset
- Dataset exploration, analysis and processing
- Code used to train, test and save the model.

Spark application

The source code for the Spark application that:

- Loads test data from the specified path.
- Loads the best_model.
- Process the test data to be able to use the model.
- Performs some predictions
- Performs a complete performance test on the test data.

Application should be able to be executed with spark-submit.

Report

The report should be a short document, describing the work done. **No source code should be included in the report**. The document must contain the following information:

1. List of variables selected for the model, and the reason for selecting them.
2. List of variables excluded from the model, and the reason for not using them.
3. Detailed description and justification of the variable transformations performed.
4. Detailed description and justification of the new variables created.
5. Description of the machine learning technique selected and its parameters, explaining why this technique was selected.
6. Description of the validation process, including justification of the decisions taken.
7. Final evaluation of the prediction model, based on the validation results.
8. Final conclusions and personal remarks.
9. Instructions on how to pass the test data path to the application

Packaged deliverable

The work must be delivered in a **single ZIP file**, containing three elements:

- notebook.ipynb: A notebook containing dataset exploration, analysis and transformations.
- app.py: A python file containing the application that loads the best model and performs testing on the test dataset.
- best_model: a file containing the best trained model.
- report.pdf: A single pdf file with the report document.
- EXTRA: requirements.txt if you are using some external libraries.

The file must be delivered using moodle. **No email deliveries will be accepted**.

Grading

The work will be graded from 0 to 10 points. A minimum of 5 points is required to pass. This section details the grading criteria.

Basics

To obtain the minimum 5 points for passing, students must at least perform the following tasks:

- Read the input data file, correctly separating all given variables.
- Perform variable selection based on some basic analysis and logical criteria.
- Properly handle variable types (numerical, categorical, etc.).
- Select at least one appropriate machine learning algorithm for the problem and correctly train the model using basic train/test data split.
- Use an appropriate validation metric to measure the model performance.
- Use MLlib tools for handling the data, training, validating and saving the best model.
- Present code that works without human intervention/alteration, aside from running the program. Deployed with spark-submit.
- Write a clear and concise report, detailing all the reasoning process and decisions made.

Going further

Once the basics are covered, there are many things that can be done to raise the assignment grade, depending on the approach the students follow. One of the purposes of the assignment is to allow students to explore the Spark library and experiment with as many of the provided tools as possible. Here is a list of possible things that can be done to improve the quality of the work and obtain a very good or excellent grade. These aspects are not presented in any particular order (priority, importance, weight, etc.). Students should try to cover as many of them as possible if they intend to obtain a high grade. This is not an exhaustive list; new additions and other interesting improvements are welcome.

- Smart use of the Spark tools provided to properly read the input file and handle possible input errors (empty files, missing columns, wrong formats, etc.).
- Proper exploratory data analysis (possibly including univariate and/or multivariate analysis) to better understand the input data and provide robust criteria for variable selection.
- Smart handling of special format in some input variables, performing relevant processing and/or transformations.
- Feature engineering and exploring additional datasets to try to find additional relevant information.
- Select more than one valid machine learning algorithm for building the model.
- Perform model hyper-parameter tuning.
- Consider more than one possible model performance metric and explain the criteria for selecting the most appropriate.
- Use cross-validation techniques to select the best model.
- Use the full capacities of Spark's MLlib, including tools for cross-validation, hyper-parameter tuning, model evaluation, pipelines, etc.
- Write code that is properly commented, well structured, clean and easy to read.
- Create an application that can be used with different input files without having to change a single line of code, both for training and applying the model.
- Write a report that is both clear and interesting, including insightful analysis and conclusions.

Final remarks

The problem presented in this exercise is the construction of a prediction model. This is a typical Data Science problem, and the application requirements try to cover the basic aspects of most problems of this sort. **The focus of the students, however, should be on developing the necessary Big Data processes required to create this model, and not only on producing a high-quality predictor.** The problem presented (estimating arrival delay in commercial flights) is not a trivial one. **The key aspect here is the correct use of the Big Data technologies (Spark, in this case).**