Monash University

FIT5202 - Data processing for Big Data

Assignment 1: Analysing Flight-delays Data

Due Date : 5th September Sunday, 11.00pm

Weight : 10 percent of the total mark

Background

The flight-delays prediction dataset has become one popular dataset used by the aviation industry to predict the delay given the historical flight data. Learning from data can be beneficial for the companies, e.g. aviation industry, so that they can minimize the delay to improve customer satisfaction. The insight of the data can be obtained by conducting some steps, including pre-processing, visualization, and data modelling. In this assignment, we use Spark to visualize, and manipulate historical flight-delays data using Spark RDD and Spark SQL.

Required Datasets (available in Moodle):

- A compressed file **flight-delays.zip.**
- This zip file consists of two csv files and a metadata file:
  - o flights.csv
  - o airports.csv
  - o metadata.pdf
- The complete dataset also can be downloaded publicly at https://www.kaggle.com/usdot/flight-delays

Information on Dataset

The flight-delays and cancellation data was collected and published by the U.S. Department of Transportation’s (DOT) Bureau of Transportation Statistics. This data records the flights operated by large air carriers and tracks the on-time performance of domestic flights. This data summarises various flight information such as the number of on-time, delayed, cancelled, and diverted flights published in DOT's monthly in 2015.

Assignment Information

This assignment consists of two parts:

- **Part 1:** Work with RDDs in PySpark to implement specific queries related to flight delays data analysis.
- **Part 2:** Work with Dataframes in PySpark to implement specific queries related to flight delays data analysis.
- **Part 3:** You are required to implement a query and compare three different approaches: RDD, DataFrame, and SparkSQL.

Getting Started

- Download the datasets from Moodle namely **flight-delays.zip**.
- Download a template jupyter file namely ***Assignment-1.ipynb*.** This template is provided in Moodle and please put your answers according to the cells provided in the file.
- You will be using Python 3+ and PySpark 3+ for this assignment.

1. Working with RDD (45%)

In this section, you will need to create RDDs from the given datasets, perform partitioning in these RDDs and use various RDD operations to answer the queries for crash analysis.

1.1 Data Preparation and Loading (15%)

1. Write the code to create a *SparkContext* object using *SparkSession*, which tells Spark how to access a cluster. To create a *SparkSession* you first need to build a *SparkConf* object that contains information about your application. Give an appropriate name for your application and run Spark locally with as many working processors as logical cores on your machine.
2. Read the "flights.csv" file into a single RDD (flights_rdd) and "airports.csv" file into a single RDD (airports_rdd). For each RDD, remove the header row and parse each comma-delimited line into a *Row object* with each column following the data type given in the *jupyter notebook cell*. Please convert some columns into the preferred format. Columns that should be converted into integer : 'YEAR', 'MONTH', 'DAY','DAY_OF_WEEK', 'FLIGHT_NUMBER'. Column that should be converted

into float data type : 'DEPARTURE_DELAY', 'ARRIVAL_DELAY', 'TAXI_OUT', 'ELAPSED_TIME', 'AIR_TIME', 'DISTANCE', 'TAXI_IN', and 'TAXI_OUT'. While the rest are kept as string format. Note that in this preprocessing task, you are asked to build a set of functions which load the csv data into the RDD object, remove the header of the RDD object, and finally parse the RDD object into the desired format (integer, float, or string).
3. For each RDD, display the number of columns, the total number of records, and display the number of columns.

1.2 Dataset Partitioning in RDD (20%)

In this section, you will need to analyse the RDD partitions.

- How many partitions do the above RDDs have?
- How is the data in these RDDs partitioned by default, when we do not explicitly specify any partitioning strategy?

Once flights_rdd is created in 1.1.2, we note that the 'ARRIVAL_DELAY' column has been converted into a float data type. This column represents the gap between arrival time and the scheduled time represented in the column 'ARRIVAL_TIME' and 'SCHEDULED_ARRIVAL' respectively. Negative value means that the arrival time is earlier than scheduled time and vice versa.

Assuming we want to keep all the data related to **flight data** in one partition and the **airport data** in another partition, please do the following tasks.

1. Obtain the maximum arrival time using RDD from flights_rdd (could be a positive value).
2. Obtain the minimum arrival time using RDD from flights_rdd (could be a negative value).
3. Make a python function and define a hash partitioning function.
4. Write the code to obtain the number of records in each partition. Number of partitions can be defined manually. Once you display the *number of records in each partition*, please give an argument about it?

1.3 Query RDD (10%)

For the flights_rdd, write relevant RDD operations to answer the following queries.

1. Display the number of flights for each month
2. Display the average delay for each month

2. Working with DataFrames (35%)

In this section, you will need to load the given datasets into PySpark DataFrames and use *DataFrames functions* to answer the queries.

2.1 Data Preparation and Loading (5%)

1. Load all flights and airports data into two separate dataframes. Name the dataframes as *flightsDf* and *airportsDf* respectively. Hint : use the module *spark.read.format("csv")*, with header option is true and inferSchema is true.
2. Display the schema of the final of two dataframes.

2.2 Query (15%)

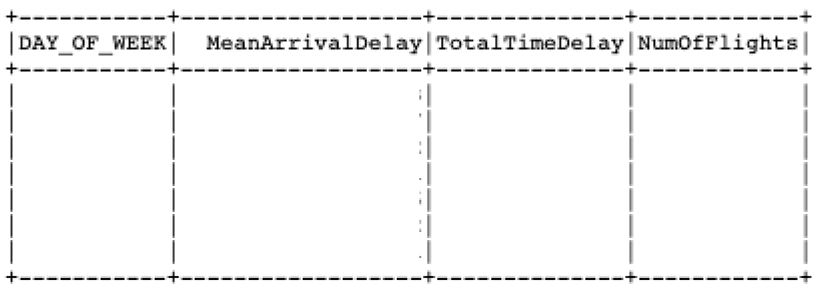
Implement the following queries using dataframes. You need to be able to perform operations like filtering, sorting, joining and group by using the functions provided by the DataFrame API.

1. Display all the flight events in January 2015 with five columns (Month, Origin Airport, Destination Airport, Distance, and Arrival Delay), where the origin airport 'ANC' and name this dataframe as *janFlightEventsAncDf*.
2. From the query results on query no.1, please display a new query. Then please group by 'ORIGIN_AIRPORT' AND 'DESTINATION_AIRPORT'. Add a new column and name it as 'AVERAGE_DELAY'. This column value is the average from all 'ARRIVAL_DELAY' values. Then sort it based on 'AVERAGE_DELAY'. Please name this dataframe as ***janFlightEventsAncAvgDf*.**
3. Join the results on query no. 2 *janFlightEventsAncAvgDf* and *airportsDf* using inner join operation. You may name this dataset as *joinedSqlDf*.

2.3 Analysis (15%)

In this section, we want to analyse when the delays are most likely to happen using Spark SQL. By obtaining the day of week and month in all history of flight, implement the following queries:

1. Find the total number of flights events, total time delay and average of arrival delay for each day of week ('DAY_OF_WEEK') sorted by the value of *NumOfFlights* in descending order. This query represents the summary of all 2015 flights. What can you analyse from this query results?



```
+-----------+-----------------+-------------+-----------+
|DAY_OF_WEEK|  MeanArrivalDelay|TotalTimeDelay|NumOfFlights|
+-----------+-----------------+-------------+-----------+
|           |                 |             |           |
|           |                 |             |           |
|           |                 |             |           |
|           |                 |             |           |
|           |                 |             |           |
|           |                 |             |           |
|           |                 |             |           |
+-----------+-----------------+-------------+-----------+
```

2. Find the average of arrival delay, total time delay, and total number of flight events for each month ('MONTH') sorted by *MeanArrivalDelay* in ascending order (default). What can you analyse from this query results?

3. Display the mean departure delay (*MeanDeptDelay*) and mean arrival delay (*MeanArrivalDelay*) for each month ('MONTH') sorted by *MeanDeptDelay* in descending order. What you can analyse from the relationship between two columns: Mean Departure Delay and Mean Arrival Delay?

3. RDDs vs DataFrame vs Spark SQL (20%)

In this part, you will be asked to compare between RDDs, DataFrame, and Spark SQL approaches to execute a query task below. Implement a query using RDDs, DataFrame and SparkSQL approaches separately.

Log the time taken for each query in each approach using the "%%time" built-in magic command in Jupyter Notebook and discuss the performance difference of these 3 approaches.

Note: Students could research and/or think of other ways to compare the performance of the 3 approaches rather than rely just on the "%%time" command.

The query task is as follows:

1. Find the MONTH and DAY_OF_WEEK, number of flights, and average delay where TAIL_NUMBER = 'N407AS'. Note number of flights and average delay should be aggregated separately. The average delay should be grouped by both MONTH and DAY_OF_WEEK.

Assignment Marking

The marking of this assignment is based on the quality of work that you have submitted rather than just quantity. The marking starts from zero and goes up based on the tasks you have successfully completed and its quality, for example how well the code submitted follows programming standards, code documentation, presentation of the assignment, readability of the code, reusability of the code, and organisation of code.

Submission Requirements

You should submit your final version of the assignment solution online via Moodle; You must submit the following:

- **A PDF** file (created from the notebook) to be submitted through Turnitin submission link. Use the browser's print function to save the notebook as PDF. Please name this pdf file based on your authcate name (e.g. **psan002.pdf**)
- **A zip file** of your Assignment 1 folder, named based on your authcate name (e.g. **psan002.zip**). This should be a ZIP file and **not** any other kind of compressed folder (e.g. .rar, .7zip, .tar). Please do not include the data files in the ZIP file. Your ZIP file should only contain Assignment-1.ipynb

Where to Get Help

You can ask questions about the assignment on the Assignments section in the Ed Forum accessible from the on the unit's Moodle Forum page. This is the preferred venue for assignment clarification-type questions. You should check this forum regularly, as the responses of the teaching staff are "official" and can constitute amendments or additions to the assignment specification. Also, you can visit the consultation sessions if the problem and the confusions are still not solved.

Plagiarism and Collusion

Plagiarism and collusion are serious academic offences at Monash University. Students must not share their work with any other students. Students should consult the policy linked below for more information.

https://www.monash.edu/students/academic/policies/academic-integrity

See also the links under *Academic Integrity Resources* in Assessments block on Moodle.

Students involved in collusion or plagiarism will be subject to disciplinary penalties, which can include:

- The work not being assessed
- A zero grade for the unit
- Suspension from the University
- Exclusion from the University

Late submissions

Late Assignments or extensions will not be accepted unless you submit a special consideration form. ALL Special Consideration, including within the semester, is now to be submitted centrally. This means that students MUST submit an online Special Consideration form via Monash Connect. For more details please refer to the **Unit Information** section in Moodle.

There is a **5% penalty per day including weekends** for the late submission.