Big Data Systems and Architectures - Spark Assignment 2021

Exploring International Flights in 2017 Data – Task 2

The requested task is to create reports on the average and median departure delays of all the airports and all the airways. Before starting the implementation, it is required to install numpy, spark, findspark and pyspark. Some packages such as SparkContext, SparkSession, SQLContext, DataFrameReader, StringType, Window should also be imported. It is important to run findspark.init(), so as to initialize the findspark.

A new entry point to Spark SQL is created by the use of SparkSession.builder with appName("FlightsAssignment"), as shown below:

After creating the temporary view, the data are loaded to a variable named `flights_data` by the use of spark.read() function with different options, such as that the dataset contains header and it is in comma delimited format.

Our scope is to omit the outliers in the columns ORIGIN and CARRIER. Firstly, the percent_rank() function is used from Window package, to calculate the rank of percentages in the count of column ORIGIN. This percent_rank id added as an extra column in the grouped ORIGIN and its count.

The datasets `flights_data`, which is the initial and the `percentiles_ap`, which contains the above calculations, are joined through column ORIGIN to one dataset named `flights_percentiles_ap`. In the initial dataset `flights_data` we filter and keep only the percent_ranks that are greater than 0.01, from the `flights_percentiles_ap`.

Subsequently, having omitted the outliers of column ORIGIN, we calculate the average delay per airport. To achieve that, a new dataset named ‘avgDelayPerAirport’ is created , having two columns. The name of each ORIGIN, as our data are grouped by it, and the average DEP_DELAY of each being round with two decimal digits and named with the alias AverageDelay. The data are sorted in descending order of the second column. The first 100 records of this outcome are exported to a csv named `task2-ap-avg` by the use of coalesce() function.

The next step is to calculate the median departure delay of each airport. A new dataset named `medDelayPerAirport` is created and contains two columns. The first columns contains the name of each ORIGIN, as our data are grouped by it, and the median, which is calculated by the use of percentile_approx() function applied in DEP_DEALY column with 0.5 parameter. The data are sorted in descending order of the second column named MedianDelay. The first 100 records of this outcome are exported to a csv named `task2-ap-med` by the use of coalesce() function.

Similar to the steps above, we calculated the average and median of each airway. The outliers in column CARRIER are omitted and in the initial dataset `flights_data` we keep only the percent_ranks that are greater than 0.01, from what have been calculated in the `flights_percentiles_aw`. Finally, the results of the average and the median at column CARRIER are exported in a csv named `task2-aw-avg` and `task2-aw-med` respectively.