THE UNIVERSITY OF SYDNEY

University of Sydney

DATA3404

Data Science Platforms

Group Assignment

Team members:
Charles Hyland 450411920
Yiran Jing 460244129
TODO: JAZLYN!!!

Semester 1, 2019

Contents

Job Design Documentation

Task 1: Top 3 Cessna Models

Aircrafts: A set of tuples with three fields(tail number, manufacturer, and model)
Flights: : A set of tuples with one field (tail number)

Before:

1. Join two data files using **join** function with join key tail number.
2. Apply **FilterFunction** to only return aircrafts with the manufacturer equaled to "CESSNA", and then use **project** function to project the "model" column only.
3. After that, apply **flatMap** with **CountFlightPerModel** object to produces one for each instance, also add "Cseena" to each instance and abstract the first three digits of each instance to fit the final output format.
4. Then, Group data by the different Cessna models using **groupBy** function, and for each group use **sum** to count all the instances of the same Cessna model.
5. Rank Cessna models in descending order using **sortPartition** function. And return the top 3 Cessna model by **first**.

After:

1. Apply **FilterFunction** to only return aircrafts with the manufacturer equaled to "CESSNA", and then use **project** function to project the tail number and model columns.
2. Join two data files using **join** with **broadcast hash** the filtered CESSNA model file and then project model column only.
3. Add **ReadFields** for the **CountFlightPerModel** object to specifies the model field is used to compute a result value. After that, apply **flatMap** with **CountFlightPerModel** object. And the following steps same as before.

Task 2 Average Departure Delay

Aircrafts: A set of tuples with one fields(tail number)
Flights: : A set of tuples with five field (carrier code, flight date, tail number, scheduled departure, actual departure)

Airlines: : A set of tuples with three field (carrier code, name, country)
A variable named *year* will save the user specific year. (We use 2004 to evaluate).

Before:

1. At the beginning, we filter airline dataset by **FilterFunction** to contain only US Airlines, and then **project** only two carrier code and name columns(delete "country" column) after this step.
2. Filter the specified year using 'flight date' field of Flights file, and then this field is removed.
3. Filter out non-delayed flights if actual departure time is not later than scheduled time, and filter out the cancelled flight by catch ParseException within **FilterFunction**
4. After that, apply **flatMap** with **TimeDifferenceMapper** object to calculate the actual delay time for each delay departure flight.
5. Join these three dataset to get the the*joinresult* dataset, project only two columns (airline name, length of delay time)
6. Apply **flatMap** with **NumMapper** object to produces one for each instance, then, group data by the different US airlines using **groupBy** function, and for each group use **sum** to count all the instances of the same US airlines, get the *joinresultNum* dataset
7. Then, Group *joinresult* data by the US airlines using **groupBy** function, and use **sum** to get the total length of delay time for each US airline. After that, join this dataset with *joinresultNum* get *joinresultNumSum* dataset.
8. Group *joinresult* data by the US airlines using **groupBy** function, and use **min** to get the min length of delay time for each US airline. After that, join this dataset with *joinresultNumSum* get *joinresultNumSumMin* dataset.
9. Group *joinresult* data by the US airlines using **groupBy** function, and use **max** to get the max length of delay time for each US airline. After that, join this dataset with *joinresultNumSumMin* get *joinresultNumSumMinMax* dataset.
10. Apply **flatMap** with **AvgMapper** object to get the average delay time for each US airline. Then Rank US airlines in alphabetical order by **sortPartition** function.

After:

1. step 1 to 4 same as before, but Rank US airlines in alphabetical order by **sortPartition** function before join.

2. To join two data files using **join** with **broadcast hash** the aircrafts file and the filtered US airlines file, then project only two columns (airline name, length of delay time)
3. After **groupBy** the US airline result, instand step 6 to 10, we apply **reduceMap** function with **Aggregation** function to count the number of delay and the average, min and max delay time for each US airline at the same time. And add **ForwardedFields** and **ReadFields** to this object.

Task 3: Most Popular Aircraft Types

Aircrafts: A set of tuples with three fields(tail number, manufacturer, model)
Flights: : A set of tuples with five field (carrier code, tail number)
Airlines: : A set of tuples with three field (carrier code, name, country)

Before:

1. We join the airlines dataset on the flights dataset based on the carrier code. Furthermore, we restrict the output to only include the airline name and the flight tail number fields.
2. We join the output of step 2 with the aircrafts dataset based on the tail number. Furthermore, we restrict the output to only include the airline name, flight tail number, aircraft manufacturer, and airline model fields.
3. We apply a **groupBy** function on the result of step 1 by the flight tail number. We then apply a **reduceGroup** function whereby we count the number of unique flight tail numbers and construct a new field with a count for each tail number to append. We then sort the data by airline name and the tail number count constructed.
4. We apply a **reduceGroup** function whereby we retrieve the top 5 aircraft type for each airlines.
5. We filtered the airlines dataset for flights based in the United States.
6. We apply a **reduceGroup** function on the output of the previous step to format the result needed for the output and sort the output by the airline name alphabetically.

After:

1. The steps are identical to before except we apply the airlines filter in step 5 to be the first step.

2. We apply a **broadcast hash join** in step 2 and 3 for reasons similar to task 2.