JIGSAW ACADEMY
THE ONLINE SCHOOL OF ANALYTICS

Case Study: PySpark – Practice (Non-Graded) Assignment

Broadcast Variable

Dataset:

We have a dataset of two tab delimited files – customers.tsv and salestxns.tsv, with the following columns.

customers.tsv: Contains customer details as below.

Customer Id	Name	City	State	Zip Code
11039	Mary Torres	Caguas	PR	725
5623	Jose Haley	Columbus	OH	43207
5829	Mary Smith	Houston	TX	77015

Salestxns. Tsv: Contains sales transaction details as follows:

Sales Txn Id	Category Id	Category Name	Product Id	Product Name	Price	Quantity	Customer Id
1	43	Camping & Hiking	957	Diamondback Women's Serene Classic Comfort Bi	299.98	1	11599
2	48	Water Sports	1073	Pelican Sunstream 100 Kayak	199.99	1	256
3	24	Women's Apparel	502	Nike Men's Dri-FIT Victory Golf Polo	50	5	256

Problem Statement:

From these two sets of data, we need to get a list of products purchased by the customers. Each record in the list should contain:

Customer Id	Customer Name	Product Id	Product Name	Price	Total Quantity	Total Amount

Note that the sales transaction files will have two records for the same product if a customer would have purchased a product two times in two separate transactions. So the Total Quantity field should have the sum of quantity from these transactions. And the total amount should have the total quantity multiplied by the price of the product.

Approach:

We have customer details in a file - customers.tsv and sales transactions details in another file - salestxn.tsv
Sales transactions contain only customer id but not name. Also the transactions may have multiple entries for a particular item if customer made separate transactions.
We have to generate output that contain customer id, customer name along with the product name, product id, price, total quantity ordered and total amount paid.
Since customer file is smaller (with 1244 records) and sales transactions file is larger (172198 records) we will put customer details in a variable and broadcast it across the cluster nodes. We will use this broadcast variable to add the customer name based on each customer id to the sales transactions.

Also in the application we need to aggregate the sales transactions of the same product in cases where a customer ordered as multiple transactions.

Solution:

Out PySpark application has the following sequence of steps:

- Read the input file customers.tsv to create the customer RDD. The RDD elements are the lines of type string from the input file.
- We now need to transform the RDD into a pairRDD having a key,value pair as the RDD elements with customer id as the key and customer name as the value.
- For this we are writing a function which performs the tasks below:
    - Splits the input string at the delimiter tab.
    - Then returns the string constructed as (customer id, customer name)
- We then use `map` lambda function to transform elements of customer RDD1 to key,value pair using the above function and generate customer RDD2 which has the above pairs as the elements.
- The RDD operation collectAsMap returns the pairRDD as a dictionary (referred to as Map in other programming languages).
- We will define a broadcast variable - custDictBroadcast and make our dictionary as the broadcast variable in our application.
- We then read the second input file salestxns.tsv to create the sales RDD.
- We now need to transform the RDD into a pairRDD having a key,value pair as the RDD elements with transaction details like (customer id, product id, product name, price) as the key and quantity purchased in the transaction as the value.
- For this we are writing a function which performs the tasks below:
    - Splits the input string at the delimiter tab.
    - Then returns the string constructed as (customer id, customer name)
- We then use `map` lambda function to transform elements of sales RDD1 to key,value pair using the above function and generate sales RDD2 which has the above pairs as the elements.
- We will use reduceByKey to sum up the quantities of same product bought by same customer in separate multiple transactions.
- Then we will map the key and value of the above RDD elements such that we will have customer id, product id, product name, price, total quantity purchased and lastly total amount paid (price * total qty).
- Once this is done we can use the broadcast variable to get the customer name for the corresponding customer id as per the requirement mentioned in the problem statement.
- We can save the final RDD using saveASTextFile function. It will create a directory with the given name and saves each partition of the RDD as a text file.

The `.ipynb` file containing code is given and can be run in Jupyter notebook.