Title: - Online Transaction Analysis

Project Author: - Sandeep Kumar Bhandoria

OBJECTIVE: I am a Hadoop Analyst in my company OTA Pvt Ltd. My company give the suggestions to other company who shares their transaction data with us.

This time they have given me the Online Transaction data of a company that is planning to surprise their customers for the events for Christmas and New Year and they also want to do some more suggestion so that they can make decision.

My objective is to analyse the data and come up which some use case and solution for that.

DATA WE HAVE:

1. Transaction's Data:

File Used: txns-large.dat

TransactionID	T_date	UserId	Price	Product_Cat	Product
00000000	06-26-2015	4000003	040.33	Exercise & Fitness	Cardio Machine Accessories
00000001	06-01-2015	4009775	005.58	Outdoor Recreation	Archery

2. Customer's Data:

File Used: Customer.dat

UserID	FirstName	LastName	Age	Profession
4000001	Kristina	Chug	55	Pilot
4000002	Paige	Chen	74	Teacher
4000003	Sherri	Melton	34	Firefighter
4000004	Karen	Puckett	74	Lawyer
4000005	Elsie	Hamilton	43	Pilot

TECHNOLOGY WE USED:

1) Apache Hadoop

2) Map Reduce programming in Java

SOFTWARE WE USED:

1) Virtual Box

2) Eclipse

3) Ubuntu

PROJECT DESCRIPTION:

Use case 1

Scenario: - Heavy price based transactions that company have.

1) We find all the transaction or products based on the user defined prices.

In the case we are expecting input from user for the value of amount on which we have to decide the transaction.

2) This can be used to find the transaction done on a specific price from where we can get products name that the users are interested in for a specific price.

Validation:

We have done a check on the user input before processing it further.

1) User can have to specify a minimum price and based on that price all transaction will be filter where price is greater than what user has specified.

2) If the user is passing String in place of number he/she will be displayed a message showing an error message to provide valid input and start the job again.

```
hduser@ubuntu64server:~$ hadoop jar CustomT1.jar /home/hduser/Transactional.dat /home/hduser/custom11
Use Case 1 : Finding the number where transaction amount is user-defined
Enter the minimum amount
he6
Please provide the amount as number. It mustn't contains any alphabets
hduser@ubuntu64server:~$
```

```
hduser@ubuntu64server:~$ hadoop jar CustomT1.jar /home/hduser/Transactional.dat /home/hduser/custom11
Use Case 1 : Finding the number where transaction amount is user-defined
Enter the minimum amount
he6
Please provide the amount as number. It mustn't contains any alphabets
hduser@ubuntu64server:~$ hadoop jar CustomT1.jar /home/hduser/Transactional.dat /home/hduser/custom12
Use Case 1 : Finding the number where transaction amount is user-defined
Enter the minimum amount
190
16/11/21 13:49:40 INFO client.RMProxy: Connecting to ResourceManager at /192.168.56.123:8032
```

Use case 2

Scenario: - Price Range Based Products

1) We will find the number of products we have for a particular range of price.

Validation:

1) We will be accepting user input for minimum and maximum limit for price.

2) The maximum price can't be less than minimum price we user pass inputs. The will be showed a message for this. And also tells the user to run the task again with proper inputs.

3) Minimum amount can't be less than 0. Message will be displayed for the same.

4) Maximum amount can't be less than 0. Message will be displayed for the same.

Output screenshot: -

```
hduser@ubuntu64server: ~
        at java.lang.reflect.Method.invoke(Method.java:497)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:221)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
hduser@ubuntu64server:~$ hadoop jar CustomT2.jar /home/hduser/Transactional.dat /home/hduser/hee
Use Case 1 : Count All the Transaction between a lower and upper amountt limit
==============================================================================
==============================================================================
Enter the minimum amount  145
Enter the maximum amount  156
16/11/21 11:11:50 INFO client.RMProxy: Connecting to ResourceManager at /192.168.56.123:8032
16/11/21 11:11:51 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Imp
lement the Tool interface and execute your application with ToolRunner to remedy this.
16/11/21 11:11:51 INFO input.FileInputFormat: Total input paths to process : 1
16/11/21 11:11:52 INFO mapreduce.JobSubmitter: number of splits:1
```

```
hduser@ubuntu64server: ~
                Reduce input groups=1
                Reduce shuffle bytes=36835
                Reduce input records=2833
                Reduce output records=1
                Spilled Records=5666
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=239
                CPU time spent (ms)=2260
                Physical memory (bytes) snapshot=300953600
                Virtual memory (bytes) snapshot=3754459136
                Total committed heap usage (bytes)=137498624
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=4418139
        File Output Format Counters
                Bytes Written=55
hduser@ubuntu64server:~$ hadoop fs -cat /home/hduser/hee/part-r-00000
Total number of transaction for your search are : 2833
```

Use case 3

Scenario: - Customers wise transaction and purchase

1) The Company is planning for a scheme to give offers to customers based on

a) Their past number of transactions.

b) Total purchase they did.

This requires an analysis to prepare a report per each user.

Output screenshot: -

```
hduser@ubuntu64server: ~
4009957 142.57   4
4009958 471.94   4
4009959 142.1   1
4009960 642.1100000000001   6
4009961 877.32   7
4009962 419.83   5
4009963 161.82999999999998   2
4009964 386.46999999999997   5
4009965 145.1   1
4009966 412.84   4
4009967 622.6800000000001   6
4009968 704.07   8
4009969 461.19   5
4009970 154.92   1
4009971 528.3399999999999   5
4009972 691.19   9
```

Use case 4

Scenario: - Monthly Wise Revenue

At the end of every year your company wants to do an analysis to know in which month people usually comes for shopping.

Output screenshot: -

```
hduser@ubuntu64server: ~
hduser@ubuntu64server:~$ hadoop jar CustomT4.jar /home/hduser/Transactional.dat /home/hduser/tpwo
Use Case 4 : Calculate total sales amt for each Month.
==========================================================================
==========================================================================
Enter the number of month for which you want to generate report
Enter 0 to generate report for each month
05
ddd
16/11/21 13:17:32 INFO client.RMProxy: Connecting to ResourceManager at /192.168.56.123:8032
16/11/21 13:17:33 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool i
erface and execute your application with ToolRunner to remedy this.
16/11/21 13:17:33 INFO input.FileInputFormat: Total input paths to process : 1
```

```
hduser@ubuntu64server: ~
                Combine input records=0
                Combine output records=0
                Reduce input groups=1
                Reduce shuffle bytes=54671
                Reduce input records=4205
                Reduce output records=1
                Spilled Records=8410
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=256
                CPU time spent (ms)=2850
                Physical memory (bytes) snapshot=297660416
                Virtual memory (bytes) snapshot=3754459136
                Total committed heap usage (bytes)=137498624
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=4418139
        File Output Format Counters
                Bytes Written=22
hduser@ubuntu64server:~$ hadoop fs -cat /home/hduser/tpwo/part-r-00000
05      432627.58000000013
hduser@ubuntu64server:~$
hduser@ubuntu64server:~$
```

Use Case 5

Scenario: - Monthly Wise Transaction Summary

Being a Hadoop Developer and Admin, You may need to partition your final data to make further processing easy.

We have been asked to divide all the transaction based on the month and store each transaction according to the months. So12 files are created for this one for each month.

```
hduser@ubuntu64server:~$ hadoop fs -la   /uio
-la: Unknown command
hduser@ubuntu64server:~$ hadoop fs -ls   /uio
Found 13 items
-rw-r--r--   1 hduser supergroup          0 2016-11-21 22:30 /uio/_SUCCESS
-rw-r--r--   1 hduser supergroup     377449 2016-11-21 22:28 /uio/part-r-00000
-rw-r--r--   1 hduser supergroup     339311 2016-11-21 22:28 /uio/part-r-00001
-rw-r--r--   1 hduser supergroup     385895 2016-11-21 22:28 /uio/part-r-00002
-rw-r--r--   1 hduser supergroup     368421 2016-11-21 22:28 /uio/part-r-00003
-rw-r--r--   1 hduser supergroup     371798 2016-11-21 22:28 /uio/part-r-00004
-rw-r--r--   1 hduser supergroup     368247 2016-11-21 22:28 /uio/part-r-00005
-rw-r--r--   1 hduser supergroup     375554 2016-11-21 22:29 /uio/part-r-00006
-rw-r--r--   1 hduser supergroup     374305 2016-11-21 22:29 /uio/part-r-00007
-rw-r--r--   1 hduser supergroup     367955 2016-11-21 22:29 /uio/part-r-00008
-rw-r--r--   1 hduser supergroup     368733 2016-11-21 22:29 /uio/part-r-00009
-rw-r--r--   1 hduser supergroup     353858 2016-11-21 22:29 /uio/part-r-00010
-rw-r--r--   1 hduser supergroup     366614 2016-11-21 22:29 /uio/part-r-00011
```

Use case 6

Scenario: -File sorting based on price

We have the transaction file and this file will be sorted based on the amounts available in each transaction.

Output screenshot: -

```
0007970,03-15-2011,4000156,199.94,Winter Sports,Snowshoeing,Montgomery,Alabama,credit
0017491,06-11-2011,4004350,199.94,Exercise & Fitness,Free Weights,Dayton,Ohio,credit
0042768,09-12-2011,4006767,199.96,Exercise & Fitness,Yoga & Pilates,Washington,District o
0032452,06-19-2011,4007666,199.97,Outdoor Recreation,Archery,Madison,Wisconsin,credit
0047835,10-17-2011,4003783,199.98,Outdoor Play Equipment,Sandboxes,Minneapolis,Minnesota,
0001263,08-31-2011,4001222,199.99,Winter Sports,Bobsledding,Columbus,Georgia,credit
0024867,11-01-2011,4009524,199.99,Water Sports,Kitesurfing,Boise,Idaho,credit
0031257,02-09-2011,4005726,199.99,Winter Sports,Bobsledding,Scottsdale,Arizona,credit
0036291,06-23-2011,4005620,200.00,Exercise & Fitness,Stopwatches,Gilbert,Arizona,credit
```

Use Case 7

Scenario: - Top profession who does shopping the most

1) Company wants to target the particular area where people are more interested in their products so we have analysed the top profession.

Output screenshot: -

```
hduser@ubuntu64server:~$ hadoop fs -cat /Olive30/part-r-00000
Pilot   1700.17
```

The customers who are pilot are doing more transactions.

Use Case 8

Scenario: -Analyze Top 3 customers to give additional rewards.

Our online shopping website wants to give rewards to some top 3 customers.

Output screenshot: -

```
hduser@ubuntu64server:~$ hadoop fs -cat /Olive31/part-r-00000
Karen   1080.42
Kristina        980.51
Elsie   719.66
```

Use Case 9

Scenario: - Month Wise top customer

1) We have analysed the data to get the top customer for a specific month July.

Output screenshot: -

```
hduser@ubuntu64server:~$ hadoop fs -cat /Olive32/part-r-00000
Karen   155.18
```

Karen is the top customer who spent the most for online shipping.

CONCLUSION - Above Data Analysis shows that we can get various information using map reduce Hadoop processing to make better decision in E-commerce Industry which will help the website owner in providing better service for their customers.