Big Data Processing

ECS640U/ECS765P

200361138

Rohitkumar Subir Keswani

Semester1

MSc Big Data Science with

Industrial Experience

PartA: Time Analysis

1:

Job id:

1. http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_2886/

In this first question we have to create a bar plot showing the number of transactions occurring every month between the start and end of the datset.

First of I started by importing all the necessary library which is in this case is mrjob. And also we imported time as the time is in epoch format we had to convert that into regular time to get the months and the years.

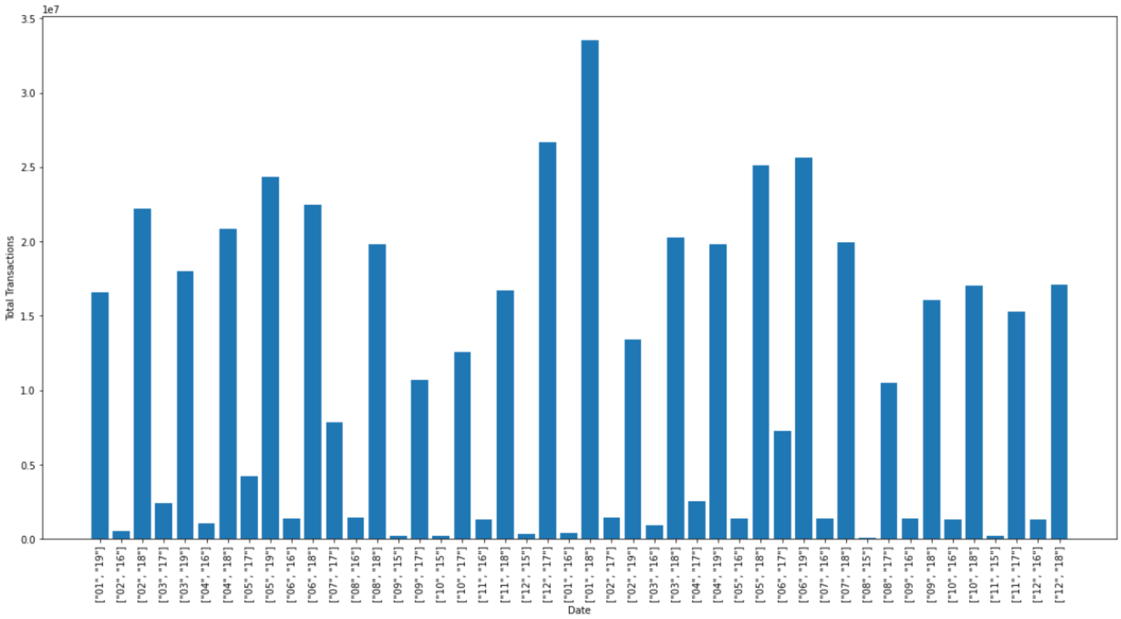
After that we counted the fields to check the fields of the dataset in this case we are using the transactions dataset schema.

After that I did a simple count program. Here I converted the epoch time in months and years and returned it as a output of the mapper and given that to the reducer which counted the number of transaction occurring every month.

Then I got the text file as an output of the map reduce program which is also attached in the PartAJob1 folder.

Then to plot the barplot of the number of transaction every month I used python for it.

I started by importing all the necessary libraries for it pandas, numpy, matplotlib. And then I read the output file in it and added the columns in it and then the plotted the bar plot with dates in x axis and transaction values in the y axis.



PARTA

2:

Jobid:

1. http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_4081/

For this question we have to plot the average value of transaction in each month between the start and end of the dataset.

Here again we started by importing all the necessary libraries we imported in previous question mrjob, re, time,statistics.

For this question also we are using the same transaction schema dataset.

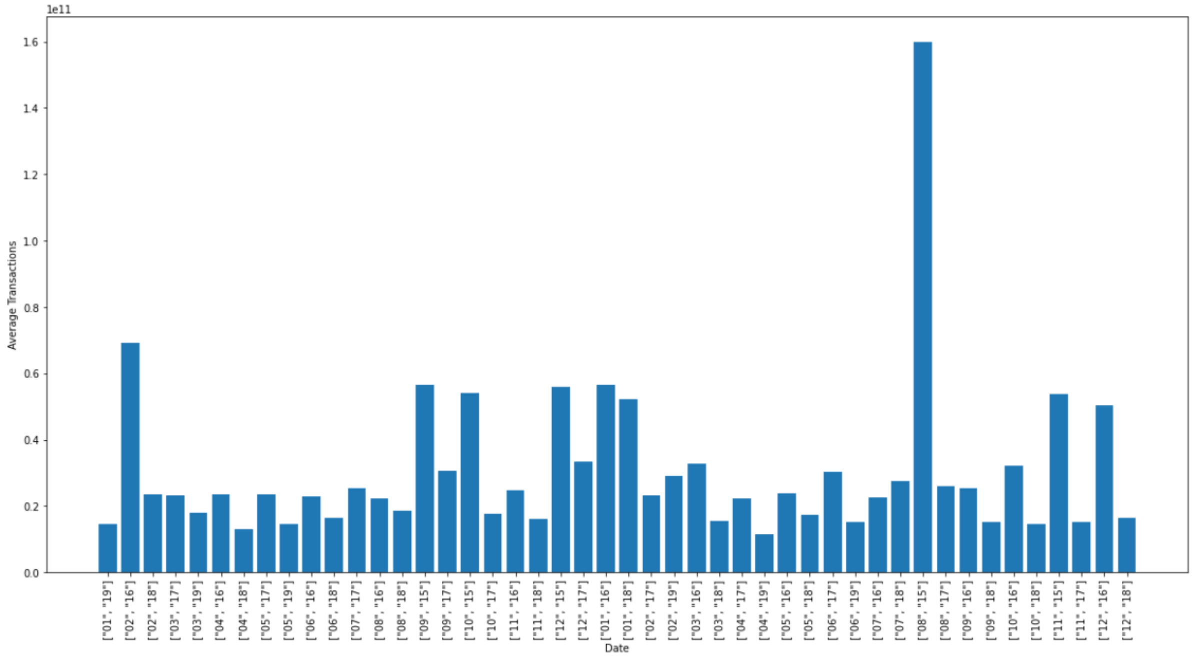
After that we did a similar code but took the gasprice also in our mapper only. Then I wrote a function which will calculate the average which calculated the sum of the count and returned the average.

Then we had a combiner which will decrease the computation time overall and for reducer also. Date was sent as a key where average was sent as a value to get average values of respective months and year.

Then we got the ouput of the program which is also attached in the PARTAJOB2 folder.

Then taking this txt file I plotted the graph in python for that again I imported all the necessary libraries such as pandas, numpy, matplotlib.

Then we name the columns as Date and Transactions and then plotted the graph using matplotlib library, which is also attached in the same folder.



PartB: Top ten most Popular Service

Jobid:

1. http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_3048/
2. http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_3111/

In this question we have to find out the top ten most popular services.

Here we will again start by importing all the necessary libraries mrjob and here we will also import MRStep.

In this code we are going to use two datasets Transactions and Contracts.

As we are using two dataset here we have to define which data is from which dataset for that we have to check the fields for it. If the fields is equal to seven it is from Transactions and if the fields is equal to five it is from contracts.

So in the first map Reduce job we are taking address as the key from both the dataset for map reduce jobs. We are taking value as a key value pair from the transaction dataset and we are passing value 1 with it which will help us to distinguish it from the other dataset. For contracts dataset also we are taking 2 as another value and also passing 1 which is the counter for the reduce stage.

Then the reducer checks the values from the values passed from the mapper and if they are true which means they exist and the record is valid. Then it sums all the values and key being the same passes this summed value as a value.

In the next mapper the mapper takes the values from the reducer and combines the key and values as the reducer has to find the top 10 contracts with None being the key. Then in the reducer we sort the values in decreasing order using lambda function then using a for loop we extract top 10 values and yield the top 10 values. The output is present in the folder PARTB.

Output:

```
"0xaa1a6e3e6ef20068f7f8d8c835d2d22fd5116444"84155100809965865822726776
"0xfa52274dd61e1643d2205169732f29114bc240b3"       45787484483189352986478805
"0x7727e5113d1d161373623e5f49fd568b4f543a9e"       45620624001350712557268573
"0x209c4784ab1e8183cf58ca33cb740efbf3fc18ef"  43170356092262468919298969
"0x6fc82a5fe25a5cdb58bc74600a40a69c065263f8"       27068921582019542499882877
"0xbfc39b6f805a9e40e77291aff27aee3c96915bdd"21104195138093660050000000
"0xe94b04a0fed112f3664e45adb2b8915693dd5ff3"15562398956802112254719409
"0xbb9bc244d798123fde783fcc1c72d3bb8c189413"       11983608729202893846818681
"0xabbb6bebfa05aa13e908eaa492bd7a8343760477"       11706457177940895521770404
"0x341e790174e3a4d35b65fdc067b6b5634a61caea"       8379000751917755624057500
```

PartC: Top Ten most Active Miners

Jobid:

1. http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_3131/
2. http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_3133/

In this question we have to find the top ten most active miners and here we are going to use the blocks dataset.

Here again we start by importing all the libraries which are MRJob and MRStep

So the value of fields will be 9 while checking the fields of the dataset as blocks dataset has 9 columns.

So in the first mapper we have miner as our key and size field as our value but we have converted it into integer as we have to perform the computation in the reducer.

So in the reducer we have summed the value we got from the mapper and yielded it as the value with key being the same.

For second mapper again as in previous question we have combined the key and value which are miner and totalsize with None being the key. In the reducer then we sort the values in the descending order then we use the for loop to get the top 10 miners then we yield those values.

Output:

```
"0xea674fdde714fd979de3edf0f56aa9716b898ec8"    23989401188

"0x829bd824b016326a401d083b33d092293333a830"          15010222714

"0x5a0b54d5dc17e0aadc383d2db43b0a0d3e029c4c" 13978859941

"0x52bc44d5378309ee2abf1539bf71de1b7d7be3b5"  10998145387

"0xb2930b35844a230f00e51431acae96fe543a0347"  7842595276

"0x2a65aca4d5fc5b5c859090a6c34d164135398226"  3628875680

"0x4bb96091ee9d802ed039c4d1a5f6216f90f81b01"   1221833144

"0xf3b9d2c81f2b24b0fa0acaaa865b7d9ced5fc2fb"     1152472379

"0x1e9939daaad6924ad004c2560e90804164900341"1080301927

"0x61c808d82a3ac53231750dadc13c777b59310bd9" 692942577
```

PartD

Price Forecasting:

Jobid: http://andromeda.student.eecs.qmul.ac.uk:8088/cluster/app/application_1607539937312_5586

In this question we had to build a price forecasting model.

For this we used the data from Ethereum — Opendatasoft website. I downloaded this data on 12th December 2020.

Now for this we started by importing all the necessary libraries which is pyspark and many other libraries from pyspark itself.

After downloading the dataset in the local system I uploaded it into hdfs and started the code by giving the path of the csv with name ethereum.csv. In this question we used spark and mllib to build a model. We start by storing the dataset in the df1 which is our data frame.

The vector assembler then takes input columns and output columns and stores it as a vector. The obtained assembler is used to transform our df1 dataframe. Then output feature is selected which is PriceUSD column which we have to predict then we have split the data into training and testing which is 70 percent training data and 30 percent test data .

Here we are using linearRegression model with max iteration 100 and then we fit the training data and then predict on our test data.

```
121.820662908

0.690286000006

|[7.230786299948E7...|82.69275857668492|

|[7.233504128073E7...|80.86410519402853|

|[7.253323940573E7...|80.63175447423782|

|[7.256135831198E7...|80.57654554553574|

|[7.258752112448E7...|80.34974013179203|

|[7.266255112448E7...|81.41401334026182|

|[7.279656456198E7...| 80.5938847939475|

|[7.282109737448E7...|82.05248791667134|

|[7.291059456198E7...|80.92334901386312|

|[7.293093378073E7...|80.69382004115516|

|[7.294932487448E7...|80.62882969904888|

|[7.319945081198E7...|79.11355299967829|

|[7.325121034323E7...|78.71298830512603|

|[7.332857940573E7...|77.59833286672074|

|[7.338166081198E7...|77.20962411739458|

|[7.343629534323E7...|78.09926408285128|

|[7.354313518698E7...| 76.7219175086542|

|[7.372256815573E7...|75.62072241995384|

|[7.382567549948E7...|75.31082533096378|

|[7.385180909323E7...|  75.783301442577|
```

Here we can see that we got root mean square value of 0.69 which is around 70 percent which is good .

Popular Scams:

Jobid:

1. http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_2763/
2. http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_2799/

Here we are using two dataset scams.json file and the transaction dataset. So we will start by importing all the necessary libraries MRJob, MRStep and json . We will start by checking the fields of the dataset if its is equal to seven then it is of transaction dataset and if its not which means it is not our required dataset and then we will load the json file.

Now from the transaction dataset we will take address as key and value as our value and we will also use 0 which will help us to distinguish between json file. Similary we take address from the scams data set which we get from the for loop and we take value as value and put 1 to distinguish it from transactions.

In the reducer we check whether the values are from transactions or json file and then is passed it to the next mapper take the key value pair from the reducer which is categories and total values and pass it on the next reducer and in the second reducer total number of values are counted of how much values are there of how many categories.

Output:

```
"Scamming"  3.833616286244437e+22

"Fake ICO"  1.35645756688963e+21

"Phishing"  2.6999375794087423e+22

"Scam"      0
```

Jobid2: http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_9609/

http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_9698/

In this code again we will start by importing the necessary libraries and importing the json. Here again we are using transactions and scams.json file. So we will check the fields first that we are using the correct dataset or not.

We will take address as akey value in the mapper and we will use 1 to distinguish it like we did in previous questions. From the reducer also we will take address as a key which we are getting from the for loop and for the value we take category and status of the scam and we will use 2 for distinguishing purposes.

We will then check for the values of it is one we will add the values, if it is not one we will add them to categories and status as they are from the json file having categories and values of the scams.

In the secong mapper we just take the values received from the previous reducer and give them to the next reducer where it performs the sum operation which is counting the total number of scams of particular category.

The below output gives total number of scams in various categories of scams.

Output3:

```
["Active", "Scamming"]  88444

["Inactive", "Phishing"]        22

["Offline", "Fake ICO"] 121

["Offline", "Phishing"] 7022

["Offline", "Scam"]     0

["Suspended", "Phishing"]       11

["Active", "Phishing"]  1584

["Offline", "Scamming"] 24692

["Suspended", "Scamming"]       56
```

GasGuzzlers:

Jobid:
http://andromeda.student.eecs.qmul.ac.uk:8088/cluster/app/application_1607539937312_6785

In this question we have to do analysis of the gas for example how gas price has changed over time how have contracts became more complicated .

For this code I have started by importing pyspark and time. For this problem we have to take three datasets which are transactions, contracts and blocks. We will start by creating three functions for each dataset to check whether we have the correct dataset or not for that we will check the dataset by counting there fields. So for transaction dataset there should be 7 fields for contracts dataset there should be 5 fields and for blocks dataset there should be 9 fields if it does not satisfy it will return false.

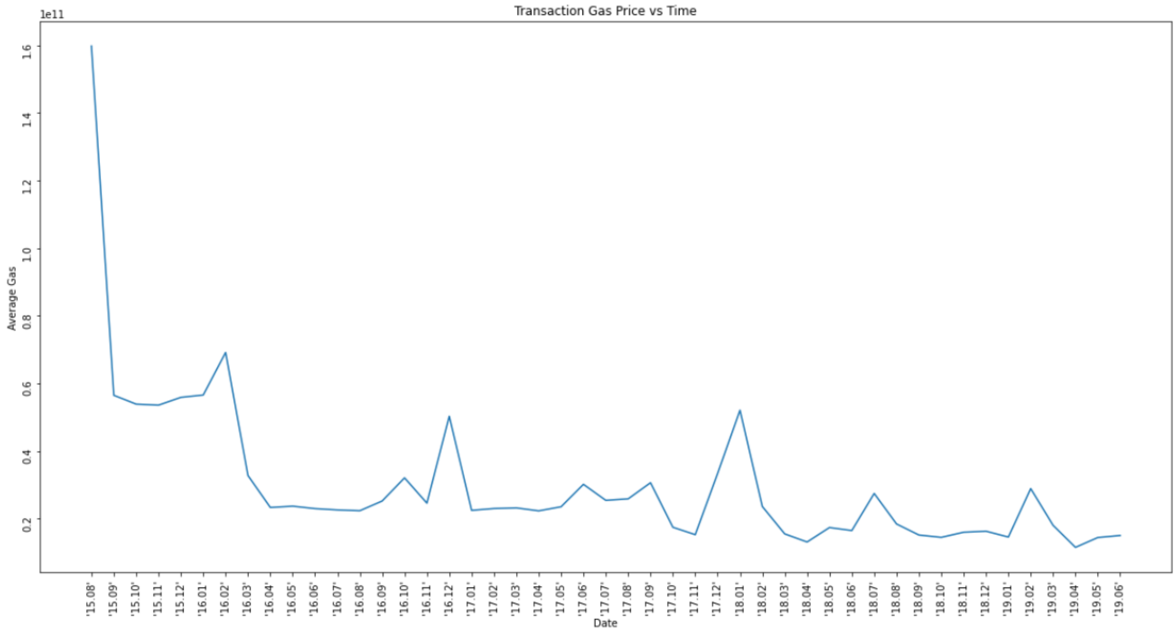
Then we will load the dataset and will check those through these functions.

In the first stage the average price of the gas is calculated for each month it is done by total sum og gas price divide by total number of transactions.

I got the output in the text file AverageGas which is also attached. Now taking this file as input I loaded this data in python. So in python I started by loading important libraries which I used which were numpy, pandas, and matplotlib.

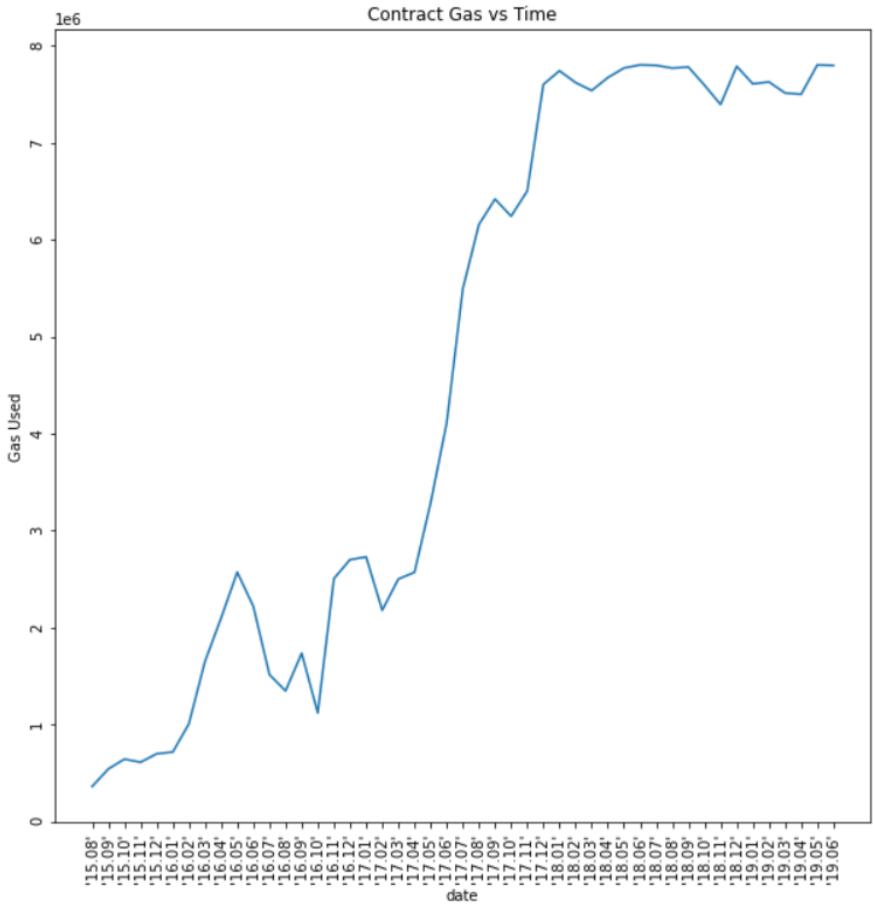
After loading the dataset I assigned the names to the columns header as date and gas respectively.

Then the data was very messed up so I have to clean it befor plotting. So first I faced was there were many square brackets so I tried to remove those using str.strip method and punctuation marks and then I converted the gas column into float, and then with the help of matplotlib library I plotted the lineplot of the Transaction Gas Price vs Time.



In the above graph we can see that the average price of gas has decreased from 2015 to 2019 and we can also see that in every year there is a spike in prices either in last few months or first few months of the year.

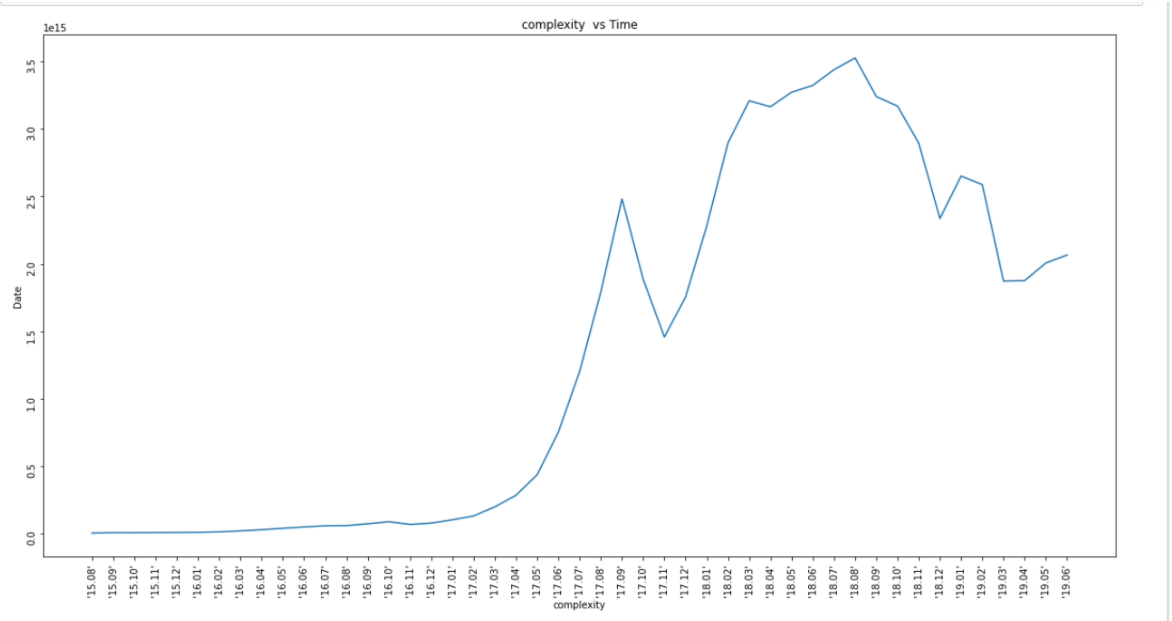
After that with the same approach i find out the transactions which were smart contract so for that we need to make sure that it is joined with the contracts. As id is unique we can use it as a joining key.



We can see that there is a high demand in last few years comparing to when it started in 2015 but now it is in steady state.

The out I got was save in Timedifference text file which is also attached.

In this text file the data was very dirty and had many brackets and commas compare to previous text file. The ipynb files are also attached in the folders.



We can clearly observe that with time contracts have became more complicated and can see the plot has 2 peaks one in 2017 and one in 2018.

Comparative Evaluation:

Jobid: http://andromeda.student.eecs.qmul.ac.uk:8088/cluster/app/application_1607539937312_3978

In this question we have done the PartB in the spark code.

Here we are using the Transactions and Contracts dataset as we done in the PartB with map reduce jobs.

We start by importing the pyspark module. Then we start by writing the function to check the fields as we did in the map reduce code if the field is not equal to 7 then we returned false for the transaction dataset and if the value of the fields is not equal to 5 then we returned false for contracts dataset.

Then we load the dataset and check the fields with the above functions .Then we take address as our key and value as our value. And then in the reduce stage with key being the same but aggregating all the values for the keys. And then for contracts we map the address as the key. Then joing operation is performed between this address and the aggregate values which we got by performing the computation on the transaction dataset in the previous reducer.

We then sort the data in the decreasing order and then top 10 values are extracted.

Output:

```
0xaa1a6e3e6ef20068f7f8d8c835d2d22fd5116444 : 84155100809965865822726776
0xfa52274dd61e1643d2205169732f29114bc240b3 : 45787484483189352986478805
0x7727e5113d1d161373623e5f49fd568b4f543a9e : 45620624001350712557268573
0x209c4784ab1e8183cf58ca33cb740efbf3fc18ef : 43170356092262468919298969
0x6fc82a5fe25a5cdb58bc74600a40a69c065263f8 : 27068921582019542499882877
0xbfc39b6f805a9e40e77291aff27aee3c96915bdd : 21104195138093660050000000
0xe94b04a0fed112f3664e45adb2b8915693dd5ff3 : 15562398956802112254719409
0xbb9bc244d798123fde783fcc1c72d3bb8c189413 : 11983608729202893846818681
0xabbb6bebfa05aa13e908eaa492bd7a8343760477 : 11706457177940895521770404
0x341e790174e3a4d35b65fdc067b6b5634a61caea : 8379000751917755624057500
```

Spark Jobs:

1st time:

Jobid: http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_8517

Time: 11mins 29 seconds

2nd Time:

Jobid: http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_8561

Time: 6 mins 7 seconds

3rd Time:

Jobid: http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_8581

Time: 10mins 26seconds

Average Time Taken: 9 min 34 seconds

Median Time Taken:10 mins 26 seconds

Hadoop jobs:

1st time: 50 mins 19 seconds

Jobid1: http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_8289/

Jobid2: http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_8289/

Time: 50 mins 19 seconds

2nd time: 47 mins 54 seconds

Jobid1: http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_9431/

Jobid2: http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_9498/

Time:

3rd time: 44 mins 46 seconds

Jobid: : http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_9515/

Jobid2:
http://andromeda.student.eecs.qmul.ac.uk:8088/proxy/application_1607539937312_9609/

Average Time Taken: 47.53

Median Time Taken:47.54

As we can take see that average time taken by map reduce was around 48 minutes and average time taken by spark was around 9 minutes.

Looking at the average time only we can find out that spark is much faster than map reduce and spark seems to be appropriate framework for this task as it takes less than 10 minutes to complete the task.

Spark computes the task at such a quick rate comparing to map reduce is because of its in memory processing, this makes Spark faster than map reduce. Also spark

RDD has a quick way to process and retrieve data in Hadoop comparing to Map reduce. Thus Spark is the most suitable framework for this task.