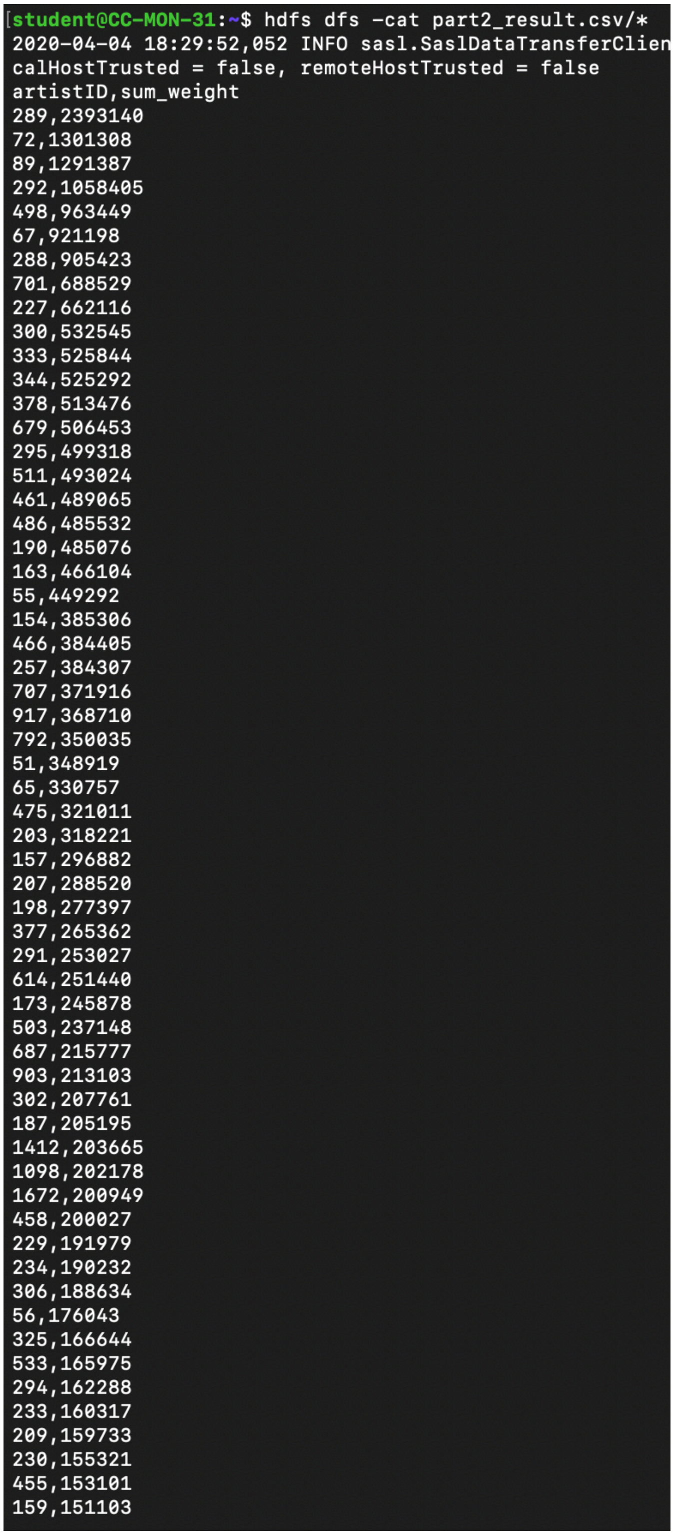
Cloud Computing MiniProject 2 Report

Ruoxi Zhang ruz39@pitt.edu
Kenny Wu kew143@pitt.edu
Zhanghaoxiang Yin zhy77@pitt.edu

Part 2. Listening Counts on ‘user_artists.dat’

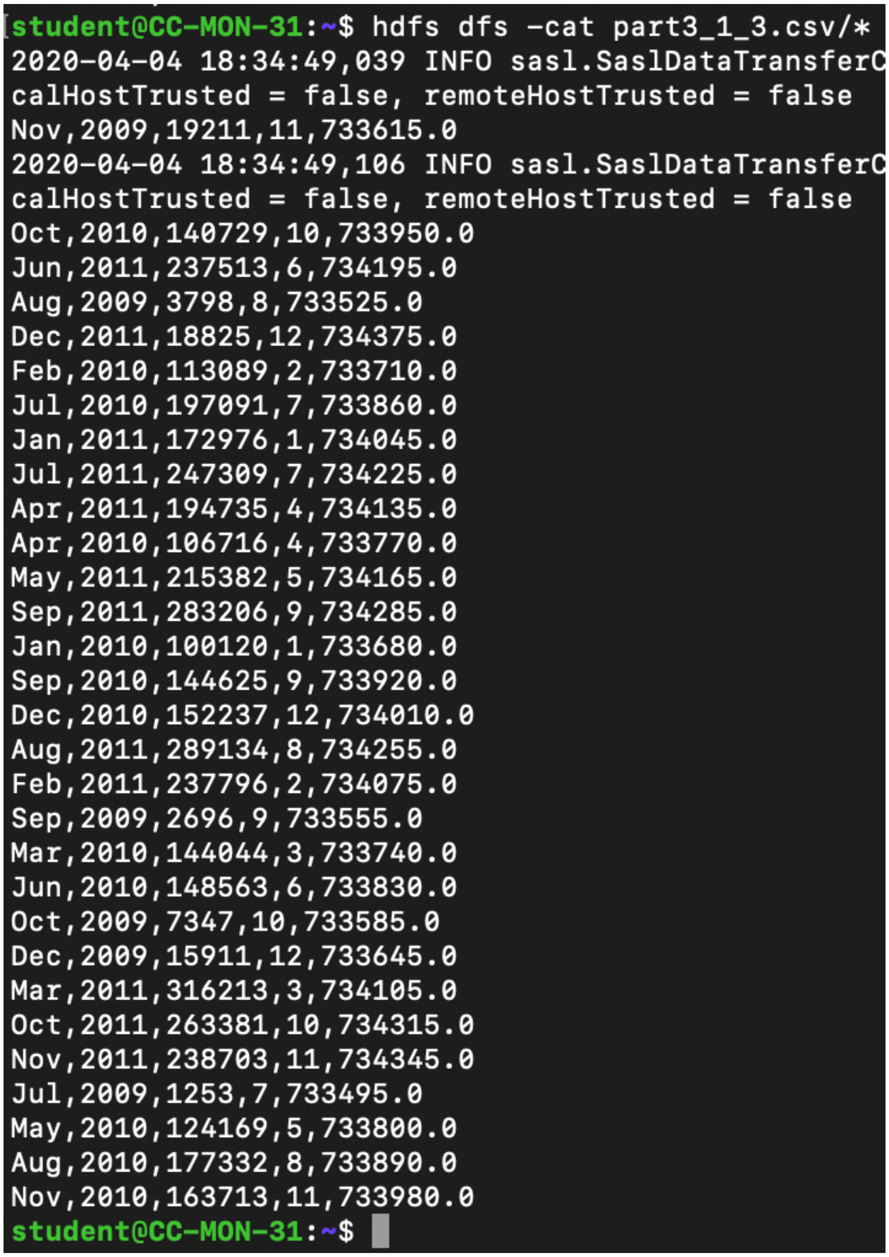


```
[student@CC-MON-31:~$ hdfs dfs -cat part2_result.csv/*
2020-04-04 18:29:52,052 INFO sasl.SaslDataTransferClien
calHostTrusted = false, remoteHostTrusted = false
artistID,sum_weight
289,2393140
72,1301308
89,1291387
292,1058405
498,963449
67,921198
288,905423
701,688529
227,662116
300,532545
333,525844
344,525292
378,513476
679,506453
295,499318
511,493024
461,489065
486,485532
190,485076
163,466104
55,449292
154,385306
466,384405
257,384307
707,371916
917,368710
792,350035
51,348919
65,330757
475,321011
203,318221
157,296882
207,288520
198,277397
377,265362
291,253027
614,251440
173,245878
503,237148
687,215777
903,213103
302,207761
187,205195
1412,203665
1098,202178
1672,200949
458,200027
229,191979
234,190232
306,188634
56,176043
325,166644
533,165975
294,162288
233,160317
209,159733
230,155321
455,153101
159,151103
```

Part 3. Linear Regression on ‘access_log’

Step 1. Group the data by timestamp in month.

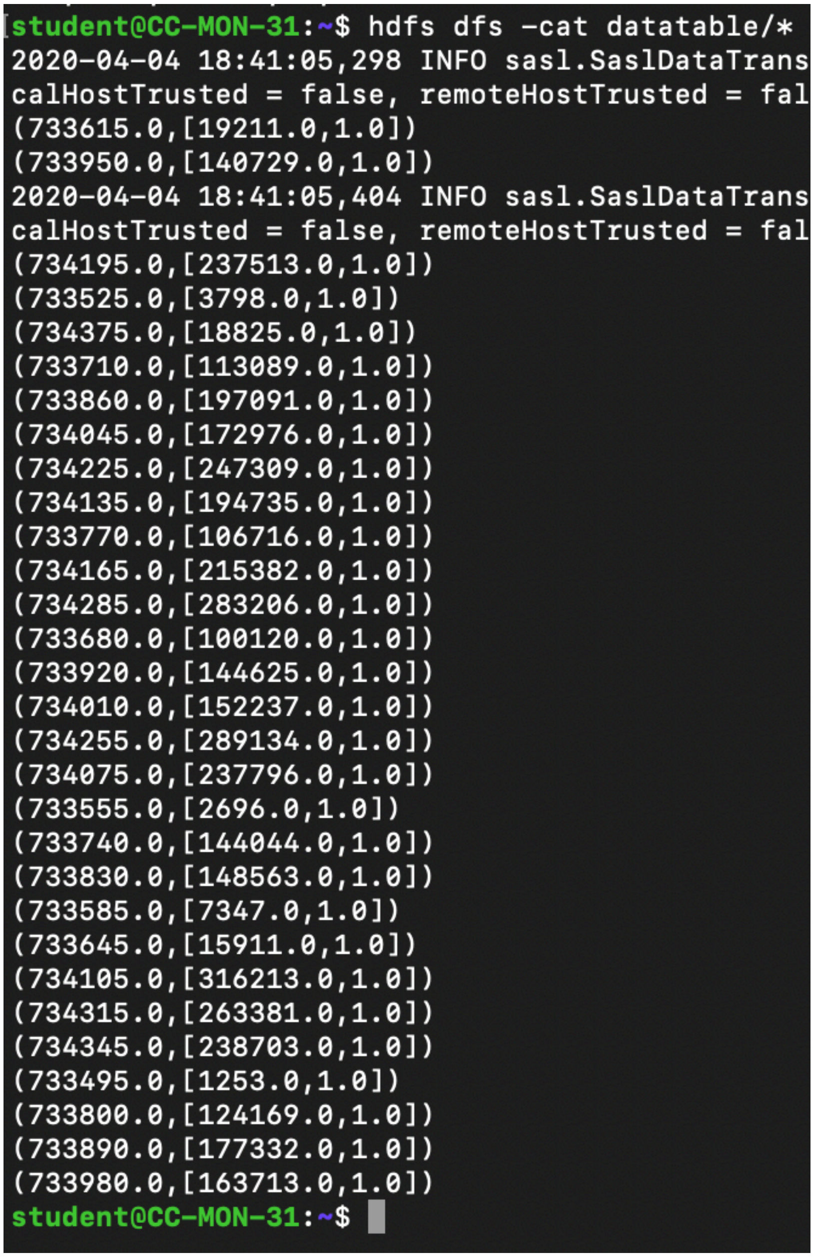
The 1st column is the month name in timestamp.
The 2nd column is the year in timestamp.
The 3rd column is the counts of IP address for each month.
The 4th column is the integer value of month.
The 5th column is the approximate of proleptic Gregorian ordinal date.
(1 year = 365 days; 1 month = 30 days.)



```
[student@CC-MON-31:~$ hdfs dfs -cat part3_1_3.csv/*
2020-04-04 18:34:49,039 INFO sasl.SaslDataTransferC
calHostTrusted = false, remoteHostTrusted = false
Nov,2009,19211,11,733615.0
2020-04-04 18:34:49,106 INFO sasl.SaslDataTransferC
calHostTrusted = false, remoteHostTrusted = false
Oct,2010,140729,10,733950.0
Jun,2011,237513,6,734195.0
Aug,2009,3798,8,733525.0
Dec,2011,18825,12,734375.0
Feb,2010,113089,2,733710.0
Jul,2010,197091,7,733860.0
Jan,2011,172976,1,734045.0
Jul,2011,247309,7,734225.0
Apr,2011,194735,4,734135.0
Apr,2010,106716,4,733770.0
May,2011,215382,5,734165.0
Sep,2011,283206,9,734285.0
Jan,2010,100120,1,733680.0
Sep,2010,144625,9,733920.0
Dec,2010,152237,12,734010.0
Aug,2011,289134,8,734255.0
Feb,2011,237796,2,734075.0
Sep,2009,2696,9,733555.0
Mar,2010,144044,3,733740.0
Jun,2010,148563,6,733830.0
Oct,2009,7347,10,733585.0
Dec,2009,15911,12,733645.0
Mar,2011,316213,3,734105.0
Oct,2011,263381,10,734315.0
Nov,2011,238703,11,734345.0
Jul,2009,1253,7,733495.0
May,2010,124169,5,733800.0
Aug,2010,177332,8,733890.0
Nov,2010,163713,11,733980.0
student@CC-MON-31:~$
```

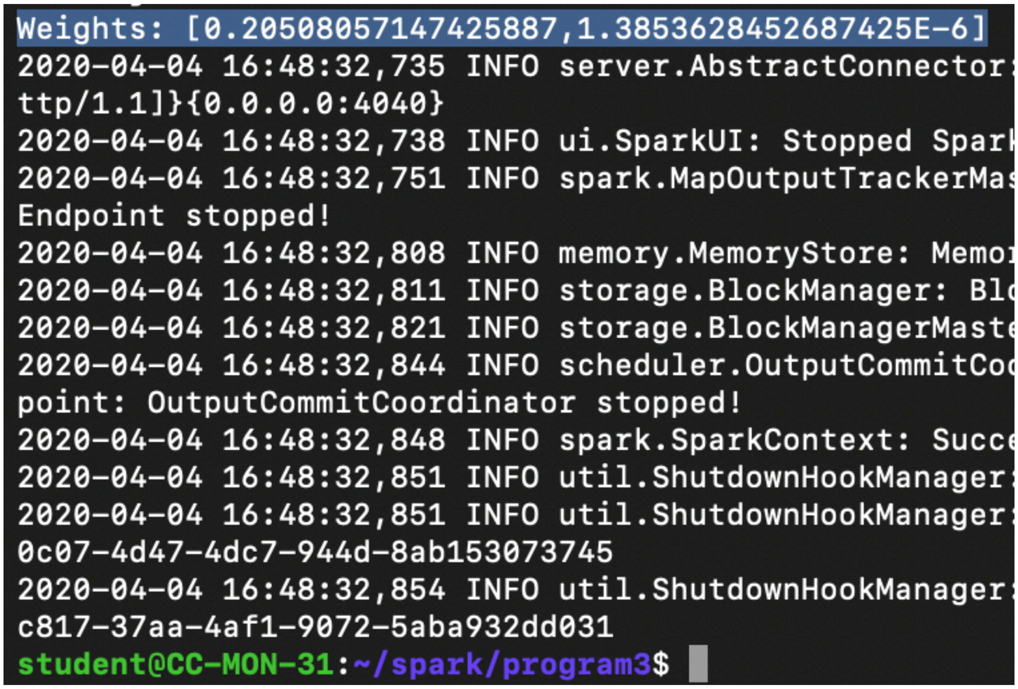
Step 2. Do simple Linear Regression on the grouped result.

Datatable for the model input:
The 1st column is label (the approximate of proleptic Gregorian ordinal date).
The 2nd column is features:
First feature is the counts of IP address for each month.
Second feature is the constant of 1, which indicates the existence of intercept of the linear model.



```
[student@CC-MON-31:~$ hdfs dfs -cat datatable/*
2020-04-04 18:41:05,298 INFO sasl.SaslDataTrans
calHostTrusted = false, remoteHostTrusted = fal
(733615.0,[19211.0,1.0])
(733950.0,[140729.0,1.0])
2020-04-04 18:41:05,404 INFO sasl.SaslDataTrans
calHostTrusted = false, remoteHostTrusted = fal
(734195.0,[237513.0,1.0])
(733525.0,[3798.0,1.0])
(734375.0,[18825.0,1.0])
(733710.0,[113089.0,1.0])
(733860.0,[197091.0,1.0])
(734045.0,[172976.0,1.0])
(734225.0,[247309.0,1.0])
(734135.0,[194735.0,1.0])
(733770.0,[106716.0,1.0])
(734165.0,[215382.0,1.0])
(734285.0,[283206.0,1.0])
(733680.0,[100120.0,1.0])
(733920.0,[144625.0,1.0])
(734010.0,[152237.0,1.0])
(734255.0,[289134.0,1.0])
(734075.0,[237796.0,1.0])
(733555.0,[2696.0,1.0])
(733740.0,[144044.0,1.0])
(733830.0,[148563.0,1.0])
(733585.0,[7347.0,1.0])
(733645.0,[15911.0,1.0])
(734105.0,[316213.0,1.0])
(734315.0,[263381.0,1.0])
(734345.0,[238703.0,1.0])
(733495.0,[1253.0,1.0])
(733800.0,[124169.0,1.0])
(733890.0,[177332.0,1.0])
(733980.0,[163713.0,1.0])
student@CC-MON-31:~$
```

Step 3. The Linear Regression Model Result:



```
Weights: [0.20508057147425887,1.3853628452687425E-6]
2020-04-04 16:48:32,735 INFO server.AbstractConnector:
ttp/1.1]}{0.0.0.0:4040}
2020-04-04 16:48:32,738 INFO ui.SparkUI: Stopped Spark
2020-04-04 16:48:32,751 INFO spark.MapOutputTrackerMas
Endpoint stopped!
2020-04-04 16:48:32,808 INFO memory.MemoryStore: Memor
2020-04-04 16:48:32,811 INFO storage.BlockManager: Blo
2020-04-04 16:48:32,821 INFO storage.BlockManagerMaste
2020-04-04 16:48:32,844 INFO scheduler.OutputCommitCoo
point: OutputCommitCoordinator stopped!
2020-04-04 16:48:32,848 INFO spark.SparkContext: Succe
2020-04-04 16:48:32,851 INFO util.ShutdownHookManager:
2020-04-04 16:48:32,851 INFO util.ShutdownHookManager:
0c07-4d47-4dc7-944d-8ab153073745
2020-04-04 16:48:32,854 INFO util.ShutdownHookManager:
c817-37aa-4af1-9072-5aba932dd031
student@CC-MON-31:~/spark/program3$
```

The model can be written as :

$$y = 0.20508057147425887 \cdot x + 1.3853628452687425E - 6$$