Christian Craig
CSC 555 Project Phase 2

CSC 555: Mining Big Data

Project, Phase 2 (due Sunday, March 21st)

In this part of the project, you will execute queries using Hive, Pig and Hadoop streaming and develop a custom version of KMeans clustering. The schema is available below:
http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/SSBM_schema_hive.sql

The data is available at http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/

In your submission, please note what cluster you are using. Please be sure to submit all code. You should also submit the command lines you use and a screenshot of a completed run (just the last page, do not worry about capturing the entire output).

I highly recommend creating a small sample input (e.g., by running head -n 1000 lineorder.tbl > lineorder.tbl.sample, you can create a small version of lineorder with 1000 lines) and testing your code with it.

Christian Craig
CSC 555 Project Phase 2

Part 1: Pig

Implement the following query using Pig:

```
select c_nation, AVG(lo_extendedprice) as AVG1
from customer, lineorder
where lo_custkey = c_custkey
  and c_region = 'AFRICA'
  and lo_discount = 5
group by c_nation
order by AVG1;
```

```
create table lineorder (
  lo_orderkey          int,
  lo_linenumber        int,
  lo_custkey           int,
  lo_partkey           int,
  lo_suppkey           int,
  lo_orderdate         int,
  lo_orderpriority     varchar(15),
  lo_shippriority      varchar(1),
  lo_quantity          int,
  lo_extendedprice     int,
  lo_ordertotalprice   int,
  lo_discount          int,
  lo_revenue           int,
  lo_supplycost        int,
  lo_tax               int,
  lo_commitdate         int,
  lo_shipmode          varchar(10)
);
```

```
create table customer (
  c_custkey     int,
  c_name        varchar(25),
  c_address     varchar(25),
  c_city        varchar(10),
  c_nation      varchar(15),
  c_region      varchar(12),
  c_phone       varchar(15),
  c_mktsegment  varchar(10)
);
```

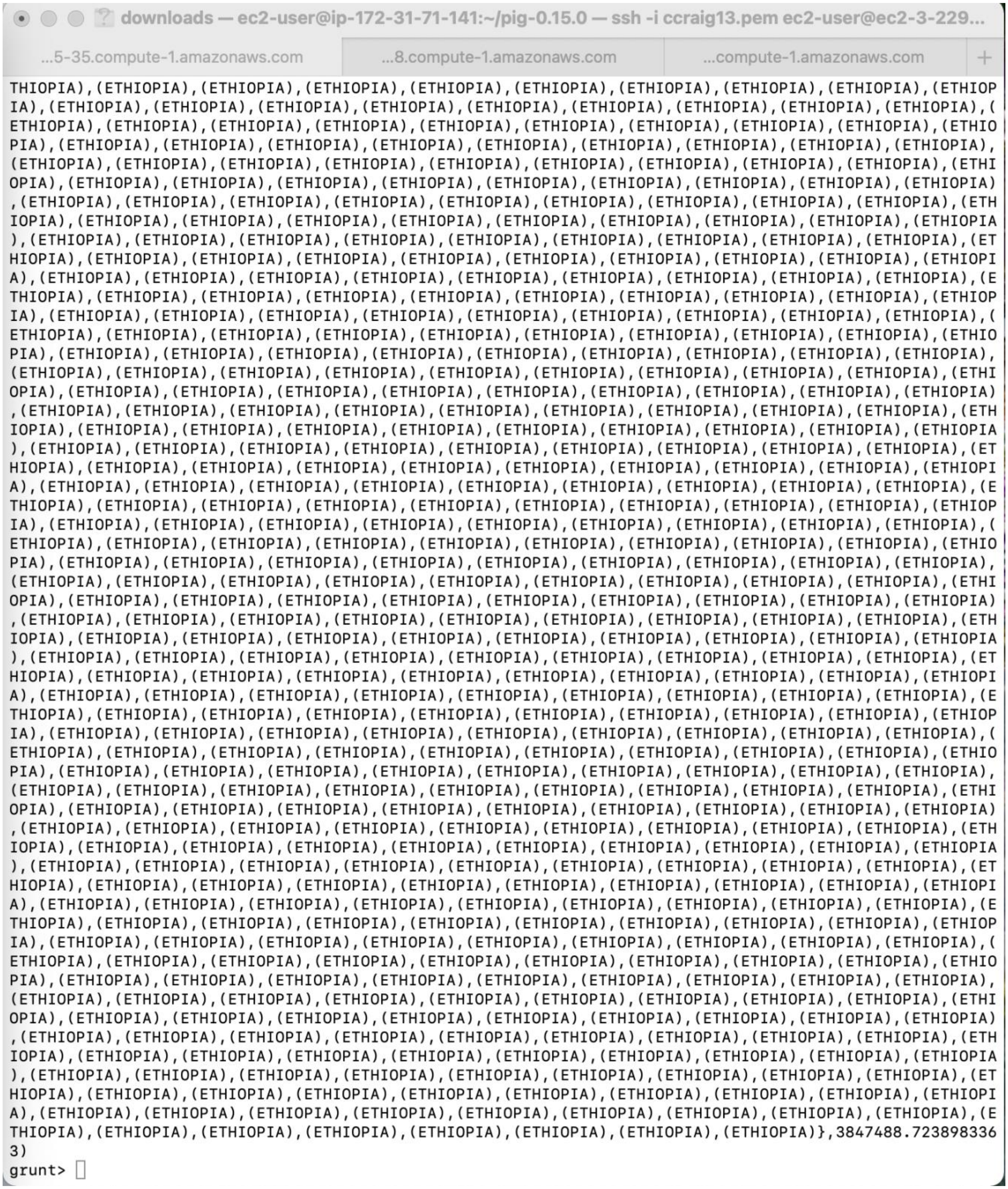
lineorder_tbl = LOAD '/user/ec2-user/lineorder.tbl.1' USING PigStorage('|') AS(lo_orderkey:int, lo_linenumber:int, lo_custkey: int, lo_partkey:int, lo_suppkey:int,lo_orderdate: int,lo_orderpriority:chararray,lo_shippriority:chararray,lo_quantity:int,lo_extendedprice:int,lo_ordertotalprice:int,lo_discount: int,lo_revenue: int,lo_supplycost:int,lo_tax: int,lo_commitdate: int, lo_shipmode:chararray);

customer_tbl = LOAD '/user/ec2-user/customer.tbl' USING PigStorage('|') AS(c_custkey:int, c_name:chararray, c_address:chararray, c_city:chararray, c_nation:chararray, c_region:chararray, c_phone:chararray, c_mktsegment:chararray);

```
join_tbls = join lineorder_tbl by lo_custkey, customer_tbl by c_custkey;

filter_tbls = filter join_tbls by (c_region == 'AFRICA') AND (lo_discount == 5);

group_tbls = group filter_tbls by c_nation;

avg_ep = foreach group_tbls generate filter_tbls.c_nation,
AVG(filter_tbls.lo_extendedprice) AS AVG1;
```

```
order_tbls = order avg_ep by AVG1;
dump order_tbls;
```

Christian Craig
CSC 555 Project Phase 2



```
THIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOP
IA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(
ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIO
PIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),
(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHI
OPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA)
,(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETH
IOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA
),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ET
HIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPI
A),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(E
THIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOP
IA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(
ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIO
PIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),
(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHI
OPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA)
,(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETH
IOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA
),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ET
HIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPI
A),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(E
THIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOP
IA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(
ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIO
PIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),
(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHI
OPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA)
,(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETH
IOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA
),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ET
HIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPI
A),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(E
THIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOP
IA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(
ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIO
PIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),
(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHI
OPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA)
,(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETH
IOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA
),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ET
HIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPI
A),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(E
THIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOP
IA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(
ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIO
PIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),
(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHI
OPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA)
,(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETH
IOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA
),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ET
HIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPI
A),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(E
THIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA),(ETHIOPIA)},3847488.723898336
3)
grunt> 
```

Christian Craig
CSC 555 Project Phase 2

Part 2: Hadoop streaming

Implement the following query using Hadoop streaming:

```
select sum(lo_revenue), d_year, p_brand1
from lineorder, dwdate, part
where lo_orderdate = d_datekey
  and lo_partkey = p_partkey
  and p_brand1 between 'MFGR#2221'
  and 'MFGR#2228'
group by d_year, p_brand1
```

In Hadoop streaming, this will use a total of 3 passes (two joins and another one for GROUP BY).

```
create table lineorder (
  0 lo_orderkey          int,
  1 lo_linenumber        int,
  2 lo_custkey           int,
  3 lo_partkey           int,
  4 lo_suppkey           int,
  5 lo_orderdate         int,
  6 lo_orderpriority
varchar(15),
  7 lo_shippriority
varchar(1),
  8 lo_quantity          int,
  9 lo_extendedprice     int,
  10 lo_ordertotalprice   int,
  11 lo_discount          int,
  12 lo_revenue           int,
  13 lo_supplycost        int,
  14 lo_tax               int,
  15 lo_commitdate         int,
  16 lo_shipmode
varchar(10)
);
```

```
create table dwdate (
  0 d_datekey            int,
  1 d_date               varchar(19),
  2 d_dayofweek          varchar(10),
  3 d_month              varchar(10),
  4 d_year               int,
  5 d_yearmonthnum       int,
  6 d_yearmonth          varchar(8),
  7 d_daynuminweek       int,
  8 d_daynuminmonth      int,
  9 d_daynuminyear       int,
  10 d_monthnuminyear     int,
  11 d_weeknuminyear      int,
  12 d_sellingseason      varchar(13),
  13 d_lastdayinweekfl    varchar(1),
  14 d_lastdayinmonthfl   varchar(1),
  15 d_holidayfl          varchar(1),
  16 d_weekdayfl          varchar(1)
);
```

```
create table part (
  0 p_partkey     int,
  1 p_name        varchar(22),
  2 p_mfgr        varchar(6),
  3 p_category    varchar(7),
  4 p_brand1      varchar(9),
  5 p_color       varchar(11),
  6 p_type        varchar(25),
  7 p_size        int,
  8 p_container   varchar(10)
);
```

Christian Craig
CSC 555 Project Phase 2

```
hadoop fs -mkdir /user/ec2-user/data/joinTbl_final_1
hadoop fs -put lineorder.tbl dwdate.tbl /user/ec2-user/data/joinTbl_final_1
```

downloads — ec2-user@ip-172-31-71-141:~/hadoop-2.6.4 — ssh -i ccraig13.pem ec2-user@ec2-3-2..

...-225.compute-1.amazonaws.com | ...8.compute-1.amazonaws.com | ...compute-1.amazonaws.com

GNU nano 2.9.8 final_mapper1.py

```python
#!/usr/bin/python
import sys

#input comes from STDIN (standard input)
for line in sys.stdin:

    line = line.strip()
    split = line.split('|')
    wk_lst = ['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']

    if split[2] in wk_lst: #dwdate
        d_datekey = split[0]
        d_year = split[4]
        print(d_datekey +'\t'+ d_year +'\t'+'dw')
    else: #lineorder
        lo_partkey = split[3]
        lo_orderdate = split[5]
        lo_revenue = split[12]
        print(lo_orderdate+'\t'+lo_revenue+'\t'+lo_partkey+'\t'+'lo')
```

Christian Craig
CSC 555 Project Phase 2

downloads — ec2-user@ip-172-31-71-141:~/hadoop-2.6.4 — ssh -i ccraig13.pem ec2-user@ec2-3-2...

...-225.compute-1.amazonaws.com | ...8.compute-1.amazonaws.com | ...compute-1.amazonaws.com

GNU nano 2.9.8 final_reducer1.py

```
#!/usr/bin/python
import sys

currentKey = None
valsDw = None
valsLo = None
TotalCount = 0

for line in sys.stdin:

    split = line.strip().split('\t')
    key = split[0]
    value = '|'.join(split[1:])

    if currentKey == key: #same key
        if value.endswith('dw'):
            valsDw.append(value[:-2])
        if value.endswith('lo'):
            valsLo.append(value[:-2])
    else:
        if currentKey:
            lenDw = len(valsDw)
            lenLo = len(valsLo)
            if (lenDw*lenLo>0):
                for l,d in zip(valsLo,valsDw):
                    print'%s%s'%(l,d)
        valsDw = []
        valsLo = []

        currentKey = key
        if value.endswith('dw'):
            valsDw = [value[:-2]]
            valsLo = []
        elif value.endswith('lo'):
            valsDw = []
            valsLo = [value[:-2]]

lenDwLast = len(valsDw)
lenLoLast = len(valsLo)
if (lenLoLast*lenDwLast>0):
    for l,d in zip(valsLo,valsDw):
        print'%s%s'%(l,d)
```

Christian Craig
CSC 555 Project Phase 2

```
hadoop jar hadoop-streaming-2.6.4.jar -input /user/ec2-user/data/joinTbl_final_1 -
mapper final_mapper1.py -file final_mapper1.py -reducer final_reducer1.py -file
final_reducer1.py -output /join2/output7777
```

hadoop fs -cat /join2/output7777/part-00000

downloads — ec2-user@ip-172-31-71-141:~/hadoop-2.6.4 — ssh -i ccraig13.pem ec2-user@ec2-3-2...

...-225.compute-1.amazonaws.com | ...8.compute-1.amazonaws.com | ...compute-1.amazonaws.com

```
4570677|103126|1998|
7555941|113963|1998|
6268744|135786|1998|
3873580|30557|1998|
3277643|169316|1998|
8368722|179775|1998|
7623919|42786|1998|
1627204|67623|1998|
5030704|104348|1998|
4941756|184414|1998|
3272525|105386|1998|
1501661|19695|1998|
1947230|112974|1998|
6805194|110918|1998|
2810566|72175|1998|
1017490|18464|1998|
2140935|107447|1998|
6732281|5796|1998|
2858464|162943|1998|
2424143|105355|1998|
127432|47327|1998|
2617802|25837|1998|
7331885|155769|1998|
3773073|64850|1998|
3560516|155729|1998|
5973598|21645|1998|
4644260|39890|1998|
2628670|61666|1998|
3624251|98988|1998|
5727576|39735|1998|
5269785|132340|1998|
3934769|85815|1998|
4741693|20466|1998|
2134383|38356|1998|
761051|60676|1998|
7236131|149808|1998|
4374591|5076|1998|
4016661|179219|1998|
3029821|142672|1998|
6226388|138916|1998|
1372672|97963|1998|
171436|84801|1998|
6706269|130870|1998|
4114889|107676|1998|
3966048|58377|1998|
4909014|76917|1998|
6833771|102528|1998|
2412210|738|1998|
1005336|193303|1998|
4402947|19815|1998|
5684775|60970|1998|
3378034|161474|1998|
3128136|85268|1998|
950294|198114|1998|
6621111|82653|1998|
7667131|89958|1998|
4040622|58304|1998|
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$
```

Christian Craig
CSC 555 Project Phase 2

downloads — ec2-user@ip-172-31-71-141:~/hadoop-2.6.4 — ssh -i ccraig13.pem ec2-user@ec2-3-...

...-78.compute-1.amazonaws.com | ...8.compute-1.amazonaws.com | ...compute-1.amazonaws.com

GNU nano 2.9.8 final_mapper2.py

```
#!/usr/bin/python

import sys

for line in sys.stdin:
        line = line.strip()
        split = line.split('|')
        if split[2].startswith('MFGR'): #part
                p_partkey = split[0]
                p_brand = split[4]
                if p_brand>'MFGR#2221' and p_brand<'MFGR#2228':
                        print(p_partkey+'\t'+p_brand+'\t'+'pt')
        else: #r1
                lo_rev = split[0]
                lo_partkey = split[1]
                d_year = split[2]
                print(lo_partkey+'\t'+lo_rev+'\t'+d_year+'\t'+'r1')
```

Christian Craig
CSC 555 Project Phase 2

downloads — ec2-user@ip-172-31-71-141:~/hadoop-2.6.4 — ssh -i ccraig13.pem ec2-user@ec2-3-.

...-78.compute-1.amazonaws.com | ...8.compute-1.amazonaws.com | ...compute-1.amazonaws.com

GNU nano 2.9.8 final_reducer2.py

```
#!/usr/bin/python
import sys

currentKey = None
valsPt = None
valsR1 = None
TotalCount = 0

for line in sys.stdin:

    split = line.strip().split('\t')
    key = split[0]
    value = '\t'.join(split[1:])

    if currentKey == key: #same key
        if value.endswith('pt'):
            valsPt.append(value[:-2])
        if value.endswith('r1'):
            valsR1.append(value[:-2])
    else:
        if currentKey:
            lenPt = len(valsPt)
            lenR1 = len(valsR1)
            if (lenPt*lenR1>0):
                for p,r in zip(valsPt,valsR1):
                    print'%s\t%s'%(p,r)
        valsPt = []
        valsR1 = []

        currentKey = key
        if value.endswith('pt'):
            valsPt = [value[:-2]]
            valsR1 = []
        elif value.endswith('r1'):
            valsPt = []
            valsR1 = [value[:-2]]

lenPtLast = len(valsPt)
lenR1Last = len(valsR1)
if (lenPtLast*lenR1Last>0):
    for p,r in zip(valsPt,valsR1):
        print'%s\t%s'%(p,r)
```

```
hadoop fs -put part-00000 part.tbl /user/ec2-user/data/joinTbl_final_2
```

Christian Craig
CSC 555 Project Phase 2

```
hadoop jar hadoop-streaming-2.6.4.jar -input /user/ec2-user/data/joinTbl_final_2 -
mapper final_mapper2.py -file final_mapper2.py -reducer final_reducer2.py -file
final_reducer2.py -output /join3/output7777

hadoop fs -cat /join3/output7777/part-00000
```

```
                Bytes Written=474
21/03/19 21:03:00 INFO streaming.StreamJob: Output directory: /join3/output7777
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ hadoop fs -cat /join3/output7777/part-00000
MFGR#2227               4011621 1993
MFGR#2222               5858421 1993
MFGR#2225               1709969 1994
MFGR#2226               2733265 1996
MFGR#2223               3381664 1994
MFGR#2224               3872266 1993
MFGR#2227               239260  1993
MFGR#2227               3848423 1998
MFGR#2222               6259150 1996
MFGR#2223               4970613 1992
MFGR#2223               4471447 1993
MFGR#2225               2702501 1997
MFGR#2223               7328765 1995
MFGR#2223               4229583 1997
MFGR#2226               2025270 1997
MFGR#2223               6252081 1995
MFGR#2224               2699021 1995
MFGR#2225               2596400 1997
MFGR#2225               3794422 1995
```

Christian Craig
CSC 555 Project Phase 2

downloads — ec2-user@ip-172-31-71-141:~/hadoop-2.6.4 — ssh -i ccraig13.pem ec2-user@ec2-...

...-78.compute-1.amazonaws.com ....compute-1.amazonaws.com ...ompute-1.amazonaws.com

sum Done

GNU nano 2.9.8 final_mapper3.py

```
#!/usr/bin/python
import sys

#input comes from STDIN (standard input)
for line in sys.stdin:

    line = line.strip()
    split = line.split('\t')
    p_brand = split[0]
    lo_rev = split[2]
    d_year = split[3]
    print '%d\t%s\t%d'%(int(d_year),p_brand,int(lo_rev))
```

Christian Craig
CSC 555 Project Phase 2

downloads — ec2-user@ip-172-31-71-141:~/hadoop-2.6.4 — ssh -i ccraig13.pem ec2-user@ec2-...

...-78.compute-1.amazonaws.com ....compute-1.amazonaws.com ...ompute-1.amazonaws.com

sum Done

GNU nano 2.9.8 final_reducer3.py

```
#!/usr/bin/python

import sys
import statistics as st
currentKey = None
key= None
vals = []
val =None

# input comes from STDIN
for line in sys.stdin:

    split = line.strip().split('\t')
    #get key
    key = split[0:2]
    #value = '\t'.join(split[2:])
    val = split[2]
    if currentKey == key: # Same key
        vals.append(int(val))
    else:
        if currentKey: #get ouput
            total_rev =sum(vals)
            print currentKey,'\t',total_rev

        currentKey = key
        vals = []
        val = split[2]
        vals.append(int(val))

if len(vals)>0:
    total_rev  = sum(vals)
    print'%s\t%d'%(key,int(total_rev))
```

```
hadoop jar hadoop-streaming-2.6.4.jar -input /user/ec2-user/data/Tbl_final_3 -
mapper final_mapper3.py -file final_mapper3.py -reducer final_reducer3.py -file
final_reducer3.py -output /final_mr/output7777
```

```
hadoop fs -cat /final_mr/output7777/part-00000
```

```
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=223
                CPU time spent (ms)=1390
                Physical memory (bytes) snapshot=695910400
                Virtual memory (bytes) snapshot=6386876416
                Total committed heap usage (bytes)=509083648
        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=711
        File Output Format Counters
                Bytes Written=557
21/03/19 23:01:14 INFO streaming.StreamJob: Output directory: /final_mr/output7777
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ hadoop fs -cat /final_mr/output7777/part-00000
['1992', 'MFGR#2223']   4970613
['1993', 'MFGR#2227']   239260
['1993', 'MFGR#2224']   3872266
['1993', 'MFGR#2222']   5858421
['1993', 'MFGR#2227']   4011621
['1993', 'MFGR#2223']   4471447
['1994', 'MFGR#2223']   3381664
['1994', 'MFGR#2225']   1709969
['1995', 'MFGR#2225']   3794422
['1995', 'MFGR#2224']   2699021
['1995', 'MFGR#2223']   13580846
['1996', 'MFGR#2222']   6259150
['1996', 'MFGR#2226']   2733265
['1997', 'MFGR#2225']   2596400
['1997', 'MFGR#2226']   2025270
['1997', 'MFGR#2223']   4229583
['1997', 'MFGR#2225']   2702501
['1998', 'MFGR#2227']   3848423
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$
```

Part 3: Clustering

Using Hadoop streaming and randomly generated data (similar to what you did in Assignment6, but generate 1M rows and 9 columns of data)

- perform **five** KMeans iterations manually, **using 7 centers.** You can randomly choose the initial centers, such as by picking 7 random points from your data. For each of five iterations, include the centers produced by your code (i.e., **do not** submit the command line five times, without the corresponding output).
- This would require passing a text file with cluster centers using -file option as discussed in class, opening the centers.txt in the mapper with open('centers.txt', 'r') and assigning a key to each point based on which center is the closest to each particular point. Your reducer would then

compute the new centers by averaging the points, which would conclude the iteration. At that point, the output of the reducer with new centers can be given to the next pass of the same map reduce code using the -file option (you would need to get the output from HDFS into a local file for that).

- The only difference between first and subsequent iterations is that in first iteration you have to pick the initial centers. Starting from the 2nd iteration, the centers will be given to you by a previous pass of KMeans, and so on. Include the centers you computed at each iteration in your answer.

downloads — ec2-user@ip-172-31-71-141:~ — ssh -i ccraig13.pem ec2-user@ec2-3-230-166-233.co...

...-233.compute-1.amazonaws.com | ...15.compute-1.amazonaws.com | compute-1.amazonaws.com

sum Done

GNU nano 2.9.8 random_data.py

```
#!/usr/bin/python
import sys
import numpy as np
from numpy import random


num = random.random_integers(777, size=(1000000, 9))
np.savetxt('random_data_final.txt', num)
```

downloads — ec2-user@ip-172-31-71-141:~/hadoop-2.6.4 — ssh -i ccraig13.pem ec2-user@ec2-3-23

...35-40-212.compute-1.amazonaws.com | ...232-59-7.compute-1.amazonaws.com | ...216-16.compute-1.ama

GNU nano 2.9.8 centroid.txt

```
1       2.4     2.4     2.4     2.4     2.4     2.4     2.4     2.4     2.4
2       10.3    2.3     12.3    1.3     2.3     1.3     2.3     12.3    2.3
3       3.4     1.4     3.4     3.4     3.4     3.4     3.4     3.4     3.4
4       4.5     2.5     12.5    4.5     10.5    11.5    10.5    4.5     4.5
5       16.6    9.6     6.6     9.6     6.6     6.6     6.6     3.6     6.6
6       4.3     4.3     4.3     4.3     4.3     4.3     13.3    1.3     4.3
7       2.3     2.3     2.3     2.3     2.3     2.3     2.3     2.3     2.3
```

Christian Craig
CSC 555 Project Phase 2

...-i ccraig13.pem ec2-user@ec2-3-235-40-212.compute-1.amazonaws.com | ...-i ccraig13.pem ec2-user@ec2-18-232-59-7.compute-1.amazonaws.com | ...raig13.pem ec2-user@ec2-3-236-216-16.compute-1.amazonaws.com

GNU nano 2.9.8 finalmapperP3.py

```
#!/usr/bin/python
import sys
import numpy as np


fd_c = open('centroid.txt','r')
lines = fd_c.readlines()
cent_dict = {}
amt = 0
for ln in lines:
        split = ln.split('\t')
        cent_key = split[0]
        cent_val = split[1:]
        cent_dict[cent_key] = [cent_val[0],cent_val[1],cent_val[2],cent_val[3],cent_val[4],cent_val[5],cent_val[6],cent_val[7],cent_val[8].strip()] #remove '\n'

fd_c.close()



for line in sys.stdin:
        line = line.strip()
        split = line.split(' ')
        groups = split[0:]
        new_key = None
        groups = ','.join(groups)
        euc_lst = []
        for key,val in cent_dict.items():
                groups_f = map(float,groups.strip().split(','))
                for i in range(9): #0-8
                        euc_dist = np.sqrt((float(val[i]) - groups_f[i])**2).sum()
                        euc_lst.append([key,euc_dist])
        euc_arr = np.array(euc_lst)
        new_key = euc_arr[np.argmin(euc_arr[:,1])][0]
        print '%d\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f'%(int(new_key),float(split[0]),float(split[1]),float(split[2]),float(split[3]),float(split[4]) ,float(split[5]),float(split[6]),float(split[7]),float$
```

...-i ccraig13.pem ec2-user@ec2-3-235-40-212.compute-1.amazonaws.com | ...-i ccraig13.pem ec2-user@ec2-18-232-59-7.compute-1.amazonaws.com | ...raig13.pem ec2-user@ec2-3-236-216-16.cc

GNU nano 2.9.8 final_reducer_p31.py

```
#!/usr/bin/python

import sys
import statistics as st
currentKey = None
amt = 0
key = None

val1 = 0
val2 = 0
val3 = 0
val4 = 0
val5 = 0
val6 = 0
val7 = 0
val8 = 0
val9 = 0
# input comes from STDIN
for line in sys.stdin:

    split = line.strip().split('\t')
    #get key
    key = int(split[0])
    if currentKey==key: # Same key
        amt+=1
        val1 = val1+float(split[1])
        val2 = val2+float(split[2])
        val3 = val3+float(split[3])
        val4 = val4+float(split[4])
        val5 = val5+float(split[5])
        val6 = val6+float(split[6])
        val7 = val7+float(split[7])
        val8 = val8+float(split[8])
        val9 = val9+float(split[9])
    else:
        if currentKey:
                print '%d\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f'%(currentKey,(val1/amt),(val2/amt),(val3/amt),(val4/amt),(val5/amt),(val6/amt),(val7/amt),(val8/amt),(val9/amt))
        currentKey = key
        amt = 1
        val1 = 0
        val2 = 0
        val3 = 0
        val4 = 0
        val5 = 0
        val6 = 0
        val7 = 0
        val8 = 0
        val9 = 0

        val1 = val1+float(split[1])
        val2 = val2+float(split[2])
        val3 = val3+float(split[3])
        val4 = val4+float(split[4])
        val5 = val5+float(split[5])
        val6 = val6+float(split[6])
        val7 = val7+float(split[7])
        val8 = val8+float(split[8])
        val9 = val9+float(split[9])

if currentKey ==key:
        print '%d\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f'%(currentKey,(val1/amt),(val2/amt),(val3/amt),(val4/amt),(val5/amt),(val6/amt),(val7/amt),(val8/amt),(val9/amt))
```

Christian Craig
CSC 555 Project Phase 2

Iteration 1

hadoop jar hadoop-streaming-2.6.4.jar -input /user/ec2-user/data/kmeans_iter_1/random_data_final.txt -

mapper finalmapperP3.py -file finalmapperP3.py -reducer final_reducer_p31.py -file

final_reducer_p31.py -file centroid.txt -output /data/kmeans_test102

```
                Bytes Written=707
21/03/23 16:24:53 INFO streaming.StreamJob: Output directory: /data/kmeans_test102
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ hadoop fs -cat /data/kmeans_test102/part-00000
1       339.164650      396.392816      372.004394      336.346643      343.788175      371.121329      370.713944      370.098127      339.343791
2       376.140428      414.159784      412.023296      408.325978      412.011436      414.254580      413.446226      222.576738      407.135317
3       356.333692      357.418990      370.045482      356.104194      355.637601      358.592151      310.724672      376.933724      355.147808
4       416.157762      417.122359      352.036310      409.228281      349.217841      348.469799      414.876641      405.344238      413.952923
5       368.206135      364.089277      412.376890      364.888002      414.304745      414.481844      416.752628      421.786642      361.987325
6       417.741403      382.489973      400.051282      414.337956      400.752159      399.884437      231.166555      411.837614      411.141180
7       275.683394      387.524468      281.089035      334.961260      392.152583      337.545310      387.695514      336.180109      391.210807
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$
```

My centroid.txt file was not able to replace properly, as I used “mv” incorrectly. When I use -get its gets the one I accidently moved to local. The remainder of the iterations use centroid_take2.txt. I adjusted the mapper for this.

I understand conceptually you are using the last output as the text and not supposed to adjust the mapper. But my mapreduce takes 10 – 15 minutes to run, and running out of time. Iterations 2-5 should use the same.

Gaps between kmeans_test# because node failures.

hadoop fs -cat /data/kmeans_test102/part-00000

hadoop fs -mv /data/kmeans_test102/part-00000 centroid_take2.txt

hadoop fs -get centroid_take2.txt

Iteration 2

hadoop jar hadoop-streaming-2.6.4.jar -input /user/ec2-user/data/kmeans_iter_1/random_data_final.txt -

mapper finalmapperP3.py -file finalmapperP3.py -reducer final_reducer_p31.py -file

final_reducer_p31.py -file centroid_take2.txt -output /data/kmeans_test104

```
21/03/23 16:50:35 INFO streaming.StreamJob: Output directory: /data/kmeans_test104
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ hadoop fs -cat /data/kmeans_test104/part-00000
1       384.255331      391.863340      394.680948      389.617004      382.448066      389.490038      390.181934      387.203562      382.506425
2       394.355119      389.050650      388.536492      387.111737      388.358299      388.450445      392.024876      361.472359      389.909037
3       387.901795      382.653900      387.230795      384.875327      388.276867      387.545465      380.541192      394.064695      384.757909
4       389.536121      396.795723      382.960140      393.558037      385.986068      386.446591      395.096709      393.581486      399.090217
5       388.840943      388.183135      399.635062      389.598379      397.745080      398.625531      401.397725      401.324819      389.547200
6       399.746634      385.803104      392.461425      396.509618      392.228999      390.645351      367.850031      395.896733      391.026480
7       378.451713      390.678800      378.211939      382.906133      389.434860      383.308187      394.582540      386.837314      390.326221
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$
```

hadoop fs -cat /data/kmeans_test108/part-00000

mv centroid_take2.txt centroid_take2_1.txt

hadoop fs -mv /data/kmeans_test104/part-00000 centroid_ph.txt

hadoop fs -get centroid_ph.txt

mv centroid_ph.txt centroid_take2.txt

Christian Craig
CSC 555 Project Phase 2

```
cat: centroids_take2.txt: No such file or directory
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ cat centroid_take2.txt
1       384.255331      391.863340      394.680948      389.617004      382.448066      389.490038      390.181934      387.203562      382.506425
2       394.355119      389.050650      388.536492      387.111737      388.358299      388.450445      392.024876      361.472359      389.989037
3       387.901795      382.653900      387.230795      384.875327      388.276867      387.545465      380.541192      394.064695      384.757909
4       389.536121      396.795723      382.960140      393.558037      385.986068      386.446591      395.096709      393.581486      399.090217
5       388.840943      388.183135      399.635062      389.598379      397.745080      398.625531      401.397725      401.324819      389.547200
6       399.746634      385.803104      392.461425      396.509618      392.228999      390.645351      367.850031      395.896733      391.026480
7       378.451713      390.678800      378.211939      382.906133      389.434860      383.308187      394.582540      386.837314      390.326221
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$
```

Iteration 3

hadoop jar hadoop-streaming-2.6.4.jar -input /user/ec2-user/data/kmeans_iter_1/random_data_final.txt -mapper finalmapperP3.py -file finalmapperP3.py -reducer final_reducer_p31.py -file final_reducer_p31.py -file centroid_take2.txt -output /data/kmeans_test108

hadoop fs -cat /data/kmeans_test108/part-00000

mv centroid_take2.txt centroid_take2_2.txt

hadoop fs -mv /data/kmeans_test108/part-00000 centroid_ph2.txt

hadoop fs -get centroid_ph2.txt

mv centroid_ph2.txt centroid_take2.txt

```
21/03/23 17:36:08 INFO streaming.StreamJob: Output directory: /data/kmeans_test108
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ hadoop fs -cat /data/kmeans_test108/part-00000
1       385.930723      391.320010      390.671417      391.964220      367.911173      388.495374      385.874905      392.603064      368.733076
2       389.105941      388.686582      391.277806      387.291576      389.500729      389.663136      390.314460      360.832521      390.623815
3       388.185157      363.733830      386.771732      388.394434      386.236537      387.737469      384.866223      388.546113      391.779933
4       391.993405      407.715430      388.371365      388.489608      389.802594      387.269706      391.500994      387.516185      407.572356
5       388.554484      387.826567      403.375186      388.440587      402.568044      401.727099      403.158242      405.095699      386.593222
6       404.467412      388.401401      388.760539      404.822228      389.992019      392.526519      372.586835      392.633419      391.883929
7       372.086590      390.022428      372.116679      373.305150      390.436418      373.698127      388.699923      382.431303      388.338843
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$
```

```
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ cat centroid_take2.txt
1       385.930723      391.320010      390.671417      391.964220      367.911173      388.495374      385.874905      392.603064      368.733076
2       389.105941      388.686582      391.277806      387.291576      389.500729      389.663136      390.314460      360.832521      390.623815
3       388.185157      363.733830      386.771732      388.394434      386.236537      387.737469      384.866223      388.546113      391.779933
4       391.993405      407.715430      388.371365      388.489608      389.802594      387.269706      391.500994      387.516185      407.572356
5       388.554484      387.826567      403.375186      388.440587      402.568044      401.727099      403.158242      405.095699      386.593222
6       404.467412      388.401401      388.760539      404.822228      389.992019      392.526519      372.586835      392.633419      391.883929
7       372.086590      390.022428      372.116679      373.305150      390.436418      373.698127      388.699923      382.431303      388.338843
```

Christian Craig
CSC 555 Project Phase 2

Iteration 4

hadoop jar hadoop-streaming-2.6.4.jar -input /user/ec2-user/data/kmeans_iter_1/random_data_final.txt -mapper finalmapperP3.py -file finalmapperP3.py -reducer final_reducer_p31.py -file final_reducer_p31.py -file centroid_take2.txt -output /data/kmeans_test113

hadoop fs -cat /data/kmeans_test113/part-00000

mv centroid_take2.txt centroid_take2_3.txt

hadoop fs -mv /data/kmeans_test113/part-00000 centroid_ph3.txt

hadoop fs -get centroid_ph3.txt

mv centroid_ph3.txt centroid_take2.txt

```
                Bytes Written=707
21/03/23 18:00:47 INFO streaming.StreamJob: Output directory: /data/kmeans_test113
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ hadoop fs -cat /data/kmeans_test113/part-00000
1       383.858960      394.949478      387.826876      393.084820      377.332288      388.950638      390.187644      387.675869      376.764844
2       389.121838      391.628914      396.102285      382.209071      387.130850      391.761384      389.337991      368.856839      390.065010
3       387.123109      371.827711      382.938981      387.242069      382.714836      390.744026      381.337593      391.431972      388.815242
4       394.467661      403.139409      387.204265      390.144604      388.872204      381.597288      394.068589      387.457432      403.597796
5       388.970860      384.042195      398.423252      388.552310      398.184119      397.881031      397.126233      399.952476      384.774112
6       401.274449      388.442850      391.272422      402.268509      389.852395      392.103267      375.895489      395.325376      395.514986
7       378.804314      389.416810      379.592872      380.103977      393.392222      380.882495      388.165627      385.196402      388.739393
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$
```

```
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ mv centroid_take2.txt centroid_take2_3.txt
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ hadoop fs -mv /data/kmeans_test113/part-00000  centroid_ph3.txt
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ hadoop fs -get centroid_ph3.txt
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ mv centroid_ph3.txt centroid_take2.txt
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ cat centroid_take2.txt
1       383.858960      394.949478      387.826876      393.084820      377.332288      388.950638      390.187644      387.675869      376.764844
2       389.121838      391.628914      396.102285      382.209071      387.130850      391.761384      389.337991      368.856839      390.065010
3       387.123109      371.827711      382.938981      387.242069      382.714836      390.744026      381.337593      391.431972      388.815242
4       394.467661      403.139409      387.204265      390.144604      388.872204      381.597288      394.068589      387.457432      403.597796
5       388.970860      384.042195      398.423252      388.552310      398.184119      397.881031      397.126233      399.952476      384.774112
6       401.274449      388.442850      391.272422      402.268509      389.852395      392.103267      375.895489      395.325376      395.514986
7       378.804314      389.416810      379.592872      380.103977      393.392222      380.882495      388.165627      385.196402      388.739393
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$
```

Iteration 5

hadoop jar hadoop-streaming-2.6.4.jar -input /user/ec2-user/data/kmeans_iter_1/random_data_final.txt -mapper finalmapperP3.py -file finalmapperP3.py -reducer final_reducer_p31.py -file final_reducer_p31.py -file centroid_take2.txt -output /data/kmeans_test116

Christian Craig
CSC 555 Project Phase 2

hadoop fs -cat /data/kmeans_test116/part-00000

mv centroid_take2.txt centroid_take2_4.txt

hadoop fs -mv /data/kmeans_test116/part-00000 centroid_ph4.txt

hadoop fs -get centroid_ph4.txt

mv centroid_ph4.txt centroid_take2.txt

```
21/03/23 18:19:14 INFO streaming.StreamJob: Output directory: /data/kmeans_test116
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ hadoop fs -cat /data/kmeans_test116/part-00000
1       386.310439      392.609372      391.698239      390.503225      371.152254      386.512317      391.089577      389.789489      371.272996
2       391.457840      390.103523      387.034853      389.534282      385.271519      390.722447      389.852144      366.274097      391.618192
3       387.085264      369.900157      390.738679      386.429506      388.699950      388.409968      387.530383      389.586340      390.993286
4       391.190665      409.349866      385.122833      388.980173      389.415340      390.590919      388.226072      388.027476      409.299332
5       388.635971      386.266039      402.240082      389.344551      401.572569      401.038407      401.611274      404.172385      387.253254
6       402.833287      387.097819      390.015934      403.923253      390.721014      390.899881      374.124904      390.121669      391.377270
7       373.719441      390.225566      373.210666      374.066629      390.447844      373.819256      385.820704      384.791855      386.725624
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$
```

```
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$ cat centroid_take2.txt
1       386.310439      392.609372      391.698239      390.503225      371.152254      386.512317      391.089577      389.789489      371.272996
2       391.457840      390.103523      387.034853      389.534282      385.271519      390.722447      389.852144      366.274097      391.618192
3       387.085264      369.900157      390.738679      386.429506      388.699950      388.409968      387.530383      389.586340      390.993286
4       391.190665      409.349866      385.122833      388.980173      389.415340      390.590919      388.226072      388.027476      409.299332
5       388.635971      386.266039      402.240082      389.344551      401.572569      401.038407      401.611274      404.172385      387.253254
6       402.833287      387.097819      390.015934      403.923253      390.721014      390.899881      374.124904      390.121669      391.377270
7       373.719441      390.225566      373.210666      374.066629      390.447844      373.819256      385.820704      384.791855      386.725624
[ec2-user@ip-172-31-71-141 hadoop-2.6.4]$
```

<u>Submit a single document containing your written answers. Be sure that this document contains your name and “CSC 555 Project Phase 2” at the top.</u>