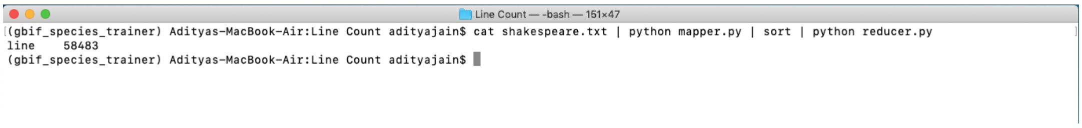
Assignment 1 Report

Aditya Jain

1 Line Count

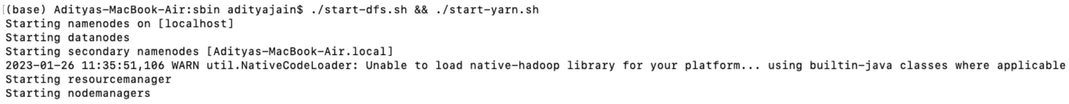
Figure 1 shows the line count result when the code is run on the local system. The count is **58,483**.



```
(gbif_species_trainer) Adityas-MacBook-Air:Line Count adityajain$ cat shakespeare.txt | python mapper.py | sort | python reducer.py
line    58483
(gbif_species_trainer) Adityas-MacBook-Air:Line Count adityajain$ 
```

Figure 1: Result on local machine

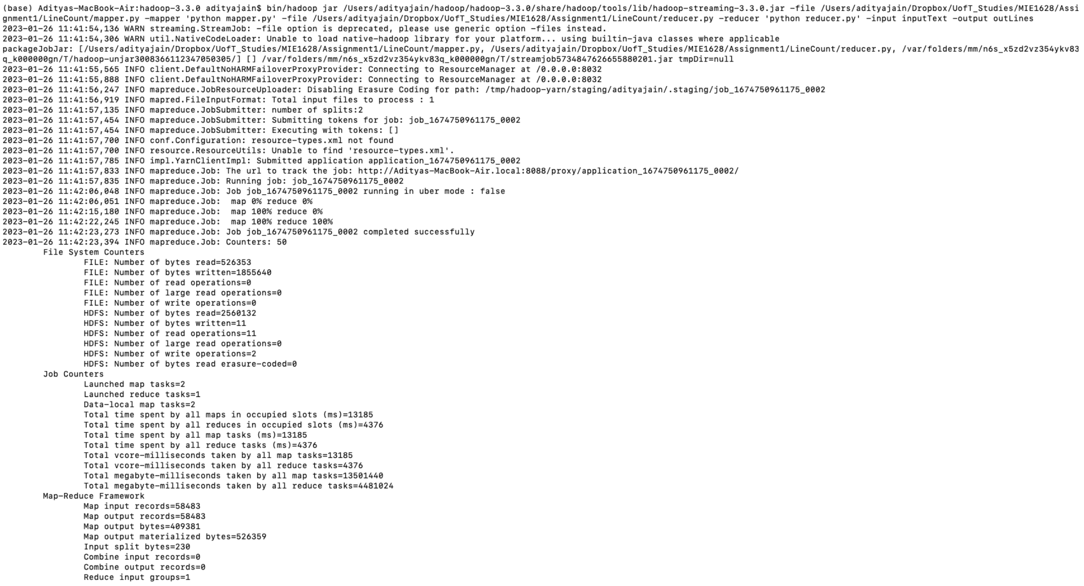
Figure 2 shows the launch of DFS and YARN daemons.



```
(base) Adityas-MacBook-Air:sbin adityajain$ ./start-dfs.sh && ./start-yarn.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [Adityas-MacBook-Air.local]
2023-01-26 11:35:51,106 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting resourcemanager
Starting nodemanagers
```

Figure 2: Start DFS and YARN daemon

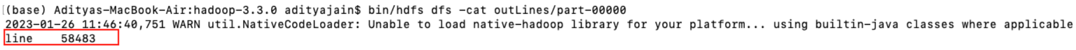
Figure 3 shows the screenshot of MapReduce task running on the Hadoop platform.



```
(base) Adityas-MacBook-Air:hadoop-3.3.0 adityajain$ bin/hadoop jar /Users/adityajain/hadoop/hadoop-3.3.0/share/hadoop/tools/lib/hadoop-streaming-3.3.0.jar -file /Users/adityajain/Dropbox/UofT_Studies/MIE1628/Assi
gnment1/LineCount/mapper.py -mapper 'python mapper.py' -file /Users/adityajain/Dropbox/UofT_Studies/MIE1628/Assignment1/LineCount/reducer.py -reducer 'python reducer.py' -input inputText -output outLines
2023-01-26 11:41:54,136 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
2023-01-26 11:41:54,306 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
packageJobJar: [/Users/adityajain/Dropbox/UofT_Studies/MIE1628/Assignment1/LineCount/mapper.py, /Users/adityajain/Dropbox/UofT_Studies/MIE1628/Assignment1/LineCount/reducer.py, /var/folders/mm/n6s_x5zd2vz354ykv83
q_k000000gn/T/hadoop-unjar3008366112347050305/] [] /var/folders/mm/n6s_x5zd2vz354ykv83q_k000000gn/T/streamjob5734847626655880201.jar tmpDir=null
2023-01-26 11:41:55,565 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2023-01-26 11:41:55,888 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2023-01-26 11:41:56,247 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/adityajain/.staging/job_1674750961175_0002
2023-01-26 11:41:56,919 INFO mapred.FileInputFormat: Total input files to process : 1
2023-01-26 11:41:57,135 INFO mapreduce.JobSubmitter: number of splits:2
2023-01-26 11:41:57,454 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1674750961175_0002
2023-01-26 11:41:57,454 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-01-26 11:41:57,700 INFO conf.Configuration: resource-types.xml not found
2023-01-26 11:41:57,700 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2023-01-26 11:41:57,785 INFO impl.YarnClientImpl: Submitted application application_1674750961175_0002
2023-01-26 11:41:57,833 INFO mapreduce.Job: The url to track the job: http://Adityas-MacBook-Air.local:8088/proxy/application_1674750961175_0002/
2023-01-26 11:41:57,835 INFO mapreduce.Job: Running job: job_1674750961175_0002
2023-01-26 11:42:06,048 INFO mapreduce.Job: Job job_1674750961175_0002 running in uber mode : false
2023-01-26 11:42:06,051 INFO mapreduce.Job:  map 0% reduce 0%
2023-01-26 11:42:15,180 INFO mapreduce.Job:  map 100% reduce 0%
2023-01-26 11:42:22,245 INFO mapreduce.Job:  map 100% reduce 100%
2023-01-26 11:42:23,273 INFO mapreduce.Job: Job job_1674750961175_0002 completed successfully
2023-01-26 11:42:23,394 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=526353
                FILE: Number of bytes written=1855640
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=2560132
                HDFS: Number of bytes written=11
                HDFS: Number of read operations=11
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=13185
                Total time spent by all reduces in occupied slots (ms)=4376
                Total time spent by all map tasks (ms)=13185
                Total time spent by all reduce tasks (ms)=4376
                Total vcore-milliseconds taken by all map tasks=13185
                Total vcore-milliseconds taken by all reduce tasks=4376
                Total megabyte-milliseconds taken by all map tasks=13501440
                Total megabyte-milliseconds taken by all reduce tasks=4481024
        Map-Reduce Framework
                Map input records=58483
                Map output records=58483
                Map output bytes=409381
                Map output materialized bytes=526359
                Input split bytes=230
                Combine input records=0
                Combine output records=0
                Reduce input groups=1
```

Figure 3: MapReduce task on Hadoop

Figure 4 shows the result of MapReduce task, which is again **58,483**.

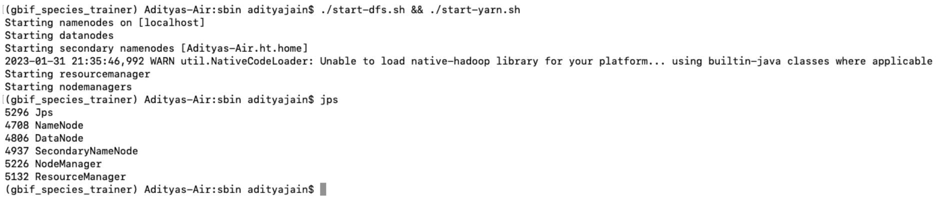


```
(base) Adityas-MacBook-Air:hadoop-3.3.0 adityajain$ bin/hdfs dfs -cat outLines/part-00000
2023-01-26 11:46:40,751 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
line    58483
```

Figure 4: MapReduce task result

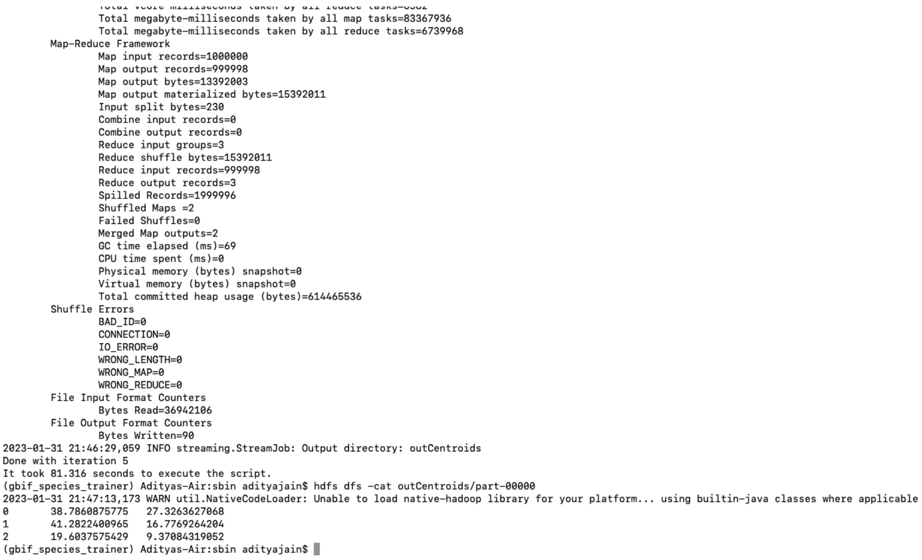
2 K-means clustering

Figure 5 shows the launch of DFS and YARN daemons. Figure 6 and figure 7 shows Hadoop running k-means for 3 and 6 clusters respectively.



```
(gbif_species_trainer) Adityas-Air:sbin adityajain$ ./start-dfs.sh && ./start-yarn.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [Adityas-Air.ht.home]
2023-01-31 21:35:46,992 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting resourcemanager
Starting nodemanagers
(gbif_species_trainer) Adityas-Air:sbin adityajain$ jps
5296 Jps
4708 NameNode
4806 DataNode
4937 SecondaryNameNode
5226 NodeManager
5132 ResourceManager
(gbif_species_trainer) Adityas-Air:sbin adityajain$ 
```

Figure 5: Start DFS and YARN daemon



```
                Total vcore-milliseconds taken by all reduce tasks=6582
                Total megabyte-milliseconds taken by all map tasks=83367936
                Total megabyte-milliseconds taken by all reduce tasks=6739968
        Map-Reduce Framework
                Map input records=1000000
                Map output records=999998
                Map output bytes=13392003
                Map output materialized bytes=15392011
                Input split bytes=230
                Combine input records=0
                Combine output records=0
                Reduce input groups=3
                Reduce shuffle bytes=15392011
                Reduce input records=999998
                Reduce output records=3
                Spilled Records=1999996
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=69
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=614465536
        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=36942106
        File Output Format Counters
                Bytes Written=90
2023-01-31 21:46:29,059 INFO streaming.StreamJob: Output directory: outCentroids
Done with iteration 5
It took 81.316 seconds to execute the script.
(gbif_species_trainer) Adityas-Air:sbin adityajain$ hdfs dfs -cat outCentroids/part-00000
2023-01-31 21:47:13,173 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
0       38.7860875775   27.3263627068
1       41.2822400965   16.7769264204
2       19.6037575429   9.37084319052
(gbif_species_trainer) Adityas-Air:sbin adityajain$ 
```

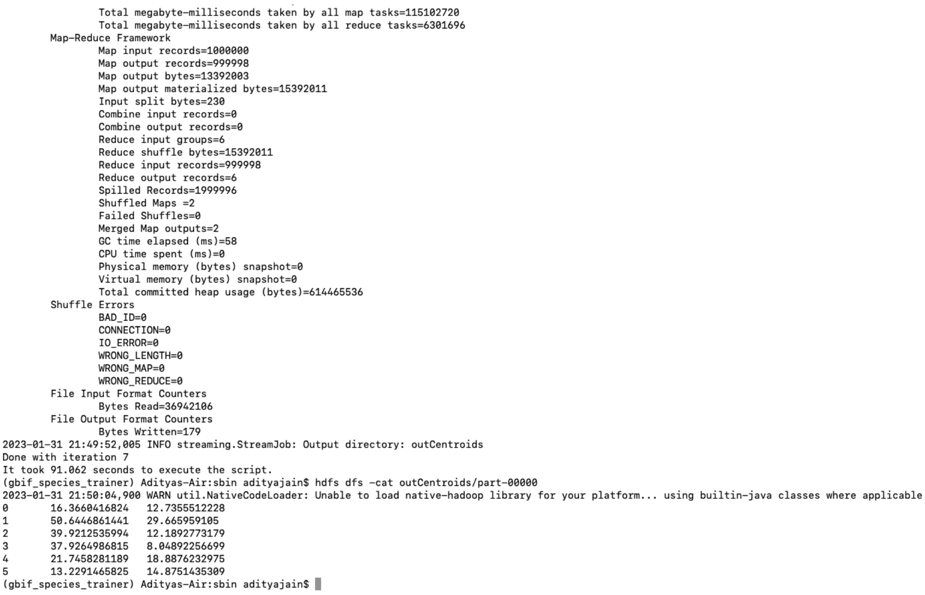
Figure 6: Running of k-means with MapReduce for k=3

The final cluster centroids for k=3 are:

- **Centroid 1**: 9.95, 15.10
- **Centroid 2**: 50.00, 30.10
- **Centroid 3**: 35.00, 1.78

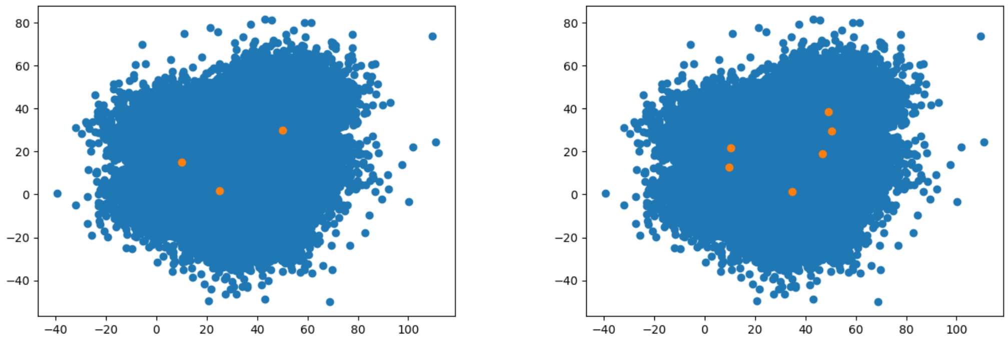
The final cluster centroids for k=6 are:

- **Centroid 1**: 49.22, 38.41
- **Centroid 2**: 9.80, 12.86
- **Centroid 3**: 10.54, 21.73
- **Centroid 4**: 34.83, 1.42
- **Centroid 5**: 46.79, 18.97
- **Centroid 6**: 50.44, 29.55



```
                Total megabyte-milliseconds taken by all map tasks=115102720
                Total megabyte-milliseconds taken by all reduce tasks=6301696
        Map-Reduce Framework
                Map input records=1000000
                Map output records=999998
                Map output bytes=13392003
                Map output materialized bytes=15392011
                Input split bytes=230
                Combine input records=0
                Combine output records=0
                Reduce input groups=6
                Reduce shuffle bytes=15392011
                Reduce input records=999998
                Reduce output records=6
                Spilled Records=1999996
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=58
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=614465536
        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=36942106
        File Output Format Counters
                Bytes Written=179
2023-01-31 21:49:52,005 INFO streaming.StreamJob: Output directory: outCentroids
Done with iteration 7
It took 91.062 seconds to execute the script.
(gbif_species_trainer) Adityas-Air:sbin adityajain$ hdfs dfs -cat outCentroids/part-00000
2023-01-31 21:50:04,900 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
0       16.3660416824   12.7355512228
1       50.6446861441   29.665959105
2       39.9212535994   12.1892773179
3       37.9264986815   8.04892256699
4       21.7458281189   18.8876232975
5       13.2291465825   14.8751435309
(gbif_species_trainer) Adityas-Air:sbin adityajain$ 
```

Figure 7: Running of k-means with MapReduce for k=6



(a) k=3

(b) k=6

Figure 8: Cluster centroids superimposed over raw data points.

Number of clusters (k)	Standard k-means	k-means with MapReduce
3	171 sec, 4 iterations	15 sec, 5 iterations
6	500 sec, 7 iterations	40 sec, 7 iterations

Table 1: Time and iteration comparison between the two methods.

Figure 8 shows the clustering result using k-means with MapReduce. Table 1 shows the time and iteration comparison for the two methods of implementing k-means.

As seen in table 1, the advantage of implementing k-means clustering with MapReduce is of significant reduction in computation time. However, MapReduce stores all intermediate results on the disk which takes up a lot of disk space and can be expensive for large datasets.

3 Canopy Selection Questions

1. Yes, we can reduce the comparisons by using the canopy selection method. For the given clustering problem of 2D data points, we can form canopies of points that lie in a given patch on the 2D plane. For example, we will form a canopy for all points lying in the range [0, 5] on the x-axis and [0, 5] on the y-axis. Now, for a centroid in this canopy, only points lying in this patch will be tested for the euclidean distance instead of all the points.

2. Yes, we can use canopy selection on MapReduce. The mapper function will be used to create the canopies and the reducer will do the remainder clustering through K-means, Greedy Agglomerative Clustering (GAC), or Expectation-Maximization (EM).

3. Yes, we can combine canopy selection with K-means on MapReduce. The mapper function will form the canopies, an intermediate combiner/function will do the cluster assignment within the canopies, and the reducer will perform the averaging to calculate new cluster centroids.