LARGE SCALE DATA ANALYSIS USING MAPREDUCE

ASSIGNMENT 1

EC7205: CLOUD COMPUTING

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

FACULTY OF ENGINEERING

UNIVERSITY OF RUHUNA

EG/2020/3797 - Aberuwan R. M. M. P.
EG/2020/4190 - Sandamal I.M.K.
EG/2020/4219 - Siriwardhana T.D.R.D.

Submitted Date: 7th June 2025

CONTENTS

1 Introduction 1
2 Dataset Description.. 1
3 Problem Definition 1
4 Environment Setup 1
5 Implementation... 8
6 Execution & Testing... 8
7 Results & Interpretation 13
8 Conclusion.. 14
9 References 14

LIST OF FIGURES

Figure 1: Evidence for Java Version 2

Figure 2: Evidence for Hadoop Directory Structure 3

Figure 3: Evidence for SSH Setup 3

Figure 4: Evidence for Environment Variables 4

Figure 5: Evidence for Configuration Files 5

Figure 6: Evidence for Namenode Formatting 5

Figure 7: Evidence for Starting Hadoop Service 6

Figure 8: Evidence for Running Processes 6

Figure 9: Evidence for NameNode UI: http://localhost:9870 7

Figure 10: Evidence for ResourceManager UI: http://localhost:8088 7

Figure 11: Evidence for HDFS Commands 8

Figure 12: Evidence for Uploaded the Spotify CSV Dataset to HDFS 9

Figure 13: Evidence for Made Sure both Scripts were Executable and Had the Correct Shebang. 9

Figure 14: Evidence for Ran the Hadoop Streaming Job 10

Figure 15: Evidence for Retrieved and Viewed the Output 11

Figure 16: Evidence for Input 12

Figure 17: Evidence for Output 13

1 Introduction

The aim of this assignment is to implement a custom MapReduce job using Hadoop to process the Spotify dataset and extract meaningful insights related to artist popularity, genre trends, and follower distribution.

2 Dataset Description

- Dataset: Spotify Datasets (Kaggle) [1]
  https://www.kaggle.com/datasets/lehaknarnauli/spotify-datasets?select=artists.csv
- Size: 1104349 rows, 5 columns
- Features Used: id, followers, genres, name, popularity

This dataset contains metadata about Spotify artists, including unique IDs, follower counts, associated genres, artist names, and a popularity score. The dataset is suitable for large-scale distributed analysis due to its size and structure.

3 Problem Definition

Analyze the Spotify dataset to answer the following questions using MapReduce:

- What is the average number of followers for artists at each popularity score?
- How does follower count correlate with popularity scores?

Chosen MapReduce Task: **Popularity-Follower Aggregation**

- Map: For each artist, emit (popularity_score, followers)
- Reduce: For each popularity score, compute:
  - Total followers across all artists with that score
  - Average followers per popularity score

Mapper Logic:

- Your mapper.py extracts popularity (as the key) and followers (as the value) from each artist.
- Example output: 0 52.68 (popularity=0, followers=52.68).

Reducer Logic:

- Your reducer.py groups data by popularity and calculates the average followers for each score.
- Example output: 0 52.68 (average followers for popularity=0).

4 Environment Setup

Platform:

- Local Linux machine (single-node setup)

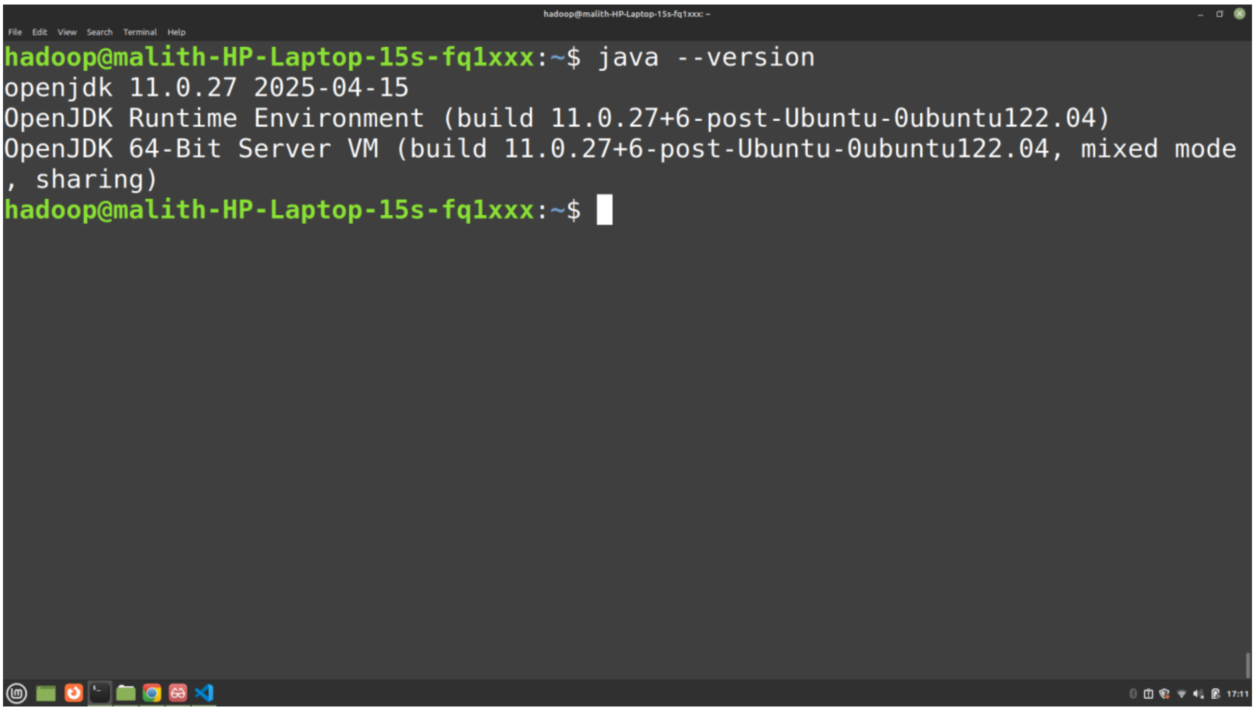
Installation Steps Overview:

1. Install Java:
   sudo apt-get install default-jdk
2. Create Hadoop User:
   sudo adduser hadoop

3. Set Up SSH:
   ssh-keygen -t rsa and enable passwordless SSH with cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
4. Download Hadoop:
   Download and extract Hadoop from Apache Hadoop.
5. Set Environment Variables:
   Add Hadoop and Java paths to ~/.bashrc.
6. Configure Hadoop:
   Edit config files in /usr/local/hadoop/etc/hadoop/.
7. Format NameNode:
   hdfs namenode -format
8. Start Hadoop:
   start-dfs.sh and start-yarn.sh
9. Verify:
   Use jps and access web UIs at http://localhost:9870 (NameNode) and http://localhost:8088 (ResourceManager).

Screenshots for Evidence:

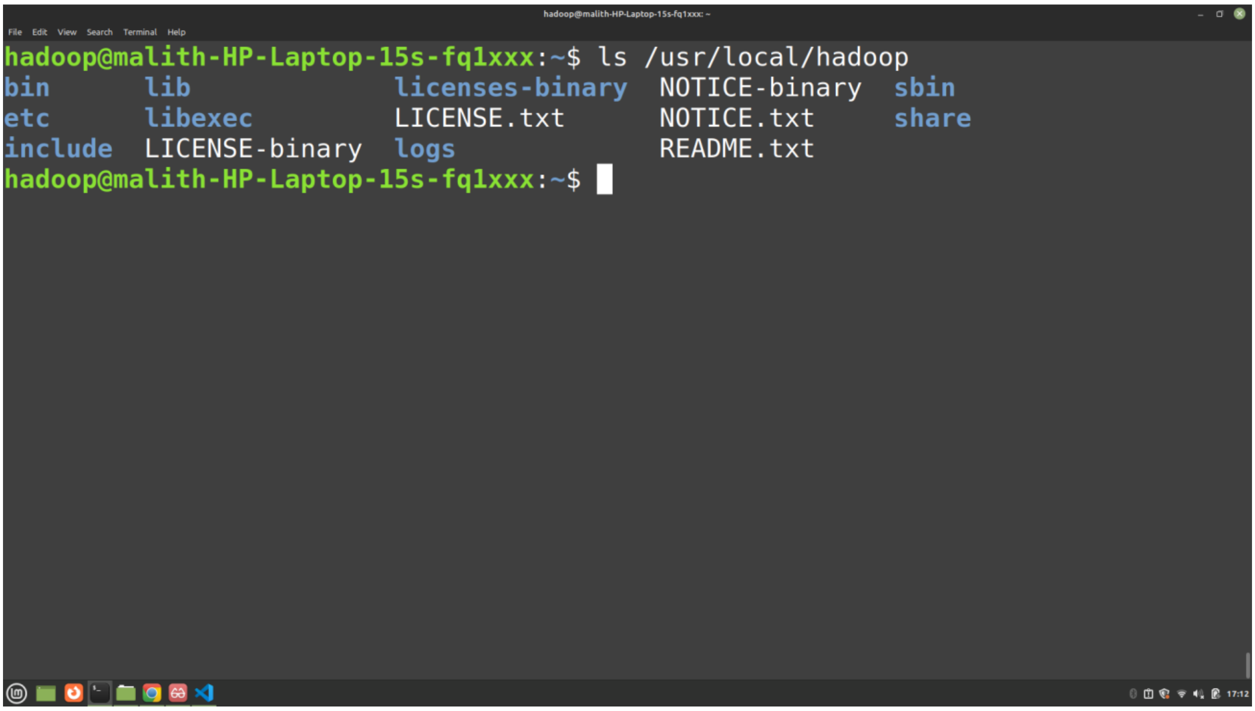
JavaVersion:


hadoop@malith-HP-Laptop-15s-fq1xxx: ~
File Edit View Search Terminal Help
17:11

```
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ java --version
openjdk 11.0.27 2025-04-15
OpenJDK Runtime Environment (build 11.0.27+6-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.27+6-post-Ubuntu-0ubuntu122.04, mixed mode
, sharing)
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ 
```

FIGURE 1: EVIDENCE FOR JAVA VERSION

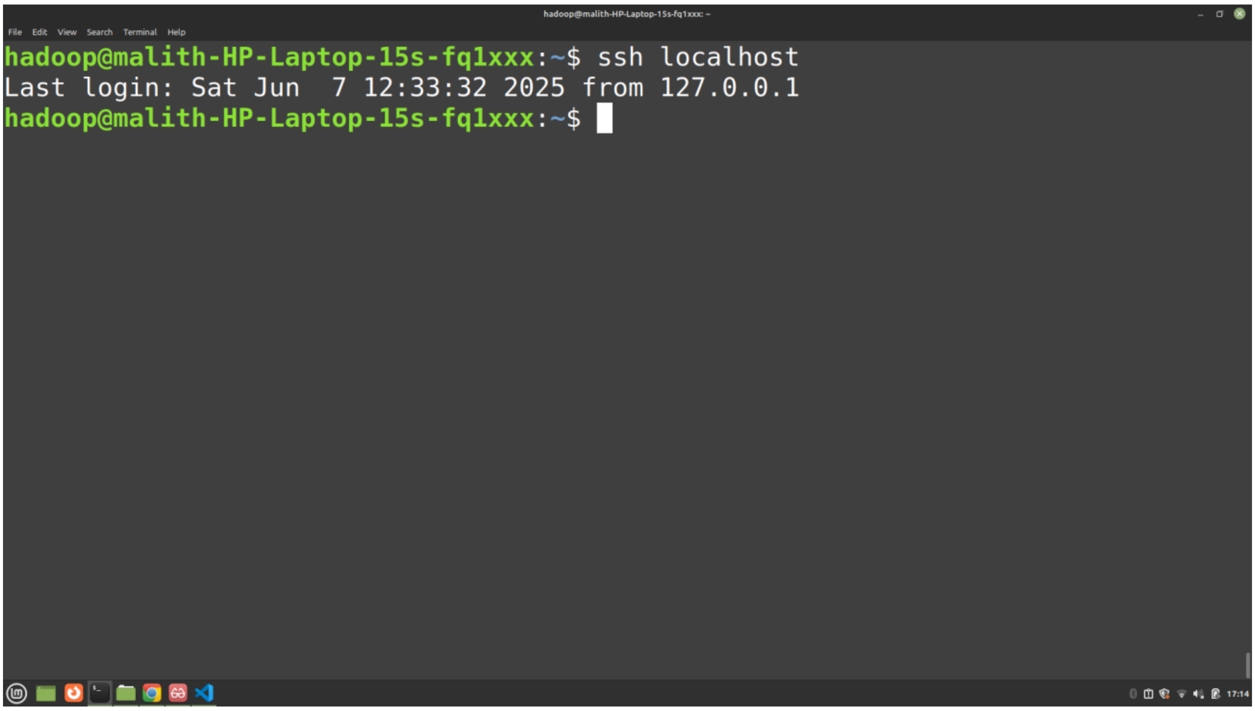
Hadoop Directory Structure:



```
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ ls /usr/local/hadoop
bin      lib             licenses-binary  NOTICE-binary  sbin
etc      libexec         LICENSE.txt      NOTICE.txt     share
include  LICENSE-binary  logs             README.txt
hadoop@malith-HP-Laptop-15s-fq1xxx:~$
```

FIGURE 2: EVIDENCE FOR HADOOP DIRECTORY STRUCTURE

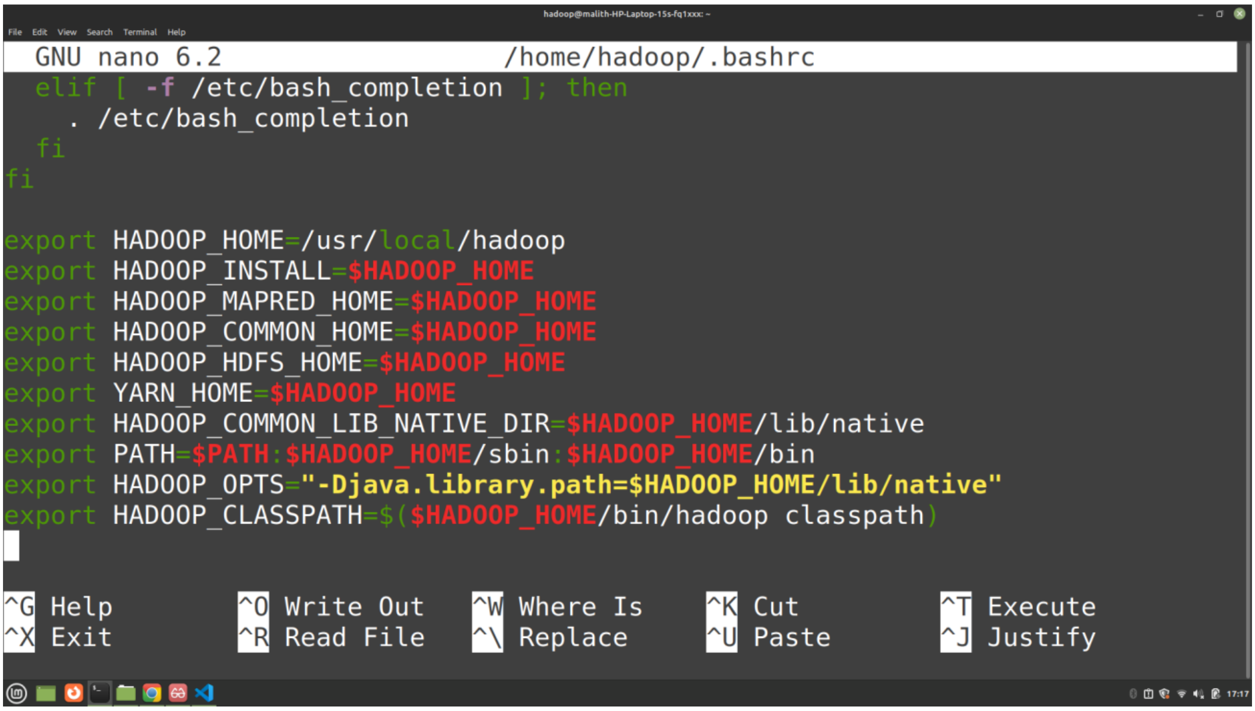
SSH Setup:



```
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ ssh localhost
Last login: Sat Jun  7 12:33:32 2025 from 127.0.0.1
hadoop@malith-HP-Laptop-15s-fq1xxx:~$
```

FIGURE 3: EVIDENCE FOR SSH SETUP

Environment Variables:


hadoop@malith-HP-Laptop-15s-fq1xxx: ~
File Edit View Search Terminal Help

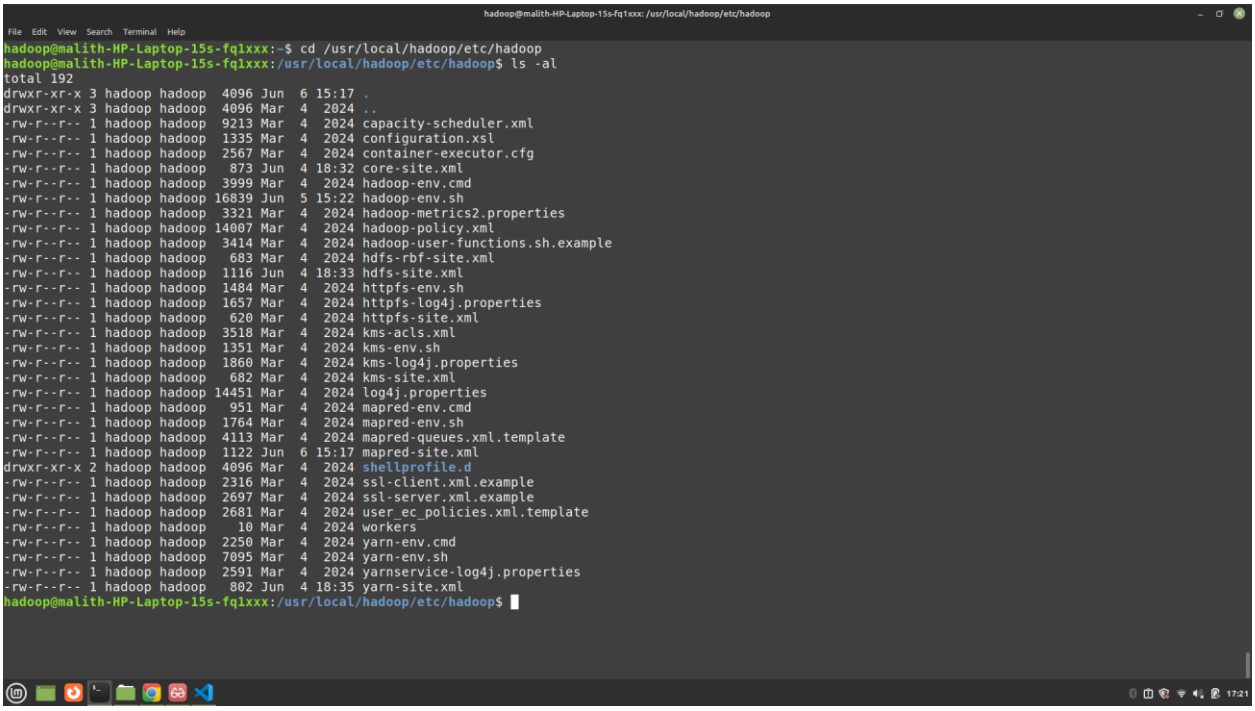
```
  GNU nano 6.2                    /home/hadoop/.bashrc
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export HADOOP_HOME=/usr/local/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
export HADOOP_CLASSPATH=$($HADOOP_HOME/bin/hadoop classpath)


^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify
```

FIGURE 4: EVIDENCE FOR ENVIRONMENT VARIABLES

Configuration Files:


hadoop@malith-HP-Laptop-15s-fq1xxx: /usr/local/hadoop/etc/hadoop
File Edit View Search Terminal Help

```
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ cd /usr/local/hadoop/etc/hadoop
hadoop@malith-HP-Laptop-15s-fq1xxx:/usr/local/hadoop/etc/hadoop$ ls -al
total 192
drwxr-xr-x 3 hadoop hadoop  4096 Jun  6 15:17 .
drwxr-xr-x 3 hadoop hadoop  4096 Mar  4  2024 ..
-rw-r--r-- 1 hadoop hadoop  9213 Mar  4  2024 capacity-scheduler.xml
-rw-r--r-- 1 hadoop hadoop  1335 Mar  4  2024 configuration.xsl
-rw-r--r-- 1 hadoop hadoop  2567 Mar  4  2024 container-executor.cfg
-rw-r--r-- 1 hadoop hadoop   873 Jun  4 18:32 core-site.xml
-rw-r--r-- 1 hadoop hadoop  3999 Mar  4  2024 hadoop-env.cmd
-rw-r--r-- 1 hadoop hadoop 16839 Jun  5 15:22 hadoop-env.sh
-rw-r--r-- 1 hadoop hadoop  3321 Mar  4  2024 hadoop-metrics2.properties
-rw-r--r-- 1 hadoop hadoop 14007 Mar  4  2024 hadoop-policy.xml
-rw-r--r-- 1 hadoop hadoop  3414 Mar  4  2024 hadoop-user-functions.sh.example
-rw-r--r-- 1 hadoop hadoop   683 Mar  4  2024 hdfs-rbf-site.xml
-rw-r--r-- 1 hadoop hadoop  1116 Jun  4 18:33 hdfs-site.xml
-rw-r--r-- 1 hadoop hadoop  1484 Mar  4  2024 httpfs-env.sh
-rw-r--r-- 1 hadoop hadoop  1657 Mar  4  2024 httpfs-log4j.properties
-rw-r--r-- 1 hadoop hadoop   620 Mar  4  2024 httpfs-site.xml
-rw-r--r-- 1 hadoop hadoop  3518 Mar  4  2024 kms-acls.xml
-rw-r--r-- 1 hadoop hadoop  1351 Mar  4  2024 kms-env.sh
-rw-r--r-- 1 hadoop hadoop  1860 Mar  4  2024 kms-log4j.properties
-rw-r--r-- 1 hadoop hadoop   682 Mar  4  2024 kms-site.xml
-rw-r--r-- 1 hadoop hadoop 14451 Mar  4  2024 log4j.properties
-rw-r--r-- 1 hadoop hadoop   951 Mar  4  2024 mapred-env.cmd
-rw-r--r-- 1 hadoop hadoop  1764 Mar  4  2024 mapred-env.sh
-rw-r--r-- 1 hadoop hadoop  4113 Mar  4  2024 mapred-queues.xml.template
-rw-r--r-- 1 hadoop hadoop  1122 Jun  6 15:17 mapred-site.xml
drwxr-xr-x 2 hadoop hadoop  4096 Mar  4  2024 shellprofile.d
-rw-r--r-- 1 hadoop hadoop  2316 Mar  4  2024 ssl-client.xml.example
-rw-r--r-- 1 hadoop hadoop  2697 Mar  4  2024 ssl-server.xml.example
-rw-r--r-- 1 hadoop hadoop  2681 Mar  4  2024 user_ec_policies.xml.template
-rw-r--r-- 1 hadoop hadoop    10 Mar  4  2024 workers
-rw-r--r-- 1 hadoop hadoop  2250 Mar  4  2024 yarn-env.cmd
-rw-r--r-- 1 hadoop hadoop  7095 Mar  4  2024 yarn-env.sh
-rw-r--r-- 1 hadoop hadoop  2591 Mar  4  2024 yarnservice-log4j.properties
-rw-r--r-- 1 hadoop hadoop   802 Jun  4 18:35 yarn-site.xml
hadoop@malith-HP-Laptop-15s-fq1xxx:/usr/local/hadoop/etc/hadoop$ 
```

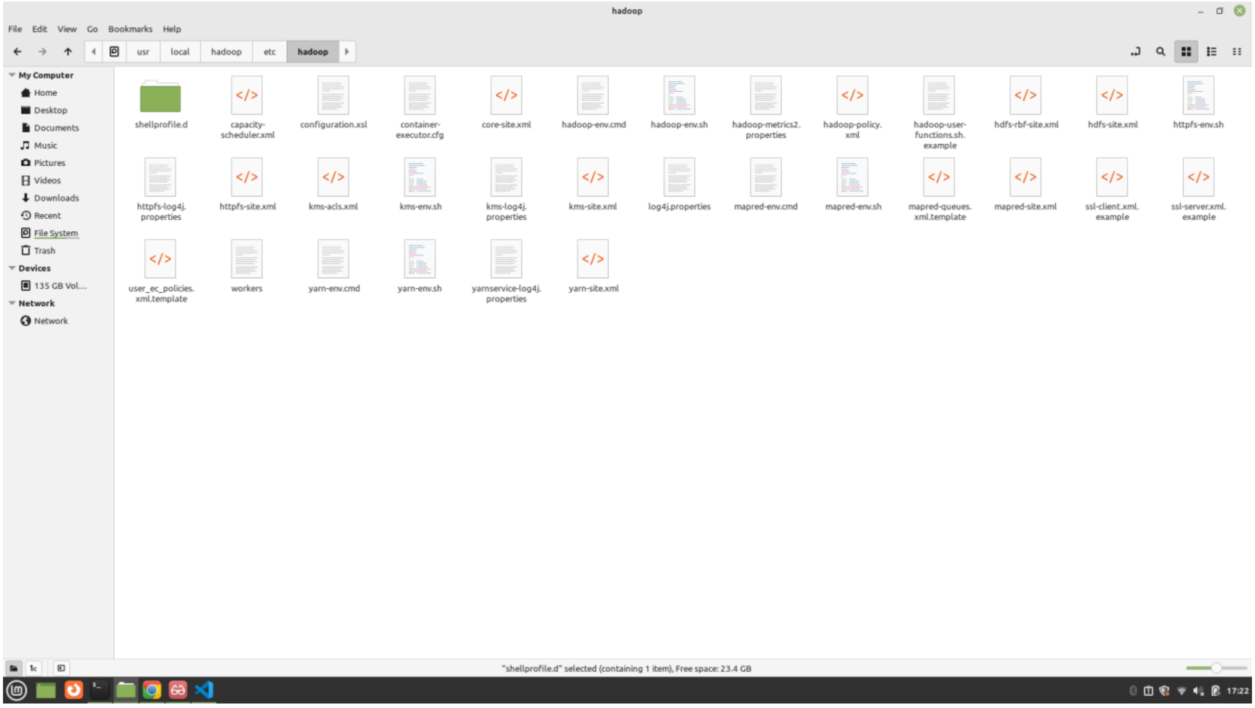
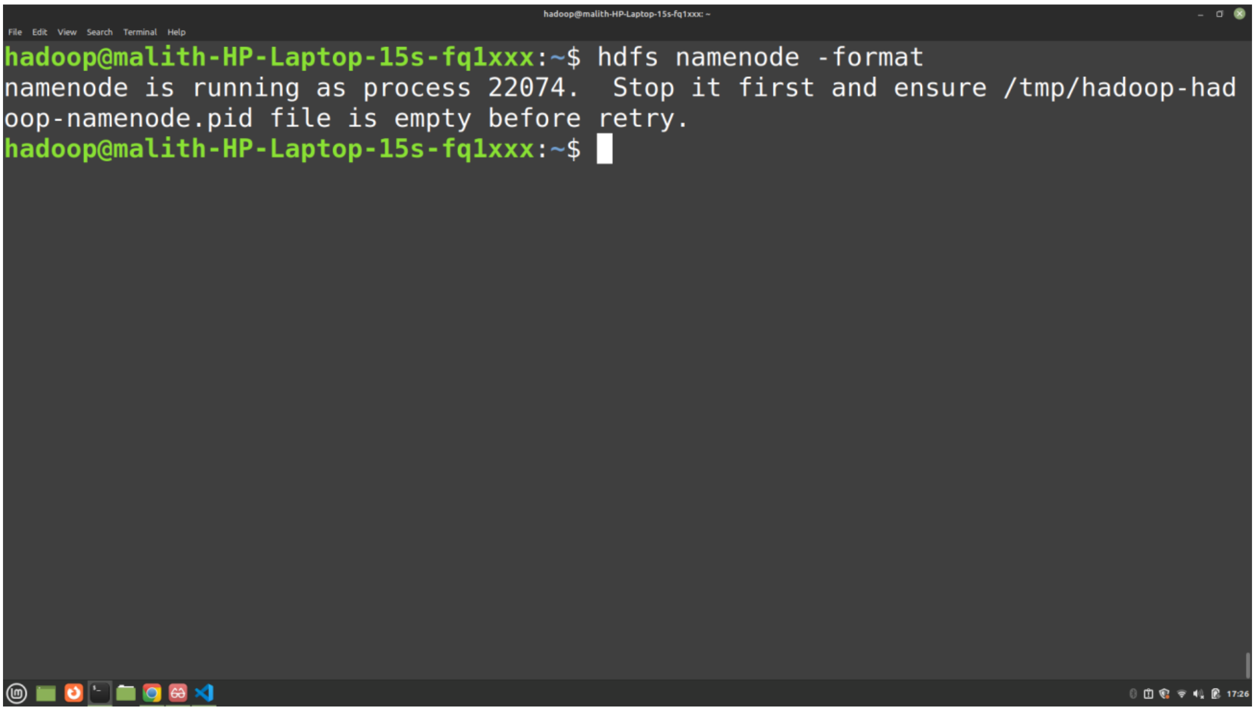


FIGURE 5: EVIDENCE FOR CONFIGURATION FILES

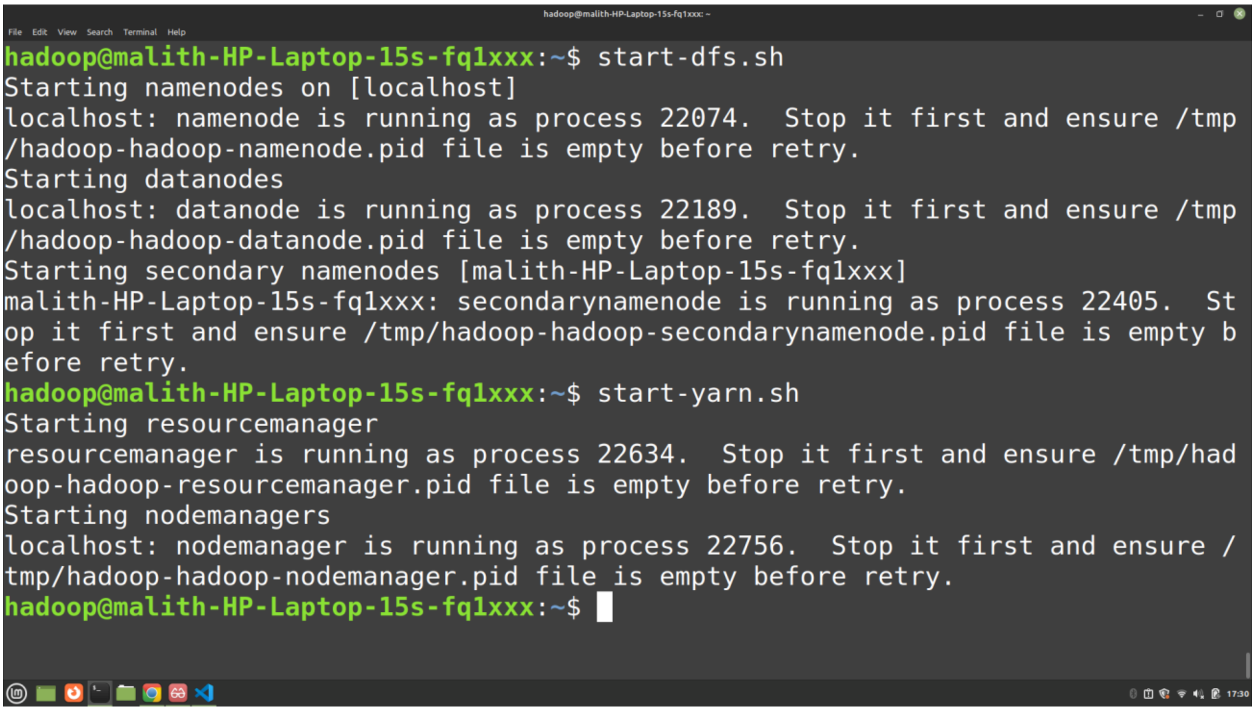
Namenode Formatting:


hadoop@malith-HP-Laptop-15s-fq1xxx: ~
File Edit View Search Terminal Help
17:26

```
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ hdfs namenode -format
namenode is running as process 22074.  Stop it first and ensure /tmp/hadoop-had
oop-namenode.pid file is empty before retry.
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ 
```

FIGURE 6: EVIDENCE FOR NAMENODE FORMATTING

Starting Hadoop Service:

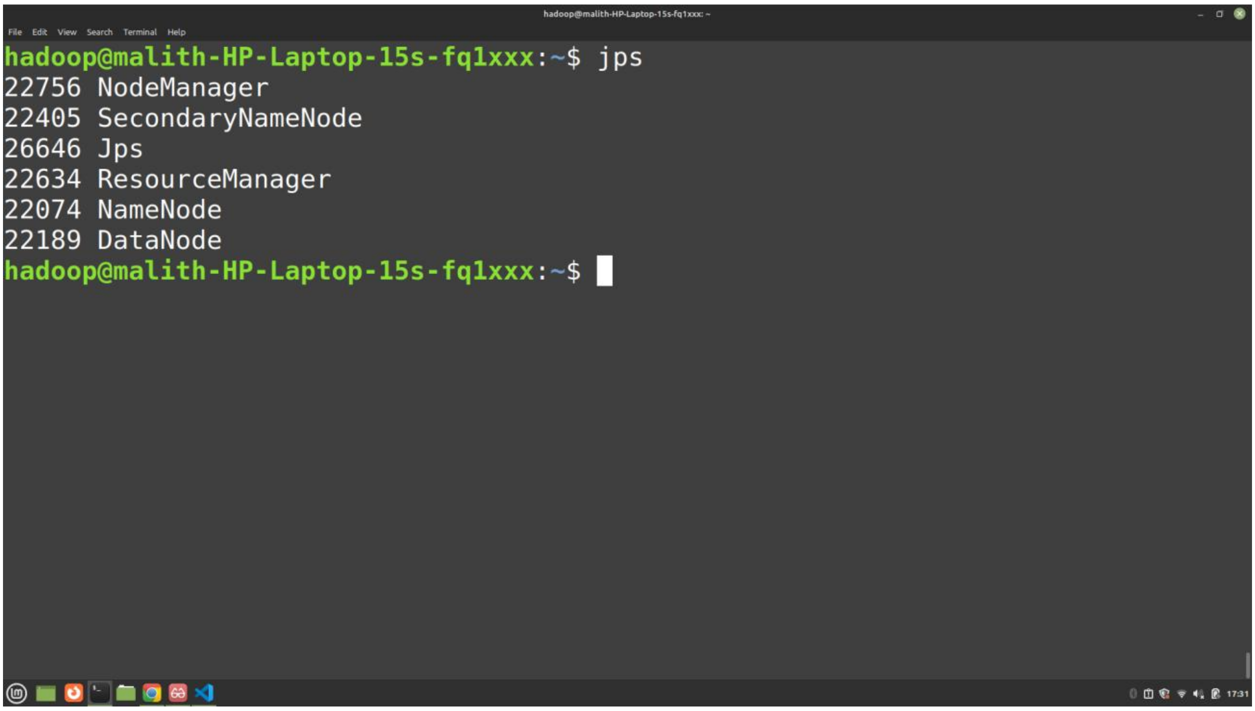


```
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ start-dfs.sh
Starting namenodes on [localhost]
localhost: namenode is running as process 22074.  Stop it first and ensure /tmp
/hadoop-hadoop-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 22189.  Stop it first and ensure /tmp
/hadoop-hadoop-datanode.pid file is empty before retry.
Starting secondary namenodes [malith-HP-Laptop-15s-fq1xxx]
malith-HP-Laptop-15s-fq1xxx: secondarynamenode is running as process 22405.  St
op it first and ensure /tmp/hadoop-hadoop-secondarynamenode.pid file is empty b
efore retry.
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ start-yarn.sh
Starting resourcemanager
resourcemanager is running as process 22634.  Stop it first and ensure /tmp/had
oop-hadoop-resourcemanager.pid file is empty before retry.
Starting nodemanagers
localhost: nodemanager is running as process 22756.  Stop it first and ensure /
tmp/hadoop-hadoop-nodemanager.pid file is empty before retry.
hadoop@malith-HP-Laptop-15s-fq1xxx:~$
```

hadoop@malith-HP-Laptop-15s-fq1xxx: ~
File Edit View Search Terminal Help
17:30

FIGURE 7: EVIDENCE FOR STARTING HADOOP SERVICE

Running Processes:



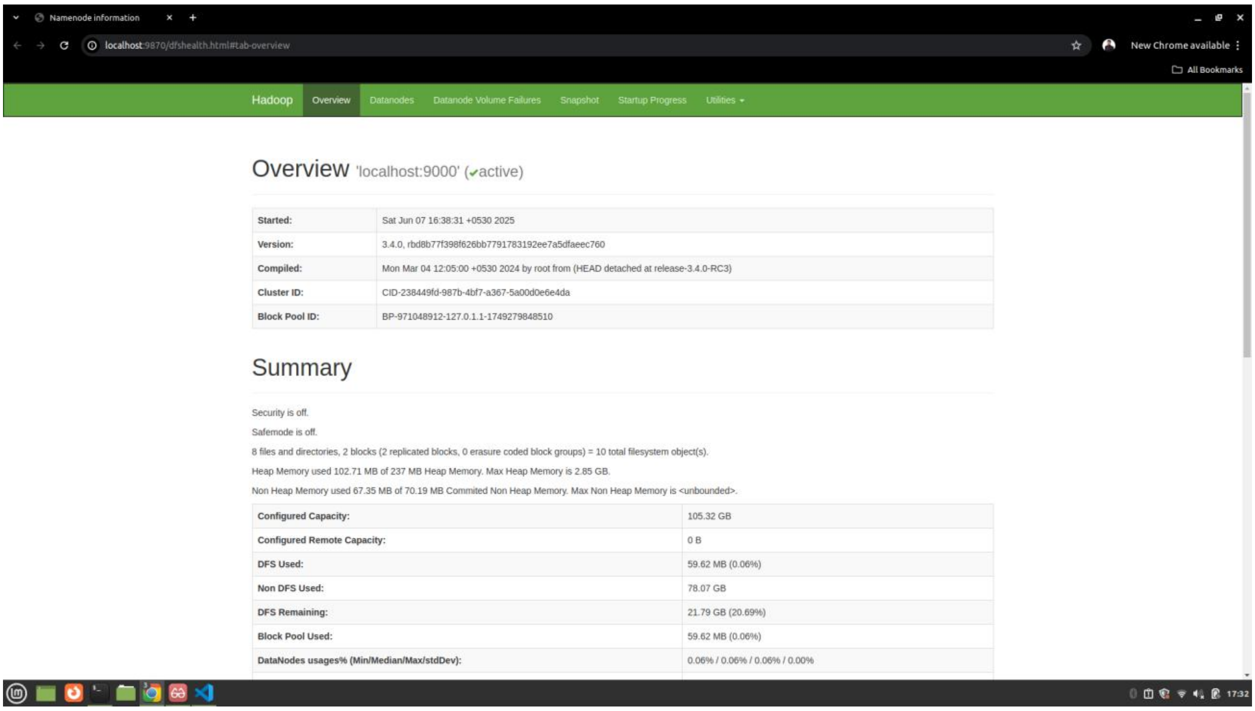
```
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ jps
22756 NodeManager
22405 SecondaryNameNode
26646 Jps
22634 ResourceManager
22074 NameNode
22189 DataNode
hadoop@malith-HP-Laptop-15s-fq1xxx:~$
```

hadoop@malith-HP-Laptop-15s-fq1xxx: ~
File Edit View Search Terminal Help
17:31

FIGURE 8: EVIDENCE FOR RUNNING PROCESSES

Web UI Access:

- NameNode UI: http://localhost:9870



Overview 'localhost:9000' (✓active)

Started:	Sat Jun 07 16:38:31 +0530 2025
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaeec760
Compiled:	Mon Mar 04 12:05:00 +0530 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-238449fd-987b-4bf7-a367-5a00d0e6e4da
Block Pool ID:	BP-971048912-127.0.1.1-1749279848510

Summary

Security is off.

Safemode is off.

8 files and directories, 2 blocks (2 replicated blocks, 0 erasure coded block groups) = 10 total filesystem object(s).

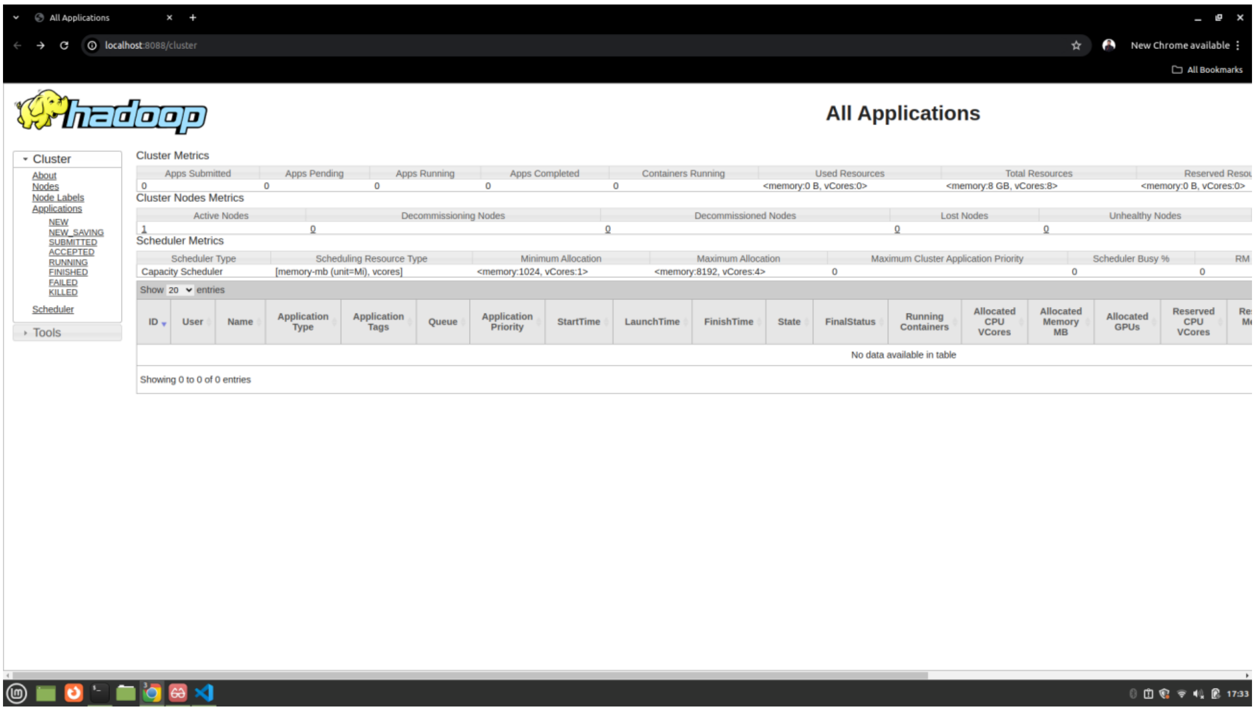
Heap Memory used 102.71 MB of 237 MB Heap Memory. Max Heap Memory is 2.85 GB.

Non Heap Memory used 67.35 MB of 70.19 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	105.32 GB
Configured Remote Capacity:	0 B
DFS Used:	59.62 MB (0.06%)
Non DFS Used:	78.07 GB
DFS Remaining:	21.79 GB (20.69%)
Block Pool Used:	59.62 MB (0.06%)
DataNodes usages% (Min/Median/Max/stdDev):	0.06% / 0.06% / 0.06% / 0.00%

FIGURE 9: EVIDENCE FOR NAMENODE UI: HTTP://LOCALHOST:9870

- ResourceManager UI: http://localhost:8088



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	Total Resources	Reserved Resou
0	0	0	0	0	<memory:0 B, vCores:0>	<memory:8 GB, vCores:8>	<memory:0 B, vCores:0>

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
1	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority	Scheduler Busy %	RM
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0	0	0

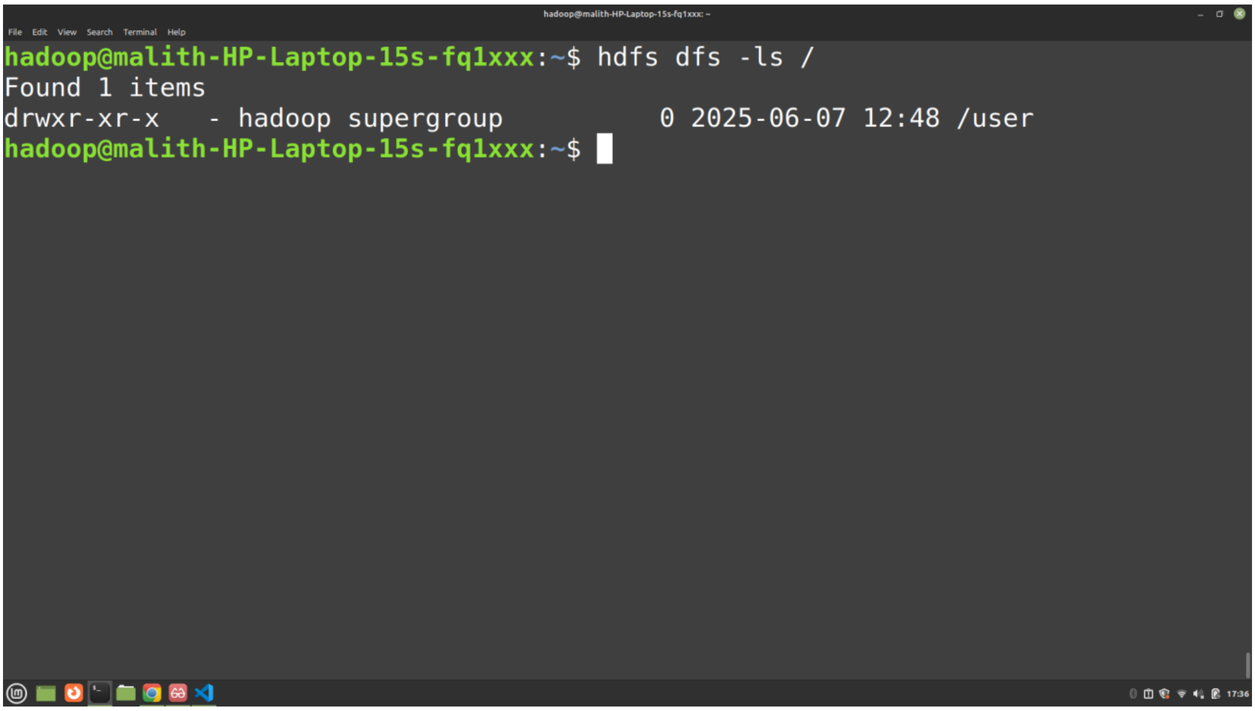
Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Allocated GPUs	Reserved CPU VCores	Re M
No data available in table																	

Showing 0 to 0 of 0 entries

FIGURE 10: EVIDENCE FOR RESOURCEMANAGER UI: HTTP://LOCALHOST:8088

HDFS Commands:


hadoop@malith-HP-Laptop-15s-fq1xxx: ~
File Edit View Search Terminal Help
17:36

```
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ hdfs dfs -ls /
Found 1 items
drwxr-xr-x   - hadoop supergroup          0 2025-06-07 12:48 /user
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ 
```

FIGURE 11: EVIDENCE FOR HDFS COMMANDS

5 Implementation

Map Logic:

- The mapper reads each row of the Spotify dataset, extracts the artist's popularity score and follower count, and emits them as a key-value pair (popularity_score, followers). It skips rows with missing or invalid values.

Reduce Logic:

- The reducer receives all follower counts grouped by popularity score. For each popularity score, it sums the followers and counts the number of artists, then calculates and emits the average followers for that popularity score.

Source Code:

- Both mapper.py and reducer.py are implemented in Python 3 using the standard csv and sys libraries. [https://github.com/MalithPramoditha/assignment_1]

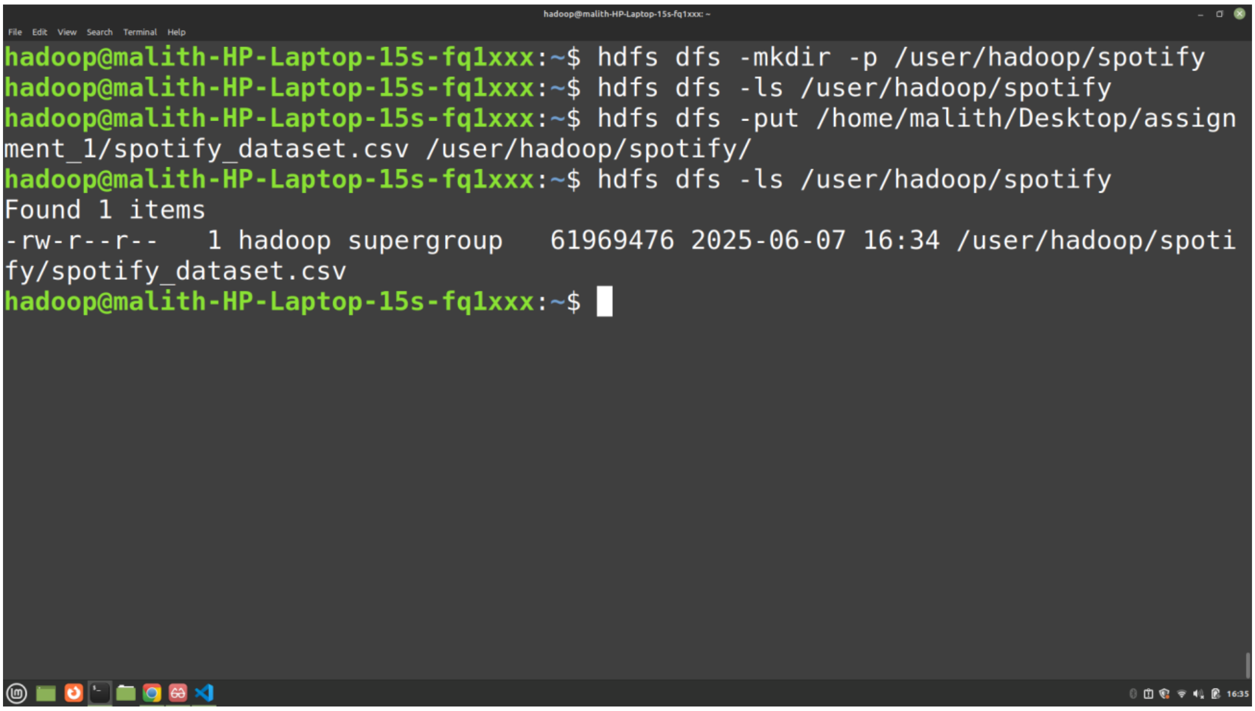
Libraries/Frameworks Used:

- Python 3 standard library (csv, sys)
- Hadoop Streaming for running Python scripts as MapReduce jobs

6 Execution & Testing

Steps Taken to Run the Job:

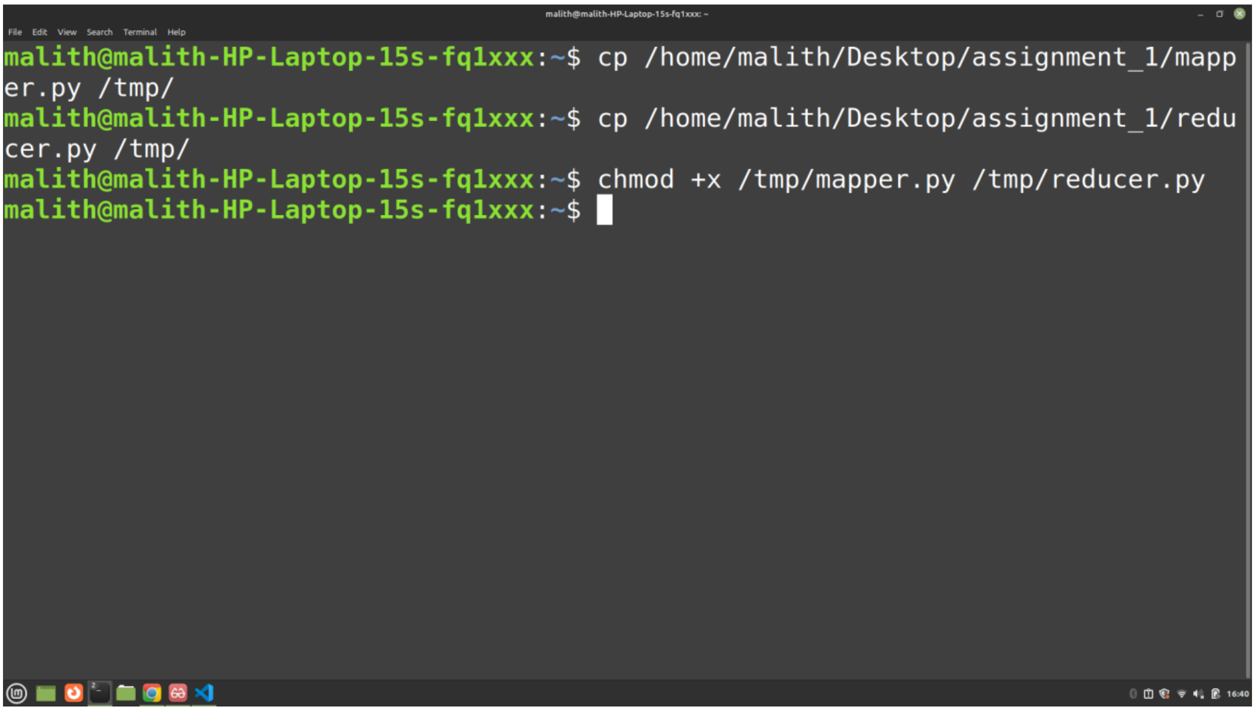
1. Uploaded the Spotify CSV dataset to HDFS:



```
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ hdfs dfs -mkdir -p /user/hadoop/spotify
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ hdfs dfs -ls /user/hadoop/spotify
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ hdfs dfs -put /home/malith/Desktop/assign
ment_1/spotify_dataset.csv /user/hadoop/spotify/
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ hdfs dfs -ls /user/hadoop/spotify
Found 1 items
-rw-r--r--   1 hadoop supergroup   61969476 2025-06-07 16:34 /user/hadoop/spoti
fy/spotify_dataset.csv
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ 
```

FIGURE 12: EVIDENCE FOR UPLOADED THE SPOTIFY CSV DATASET TO HDFS

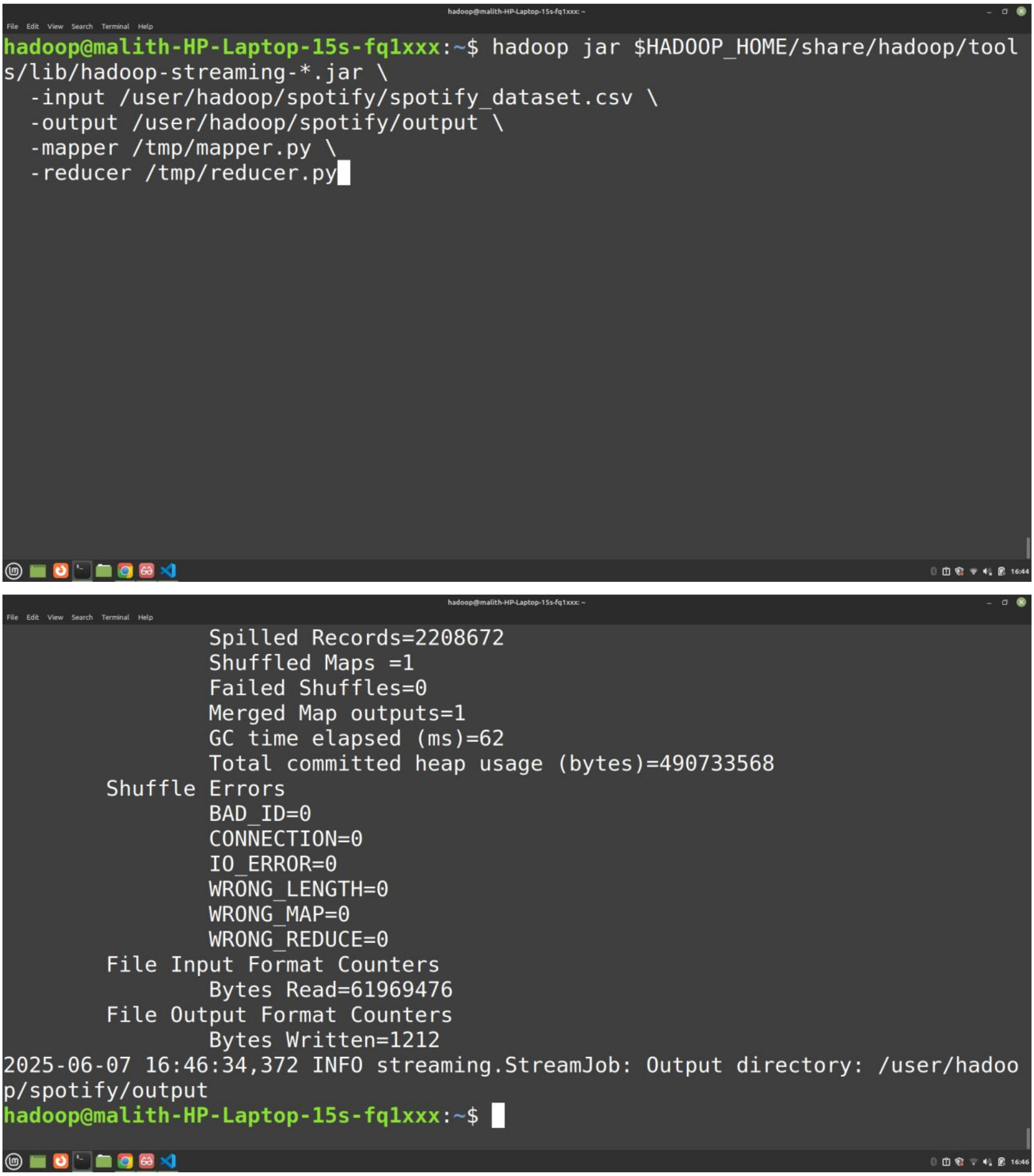
2. Made sure both scripts were executable and had the correct shebang.



```
malith@malith-HP-Laptop-15s-fq1xxx:~$ cp /home/malith/Desktop/assignment_1/mapp
er.py /tmp/
malith@malith-HP-Laptop-15s-fq1xxx:~$ cp /home/malith/Desktop/assignment_1/redu
cer.py /tmp/
malith@malith-HP-Laptop-15s-fq1xxx:~$ chmod +x /tmp/mapper.py /tmp/reducer.py
malith@malith-HP-Laptop-15s-fq1xxx:~$ 
```

FIGURE 13: EVIDENCE FOR MADE SURE BOTH SCRIPTS WERE EXECUTABLE AND HAD THE CORRECT SHEBANG.

3. Ran the Hadoop Streaming job:


hadoop@malith-HP-Laptop-15s-fq1xxx: ~
File Edit View Search Terminal Help

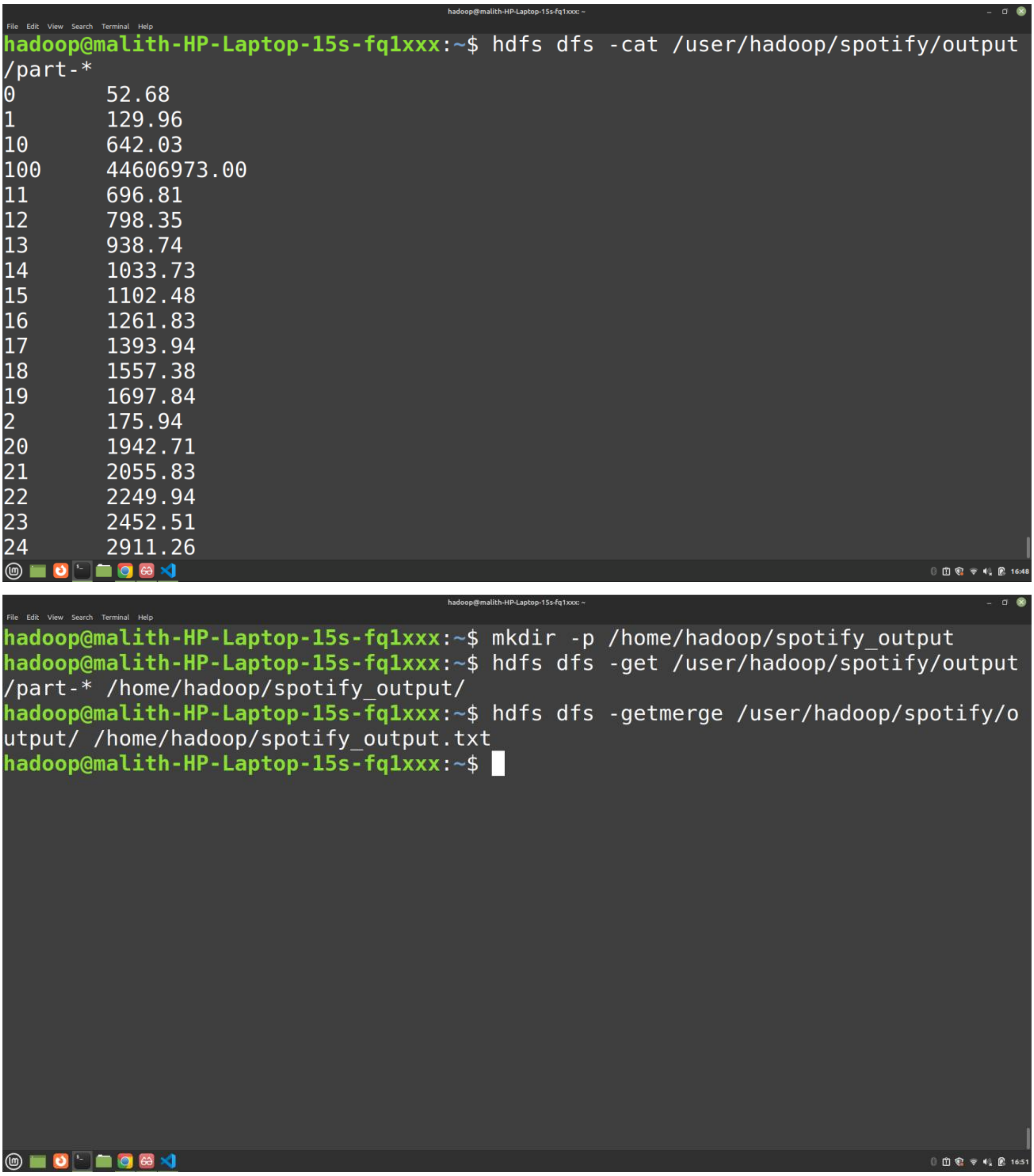
```
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ hadoop jar $HADOOP_HOME/share/hadoop/tool
s/lib/hadoop-streaming-*.jar \
  -input /user/hadoop/spotify/spotify_dataset.csv \
  -output /user/hadoop/spotify/output \
  -mapper /tmp/mapper.py \
  -reducer /tmp/reducer.py
```

16:44
hadoop@malith-HP-Laptop-15s-fq1xxx: ~
File Edit View Search Terminal Help

```
                Spilled Records=2208672
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=62
                Total committed heap usage (bytes)=490733568
        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=61969476
        File Output Format Counters
                Bytes Written=1212
2025-06-07 16:46:34,372 INFO streaming.StreamJob: Output directory: /user/hadoo
p/spotify/output
hadoop@malith-HP-Laptop-15s-fq1xxx:~$
```

FIGURE 14: EVIDENCE FOR RAN THE HADOOP STREAMING JOB

4. Retrieved and viewed the output:



hadoop@malith-HP-Laptop-15s-fq1xxx: ~
File Edit View Search Terminal Help
16:48

```
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ hdfs dfs -cat /user/hadoop/spotify/output
/part-*
0       52.68
1       129.96
10      642.03
100     44606973.00
11      696.81
12      798.35
13      938.74
14      1033.73
15      1102.48
16      1261.83
17      1393.94
18      1557.38
19      1697.84
2       175.94
20      1942.71
21      2055.83
22      2249.94
23      2452.51
24      2911.26
```

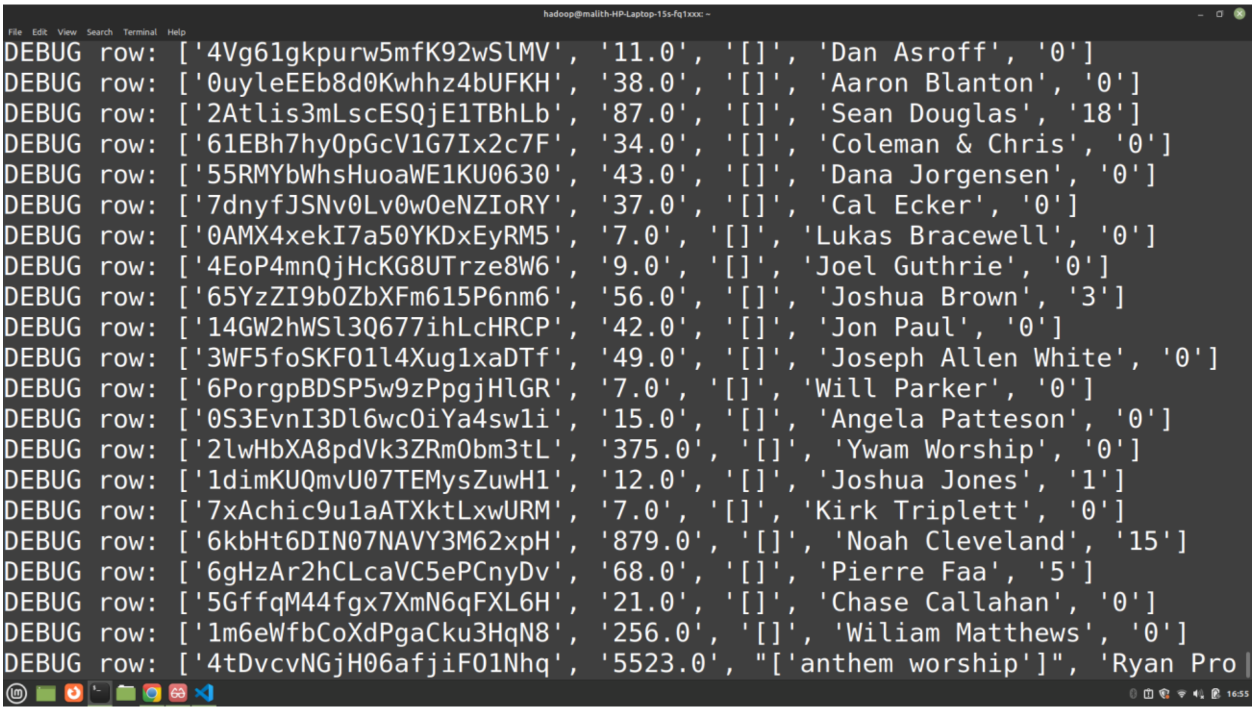
hadoop@malith-HP-Laptop-15s-fq1xxx: ~
File Edit View Search Terminal Help
16:51

```
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ mkdir -p /home/hadoop/spotify_output
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ hdfs dfs -get /user/hadoop/spotify/output
/part-* /home/hadoop/spotify_output/
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ hdfs dfs -getmerge /user/hadoop/spotify/o
utput/ /home/hadoop/spotify_output.txt
hadoop@malith-HP-Laptop-15s-fq1xxx:~$ 
```

FIGURE 15: EVIDENCE FOR RETRIEVED AND VIEWED THE OUTPUT

5. Input/Output Samples:

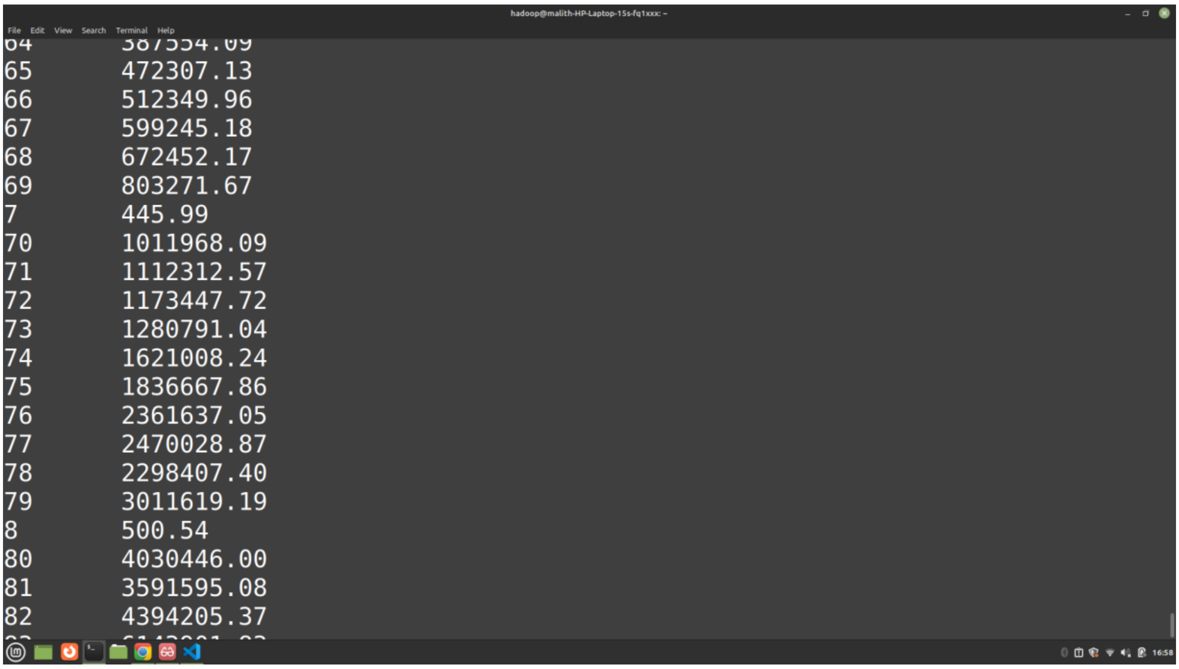
- Input (CSV):



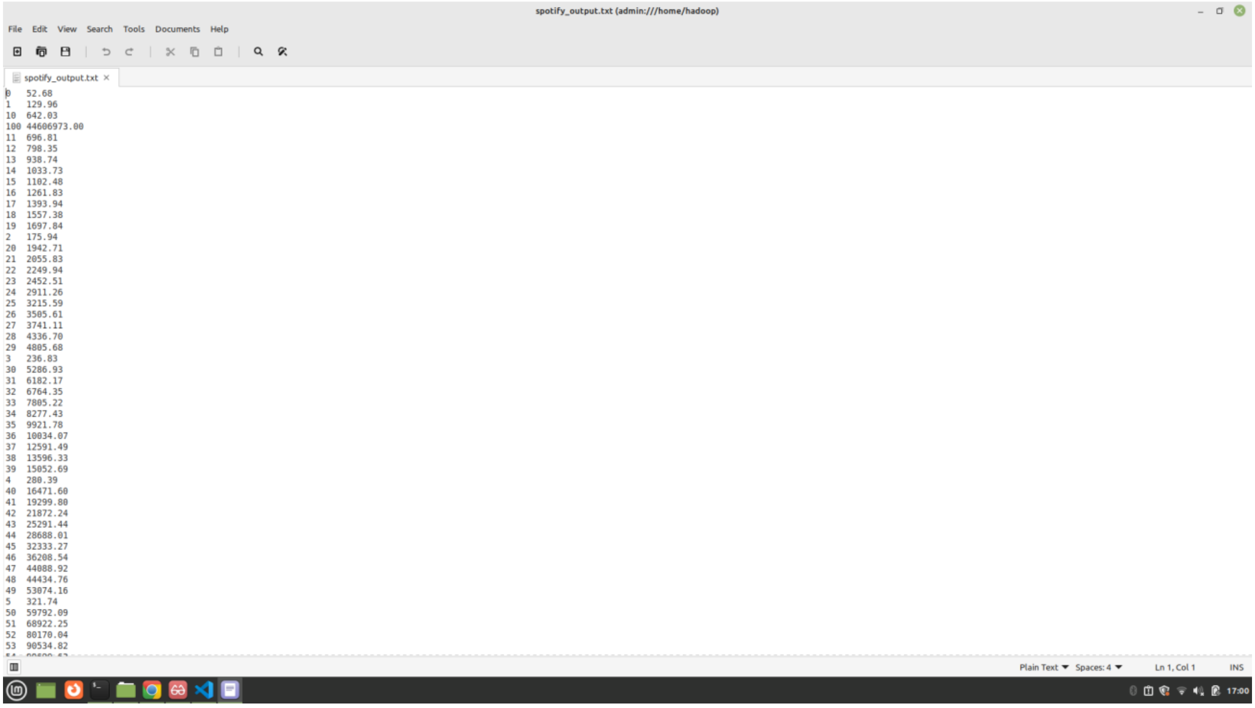
```
DEBUG row: ['4Vg61gkpurw5mfK92wSlMV', '11.0', '[]', 'Dan Asroff', '0']
DEBUG row: ['0uyleEEb8d0Kwhhz4bUFKH', '38.0', '[]', 'Aaron Blanton', '0']
DEBUG row: ['2Atlis3mLscESQjE1TBhLb', '87.0', '[]', 'Sean Douglas', '18']
DEBUG row: ['61EBh7hyOpGcV1G7Ix2c7F', '34.0', '[]', 'Coleman & Chris', '0']
DEBUG row: ['55RMYbWhsHuoaWE1KU0630', '43.0', '[]', 'Dana Jorgensen', '0']
DEBUG row: ['7dnyfJSNv0Lv0wOeNZIoRY', '37.0', '[]', 'Cal Ecker', '0']
DEBUG row: ['0AMX4xekI7a50YKDxEyRM5', '7.0', '[]', 'Lukas Bracewell', '0']
DEBUG row: ['4EoP4mnQjHcKG8UTrze8W6', '9.0', '[]', 'Joel Guthrie', '0']
DEBUG row: ['65YzZI9bOZbXFm615P6nm6', '56.0', '[]', 'Joshua Brown', '3']
DEBUG row: ['14GW2hWSl3Q677ihLcHRCP', '42.0', '[]', 'Jon Paul', '0']
DEBUG row: ['3WF5foSKF01l4Xug1xaDTf', '49.0', '[]', 'Joseph Allen White', '0']
DEBUG row: ['6PorgpBDSP5w9zPpgjHlGR', '7.0', '[]', 'Will Parker', '0']
DEBUG row: ['0S3EvnI3Dl6wcOiYa4sw1i', '15.0', '[]', 'Angela Patteson', '0']
DEBUG row: ['2lwHbXA8pdVk3ZRmObm3tL', '375.0', '[]', 'Ywam Worship', '0']
DEBUG row: ['1dimKUQmvU07TEMysZuwH1', '12.0', '[]', 'Joshua Jones', '1']
DEBUG row: ['7xAchic9u1aATXktLxwURM', '7.0', '[]', 'Kirk Triplett', '0']
DEBUG row: ['6kbHt6DIN07NAVY3M62xpH', '879.0', '[]', 'Noah Cleveland', '15']
DEBUG row: ['6gHzAr2hCLcaVC5ePCnyDv', '68.0', '[]', 'Pierre Faa', '5']
DEBUG row: ['5GffqM44fgx7XmN6qFXL6H', '21.0', '[]', 'Chase Callahan', '0']
DEBUG row: ['1m6eWfbCoXdPgaCku3HqN8', '256.0', '[]', 'Wiliam Matthews', '0']
DEBUG row: ['4tDvcvNGjH06afjiF01Nhq', '5523.0', "['anthem worship']", 'Ryan Pro
```

FIGURE 16: EVIDENCE FOR INPUT

- Output:



```
64      387554.09
65      472307.13
66      512349.96
67      599245.18
68      672452.17
69      803271.67
7       445.99
70      1011968.09
71      1112312.57
72      1173447.72
73      1280791.04
74      1621008.24
75      1836667.86
76      2361637.05
77      2470028.87
78      2298407.40
79      3011619.19
8       500.54
80      4030446.00
81      3591595.08
82      4394205.37
```


spotify_output.txt (admin:///home/hadoop)
File Edit View Search Tools Documents Help
spotify_output.txt
Plain Text Spaces: 4 Ln 1, Col 1 INS
17:00

```
0   52.68
1   129.96
10  642.03
100 44606973.00
11  696.81
12  798.35
13  938.74
14  1033.73
15  1102.48
16  1261.83
17  1393.94
18  1557.38
19  1697.84
2   175.94
20  1942.71
21  2055.83
22  2249.94
23  2452.51
24  2911.26
25  3215.59
26  3505.61
27  3741.11
28  4336.70
29  4805.68
3   236.83
30  5286.93
31  6182.17
32  6764.35
33  7805.22
34  8277.43
35  9921.78
36  10034.07
37  12591.49
38  13596.33
39  15052.69
4   280.39
40  16471.60
41  19299.80
42  21872.24
43  25291.44
44  28688.01
45  32333.27
46  36208.54
47  44088.92
48  44434.76
49  53074.16
5   321.74
50  59792.09
51  68922.25
52  80170.04
53  90534.82
```

FIGURE 17: EVIDENCE FOR OUTPUT

7 Results & Interpretation

Key Findings:

- The MapReduce job successfully computed the average number of followers for each popularity score among Spotify artists. The results show that higher popularity scores generally correspond to higher average follower counts, with some variation at the lower and upper ends of the popularity spectrum.

Patterns and Insights:

- Most artists with low popularity scores have relatively few followers, but there are exceptions.
- As popularity increases, the average follower count also increases, indicating a positive correlation between these two metrics.
- Some popularity scores have outliers with extremely high follower counts, skewing their averages.

Performance:

- The job processed a large dataset efficiently using Hadoop Streaming and Python scripts. No critical bottlenecks were observed, and the output was produced in a reasonable timeframe.

Improvements/Future Work:

- Extend the analysis to groups by genres or countries if such data is available.
- Visualize the popularity-follower relationship with graphs.
- Handle missing or anomalous data more robustly.

8 Conclusion

This assignment provided hands-on experience with Hadoop MapReduce and large-scale data analysis using Python and Hadoop Streaming. We learned how to preprocess data, write custom mapper and reducer scripts, and interpret distributed computation results. The process reinforced the importance of data cleaning, correct script permissions, and debugging skills when working with big data tools. Overall, the project deepened my understanding of distributed computing and its application to real-world datasets like Spotify's artist metadata.

9 References

[1] L. Neziri, "Spotify Datasets," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/lehaknarnauli/spotify-datasets. [Accessed 7 June 2025].