

Analysis of Stock Market using Hadoop Map Reduce

Vishal Yadav

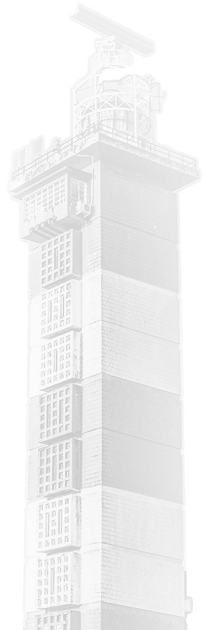
Department of Mechanical Engineering
Indian Institute of Technology Kanpur

May 30, 2022



Contents

- AIM
- Objective
- Plan Of Work
- Algorithm
- Results
- Contribution

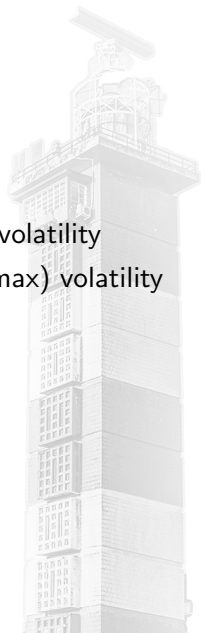


- Finding Volatility Of Stocks



Objective

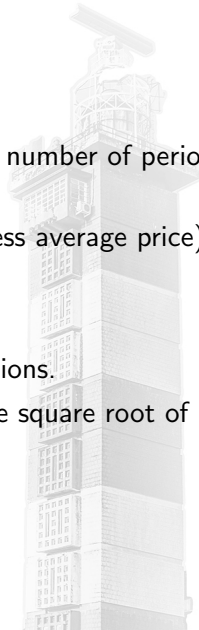
- Find the top 10 stocks with Lowest (min) volatility
- Find the top 10 stocks with the Highest (max) volatility



Plan Of Work

Calculation steps of Volatility is as follows:

- Calculate the average (mean) price for the number of periods or observations.
- Determine each period's deviation (close less average price).
- Square each period's deviation.
- Sum the squared deviations.
- Divide this sum by the number of observations.
- The standard deviation is then equal to the square root of that number.

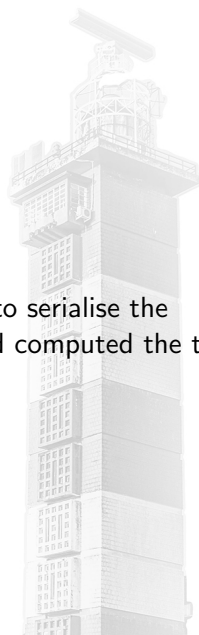


Plan Of Work(Contd..)

- The equation for standard deviation is $s =$

$$\sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}.$$

- We used mapreduce paradigm of Hadoop to serialise the calculation of volatility for each month and computed the top 10 and bottom values.
- Number of Mapper Implementation : 3
- Number of Reducer Implementation : 3



Algorithm

Roles Of Each Mapper and Reducer: Mapper1:

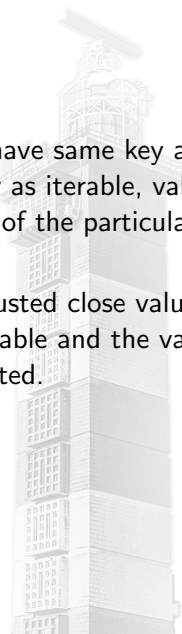
- splits the input data and options the date and close adjusted value.
- key - stock name + month + year
- value - date + adjusted close value



Algorithm(Contd..)

Reducer1:

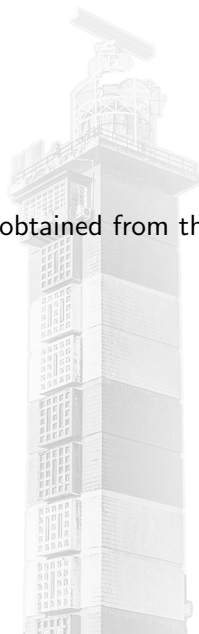
- Since after the map step the values which have same key are grouped together and passed to the reducer as iterable, values that correspond to specific month and year of the particular stock are grouped together.
- Beginning adjusted close value and end adjusted close value are obtained by integrating through the iterable and the value of x_i for the corresponding month is computed.
- key - Company Name
- Value - Computed X_i .



Algorithm(Contd..)

Mapper2:

- Now we have to consolidate all the values obtained from the reducer with respect to company name.
- Key - Company Name
- Value - X_i



Algorithm (Contd..)

Reducer2:

- All the x_i corresponding to the the respective companies are grouped together.
- Volatility for the particular company is obtained from these values.
- Key - Company Name
- Value - Volatility



Algorithm (Contd..)

Mapper3:

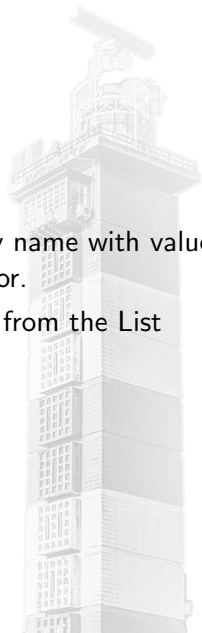
- All the companies are grouped together with a common key.
- Key - Common
- Value = Company Name + Volatility



Algorithm (Contd..)

Reducer3:

- Obtained iterable contains all the company name with values and they are sorted by a custom comparator.
- top 10 and bottom 10 values are obtained from the List



Results

```
hadoop@milind-VirtualBox: ~/Analysis-of-Stock-Market-using-Hadoop-Map-Reduce
hadoop@milind-VirtualBox:~/Analysis-of-Stock-Market-using-Hadoop-Map-Reduce$ hdfs dfs -put /home/hadoop/Analysis-of-Stock-Market-using-Hadoop-Map-Reduce/input/* stock_input
2022-05-24 06:59:38,486 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@milind-VirtualBox:~/Analysis-of-Stock-Market-using-Hadoop-Map-Reduce$ hadoop jar stock.jar MainStockAnalysis /stock_input /stock_output
2022-05-24 07:01:15,692 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
*****Stock_Analysis_Hadoop_MapReduce-> Start*****
2022-05-24 07:01:17,449 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2022-05-24 07:01:18,494 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the tool interface and execute your application with ToolRunner to remedy this.
2022-05-24 07:01:18,583 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1653355443556_0001
2022-05-24 07:01:19,232 INFO mapreduce.JobSubmitter: Cleaning up the staging area /tmp/hadoop-yarn/staging/hadoop/.staging/job_1653355443556_0001
Exception in thread "main" org.apache.hadoop.mapreduce.lib.input.InvalidInputException: Input path does not exist: hdfs://127.0.0.1:9000/stock_input
    at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.singleThreadedListStatus(FileInputFormat.java:332)
    at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.listStatus(FileInputFormat.java:274)
    at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.get_splits(FileInputFormat.java:396)
    at org.apache.hadoop.mapreduce.JobSubmitter.writeNew_splits(JobSubmitter.java:310)
    at org.apache.hadoop.mapreduce.JobSubmitter.write_splits(JobSubmitter.java:327)
    at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitter.java:200)
    at org.apache.hadoop.mapreduce.Job$11.run(Job.java:1565)
    at org.apache.hadoop.mapreduce.Job$11.run(Job.java:1562)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1762)
    at org.apache.hadoop.mapreduce.Job.submit(Job.java:1562)
```

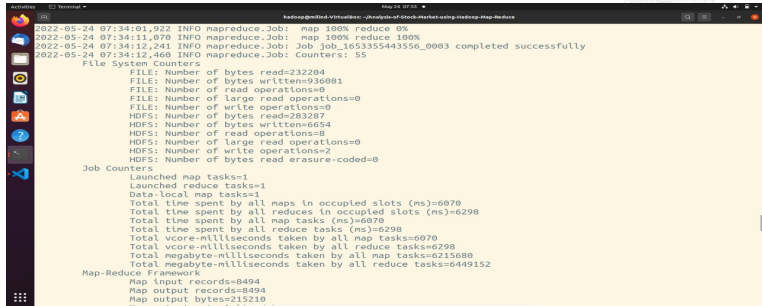
Figure: commands to put our input on hadoop distributed file system and run the program

Results(Contd..)

```
2022-05-24 07:34:01,922 INFO mapreduce.Job: map 100% reduce 0%
2022-05-24 07:34:11,070 INFO mapreduce.Job: map 100% reduce 100%
2022-05-24 07:34:12,241 INFO mapreduce.Job: Job job_1653355443556_0003 completed successfully
2022-05-24 07:34:12,460 INFO mapreduce.Job: Counters: 55
  File System Counters
    FILE: Number of bytes read=232204
    FILE: Number of bytes written=936081
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=283287
    HDFS: Number of bytes written=6654
    HDFS: Number of read operations=8
    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=1
    Launched reduce tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=6070
    Total time spent by all reduces in occupied slots (ms)=6298
    Total time spent by all map tasks (ms)=6070
    Total time spent by all reduce tasks (ms)=6298
    Total vcore-milliseconds taken by all map tasks=6070
    Total vcore-milliseconds taken by all reduce tasks=6298
    Total megabyte-milliseconds taken by all map tasks=6215680
    Total megabyte-milliseconds taken by all reduce tasks=6449152
  Map-Reduce Framework
    Map input records=8494
    Map output records=8494
    Map output bytes=215210
```

Figure: Map Reduce jobs running in parallel manner facilitated by the distributed file system

Results(Contd..)



```
2022-05-24 07:34:01,922 INFO mapreduce.Job: map 100% reduce 0%
2022-05-24 07:34:11,070 INFO mapreduce.Job: map 100% reduce 100%
2022-05-24 07:34:12,241 INFO mapreduce.Job: Job job_1653355443556_0003 completed successfully
2022-05-24 07:34:12,460 INFO mapreduce.Job: Counters: 55
  File System Counters
    FILE: Number of bytes read=232204
    FILE: Number of bytes written=936081
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=283287
    HDFS: Number of bytes written=6654
    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=1
    Launched reduce tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=6070
    Total time spent by all reduces in occupied slots (ms)=6298
    Total time spent by all map tasks (ms)=6070
    Total time spent by all reduce tasks (ms)=6298
    Total vcore-milliseconds taken by all map tasks=6070
    Total vcore-milliseconds taken by all reduce tasks=6298
    Total megabyte-milliseconds taken by all map tasks=6215680
    Total megabyte-milliseconds taken by all reduce tasks=6449152
  Map-Reduce Framework
    Map input records=8494
    Map output records=8494
    Map output bytes=215210
```

Figure: Completion Of Tasks

Results(Contd..)

```
*****Stock_Analysis_Hadoop_MapReduce-> End*****

hadoop@milind-VirtualBox:~/Analysis-of-Stock-Market-using-Hadoop-Map-Reduce$ hdfs dfs -ls
2022-05-24 07:45:57,666 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using built-in java classes where applicable
Found 5 items
drwxr-xr-x - hadoop supergroup          0 2022-05-24 07:33 Intermediate1
drwxr-xr-x - hadoop supergroup          0 2022-05-24 07:34 Intermediate2
drwxr-xr-x - hadoop supergroup          0 2022-05-24 05:45 stock_analysis
drwxr-xr-x - hadoop supergroup          0 2022-05-24 07:00 stock_input
hadoop@milind-VirtualBox:~/Analysis-of-Stock-Market-using-Hadoop-Map-Reduce$ hdfs dfs -get stock_output copyFromHadoop2
2022-05-24 07:47:07,309 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using built-in java classes where applicable
hadoop@milind-VirtualBox:~/Analysis-of-Stock-Market-using-Hadoop-Map-Reduce$
hadoop@milind-VirtualBox:~/Analysis-of-Stock-Market-using-Hadoop-Map-Reduce$
```

Figure: commands to get the output back from the distributed file system onto our own file system

Results(Contd..)


```
copyFromHadoop2 >  part-r-00000
1 Top 10 stocks with Minimum volatility 0.0
2 AGZD 0.003938593878697365
3 AXPWW 0.0044388372955839524
4 AUMAU 0.006017144863314729
5 AGND 0.010751963436794309
6 AGNCP 0.01669670030720715
7 AGNCB 0.016781408595782567
8 ALLB 0.021866756279518028
9 AGIIL 0.022955571847192706
10 ASRVP 0.028529779716934052
11 ACNB 0.028565410375761102
12 Top 10 stocks with Maximum volatility 1.0
13 APDN 0.3773818041663614
14 ALDR 0.39064070974779724
15 ANY 0.4118627840513947
16 AFMD 0.41919685205354573
17 AMCF 0.4279202890844407
18 ATRA 0.42898449574226183
19 ASPX 0.43506357182854893
20 ADXS 0.4411702287863926
21 APDNW 0.6975880360551902
22 ACST 9.271589761859984
23
```

Figure: Output