The Godfather of Talent | Futurense

Stock Analysis Data Engineering Solution

Prepared By

Kousik Dutta

Data Specialist

PROJECT OVERVIEW

Objective :

Build an End-to-End Data Engineering Pipeline to analyze the Stock Data using Big Data technologies.

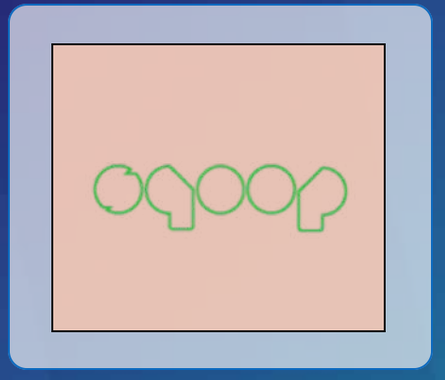
Data Description :

The dataset includes information such as the opening price, closing price, highest price, lowest price, and trading volume of these stocks. Stocks we have analyzed on are AAL, AAOI, ABMD, ABIO, Walmart

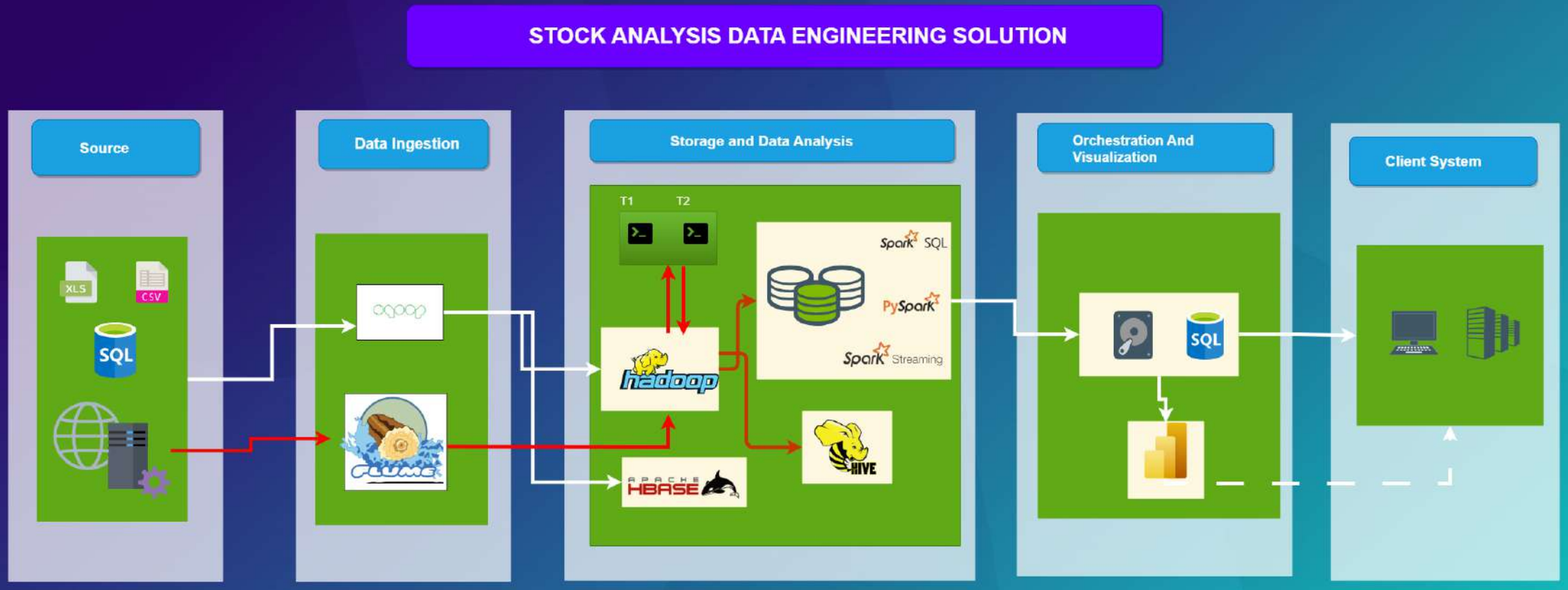
Target Use Case :

Valuable use case for this data is to generate an investor-ready dashboard. This dashboard can provide various investors with the tools they need to carry out targeted analysis before adding these stocks to their portfolios.

TOOLS & FRAMEWORKS USED



STOCK ANALYSIS DATA ENGINEERING SOLUTION

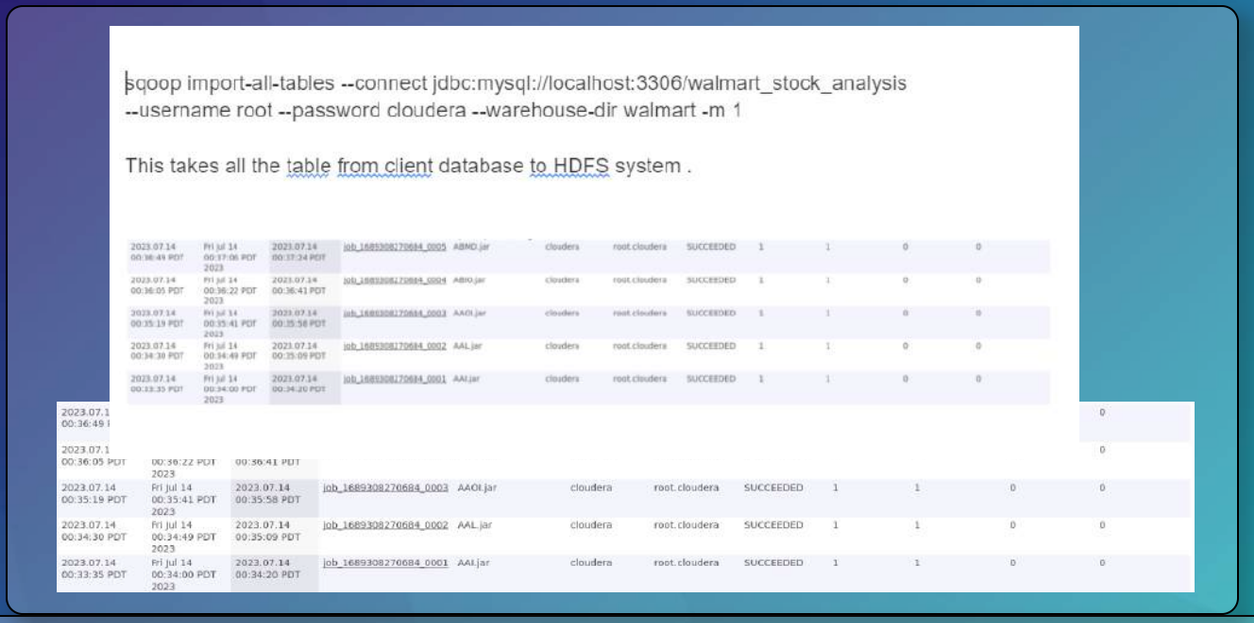


PIPELINE MADE

Importing All Tables

```
sqoop import-all-tables --connect jdbc:mysql://localhost:3306/walmart_stock_analysis
--username root --password cloudera --warehouse-dir walmart -m 1
```

This takes all the table from client database to HDFS system .



Exporting Result

RESPONSIBILITIES

1. **Loading Data with Utmost Security from Static Data Source:**
   Implement secure data loading procedures from static data sources.

2. **Data Ingestion Pipeline Creation:**
   Configure and optimize data ingestion workflows.
   Implement fault-tolerant mechanisms to handle data ingestion failures.

3. **Handling and Managing Data using Hadoop Distributed File System (HDFS) and YARN:**
   Optimize data storage and retrieval using compression, Bucketing and indexing techniques.

4. **Streamlining Data Flow between Hadoop Ecosystem Components (HDFS and Hive)**

5. **Memory-Based Analysis for Batch Processing using PySpark:**
   Leverage in-memory computing for high-performance data processing.

6. **Interactive Memory Processing using Spark SQL**
7. **Using Spark Streaming to analyze the stocks data**
8. **Power BI Report Generation based on Analytical Results:**
   Design interactive dashboards and visualizations to convey analytical insights.

ENVIRONMENT & VERSIONS

Versions

Hadoop	2.6.0
Sqoop	1.4.6
HBASE	1.2.0
SPARK	3.3.2
HIVE	1.1.0
MYSQL	14.14
Python	3.10.1

Environment

Cloudera 5.8.0
Jupyter Notebook

Stock Name

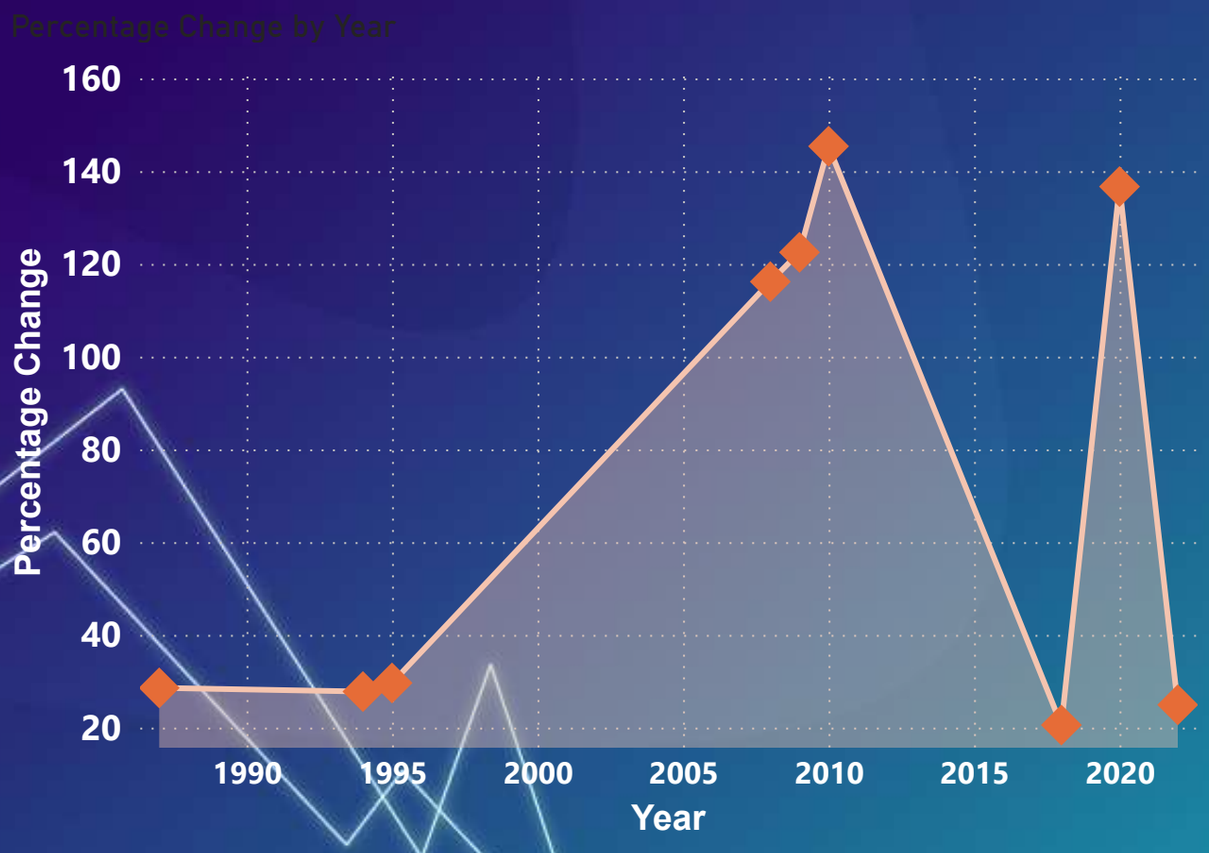
All

Year

All

Quarter

All



Year	Quarter	Month	Day	Percentage Change
2022	Qtr 3	September	16	24.92
2020	Qtr 1	March	19	18.87
2020	Qtr 2	May	28	117.80
2018	Qtr 4	November	8	20.56
2010	Qtr 1	March	26	145.37
2009	Qtr 1	January	28	122.50
2008	Qtr 3	July	22	45.24
2008	Qtr 4	October	10	40.67
2008	Qtr 4	November	12	30.27
1995	Qtr 1	February	22	29.63
1994	Qtr 4	December	30	27.78
1987	Qtr 4	December	31	28.57

Percentage Change VS Year Analysis

Observations :

1. Volatility in Stock Prices - The high percentage difference between the open and close prices of stocks indicates a significant level of volatility in the market.

2. Intraday Trading Opportunities - The substantial percentage difference between the open and close prices presents potential opportunities for intraday traders. This difference suggests that there is significant price movement within a single trading day

Stock Name

All

Year

All

Quarter

All

Year	Quarter	Month	Day	Average Month Low	Sum of low
1988	Qtr 1	January	4	4.10	3.31
1988	Qtr 1	January	5	4.10	3.88
1988	Qtr 1	January	6	4.10	4.06
1988	Qtr 1	January	12	4.10	3.94
1988	Qtr 1	January	13	4.10	3.94
1988	Qtr 1	January	20	4.10	4.06
1988	Qtr 1	January	21	4.10	3.94
1988	Qtr 1	January	22	4.10	4.00
1988	Qtr 1	January	25	4.10	4.06
1988	Qtr 1	January	27	4.10	4.06
1988	Qtr 1	January	28	4.10	4.06
1988	Qtr 1	January	29	4.10	4.00
1988	Qtr 1	February	11	4.11	3.94
1988	Qtr 1	February	12	4.11	4.06
1988	Qtr 1	February	18	4.11	4.00
1988	Qtr 1	February	19	4.11	4.06
1988	Qtr 1	February	22	4.11	4.00
1988	Qtr 1	February	23	4.11	4.06
1988	Qtr 1	February	24	4.11	4.00

Comparison Between Low Price & Average Month Low Price

Observations :

1. If the low price consistently stays above the average monthly low, it may indicate a strong support level, suggesting that the stock has a higher probability of bouncing back from that price range , if the low price repeatedly falls below the average monthly low, it could suggest a potential breakdown of support. Traders can use these insights to make more informed decisions regarding their trading strategies.

2. Market Sentiment and Investor Behavior : Comparing the low price to the average monthly low can provide insights into market sentiment (Positive or Negative) and investor behavior.

Stock Name

All

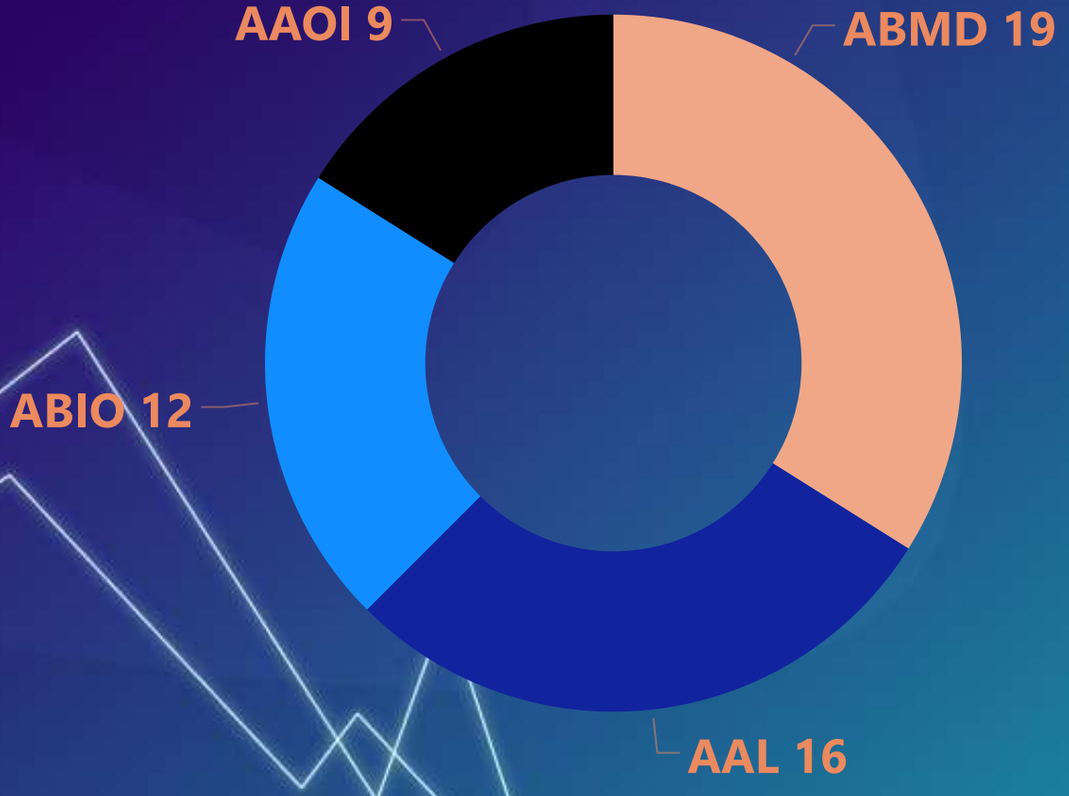
Year

All

Quarter

All

Streak Length According to Stock



Year	Quarter	Month	Day	Close Price	Serial Number
1988	Qtr 2	June	13	4.06	1
1988	Qtr 2	June	14	4.06	2
1988	Qtr 2	June	15	4.06	3
1988	Qtr 2	June	16	4.19	4
1988	Qtr 2	June	17	4.19	5
1988	Qtr 2	June	20	4.25	9
1988	Qtr 2	June	21	4.25	8
1988	Qtr 2	June	22	4.25	7
1988	Qtr 2	June	23	4.25	6
1988	Qtr 2	June	24	4.25	11
1988	Qtr 2	June	27	4.25	10
1988	Qtr 2	June	28	4.31	13
1988	Qtr 2	June	29	4.31	12
1988	Qtr 2	June	30	4.31	15
1988	Qtr 3	July	1	4.31	14
1988	Qtr 3	July	5	4.69	16
1988	Qtr 3	July	6	4.69	17
1988	Qtr 3	July	7	4.94	18
1988	Qtr 3	July	8	5.00	19
2005	Qtr 4	October	25	22.00	1

Longest Streak Analysis

Observations :

1. Momentum Analysis: Streaks can indicate the momentum of a stock. A prolonged positive streak may indicate strong buying pressure and upward momentum, while a prolonged negative streak may indicate selling pressure and downward momentum. Momentum analysis can help identify stocks that are likely to continue their current trend or experience a reversal.

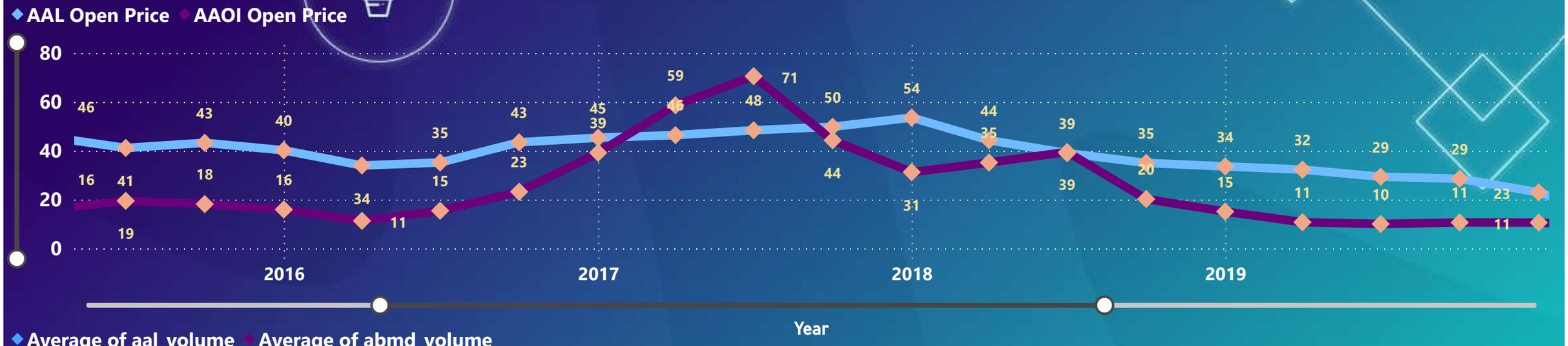
Quarter

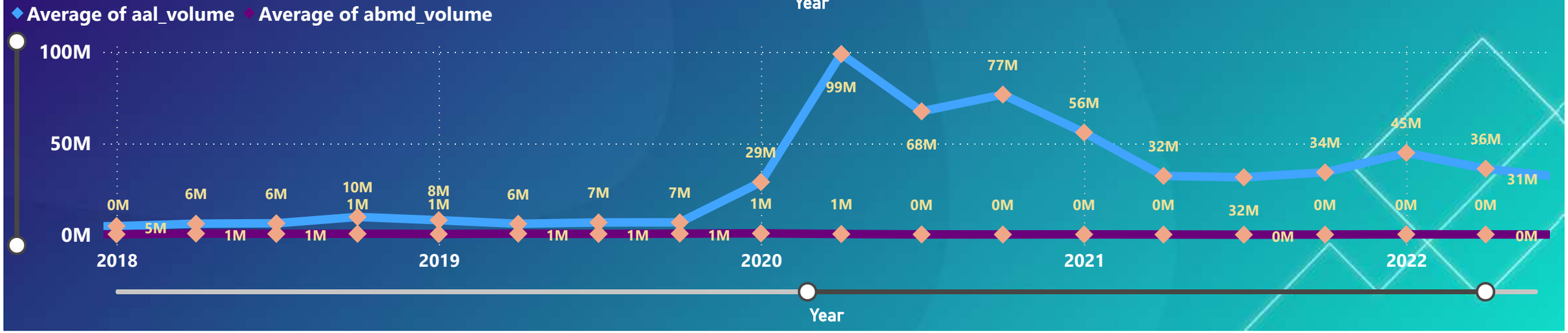
All

Year

All

Year	Quarter	Month	Day	AAL Open Price	AAOI Open Price	AAL Volume	abmd_volume
2022	Qtr 1	January	3	18.23	5.20	42781000	240000
2022	Qtr 1	January	4	19.22	5.30	29266600	316200
2022	Qtr 1	January	5	19.13	5.26	34447900	359200
2022	Qtr 1	January	6	18.89	5.00	19097700	245600





Comparison Between Different Stocks

Observations :

1. Relative Strength: If the open price of AAL is consistently higher than AAOI on certain dates, it suggests that investors have more confidence in the performance and potential of American Airlines compared to AAOI.

1. 2. volume Comparison: The query also considers the volume of AAL compared to the volume of ABMD. Volume represents the number of shares traded, and higher volume generally indicates increased market activity and interest in a particular stock.

Stock Name

All

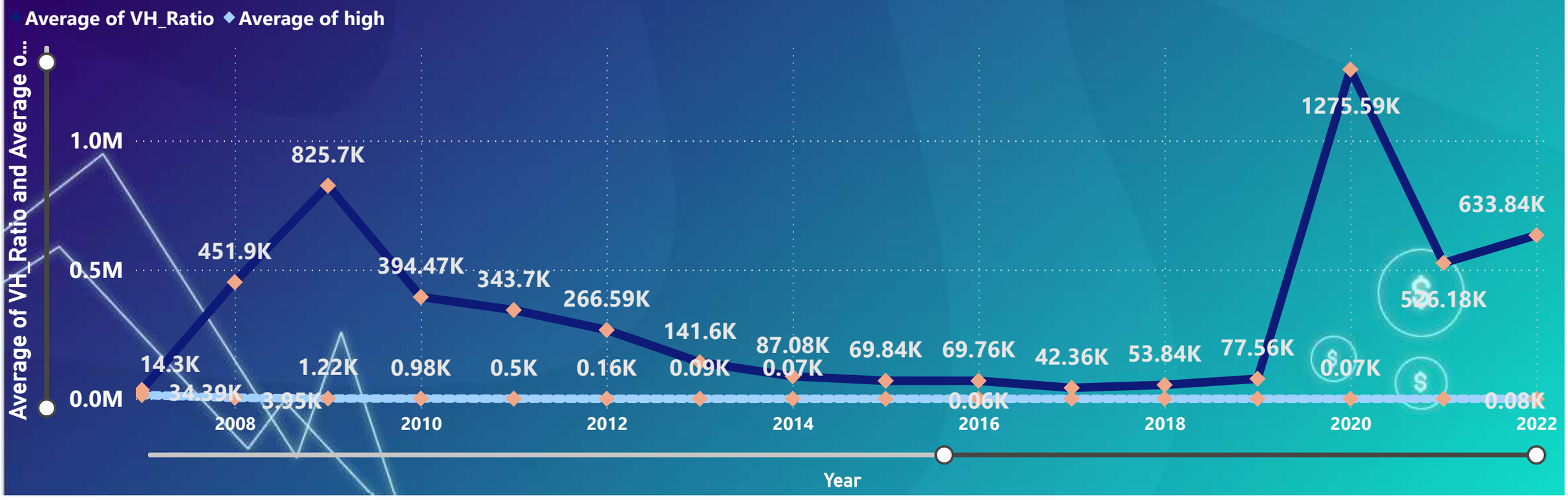
Year

All

Quarter

All

Year	Quarter	Month	Day	Volume	High	VH_Ratio
1987	Qtr 3	July	29	201200	5.50	36,581.82
1987	Qtr 3	July	30	107000	5.56	19,235.96
1987	Qtr 3	July	31	35400	5.56	6,364.04
1987	Qtr 3	August	3	23800	5.56	4,278.65
1987	Qtr 3	August	4	35800	5.56	6,435.96
1987	Qtr 3	August	5	23800	5.56	4,278.65
1987	Qtr 3	August	6	34000	5.56	6,112.36
1987	Qtr 3	August	7	11400	5.56	2,049.44
1987	Qtr 3	August	10	0	5.56	0.00

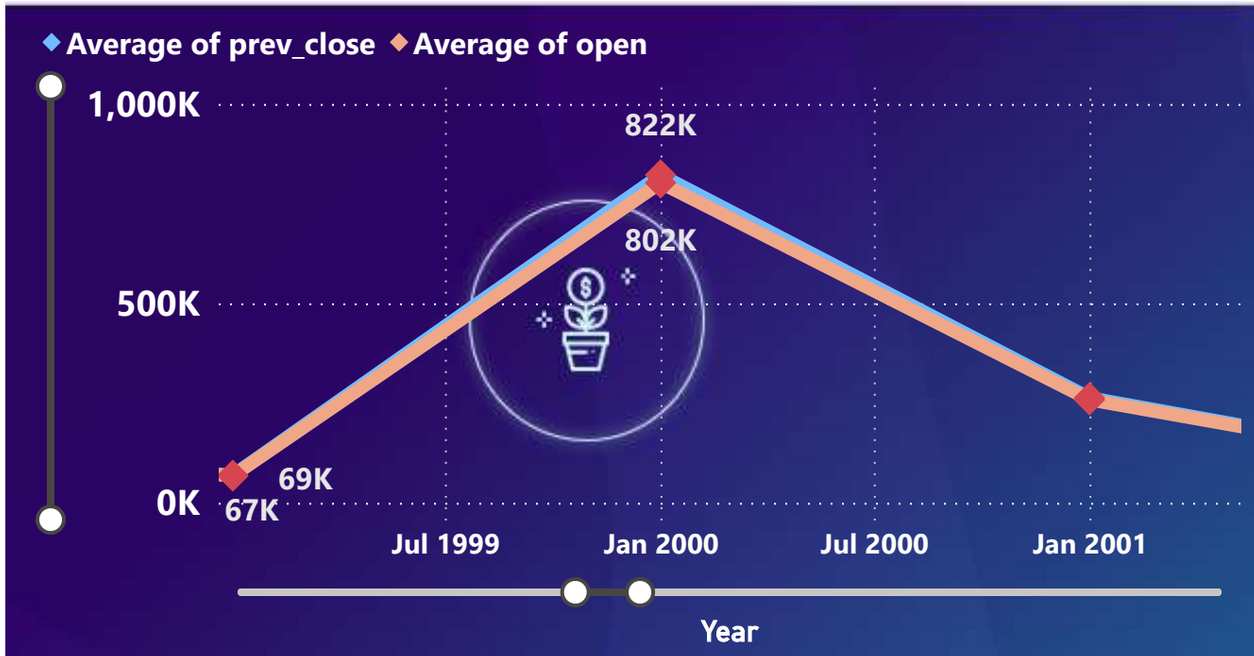


Volume To High Ratio

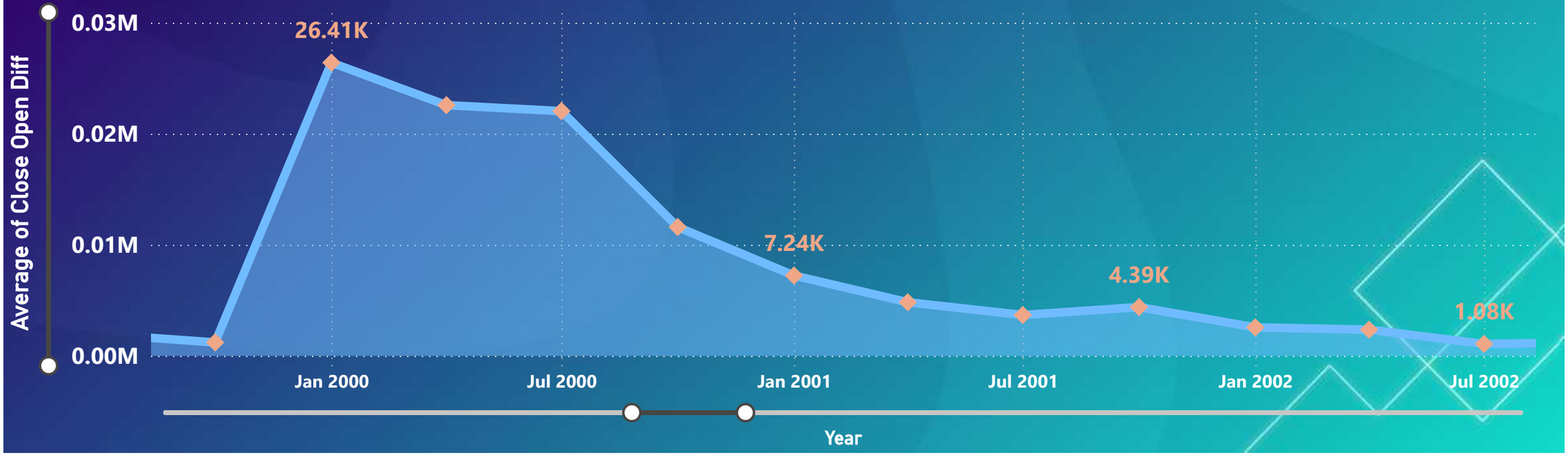
Observations :

1. Market Interest and Liquidity: The volume-to-high ratio provides insights into the level of market interest and liquidity for a particular stock. If the ratio is high, it indicates that there is a significant amount of trading activity occurring at or near the stock's high price.

2. Also it may not be a ideal time to sale your stock, because many people are buying at very low price.



Year	Quarter	Month	Day	prev_close	open	Difference
1987	Qtr 3	August	10	5.56	5.50	0.06
1987	Qtr 3	August	26	6.94	6.88	0.06
1987	Qtr 3	September	3	6.38	6.25	0.13
1987	Qtr 3	September	15	5.94	5.88	0.06
1987	Qtr 3	September	16	5.88	5.81	0.06
1987	Qtr 3	September	17	5.63	5.56	0.06
1987	Qtr 3	September	29	5.69	5.63	0.06
1987	Qtr 4	October	9	5.75	5.69	0.06
1987	Qtr 4	October	12	5.69	5.63	0.06
1987	Qtr 4	October	19	4.50	4.31	0.19
1987	Qtr 4	October	26	3.63	3.56	0.06

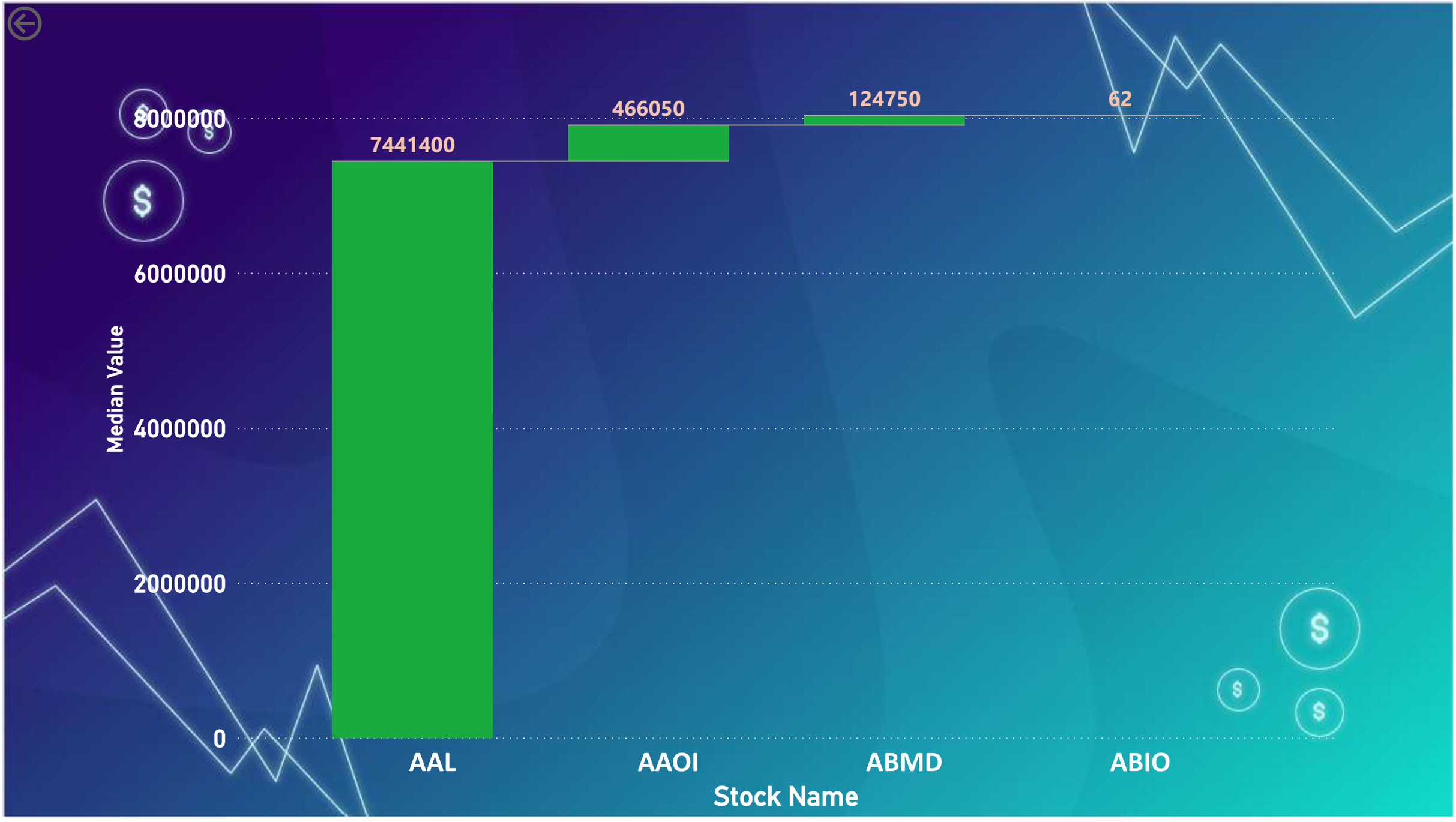


Previous Day Close & Current Day Open Price Difference

Observations :

1. Overnight Sentiment and Price Gaps: The query helps identify dates where there is a positive difference between the previous day's close and the current day's open for each stock. This difference indicates a price gap, reflecting overnight sentiment and potential market events that occurred outside of regular trading hours. A positive difference suggests that there was generally positive sentiment or news overnight that caused the stock to open higher than the previous day's close. This insight can be valuable for investors and traders to understand the impact of overnight developments on stock prices and to potentially capitalize on price gaps.

1.


8000000
6000000
4000000
2000000
0
Median Value
7441400
466050
124750
62
AAL
AAOI
ABMD
ABIO
Stock Name

Median Value Analysis

Observations :

Central Tendency - If the Median value is near to mean value then it will follow the central limit theorem and the prediction can be good, investor can make use of this to invest in stocks.

1.

HBASE ANALYSIS

Creation of an Empty Table in HBASE :

```
hbase(main):003:0> create table 'walmart' values
```

Importing data in HBASE :

```
[cloudera@quickstart Desktop]$ sqoop import --connect jdbc:mysql://localhost:3306/stoc
k_analysis --username root --password cloudera --table walmart_stock --hbase-table wal
mart --columns "date,Open,High,Low,Close,Volume,Adj_Close"  --hbase-row-key date --col
umn-family values --hbase-create-table -m 1
```

Top 5 rows in HBASE :

```
hbase(main):002:0> scan 'walmart', {LIMIT=>5}
ROW                   COLUMN+CELL
 2012-01-03           column=values:Adj_Close, timestamp=1689345828549, value=52
                      .6192
 2012-01-03           column=values:Close, timestamp=1689345828549, value=60.33
 2012-01-03           column=values:High, timestamp=1689345828549, value=61.06
 2012-01-03           column=values:Low, timestamp=1689345828549, value=59.87
 2012-01-03           column=values:Open, timestamp=1689345828549, value=59.97
 2012-01-03           column=values:Volume, timestamp=1689345828549, value=12668
                      800
 2012-01-04           column=values:Adj_Close, timestamp=1689345828549, value=52
                      .0785
 2012-01-04           column=values:Close, timestamp=1689345828549, value=59.71
 2012-01-04           column=values:High, timestamp=1689345828549, value=60.35
 2012-01-04           column=values:Low, timestamp=1689345828549, value=59.47
 2012-01-04           column=values:Open, timestamp=1689345828549, value=60.21
 2012-01-04           column=values:Volume, timestamp=1689345828549, value=95933
                      00
 2012-01-05           column=values:Adj_Close, timestamp=1689345828549, value=51
                      .8255
 2012-01-05           column=values:Close, timestamp=1689345828549, value=59.42
 2012-01-05           column=values:High, timestamp=1689345828549, value=59.62
 2012-01-05           column=values:Low, timestamp=1689345828549, value=58.37
 2012-01-05           column=values:Open, timestamp=1689345828549, value=59.35
 2012-01-05           column=values:Volume, timestamp=1689345828549, value=12768
                      200
 2012-01-06           column=values:Adj_Close, timestamp=1689345828549, value=51
                      .4592
 2012-01-06           column=values:Close, timestamp=1689345828549, value=59.0
 2012-01-06           column=values:High, timestamp=1689345828549, value=59.45
 2012-01-06           column=values:Low, timestamp=1689345828549, value=58.87
 2012-01-06           column=values:Open, timestamp=1689345828549, value=59.42
 2012-01-06           column=values:Volume, timestamp=1689345828549, value=80694
                      00
 2012-01-09           column=values:Adj_Close, timestamp=1689345828549, value=51
                      .6162
 2012-01-09           column=values:Close, timestamp=1689345828549, value=59.18
 2012-01-09           column=values:High, timestamp=1689345828549, value=59.55
 2012-01-09           column=values:Low, timestamp=1689345828549, value=58.92
 2012-01-09           column=values:Open, timestamp=1689345828549, value=59.03
 2012-01-09           column=values:Volume, timestamp=1689345828549, value=66793
                      00
5 row(s) in 0.0770 seconds

hbase(main):003:0>
```

OPTIMIZATION TECHNIQUES USED

Following are the optimization techniques used :

Indexing (User story 6) :

Old Result : Time -158 sec

```
2022-10-10      2.0     2.07    0.06999993
2022-10-11      2.01    2.04    0.029999971
2022-10-13      1.98    2.02    0.03999996
2022-10-18      2.06    2.1     0.03999996
2022-10-21      2.04    2.06    0.01999998
2022-10-24      2.05    2.08    0.029999971
2022-10-28      2.05    2.09    0.03999996
2022-10-31      2.12    2.18    0.06000018
2022-11-03      2.1     2.13    0.03000021
2022-11-04      2.05    2.08    0.029999971
2022-11-07      2.02    2.06    0.03999996
2022-11-14      2.0     2.01    0.00999999
2022-11-22      2.05    2.07    0.01999998
2022-11-23      2.07    2.08    0.00999999
2022-11-25      2.06    2.1     0.03999996
2022-11-28      2.01    2.08    0.06999993
2022-12-01      2.03    2.09    0.059999943
2022-12-02      2.03    2.07    0.03999996
2022-12-05      2.1     2.11    0.00999999
2022-12-06      2.11    2.13    0.02000022
2022-12-07      2.07    2.1     0.029999971
2022-12-09      2.14    2.2     0.059999943
Time taken: 158.059 seconds, Fetched: 2405 row(s)
```

New Result after indexing : Time -28 sec

```
2022-10-10      2.0     2.07    0.06999993
2022-10-11      2.01    2.04    0.029999971
2022-10-13      1.98    2.02    0.03999996
2022-10-18      2.06    2.1     0.03999996
2022-10-21      2.04    2.06    0.01999998
2022-10-24      2.05    2.08    0.029999971
2022-10-28      2.05    2.09    0.03999996
2022-10-31      2.12    2.18    0.06000018
2022-11-03      2.1     2.13    0.03000021
2022-11-04      2.05    2.08    0.029999971
2022-11-07      2.02    2.06    0.03999996
2022-11-14      2.0     2.01    0.00999999
2022-11-22      2.05    2.07    0.01999998
2022-11-23      2.07    2.08    0.00999999
2022-11-25      2.06    2.1     0.03999996
2022-11-28      2.01    2.08    0.06999993
2022-12-01      2.03    2.09    0.059999943
2022-12-02      2.03    2.07    0.03999996
2022-12-05      2.1     2.11    0.00999999
2022-12-06      2.11    2.13    0.02000022
2022-12-07      2.07    2.1     0.029999971
2022-12-09      2.14    2.2     0.059999943
Time taken: 28.721 seconds, Fetched: 2405 row(s)
```

OPTIMIZATION TECHNIQUES USED

Following are the optimization techniques used :

Bucketing (User Story 7) :

Old Result : Time -392 sec

```
MapReduce Total cumulative CPU time: 4 seconds 690 msec
Ended Job = job_1689150900907_0081
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 9.48 sec   HDFS Read: 377333
HDFS Write: 141078 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 9.87 sec   HDFS Read: 150139
HDFS Write: 127 SUCCESS
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 4.69 sec   HDFS Read: 5105 HD
FS Write: 5 SUCCESS
Total MapReduce CPU Time Spent: 24 seconds 40 msec
OK
62.0
Time taken: 392.513 seconds, Fetched: 1 row(s)
```

New Result after Bucketing : Time - 148

```
2023-07-16 03:41:11,787 Stage-3 map = 100%,  reduce = 100%, Cumulative CPU 4.47
sec
MapReduce Total cumulative CPU time: 4 seconds 470 msec
Ended Job = job_1689395438209_0036
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 8.51 sec   HDFS Read: 333866
HDFS Write: 141078 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 9.11 sec   HDFS Read: 150153
HDFS Write: 127 SUCCESS
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 4.47 sec   HDFS Read: 5105 HD
FS Write: 5 SUCCESS
Total MapReduce CPU Time Spent: 22 seconds 90 msec
OK
62.0
Time taken: 148.513 seconds, Fetched: 1 row(s)
```

OPTIMIZATION TECHNIQUES USED

Following are the optimization techniques used :

Compressing To ORC file format (User Story 4) :

Old Result : Time - 68 sec

```
10-11-2022      14.46   2.23    35599100        2849100
11-11-2022      15.0    2.24    24134000        1665200
14-11-2022      14.79   2.28    26266600        2754200
15-11-2022      15.02   2.4     29423100        3660200
16-11-2022      14.45   2.3     28858000        1667800
17-11-2022      13.82   2.1     24041700        1621700
18-11-2022      14.26   2.18    25968300        1678600
21-11-2022      14.02   2.12    25708400        964500
22-11-2022      13.85   2.2     26387000        661500
23-11-2022      13.98   2.268   23686400        801800
25-11-2022      14.4    2.18    9903900 258900
28-11-2022      14.34   2.18    21313800        830300
29-11-2022      13.89   2.11    17335300        592000
30-11-2022      14.12   2.1     21195200        733900
01-12-2022      14.46   2.18    26519600        644000
02-12-2022      13.82   2.17    24094600        704800
05-12-2022      13.84   2.33    27029000        886200
06-12-2022      14.14   2.28    20781500        672300
07-12-2022      14.24   2.1     28161400        974100
08-12-2022      13.65   2.13    25300900        1506900
09-12-2022      13.52   2.15    18489800        990100
12-12-2022      13.49   2.11    8048550 412703
Time taken: 67.921 seconds, Fetched: 2320 row(s)
```

New Result after Compressing to ORC : Time - 26 sec

```
10-11-2022      14.46   2.23    35599100        2849100
11-11-2022      15.0    2.24    24134000        1665200
14-11-2022      14.79   2.28    26266600        2754200
15-11-2022      15.02   2.4     29423100        3660200
16-11-2022      14.45   2.3     28858000        1667800
17-11-2022      13.82   2.1     24041700        1621700
18-11-2022      14.26   2.18    25968300        1678600
21-11-2022      14.02   2.12    25708400        964500
22-11-2022      13.85   2.2     26387000        661500
23-11-2022      13.98   2.268   23686400        801800
25-11-2022      14.4    2.18    9903900 258900
28-11-2022      14.34   2.18    21313800        830300
29-11-2022      13.89   2.11    17335300        592000
30-11-2022      14.12   2.1     21195200        733900
01-12-2022      14.46   2.18    26519600        644000
02-12-2022      13.82   2.17    24094600        704800
05-12-2022      13.84   2.33    27029000        886200
06-12-2022      14.14   2.28    20781500        672300
07-12-2022      14.24   2.1     28161400        974100
08-12-2022      13.65   2.13    25300900        1506900
09-12-2022      13.52   2.15    18489800        990100
12-12-2022      13.49   2.11    8048550 412703
Time taken: 26.703 seconds, Fetched: 2320 row(s)
```

SPARK ANALYSIS WITH RDD & SPARK SQL

Scenario 2:There are too many decimal places for mean and stddev in the describe() dataframe. Format the numbers to just show up to two decimal places. Pay careful attention to the datatypes that .describe() returns, we didn't cover how to do this exact formatting, but we covered something very similar.

```
: df.describe().show()
```

...

```
: df.describe().withColumn("Open", format_number(col("Open").cast('float'),2))\
.withColumn("High", format_number(col("High").cast('float'),2))\
.withColumn("Low", format_number(col("Low").cast('float'),2))\
.withColumn("Close", format_number(col("Close").cast('float'),2))\
.withColumn("Volume", format_number(col("Volume").cast('float'),2))\
.withColumn("Adj Close", format_number(col("Adj Close").cast('float'),2))\
.show()
```

```
+-------+--------+--------+--------+--------+-------------+---------+
|summary|    Open|    High|     Low|   Close|       Volume|Adj Close|
+-------+--------+--------+--------+--------+-------------+---------+
|  count|1,258.00|1,258.00|1,258.00|1,258.00|     1,258.00| 1,258.00|
|   mean|   72.36|   72.84|   71.92|   72.39| 8,222,093.50|    67.24|
| stddev|    6.77|    6.77|    6.74|    6.76| 4,519,781.00|     6.72|
|    min|   56.39|   57.06|   56.30|   56.42| 2,094,900.00|    50.36|
|    max|   90.80|   90.97|   89.25|   90.47|80,898,096.00|    84.91|
+-------+--------+--------+--------+--------+-------------+---------+
```

SPARK ANALYSIS WITH RDD & SPARK SQL

Scenario 3: Create a new dataframe with a column called HV Ratio that is the ratio of the High Price versus volume of stock traded for a day.?

```
# Spark - SQL Technique

ss.sql("select *, round(cast(Volume as float)/cast(High as float),2) as HV_Ratio from walmartstock").show()
```

```
+----------+-----+-----+-----+-----+--------+---------+---------+
|      Date| Open| High|  Low|Close|  Volume|Adj Close| HV_Ratio|
+----------+-----+-----+-----+-----+--------+---------+---------+
|2012-01-03|59.97|61.06|59.87|60.33|12668800|52.619236|207481.16|
|2012-01-04|60.21|60.35|59.47|59.71| 9593300|52.078476|158961.06|
|2012-01-05|59.35|59.62|58.37|59.42|12768200| 51.82554|214159.68|
|2012-01-06|59.42|59.45|58.87| 59.0| 8069400| 51.45922|135734.23|
|2012-01-09|59.03|59.55|58.92|59.18| 6679300|51.616215|112162.89|
|2012-01-10|59.43|59.71|58.98|59.04| 6907300| 51.49411|115680.79|
|2012-01-11|59.06|59.53|59.04| 59.4| 6365600|51.808098|106930.96|
|2012-01-12|59.79| 60.0| 59.4| 59.5| 7236400|51.895317|120606.67|
```

```
# DSL Technique

df.withColumn('HV Ratio',round(lit(col('Volume').cast('float')/col('High').cast('float')),2)).show()
```

```
+----------+-----+-----+-----+-----+--------+---------+---------+
|      Date| Open| High|  Low|Close|  Volume|Adj Close| HV Ratio|
+----------+-----+-----+-----+-----+--------+---------+---------+
|2012-01-03|59.97|61.06|59.87|60.33|12668800|52.619236|207481.16|
|2012-01-04|60.21|60.35|59.47|59.71| 9593300|52.078476|158961.06|
|2012-01-05|59.35|59.62|58.37|59.42|12768200| 51.82554|214159.68|
|2012-01-06|59.42|59.45|58.87| 59.0| 8069400| 51.45922|135734.23|
|2012-01-09|59.03|59.55|58.92|59.18| 6679300|51.616215|112162.89|
|2012-01-10|59.43|59.71|58.98|59.04| 6907300| 51.49411|115680.79|
|2012-01-11|59.06|59.53|59.04| 59.4| 6365600|51.808098|106930.96|
|2012-01-12|59.79| 60.0| 59.4| 59.5| 7236400|51.895317|120606.67|
```

SPARK ANALYSIS WITH RDD & SPARK SQL

Scenario 4: What day had the Peak High in Price?

In [28]:
```
# Spark-SQL techinique

ss.sql("select Date, High from walmartstock order by High desc limit 1").show()
```

```
+----------+-----+
|      Date| High|
+----------+-----+
|2015-01-13|90.97|
+----------+-----+
```

In [29]:
```
# DSL

df.orderBy(desc('High')).select("Date", "High").limit(1).show()
```

```
+----------+-----+
|      Date| High|
+----------+-----+
|2015-01-13|90.97|
+----------+-----+
```

SPARK ANALYSIS WITH RDD & SPARK SQL

Scenario 5: What is the mean of the Close column?

```
from pyspark.sql.functions import round
```

```
# Spark - SQL Technique

ss.sql("select round(avg(Close),2) as mean_close from walmartstock").show()
```

```
+----------+
|mean_close|
+----------+
|     72.39|
+----------+
```

```
# DSL Technique

df.select(round(mean(df.Close),2).alias("Mean_Close")).show()
```

```
+----------+
|Mean_Close|
+----------+
|     72.39|
+----------+
```

SPARK ANALYSIS WITH RDD & SPARK SQL

Scenario 6: What is the max and min of the Volume column?

```
]: # Spark - Sql Technique

ss.sql("select round(max(Volume)) as max_volume, round(min(Volume)) as min_volume from walmartstock").show()
```

```
+----------+----------+
|max_volume|min_volume|
+----------+----------+
|  80898100|   2094900|
+----------+----------+
```

```
]: # DSL Technique

df.select(round(max(df.Volume),2).alias("Max_Volume"), round(min(df.Volume),2).alias("Min_Volume")).show()
```

```
+----------+----------+
|Max_Volume|Min_Volume|
+----------+----------+
|  80898100|   2094900|
+----------+----------+
```

SPARK ANALYSIS WITH RDD & SPARK SQL

Scenario 6: What is the max and min of the Volume column?

```
]: # Spark - Sql Technique

ss.sql("select round(max(Volume)) as max_volume, round(min(Volume)) as min_volume from walmartstock").show()
```

```
+----------+----------+
|max_volume|min_volume|
+----------+----------+
|  80898100|   2094900|
+----------+----------+
```

```
]: # DSL Technique

df.select(round(max(df.Volume),2).alias("Max_Volume"), round(min(df.Volume),2).alias("Min_Volume")).show()
```

```
+----------+----------+
|Max_Volume|Min_Volume|
+----------+----------+
|  80898100|   2094900|
+----------+----------+
```

SPARK ANALYSIS WITH RDD & SPARK SQL

-- RDD --

Scenario 7: How many days was the Close lower than 60 dollars?

```
walmartrdd2 = walmartrdd.map(lambda line:(line[0],int(line[4])))\
.filter(lambda item:float(item[1]) < 60)\
.map(lambda x: (x[0],1))\
.count()
```

```
print(f"{walmartrdd2} days was the close lower than 60 dollars.")
```

```
81 days was the close lower than 60 dollars.
```

Scenario 8: What percentage of the time was the High greater than 80 dollars ?

[13]:
```
walmartrdd3 = (walmartrdd.map(lambda line:(line[0],int(line[2])))\
.filter(lambda item:float(item[1]) > 80)\
.map(lambda x: (x[0],1))\
.count())/walmartrdd.count()*100
```

[14]:
```
print(f"{round(walmartrdd3, 2)} % of the time was the high greater than 80 dollars.")
```

```
8.43 % of the time was the high greater than 80 dollars.
```

Scenario 9: What is the max High per year?

[15]:
```
walmartrdd4 = walmartrdd.map(lambda x: (int(x[0].split('-')[0]), x[2]))\
.reduceByKey(lambda a, b: round(a,2) if a>b else round(b,2))
```

[16]:
```
print("Max High Per Year: %s"%walmartrdd4.collect())
```

```
Max High Per Year: [(2016, 75.19), (2012, 77.6), (2013, 81.37), (2014, 88.09), (2015, 90.97)]
```

SPARK STREAMING ANALYSIS

```
import pyspark
from pyspark.streaming import StreamingContext
sc = pyspark.SparkContext('local[2]','StreamingWordCount')
ssc=StreamingContext(sc,4)
sts=ssc.socketTextStream('localhost',9999)
price=sc.textFile("/user/cloudera/previous_max_price.csv")
p1=price.map(lambda x:x.split(',')).map(lambda x:(x[0],float(x[1])))
p1.take(2)
p2=p1.collectAsMap()
price_broad=sc.broadcast(p2)

r1=sts.map(lambda x:x.split(',')).map(lambda x:(x[0],x[1],float(x[2])))
r2=r1.map(lambda x: (x[0],x[1],x[2],x[2]>=price_broad.value.get(x[0]) if x[0] in price_broad.value else 'None')
)
r2.pprint()
ssc.start()
ssc.awaitTermination()
```

```
-------------------------------------------
Time: 2023-07-18 04:14:36
-------------------------------------------
(u'JAS', u'08:22:33', 26.699999999999999, False)
(u'JBR', u'08:22:34', 22.5, False)
(u'JAS', u'08:22:35', 27.210000000000001, False)
(u'JWF', u'08:22:36', 22.859999999999999, False)
(u'JEQ', u'08:22:37', 15.539999999999999, False)
(u'JAH', u'08:22:38', 38.979999999999997, False)
(u'JHX', u'08:22:39', 37.630000000000003, False)
(u'JFP', u'08:22:40', 12.58, 'None')
```

Conclusion

In conclusion, the stock analysis project serves as a valuable tool for individuals who aim to earn from the stock market but may lack the capacity for in-depth analysis. With its customizable dashboard and market-ready insights, it empowers common investors to confidently navigate the stock market and make informed investment decisions, even without extensive knowledge of the market. By providing accessible and user-friendly analytics, this project opens doors for individuals to participate in the stock market with ease and potential success.

1.

Credits

Thank You Venkat Sir, Bharathan & Satwik for guiding me through the project

1.