Distributed Financial Risk Assessment

Sai Preetham Bojja (Group 7)

This report outlines the process of data profiling, cleaning, and ingestion of stock market data using Apache Spark.

1 Data Source

I am working with stocks listed on the London Stock Exchange. I have downloaded the latest stock ticker symbols from the official website. Using the Python module yfinance, I obtained historical stock data from Yahoo Finance via API calls. The data spans from 1st January 2000 to 31st October 2024 and was retrieved using a Bash script that utilizes the Python module. This dataset is stored in the Hadoop Distributed File System (HDFS).

Each CSV file corresponds to a different stock symbol and contains daily closing prices along with other related financial metrics. The files are named following the pattern <StockSymbol¿.L.csv. An example snippet of one of the CSV files is shown in Table 1.

Date	AdjClose	Close	Open	High	Low	Volume
2021-01-04	150.0	150.0	152.0	153.0	149.0	1000000
2021-01-05	151.0	151.0	150.5	152.5	149.5	1100000
2021-01-06	149.5	149.5	151.0	151.5	148.0	1050000

Table 1: Snippet of the stock data CSV file

2 Data Profiling

This information is crucial for understanding the quality of the data and identifying any anomalies or missing values that need to be addressed during the cleaning process. In profiling we calculated the following metrics:

- The range of dates for which data is available for each stock.
- The minimum and maximum closing prices.
- The number of null values in the `Close` column.
- The standard deviation of the closing prices.

The following figures show the metrics of each stock and the no of stocks listed in each year with data profiling.

StockSymbol	StartDate	MaxValue	MinValue	NullCount	StdDev
AAEV	2007-04-05	126.0	67.0	0	16.314816655233766
AAF	2019-06-28	170.89999389648438	27.799999237060547	0	30.317972992775456
AAIF	2005-12-20	244.75	81.5	0	42.65895486669987
AAL	2000-01-03	4170.5	220.0312042236328	0	782.5772395110702
AAS	2000-01-03	664.0	16.100000381469727	0	119.40790251413108
AATG	2001-01-17	108.0	63.45000076293945	0	10.296072846604547
AAU	2005-07-28	17.5	0.6999999988079071	0	2.849958972918156
AAVC	2000-01-03	125.0	0.41600000858306885	0	23.563625187841094
AAZ	2005-07-29	177.0	2.75	0	43.362162519625386
ABDN	2006-07-10	640.4442749023438	133.0	0	114.73140789146125
ABDP	2013-05-22	2850.0	101.85199737548828	0	768.2431900151099
ABDX	2020-12-15	125.62550354003906	3.75	0	26.79249762403279
ABF	2000-01-03	3599.0	294.25	0	908.9083963874846
AC8	2021-07-19	425.0	0.8999999761581421	0	103.67077276487083

Figure 1: Metric of each stock

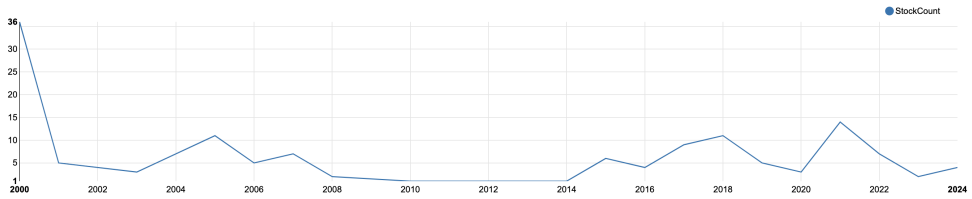

Figure 2: Number of stocks listed in each year

3 Data Cleaning

Data cleaning involves handling missing values and ensuring data consistency. In this case, missing closing prices are forward-filled and backward-filled to maintain continuity.

3.1 Cleaning Strategy

The cleaning strategy includes:

- **Forward Filling**: For each null value in the `Close` column, replace it with the last non-null value encountered in previous rows.
- **Backward Filling**: If any null values remain at the beginning of the dataset, replace them with the next non-null value.

This approach ensures that all missing closing prices are filled with the most recent available data, maintaining the integrity of time series analyses.

The initial data had all Date, Adj Close, Close, Open, High, Low and Volume, after using the cleaning strategy for the project we only need the Date and Close so we only select them, so here is the snippet of cleansed data of one of the stock.

```
+----------+------+
|      Date| Close|
+----------+------+
|2005-06-21|1120.0|
|2005-06-22|1135.0|
|2005-06-23|1160.0|
|2005-06-24|1195.0|
|2005-06-27|1190.0|
|2005-06-28|1225.0|
|2005-06-29|1200.0|
|2005-06-30|1200.0|
|2005-07-01|1185.0|
|2005-07-04|1185.0|
|2005-07-05|1240.0|
|2005-07-06|1325.0|
|2005-07-07|1365.0|
|2005-07-08|1375.0|
```

Figure 3: Number of stocks listed in each year

4 Data Ingestion

After cleaning, the data is ingested by writing the cleaned DataFrames back to HDFS in CSV format, stored in HDFS. The data ingestion step is integrated into the data cleaning code, as shown in the last line of the cleaning code snippet:

```
cleanedDf.write.option("header", "true")
  .mode("overwrite")
  .csv(s"/user/sb9509_nyu_edu/stocks_cleaned/$stockName")
```