BÁO CÁO PROJECT: TRIỂN KHAI HỆ THỐNG PHÂN TÍCH VÀ XỬ LÝ DỮ LIỆU LỚN

Case Study: Phân tích Dữ liệu Giá Chứng khoán Việt Nam

Hoang Manh Hung
23020371
UET-VNU

Mục lục

1 Giới thiệu Bài toán

Thị trường chứng khoán là một môi trường đầy biến động, với hàng triệu lượt giao dịch diễn ra mỗi ngày. Việc áp dụng xử lý dữ liệu lớn (Big Data) vào phân tích chứng khoán đã trở thành một công cụ hữu hiệu, giúp các nhà đầu tư đưa ra quyết định thông minh và hạn chế rủi ro.

Project này hướng đến việc mô phỏng một hệ thống big data sử dụng **Hadoop** để lưu trữ và **Spark** để xử lý, phân tích dữ liệu lịch sử giá của các cổ phiếu trong rổ VN30, được lấy từ bộ dữ liệu công khai trên Kaggle.

Mục tiêu:

- Vận dụng kỹ thuật phân tích và xử lý dữ liệu lớn.
- Xây dựng, triển khai hệ thống dữ liệu lớn mô phỏng với Docker.
- Trích xuất thông tin chi tiết (insights) có giá trị từ dữ liệu giá chứng khoán.

1.1 Mô tả Dữ liệu

Dữ liệu được sử dụng trong project là bộ dữ liệu lịch sử giá của các cổ phiếu trong rổ VN30 (Index Việt Nam), được thu thập từ Kaggle. Dữ liệu bao gồm các file CSV riêng lẻ cho từng mã cổ phiếu, với cấu trúc các trường như sau:

- `<Ticker>`: Mã định danh của cổ phiếu (ví dụ: FPT, HPG, VCB).
- `<DTYYYYMMDD>`: Ngày giao dịch theo định dạng `YYYYMMDD`.
- `<Open>`: Giá cổ phiếu lúc mở cửa phiên giao dịch.
- `<High>`: Giá cổ phiếu cao nhất đạt được trong phiên giao dịch.
- `<Low>`: Giá cổ phiếu thấp nhất trong phiên giao dịch.
- `<Close>`: Giá cổ phiếu lúc đóng cửa phiên giao dịch.
- `<Volume>`: Tổng khối lượng cổ phiếu được giao dịch trong ngày.

2 Kiến trúc Hệ thống

Hệ thống được xây dựng mô phỏng dựa trên nền tảng Docker và các image của Big Data Europe (bde2020), bao gồm các thành phần chính:

- **Cụm Hadoop HDFS (Lưu trữ):**
  - **1 Namenode:** Quản lý metadata của hệ thống file.
  - **4 Datanode:** Lưu trữ dữ liệu thực tế (các block dữ liệu).
- **Cụm Spark (Xử lý):**
  - **1 Spark Master:** Quản lý và điều phối tài nguyên cho các ứng dụng Spark.
  - **4 Spark Worker:** Các node thực thi các tác vụ (tasks) xử lý dữ liệu.
- **Giao diện Phân tích:**
  - **1 Jupyter Notebook (PySpark):** Môi trường để lập trình, gửi các tác vụ phân tích đến cụm Spark và trực quan hóa kết quả.

3 Triển khai và Xử lý Dữ liệu (ETL)

3.1 Nạp Dữ liệu (Data Ingestion)

Dữ liệu thô (tập hợp các file `.csv` của các mã chứng khoán) được tải lên container Jupyter, sau đó được nạp vào Hệ thống tệp phân tán HDFS tại đường dẫn `hdfs://namenode:8020/datack/` để sẵn sàng cho việc phân tích.

3.2 Đọc và Làm sạch Dữ liệu (Data Cleansing)

Chúng tôi sử dụng PySpark để đọc toàn bộ dữ liệu CSV từ HDFS vào một DataFrame Spark duy nhất.

```
df = spark.read.csv("hdfs://namenode:8020/datack/*.csv",
                    header=True, inferSchema=True)
```

Quá trình làm sạch và chuyển đổi (Transformation) bao gồm:

1. **Chuyển đổi Kiểu dữ liệu:** Cột `<DTYYYYMMDD>` (ví dụ: `20231024`) được chuyển đổi sang kiểu `Date` để phân tích theo thời gian.

```
from pyspark.sql.functions import col, to_date

df = df.withColumn("Date",
    to_date(col("<DTYYYYMMDD>").cast("string"), "yyyyMMdd"))
```

2. **Định kiểu Cột:** Đảm bảo các cột số như `<Close>`, `<Volume>` được ép kiểu chính xác sang `DoubleType` và `IntegerType`.

3. **Xử lý Dữ liệu Khuyết (Null):** Loại bỏ bất kỳ dòng nào có dữ liệu bị thiếu (`df.dropna()`).

4. **Tạo Cột Phân tích (Feature Engineering):**

   - `ChangePct`: Tính toán phần trăm thay đổi giá trong ngày `(<Close> - <Open>) / <Open>`.
   - `Daily_Spread`: Tính toán mức biến động giá trong ngày `(<High> - <Low>)`.

4 Phân tích Dữ liệu

Sau khi dữ liệu đã được làm sạch, chúng tôi tiến hành các phân tích sau để trích xuất thông tin chi tiết:

4.1 Phân tích Thống kê Mô tả

Tính toán các chỉ số thống kê cơ bản như giá đóng cửa trung bình, giá cao nhất, giá thấp nhất và khối lượng giao dịch trung bình cho từng mã cổ phiếu (`<Ticker>`). Kết quả này cho phép so sánh tổng quan về hiệu suất và quy mô giao dịch của các mã trong rổ VN30.

```
from pyspark.sql.functions import avg, max, min
basic_stats = df.groupBy("<Ticker>") \
    .agg(
        avg("<Close>").alias("Avg_Close"),
        max("<Close>").alias("Max_Close"),
        min("<Close>").alias("Min_Close"),
        avg("<Volume>").alias("Avg_Volume")
    )
basic_stats.show()
```

```
+--------+------------------+---------+---------+------------------+
|<Ticker>|         Avg_Close|Max_Close|Min_Close|        Avg_Volume|
+--------+------------------+---------+---------+------------------+
|     SSI|15.190702662229558|     53.7|   4.1745|2670544.1092623407|
|     SBT|11.576043729323354|  36.6705|   1.8026|1356472.7978947368|
|     PNJ| 31.90247203278712|    101.9|   4.2822|222202.55704918032|
|     MBB|11.821110552555012|     42.1|   3.8988| 3885864.333194848|
|     TCH|20.075031186440693|  43.1156|  12.9066|2465284.2118644067|
|     VNM|46.306322358146765| 132.7491|    2.728| 558889.2277459656|
|     VPB|27.287067289719666|     72.0|    16.95| 5010468.660436137|
|     TPB|19.957316687578437|     39.0|     13.5| 1188471.957340025|
|     VIC|41.922699270711774|    144.0|   2.4293| 639301.5306301051|
|     CTG| 17.79040640724583|     54.0|   8.0049|2583112.8329419657|
|     KDH| 12.63406981331453|    37.35|    2.481| 317416.0852412821|
|     BID|25.824845533297072|  54.5776|  10.1164|1895361.8733080672|
|     NVL| 45.30150633928572|    119.1|  32.1527| 1290022.080357143|
|     HDB| 24.37893414351854|  38.4277|  13.2653|2432274.8032407407|
|     TCB|26.574522656250014|     54.8|     14.9| 5588575.442708333|
|     STB|13.103224644295187|     33.8|   5.0273|4164020.8614765103|
|     REE|14.946464042721546|     60.0|   0.8574| 609855.1212420886|
|     FPT|19.697239424937724|     87.1|   3.5803| 771563.0030411944|
|     HPG| 8.153285098849206|     55.5|   0.9945|3466311.2475656536|
|     VCB| 37.27980478100963|    112.6|  10.0958| 916478.8585757272|
+--------+------------------+---------+---------+------------------+
only showing top 20 rows
```

4.2 Phân tích Khối lượng Giao dịch (Thanh khoản)

Tính tổng khối lượng giao dịch (<Volume>) theo từng mã cổ phiếu trong toàn bộ khoảng thời gian dữ liệu. Phân tích này giúp xác định những mã cổ phiếu có tính thanh khoản cao, tức là được mua bán nhiều nhất trên thị trường.

```
1 from pyspark.sql.functions import sum
2 df.groupBy("<Ticker>").sum("<Volume>").orderBy("sum(<Volume>)", ascending=False).show
    (10, truncate=False)
```

```
Tổng khối lượng giao dịch theo mã:
+--------+--------------+
|<Ticker>|sum(<Volume>)|
+--------+--------------+
|STB     |15510977709   |
|HPG     |11747328818   |
|SSI     |9629982058    |
|MBB     |9353275450    |
|CTG     |7700259355    |
|VPB     |4825081320    |
|SBT     |4510272053    |
|TCB     |4292025940    |
|POW     |3796262502    |
|BID     |3500733380    |
+--------+--------------+
only showing top 10 rows
```

4.3 Phân tích Biến động giá (Volatility)

Tính toán mức chênh lệch trung bình hàng ngày giữa giá cao nhất (`<High>`) và giá thấp nhất (`<Low>`) cho mỗi mã cổ phiếu. Chỉ số `Avg_Daily_Spread` cao cho thấy mã cổ phiếu có biến động giá lớn trong ngày, phù hợp với các chiến lược đầu tư ngắn hạn hoặc "lướt sóng".

```
df = df.withColumn("Daily_Spread", col("<High>") - col("<Low>"))
volatility = df.groupBy("<Ticker>") \
    .agg(avg("Daily_Spread").alias("Avg_Daily_Spread")) \
    .orderBy(col("Avg_Daily_Spread").desc())
volatility.show()
```

```
+--------+------------------+
|<Ticker>|   Avg_Daily_Spread|
+--------+------------------+
|     VJC|    2.4267292051756|
|     VHM| 2.1223289203084876|
|     MSN| 1.6634417643004336|
|     MWG| 1.6611279792746232|
|     GAS| 1.6438412411971839|
|     BVH| 1.5729290387182913|
|     PLX| 1.3748940669856473|
|     NVL| 1.0763658928571438|
|     VIC|  1.007780280046671|
|     VRE| 0.9589857300884957|
|     VCB| 0.9043438649281162|
|     VNM| 0.8757835502342615|
|     VPB| 0.8548030114226376|
|     PNJ| 0.7870733442622919|
|     BID| 0.7636200324851152|
|     TCB| 0.7072265625000006|
|     HDB| 0.6785876157407408|
|     TCH| 0.6722802542372888|
|     TPB|    0.5234991217064|
|     CTG|0.47108802415296813|
+--------+------------------+
only showing top 20 rows
```

4.4 Phân tích Xu hướng Giá theo Thời gian

Phân tích diễn biến giá đóng cửa (`<Close>`) theo thời gian (`Date`):

- So sánh xu hướng giá của một số mã cổ phiếu tiêu biểu (ví dụ: VCB, FPT, HPG, MWG) để thấy sự khác biệt trong tăng trưởng giữa các ngành.
- Tính giá đóng cửa trung bình của toàn bộ rổ VN30 theo từng năm để đánh giá xu hướng chung và sức khỏe của thị trường qua các năm.

```
from pyspark.sql.functions import year
df_with_year = df.withColumn("Year", year(col("Date")))
yearly_trend = df_with_year.groupBy("Year") \
    .agg(avg("<Close>").alias("Avg_Close_VN30")) \
    .orderBy("Year")
yearly_trend.show()
```

```
+----+------------------+
|Year|    Avg_Close_VN30|
+----+------------------+
|2000|1.9333151515151519|
|2001|4.1738960264900635|
|2002|2.2026152542372888|
|2003| 1.162368825910932|
|2004|2.1143787999999986|
|2005| 2.578967729083667|
|2006| 7.513624486571885|
|2007|18.716431409295357|
|2008| 7.057379719917011|
|2009| 9.439350379362669|
|2010|13.432002061855673|
|2011| 16.61505988310308|
|2012|15.707345994643289|
|2013| 18.43317929050814|
|2014|20.844829701657456|
|2015|21.675243309260825|
|2016|25.744144405377444|
|2017| 35.03804804362881|
|2018|  48.0773168487395|
|2019| 47.27430332354514|
+----+------------------+
only showing top 20 rows
```

4.5 Phân tích Giao dịch Đột biến (Window Functions)

Sử dụng Window Functions trong Spark để xác định các sự kiện giao dịch bất thường hoặc quan trọng:

1. **Ngày có khối lượng giao dịch cao nhất:** Tìm ra ngày mà mỗi mã cổ phiếu có khối lượng giao dịch (`<Volume>`) đạt đỉnh, thường liên quan đến các sự kiện đặc biệt hoặc tin tức quan trọng.

2. **Ngày tăng/giảm giá mạnh nhất:** Xác định ngày mà mỗi mã cổ phiếu có phần trăm thay đổi giá trong ngày (`ChangePct`) lớn nhất (cả chiều tăng và chiều giảm).

Những ngày này thường đánh dấu các bước ngoặt hoặc phản ứng mạnh của thị trường đối với thông tin liên quan đến cổ phiếu đó.

```
from pyspark.sql import Window
from pyspark.sql import functions as F
window_up = Window.partitionBy("<Ticker>").orderBy(F.desc("ChangePct"))
max_up = df.withColumn("rank", F.row_number().over(window_up)) \
           .filter(F.col("rank") == 1) \
           .select("<Ticker>", "Date", "ChangePct")

window_down = Window.partitionBy("<Ticker>").orderBy(F.asc("ChangePct"))
max_down = df.withColumn("rank", F.row_number().over(window_down)) \
             .filter(F.col("rank") == 1) \
             .select("<Ticker>", "Date", "ChangePct")

max_up.show(10)
max_down.show(10)

```

```
from pyspark.sql.window import Window

windowSpec = Window.partitionBy("<Ticker>").orderBy(col("<Volume>").desc())


from pyspark.sql.functions import row_number

top_volume_days = df.withColumn("rank", row_number().over(windowSpec)) \
                    .filter(col("rank") == 1) \
                    .select("<Ticker>", "Date", "<Volume>")

top_volume_days.show()
```

5 Trực quan hóa Kết quả

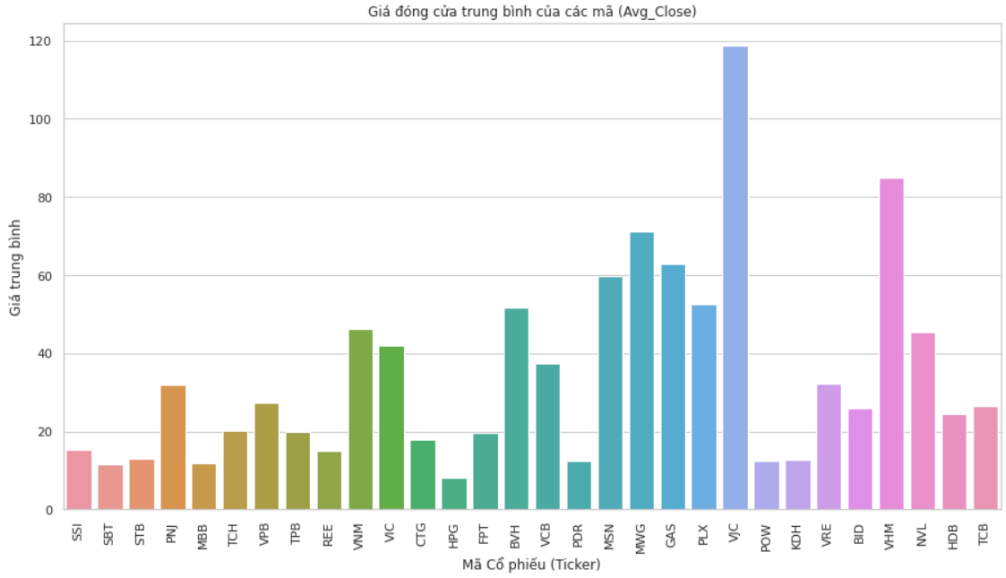
Các kết quả phân tích được trực quan hóa bằng biểu đồ để dễ dàng nắm bắt thông tin:

5.1 Xu hướng giá đóng cửa của một số mã VN30

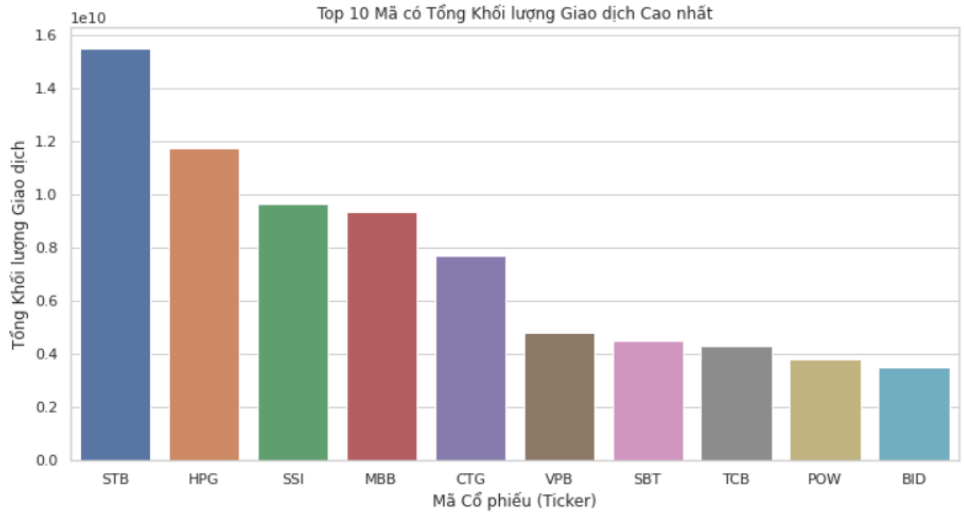
Biểu đồ đường này trực quan hóa xu hướng giá đóng cửa theo thời gian của 4 mã tiêu biểu: VCB, FPT, HPG, và MWG. Có thể thấy rõ sự tăng trưởng mạnh mẽ của MWG và FPT, đặc biệt từ năm 2016 trở đi. VCB cũng cho thấy xu hướng tăng trưởng ổn định. Trong khi đó, HPG có những giai đoạn biến động mạnh hơn, phản ánh tính chu kỳ của ngành thép. Biểu đồ giúp so sánh trực quan hiệu suất đầu tư giữa các mã này trong dài hạn.

5.2 Giá đóng cửa trung bình của các mã (AvgClose)

Biểu đồ cột này cho thấy sự phân hóa rõ rệt về mức giá trung bình giữa các cổ phiếu trong rổ VN30. Các mã như VIC, VHM có mức giá trung bình cao vượt trội so với phần còn lại, trong khi nhóm ngân hàng (ví dụ: STB, TCB) và một số mã khác có mức giá thấp hơn đáng kể. Điều này phản ánh sự khác biệt về giá trị thị trường và có thể là cả tiềm năng tăng trưởng được định giá khác nhau giữa các cổ phiếu.

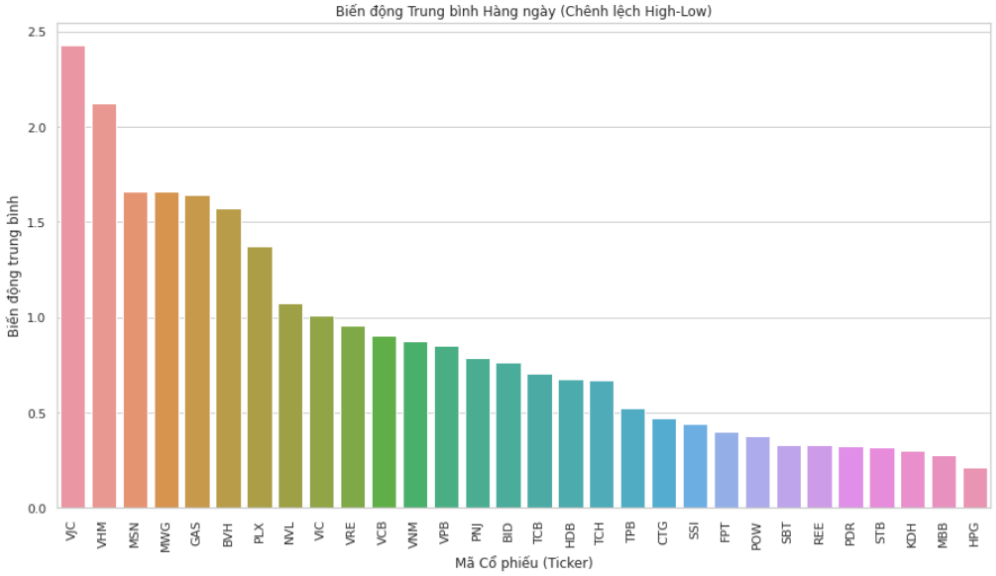


5.3 Top 10 Mã có Tổng Khối lượng Giao dịch Cao nhất



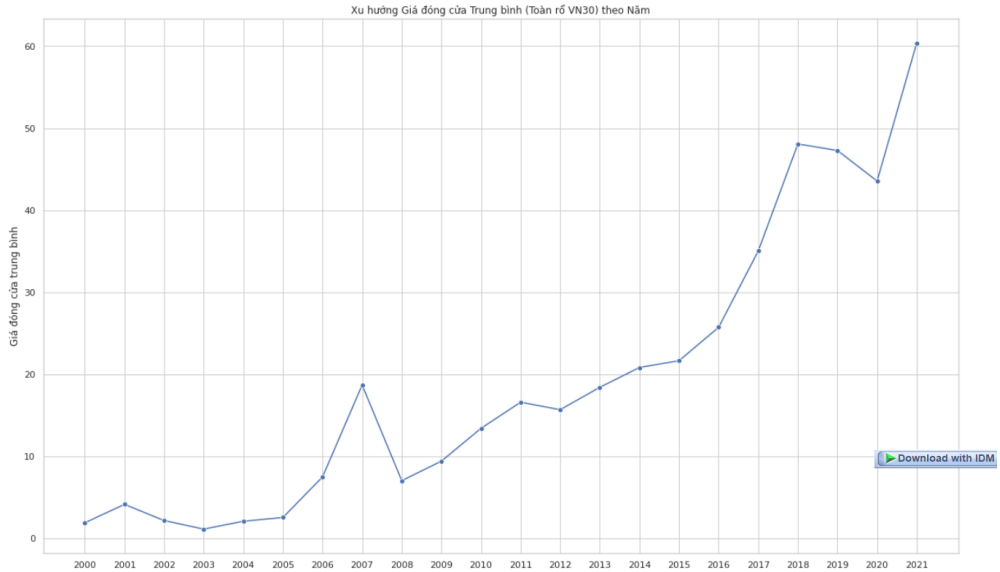
Biểu đồ thể hiện Top 10 cổ phiếu có tính thanh khoản cao nhất, dựa trên tổng khối lượng giao dịch trong suốt giai đoạn phân tích. Mã STB dẫn đầu với khối lượng giao dịch rất lớn, theo sau là HPG và SSI. Đây thường là những cổ phiếu thu hút sự quan tâm lớn của nhà đầu tư, thể hiện qua khối lượng mua bán sôi động.

5.4 Biến động Trung bình Hàng ngày (Chênh lệch High-Low)



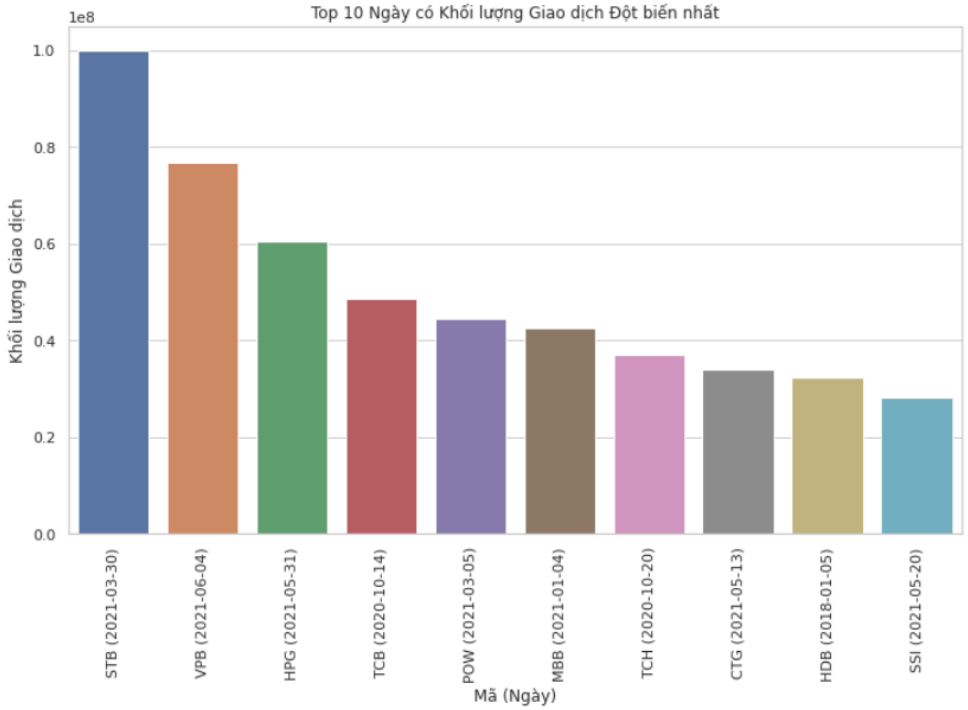
Biểu đồ này so sánh mức độ biến động giá trung bình trong ngày của các cổ phiếu. Mã VJC và VHM cho thấy mức biến động cao nhất, nghĩa là giá có sự chênh lệch lớn giữa mức cao nhất và thấp nhất trong phiên. Những cổ phiếu có biến động cao thường hấp dẫn các nhà đầu tư lướt sóng nhưng cũng đi kèm rủi ro cao hơn. Ngược lại, HPG có mức biến động trung bình thấp nhất trong danh sách.

5.5 Xu hướng Giá đóng cửa Trung bình (Toàn rổ VN30) theo Năm



Đường xu hướng này thể hiện "sức khỏe"chung của thị trường chứng khoán Việt Nam (đại diện bởi rổ VN30) qua các năm. Nhìn chung, thị trường có xu hướng tăng trưởng dài hạn, đặc biệt có sự bứt phá mạnh mẽ từ năm 2016 đến 2018 và giai đoạn 2020-2021. Tuy nhiên, cũng có những năm điều chỉnh giảm như 2008 (khủng hoảng tài chính toàn cầu) và 2018-2019.

5.6 Top 10 Ngày có Khối lượng Giao dịch Đột biến nhất



Biểu đồ cột này chỉ ra 10 ngày giao dịch có khối lượng đột biến nhất đối với các mã cụ thể. Ngày 30/03/2021 mã STB có khối lượng giao dịch cao kỷ lục. Các ngày khác với khối lượng giao dịch lớn của VPB, HPG, TCB,... thường trùng với các sự kiện quan trọng như báo cáo kết quả kinh doanh, chia cổ tức, thông tin vĩ mô hoặc các đợt tái cơ cấu danh mục của quỹ đầu tư lớn.

6 Lưu kết quả (Data Sink)

Sau khi hoàn tất quá trình xử lý và phân tích, DataFrame cuối cùng (đã được làm sạch và bổ sung các cột `ChangePct`, `Daily_Spread`) được lưu trở lại HDFS dưới định dạng CSV để phục vụ cho các phân tích sâu hơn hoặc cho các hệ thống báo cáo khác.

```
df.write.csv("hdfs://namenode:8020/user/jovyan/output/vn30_cleaned_final",
             header=True, mode="overwrite")
```

7 Kết luận

Project đã thành công trong việc mô phỏng và triển khai một hệ thống xử lý dữ liệu lớn đơn giản, ứng dụng vào bài toán thực tế là phân tích dữ liệu lịch sử giá chứng khoán của rổ VN30. Các mục tiêu chính đề ra ban đầu đều đã đạt được:

- **Xây dựng hệ thống:** Đã thiết lập thành công môi trường Big Data mô phỏng sử dụng Docker, bao gồm cụm lưu trữ HDFS (Namenode, Datanode) và cụm xử lý Spark (Master, Worker), cùng với giao diện Jupyter Notebook để tương tác.
- **Vận dụng kỹ thuật xử lý dữ liệu lớn:** Đã thực hiện quy trình ETL (Extract - Transform - Load) hoàn chỉnh, từ việc nạp dữ liệu thô vào HDFS, làm sạch, chuyển đổi kiểu dữ liệu, đến việc tạo ra các thuộc tính mới (`ChangePct`, `Daily_Spread`) bằng PySpark DataFrame API.

- **Trích xuất thông tin giá trị:** Thông qua các phân tích thống kê mô tả, phân tích xu hướng theo thời gian, phân tích biến động và thanh khoản, project đã rút ra được những insight quan trọng về thị trường:
  - Xác định được xu hướng tăng trưởng chung của thị trường qua các năm, cũng như các giai đoạn biến động mạnh.
  - Nhận diện các cổ phiếu có tính thanh khoản cao (ví dụ: STB, HPG, SSI) và các cổ phiếu có mức biến động giá lớn trong ngày (ví dụ: VJC, VHM).
  - So sánh được hiệu suất và đặc điểm biến động giá khác nhau giữa các mã cổ phiếu tiêu biểu thuộc các nhóm ngành khác nhau.
  - Phát hiện các ngày giao dịch có khối lượng đột biến, gợi ý về các sự kiện thị trường đáng chú ý.
- **Lưu trữ kết quả:** Dữ liệu đã qua xử lý được lưu trữ lại trên HDFS, sẵn sàng cho các phân tích sâu hơn.

Việc sử dụng công nghệ Hadoop và Spark đã chứng tỏ hiệu quả trong việc quản lý và xử lý song song khối lượng lớn dữ liệu chuỗi thời gian như dữ liệu chứng khoán. Các phân tích và trực quan hóa kết quả cung cấp cái nhìn tổng quan và chi tiết, hỗ trợ nhà đầu tư trong việc đưa ra quyết định dựa trên dữ liệu.

Hướng phát triển tiếp theo:

- **Phân tích nâng cao:** Áp dụng các mô hình Machine Learning (như ARIMA, LSTM) để dự đoán xu hướng giá cổ phiếu, hoặc các thuật toán phân cụm (Clustering) để nhóm các cổ phiếu có đặc điểm biến động tương tự.
- **Xử lý thời gian thực:** Mở rộng hệ thống bằng cách tích hợp Apache Kafka và Spark Streaming để có thể phân tích dữ liệu giao dịch chứng khoán theo thời gian thực, cung cấp thông tin cập nhật liên tục.
- **Giao diện tương tác:** Xây dựng một dashboard trực quan (ví dụ: sử dụng Dash, Streamlit hoặc tích hợp với Power BI/Tableau) để người dùng cuối có thể tương tác trực tiếp với kết quả phân tích.