



**Social Media Impact on Financial Markets Predictability:
GameStop and r'Wallstreetbets Episode**

**Data Processing Framework
MSBA 305**

Prepared by:

Makram Nouredine
Johnny El Achkar
Tarek Oueidat

Presented to:

Dr. Elie Semaan Nasr

Table of Contents

Introduction and Problem Statement	3
Literature Review	5
Solution & Framework	7
Reddit Wallstreetbets cleaning and feature extraction steps.....	8
Yahoo Finance historical price importing and feature extraction steps.....	9
Yahoo Finance Call/Put Options.....	10
Openinsider Insider trading extraction for GME and BBY	10
Data Analysis.....	11
Contrasting Reddit Bullish Aspects and GME Stock Market Performance	11
Storing Tables in MySQL and retrieving them for querying.....	13
Cassandra.....	19
Reading Cassandra from Python:	23
Conclusion & Recommendations.....	24
References	25

Introduction and Problem Statement

Market valuation and stock pricing in finance is a crucial market driver towards equilibrium between supply and demand of financial products such as companies' stocks. Since the dawn of the financial system, the discipline of investing and market pricing had been governed by what is called "market economic fundamentals." These fundamentals represent the intrinsic KPIs specific to each firm in relation with the overall economic performance throughout business cycles. The latter practice can be described as a standardized stable set of rules that assess the firm's worth. However, the dynamics of the market have changed dramatically and are in a continuous, sometimes exponential, change after factoring in the effects of globalization and the technological rapid evolution. Despite the relative predictability of the trade globalization's effects, the massive and speedy transfer of information across the Internet and throughout innovative social media platforms targeting all levels of the society is significantly disrupting the finance industry. The disruption is originated by integrating unperceived shifts in the investors' Sentiment beyond what is detected in traditional sentiment analysis surveys. Such shifts may lead to incite massive demand financial products and/or massive rejection of others, causing the supply and demand balance move away from tangible fundamentals and requiring market corrections. The issue here is not 'how to conduct market corrections' because in free market exchanges the so called 'invisible hand' of the economy will automatically push towards rational decisions and balance. The problem here is that such moves are so hard to predict using traditional business as usual ways and require "predictive analytics over similar and dissimilar scenarios," therefore finding efficient ways to fetch, store, and prepare Big Data to be analyzed via Machine Learning algorithms.

Why bother doing that when the market is already capable of correcting itself post-events? The answer is simple. Disruptions in market sentiment and eventually the investment decisions have tremendous negative effects on the average investors' savings causing it to be misled and wiped out completely under severe scenarios. In addition to that, we should always consider the Dominos effect of any financial disruption on the overall investing spectrum. The most sensitive and vulnerable part of this spectrum is the pensions' component of the market, which is supposed to be the safety net of the hardworking class of the society to secure their retirement plans. Knowing that, most of the investment funds dealing with sales and trading or investment management are training its staff to be able to collect Big Data, not only in finance but also in politics/military/environment etc., clean it, normalize it, and store it efficiently to retrieve the data and apply statistical methods on it to generate explanatory insights to help cope with any repercussion ahead of time to minimize the downside of crises.

The issue of investment sentiment shifts is not new to the industry; in fact, it was tackled as part of the "Behavior Finance." Based on the CFA institute's definition, "Behavior finance is a field of finance that proposes psychology-based theories

to explain stock market anomalies.” What changed, however, was the proliferation of the means of getting information and its sources, with the technology playing a huge role in making information accessibility more widespread. This brings up the concern about upholding the characteristics integrity, variety, and veracity. Where to look for data, how to process it, and where to store it? It is not as convoluted as it seems. Data availability is increasing significantly due to factors not restricted to the finance industry regulations mandating a periodic disclosure of financial figures of firms, and the massive amount of unstructured data surfing the Internet of Things and social media platforms. As main sources of credible updated financial metrics of corporations, we can cite the live publishing of data on “Yahoo Finance” and the periodic publication of firms’ information on the “Security Stock Exchange (SEC)” official website. These websites provide each an API to access for information request and fetching. Additionally, to access the social media sentiment unstructured data, one can find numerous sources where average and professional investors interact to share their opinion in the market direction: Twitter and Reddit can be the focus here.

Now we need to place emphasis on the main topic or event that brought up the problematic question of this experiment, which is ***“How to find financial market relevant data and prepare it for stock performance trend prediction in time of anomalous market hits and Bubble-like investment schemes?”*** The problematic is the latest market fraud causing massive losses of investors’ savings and financial disruption. It is the GAMESTOP INC stock dilemma. Briefly, what happened was that GAMESTOP (GME), which is a video games publicly traded corporation, was placed under a SHORTING (in other terms SELLING recommendation) Strategy by huge hedge funds due to their research concluding that the firm’s demand and performance is declining considering cannibalization and technological substitution. These prominent hedge funds borrowed large amount money to short sell the GME stock, a strategy called “Leveraged Shorting Strategy.” Under the terms of the Short strategy, these hedge funds will benefit from the decrease in GME’s stock price while being prone to huge losses in case their speculation went off and the stock went up in market value. The up scenario can only be justified by increased demand for the stock, which was not “fundamentally rational.” So, what happened? A Sub-Reddit called Wallstreetbets on Reddit spread the information about the hedge funds strategy and decided that GME is undervalued due to the scheme of hedge funds. Consequently, a huge number of the subreddit subscribers decided to push-to-push the GME price up by buying the stock so that they may profit from the loss of hedge funds. It was called the “Short Squeeze.” Therefore, the stock price hiked insanely pushed by unjustified demand, but the market correction brought it back down causing the majority of late adopter of Wallstreetbets scheme to lose their money. Such a behavior in the financial market is detrimental and need to be forecasted to hedge its effects ahead of time. The only way of understanding what happened and predict similar scenarios is by tackling the Big Data and using it accordingly. This experiment will aim at gathering, cleaning, organizing, and storing the structured and unstructured financial data of GAMESTOP to retrieve later and query the findings, making sense of the real-life results.

Literature Review

Many previous and recent studies addressed the issue of social media influence on market demand movement as well as addressing the GME short squeeze event. We can cite some of the relevant papers as follows, summarizing its findings:

- Predicting Market Volatility Using Semantic Vectors and Google Trends

By: Anusha Balakrishnan and Kalpit Dixit.

The aim of the paper was to find and predict the market volatility considering social media influence. They hypothesized that social media is capable of accurately reflecting market sentiment and therefore its data can be used to predict market movement. To do so they needed to access two APIs to gather data. The first step is querying the New York Times (NYT) Article Search API for time series financial data relative the Futures of S&P500 Index traded at the Chicago Mercantile Exchange (CME). Next, they needed to fetch, clean, normalize, and apply NLTK Python library to extract insights; therefore, they accessed Google Trends API to extract the important financial keywords trending to be used in the experiment. The yielded result was a corpus of snippets from 20,000 articles since 2012 (weekly Search Volume Index SVI), which is then used to generate word2vec. After successfully processing and testing the data the results confirmed the ability of predicting, the market trend based on social media interactions.

- News and social media emotions in the commodity market

By: Jiancheng Shen, Mohammad Najand, Feng Dong, and Wu He.

The paper fall under the topic of Behavioral Finance and its purpose is to investigate the media-based sentiments role in predicting future commodity returns. To confirm the hypothesis, the authors decided to check the influence of media interaction on the following five days' commodity returns and used a proprietary data of commodity-specific market emotions stored from textual media sources such as newswires, Internet news and social media. They compared the textual data analysis to the time series daily observations of the CRB commodity market index, crude oil and gold returns using econometrics models such as the Auto-Regressive (AR) model on 14 years of historical data to contrast the inputs. The market sentiments considered were optimism, fear, and joy showing its reflection on the consecutive returns. The finding was that media emotions and sentiment have significant influence over the commodity returns in question.

- Securities regulation and class warfare

By: Jonathan R. Macey

A research pivoting around market regulation and market efficiency considering the disruption caused by the GameStop Inc. trading bubble through the Robinhood Financial LLC, the free trading easily accessible platform. Since the main definition of market efficiency is that, the prices in the market are based on fundamental

values and other variables that are already factored in the price or correcting the price systematically, they found that GME trade led to unfair results and misleading information causing disadvantage of the average investors to the benefit of the market financial elite. This finding was further reinforced by the research outcomes. The results of the analysis also pointed fingers at the equal responsibility of retail traders, who originally publicized their opinion of the GME underperformance, and the bloggers on Reddit Wallstreetbets who fueled the wave of deceiving trades pushing the GME price to an all-time high followed by a huge crash due to the demand volatility.

- Squeezing Shorts Through Social News Platforms

By: Franklin Allen, Eric Nowak, Matteo Pirovano, and Angel Tengulov.

The paper tackles the issue of short squeezes such as the one conducted on the short positions of GME stock. It argues that the results are a product of coordinated trading by investors discussing their strategies over social media platforms, such as Reddit in our experiment's case. They even expanded the scope of their research towards gathering option markets data as playing a central role in the short squeeze events. The data was a unique primary data extracted by monitoring media news and activities over the social media platforms Reddit, Twitter, Stocktwits, as well as extensive public press searches. It was tested to show that such social media events contributed to the impeding of the market quality. This information was matched against accounting and stock price information from Compustat as well as the annual reports of the financial product under the scope. The data sample consisted of 13 stocks experiencing trading ban because of the short squeeze during January and February 2021; therefore, GME is key here. To further test the variation in market quality and differentiate between pre- and post-Squeeze, the researchers adopted a linear regression model. The observed results were that the bid-ask spreads indicating risk or liquidity increased by 35% during the short squeeze incited by social media, and the volatility increase by 129% with, of course, a hike in the trading volume during the event by 121%. It later dropped substantially after the event. Consequently, the authors validated the evidence of inter-linkages between the social media platforms, derivatives market, and equity markets.

- Sentiment Analysis of Events from Twitter Using Open-Source Tool

By: Rajkumar S. Jagdale, Vishal S. Shirsat, and Sachin N. Deshmukh.

This paper addressed the sentiment analysis specifically as social media feeds through online forum' discussions, blogs, micro-blogs, Twitter, and social networks that was increasing in availability exponentially. They show that data collected from these sources provide content about market demand's trends. For this sentiment analysis and opinion mining research, the paper used Natural Language processing and R programming as a statistical open-source tool. Data was collected through Twitter's API using R, cleaning the data, preprocessing it, and labeling it according to the emotional state combined with each set of tokens such as "Happy", "Sad", "Angry", "Fear", or "Surprise" and they created word clouds for visualization. They

also calculated the numbers of positive, negative, and neutral tweets from each event. After reviewing results, they found correlation between both social media sentiment and market event, which makes sentiment analysis a good reference for consumer product decision making, for instance for e-commerce websites.

Solution & Framework

In order to figure out the impact of the Reddit's Wallstreetbets on (GME) GameStop's market price itself, we need to acquire a meaningful data capable of providing an insight into both GME's financial historical data as well as the output of the subreddit tokens' analysis during the period of abnormal volatility. Additionally, we can conduct a pre- and post-event analysis combined by an insider look into the firm. Therefore, we need to use python libraries to retrieve and request data from the public APIs of the following websites:

- reddit.com/r/wallstreetbets/
- finance.yahoo.com/quote/GME/history?p=GME
- finance.yahoo.com/quote/GME/options?p=GME
- openinsider.com/screener?s=GME
- openinsider.com/screener?s=BBY

We thereby access these websites using two methods, which are using web scraping for the insider trade tables and traditional use of provided web-APIs for the Reddit's comments collection and the Yahoo Finance's GME historical stock price as well as the GME's historical Call and Put options. Note that the call and put tables will be used for post-fluctuation analysis to contrast it with the previous weeks.

In terms of data structure, it is worth noting that tables imported from Yahoo Finance were somewhat structured, whereas the data imported from Reddit and Open insider needed restructuring to apply further operations on it. We elected to add an insider-trading table of GameStop GME's competitor Best-Buy BBY to make sense of the insider trade in the video games industry.

```
import nbconvert
nbconvert.NotebookExporter.output_mimetype = 'application/x-ipyndb+json'

gamestop = web.DataReader('GME', 'yahoo', dt.datetime(2020,12,31), dt.datetime(2021,2,22))

url = r"http://openinsider.com/screener?s=gme&o=&pl:"
gme_insider = pd.read_html(url)

df = pd.read_html('https://finance.yahoo.com/quote/GME/options')

print(df[0])
```

```

import praw
import pandas as pd
import datetime as dt
from praw import PushshiftAPI

pd.set_option('max_colwidth', 500)
pd.set_option('max_columns', 50)

r = praw.Reddit(client_id='@000', \
                client_secret='@0000000', \
                user_agent='MSBA305', \
                username='@0000', \
                password='@0000')

api = PushshiftAPI()

start_epoch=int(dt.datetime(2021, 1, 1).timestamp())
end_epoch=int(dt.datetime(2021, 1,31).timestamp())

api_request_generator = api.search_submissions(subreddit='wallstreetbets',after = start_epoch, before=end_epoch)

aita_submissions = pd.DataFrame([submission.d_ for submission in api_request_generator])

```

Next step is cleaning the tabularized data in each of the six acquired tables. However, the most important cleaning and normalization is the one addressing the Reddit comments since it involves pattern detection and feature extraction. We will go over the cleaning process if each of the tables until we reached the final versions intended to be stored in our database.

Reddit Wallstreetbets cleaning and feature extraction steps

1. We accessed the Reddit “Pushshift” API after creating a Reddit application using a registered client id and client secret.
2. Due to the heavy load of data in the targeted subreddit, we extracted posts over two stages, one stage per month covering the period of January 1st till February 28th.
3. After having the imported raw data in Pandas Data Frame, we shortlisted the feature columns to cover the list of: ['id', 'url', 'date', 'title', 'created', 'score', 'num_comments'].

Table 1 Reddit comments

```
reddit.tail(10)
```

	title	score	id	num_comments	date	url
863486	Going Full Retard	1	lu3poe	0	2021-02-28	https://www.reddit.com/r/wallstreetbets/commen...
863487	Great research produces great results.	1	lu3pmn	0	2021-02-28	https://www.reddit.com/r/wallstreetbets/commen...
863488	To the moon!	1	lu3pja	0	2021-02-28	https://i.redd.it/vriggcjav4k61.jpg
863489	GOOOOOOOOOD MORNINGGGG GMEEEEEEEEE	2	lu3p90	0	2021-02-28	https://v.redd.it/3bhmyueku4k61
863490	A Reminder: Worst Case Scenario with AMC is a ...	19	lu3p87	1	2021-02-28	https://www.reddit.com/r/wallstreetbets/commen...
863491	Doing my small part by passing this along 💎	2	lu3p5u	0	2021-02-28	https://i.redd.it/tas5usm6v4k61.jpg
863492	🦊🦊 APES STIMULUS - LOCKED AND LOADED 🦊🦊	25308	lu3p20	404	2021-02-28	https://v.redd.it/fjnt1ghvu4k61
863493	STAY STRONG APES THIS WEEK IS GONNA BE A BUMPY...	1	lu3ors	0	2021-02-28	https://v.redd.it/xqtbbc12v4k61
863494	Affordable guide to trading. From A-Z	1	lu3oec	0	2021-02-28	https://beginnertradingbook.com
863495	Fun and Games Switching From Margin to Cash on ET	60	lu7g5l	21	2021-02-28	https://www.reddit.com/r/wallstreetbets/commen...

4. We adjusted the data types to represent the datetime64 for the timestamp and object for the titles, keeping the type integer for the num_comments column.

5. We dropped the NAs because we could afford it, having a large load of representative data. We ended up having 863495 records.
6. Applied a loop function to remove punctuations.
7. We lowercased the titles row-by-row through a lambda function followed by a word tokenization splitting the rows into a list of tokens.
8. We inspected the unique values in the remaining data to extract a list of Bullish Expressions, which is a list of words used by subscriber to hint at upside expectations of the stock price. This list will serve as an indication of buying prospects and positive investors' sentiment.

```
bull_lst = ['gme', '🚀', '🚀🚀', '🚀🚀🚀', 'yolo', 'gme', '💎', '$', 'rkt', 'rocket', 'gamestop']
```

9. Feature extraction: Once we figured out the most frequent bullish terms, we apply a bullish terminology count row-by-row via lambda function and storing the result in a new counter' column.
10. Filtered the outstanding rows accordingly and grouped the observations by date to have a unique count number per date. Since it is a timeseries case, we elected to keep these observations unique by date, which in turns will be our primary key.
11. The final version of the table of 59 observations covering the prespecified period as follows:

Table 2 Reddit Final table

	Date	frequency_bullish	length	num_comments	score
0	2021-01-01	101	1151	2300	101
1	2021-01-02	111	1181	2420	110
2	2021-01-03	77	724	1900	77
3	2021-01-04	176	1761	2935	176
4	2021-01-05	153	1614	4339	153

Yahoo Finance historical price importing and feature extraction steps

1. We accessed the Yahoo Finance using its “web.DataReader” API and specifying the range of daily prices between January 1st and February 28th.



Figure 1 Candle-Chart GME Price

2. The data is straightforward, however we shortlisted the column features to cover only the date as a unique index, trade volume, and adjusted closing price per day.

Table 3 Yahoo Finance Historical Price

	High	Low	Open	Close	Volume	Adj Close
Date						
2020-12-31	19.799999	18.799999	19.250000	18.840000	6922700	18.840000
2021-01-04	19.100000	17.150000	19.000000	17.250000	10022500	17.250000
2021-01-05	18.080000	17.230000	17.350000	17.370001	4961500	17.370001
2021-01-06	18.980000	17.330000	17.340000	18.360001	6056200	18.360001
2021-01-07	19.450001	18.020000	18.469999	18.080000	6129300	18.080000

3. We then applied the percentage change function to extract the daily stock return and stored it in a new column.

Yahoo Finance Call/Put Options

1. We scraped the webpage containing both tables of Call options, Put options, and stored them into two separate Data Frames. Note here that unlike the historical price, options have multiple trading records per day, making the date column subject to redundancy.
2. We removed the irrelevant columns. The remaining columns we needed are 'Last_Trade_Date', 'Strike', and 'Implied_Volatility.'
3. We adjusted the format of the date into YY-MM-DD, and changed the volatility into float64 after stripping the comma from the number.
4. We then grouped the observations by date and applied the Median function to represent each daily value in the table.

Hint: what we need from these two tables is to assess the post-fluctuation behavior of the market speculators since the higher the expectations of price hike will result in more volume of calls to the expense of the puts. We also can observe the volatility change and the strike price, which is the exercise price in case the investor wanted to use the option whether to buy the underlying stock (in the case of Calls), or sell the underlying stock (in case of Puts). Such comparison will provide us with more GME investors' insight.

Open insider trading extraction for GME and BBY

Before laying down the extraction and data treatment steps, here is a brief explanation about what insider trading means:

“Trading can both be **legal** and illegal. Illegal **insider trading** is when the **insiders** want to benefit from the company information at the cost of the

company. **Legal insider trading** is when the **insiders** of the company trade shares but, at the same time, report the **trade** to the Securities and Exchanges Commission (SEC)."¹

1. We accessed the Open insider records of both GME and BBY separately for data covering 1 year of insider trading filings.
2. We fetched the data through scraping using "pd.read_html()."
3. The tables' format was frozen rendering the data inaccessible for manipulation. Therefore, we stored the acquired table as .csv format then imported it to clean it.
4. We renamed the columns, dropped the unwanted ones, adjusted the "Trade_Date" format to "YY-MM-DD," stripped the "\$" from the price and the value columns, and finally concatenated the GME and BBY observations to be enclosed in one final table.

Hint: this table is meant to be considered for a stand-alone analysis.

Data Analysis

Contrasting Reddit Bullish Aspects and GME Stock Market Performance

What we hypothesize here is that Reddit's virtual investors are enticing a high market volatility causing the GME price to fluctuate abnormally by encouraging each other to go Long or Short the stock. This issue should be revealed or at least promoted through data visualization of the price and return trends comparing them to the bullish term frequency on Reddit.



Figure 2 GME daily return

Based on the plot of the stock return above, which is usually mean reverting, the period between January 25th and February 1st stands out as not being covariant stationary. Such behavior may indicate an abnormal event forcing the stock price to

¹ www.wallstreetmojo.com/insider-trading

move up and down based on factors other than its underlying firm's fundamental facts. We further inspect the price trend compared to the Reddit bullish trend looking for any correlation or causality prospects.



Figure 3 GME daily price

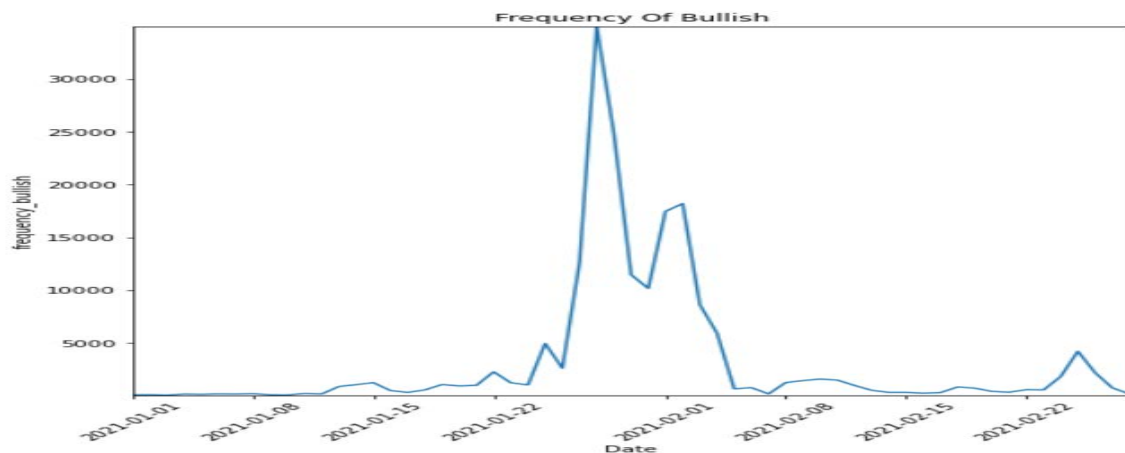


Figure 4 Reddit bullish frequency plot

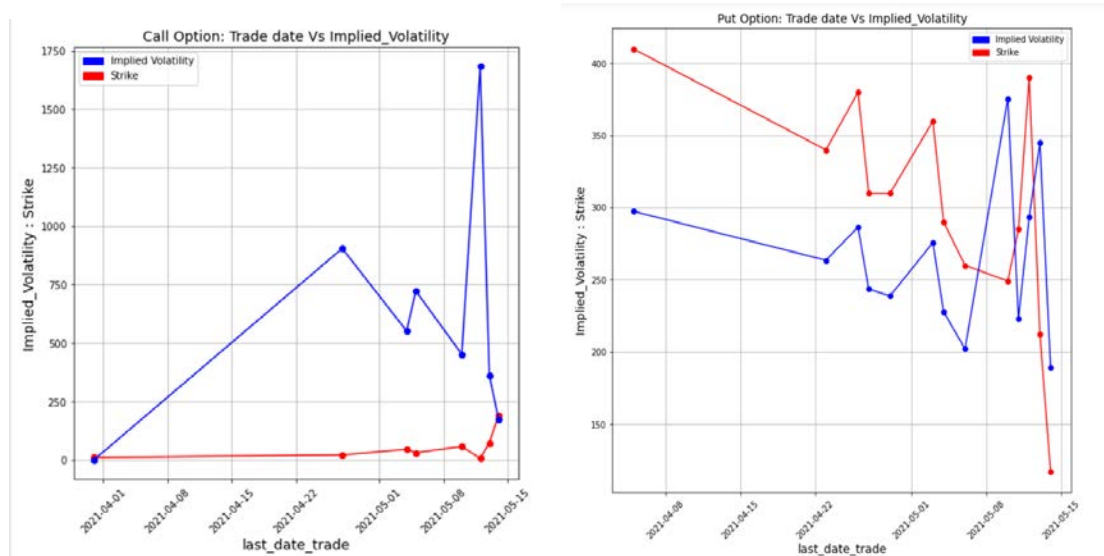
When we compare the above plots representing GME's stock price and Reddit's frequency of bullish terminology employment, we see significant association between the social media bullish trend and the bullishness of the stock itself. This shall validate the hypothesis previously proposed and can prove to a certain extent that social media platforms, such as Reddit, are contemporary influencers in the financial market.

It also highlights a very important aspect, which is the inconsistency of such influence and the unsustainability of an impulsive unfounded investing strategy. We can see it clear by observing the free fall in price level following the peak.

We can see a second wave of bullish terminologies following the price crash. This wave was also based solely on people's investment sentiment and it resulted in a

desperate try to hold the price up, however it gave it a short-term support before it continued to drop even more.

Furthermore, to investigate the post-event speculative behavior to see whether the regular investors tend to shift their expectations and lead to a market correction, we plot the median strike price and volatility for the Call and Put options.



The pair of figures above show a continuation in the market's speculation, expecting a new wave of price hike and therefore call option' strike price stabilized with a slight upward slope but with a volatility reaching unexpected levels towards mid-May. On the other hand put option' price kept falling then fluctuated while exhibiting high volatility variations towards the same period in May. We can see that even though the social media's impact on the stock market lacks fundamental foundations and is unsustainable, its effect post-crash is critical and stands in the way of rapid market corrections.

Hint: The insider trading is to be tackled for insights in the database queries.

Storing Tables in MySQL and retrieving them for querying

After cleaning and preparing the tables relevant to the GME-Reddit event, we created a new database 'REDDIT' and stored all of the tables except the insider-trading table into this database. The latter was kept for making use of redundant trade date observations in the Cassandra Keyspace. Therefore, we specified the unique date values to be the Primary Key.

First, we started our process by creating and establishing a connection between our Python notebook and MySQL database. The steps are as follow:

- Create a server connection (host name, username and password) and connect it to our localhost.

```
import mysql.connector
from mysql.connector import Error
```

```
def create_server_connection(host_name, user_name, user_password):
    connection = None
    try:
        connection = mysql.connector.connect(
            host=host_name,
            user=user_name,
            passwd=user_password
        )
        print("MySQL database connection successful")
    except Error as err:
        print(f"Error: '{err}'")

    return connection
```

```
connection = create_server_connection("localhost", "root", "@@@@")
```

MySQL database connection successful

- Create a database function (connection and query) and establish our 'REDDIT' database. We can see in our MySQL that Reddit's database is now available.

```
def create_database(connection, query):
    cursor = connection.cursor()
    try:
        cursor.execute(query)
        print("Database created successfully")
    except Error as err:
        print(f"Error: '{err}'")
```

```
create_database_query = "CREATE DATABASE REDDIT"
create_database(connection, create_database_query)
```

Error: '1007 (HY000): Can't create database 'reddit'; database exists'

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| reddit |
| sakila |
| sys |
| world |
+-----+
7 rows in set (0.00 sec)
```

- Create a database connection (host name, username, password and database name)

```
def create_db_connection(host_name, user_name, user_password, db_name):
    connection = None
    try:
        connection = mysql.connector.connect(
            host=host_name,
            user=user_name,
            passwd=user_password,
            database=db_name
        )
        print("MySQL Database connection successful")
    except Error as err:
        print(f"Error: '{err}'")

    return connection
```

- Execute a connection query between Python and MySQL

```
def execute_query(connection, query):
    cursor = connection.cursor()
    try:
        cursor.execute(query)
        connection.commit()
        print("Query successful")
    except Error as err:
        print(f"Error: '{err}'")
```

After creating our connections, we imported from the 'sqlalchemy' library the 'create_engine' function in order to create a connection that stores our cleaned Python dataframes directly into MySQL created database 'reddit'. In addition to that we specified using the created engine query the primary key of each of stored tables directly from Python without the need to set it manually from MySQL.

```
from sqlalchemy import create_engine

# Credentials to database connection
hostname="localhost"
dbname="REDDIT"
uname="root"
pwd="@@@@"

# Create SQLAlchemy engine to connect to MySQL Database
engine = create_engine("mysql+mysqlconnector://{user}:{pw}@{host}/{db}".format(host=hostname, db=dbname, user=uname, pw=pwd))
```

The list of created tables in MySQL is as follow:

- Convert cleaned Reddit dataframe to MySQL table with Date as primary key

Convert cleaned Reddit dataframe to sql table

```
# Convert cleaned Reddit to sql table
Reddit_clean.to_sql('reddit', engine, index=True, if_exists='replace')
```

```
with engine.connect() as engine:
    engine.execute('ALTER TABLE `reddit` ADD PRIMARY KEY (`Date`);')
```

```
mysql> describe reddit;
+-----+-----+-----+-----+-----+-----+
| Field          | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Date           | datetime  | NO   | PRI | NULL    |       |
| frequency_bullish | bigint    | YES  |     | NULL    |       |
| length         | bigint    | YES  |     | NULL    |       |
| num_comments   | bigint    | YES  |     | NULL    |       |
| score          | bigint    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select * from reddit;
+-----+-----+-----+-----+-----+
| Date           | frequency_bullish | length | num_comments | score |
+-----+-----+-----+-----+-----+
| 2021-01-01 00:00:00 | 101 | 1151 | 2300 | 101 |
| 2021-01-02 00:00:00 | 111 | 1181 | 2420 | 110 |
| 2021-01-03 00:00:00 | 77  | 724  | 1900 | 77  |
| 2021-01-04 00:00:00 | 176 | 1761 | 2935 | 176 |
| 2021-01-05 00:00:00 | 153 | 1614 | 4339 | 153 |
| 2021-01-06 00:00:00 | 191 | 1946 | 2583 | 191 |
| 2021-01-07 00:00:00 | 179 | 2028 | 4537 | 179 |
| 2021-01-08 00:00:00 | 214 | 2208 | 2927 | 214 |
+-----+-----+-----+-----+-----+
```

- Convert cleaned GME Yahoo Stock Prices dataframe to MySQL table with Date as primary key

Convert cleaned GME Yahoo Stock Prices dataframe to MySQL table

```
# Convert GME stock prices to sql table
gme.to_sql('gme', engine, index=True, if_exists='replace')

with engine.connect() as engine:
    engine.execute('ALTER TABLE `gme` ADD PRIMARY KEY (`Date`);')
```

- Convert Combined REDDIT & GME stock prices to MySQL table with Date as primary key

Convert Combined REDDIT & GME stock prices to MySQL table

```
# Convert Combined REDDIT & GME stock prices to sql table
reddit_gme.to_sql('reddit_gme', engine, index=True, if_exists='replace')

with engine.connect() as engine:
    engine.execute('ALTER TABLE `reddit_gme` ADD PRIMARY KEY (`Date`);')
```

- Convert the Puts data frame to MySQL table with 'Last_Trade_Date' as primary key

Convert the Puts dataframe to MySQL table

```
# Convert Puts table to sql table
puts1.to_sql('puts', engine, index=True, if_exists='replace')

with engine.connect() as engine:
    engine.execute('ALTER TABLE `puts` ADD PRIMARY KEY (`Last_Trade_Date`);')
```

- Convert the Calls data frame to MySQL table with 'Last_Trade_Date' as primary key

Convert the Calls dataframe to MySQL table

```
# Convert Calls table to sql table
calls1.to_sql('calls', engine, index=True, if_exists='replace')

with engine.connect() as engine:
    engine.execute('ALTER TABLE `calls` ADD PRIMARY KEY (`Last_Trade_Date`);')
```

- We can clearly see on MySQL that all tables were created successfully

```
mysql> show tables;
+-----+
| Tables_in_reddit |
+-----+
| calls             |
| gme               |
| puts             |
| reddit            |
| reddit_gme        |
+-----+
5 rows in set (0.00 sec)
```


Queries Examples:

In order to test if we our created MySQL tables are ready for usage, we will perform a couple of queries that will retrieve us the needed information.

- Query 1:

In this second query we will retrieve from reddit table, the index, date, number of comments and frequency of bullish between the dates 27/01/2021 and 31/01/2021

```
from sqlalchemy import create_engine
import pandas as pd

db_connection = create_engine("mysql+mysqlconnector://{user}:{pw}@{host}/{db}".format(host=hostname,
                                                                                       db=dbname, user=username, pw=pwd))

df2 = pd.read_sql('SELECT Date,frequency_bullish, num_comments from reddit where (Date > "2021-01-27" AND Date< "2021-01-31")',
                  con=db_connection,index_col='Date')
```

As we can see, we successfully managed to retrieve the data we required

	frequency_bullish	num_comments
Date		
2021-01-28	34975	534346
2021-01-29	24799	769783
2021-01-30	11467	443545

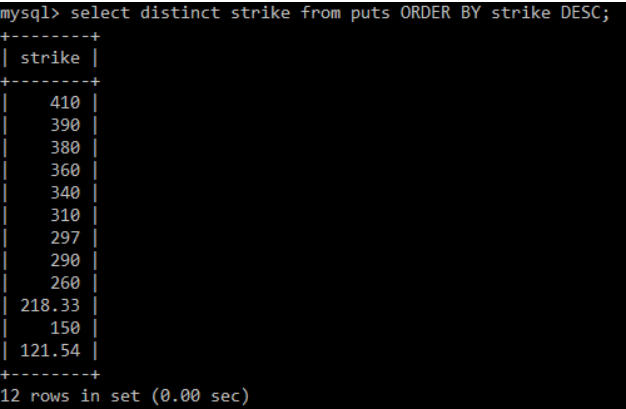
- Query 2:

Select the distinct strike values from the puts table and order the output in descending order

```
db_connection = create_engine("mysql+mysqlconnector://{user}:{pw}@{host}/{db}".format(host=hostname,
                                                                                       db=dbname, user=username, pw=pwd))

df3= pd.read_sql('SELECT distinct strike from puts ORDER BY strike DESC', con=db_connection)
```

	strike
0	410.00
1	390.00
2	380.00
3	360.00
4	340.00
5	310.00
6	297.00
7	290.00
8	260.00
9	218.33
10	150.00
11	121.54



Comparing the values obtained from MySQL directly and from the Python query, we can see that both functions gave us the same results.

- Query 3:

Retrieve the date of the maximum adjusted closing price achieved by the GME stock

```
db_connection = create_engine("mysql+mysqlconnector://{user}:{pw}@{host}/{db}".format(host=hostname,
                                                                                       db=dbname, user=username, pw=pwd))

df4= pd.read_sql('select Date,Adj_close from gme WHERE Adj_close=(Select MAX(Adj_close) from gme);', con=db_connection)
```

	Date	Adj_close
0	2021-01-27	347.51001

```
mysql> select Date,Adj_close from gme WHERE Adj_close=(Select MAX(Adj_close) from gme);
+-----+-----+
| Date | Adj_close |
+-----+-----+
| 2021-01-27 00:00:00 | 347.510009765625 |
+-----+-----+
1 row in set (0.00 sec)
```

As we can see, the returned maximum Closing Adjusted Price of 347.5\$ tallies 100% with our Data, as on 27/01/2021 GME Stock reached its historical maximum Adjusted Closing Price.

- Query 4:

Joining the reddit cleaned table and the GME table using the Date as the common attribute

```
db_connection = create_engine("mysql+mysqlconnector://{user}:{pw}@{host}/{db}".format(host=hostname,
                                                                                       db=dbname, user=username, pw=pwd))

df5= pd.read_sql('select * from reddit.reddit, reddit.gme WHERE reddit.Date=gme.Date;', con=db_connection)
```

	Date	frequency_bullish	length	num_comments	score	Date	Volume	Adj_Close	return
0	2021-01-04	176	1761	2935	176	2021-01-04	10022500	17.250000	-0.084395
1	2021-01-05	153	1614	4339	153	2021-01-05	4961500	17.370001	0.006957
2	2021-01-06	191	1946	2583	361	2021-01-06	6056200	18.360001	0.056995
3	2021-01-07	179	2028	4537	421	2021-01-07	6129300	18.080000	-0.015251
4	2021-01-08	214	2208	2927	362	2021-01-08	6482000	17.690001	-0.021571
5	2021-01-11	235	2375	6245	675	2021-01-11	14908000	19.940001	0.127190
6	2021-01-12	186	2220	4008	510	2021-01-12	7060700	19.950001	0.000502

```
mysql> select * from reddit.reddit, reddit.gme WHERE reddit.Date=gme.Date;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Date | frequency_bullish | length | num_comments | score | Date | Volume | Adj_Close | return |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2021-01-04 00:00:00 | 176 | 1761 | 2935 | 176 | 2021-01-04 00:00:00 | 10022500 | 17.25 | -0.084394911874217 |
| 2021-01-05 00:00:00 | 153 | 1614 | 4339 | 153 | 2021-01-05 00:00:00 | 4961500 | 17.3700008392334 | 0.006956570390342032 |
| 2021-01-06 00:00:00 | 191 | 1946 | 2583 | 361 | 2021-01-06 00:00:00 | 6056200 | 18.360000610351562 | 0.056994802722292626 |
| 2021-01-07 00:00:00 | 179 | 2028 | 4537 | 421 | 2021-01-07 00:00:00 | 6129300 | 18.079999923706055 | -0.015250581554318687 |
| 2021-01-08 00:00:00 | 214 | 2208 | 2927 | 362 | 2021-01-08 00:00:00 | 6482000 | 17.690000534057617 | -0.021570762792818332 |
| 2021-01-11 00:00:00 | 235 | 2375 | 6245 | 675 | 2021-01-11 00:00:00 | 14908000 | 19.940000534057617 | 0.12719049926924497 |
| 2021-01-12 00:00:00 | 186 | 2220 | 4008 | 510 | 2021-01-12 00:00:00 | 7060700 | 19.950000762939453 | 0.0005015159786356804 |
```

We can see that our join was successful and it yielded the same results we did in our notebook, where we used the merge function in order to combine the reddit cleaned table and the GME table on the attribute date

Cassandra

After cleaning and preparing the tables relevant to the GME-Reddit event using Python, we extracted the tables into external CSV files in order to store and query over them using CASSANDRA. The procedure adopted was as follow:

- Create the KeySpace 'Reddit' using the SimpleStrategy method and a replication factor of 3:

```
cqlsh> create KEYSPACE Reddit with  
replication={'class':'SimpleStrategy','replication_factor':3};
```

- Use KeySpace 'Reddit'

```
cqlsh> use reddit;
```

After creating the Keyspace reddit and selecting it, we will now store some of the tables we extracted from Python, store them into Cassandra using the 'COPY function from CSV', and perform some queries over it:

1. Reddit Table:

- Create the reddit table, with "Date" as partition key and "frequency_bullish" as clustering key:

```
cqlsh:reddit> create table reddit(Date date, frequency_bullish int, length  
int, num_comments int, score int, PRIMARY  
KEY((Date),frequency_bullish));
```

```
cqlsh:reddit> select * from reddit;
```

```
cqlsh:reddit> COPY reddit(date,frequency_bullish,length, num_comments,  
score) FROM 'C:/Users/Makram/Desktop/305/reddit_clean.csv' WITH  
DELIMITER =',' AND HEADER=TRUE;
```

```
cqlsh:reddit> select * from reddit;
```

```
cqlsh:reddit> select * from reddit;
```

date	frequency_bullish	length	num_comments	score
2021-01-16	504	5841	12666	1231
2021-01-10	87	990	3738	160
2021-02-24	1827	18629	221066	280706
2021-02-28	188	2423	3584	44779
2021-02-26	2199	26957	282223	295894
2021-02-12	988	11512	20429	25990
2021-02-17	312	3968	94172	127242
2021-01-28	34975	444812	534346	214955
2021-02-05	671	8648	20212	137282

- Queries:

- Extract all the dates with a frequency of bullish higher than 10,000

```
cqlsh:reddit> SELECT date,frequency_bullish FROM reddit WHERE frequency_bullish > 10000 ALLOW FILTERING;
```

date	frequency_bullish	num_comments
2021-01-28 22:00:00+0000	24799	769783
2021-02-01 22:00:00+0000	18195	495273
2021-01-31 22:00:00+0000	17458	809892
2021-01-29 22:00:00+0000	11467	443545
2021-01-30 22:00:00+0000	10166	294336
2021-01-27 22:00:00+0000	34975	534346
2021-01-26 22:00:00+0000	12631	450177

We can see that during the days between 26/01/2021 and 01/02/2021, the activity on reddit was the highest ever.

- Update the row on 24/01/2021 to have a new num_comments = 100000 and score of 45000.

```
cqlsh:reddit> update reddit SET num_comments=1000000, score=45000 WHERE date = '2021-01-24' and frequency_bullish=1037;
```

```
cqlsh:reddit> select * from reddit Where date='2021-01-24';
```

date	frequency_bullish	length	num_comments	score
2021-01-24	1037	12324	4489	1042

date	frequency_bullish	length	num_comments	score
2021-01-24	1037	12324	1000000	45000

2. GME STOCK prices Table:

- Create the GME table, with “Date” as partition key and “Volume” as clustering key:

```
cqlsh:reddit> create table GME(Date date, Volume int, Adj_Close float, return float, PRIMARY KEY((Date),Volume));
```

```
cqlsh:reddit> COPY gme(Date,Volume,Adj_Close,return) FROM 'C:/Users/Makram/Desktop/305/gme.csv' WITH DELIMITER =',' AND HEADER=TRUE;
```

```
cqlsh:reddit> select * from gme;
```

```
cqlsh:reddit> select * from gme;
```

date	volume	adj_close	return
2021-02-24	83111700	91.71	1.03936
2021-02-26	91963000	101.74	-0.064288
2021-02-12	14573300	52.4	0.02544
2021-02-17	9186800	45.94	-0.072107
2021-01-28	58815800	193.60001	-0.442894
2021-02-05	81345000	63.77	0.191963
2021-01-15	46866400	35.5	-0.110499
2021-01-20	33471800	39.12	-0.006098
2021-02-09	26843100	50.31	-0.1615
2021-01-26	178588000	147.98	0.927074
2021-01-06	6056200	18.36	0.056995

- Queries:

- Extract all the data where the trading volume was higher than 100,000,000

```
cqlsh:reddit> select * from gme where volume > 100000000 Allow filtering;
```

date	volume	adj_close	return
2021-01-26	178588000	147.98	0.927074
2021-01-22	197157900	65.01	0.510807
2021-01-25	177874000	76.79	0.181203
2021-02-25	150308800	108.73	0.185585
2021-01-13	144501700	31.4	0.573935

- We will apply a batch where we add a new row to the table with values ('2021-05-14',150000000,159.52,-0.5), update the return to 0.8 instead of -0.5, and then delete the created row

```
cqlsh:reddit> begin batch
... insert into gme(date,volume,adj_close,return) values('2021-05-14',150000000,159.52,-0.5);
... update gme set return=0.8 where date='2021-05-14'and volume=150000000;
... delete from gme where date='2021-05-14';
... apply batch;
cqlsh:reddit> select * from gme;
```

```
cqlsh:reddit> select * from gme Where date='2021-05-14';

date | volume | adj_close | return
-----+-----+-----+-----
(0 rows)
cqlsh:reddit>
```

We can see that our newly created row is no longer available.

3. Insider Trading Table:

- Create the Insider Trading table, with “ID” as partition key and “ticker” as clustering key:

```
cqlsh:reddit> create table insider(Trade_date date, Ticker varchar,  
Insider_Name varchar, Trade_Type varchar, Price float, Qty int, Value  
float, id int, PRIMARY KEY((id), Ticker));
```

```
cqlsh:reddit> COPY  
insider(trade_date,ticker,insider_name,trade_type,price,qty,value,id)  
FROM 'C:/Users/Makram/Desktop/insider_trade_test.csv' WITH  
DELIMITER =',' AND HEADER=TRUE;
```

```
cqlsh:reddit> select * from insider;
```

id	ticker	insider_name	price	qty	trade_date	trade_type	value
23	BBY	Barry Corie S	119.03	-10855	2021-03-23	S - Sale	-1.2921e+06
53	BBY	Hartman Todd G.	79	-3165	2020-12-06	S - Sale	-2.5004e+05
55	BBY	Scarlett Kathleen	77.58	-1348	2020-02-06	S - Sale	-1.0458e+05
33	BBY	Alexander Whitney L	113.88	-968	2021-03-16	S - Sale	-1.1023e+05
5	GME	Cohen Ryan	15.51	1226400	2020-12-17	P - Purchase	1.9026e+07
28	BBY	Peterson Allison	113.88	-433	2021-03-16	S - Sale	-49308
42	BBY	Mohan Rajendra M	112.9	-65000	2020-08-26	S - Sale+OE	-7.3382e+06
50	BBY	Schulze Richard M	86.29	-1000000	2020-06-26	S - Sale	-8.6291e+07
49	BBY	Schulze Richard M	86.64	-6000	2020-08-07	S - Sale	-5.1983e+05
10	BBY	Watson Mathew	121.74	-13	2021-04-13	S - Sale	-1583

- Queries:

- Create an index on the clustering column ‘Ticker’:

```
cqlsh:reddit> CREATE INDEX ON insider(ticker);
```

- Retrieve all the data with GME ticker only:

```
cqlsh:reddit> select * from insider where ticker='GME';
```

```
cqlsh:reddit> select * from insider where ticker='GME';
```

id	ticker	insider_name	price	qty	trade_date	trade_type	value
5	GME	Cohen Ryan	15.51	1226400	2020-12-17	P - Purchase	1.9026e+07
1	GME	Vrabeck Kathy P	27.99	-50000	2021-01-13	S - Sale	-1.3993e+06
0	GME	Fernandez Raul J	37.71	-3500	2021-01-15	S - Sale	-1.3198e+05
2	GME	Fernandez Raul J	35.28	-34619	2021-01-13	S - Sale	-1.2212e+06
4	GME	Wolf Kurt James	21.22	-810000	2021-12-01	S - Sale	-1.719e+07
3	GME	Dunn Lizabeth	31.34	-5000	2021-01-13	S - Sale	-1.567e+05

(6 rows)
cqlsh:reddit>

- Create a secondary index on column ‘qty’:

```
cqlsh:reddit> CREATE INDEX ON insider(qty);
```

- Retrieve the data of the person who had a qty above 1 million and whom belonged to GME:

```
cqlsh:reddit> select * from insider where ticker='GME' and
qty>1000000 allow filtering;
```

```
id | ticker | insider_name | price | qty | trade_date | trade_type | value
-----+-----+-----+-----+-----+-----+-----+-----
5 | GME | Cohen Ryan | 15.51 | 1226400 | 2020-12-17 | P - Purchase | 1.9026e+07
(1 rows)
```

- TRUNCATE the created insider table

```
cqlsh:reddit> truncate insider;
cqlsh:reddit> select * from insider;
```

```
id | ticker | insider_name | price | qty | trade_date | trade_type | value
-----+-----+-----+-----+-----+-----+-----+-----
```

We can see that all values from the insider table are now deleted.

Reading Cassandra from Python:

In the below section, we will provide a brief description of how we were able to read some data from the stored Cassandra table “REDDIT”, using queries from Python.

- First we created a cluster connection to connect to our “localhost”
- Then we created a session to connect to our cluster
- We connected to our previously created keyspace “reddit”.

```
from cassandra.cluster import Cluster
from cassandra.policies import DCAwareRoundRobinPolicy

cluster = Cluster(['localhost'],load_balancing_policy=DCAwareRoundRobinPolicy(),port=9042,protocol_version=4)
session = cluster.connect()
session.set_keyspace('reddit')
```

- We created a session to retrieve all the data under reddit table and display them

```
rows = session.execute('SELECT * FROM reddit')
for user_row in rows:
    print(user_row.date,user_row.frequency_bullish, user_row.length,user_row.num_comments,user_row.score)

2021-02-07 191 2912 22850 260791
2021-02-14 535 3814 11045 114977
2021-01-31 10166 132261 294336 2348238
2021-02-08 1263 15125 63191 2302883
2021-02-18 846 9934 142427 292109
2021-02-28 188 2423 3584 44779
2021-02-10 1595 16550 98984 236327
2021-01-29 24799 314870 769703 1277809
2021-01-17 535 3435 11490 378
2021-02-19 732 8823 57565 139968
2021-02-11 1098 15895 51072 98392
2021-02-20 427 5552 13928 566
2021-01-10 87 990 3738 150
2021-01-09 103 1118 1230 145
2021-01-12 186 2220 4089 510
2021-01-25 4961 52028 3302 5168
2021-01-19 1076 12164 71009 5247
2021-02-13 531 5858 25485 93191
2021-01-26 2617 29000 2050 2662
2021-01-15 1252 13971 26084 7633
```

- We created a session to execute and retrieve all the data corresponding to date '2021-02-13'.

```
rows = session.execute("SELECT * FROM reddit WHERE date='2021-02-13'")
for user_row in rows:
    print(user_row.date, user_row.frequency_bullish, user_row.length, user_row.num_comments, user_row.score)
```

```
2021-02-13 531 5858 25486 93191
```

- We created another session to retrieve all the date from Reddit table where the bullish frequency was greater than 10000 and display the full features of the first retrieved record.

```
rows = session.execute("SELECT * FROM reddit WHERE frequency_bullish >= 10000 ALLOW FILTERING")
print (rows[0])
for user_row in rows:
    print(user_row.date, user_row.frequency_bullish, user_row.length, user_row.num_comments, user_row.score)
```

```
Row(date='2021-01-31', frequency_bullish=10166, length=132261, num_comments=294336, score=2348238)
2021-01-31 10166 132261 294336 2348238
2021-01-29 24799 314870 769783 1277809
2021-01-27 12631 200319 450177 116086
2021-02-02 18195 242751 495273 1922369
2021-01-30 11467 154669 443545 2631208
2021-01-28 34975 444812 534346 214955
2021-02-01 17458 231787 809892 3228585
```

Conclusion & Recommendations

After going through the process of studying the variation of the GME stock price during the abnormal period ranging between the end of January-2021 and early February-2021, we observe that there was a significant impact of the social media forum on the supply and demand of the stock price; therefore, resulting in a synthetic rise in its market price. However, this price hike was not backed by the firm's fundamental data performance nor by the sentiment of the firm's employees as evident in the short positions of its insider trading pre-event and even merely a shy sales post-event.

Data collected was extremely usefully for ad hoc analysis. However, if we can track social media publications and reactions and create a real-time streaming process in which data is updated in the database, especially in Apache Cassandra since it is highly efficient in this case, then we can probably create an alert mechanism to prevent future disruptions from happening.

As for the recommendations, the results of this project raise a red flag for the market legislators to address this issue and to find a way to minimize the impact of mainstream bloggers on serious investing activities. This is considered a problem since it resulted in a market manipulation and wiping out small investors' money, even putting their pensions in jeopardy. Another recommendation that may be issued in this case is to consider social media in behavioral finance researches and elaborating more on what might be a threat to the industry's sustainability.

References

1. "Predicting Market Volatility Using Semantic Vectors and Google Trends", by: Anusha Balakrishnan and Kalpit Dixit.
2. "News and social media emotions in the commodity market", by: Jiancheng Shen, Mohammad Najand, Feng Dong, Wu He.
3. "Securities regulation and class warfare", by: Jonathan R. Macey.
4. "Squeezing Shorts Through Social News Platforms", by: Franklin Allen, Eric Nowak, Matteo Pirovano, and Angel Tengulov.
5. "Sentiment Analysis of Events from Twitter Using Open-Source Tool", by: Rajkumar S. Jagdale, Vishal S. Shirsat , Sachin N. Deshmukh.
6. [www. reddit.com/r/wallstreetbets_/](http://www.reddit.com/r/wallstreetbets_/)
7. [www. finance.yahoo.com](http://www.finance.yahoo.com)
8. www.openinsider.com/screener?s=GME
9. [www. openinsider.com/screener?s=BBY](http://www.openinsider.com/screener?s=BBY)