Project Write-up

Introduction

For our project, we chose to use the Big tech stock price dataset which contains daily stock prices of major technology companies in US like Apple, Amazon, Google, etc. The cutoff of the dataset is from January 2010 to end of 2023. Besides that, we also collected manually data on big events which might have major effects on these companies' stock for example: updates of software, new products rolling out, the CEO left. The data is collected from multiple trusted sources like: the New York Times, Bloomberg, Wikipedia, The Guardians and were cross referenced checked. Understanding the trends and fluctuations in the stock values of leading technology companies is crucial, especially given their influential role in global markets and technological innovation. Analyzing this data allows investors, analysts, and stakeholders to gain deeper insights into how significant events, market dynamics, and emerging technologies such as large language models (LLMs) influence stock performance.

Question 01:

What are some fundamentals and useful indicators that newbies must know to perform basic technical analysis?

Introduction

Technical analysis has always been the backbone of many trading strategies for years and are still very common these days. In this question, I focus on price data (open, high, low, close), volume, and derive technical indicators from the Big Tech Stock Price dataset. I want to show a comprehensive entry point for those who are new to stock analysis, including myself, by visualizing essential technical indicators alongside historical prices. And this implementation can guide us to better understand how some metrics are calculated, visualized and used.

Approach

We implemented two different connected graph:

1. **Candlestick Chart with Volume Panel**: Candlestick charts are the golden standard for price visualization as they efficiently illustrate four **data points (open, high, low, close)** in a single element while clearly indicating price movement with color. I paired this with a subplot for **volume**, using color mapping to indicate buying pressure (green) versus selling pressure (red).
2. **Interactive Technical Indicator Overlay**: We added toggle technical indicators (SMA, EMA, Bollinger Bands) as line and area charts overlaid on the candlestick chart. This approach uses color mapping to distinguish different indicators
   - Blue for SMA20 and purple for SMA50
   - Green dashed line for EMA20 and magenta dashed line for EMA50
   - Gray bands for Bollinger Bands

By doing this, we can easily see how these separate indicators interact with the price movements without making the chart overwhelming.

Analysis

The implementation uses `Plotly` and `Dash` to create an interactive dashboard that allows us to:

- Calculate the technical indicators

```
def calculate_technical_indicators(data, ma_periods=[20, 50], bb_period=20, bb_std=2):
    """Calculate technical indicators for stock data"""
    indicators = {}

    # Calculate Simple Moving Averages
    for period in ma_periods:
        if len(data) >= period:
            indicators[f'SMA_{period}'] = data['close'].rolling(window=period).mean()

    # Calculate Exponential Moving Averages
    for period in ma_periods:
        if len(data) >= period:
            indicators[f'EMA_{period}'] = data['close'].ewm(span=period, adjust=False).mean()

    # Calculate Bollinger Bands
    if len(data) >= bb_period:
        middle_band = data['close'].rolling(window=bb_period).mean()
        std_dev = data['close'].rolling(window=bb_period).std()
        indicators['BB_middle'] = middle_band
        indicators['BB_upper'] = middle_band + (std_dev * bb_std)
        indicators['BB_lower'] = middle_band - (std_dev * bb_std)

    return indicators
```

$$\mathrm{SMA} = \frac{A_1 + A_2 + \ldots + A_n}{n}$$

$$\textbf{where:}$$

$$A_n = \text{the price of an asset at period } n$$

$$n = \text{the number of total periods}$$

Simple Moving Averages Formula

$$EMA = \text{Price}(t) \times k + EMA(y) \times (1 - k)$$

where:

$t = \text{today}$

$y = \text{yesterday}$

$N = \text{number of days in EMA}$

$k = 2 \div (N + 1)$

Exponential Moving Averages Formula

\# of standard deviations away from mean (bandwidth / 2)

SMA of price for given period

$$BBands_i = x_i \pm \sigma_i * d$$

observed value for single period

standard devitation of SMA over a given perioa

α

Bollinger Bands Formula

- Create candlestick charts for all 15 big techs to show daily trading data, and a volume subplot, with an option to choose the specific companies and multiple indicators to display.

```
app = JupyterDash(__name__)
app.layout = html.Div([
    # Layout components including dropdowns and toggle controls
    # ...
    dcc.Graph(id='candlestick-chart')
])

@app.callback(
    Output('candlestick-chart', 'figure'),
    [Input('company-dropdown', 'value'),
     Input('sma-toggle', 'value'),
     Input('ema-toggle', 'value'),
     Input('bb-toggle', 'value')]
```

```
)
def update_chart(symbol, sma_toggle, ema_toggle, bb_toggle):
    # Update chart based on user selections
    # ...
```

Figure 01: Big Tech Stock Dashboard

Discussion

Each technical indicators reveal different aspects of market behavior.

- **Simple Moving Averages** effectively smooth out price noise and highlight the overall trend. In common practice, it is like a smooth line that helps we see if the overall trend is up, down, or flat.
  - *Example:* A 20-day SMA takes the **last 20 days of closing prices**, adds them up, and divides by 20.
- **Exponential Moving Averages** is very similar to SMA, but a bit smarter, it responds more quickly to recent price changes, making it more sensitive to emerging trends but also more prone to false signals. It is usually used for quicker trend detection or more responsive trading signals.
- **Bollinger Bands** demonstrate volatility expansion and contraction cycles.
  - **Two outer bands** show how “wide” or “volatile” the price has been. When the bands are **wide**, the market is volatile. When they’re **narrow**, it’s calm.

- It can help us to spot when prices might be **too high** (touching the upper band) or **too low** (touching the lower band) and get a feel when a **breakout** might happen after a squeeze (when bands get super tight).
- For Big Tech stocks, we often see periods of **narrow band** followed by **explosive moves**. This pattern is particularly around e**arnings announcements and major product launches**, where the stock price tests the upper or lower bands before reverting to the mean.

Question 2

How did product/service launched dates, company-related events and major economic events (COVID-19 pandemic, the rise of LLMs) affected the volatility and recovery patterns of different tech stocks? What are the correlations between stock price trends of these big techs?

Introduction

In this part, we aim to understand the impact of major events, such as product launches and COVID-19 on stock volatility and recovery patterns, which is very important. We explore how these events influenced the stock price of big tech companies and also examine the correlations between their price trends.

Approach

We implemented two visualization approaches to analyze event impacts:

1. **Multi-company Line Chart**: We chose line charts for comparing multiple companies during the same event period simply because lines are better than candle sticks at showing trends over time and make it easy to compare patterns across different stocks. We applied color mapping to distinguish between different companies, making it simple to track each stock's performance. The chart also includes vertical markers for key dates within each event period for understanding price movements.
2. **Normalized Percentage Change View**: To address the challenge of comparing stocks with vastly different price scales (e.g., Apple vs. Amazon), I implemented a toggle feature that normalizes all prices to percentage changes from the starting point. This transformation allows for direct comparison of relative performance regardless of absolute price levels. This approach uses the same line chart structure but transforms the y-axis from absolute prices to percentage changes, enabling us to see which companies were most significantly impacted by specific events relative to their starting values.

And 1 visualization for companies correlations:

Stock Price Correlation Heatmap: We selected a heatmap as our visualization method because it effectively illustrates the correlations between the stock prices of these companies. By applying a color gradient—red to represent strong positive correlations and blue to indicate strong negative correlations—we can easily identify the strength and direction of these relationships. This approach allows us to clearly observe how the stock price of one company influences another, determine which companies tend to follow similar trends, and explore the underlying reasons driving these patterns.

Analysis

This contains 2 parts: **Tech Stock Response to Major Events** and **Companies Correlation**

- Tech Stock Response to Major Events:
    - Uses `Plotly` and `Dash` to create an interactive dashboard that:
        - Filters stock data to the date range for the selected event
        - Creates line charts showing price movements for all tech companies during the selected event period
        - Adds a toggle button that converts **Raw Prices** to **Percentage Changes** and vice-versa
        - Overlay annotations for key milestone dates (like "Black Thursday" or "ChatGPT Released")

```
def generate_event_comparison_chart(event_name, normalize=False):
    """Generate a line chart comparing all companies during a specific event"""

    # Get event configuration
    event_config = events_config[event_name]
    start_date = pd.to_datetime(event_config["date_range"][0])
    end_date = pd.to_datetime(event_config["date_range"][1])

    # Filter stock data for the event time period
    event_data = stock_df[(stock_df['date'] >= start_date) & (stock_df['date'] <= end_date)].

    fig = go.Figure()
    for symbol in stock_symbols:
        company_data = event_data[event_data['stock_symbol'] == symbol].copy()

        if len(company_data) == 0:
            continue
        company_data = company_data.sort_values('date')
        y_values = company_data['close']
        if normalize and len(company_data) > 0:
            first_value = company_data['close'].iloc[0]
            if first_value > 0:
                y_values = (company_data['close'] / first_value - 1) * 100
                y_axis_title = "Percentage Change (%)"
            else:
                y_axis_title = "Price (USD)"
        else:
            y_axis_title = "Price (USD)"

        # Add traces for each company
        # ...
```

```
@app_events.callback(
    Output('event-comparison-chart', 'figure'),
```

```
    [Input('event-dropdown', 'value'), Input('normalize-toggle', 'value')]
)
def update_event_chart(event_name, normalize_option):
    normalize = normalize_option == 'percentage'
    return generate_event_comparison_chart(event_name, normalize)
```

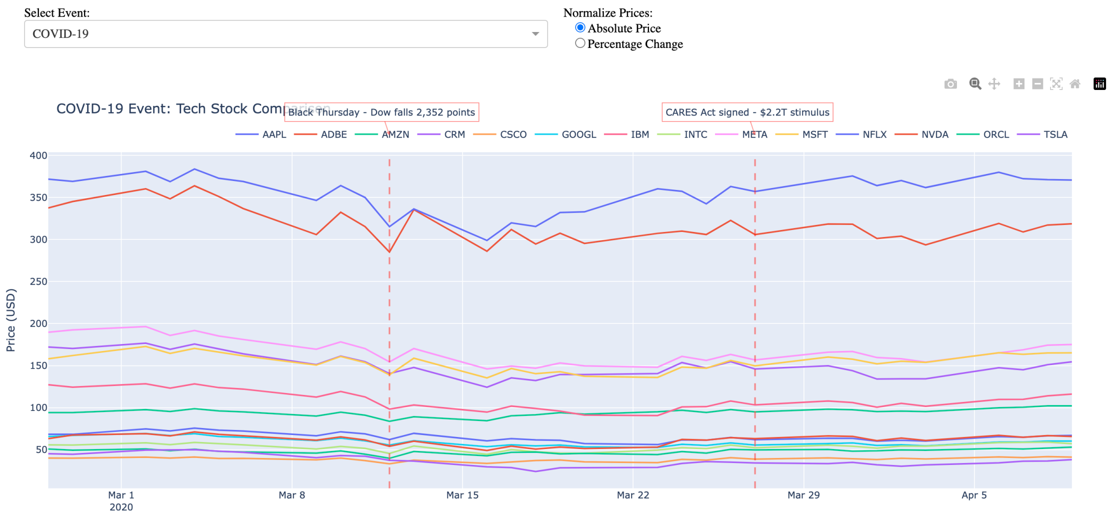


Figure 02: Tech Stock Response to COVID-19 (by Absolute Change)

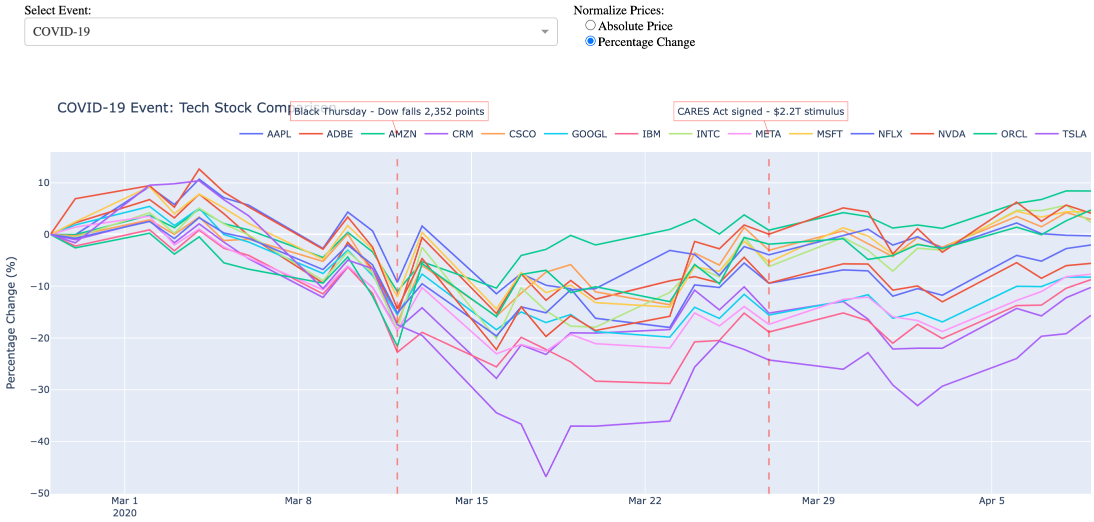


Figure 03: Tech Stock Response to COVID-19 (by Percentage Change)

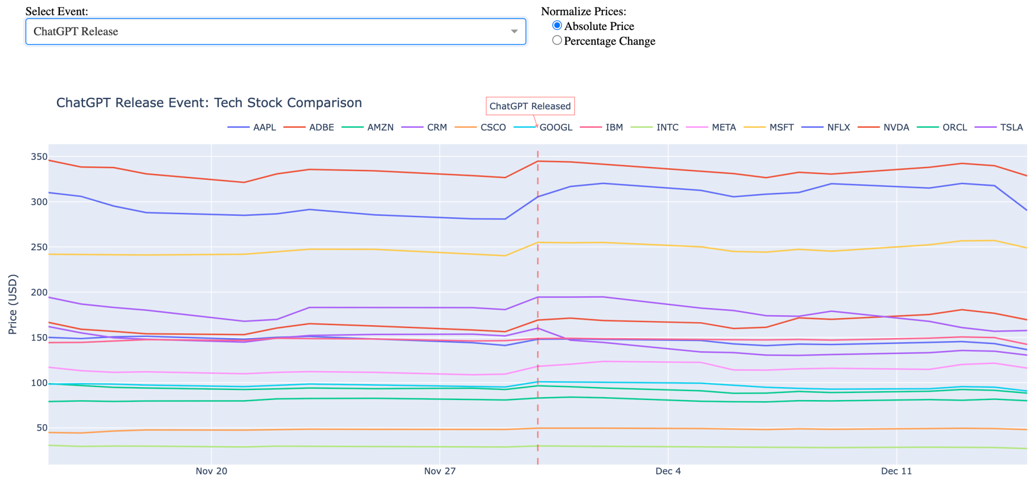


Figure 04: Tech Stock Response to ChatGPT Release (by Percentage Change)

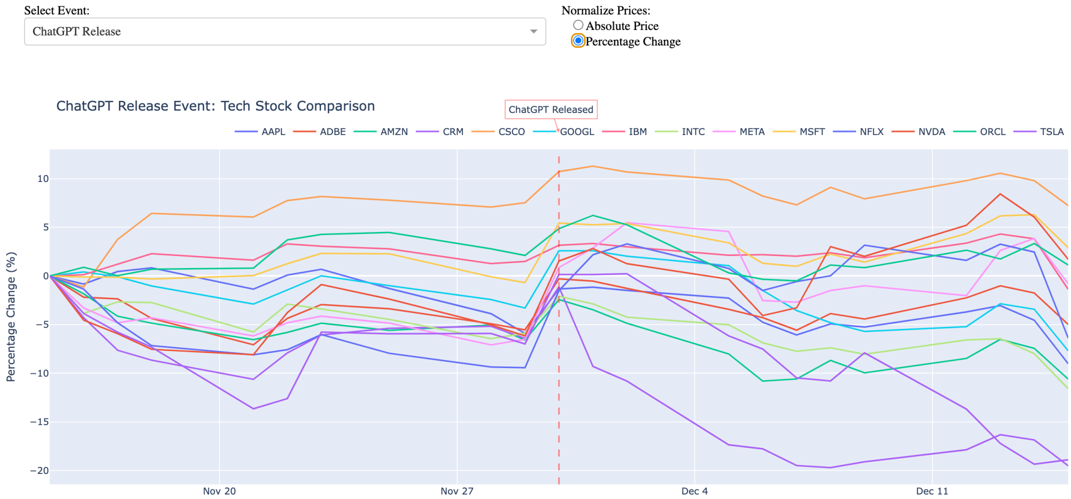
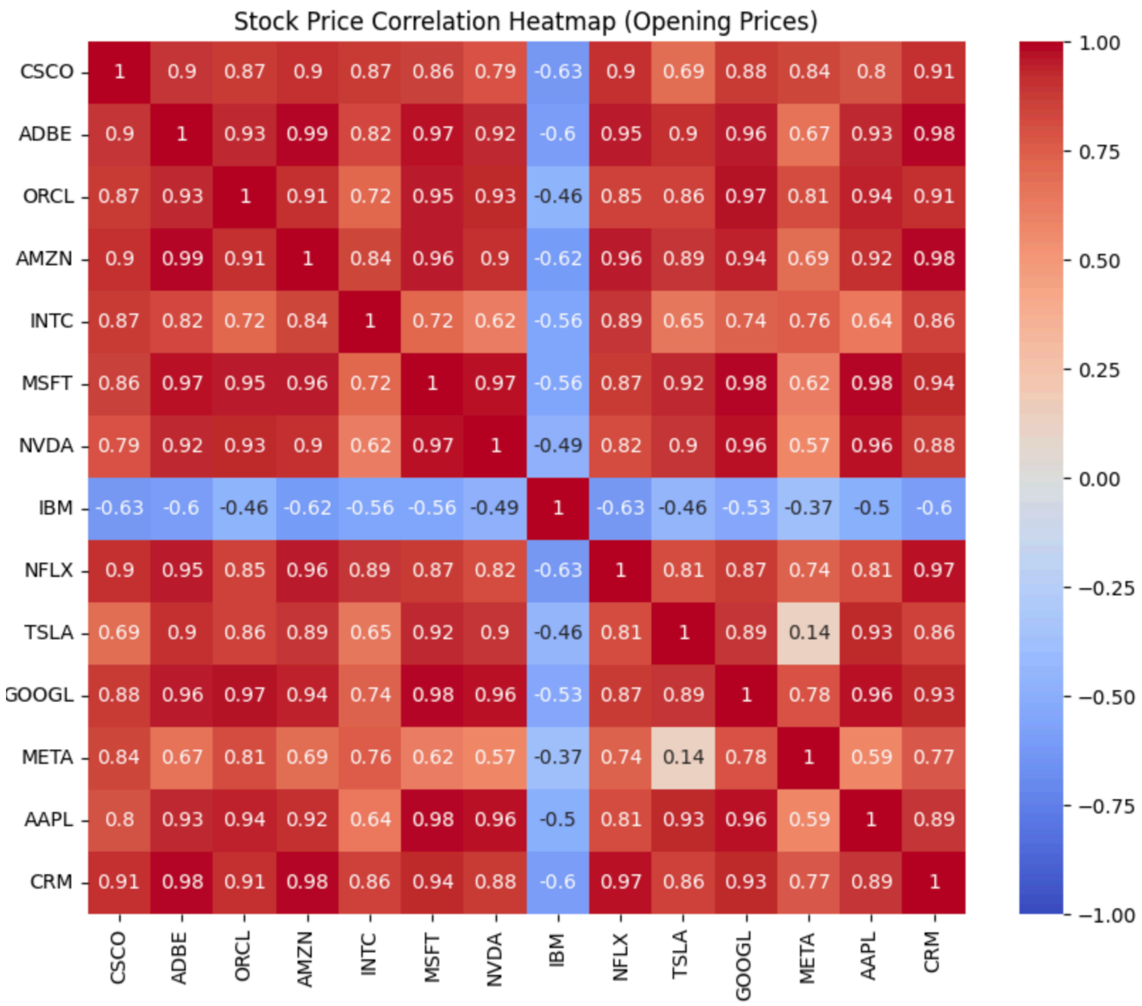


Figure 05: Tech Stock Response to ChatGPT Release (by Percentage Change)

- Companies correlation:
  - Use `seaborn` and `matplotlib` to create and plot the heatmap

```
folder_path = "data_original"

stock_data = {}
for file in os.listdir(folder_path):
    if file.endswith(".csv"):
        df = pd.read_csv(os.path.join(folder_path, file))
        stock_name = file.replace(".csv", "")
        stock_data[stock_name] = df["Close"]

combined_df = pd.DataFrame(stock_data)

correlation_matrix = combined_df.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", vmin=-1, vmax=1, center
```

```
plt.title("Stock Price Correlation Heatmap (Opening Prices)")
plt.show()
```



Discussion

- For Tech Stock response to major events:
  - On Black Thursday
    - On March 12, 2020, global stock markets experienced one of the worst single-day declines in history, driven by fears surrounding the escalating COVID-19 pandemic.
    - As you can see, all the companies took a major dive, it is clearer in the percentage plot.
  - CARES Act
    - The CARES Act (Coronavirus Aid, Relief, and Economic Security Act) was a historic $2.2 trillion economic stimulus package signed into law by President Donald Trump on March 27, 2020, in response to the economic fallout from the COVID-19 pandemic.
      It aimed to provide financial relief to individuals, businesses, healthcare systems, and state/local governments to mitigate the impact of widespread lockdowns and economic disruptions.
    - However, despite of the support, companies's stock price still slightly declines.

- ChatGPT Release
  - We all know what ChatGPT is nowadays. When it was released for the first time, the world was thrilled about it. Therefore, all the stock price go up in values, as for the big reasons that most of them are tech companies.
- For company correlation:
  - ***Tech/Software Stocks (AMZN, ADBE, MSFT, ORCL, CRM, GOOGL, AAPL):*** These stocks show consistently high correlations (0.80–0.99) with each other. This suggests that the tech sector moves together, likely driven by shared market trends like cloud computing, AI, or consumer tech demand.
  - **Semiconductor Stocks (Nvidia and Intel):** Nvidia's kind of in the mix with the tech crowd, with correlations ranging from 0.57 to 0.96. That makes sense since they're killing it with AI chips and GPUs. But Intel? It's lagging a bit, with correlations only between 0.64 and 0.87. I guess that's not too surprising—Intel's been having a tough time lately, while Nvidia's been on fire.
  - **Streaming with Netflix:** Netflix is hanging out with the tech stocks, showing correlations from 0.74 to 0.95, which is pretty solid. But it's not as buddy-buddy with Tesla, where the correlation drops to 0.81, or Meta, at 0.74. I think that's because Netflix is doing its own thing in the streaming world, not totally tied to what's driving those other companies.
  - **Tesla:** Tesla's a bit of a lone wolf here. Its got decent correlations with most stocks, between 0.81 and 0.93, but then there's this weirdly low correlation with Meta at just 0.14. It's like Tesla's off doing its own thing—maybe because it's more about cars, energy, and Elon's big ideas, which don't really overlap with what's moving Meta.
  - **IBM**: IBM's the oddball of the group, with negative correlations across the board. It's almost like when everyone else is up, IBM down, and vice verka. Maybe it's because IBM's more old-school, focusing on stuff that doesn't quite vibe with the hot tech trends the others are chasing.

Conclusion

In this analysis, we explored the fundamentals of technical indicators crucial for beginners, emphasizing their roles in interpreting stock market trends through Simple Moving Averages (SMA), Exponential Moving Averages (EMA), and Bollinger Bands. Our detailed examination of major product launches, significant company-related events, and economic occurrences like the COVID-19 pandemic and the rise of large language models (LLMs) highlighted notable impacts on stock volatility and recovery patterns across major tech firms.

Through event impact analysis and correlation studies, we observed distinct and consistent industry-driven relationships among tech companies. Companies within similar technological domains displayed highly correlated stock movements, reflecting shared market dynamics and responses to significant economic events. We also identified specific events, such as the release of ChatGPT and economic stimulus packages like the CARES Act, significantly influencing market behaviors.

To extend these insights further, future analyses could integrate higher frequency trading data to capture intra-day dynamics, employ more sophisticated predictive modeling to forecast stock responses to future events, and examine underlying market factors driving correlation shifts over time. These steps could greatly enhance our understanding and assist stakeholders in making informed investment decisions