1. **How have the stock prices of the top 5 S&P 500 companies trended over the past years?**
   1. Obtain historical Data.
   2. Organise data
   3. Analyse the trend – *use line chart or candle stick chart*
   4. Compare the trends
   5. Interpret the results
2. **Which of the top 5 S&P 500 companies had the highest return on investment over the selected time period?**
3. Net Profit = Final Investment Value - Initial Investment Cost
4. ROI = (Net Profit / Cost of Investment) \* 100

import requests

import pandas as pd

# Step 1: Obtain an API key

api\_key = 'your\_api\_key'

# Step 2: Install the necessary libraries

# Step 3: Make API requests

symbols = ['AAPL', 'MSFT', 'AMZN', 'NVDA', 'GOOGL']

rois = {}

for symbol in symbols:

endpoint = f'https://www.alphavantage.co/query?function=TIME\_SERIES\_DAILY\_ADJUSTED&symbol={symbol}&apikey={api\_key}'

response = requests.get(endpoint)

# Process the API response

if response.status\_code == 200:

data = response.json()

time\_series = data['Time Series (Daily)']

# Step 4: Process the API response

df = pd.DataFrame.from\_dict(time\_series, orient='index')

df = df.astype(float)

df.sort\_index(ascending=True, inplace=True)

# Step 5: Calculate ROI

initial\_investment = df.iloc[0]['4. close']

final\_value = df.iloc[-1]['4. close']

net\_profit = final\_value - initial\_investment

roi = (net\_profit / initial\_investment) \* 100

rois[symbol] = roi

else:

print(f'Request failed for {symbol} with status code:', response.status\_code)

# Print the calculated ROIs

for symbol, roi in rois.items():

print(f"The ROI for {symbol} is: {roi:.2f}%")

1. **How volatile are the stock prices of the top 5 S&P 500 companies?**

import pandas as pd

import numpy as np

# Step 1: Obtain historical stock price data

# historical stock price data for each company in separate DataFrames: df\_aapl, df\_msft, df\_amzn, df\_nvda, df\_googl

# Step 2: Calculate daily returns

df\_aapl['Daily\_Return'] = df\_aapl['Close'].pct\_change()

df\_msft['Daily\_Return'] = df\_msft['Close'].pct\_change()

df\_amzn['Daily\_Return'] = df\_amzn['Close'].pct\_change()

df\_nvda['Daily\_Return'] = df\_nvda['Close'].pct\_change()

df\_googl['Daily\_Return'] = df\_googl['Close'].pct\_change()

# Step 3: Calculate standard deviation

volatility\_aapl = np.std(df\_aapl['Daily\_Return'])

volatility\_msft = np.std(df\_msft['Daily\_Return'])

volatility\_amzn = np.std(df\_amzn['Daily\_Return'])

volatility\_nvda = np.std(df\_nvda['Daily\_Return'])

volatility\_googl = np.std(df\_googl['Daily\_Return'])

# Step 4: Interpret the results

print("Volatility of Apple Inc. (AAPL):", volatility\_aapl)

print("Volatility of Microsoft Corp. (MSFT):", volatility\_msft)

print("Volatility of Amazon.com Inc. (AMZN):", volatility\_amzn)

print("Volatility of NVIDIA Corp. (NVDA):", volatility\_nvda)

print("Volatility of Alphabet Inc. Class A (GOOGL):", volatility\_googl)

**4. Which company showed the highest price stability in the past year? (Think we can merge to question 3. – use standard deviation)**

5. **How do the companies compare to each other in terms of stock performance and volatility?**

1. Calculate relevant stock performance metrics for each company, thinking of exploring absolute return or total return, or annualized return.
2. Calculate stock volatility: Use standard deviation to gain insights into the stability or variability of the stock prices for each company.
3. Visualize the data: Use line charts or bar charts or candle charts, to compare the stock performance and volatility of the top 5 companies side by side. This will allow for a visual comparison of their performance over time.

**6. Are there any correlations in the performance of the top 5 companies that can be identified?**

import requests

import pandas as pd

# Step 1: Obtain API keys

api\_key1 = 'your\_api\_key1'

api\_key2 = 'your\_api\_key2'

# Step 2: Install the necessary libraries

# Step 3: Make API requests

url1 = 'https://api.example.com/endpoint1'

url2 = 'https://api.example.com/endpoint2'

headers = {

'Authorization': 'Bearer ' + api\_key1,

'Content-Type': 'application/json'

}

response1 = requests.get(url1, headers=headers)

response2 = requests.get(url2, headers=headers)

# Process the API responses

if response1.status\_code == 200 and response2.status\_code == 200:

data1 = response1.json()

data2 = response2.json()

# Step 4: Process the API response

# Extract the required data and store in pandas DataFrames or Series

# Step 5: Calculate the correlation

correlation\_coefficient = data1['variable1'].corr(data2['variable2'])

# Step 6: Interpret the result

print("Correlation coefficient:", correlation\_coefficient)

else:

print('Request failed with status code:', response1.status\_code, response2.status\_code)

1. **What are the short-term and long-term trends observable in the trading data for these companies? – MAY BE WE CAN MERGE THIS QUESTION TO NO. 1**

import pandas as pd

import matplotlib.pyplot as plt

# Step 1: Obtain historical stock price data

# Assuming you have the historical stock price data for each company in separate DataFrames: df\_aapl, df\_msft, df\_amzn, df\_nvda, df\_googl

# Step 2: Define the time periods for short-term and long-term trends

short\_term\_period = 30 # Example: 30 days

long\_term\_period = 180 # Example: 180 days

# Step 3: Calculate moving averages

df\_aapl['SMA\_short\_term'] = df\_aapl['Close'].rolling(window=short\_term\_period).mean()

df\_aapl['SMA\_long\_term'] = df\_aapl['Close'].rolling(window=long\_term\_period).mean()

# Repeat the above visualization steps for the other companies

# Repeat the above steps for the other companies

# Step 4: Analyze the trends

# Compare the short-term and long-term moving averages to identify the trends

# Step 5: Visualize the trends

plt.plot(df\_aapl['Date'], df\_aapl['Close'], label='AAPL')

plt.plot(df\_aapl['Date'], df\_aapl['SMA\_short\_term'], label='Short-term SMA')

plt.plot(df\_aapl['Date'], df\_aapl['SMA\_long\_term'], label='Long-term SMA')

plt.xlabel('Date')

plt.ylabel('Stock Price')

plt.title('Short-term and Long-term Trends of AAPL')

plt.legend()

plt.show()

1. **Conclusions and recommendations to potential employers.**