

# Journey To DataScience Project (Group\_13)

Salam AL-Kaissi Mayra Suarez Ha LE

## Financial Draft Reports on Stock Market Historical Insights (FinDraft)

In this analysis, we aimed to address the problem of generating finance-readable insights for individual investors, specifically focusing on Tesla's stock performance. We utilized multiple datasets including IPO data, top 5 tech companies' stock data, sentiment analysis from news articles, and S&P 500 index data (SPY) to understand market behavior.

We initially explored the top 5 tech companies, and due to Tesla's significant price movements in 2023, we focused more on Tesla for detailed analysis. Exploratory analysis was conducted through visualizations and correlation studies to link sentiment trends with stock price fluctuations. The findings revealed notable peaks in both sentiment and price during mid-June to mid-July 2023, offering valuable insights into the factors affecting Tesla's stock performance during this period.

## Statistical Data Analysis

### Import Libraries

```
In [1]: import pandas as pd
import numpy as np
import yfinance as yf
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import requests
import datetime
import time
```

### Data Collection

In this step, we will collect the stock data for the top 5 companies based on their stock symbols. We will use Yahoo Finance API to get the historical stock data for the year 2023.

```
In [2]: # Define the top 5 companies by their stock symbols
top_companies = ["AAPL", "MSFT", "GOOG", "AMZN", "TSLA"]

# Define the time range for data collection (Year 2023)
start_date = "2023-01-01"
end_date = "2023-12-31"
```

```
# Initialize an empty List to store data
data = []

for ticker in top_companies:
    stock_data = yf.download(ticker, start=start_date, end=end_date)
    stock_data.reset_index(inplace=True)
    stock_data['Symbol'] = ticker
    data.append(stock_data)

# Concatenate all data into a single DataFrame
combined_df_2023 = pd.concat(data, ignore_index=True)

combined_df_2023.head(2)
```

```
[*****100%*****] 1 of 1 completed
```

Out[2]:

	Date	Open	High	Low	Close	Adj Close	Volume	Symb
0	2023-01-03	130.279999	130.899994	124.169998	125.070000	123.904625	112117500	AA
1	2023-01-04	126.889999	128.660004	125.080002	126.360001	125.182610	89113600	AA

Column Definitions:

Open: The opening price of the stock for the given day.

High: The highest price of the stock for the given day.

Low: The lowest price of the stock for the given day.

Close: The closing price of the stock for the given day.

Volume: The number of shares traded during the given day.

Symbol: The ticker symbol representing the stock.

In [3]:

```
# Add SPY500 index
spy_data_2023 = yf.download(tickers = "^GSPC", start=start_date, end=end_date,in
spy_data_2023.head(2)
```

```
[*****100%*****] 1 of 1 completed
```

Out[3]:	Open	High	Low	Close	Adj Close	Volume
<b>Date</b>						
2023-01-03	3853.290039	3878.459961	3794.330078	3824.139893	3824.139893	3959140000
2023-01-04	3840.360107	3873.159912	3815.770020	3852.969971	3852.969971	4414080000

Define SPY500 columns Open: The opening value of the S&P 500 index for the given day.

High: The highest value of the S&P 500 index for the given day.

Low: The lowest value of the S&P 500 index for the given day.

Close: The closing value of the S&P 500 index for the given day.

Volume: The trading volume for the S&P 500 index for the given day.

Symbol: The ticker symbol representing the index (SPY500).

```
In [4]: # IPO data collection
# Fetch IPO data from the given URL https://stockanalysis.com/ilos/2023/
import pandas as pd
import requests

headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
}

url = "https://stockanalysis.com/ilos/2023/"
response = requests.get(url, headers=headers)

ipo_dfs = pd.read_html(response.text)
```

C:\Users\USER\AppData\Local\Temp\ipykernel\_15076\202373422.py:13: FutureWarning:  
Passing literal html to 'read\_html' is deprecated and will be removed in a future  
version. To read from a literal string, wrap it in a 'StringIO' object.  
ipo\_dfs = pd.read\_html(response.text)

```
In [5]: ipo_data_2023 = ipo_dfs[0]
ipo_data_2023.head(2)
```

	IPO Date	Symbol	Company Name	IPO Price	Current	Return
0	Dec 27, 2023	IROH	Iron Horse Acquisitions Corp.	\$10.00	\$10.23	2.30%
1	Dec 19, 2023	LGCB	Linkage Global Inc	\$4.00	\$0.27	-93.31%

IPO Columns: IPO Date: The date when the company went public (IPO date).

Symbol: The ticker symbol representing the stock.

Company Name: The name of the company that went public.

IPO\_Price: The initial price of the stock during the IPO.

Current\_Price: The current trading price of the stock.

Return: The percentage return since the IPO (calculated as (Current\_Price - IPO\_Price) / IPO\_Price).

## Exploratory Data Analysis (EDA) and Add Feature Engineering

In this step, we will perform EDA to understand the data distribution and trends. and will add technical indicators to the dataset to enhance the analysis.

In [6]: `combined_df_2023.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1250 entries, 0 to 1249
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype  
---  --          -----          ----- 
 0   Date        1250 non-null   datetime64[ns]
 1   Open         1250 non-null   float64 
 2   High         1250 non-null   float64 
 3   Low          1250 non-null   float64 
 4   Close        1250 non-null   float64 
 5   Adj Close    1250 non-null   float64 
 6   Volume       1250 non-null   int64  
 7   Symbol       1250 non-null   object 
dtypes: datetime64[ns](1), float64(5), int64(1), object(1)
memory usage: 78.3+ KB
```

In [7]: `# Display basic statistics of the combined dataset`  
`print(combined_df_2023.describe())`

		Date	Open	High	Low	\
count		1250	1250.000000	1250.000000	1250.000000	
mean	2023-07-02 09:18:43.200000	188.719091	191.173623	186.493094		
min	2023-01-03 00:00:00	83.029999	85.419998	81.430000		
25%	2023-04-03 00:00:00	129.337502	130.898495	127.897497		
50%	2023-07-04 00:00:00	169.794998	173.049995	167.579994		
75%	2023-10-02 00:00:00	245.122501	250.959999	242.065002		
max	2023-12-29 00:00:00	383.760010	384.299988	378.160004		
std		NaN	78.081359	78.795057	77.273957	
		Close	Adj Close	Volume		
count	1250.000000	1250.000000	1.250000e+03			
mean	188.991980	188.134148	6.172171e+07			
min	83.120003	83.120003	8.828600e+06			
25%	129.122498	128.962963	2.637970e+07			
50%	170.730003	169.980309	4.837415e+07			
75%	246.492500	244.977497	7.641610e+07			
max	382.700012	380.619843	3.065906e+08			
std	78.039795	77.242566	4.575780e+07			

In [8]: `## Check for missing values`

```
print("Missing values in the dataset:")
print(combined_df_2023.isna().sum())
```

Missing values in the dataset:

```
Date      0
Open      0
High      0
Low       0
Close     0
Adj Close 0
Volume    0
Symbol    0
dtype: int64
```

In [9]: `spy_data_2023.info()`

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 250 entries, 2023-01-03 to 2023-12-29
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
 ---  --          -----          --    
 0   Open         250 non-null    float64 
 1   High         250 non-null    float64 
 2   Low          250 non-null    float64 
 3   Close        250 non-null    float64 
 4   Adj Close   250 non-null    float64 
 5   Volume       250 non-null    int64  
dtypes: float64(5), int64(1)
memory usage: 13.7 KB
```

In [10]: `ipo_data_2023.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 154 entries, 0 to 153
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
 ---  --          -----          --    
 0   IPO Date    154 non-null    object  
 1   Symbol       154 non-null    object  
 2   Company Name 154 non-null    object  
 3   IPO Price   154 non-null    object  
 4   Current      154 non-null    object  
 5   Return       154 non-null    object  
dtypes: object(6)
memory usage: 7.3+ KB
```

In [11]: `# convert to datetime`

```
ipo_data_2023['IPO Date'] = pd.to_datetime(ipo_data_2023['IPO Date'])
```

```
ipo_data_2023.head(2)
```

Out[11]:

	<b>IPO Date</b>	<b>Symbol</b>	<b>Company Name</b>	<b>IPO Price</b>	<b>Current</b>	<b>Return</b>
<b>0</b>	2023-12-27	IROH	Iron Horse Acquisitions Corp.	\$10.00	\$10.23	2.30%
<b>1</b>	2023-12-19	LGCB	Linkage Global Inc	\$4.00	\$0.27	-93.31%

In [12]: `missing_prices_df = ipo_data_2023[ipo_data_2023['IPO Price'].astype(str).str.find`

Out[12]:

	<b>IPO Date</b>	<b>Symbol</b>	<b>Company Name</b>	<b>IPO Price</b>	<b>Current</b>	<b>Return</b>
--	-----------------	---------------	---------------------	------------------	----------------	---------------

```
In [13]: # Remove '$'
ipo_data_2023['IPO Price'] = pd.to_numeric(ipo_data_2023['IPO Price'].str.replace('$', ''))
# call it again to transform 'object' to 'float64'
ipo_data_2023['IPO Price'] = pd.to_numeric(ipo_data_2023['IPO Price'])

# Convert "Current" column
ipo_data_2023['Current'] = pd.to_numeric(ipo_data_2023['Current'].str.replace('$', ''))

# Convert 'Return' to numeric format (percentage)
ipo_data_2023['Return'] = pd.to_numeric(ipo_data_2023['Return'].str.replace('%', ''))

ipo_data_2023.head(2)
```

```
Out[13]:
```

	<b>IPO Date</b>	<b>Symbol</b>	<b>Company Name</b>	<b>IPO Price</b>	<b>Current</b>	<b>Return</b>
<b>0</b>	2023-12-27	IROH	Iron Horse Acquisitions Corp.	10.0	10.23	0.0230
<b>1</b>	2023-12-19	LGCB	Linkage Global Inc	4.0	0.27	-0.9331

```
In [14]: ipo_data_2023.isnull().sum()
```

```
Out[14]:
```

IPO Date	0
Symbol	0
Company Name	0
IPO Price	0
Current	0
Return	0
dtype: int64	

```
In [15]: ipo_data_2023[ipo_data_2023.Return.isnull()]
```

```
Out[15]:
```

	<b>IPO Date</b>	<b>Symbol</b>	<b>Company Name</b>	<b>IPO Price</b>	<b>Current</b>	<b>Return</b>
--	-----------------	---------------	---------------------	------------------	----------------	---------------

```
In [16]: # operate with columns as a numeric type
ipo_data_2023['IPO Price'].mean()
```

```
Out[16]: 9.859935064935065
```

```
In [17]: # generate a new field -- SIMPLE calculation
ipo_data_2023['Price Increase'] = ipo_data_2023['Current'] - ipo_data_2023['IPO Price']
ipo_data_2023.head(1)
```

```
Out[17]:
```

	<b>IPO Date</b>	<b>Symbol</b>	<b>Company Name</b>	<b>IPO Price</b>	<b>Current</b>	<b>Return</b>	<b>Price Increase</b>
<b>0</b>	2023-12-27	IROH	Iron Horse Acquisitions Corp.	10.0	10.23	0.023	0.23

```
In [18]: # Descriptive Analytics of a dataset
ipo_data_2023.describe()
```

Out[18]:

	IPO Date	IPO Price	Current	Return	Price Increase
<b>count</b>	154	154.000000	154.000000	154.000000	154.000000
<b>mean</b>	2023-06-25 10:26:29.610389504	9.859935	10.071039	-0.247890	0.211104
<b>min</b>	2023-01-13 00:00:00	2.500000	0.000000	-0.999500	-21.000000
<b>25%</b>	2023-03-29 06:00:00	4.000000	1.012500	-0.817600	-3.990000
<b>50%</b>	2023-06-29 12:00:00	7.500000	3.055000	-0.470000	-2.710000
<b>75%</b>	2023-09-21 18:00:00	10.000000	10.880000	0.077750	0.767500
<b>max</b>	2023-12-27 00:00:00	51.000000	142.410000	5.207300	114.560000
<b>std</b>	NaN	8.046308	18.843418	0.846993	14.183750

In [19]:

```

import matplotlib.pyplot as plt
import seaborn as sns

# Sort by "Price Increase" and select the top 5 companies
top_5_ipos = ipo_data_2023.sort_values(by='Price Increase', ascending=False).head(5)

# Plotting the Price Increase for each IPO (top 5 only)
plt.figure(figsize=(12, 6))
sns.barplot(x='Price Increase', y='Company Name', data=top_5_ipos, palette='viridis')

# Add labels and title
plt.xlabel('Price Increase (USD)')
plt.ylabel('Company Name')
plt.title('Top 5 IPO Price Increases in 2023')
plt.grid(True)

# Display the plot
plt.show()

```

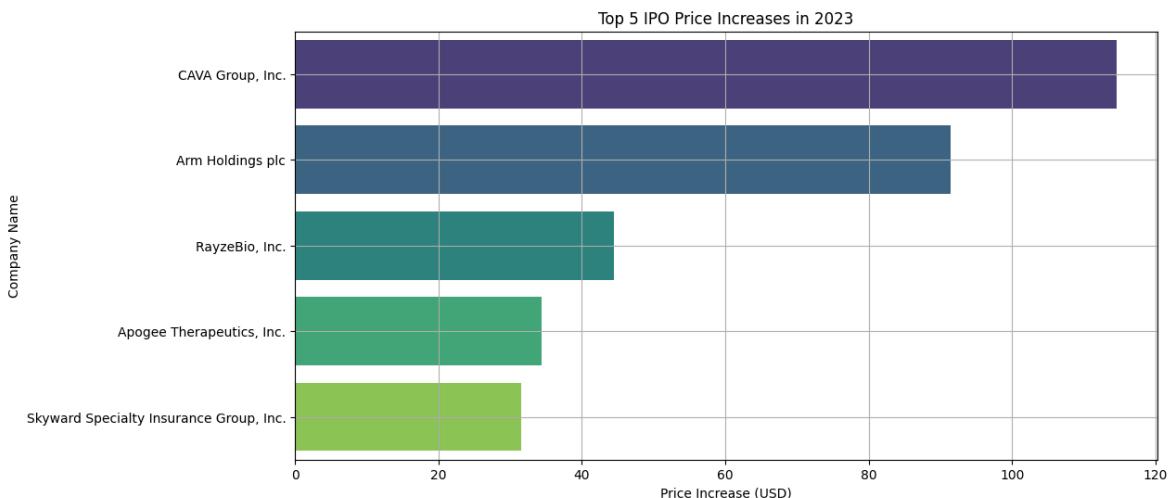
C:\Users\USER\AppData\Local\Temp\ipykernel\_15076\1005678826.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```

sns.barplot(x='Price Increase', y='Company Name', data=top_5_ipos, palette='viridis')

```



```
In [20]: # Truncate to the first day in the month - for Bar names
ipo_data_2023['Date_monthly'] = ipo_data_2023['IPO Date'].dt.to_period('M').dt.t
ipo_data_2023.head(1)
```

```
Out[20]:
```

	IPO Date	Symbol	Company Name	IPO Price	Current	Return	Price Increase	Date_monthly
0	2023-12-27	IROH	Iron Horse Acquisitions Corp.	10.0	10.23	0.023	0.23	2023-12-01

```
In [21]: # Count the number of deals for each month and year
monthly_deals = ipo_data_2023['Date_monthly'].value_counts().reset_index().sort_
monthly_deals.columns = ['Date_monthly', 'Number of Deals']

monthly_deals.head(1)
```

```
Out[21]:
```

	Date_monthly	Number of Deals
10	2023-01-01	8

```
In [22]: # some visualisation: bar chart using Plotly Express
import plotly.express as px

# Plotting the bar chart using Plotly Express
fig = px.bar(monthly_deals,
              x='Date_monthly',
              y='Number of Deals',
              labels={'Month_Year': 'Month and Year', 'Number of Deals': 'Number of IPO Deals per Month and Year',
                      'text': 'Number of Deals'},
              )
fig.update_traces(textposition='outside', # Position the text outside the bars
                  textfont=dict(color='black', size=14), # Adjust the font size of the text
                  )
fig.update_layout(title_x=0.5) # Center the title
fig.show()
```

## after IPO analysis, now continue analysis with top 5 tech companies

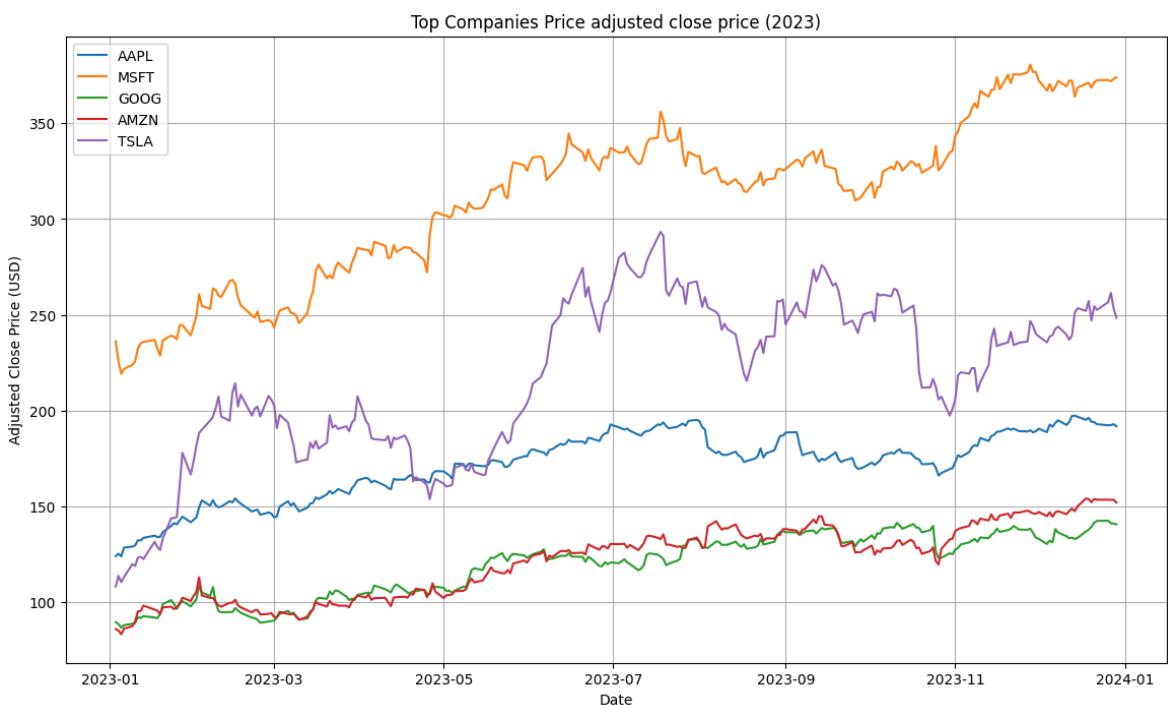
```
In [23]: import matplotlib.pyplot as plt

# Plotting each company's adjusted close price and the S&P 500 (SPY)
plt.figure(figsize=(14, 8))

# Plot each top company's adjusted close price
for company in top_companies:
    # Filter combined dataset for each symbol
    company_data = combined_df_2023[combined_df_2023['Symbol'] == company]
    plt.plot(company_data['Date'], company_data['Adj Close'], label=company)

# Add titles and labels
plt.title('Top Companies Price adjusted close price (2023)')
plt.xlabel('Date')
plt.ylabel('Adjusted Close Price (USD)')
plt.legend() # Add Legend to distinguish between companies and SPY
plt.grid(True)

# Show the plot
plt.show()
```



```
In [24]: import matplotlib.pyplot as plt

fig, ax1 = plt.subplots(figsize=(14, 8))

# Plot each company's adjusted close price on the left y-axis
for company in top_companies:
    company_data = combined_df_2023[combined_df_2023['Symbol'] == company]
    ax1.plot(company_data['Date'], company_data['Adj Close'], label=company)
ax1.set_xlabel('Date')
ax1.set_ylabel('Adjusted Close Price (USD)')
ax1.tick_params(axis='y')
```

```

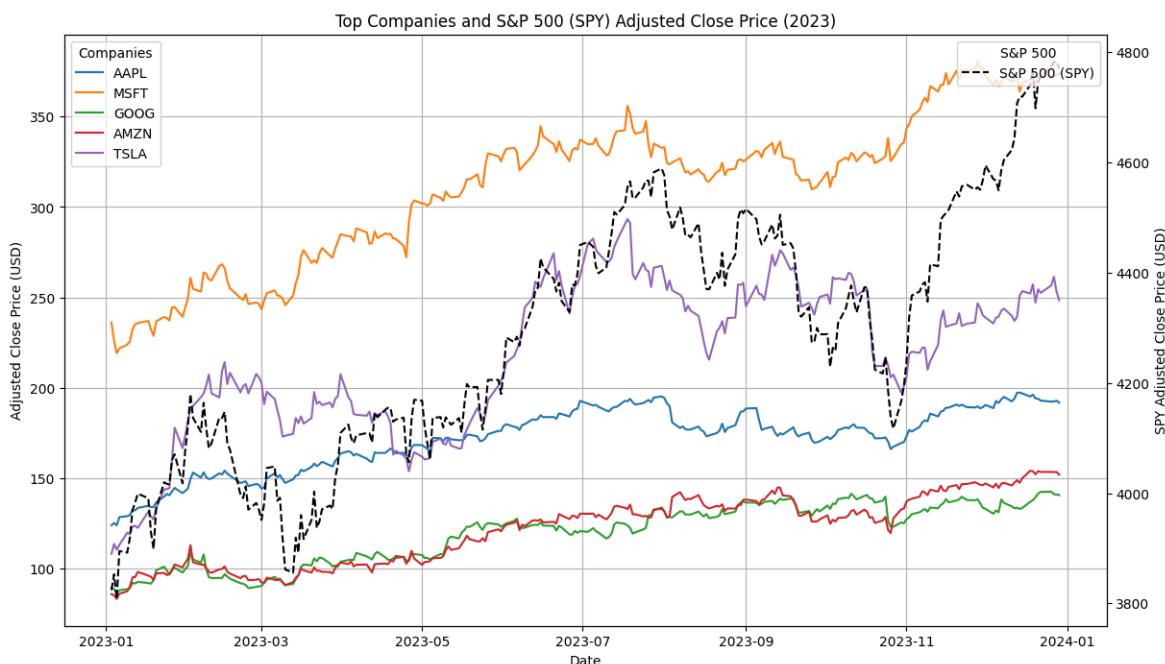
ax1.legend(title="Companies", loc='upper left')
ax1.grid(True)

# Create a second y-axis for SPY
ax2 = ax1.twinx()
ax2.plot(spy_data_2023.index, spy_data_2023['Adj Close'], label='S&P 500 (SPY)', color='orange')
ax2.set_ylabel('SPY Adjusted Close Price (USD)')
ax2.tick_params(axis='y')
ax2.legend(title="S&P 500", loc='upper right')

# Add title and rotate date labels for better visibility
plt.title('Top Companies and S&P 500 (SPY) Adjusted Close Price (2023)')
plt.xticks(rotation=45)

# Show the plot
plt.show()

```



In [25]: `combined_df_2023.head(2)`

Out[25]:

	Date	Open	High	Low	Close	Adj Close	Volume	Symb
0	2023-01-03	130.279999	130.899994	124.169998	125.070000	123.904625	112117500	AA
1	2023-01-04	126.889999	128.660004	125.080002	126.360001	125.182610	89113600	AA

In [26]: `# Set the 'Date' column as the index`  
`combined_df_2023.set_index('Date', inplace=True)`

```

# Extract year, month, and weekday from the datetime index
combined_df_2023['Year'] = combined_df_2023.index.year
combined_df_2023['Month'] = combined_df_2023.index.month
combined_df_2023['Weekday'] = combined_df_2023.index.weekday

# View the first few rows of combined_df_2023
combined_df_2023.head(2)

```

Out[26]:

	Open	High	Low	Close	Adj Close	Volume	Symbol
Date							
2023-01-03	130.279999	130.899994	124.169998	125.070000	123.904625	112117500	AAPL
2023-01-04	126.889999	128.660004	125.080002	126.360001	125.182610	89113600	AAPL

In [27]:

```
# shift ALL values (on period forward (+1) and backward (-1))
# equivalent of joining with a dataframe of the same vector, but with shifted data
combined_df_2023['adj_close_minus_1'] = combined_df_2023['Adj Close'].shift(-1)
combined_df_2023['adj_close_plus_1'] = combined_df_2023['Adj Close'].shift(1)
combined_df_2023.head(2)
```

Out[27]:

	Open	High	Low	Close	Adj Close	Volume	Symbol
Date							
2023-01-03	130.279999	130.899994	124.169998	125.070000	123.904625	112117500	AAPL
2023-01-04	126.889999	128.660004	125.080002	126.360001	125.182610	89113600	AAPL

- adj\_close\_minus\_1: This column shows the adjusted closing price of the stock for the next day. It helps in analyzing what the price will be on the following trading day compared to today.
- adj\_close\_plus\_1: This column shows the adjusted closing price of the stock for the previous day. It is useful for comparing today's price to the price from the day before.

In [28]:

```
# historical growth
combined_df_2023['growth_1d'] = combined_df_2023['Adj Close'] / combined_df_2023
combined_df_2023['growth_30d'] = combined_df_2023['Adj Close'] / combined_df_2023

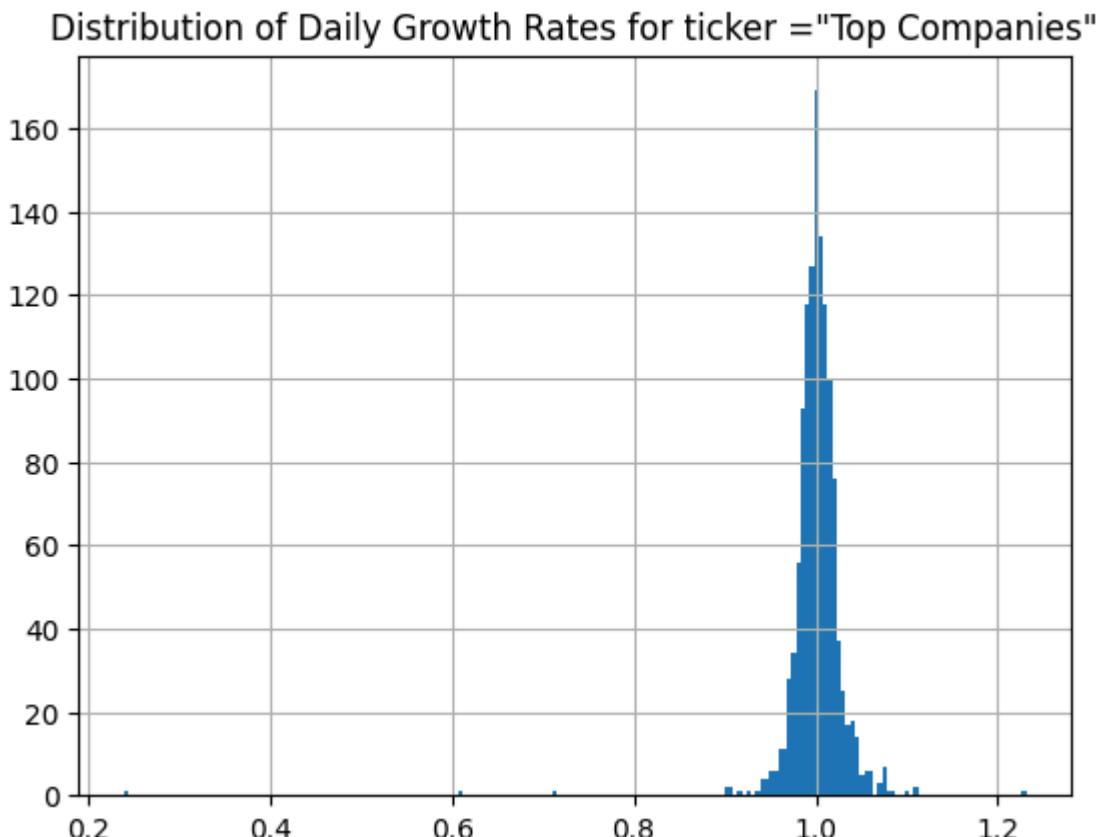
combined_df_2023.head(2)
```

Out[28]:

	Open	High	Low	Close	Adj Close	Volume	Symbol
Date							
2023-01-03	130.279999	130.899994	124.169998	125.070000	123.904625	112117500	AAPL
2023-01-04	126.889999	128.660004	125.080002	126.360001	125.182610	89113600	AAPL

- growth\_1d: This column shows the daily growth of the stock price. It calculates how much the stock price changed compared to the previous day. It is a ratio that helps to see if the stock went up or down from one day to the next.
- growth\_30d: This column shows the monthly growth of the stock price, comparing the current price to the price 30 days ago. It helps in understanding how the stock price has changed over the past month.

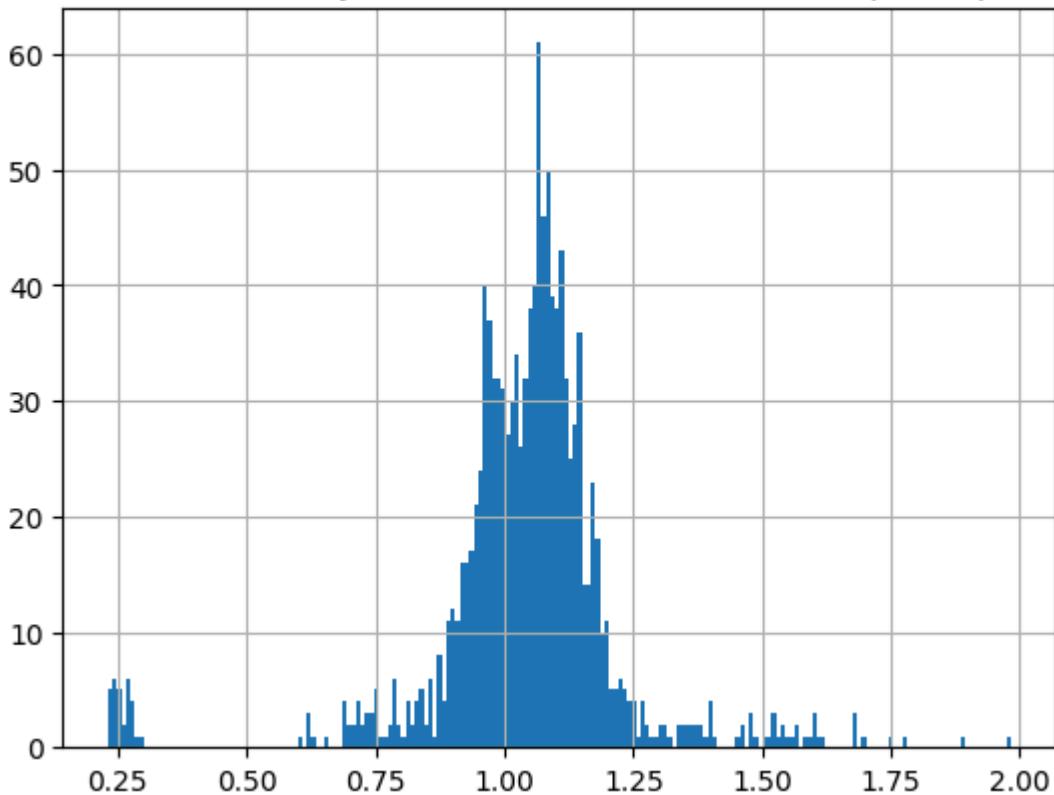
```
In [29]: plt.title('Distribution of Daily Growth Rates for ticker ="Top Companies"')
combined_df_2023.growth_1d.hist(bins=200)
plt.show()
```



```
In [30]: # you can see that the growth is can be much wider in 30 days (+- 30%)
plt.title('Distribution of 2-days Growth Rates for ticker = "Top Companies"')
combined_df_2023.growth_30d.hist(bins=200)
```

```
Out[30]: <Axes: title={'center': 'Distribution of 2-days Growth Rates for ticker = "Top Companies"'>
```

## Distribution of 2-days Growth Rates for ticker = "Top Companies"



the growth in 1 day is +-10%, while a lot of it is around 0% (around 1)

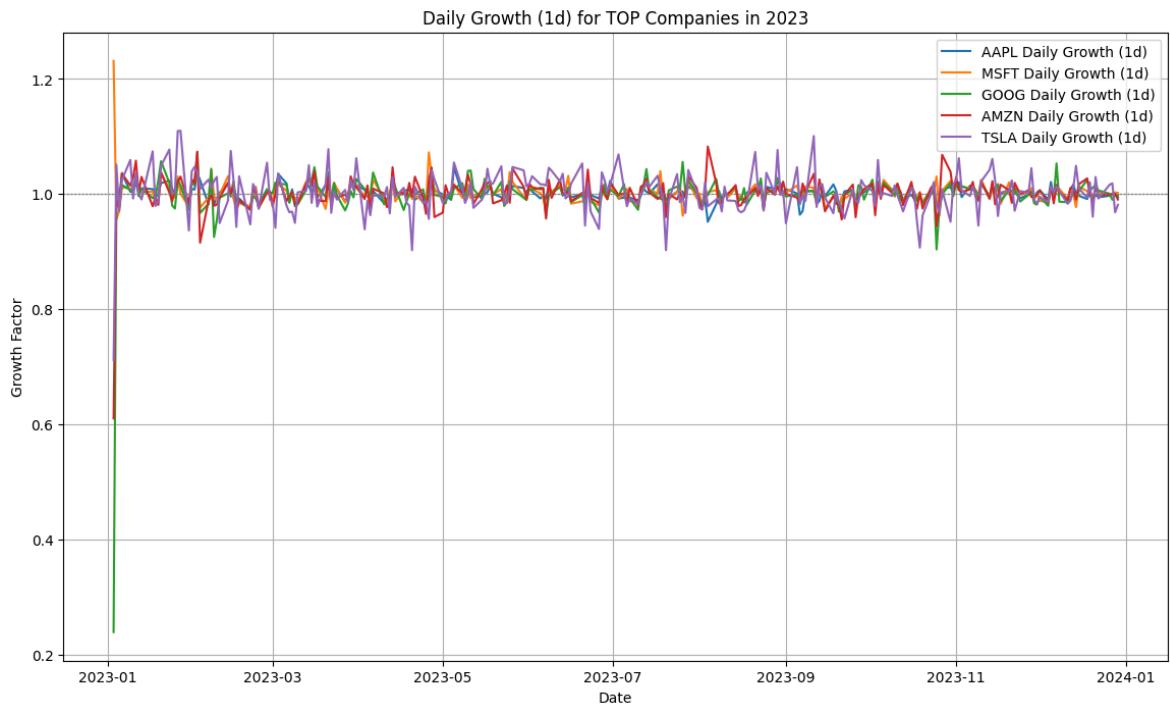
```
In [31]: import matplotlib.pyplot as plt

plt.figure(figsize=(14, 8))

# Plot daily growth for each company
for company in top_companies:
    company_data = combined_df_2023[combined_df_2023['Symbol'] == company]
    plt.plot(company_data.index, company_data['growth_1d'], label=f'{company} Daily Growth')

# Add titles and labels
plt.title('Daily Growth (1d) for TOP Companies in 2023')
plt.xlabel('Date')
plt.ylabel('Growth Factor')
plt.axhline(1, color='gray', linestyle=':', linewidth=1) # Reference Line for growth
plt.legend() # Add Legend to distinguish between companies
plt.grid(True)

# Show the plot
plt.show()
```

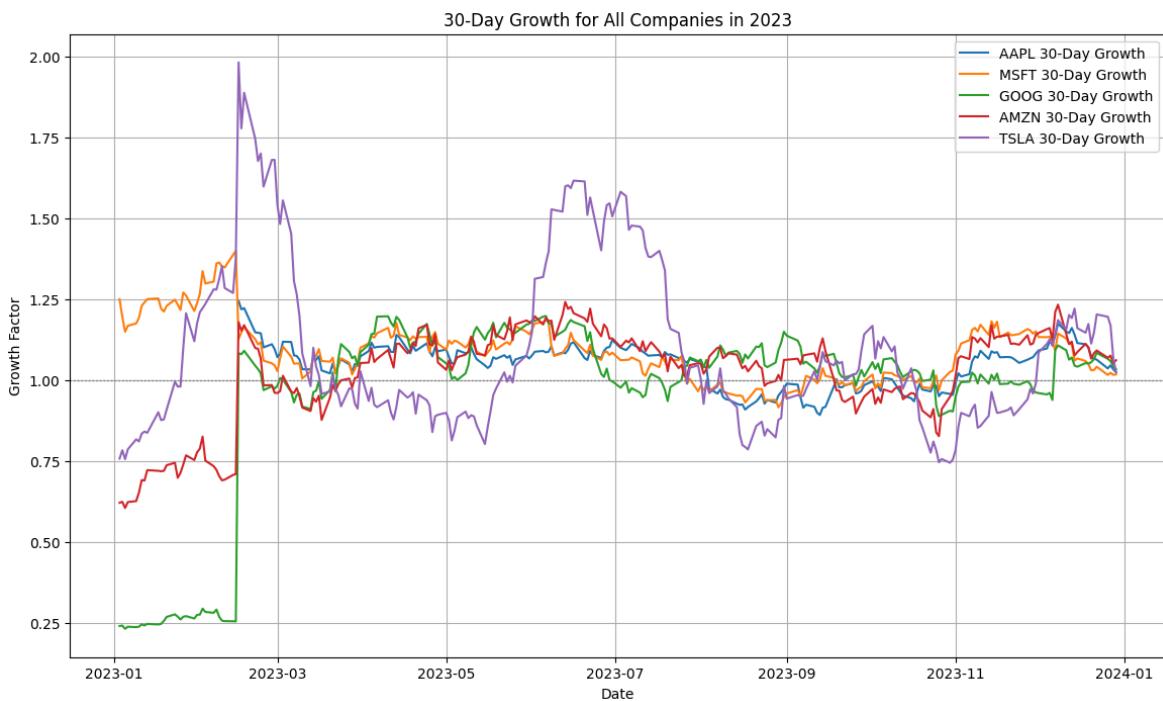


```
In [32]: plt.figure(figsize=(14, 8))

# Plot 30-day growth for each company
for company in top_companies:
    company_data = combined_df_2023[combined_df_2023['Symbol'] == company]
    plt.plot(company_data.index, company_data['growth_30d'], label=f'{company} 30d')

# Add titles and labels
plt.title('30-Day Growth for All Companies in 2023')
plt.xlabel('Date')
plt.ylabel('Growth Factor')
plt.axhline(1, color='gray', linestyle=':', linewidth=1) # Reference line for growth
plt.legend() # Add legend to distinguish between companies
plt.grid(True)

# Show the plot
plt.show()
```



```
In [33]: spy_data_2023['growth_1d'] = spy_data_2023['Adj Close'] / spy_data_2023['Adj Cl
spy_data_2023['growth_30d'] = spy_data_2023['Adj Close'] / spy_data_2023['Adj Cl
spy_data_2023.head(2)
```

Out[33]:

Date	Open	High	Low	Close	Adj Close	Volume	g
2023-01-03	3853.290039	3878.459961	3794.330078	3824.139893	3824.139893	3959140000	
2023-01-04	3840.360107	3873.159912	3815.770020	3852.969971	3852.969971	4414080000	

```
In [34]: import matplotlib.pyplot as plt

# Plot 1: SPY Growth Metrics (Daily and 30-Day Growth)
plt.figure(figsize=(14, 8))

# Plot daily growth
plt.plot(spy_data_2023.index, spy_data_2023['growth_1d'], label='SPY Daily Growth')

# Plot 30-day growth
plt.plot(spy_data_2023.index, spy_data_2023['growth_30d'], label='SPY 30-Day Growth')

# Add titles and labels
plt.title('SPY Growth Metrics (Daily and 30-Day Growth) for 2023')
plt.xlabel('Date')
plt.ylabel('Growth Factor')
plt.axhline(1, color='gray', linestyle=':', linewidth=1) # Reference Line for g
plt.legend() # Add legend to distinguish between metrics
plt.grid(True)

# Show the first plot
plt.show()
```

```

# Plot 2: Open and Close Prices for Each Company
for company in top_companies:
    plt.figure(figsize=(14, 8))

    # Filter data for the current company
    company_data = combined_df_2023[combined_df_2023['Symbol'] == company]

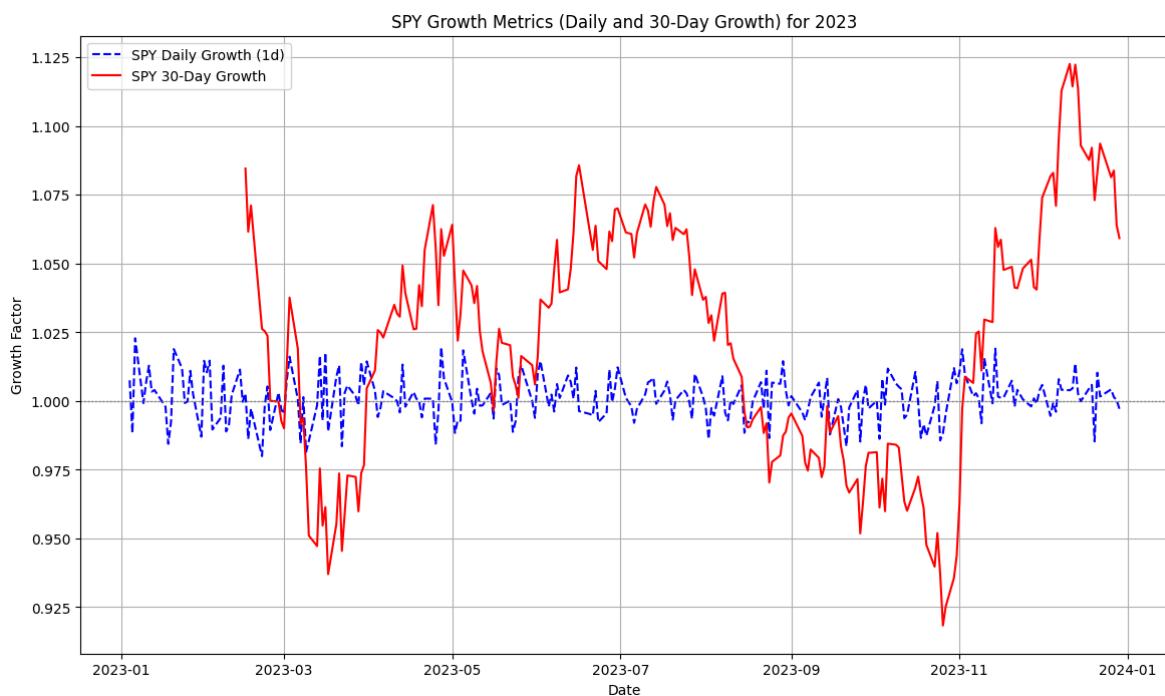
    # Plot open prices
    plt.plot(company_data.index, company_data['Open'], label=f'{company} Open Price')

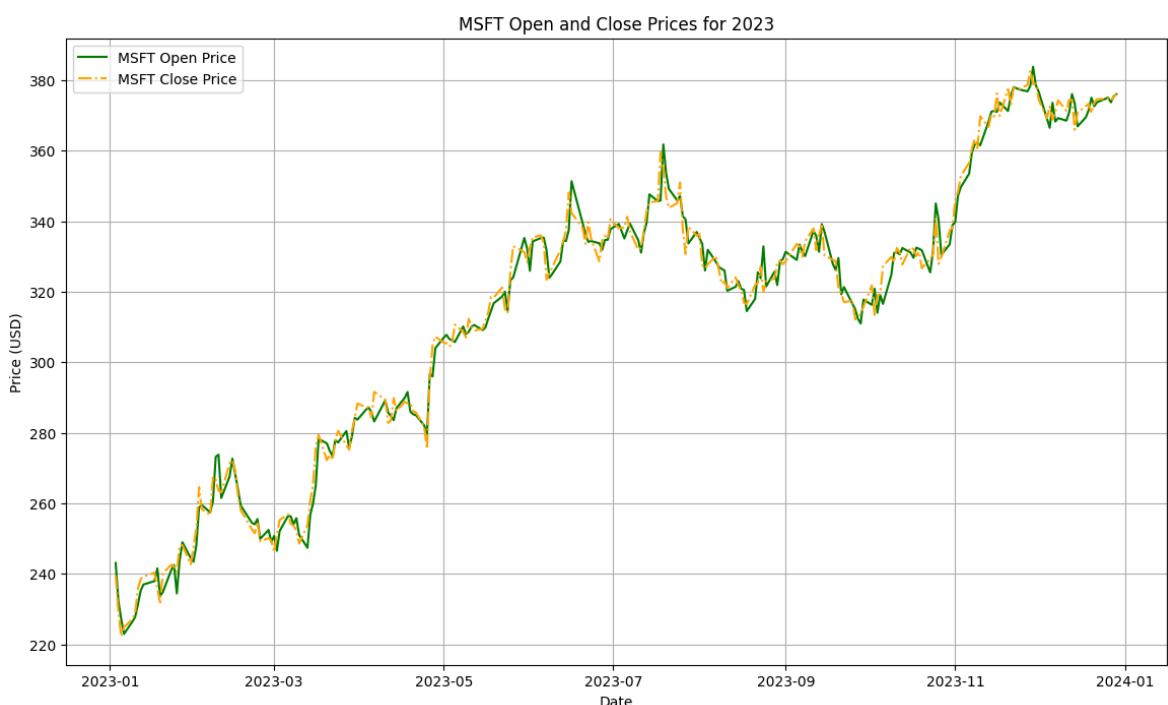
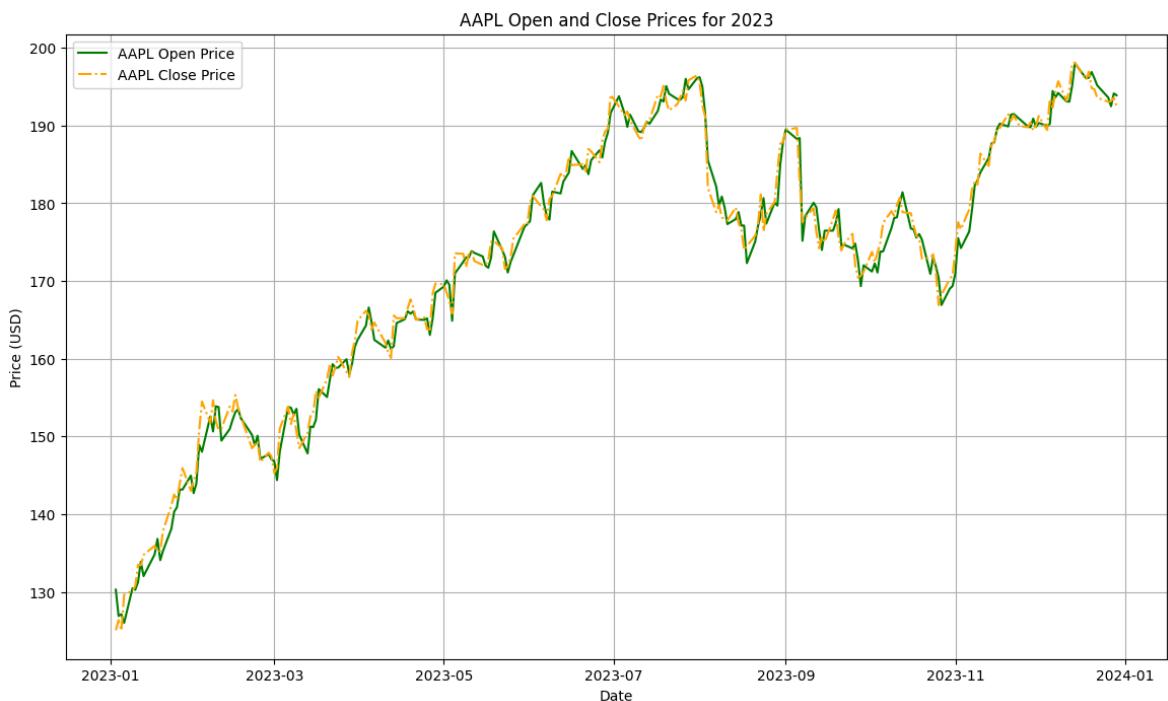
    # Plot close prices
    plt.plot(company_data.index, company_data['Close'], label=f'{company} Close Price')

    # Add titles and labels
    plt.title(f'{company} Open and Close Prices for 2023')
    plt.xlabel('Date')
    plt.ylabel('Price (USD)')
    plt.legend() # Add Legend to distinguish between Open and Close prices
    plt.grid(True)

    # Show the plot for each company
    plt.show()

```





## GOOG Open and Close Prices for 2023



## AMZN Open and Close Prices for 2023



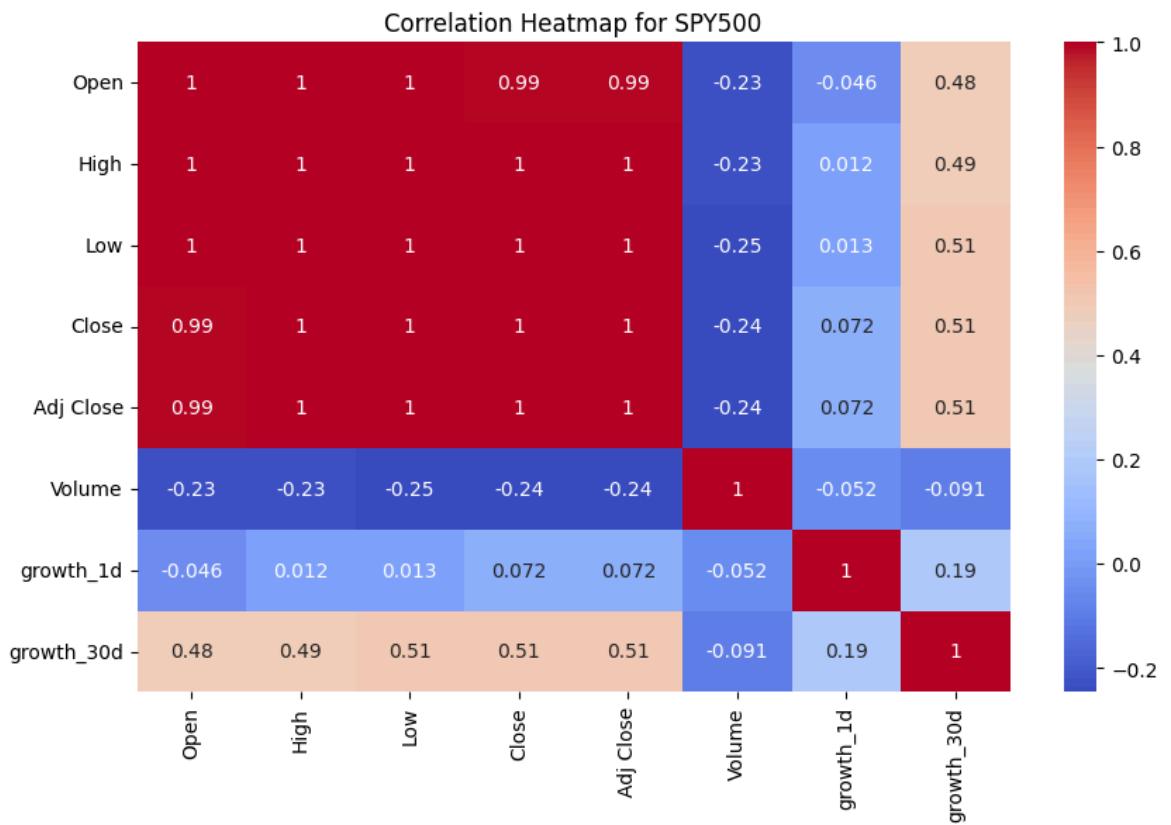


## Correlation Analysis

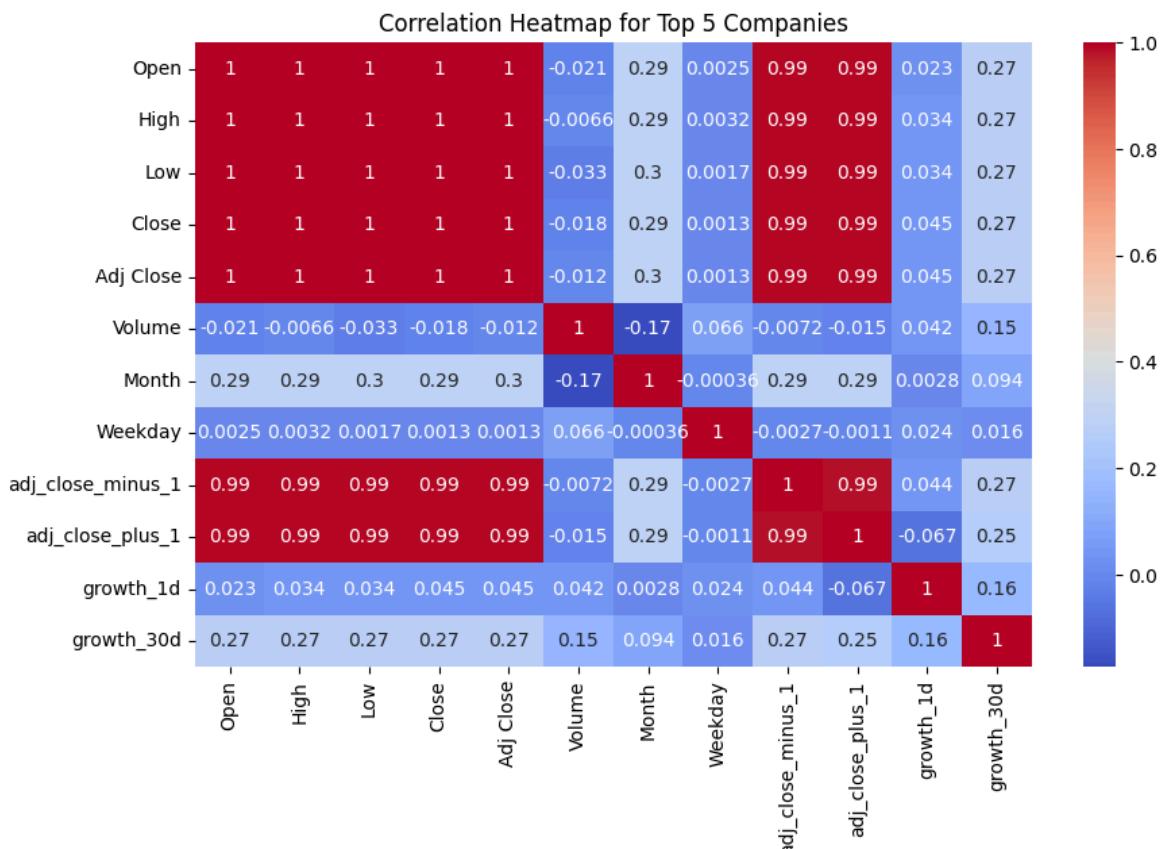
In this step, we will analyze the correlation between different features within the dataset

Open, High, Low, and Close prices are highly correlated (values close to 1). This is typical, as these metrics are derived from the same trading data.

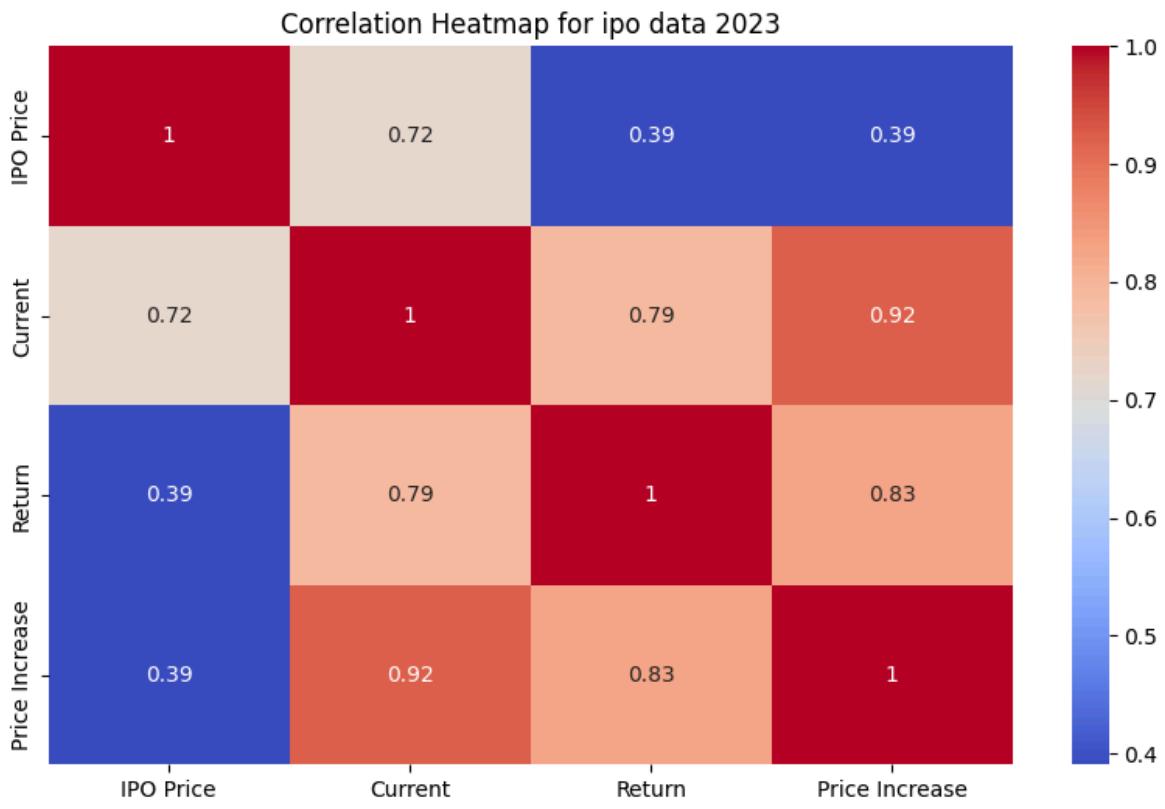
```
In [35]: plt.figure(figsize=(10, 6))
numeric_data = spy_data_2023.select_dtypes(include=[np.number]) # Select only numeric columns
numeric_data = numeric_data.drop(columns=['Year'], errors='ignore') # Exclude 'Year' column
sns.heatmap(numeric_data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap for SPY500')
plt.show()
```



```
In [36]: plt.figure(figsize=(10, 6))
numeric_data = combined_df_2023.select_dtypes(include=[np.number]) # Select only numeric data
numeric_data = numeric_data.drop(columns=['Year'], errors='ignore') # Exclude 'Year' column
sns.heatmap(numeric_data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap for Top 5 Companies')
plt.show()
```



```
In [37]: plt.figure(figsize=(10, 6))
numeric_data = ipo_data_2023.select_dtypes(include=[np.number]) # Select only numeric data
sns.heatmap(numeric_data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap for ipo data 2023')
plt.show()
```



The Column Metrics that we would add:

SMA\_50: 50-day Simple Moving Average of the closing price.

EMA\_20: 20-day Exponential Moving Average of the closing price.

RSI: Relative Strength Index used to measure the magnitude of recent price changes.

```
In [38]: combined_df_2023.head(1)
```

	Open	High	Low	Close	Adj Close	Volume	Symbol	Year
Date								
2023-01-03	130.279999	130.899994	124.169998	125.07	123.904625	112117500	AAPL	2023

```
In [39]: # Adding Technical Indicators
```

```
# Copy the combined data to avoid modifying the original directly
data = combined_df_2023.copy()

# Simple Moving Average (SMA) with a window of 50 days
data['SMA_50'] = data.groupby('Symbol')['Close'].transform(lambda x: x.rolling(windows=50, min_periods=1).mean())

# Exponential Moving Average (EMA) with a span of 20 days
data['EMA_20'] = data.groupby('Symbol')['Close'].transform(lambda x: x.ewm(span=20, min_periods=1).mean())

# Relative Strength Index (RSI) with a window of 14 days
data['RSI'] = data.groupby('Symbol')['Close'].transform(lambda x: 100 - (100 / (1 + ((x - x.rolling(14).mean()) / x.rolling(14).std()))))
```

```

data['EMA_20'] = data.groupby('Symbol')['Close'].transform(lambda x: x.ewm(span=20).mean())

# Update combined_df_2023 with the new columns
combined_df_2023 = data

# Display the updated DataFrame
print(combined_df_2023.head())

```

Date	Open	High	Low	Close	Adj Close	\
2023-01-03	130.279999	130.899994	124.169998	125.070000	123.904625	
2023-01-04	126.889999	128.660004	125.080002	126.360001	125.182610	
2023-01-05	127.129997	127.769997	124.760002	125.019997	123.855103	
2023-01-06	126.010002	130.289993	124.889999	129.619995	128.412231	
2023-01-09	130.470001	133.410004	129.889999	130.149994	128.937286	

Date	Volume	Symbol	Year	Month	Weekday	adj_close_minus_1	\
2023-01-03	112117500	AAPL	2023	1	1	125.182610	
2023-01-04	89113600	AAPL	2023	1	2	123.855103	
2023-01-05	80962700	AAPL	2023	1	3	128.412231	
2023-01-06	87754700	AAPL	2023	1	4	128.937286	
2023-01-09	70790800	AAPL	2023	1	0	129.511887	

Date	adj_close_plus_1	growth_1d	growth_30d	SMA_50	EMA_20
2023-01-03	NaN	NaN	NaN	NaN	125.070000
2023-01-04	123.904625	1.010314	NaN	NaN	125.192857
2023-01-05	125.182610	0.989395	NaN	NaN	125.176394
2023-01-06	123.855103	1.036794	NaN	NaN	125.599594
2023-01-09	128.412231	1.004089	NaN	NaN	126.032966

In [40]:

```

import matplotlib.pyplot as plt

# Plotting Close Price, SMA_50, and EMA_20 for Each Company
for company in top_companies:
    plt.figure(figsize=(14, 8))

    # Filter data for the current company
    company_data = combined_df_2023[combined_df_2023['Symbol'] == company]

    # Plot Close Price
    plt.plot(company_data.index, company_data['Close'], label=f'{company} Close')

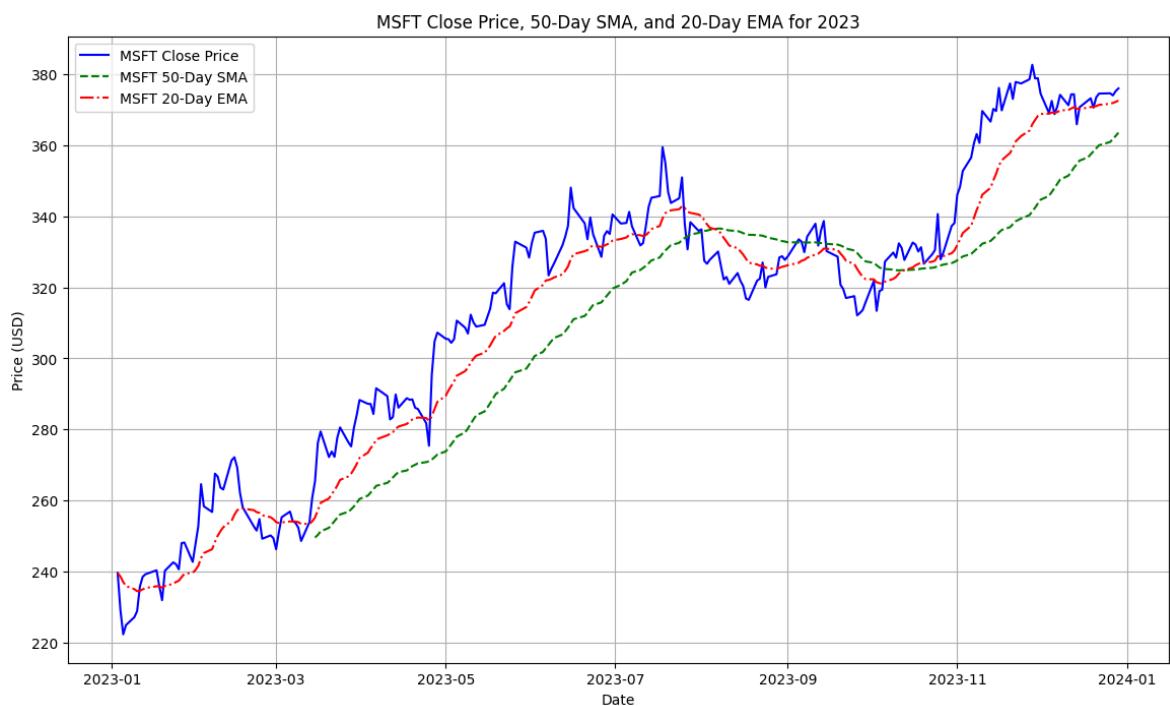
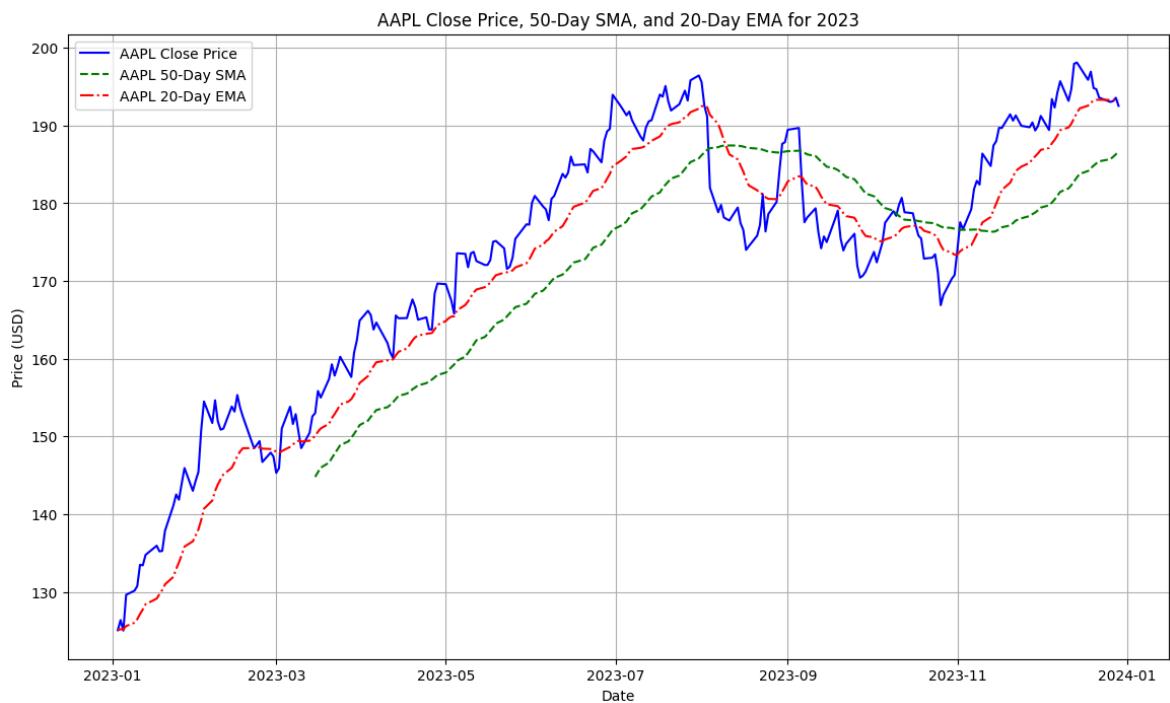
    # Plot 50-Day Simple Moving Average (SMA)
    plt.plot(company_data.index, company_data['SMA_50'], label=f'{company} 50-Day SMA')

    # Plot 20-Day Exponential Moving Average (EMA)
    plt.plot(company_data.index, company_data['EMA_20'], label=f'{company} 20-Day EMA')

    # Add titles and labels
    plt.title(f'{company} Close Price, 50-Day SMA, and 20-Day EMA for 2023')
    plt.xlabel('Date')
    plt.ylabel('Price (USD)')
    plt.legend() # Add legend to distinguish between Close, SMA, and EMA
    plt.grid(True)

    # Show the plot for each company
    plt.show()

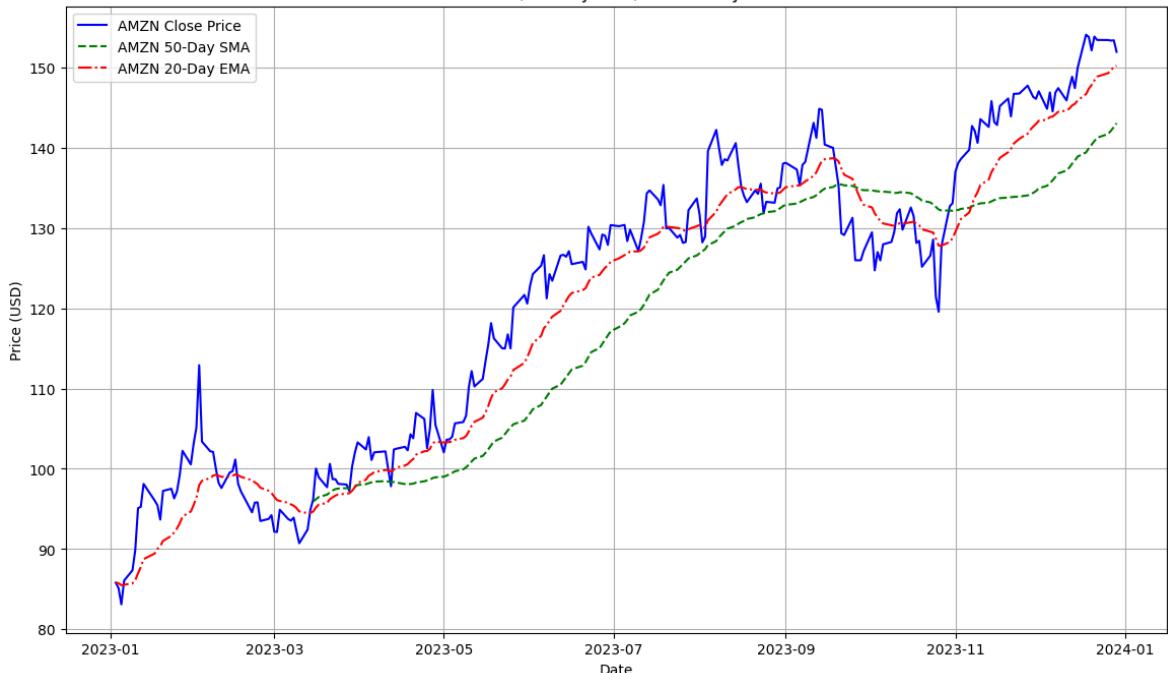
```



GOOG Close Price, 50-Day SMA, and 20-Day EMA for 2023



AMZN Close Price, 50-Day SMA, and 20-Day EMA for 2023





The plot shows the moving averages (SMA and EMA) for AAPL, which helps in understanding the stock trends and potential buy/sell signals.

## Create a new DataFrame for Tesla

since Tesla has the most change patterns during the year-2023

```
In [41]: # Filter combined_df_2023 to create a new DataFrame containing only Tesla data
tesla_df = combined_df_2023[combined_df_2023['Symbol'] == 'TSLA'].reset_index()

# Display the first few rows of tesla_df to verify
print(tesla_df.head())
```

	Open	High	Low	Close	Adj Close	Volume	\
0	118.470001	118.800003	104.639999	108.099998	108.099998	231402800	
1	109.110001	114.589996	107.519997	113.639999	113.639999	180389000	
2	110.510002	111.750000	107.160004	110.339996	110.339996	157986300	
3	103.000000	114.389999	101.809998	113.059998	113.059998	220911100	
4	118.959999	123.519997	117.110001	119.769997	119.769997	190284000	

	Symbol	Year	Month	Weekday	adj_close_minus_1	adj_close_plus_1	\
0	TSLA	2023	1	1	113.639999	151.940002	
1	TSLA	2023	1	2	110.339996	108.099998	
2	TSLA	2023	1	3	113.059998	113.639999	
3	TSLA	2023	1	4	119.769997	110.339996	
4	TSLA	2023	1	0	118.849998	113.059998	

	growth_1d	growth_30d	SMA_50	EMA_20
0	0.711465	0.756844	NaN	108.099998
1	1.051249	0.782752	NaN	108.627618
2	0.970961	0.755081	NaN	108.790701
3	1.024651	0.785685	NaN	109.197301
4	1.059349	0.816372	NaN	110.204224

```
In [42]: tesla_close = pd.pivot_table(tesla_df, values = "Close", columns = "Month", index
```

```
tesla_close.head()
```

```
Out[42]: Month      1      2      3      4      5      6      7
Year
2023  134.195999  199.775264  188.522609  176.955262  175.79909  246.090476  273.5055
```

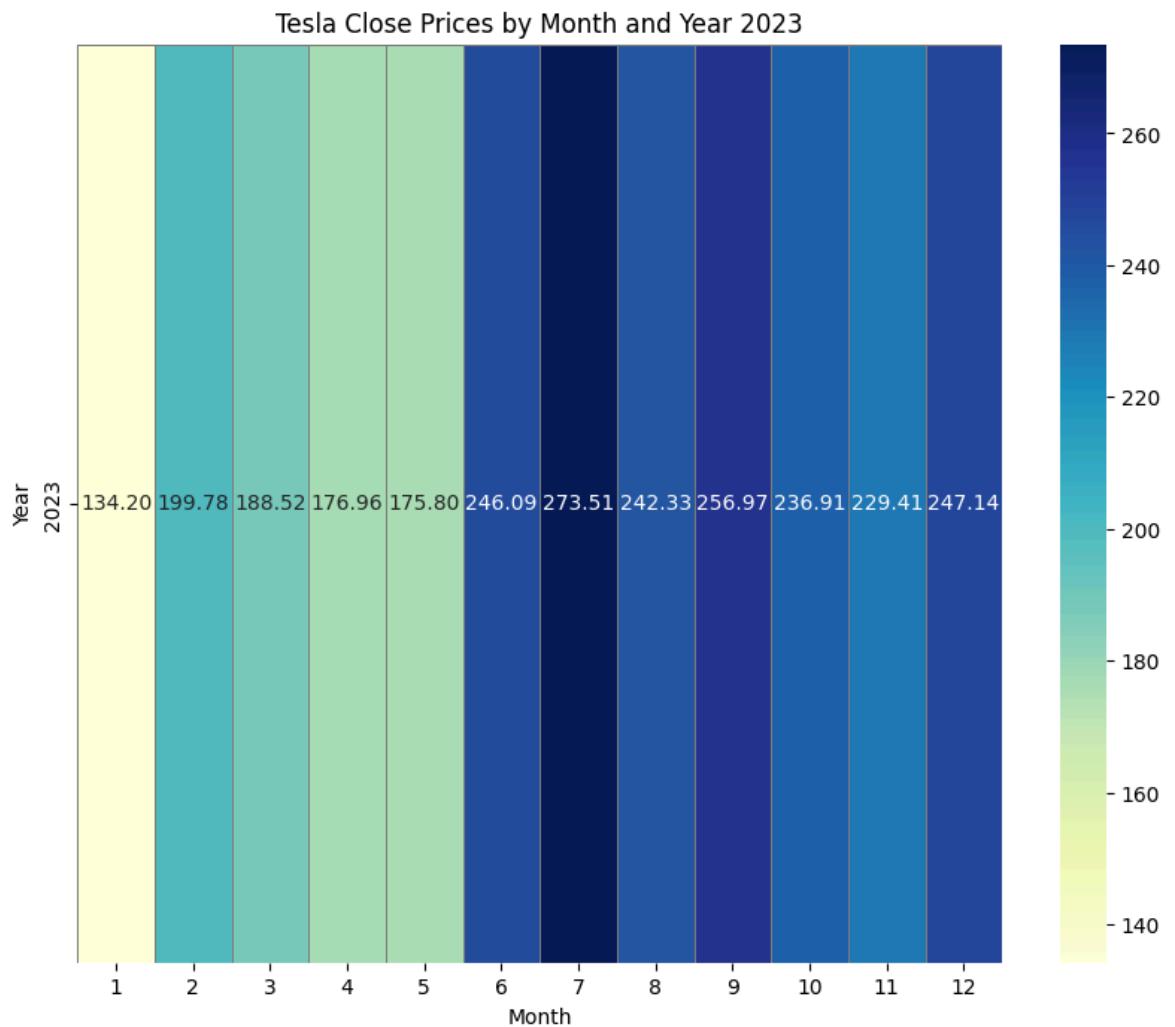
```
In [43]: import seaborn as sns
import matplotlib.pyplot as plt

# Create the pivot table
tesla_close = pd.pivot_table(tesla_df, values="Close", columns="Month", index="Year")

# Plotting the heatmap of Tesla's closing prices by month and year
plt.figure(figsize=(10, 8))
sns.heatmap(tesla_close, annot=True, fmt=".2f", cmap="YlGnBu", linewidths=0.5, linecolor="white")

# Add titles and labels
plt.title("Tesla Close Prices by Month and Year 2023")
plt.xlabel("Month")
plt.ylabel("Year")

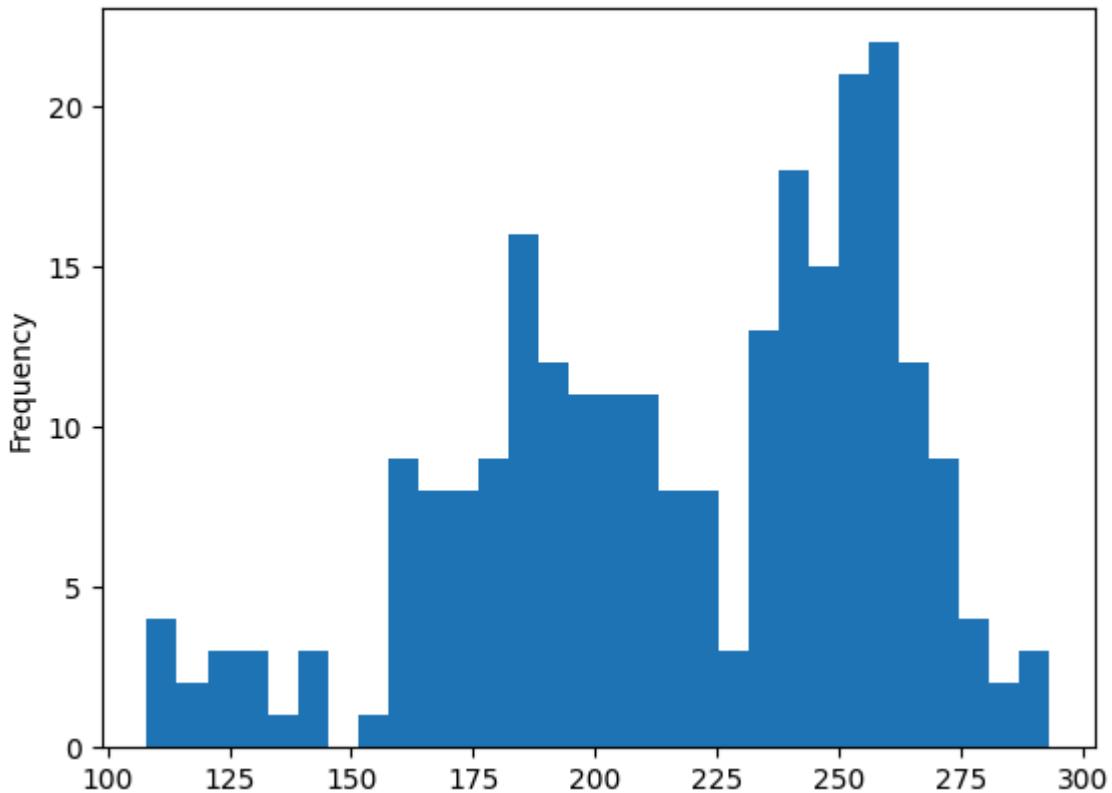
# Show the plot
plt.show()
```



The highest price for Tesla was at period mid of June to mid of July and the lowest price was on January

```
In [44]: tesla_df.Close.plot(kind = "hist", bins = 30)
```

```
Out[44]: <Axes: ylabel='Frequency'>
```



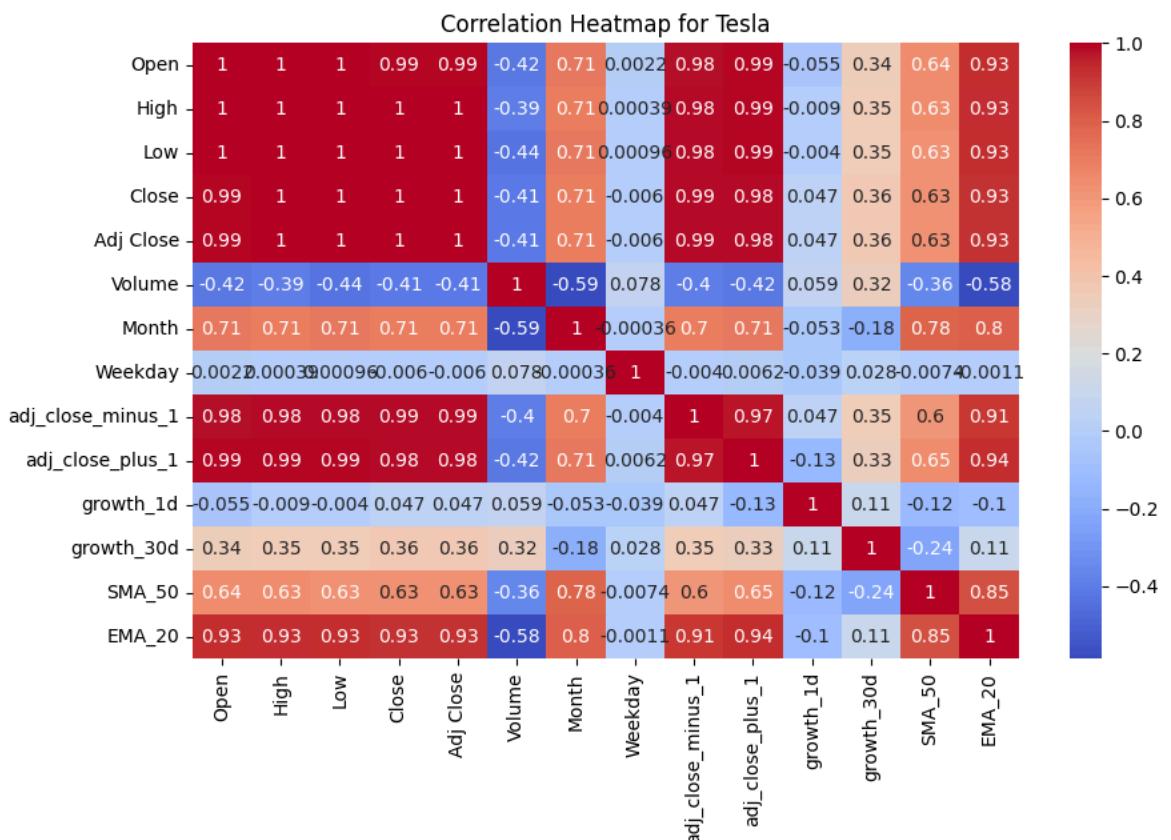
The basic statistics provide an overview of the data distribution, including mean, min, max, and standard deviation for each feature.

The line plot shows the trend of closing prices for each company throughout 2023, allowing us to visually compare performance.

The missing values summary helps identify any gaps in the data that need to be addressed for further analysis.

```
In [45]: # Correlation heatmap for Tesla
```

```
plt.figure(figsize=(10, 6))
numeric_data = tesla_df.select_dtypes(include=[np.number]) # Select only numeric
numeric_data = numeric_data.drop(columns=['Year'], errors='ignore') # Exclude 'Year'
sns.heatmap(numeric_data.corr(), annot=True, cmap='coolwarm')
plt.title(f'Correlation Heatmap for Tesla')
plt.show()
```



## Sentiment Analysis on Tesla 2023

In this step, we will collect and analyze sentiment data related to the selected companies. We will use news articles to extract sentiment scores for each company.

```
In [46]: import pandas as pd

# Load the CSV file from your provided path
file_path = r'D:/Download/Tesla_newssent_extraction.csv'
tesla_sent_2023 = pd.read_csv(file_path)

# Select relevant columns
tesla_sent_2023 = tesla_sent_2023[['request_id', 'companyname', 'url', 'title',
                                      'company_relevance', 'environment_sentiment_score',
                                      'environment_sentiment_reasoning', 'social_sentiment_score',
                                      'social_sentiment_reasoning', 'governance_sentiment_score',
                                      'governance_sentiment_reasoning']]

# Display the filtered DataFrame with the selected columns
print(tesla_sent_2023)
```

```

request_id  companyname  \
0           52385  Tesla, Inc.
1           52385  Tesla, Inc.
2           52386  Tesla, Inc.
3           52386  Tesla, Inc.
4           52386  Tesla, Inc.
...
1307        ...      ...
1308        52566  Tesla, Inc.
1308        52566  Tesla, Inc.
1309        52566  Tesla, Inc.
1310        52566  Tesla, Inc.
1311        52566  Tesla, Inc.

url  \
0           https://t.co/Fkdopyku9m
1           https://t.co/ZSdYvKjjHo
2           https://www.reuters.com/business/autos-transpo...
3           https://www.theguardian.com/us-news/2023/feb/2...
4           https://www.reuters.com/lifestyle/science/spac...
...
1307        ...      ...
1308        https://www.cnbc.com/2023/08/25/tesla-autopilo...
1308        https://seekingalpha.com/article/4630877-tesla...
1309        https://www.reuters.com/article/tesla-india-po...
1310        https://seekingalpha.com/news/4006311-teslas-s...
1311        https://www.reuters.com/business/autos-transpo...

title      date  \
0           Yahoo Finance 2023-02-25
1           Yahoo Finance 2023-02-25
2           Analysis: CATL's 'price war' shows its power, ... 2023-02-26
3           Old-school union busting: how US corporations ... 2023-02-26
4           SpaceX set to launch next International Space ... 2023-02-26
...
1307        ...      ...
1307        Tesla Autopilot safety probe by federal vehicel... 2023-08-25
1308        Tesla Stock: Undervalued With Cybertruck Deliv... 2023-08-25
1309        EXCLUSIVE-With Tesla push, India mulls import ... 2023-08-25
1310        Tesla's Supercharger business is seen growing ... 2023-08-25
1311        Exclusive: With Tesla push, India mulls import... 2023-08-25

company_relevance  environment_sentiment_score  \
0           100          0.0
1           80           0.0
2           100          0.2
3           100          0.0
4           80           0.0
...
1307        ...      ...
1307        100          0.1
1308        100          0.2
1309        80           0.2
1310        100          0.2
1311        100          0.2

environment_sentiment_reasoning  \
0           The article does not mention any specific info...
1           The article does not mention any specific envi...
2           The article mentions that CATL, the world's la...
3           The article does not mention any specific envi...
4           The article does not mention any specific envi...
...
1307        ...      ...
1307        The article does not provide a significant amo...

```

```
1308 The article does not provide specific informat...
1309 The article mentions that India is considering...
1310 The article mentions Tesla's supercharger netw...
1311 The article mentions that India is considering...
```

```
social_sentiment_score \
0 0.0
1 -0.5
2 0.1
3 -0.5
4 0.0
...
1307 0.2
1308 0.5
1309 0.1
1310 0.3
1311 0.1
```

```
social_sentiment_reasoning \
0 The article does not provide any information a...
1 The article mentions Gerber leaving Tesla's bo...
2 The article mentions that CATL has faced pushb...
3 The article mentions Tesla firing dozens of wo...
4 The article does not provide any information a...
...
1307 The article briefly mentions Tesla CEO Elon Mu...
1308 The article mentions Tesla's ability to introd...
1309 The article mentions that India is considering...
1310 The article highlights Tesla's strategic move ...
1311 The article mentions that the proposed policy ...
```

```
governance_sentiment_score \
0 0.0
1 -0.5
2 0.1
3 -0.5
4 0.0
...
1307 0.1
1308 0.5
1309 0.1
1310 0.4
1311 0.2
```

```
governance_sentiment_reasoning
0 The article does not discuss Tesla's governanc...
1 The article mentions Gerber leaving Tesla's bo...
2 The article mentions that CATL has faced pushb...
3 The article mentions Tesla firing workers in r...
4 The article does not provide any information a...
...
1307 The article does not provide significant infor...
1308 The article mentions Elon Musk's behavior caus...
1309 The article mentions that Tesla has been in ta...
1310 The article mentions Tesla's long-term strateg...
1311 The article mentions that the Indian governmen...
```

[1312 rows x 12 columns]

In [47]: `tesla_sent_2023.tail(1)`

Out[47]:

	request_id	companyname	url	title	date
1311	52566	Tesla, Inc.	https://www.reuters.com/business/autos-transpo...	Exclusive: With Tesla push, India mulls import...	2023-08-25

In [48]: `tesla_sent_2023.tail()`

Out[48]:

	request_id	companyname	url	title
1307	52566	Tesla, Inc.	https://www.cnbc.com/2023/08/25/tesla-autopilo...	Tesla Autopilot safety probe by federal vehicl...
1308	52566	Tesla, Inc.	https://seekingalpha.com/article/4630877-tesla...	Tesla Stock: Undervalued With Cybertruck Deliv...
1309	52566	Tesla, Inc.	https://www.reuters.com/article/tesla-india-po...	EXCLUSIVE- With Tesla push, India mulls import ...
1310	52566	Tesla, Inc.	https://seekingalpha.com/news/4006311-teslas-s...	Tesla's Supercharger business is seen growing ...
1311	52566	Tesla, Inc.	https://www.reuters.com/business/autos-transpo...	Exclusive: With Tesla push, India mulls import...

Calculate the average sentiment scores for each category (environmental, social, and governance) across all Tesla articles to get an overview of the sentiment trends related to Tesla

In [49]:

```
# Calculate average sentiment scores
avg_environment_sentiment = tesla_sent_2023['environment_sentiment_score'].mean()
avg_social_sentiment = tesla_sent_2023['social_sentiment_score'].mean()
avg_governance_sentiment = tesla_sent_2023['governance_sentiment_score'].mean()

print("Average Environmental Sentiment Score:", avg_environment_sentiment)
```

```
print("Average Social Sentiment Score:", avg_social_sentiment)
print("Average Governance Sentiment Score:", avg_governance_sentiment)
```

Average Environmental Sentiment Score: 0.10457317073170733  
 Average Social Sentiment Score: 0.0854420731707317  
 Average Governance Sentiment Score: 0.09855182926829269

Sentiment Distribution Visualization: Visualize the distribution of sentiment scores to see how the sentiments vary across articles. You can use histograms or box plots for this purpose.

```
In [50]: import matplotlib.pyplot as plt
import seaborn as sns

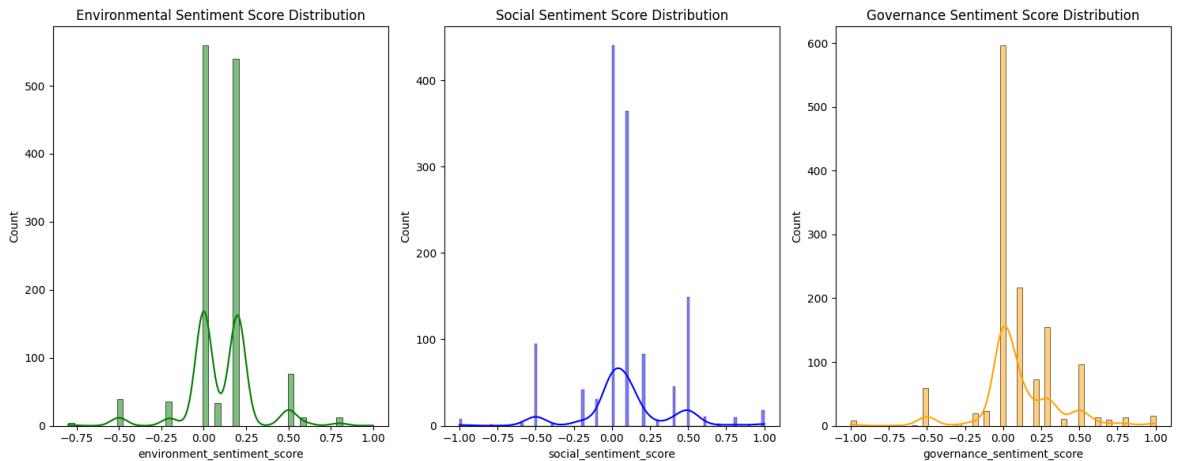
plt.figure(figsize=(15, 6))

# Environmental Sentiment Distribution
plt.subplot(1, 3, 1)
sns.histplot(tesla_sent_2023['environment_sentiment_score'], kde=True, color='green')
plt.title('Environmental Sentiment Score Distribution')

# Social Sentiment Distribution
plt.subplot(1, 3, 2)
sns.histplot(tesla_sent_2023['social_sentiment_score'], kde=True, color='blue')
plt.title('Social Sentiment Score Distribution')

# Governance Sentiment Distribution
plt.subplot(1, 3, 3)
sns.histplot(tesla_sent_2023['governance_sentiment_score'], kde=True, color='orange')
plt.title('Governance Sentiment Score Distribution')

plt.tight_layout()
plt.show()
```



Plot the sentiment scores over time to observe how the sentiments related to Tesla's environmental, social, and governance issues evolved throughout 2023.

```
In [51]: # Convert date column to datetime format
tesla_sent_2023['date'] = pd.to_datetime(tesla_sent_2023['date'], errors='coerce')

# Plotting sentiment scores over time
plt.figure(figsize=(15, 6))

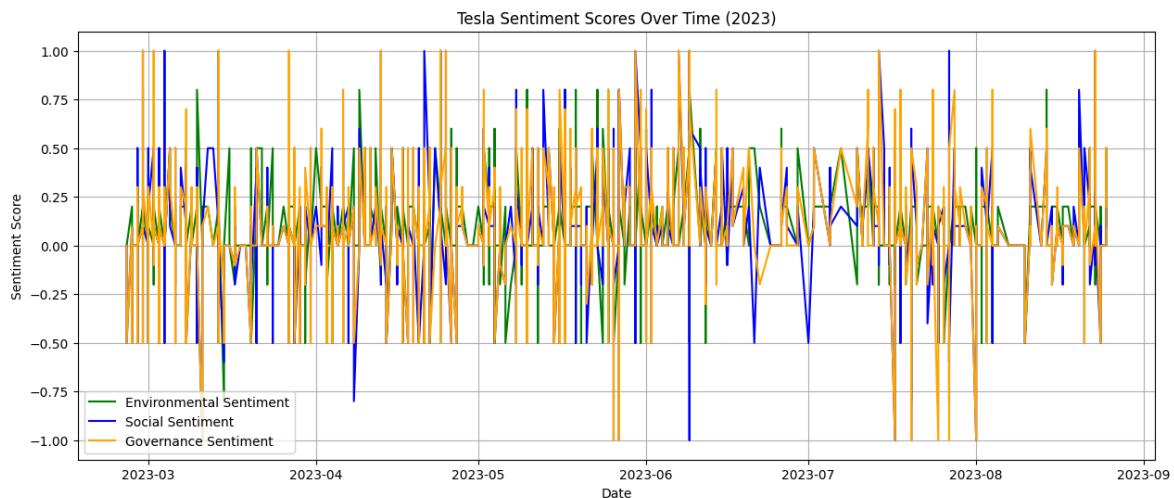
plt.plot(tesla_sent_2023['date'], tesla_sent_2023['environment_sentiment_score'])
```

```

plt.plot(tesla_sent_2023['date'], tesla_sent_2023['social_sentiment_score'], lab
plt.plot(tesla_sent_2023['date'], tesla_sent_2023['governance_sentiment_score'], lab

plt.xlabel('Date')
plt.ylabel('Sentiment Score')
plt.title('Tesla Sentiment Scores Over Time (2023)')
plt.legend()
plt.grid(True)
plt.show()

```



## Word Cloud for Key Words

generate a word cloud to visualize the most frequently used words in the news headlines.

```

In [52]: import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Combine all titles for word cloud generation
all_titles = " ".join(tesla_sent_2023['title'])

# Generate word cloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(
    all_titles)

# Display the word cloud
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud of Tesla News Headlines')
plt.show()

```

## Word Cloud of Tesla News Headlines



# Analyze Positive, Negative, and Neutral Articles

Split the articles based on their sentiment scores to analyze positive, negative, and neutral news separately.

## Define the Thresholds

- Positive Sentiment: Sentiment score  $> 0$
  - Negative Sentiment: Sentiment score  $< 0$
  - Neutral Sentiment: Sentiment score  $\approx 0$

```
In [53]: # Define a function to categorize sentiment
def categorize_sentiment(score):
    if score > 0:
        return 'Positive'
    elif score < 0:
        return 'Negative'
    else:
        return 'Neutral'

# Create new columns categorizing each sentiment type
tesla_sent_2023['environment_sentiment_category'] = tesla_sent_2023['environment']
tesla_sent_2023['social_sentiment_category'] = tesla_sent_2023['social']
tesla_sent_2023['governance_sentiment_category'] = tesla_sent_2023['governance']

# Filter articles by sentiment category
positive_environment_df = tesla_sent_2023[tesla_sent_2023['environment'] > 0]
negative_environment_df = tesla_sent_2023[tesla_sent_2023['environment'] < 0]
neutral_environment_df = tesla_sent_2023[tesla_sent_2023['environment'] == 0]

positive_social_df = tesla_sent_2023[tesla_sent_2023['social'] > 0]
negative_social_df = tesla_sent_2023[tesla_sent_2023['social'] < 0]
neutral_social_df = tesla_sent_2023[tesla_sent_2023['social'] == 0]

positive_governance_df = tesla_sent_2023[tesla_sent_2023['governance'] > 0]
negative_governance_df = tesla_sent_2023[tesla_sent_2023['governance'] < 0]
neutral_governance_df = tesla_sent_2023[tesla_sent_2023['governance'] == 0]
```

```

# Print the count of each type
print("Positive Environment Articles:", len(positive_environment_df))
print("Negative Environment Articles:", len(negative_environment_df))
print("Neutral Environment Articles:", len(neutral_environment_df))

print("Positive Social Articles:", len(positive_social_df))
print("Negative Social Articles:", len(negative_social_df))
print("Neutral Social Articles:", len(neutral_social_df))

print("Positive Governance Articles:", len(positive_governance_df))
print("Negative Governance Articles:", len(negative_governance_df))
print("Neutral Governance Articles:", len(neutral_governance_df))

```

```

Positive Environment Articles: 674
Negative Environment Articles: 79
Neutral Environment Articles: 559
Positive Social Articles: 690
Negative Social Articles: 182
Neutral Social Articles: 440
Positive Governance Articles: 602
Negative Governance Articles: 114
Neutral Governance Articles: 596

```

```

In [54]: import matplotlib.pyplot as plt

# Count the number of articles by sentiment type for each category
environment_counts = tesla_sent_2023['environment_sentiment_category'].value_counts()
social_counts = tesla_sent_2023['social_sentiment_category'].value_counts()
governance_counts = tesla_sent_2023['governance_sentiment_category'].value_counts()

# Plotting the counts for each sentiment category
plt.figure(figsize=(18, 6))

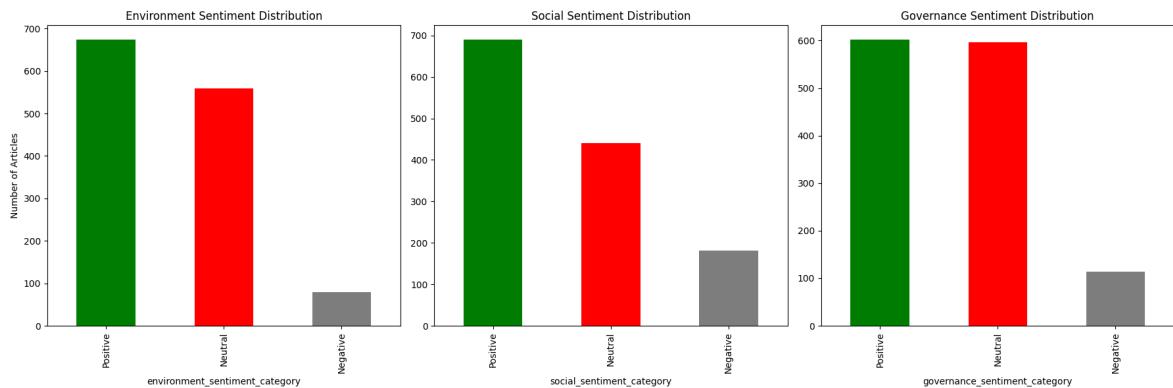
plt.subplot(1, 3, 1)
environment_counts.plot(kind='bar', color=['green', 'red', 'gray'])
plt.title('Environment Sentiment Distribution')
plt.ylabel('Number of Articles')

plt.subplot(1, 3, 2)
social_counts.plot(kind='bar', color=['green', 'red', 'gray'])
plt.title('Social Sentiment Distribution')

plt.subplot(1, 3, 3)
governance_counts.plot(kind='bar', color=['green', 'red', 'gray'])
plt.title('Governance Sentiment Distribution')

plt.tight_layout()
plt.show()

```



```
In [92]: # Filter the combined_df_2023 DataFrame to get only Tesla data
tesla_df = combined_df_2023[combined_df_2023['Symbol'] == 'TSLA']

# Display the first few rows of the filtered Tesla DataFrame to verify
tesla_df.head(2)
```

Out[92]:

	Open	High	Low	Close	Adj Close	Volume	Symbol
Date							
2023-01-03	118.470001	118.800003	104.639999	108.099998	108.099998	231402800	TSLA
2023-01-04	109.110001	114.589996	107.519997	113.639999	113.639999	180389000	TSLA

In [93]:

```
# Filter tesla_df for the specific date range: mid-June to mid-July
start_date = '2023-06-15'
end_date = '2023-07-15'
tesla_price_filtered = tesla_df[(tesla_df.index >= start_date) & (tesla_df.index <= end_date)]

# Filter sentiment data for the same date range
sentiment_filtered = tesla_sent_2023[(tesla_sent_2023['date'] >= start_date) & (tesla_sent_2023['date'] <= end_date)]

# Display the filtered price and sentiment data
print(tesla_price_filtered)
print(sentiment_filtered)
```

Date	Open	High	Low	Close	Adj Close	\
2023-06-15	248.399994	258.950012	247.289993	255.899994	255.899994	
2023-06-16	258.920013	263.600006	257.209991	260.540009	260.540009	
2023-06-20	261.500000	274.750000	261.119995	274.450012	274.450012	
2023-06-21	275.130005	276.989990	257.779999	259.459991	259.459991	
2023-06-22	250.770004	265.000000	248.250000	264.609985	264.609985	
2023-06-23	259.290009	262.450012	252.800003	256.600006	256.600006	
2023-06-26	250.070007	258.369995	240.699997	241.050003	241.050003	
2023-06-27	243.240005	250.389999	240.850006	250.210007	250.210007	
2023-06-28	249.699997	259.880005	248.889999	256.239990	256.239990	
2023-06-29	258.029999	260.739990	253.610001	257.500000	257.500000	
2023-06-30	260.600006	264.450012	259.890015	261.769989	261.769989	
2023-07-03	276.489990	284.250000	275.109985	279.820007	279.820007	
2023-07-05	278.820007	283.850006	277.600006	282.480011	282.480011	
2023-07-06	278.089996	279.970001	272.880005	276.540009	276.540009	
2023-07-07	278.429993	280.779999	273.769989	274.429993	274.429993	
2023-07-10	276.470001	277.519989	265.100006	269.609985	269.609985	
2023-07-11	268.649994	270.899994	266.369995	269.790009	269.790009	
2023-07-12	276.329987	276.519989	271.459991	271.989990	271.989990	
2023-07-13	274.589996	279.450012	270.600006	277.899994	277.899994	
2023-07-14	277.010010	285.299988	276.309998	281.380005	281.380005	

Date	Volume	Symbol	Year	Month	Weekday	adj_close_minus_1	\
2023-06-15	160171200	TSLA	2023	6	3	260.540009	
2023-06-16	167563700	TSLA	2023	6	4	274.450012	
2023-06-20	165611200	TSLA	2023	6	1	259.459991	
2023-06-21	211797100	TSLA	2023	6	2	264.609985	
2023-06-22	166875900	TSLA	2023	6	3	256.600006	
2023-06-23	176584100	TSLA	2023	6	4	241.050003	
2023-06-26	179990600	TSLA	2023	6	0	250.210007	
2023-06-27	164968200	TSLA	2023	6	1	256.239990	
2023-06-28	159770800	TSLA	2023	6	2	257.500000	
2023-06-29	131283400	TSLA	2023	6	3	261.769989	
2023-06-30	112267600	TSLA	2023	6	4	279.820007	
2023-07-03	119685900	TSLA	2023	7	0	282.480011	
2023-07-05	131530900	TSLA	2023	7	2	276.540009	
2023-07-06	120332100	TSLA	2023	7	3	274.429993	
2023-07-07	113602000	TSLA	2023	7	4	269.609985	
2023-07-10	119425400	TSLA	2023	7	0	269.790009	
2023-07-11	91972400	TSLA	2023	7	1	271.989990	
2023-07-12	95672100	TSLA	2023	7	2	277.899994	
2023-07-13	112681500	TSLA	2023	7	3	281.380005	
2023-07-14	119771100	TSLA	2023	7	4	290.380005	

Date	adj_close_plus_1	growth_1d	growth_30d	SMA_50	EMA_20
2023-06-15	256.790009	0.996534	1.593300	188.556999	221.043004
2023-06-16	255.899994	1.018132	1.616253	190.057399	224.804623
2023-06-20	260.540009	1.053389	1.613842	191.845199	229.532756
2023-06-21	274.450012	0.945382	1.510332	193.344199	232.382968
2023-06-22	259.459991	1.019849	1.564351	194.900599	235.452208
2023-06-23	264.609985	0.969729	1.522487	196.421799	237.466284
2023-06-26	256.600006	0.939400	1.400802	197.524799	237.807591
2023-06-27	241.050003	1.038000	1.489523	198.829000	238.988773
2023-06-28	250.210007	1.024100	1.540367	200.213000	240.631746
2023-06-29	256.239990	1.004917	1.546361	201.676800	242.238247
2023-06-30	257.500000	1.016582	1.505637	203.300399	244.098413
2023-07-03	261.769989	1.068954	1.581887	205.637000	247.500469

	date	request_id	companyname	url	title	date	company_relevance	environment_sentiment_score
2023-07-05	279.820007	1.009506	1.568114	207.985000	250.831854			
2023-07-06	282.480011	0.978972	1.464182	210.264800	253.280250			
2023-07-07	276.540009	0.992370	1.477257	212.540000	255.294511			
2023-07-10	274.429993	0.982436	1.474084	214.857199	256.657890			
2023-07-11	269.609985	1.000668	1.462514	217.049200	257.908568			
2023-07-12	269.790009	1.008154	1.408034	219.202799	259.249655			
2023-07-13	271.989990	1.021729	1.381487	221.524199	261.025878			
2023-07-14	277.899994	1.012523	1.379787	223.945599	262.964366			
935	52495	Tesla, Inc.						
936	52495	Tesla, Inc.						
937	52495	Tesla, Inc.						
938	52495	Tesla, Inc.						
939	52495	Tesla, Inc.						
...	...	...						
1041	52524	Tesla, Inc.						
1042	52524	Tesla, Inc.						
1043	52525	Tesla, Inc.						
1044	52525	Tesla, Inc.						
1045	52525	Tesla, Inc.						
935	https://www.cnbc.com/2023/06/15/stock-markets-...							
936	https://seekingalpha.com/news/3980049-xpeng-ro...							
937	https://www.cnbc.com/2023/06/15/toyota-stock-h...							
938	https://www.reuters.com/business/autos-transpo...							
939	https://www.cnbc.com/2023/06/15/elon-musk-met-...							
...	...	...						
1041	https://www.cnbc.com/2023/07/14/elon-musk-plan...							
1042	https://www.reuters.com/markets/asia/malaysian...							
1043	https://www.reuters.com/business/autos-transpo...							
1044	https://seekingalpha.com/news/3987864-tesla-en...							
1045	https://seekingalpha.com/article/4617194-stock...							
935	CNBC Daily Open: The Fed paused, but so did ma...	2023-06-15						
936	XPeng rolls out assisted driving tech in Beijing	2023-06-15						
937	Toyota stock having best week since 2009 after...	2023-06-15						
938	Latest Car & Auto Stories	2023-06-15						
939	Elon Musk met with Italy Prime Minister to tal...	2023-06-15						
...	...	...						
1041	Elon Musk plans Tesla and Twitter collaboratio...	2023-07-14						
1042	Malaysian PM Anwar discusses Tesla investment,...	2023-07-14						
1043	Tesla builds first Cybertruck after two years ...	2023-07-15						
1044	Tesla enters the electric truck battle after f...	2023-07-15						
1045	Stocks To Watch: Tesla Earnings, Labor Strikes...	2023-07-15						
935	company_relevance	environment_sentiment_score	\					
936	60	0.0						
937	70	0.0						
938	80	0.5						
939	80	0.2						
...	...	...						
1041	80	0.0						
1042	100	0.0						
1043	100	0.0						
1044	100	0.2						
1045	100	0.0						

```

environment_sentiment_reasoning \
935 The article does not mention any specific envi...
936 The article does not mention any specific envi...
937 The article mentions Toyota's plans for its ne...
938 The article does not provide specific informat...
939 The article does not mention any specific envi...
...
1041 The article does not provide specific informat...
1042 The article does not mention any specific envi...
1043 The article does not mention any specific envi...
1044 The article mentions the features of the new T...
1045 The article does not mention any specific envi...

social_sentiment_score \
935 0.2
936 0.0
937 0.0
938 0.1
939 0.5
...
1041 0.1
1042 1.0
1043 0.5
1044 0.0
1045 0.0

social_sentiment_reasoning \
935 The article briefly mentions France courting E...
936 The article does not provide any information a...
937 The article does not provide any information a...
938 The article does not provide significant infor...
939 The article mentions Elon Musk meeting with th...
...
1041 The article briefly mentions some controversie...
1042 The article mentions that the Malaysian Prime ...
1043 The article mentions Tesla's entry into the el...
1044 The article does not provide any specific info...
1045 The article does not provide any information a...

governance_sentiment_score \
935 0.0
936 0.0
937 0.5
938 0.3
939 0.5
...
1041 0.3
1042 1.0
1043 0.0
1044 0.0
1045 0.0

governance_sentiment_reasoning \
935 The article does not provide any sentiment tow...
936 The article does not provide any information a...
937 The article mentions shareholders voting in fa...
938 The article does not provide specific informat...
939 The article mentions Elon Musk meeting with th...
...
1041 The article mentions Elon Musk's involvement i...

```

```
1042 The article mentions that the Malaysian Prime ...
1043 The article does not provide any specific sent...
1044 The article does not discuss Tesla's governanc...
1045 The article does not discuss Tesla's governanc...
```

```
environment_sentiment_category social_sentiment_category \
935 Neutral Positive \
936 Neutral Neutral \
937 Positive Neutral \
938 Positive Positive \
939 Neutral Positive \
... ...
1041 Positive Positive \
1042 Neutral Positive \
1043 Neutral Positive \
1044 Positive Neutral \
1045 Neutral Neutral \
governance_sentiment_category
935 Neutral \
936 Neutral \
937 Positive \
938 Positive \
939 Positive \
... ...
1041 Positive \
1042 Positive \
1043 Neutral \
1044 Neutral \
1045 Neutral
```

[111 rows x 15 columns]

In [94]: `sentiment_filtered.head(1)`

Out[94]:

	request_id	companyname	url	title	date
935	52495	Tesla, Inc.	https://www.cnbc.com/2023/06/15/stock-markets...	CNBC Daily Open: The Fed paused, but so did ma...	2023-06-15

Extract Keywords from Article Titles from mid June-2023 to mid July-2023 (in this period the prices got increased)

In [95]: `from sklearn.feature_extraction.text import CountVectorizer`

```
# Extract keywords from the titles of filtered articles
vectorizer = CountVectorizer(stop_words='english', max_features=10) # Extract t
title_data = sentiment_filtered['title'].dropna() # Ensure no null values
title_keywords_matrix = vectorizer.fit_transform(title_data)
```

```
# Get the top keywords
top_keywords = vectorizer.get_feature_names_out()

# Display top keywords
print("Top Keywords from Article Titles (Mid-June to Mid-July 2023):")
print(top_keywords)
```

```
Top Keywords from Article Titles (Mid-June to Mid-July 2023):
['china' 'dominance' 'earths' 'elon' 'focus' 'lithium' 'musk' 'rare'
 'stocks' 'tesla']
```

```
In [96]: #Filter Titles for the Period (Mid-June to Mid-July 2023)
# Extract titles of articles from filtered sentiment data
filtered_titles = sentiment_filtered['title']

# Display titles of articles during the period
print("Titles of Articles from Mid-June to Mid-July 2023:")
for title in filtered_titles:
    print("-", title)
```

**Titles of Articles from Mid-June to Mid-July 2023:**

- CNBC Daily Open: The Fed paused, but so did markets
- XPeng rolls out assisted driving tech in Beijing
- Toyota stock having best week since 2009 after annual meeting, new EV goals
- Latest Car & Auto Stories
- Elon Musk met with Italy Prime Minister to talk A.I., birth rates, as hunt for new factory location continues
- Putting a number to Tesla's new Supercharger-for-all business
- Japan to give Toyota \$854 million support for domestic EV battery output, Nikkei reports
- These stocks left behind by the bull are finally starting to catch up
- Italy's Meloni meets Musk in Rome
- SK Signet to launch EV chargers with Tesla's charging standard this year
- Tesla snaps 13-day record-winning streak. Here's what the pros are saying
- Here are Thursday's biggest analyst calls: Nvidia, Tesla, Chipotle, Chevron, Netflix, Meta, Target and more
- Tesla Makes A Great Move, But Shares Are Still Overvalued (NASDAQ: TSLA)
- Tesla's U.S. electric vehicle market share will drop to 18% by 2026, BofA estimates
- Tesla, Boeing and more: CNBC's 'Halftime Report' traders answer your questions
- Tesla: German plant reducing shifts, production targets on track
- Twitter has suspended the accounts of a prominent Tesla and Elon Musk critic, PlainSite founder Aaron Greenspan
- Tesla's Concerning Q1: Forensic Analysis Of Price Elasticity, Downgrade To Hold (TSLA)
- Elon Musk says that Tesla's market cap is directly tied to whether it solves autonomous driving
- Tesla's Concerning Q1: Forensic Analysis Of Price Elasticity, Downgrade To Hold (TSLA)
- From AI to Twitter: What Elon Musk did and didn't discuss in his appearance at VivaTech in Paris
- Catalyst Watch: FedEx Earnings, Sports Teams Takeovers
- From AI to Twitter: What Elon Musk did and didn't discuss in his appearance at VivaTech in Paris
- Catalyst Watch: FedEx Earnings, Sports Teams Takeovers
- Intel spends \$33 billion in Germany in landmark expansion
- Sigma Lithium And Lithium Royalty: 2 Distinct Lithium Plays For Electric Vehicle Bulls
- Sigma Lithium And Lithium Royalty: 2 Distinct Lithium Plays For Electric Vehicle Bulls
- Sigma Lithium And Lithium Royalty: 2 Distinct Lithium Plays For Electric Vehicle Bulls
- Musk to brief Modi on Tesla's India investment plans in U.S. meeting-source
- Tesla urges EPA to finalize tougher US heavy-duty emissions cuts
- Tesla Returns to S&P 500 ESG Index after Adding More Environmental Disclosures
- 'Greenwashing vs greenhushing': Brands missing out on 'untapped' revenue by failing to communicate sustainability
- 'Greenwashing vs greenhushing': Brands missing out on 'untapped' revenue by failing to communicate sustainability
- How To Build A Dividend Portfolio With \$25,000 Among June's Top 30 Stocks
- How To Build A Dividend Portfolio With \$25,000 Among June's Top 30 Stocks
- How To Build A Dividend Portfolio With \$25,000 Among June's Top 30 Stocks
- How To Build A Dividend Portfolio With \$25,000 Among June's Top 30 Stocks
- The Canary In The Coal Mine
- Tesla's searing rally elicits downgrades, caution on Wall Street
- Shell: Don't Fear Declining Oil Prices (OTCMKTS: RYDAF)
- Shell: Don't Fear Declining Oil Prices (OTCMKTS: RYDAF)
- Six Nations leading the charge on Canada's largest battery farm
- S&P 500 closes near flat as Powell warns that more restrictive policy could be on the way: Live updates

- Tesla charging technology put on fast track to become US standard
- S&P 500 closes near flat as Powell warns that more restrictive policy could be on the way: Live updates
- Tesla charging technology put on fast track to become US standard
- Stocks slip as Powell hints at further rate hikes
- Tesla's searing rally elicits downgrades, caution on Wall Street
- Bolivia taps China, Russia's Rosatom in bid to unlock huge lithium riches
- Stocks slip as Powell hints at further rate hikes
- Tesla's searing rally elicits downgrades, caution on Wall Street
- Bolivia taps China, Russia's Rosatom in bid to unlock huge lithium riches
- Twitter applies temporary reading limits for all users, Elon Musk announces
- Tesla reported 466,140 deliveries for the second quarter, and production of 47,970 vehicles
- Stocks eke out small gain to start second half of 2023, Tesla rises 6%: Live updates
- Tesla beats second-quarter delivery estimates as price cuts pay off
- It was an accident: the scientists who have turned humid air into renewable power
- Tesla delivered a record number of cars last quarter, beating expectations
- Tesla reported 466,140 deliveries for the second quarter, and production of 47,970 vehicles
- Stocks eke out small gain to start second half of 2023, Tesla rises 6%: Live updates
- Tesla beats second-quarter delivery estimates as price cuts pay off
- It was an accident: the scientists who have turned humid air into renewable power
- Tesla delivered a record number of cars last quarter, beating expectations
- Tesla reported 466,140 deliveries for the second quarter, and production of 47,970 vehicles
- Stocks eke out small gain to start second half of 2023, Tesla rises 6%: Live updates
- Tesla beats second-quarter delivery estimates as price cuts pay off
- It was an accident: the scientists who have turned humid air into renewable power
- Tesla delivered a record number of cars last quarter, beating expectations
- Explainer-China's rare earths dominance in focus after mineral export curbs
- Automakers facing PS660m in costs due to sluggish EV rollouts
- Explainer-China's rare earths dominance in focus after mineral export curbs
- Explainer-China's rare earths dominance in focus after mineral export curbs
- China's rare earths dominance in focus after it limits germanium and gallium exports
- Explainer-China's rare earths dominance in focus after mineral export curbs
- Explainer-China's rare earths dominance in focus after mineral export curbs
- Automakers facing PS660m in costs due to sluggish EV rollouts
- Explainer-China's rare earths dominance in focus after mineral export curbs
- Explainer-China's rare earths dominance in focus after mineral export curbs
- China's rare earths dominance in focus after it limits germanium and gallium exports
- Explainer-China's rare earths dominance in focus after mineral export curbs
- Tesla wants EPA to finalize tougher vehicle emissions rules
- Tesla wants EPA to finalize tougher vehicle emissions rules
- Tesla wants EPA to finalize tougher vehicle emissions rules
- We Like NIO, But Heres Why The Rally Likely Wont Sustain (NYSE:NIO)
- John Goodenough obituary
- Explainer: What is Nasdaq's special rebalancing and its impact?
- Bloomberg
- China auto sales cool off in June but end the first half of the year up 10% (NASDAQ:LI)
- Elon Musk launches AI firm xAI as he looks to take on OpenAI
- Elon Musk thinks China is interested in an international AI framework

- Elon Musk's Tesla plans to launch electricity supplier in Britain
- Tesla's push into humanoid robots is set to benefit these 5 global stocks in the supply chain, HSBC says
- Oakmark Equity And Income Fund: Q2 2023 Commentary
- Tesla Stock: A Deep Dive Of Full Self-Driving - And A Reality Check (NASDAQ:TSLA)
- Tesla looking to make about half million EVs annually in India, Times of India reports
- India's finance ministry not considering tax waivers for Tesla, senior official says
- Tesla eyes producing 500K vehicles a year locally in India
- Nearly 100 US lawmakers urge EPA to finalize tougher vehicle emissions cuts
- Elon Musk launches AI startup and warns of a Terminator future
- Take Five: Buddy, can you spare a dollar?
- The bull and bear case for Tesla ahead of earnings
- Elon Musk says xAI will examine universe, work with Twitter and Tesla
- Tesla begins sale of cheaper Model Y vehicle in South Korea
- EVgo Stock: Risks May Overshadow The Potential Reward (NASDAQ:EVGO)
- Tesla, PepsiCo and more: CNBC's 'Halftime Report' traders answer your questions
- Tesla's Shanghai factory workers welcome bonus increase after record deliveries
- Elon Musk plans Tesla and Twitter collaborations with xAI, his new startup
- Malaysian PM Anwar discusses Tesla investment, Starlink with Musk
- Tesla builds first Cybertruck after two years of delays
- Tesla enters the electric truck battle after first Cybertruck is produced
- Stocks To Watch: Tesla Earnings, Labor Strikes And Russell 2000 Revival In The Spotlight

```
In [97]: from textblob import TextBlob
```

```
# Analyze sentiment of each title
for title in filtered_titles:
    analysis = TextBlob(title)
    sentiment_score = analysis.sentiment.polarity # Sentiment polarity between
    sentiment = "Positive" if sentiment_score > 0 else "Negative" if sentiment_s
    print(f"Title: {title}\nSentiment: {sentiment}\n")
```

Title: CNBC Daily Open: The Fed paused, but so did markets  
Sentiment: Neutral

Title: XPeng rolls out assisted driving tech in Beijing  
Sentiment: Neutral

Title: Toyota stock having best week since 2009 after annual meeting, new EV goal  
s  
Sentiment: Positive

Title: Latest Car & Auto Stories  
Sentiment: Positive

Title: Elon Musk met with Italy Prime Minister to talk A.I., birth rates, as hunt  
for new factory location continues  
Sentiment: Positive

Title: Putting a number to Tesla's new Supercharger-for-all business  
Sentiment: Positive

Title: Japan to give Toyota \$854 million support for domestic EV battery output,  
Nikkei reports  
Sentiment: Neutral

Title: These stocks left behind by the bull are finally starting to catch up  
Sentiment: Negative

Title: Italy's Meloni meets Musk in Rome  
Sentiment: Neutral

Title: SK Signet to launch EV chargers with Tesla's charging standard this year  
Sentiment: Neutral

Title: Tesla snaps 13-day record-winning streak. Here's what the pros are saying  
Sentiment: Neutral

Title: Here are Thursday's biggest analyst calls: Nvidia, Tesla, Chipotle, Chevro  
n, Netflix, Meta, Target and more  
Sentiment: Positive

Title: Tesla Makes A Great Move, But Shares Are Still Overvalued (NASDAQ: tsla)  
Sentiment: Positive

Title: Tesla's U.S. electric vehicle market share will drop to 18% by 2026, BofA  
estimates  
Sentiment: Neutral

Title: Tesla, Boeing and more: CNBC's 'Halftime Report' traders answer your quest  
ions  
Sentiment: Positive

Title: Tesla: German plant reducing shifts, production targets on track  
Sentiment: Neutral

Title: Twitter has suspended the accounts of a prominent Tesla and Elon Musk crit  
ic, PlainSite founder Aaron Greenspan  
Sentiment: Positive

Title: Tesla's Concerning Q1: Forensic Analysis Of Price Elasticity, Downgrade To  
Hold (tsla)

Sentiment: Neutral

Title: Elon Musk says that Tesla's market cap is directly tied to whether it solves autonomous driving

Sentiment: Positive

Title: Tesla's Concerning Q1: Forensic Analysis Of Price Elasticity, Downgrade To Hold (TSLA)

Sentiment: Neutral

Title: From AI to Twitter: What Elon Musk did and didn't discuss in his appearance at VivaTech in Paris

Sentiment: Neutral

Title: Catalyst Watch: FedEx Earnings, Sports Teams Takeovers

Sentiment: Neutral

Title: From AI to Twitter: What Elon Musk did and didn't discuss in his appearance at VivaTech in Paris

Sentiment: Neutral

Title: Catalyst Watch: FedEx Earnings, Sports Teams Takeovers

Sentiment: Neutral

Title: Intel spends \$33 billion in Germany in landmark expansion

Sentiment: Neutral

Title: Sigma Lithium And Lithium Royalty: 2 Distinct Lithium Plays For Electric Vehicle Bulls

Sentiment: Positive

Title: Sigma Lithium And Lithium Royalty: 2 Distinct Lithium Plays For Electric Vehicle Bulls

Sentiment: Positive

Title: Sigma Lithium And Lithium Royalty: 2 Distinct Lithium Plays For Electric Vehicle Bulls

Sentiment: Positive

Title: Musk to brief Modi on Tesla's India investment plans in U.S. meeting-source

Sentiment: Neutral

Title: Tesla urges EPA to finalize tougher US heavy-duty emissions cuts

Sentiment: Neutral

Title: Tesla Returns to S&P 500 ESG Index after Adding More Environmental Disclosures

Sentiment: Positive

Title: 'Greenwashing vs greenhushing': Brands missing out on 'untapped' revenue by failing to communicate sustainability

Sentiment: Negative

Title: 'Greenwashing vs greenhushing': Brands missing out on 'untapped' revenue by failing to communicate sustainability

Sentiment: Negative

Title: How To Build A Dividend Portfolio With \$25,000 Among June's Top 30 Stocks

Sentiment: Positive

Title: How To Build A Dividend Portfolio With \$25,000 Among June's Top 30 Stocks  
Sentiment: Positive

Title: How To Build A Dividend Portfolio With \$25,000 Among June's Top 30 Stocks  
Sentiment: Positive

Title: How To Build A Dividend Portfolio With \$25,000 Among June's Top 30 Stocks  
Sentiment: Positive

Title: The Canary In The Coal Mine  
Sentiment: Neutral

Title: Tesla's searing rally elicits downgrades, caution on Wall Street  
Sentiment: Neutral

Title: Shell: Don't Fear Declining Oil Prices (OTCMKTS:RYDAF)  
Sentiment: Neutral

Title: Shell: Don't Fear Declining Oil Prices (OTCMKTS:RYDAF)  
Sentiment: Neutral

Title: Six Nations leading the charge on Canada's largest battery farm  
Sentiment: Neutral

Title: S&P 500 closes near flat as Powell warns that more restrictive policy could be on the way: Live updates  
Sentiment: Positive

Title: Tesla charging technology put on fast track to become US standard  
Sentiment: Positive

Title: S&P 500 closes near flat as Powell warns that more restrictive policy could be on the way: Live updates  
Sentiment: Positive

Title: Tesla charging technology put on fast track to become US standard  
Sentiment: Positive

Title: Stocks slip as Powell hints at further rate hikes  
Sentiment: Neutral

Title: Tesla's searing rally elicits downgrades, caution on Wall Street  
Sentiment: Neutral

Title: Bolivia taps China, Russia's Rosatom in bid to unlock huge lithium riches  
Sentiment: Positive

Title: Stocks slip as Powell hints at further rate hikes  
Sentiment: Neutral

Title: Tesla's searing rally elicits downgrades, caution on Wall Street  
Sentiment: Neutral

Title: Bolivia taps China, Russia's Rosatom in bid to unlock huge lithium riches  
Sentiment: Positive

Title: Twitter applies temporary reading limits for all users, Elon Musk announces  
Sentiment: Neutral

Title: Tesla reported 466,140 deliveries for the second quarter, and production of 479,700 vehicles  
Sentiment: Neutral

Title: Stocks eke out small gain to start second half of 2023, Tesla rises 6%: Live updates  
Sentiment: Negative

Title: Tesla beats second-quarter delivery estimates as price cuts pay off  
Sentiment: Neutral

Title: It was an accident: the scientists who have turned humid air into renewable power  
Sentiment: Neutral

Title: Tesla delivered a record number of cars last quarter, beating expectations  
Sentiment: Neutral

Title: Tesla reported 466,140 deliveries for the second quarter, and production of 479,700 vehicles  
Sentiment: Neutral

Title: Stocks eke out small gain to start second half of 2023, Tesla rises 6%: Live updates  
Sentiment: Negative

Title: Tesla beats second-quarter delivery estimates as price cuts pay off  
Sentiment: Neutral

Title: It was an accident: the scientists who have turned humid air into renewable power  
Sentiment: Neutral

Title: Tesla delivered a record number of cars last quarter, beating expectations  
Sentiment: Neutral

Title: Tesla reported 466,140 deliveries for the second quarter, and production of 479,700 vehicles  
Sentiment: Neutral

Title: Stocks eke out small gain to start second half of 2023, Tesla rises 6%: Live updates  
Sentiment: Negative

Title: Tesla beats second-quarter delivery estimates as price cuts pay off  
Sentiment: Neutral

Title: It was an accident: the scientists who have turned humid air into renewable power  
Sentiment: Neutral

Title: Tesla delivered a record number of cars last quarter, beating expectations  
Sentiment: Neutral

Title: Explainer-China's rare earths dominance in focus after mineral export curbs  
Sentiment: Positive

Title: Automakers facing PS660m in costs due to sluggish EV rollouts

Sentiment: Negative

Title: Explainer-China's rare earths dominance in focus after mineral export curbs

Sentiment: Positive

Title: Explainer-China's rare earths dominance in focus after mineral export curbs

Sentiment: Positive

Title: China's rare earths dominance in focus after it limits germanium and gallium exports

Sentiment: Positive

Title: Explainer-China's rare earths dominance in focus after mineral export curbs

Sentiment: Positive

Title: Explainer-China's rare earths dominance in focus after mineral export curbs

Sentiment: Positive

Title: Automakers facing PS660m in costs due to sluggish EV rollouts

Sentiment: Negative

Title: Explainer-China's rare earths dominance in focus after mineral export curbs

Sentiment: Positive

Title: Explainer-China's rare earths dominance in focus after mineral export curbs

Sentiment: Positive

Title: China's rare earths dominance in focus after it limits germanium and gallium exports

Sentiment: Positive

Title: Explainer-China's rare earths dominance in focus after mineral export curbs

Sentiment: Positive

Title: Tesla wants EPA to finalize tougher vehicle emissions rules

Sentiment: Positive

Title: Tesla wants EPA to finalize tougher vehicle emissions rules

Sentiment: Positive

Title: Tesla wants EPA to finalize tougher vehicle emissions rules

Sentiment: Positive

Title: We Like NIO, But Heres Why The Rally Likely Wont Sustain (NYSE:NIO)

Sentiment: Neutral

Title: John Goodenough obituary

Sentiment: Neutral

Title: Explainer: What is Nasdaq's special rebalancing and its impact?

Sentiment: Positive

Title: Bloomberg

Sentiment: Neutral

Title: China auto sales cool off in June but end the first half of the year up 10% (NASDAQ:LI)

Sentiment: Positive

Title: Elon Musk launches AI firm xAI as he looks to take on OpenAI

Sentiment: Negative

Title: Elon Musk thinks China is interested in an international AI framework

Sentiment: Positive

Title: Elon Musk's Tesla plans to launch electricity supplier in Britain

Sentiment: Neutral

Title: Tesla's push into humanoid robots is set to benefit these 5 global stocks in the supply chain, HSBC says

Sentiment: Neutral

Title: Oakmark Equity And Income Fund: Q2 2023 Commentary

Sentiment: Neutral

Title: Tesla Stock: A Deep Dive Of Full Self-Driving - And A Reality Check (NASDAQ:TSLA)

Sentiment: Positive

Title: Tesla looking to make about half million EVs annually in India, Times of India reports

Sentiment: Negative

Title: India's finance ministry not considering tax waivers for Tesla, senior official says

Sentiment: Neutral

Title: Tesla eyes producing 500K vehicles a year locally in India

Sentiment: Neutral

Title: Nearly 100 US lawmakers urge EPA to finalize tougher vehicle emissions cuts

Sentiment: Positive

Title: Elon Musk launches AI startup and warns of a Terminator future

Sentiment: Neutral

Title: Take Five: Buddy, can you spare a dollar?

Sentiment: Neutral

Title: The bull and bear case for Tesla ahead of earnings

Sentiment: Neutral

Title: Elon Musk says xAI will examine universe, work with Twitter and Tesla

Sentiment: Neutral

Title: Tesla begins sale of cheaper Model Y vehicle in South Korea

Sentiment: Neutral

Title: EVgo Stock: Risks May Overshadow The Potential Reward (NASDAQ:EVGO)

Sentiment: Neutral

Title: Tesla, PepsiCo and more: CNBC's 'Halftime Report' traders answer your ques

tions

Sentiment: Positive

Title: Tesla's Shanghai factory workers welcome bonus increase after record deliveries

Sentiment: Positive

Title: Elon Musk plans Tesla and Twitter collaborations with xAI, his new startup

Sentiment: Positive

Title: Malaysian PM Anwar discusses Tesla investment, Starlink with Musk

Sentiment: Neutral

Title: Tesla builds first Cybertruck after two years of delays

Sentiment: Positive

Title: Tesla enters the electric truck battle after first Cybertruck is produced

Sentiment: Positive

Title: Stocks To Watch: Tesla Earnings, Labor Strikes And Russell 2000 Revival In The Spotlight

Sentiment: Neutral

## Correlate Titles with Price Increase

To see if the positive news aligns with the observed price increase:

Correlation: Compare the number of positive articles with the increase in stock price during the same period.

```
In [108...]: # Step 1: Reset the index to make 'Date' a column in tesla_price_filtered
tesla_price_filtered_daily = tesla_price_filtered.reset_index() # Date becomes a column

# Step 2: Ensure the sentiment data also has a column named 'Date' (adjust if it
avg_sentiment_by_date = sentiment_filtered.groupby(sentiment_filtered['date']).dt
avg_sentiment_by_date.rename(columns={'date': 'Date'}, inplace=True) # Rename the column

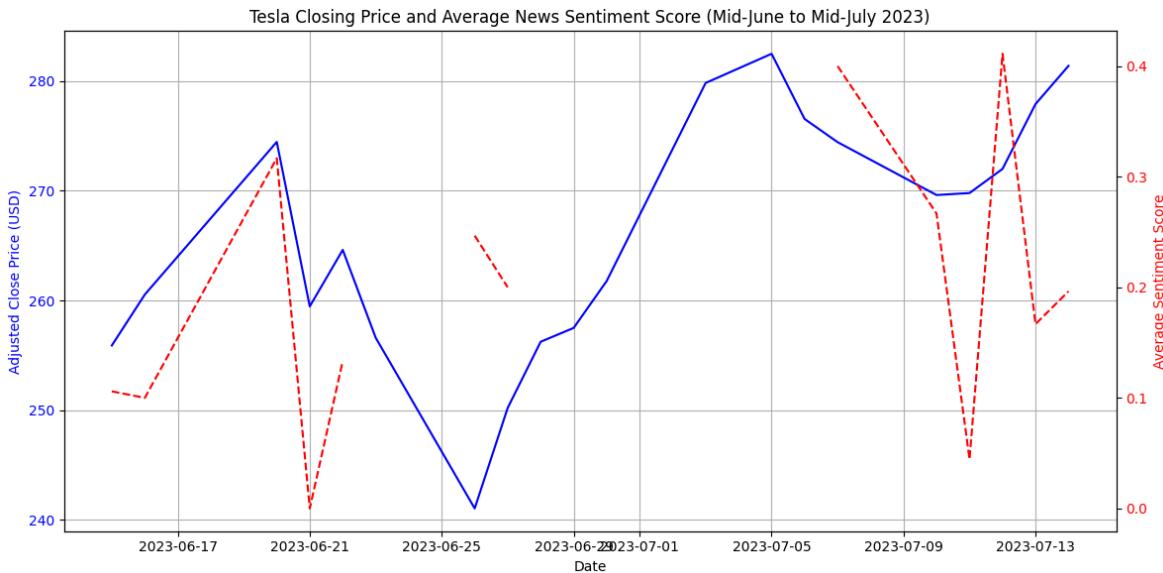
# Step 3: Merge the Tesla price data with the average sentiment data
tesla_price_filtered_daily = tesla_price_filtered_daily.merge(
    avg_sentiment_by_date[['Date', 'avg_sentiment_score']], on='Date', how='left'
)

# Step 4: Plot the updated Tesla stock price and average sentiment score
fig, ax1 = plt.subplots(figsize=(12, 6))

# Plot adjusted closing price on the left y-axis
ax1.set_xlabel('Date')
ax1.set_ylabel('Adjusted Close Price (USD)', color='blue')
ax1.plot(tesla_price_filtered_daily['Date'], tesla_price_filtered_daily['Close'])
ax1.tick_params(axis='y', labelcolor='blue')
ax1.grid(True)

# Create a second y-axis for the average sentiment score
ax2 = ax1.twinx()
ax2.set_ylabel('Average Sentiment Score', color='red')
ax2.plot(tesla_price_filtered_daily['Date'], tesla_price_filtered_daily['avg_sentiment_score'])
ax2.tick_params(axis='y', labelcolor='red')
```

```
# Add titles and legends
plt.title('Tesla Closing Price and Average News Sentiment Score (Mid-June to Mid-July 2023)')
fig.tight_layout()
plt.show()
```



In [109...]

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer

# Assuming sentiment_filtered contains article data with 'date', 'title', and 'keywords' columns

# Step 1: Identify the up peak and down peak in Tesla price data
up_peak_date = tesla_price_filtered_daily.loc[tesla_price_filtered_daily['Close'] == np.max(tesla_price_filtered_daily['Close'])]
down_peak_date = tesla_price_filtered_daily.loc[tesla_price_filtered_daily['Close'] == np.min(tesla_price_filtered_daily['Close'])]

# Step 2: Extract articles on up peak and down peak dates
up_peak_articles = sentiment_filtered[sentiment_filtered['date'] == up_peak_date['date'].values[0]]
down_peak_articles = sentiment_filtered[sentiment_filtered['date'] == down_peak_date['date'].values[0]]

# Step 3: Display the article titles and keywords for up peak
print(f"Articles on Up Peak ({up_peak_date['date'].values[0]}):")
for idx, row in up_peak_articles.iterrows():
    print(f"- Title: {row['title']}")
    # If 'keywords' is in a list-like format
    if 'keywords' in row and isinstance(row['keywords'], (list, str)):
        print(f"  Keywords: {row['keywords']}")
    else:
        # Extract keywords from title if there's no explicit 'keywords' column
        vectorizer = CountVectorizer(stop_words='english', max_features=5)
        title_keywords_matrix = vectorizer.fit_transform([row['title']])
        keywords = vectorizer.get_feature_names_out()
        print(f"  Extracted Keywords: {keywords}")

    print("\n")

# Step 4: Display the article titles and keywords for down peak
print(f"Articles on Down Peak ({down_peak_date['date'].values[0]}):")
for idx, row in down_peak_articles.iterrows():
    print(f"- Title: {row['title']}")
    # If 'keywords' is in a list-like format
    if 'keywords' in row and isinstance(row['keywords'], (list, str)):
```

```

        print(f"  Keywords: {row['keywords']}")")
else:
    # Extract keywords from title if there's no explicit 'keywords' column
    vectorizer = CountVectorizer(stop_words='english', max_features=5)
    title_keywords_matrix = vectorizer.fit_transform([row['title']])
    keywords = vectorizer.get_feature_names_out()
    print(f"  Extracted Keywords: {keywords}")

```

Articles on Up Peak (2023-07-05 00:00:00):

- Title: Explainer-China's rare earths dominance in focus after mineral export curbs
  - Extracted Keywords: ['china' 'curbs' 'dominance' 'earths' 'explainer']
- Title: Automakers facing PS660m in costs due to sluggish EV rollouts
  - Extracted Keywords: ['automakers' 'costs' 'ev' 'facing' 'ps660m']
- Title: Explainer-China's rare earths dominance in focus after mineral export curbs
  - Extracted Keywords: ['china' 'curbs' 'dominance' 'earths' 'explainer']
- Title: Explainer-China's rare earths dominance in focus after mineral export curbs
  - Extracted Keywords: ['china' 'curbs' 'dominance' 'earths' 'explainer']
- Title: China's rare earths dominance in focus after it limits germanium and gallium exports
  - Extracted Keywords: ['china' 'dominance' 'earths' 'exports' 'focus']
- Title: Explainer-China's rare earths dominance in focus after mineral export curbs
  - Extracted Keywords: ['china' 'curbs' 'dominance' 'earths' 'explainer']
- Title: Explainer-China's rare earths dominance in focus after mineral export curbs
  - Extracted Keywords: ['china' 'curbs' 'dominance' 'earths' 'explainer']
- Title: Explainer-China's rare earths dominance in focus after mineral export curbs
  - Extracted Keywords: ['china' 'curbs' 'dominance' 'earths' 'explainer']
- Title: Explainer-China's rare earths dominance in focus after mineral export curbs
  - Extracted Keywords: ['china' 'dominance' 'earths' 'exports' 'focus']
- Title: Explainer-China's rare earths dominance in focus after mineral export curbs
  - Extracted Keywords: ['china' 'curbs' 'dominance' 'earths' 'explainer']

Articles on Down Peak (2023-06-26 00:00:00):

- Title: The Canary In The Coal Mine
  - Extracted Keywords: ['canary' 'coal']
- Title: Tesla's searing rally elicits downgrades, caution on Wall Street
  - Extracted Keywords: ['caution' 'downgrades' 'elicits' 'rally' 'searing']
- Title: Shell: Don't Fear Declining Oil Prices (OTCMKTS:RYDAF)
  - Extracted Keywords: ['declining' 'don' 'fear' 'oil' 'otcmkts']
- Title: Shell: Don't Fear Declining Oil Prices (OTCMKTS:RYDAF)
  - Extracted Keywords: ['declining' 'don' 'fear' 'oil' 'otcmkts']
- Title: Six Nations leading the charge on Canada's largest battery farm
  - Extracted Keywords: ['battery' 'canada' 'charge' 'farm' 'largest']

**Generate a document realted to the period from mid June to mid July**

```
In [113...]: import cohere
from docx import Document

# Set your Cohere API key
co = cohere.Client('REDACTED')

# Prepare the data summary for mid-June to mid-July 2023
start_date = '2023-06-15'
end_date = '2023-07-15'

# filtered for this date range
min_price = tesla_price_filtered['Close'].min()
max_price = tesla_price_filtered['Close'].max()
avg_sentiment_score = avg_sentiment_by_date['avg_sentiment_score'].mean()

data_summary = f"""
Between mid-June to mid-July 2023, Tesla's stock price showed significant fluctuation. During this period, sentiment scores derived from news articles correlated with the price movement. Overall, the average sentiment score was {avg_sentiment_score:.2f}. Major news articles include:
"""

# Generate the draft article using Cohere API
response = co.generate(
    model='command-xlarge-nightly',
    prompt=data_summary,
    max_tokens=500,
    temperature=0.7
)
draft_article = response.generations[0].text

# Save the generated draft article to a Word document
document = Document()
document.add_heading('Financial Draft Article: Tesla Stock Analysis (Mid-June to Mid-July 2023)')
document.add_paragraph(draft_article)
document.save('finance_draft_article_TESLA_midJune_midJuly_2023.docx')
```

Sample of the generate doc based only on Analysis

finance\_draft\_article\_TESLA\_midJune\_midJuly\_2023.docx - Compatibility Mode - Saved to this PC -

References Mailings Review View Zotero Help Nitro Pro 10 Acrobat Foxit PDF

**Financial Draft Article: Tesla Stock Analysis (Mid-June to Mid-July 2023)**

It's fascinating to observe how news sentiment can directly impact stock price volatility, as exemplified by Tesla's stock journey from mid-June to mid-July 2023. The correlation between sentiment scores from news articles and stock price changes is a testament to the power of public perception and investor sentiment in the financial markets.

Here's a more detailed analysis of the situation:

**\*\*Price Fluctuation:\*\***  
The provided data indicates a substantial price swing in Tesla's stock within a one-month period. Starting at \$241.05 in mid-June, the price rose to \$282.48 by mid-July, representing an increase of approximately 17%. This volatility highlights the dynamic nature of the stock market, where various factors, including news and investor sentiment, can significantly influence stock prices.

**\*\*Sentiment Scores and Stock Price Correlation:\*\***  
The sentiment scores derived from news articles during this period demonstrated a clear correlation with the stock price changes. Positive news, often associated with higher sentiment scores, tends to boost investor confidence, leading to increased buying activity and a potential rise in the stock price. Conversely, negative news can evoke investor skepticism, resulting in selling pressure and a potential decline in the stock price.

The average sentiment score of 0.16 during this period suggests a slightly positive overall sentiment, which could have contributed to the upward trend in Tesla's stock price. However, individual news articles with more significant sentiment scores might have played a pivotal role in specific price movements.

**\*\*Impact of Major News Articles:\*\***

- **\*\*Positive News Impact:\*\*** News articles with high positive sentiment scores can significantly influence investor behavior. For instance, a news piece highlighting Tesla's groundbreaking innovation or successful expansion might generate excitement among investors, leading to increased buying interest and a subsequent rise in the stock price.
- **\*\*Negative News Impact:\*\*** On the other hand, negative news can have the opposite effect. News of production delays, regulatory issues, or competitive threats could evoke fear or uncertainty among investors, potentially causing a sell-off and a decline in the stock price.

## Time Series Analysis of News Impact (need to build a model)

Fetch historical stock price data using a service like yahoo finance or alpha api

Merge sentiment data with stock price data on the same date.