Project Report

Financial News Sentiment Analysis Using Natural language processing

FA690: Machine Learning in Finance

Rishabh Rawat

Fall 2022 Semester
November 06, 2022

Overview

The purpose of this project is to build an algorithm that easily and quickly parses the financial news from websites and Twitter news accounts and calculates the sentiment of the news headlines for any input ticker or index. I have tried to visualize the trending news topics for a particular stock or index using word cloud. Further, I have compared daily log returns and daily average sentiment using Pearson's correlation and Spearman's correlation. I used Python Programming Language and Jupyter Notebooks IDE to build the NLP application. I chose the Twitter news accounts of Investing.com (@Investing.com), New York Times business (@nytimesbusiness), and financial juice (@financialjuice) to scrape the financial news. For stock-specific news, I have chosen the FinViz stock screener. FinViz is one of the most powerful stock screeners giving free access to an incredible amount of information and updated news headlines for virtually any stock.

Data Collection and Discussion

In order to achieve the goals outlined for the project, I scrapped the twitter data from the three sources for the past month (Oct 5, 2022 to Nov 5, 2022) using the "snscrape library" in python. Initially, the data is sorted on the basis of news accounts. I have manipulated it and sorted it on the basis of date index. The three twitter account used are the most prominent sources of stock market.

```
                          Time  \
0    2022-11-04 18:12:45-04:00
1    2022-11-04 17:21:47-04:00
2    2022-11-04 16:47:51-04:00
3    2022-11-04 16:29:54-04:00
4    2022-11-04 16:09:57-04:00
...                        ...
7735 2022-10-04 20:32:43-04:00
7736 2022-10-04 20:30:12-04:00
7737 2022-10-04 20:20:37-04:00
7738 2022-10-04 20:19:51-04:00
7739 2022-10-04 20:03:06-04:00

                                              Tweets        Date
0     Investingcom$⚠️BREAKING:  *U.S. DOLLAR TUMBLES...  2022-11-04
1     Investingcom$⚠️WALL STREET WEEKLY PERFORMANCE:...  2022-11-04
2     Investingcom$*DRAFTKINGS CLOSES 28% LOWER IN B...  2022-11-04
3     Investingcom$*TWILIO PLUNGES 35% TO CLOSE AT L...  2022-11-04
4     Investingcom$⚠️BREAKING:  *DOW JUMPS 400 POINT...  2022-11-04
...                                                ...         ...
7735  financialjuice$JAPANESE SERVICES PMI, SEPTEMBE...  2022-10-04
7736  financialjuice$JAPANESE SERVICES PMI ACTUAL 52...  2022-10-04
7737  financialjuice$SOUTH KOREAN CENTRAL BANK: A WE...  2022-10-04
7738  financialjuice$SOUTH KOREAN CENTRAL BANK: CONS...  2022-10-04
7739  financialjuice$AUSTRALIA SELLS A$800 MLN OF 20...  2022-10-04

[7740 rows x 3 columns]
```

Sentiment Analysis

Sentiment analysis is a text analysis method that detects polarity (e.g. a *positive* or *negative* opinion) within the text, whether a whole document, paragraph, sentence, or clause.

Sentiment analysis aims to measure the attitude, sentiments, evaluations, attitudes, and emotions of a speaker/writer based on the computational treatment of subjectivity in a text. I have used VADER Sentiment analysis in the algorithm. **VADER** (Valence Aware Dictionary for Sentiment Reasoning) is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. It is available in the NLTK package and can be applied directly to unlabeled text data. VADER sentimental analysis relies on a dictionary that maps lexical features to emotion intensities known as sentiment scores.

Positive sentiment: compound score >= 0.05

Neutral sentiment: (compound score > -0.05) and (compound score < 0.05)

Negative sentiment: compound score <= -0.05

Sorting the rows by date-

```
                        User  \
Date
2022-10-04   financialjuice
2022-10-04  nytimesbusiness
2022-10-04  nytimesbusiness
2022-10-04  nytimesbusiness
2022-10-04  nytimesbusiness
...                     ...
2022-11-04   financialjuice
2022-11-04   financialjuice
2022-11-04   financialjuice
2022-11-04   financialjuice
2022-11-04     Investingcom

                                                   Tweet  Sentiment
Date
2022-10-04  AUSTRALIA SELLS A$800 MLN OF 2032 BONDS WITH A...     0.0000
2022-10-04  Amazon’s move is part of a wider cooling in th...    -0.6808
2022-10-04  Higher interest rates could add $1 trillion to...     0.7184
2022-10-04  The last time the S&amp;P 500 posted two conse...     0.3400
2022-10-04  Many economists in Britain and around the worl...    -0.3089
...                                                       ...        ...
2022-11-04  FED'S COLLINS: RATES MAY NEED TO RISE MORE THA...     0.0000
2022-11-04  FED'S COLLINS: IT MAKES PERFECT SENSE FOR THE ...     0.5719
2022-11-04  FED'S COLLINS: LAGS IN MONETARY POLICY AFFECTI...    -0.3612
2022-11-04  BANK OF AMERICA: AS THE FEDERAL RESERVE REMAIN...     0.4215
2022-11-04  ⚠️BREAKING:  *U.S. DOLLAR TUMBLES 1.9% IN BIGG...    -0.4278

[7740 rows x 3 columns]
```

Placing the data into a data frame -

```
df.head()
```

	Date	User	Tweet	Sentiment
0	2022-10-04	financialjuice	AUSTRALIA SELLS A$800 MLN OF 2032 BONDS WITH A...	0.0000
1	2022-10-04	nytimesbusiness	Amazon’s move is part of a wider cooling in th...	-0.6808
2	2022-10-04	nytimesbusiness	Higher interest rates could add $1 trillion to...	0.7184
3	2022-10-04	nytimesbusiness	The last time the S&P 500 posted two conse...	0.3400
4	2022-10-04	nytimesbusiness	Many economists in Britain and around the worl...	-0.3089

Next, I downloaded the daily SPY data for the past month and calculated its daily log returns. Getting data from yahoo finance and calculating the daily log values. Add a sentiment column to the spy data

Date	Adj Close	log_ret	sentiment
2022-10-06	373.200012	-0.028290	0.000000
2022-10-07	362.790009	-0.007665	0.080794
2022-10-10	360.019989	-0.006353	0.033166
2022-10-11	357.739990	-0.003304	0.020010
2022-10-12	356.559998	0.026049	0.044381
2022-10-13	365.970001	-0.023052	0.065815
2022-10-14	357.630005	0.025372	0.059129
2022-10-17	366.820007	0.011681	0.082488
2022-10-18	371.130005	-0.007112	0.019253
2022-10-19	368.500000	-0.008421	0.049454
2022-10-20	365.410004	0.024011	0.045488
2022-10-21	374.290009	0.012162	0.063610
2022-10-24	378.869995	0.015842	0.038875
2022-10-25	384.920013	-0.007563	0.067284
2022-10-26	382.019989	-0.005354	0.061485
2022-10-27	379.980011	0.023512	0.098835
2022-10-28	389.019989	-0.007249	0.026647
2022-10-31	386.209991	-0.004385	0.059550
2022-11-01	384.519989	-0.025416	0.033279
2022-11-02	374.869995	-0.010350	0.014583
2022-11-03	371.010010	0.014291	0.010175

After comparing the sentiments with the SPY log values using Pearson's correlation and Spearman's correlation, I conclude that the spy movements and sentiments are somewhat correlated.

Pearson's Correlation- The Pearson correlation measures the strength of the linear relationship between two variables. A correlation of **0.33** means these values are somewhat linearly correlated.

```
spy['log_ret'].corr(spy['sentiment'])
```

```
0.33302250976769976
```

Spearman correlation – To measure the strength and direction of monotonic association between two variables.

Coefficient= 0.198, rho measures the strength of association between two variables, which indicates that a **very weak** correlation exists.

p= 0.38, which means that very weak evidence of rejecting null hypothesis

```
from scipy.stats import spearmanr
coef, p = spearmanr(nasdaq['log_ret'], nasdaq['sentiment'])
p
```

```
0.3878822634645567
```

```
coef
```

```
0.1987012987012987
```

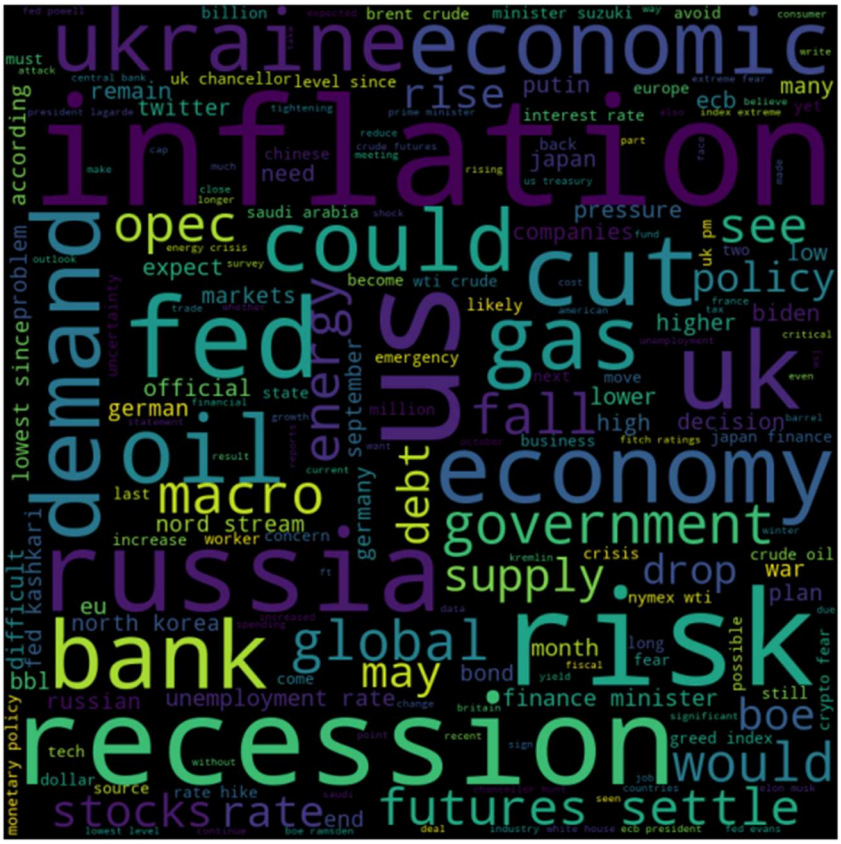
Visualization-

To visualize the news that changes the market sentiment, I have generated a wordcloud.

A word cloud is **a collection, or cluster, of words depicted in different sizes**. The bigger and bolder the word appears, the more often it's mentioned within a given text and the more important it is. Dividing the data frame into two data frames on the basis of positive and negative sentiment.

Positive Sentiment Words-

Negative Sentiment words -



The words such as growth, interest rates, banks earnings and increase boosts the market mood, whereas words such as recession, global risk, fed, opec, cut, gas, russia weakens the market sentiment.

Stock Specific Sentiments-

The news headlines stock-specific data for the **past three days.** I have chosen the five heavyweight stocks (AAPL, TSLA, AMZN, GOOGL, MSFT), which comprise nearly 38% of the NASDAQ weightage.

Extracting the news headlines and arranging them on basis of stocks-

```
Recent News Headlines for AAPL:
UPDATE 2-Musk's Twitter updates app to start charging $8 for blue checkmark ( Nov-05-22 06:09PM )
Apples AR glasses could be pushed back to 2025 or 2026 amid design issues, says analyst ( 11:52AM )
Apple and Google stocks just had their worst week in more than two years ( 11:48AM )
Amazon, Apple, and Other Hot Names Investors Are Watching ( 08:34AM )
Market Rally Buckling From Fed, Apple, Tesla, Cloud Stocks; What To Do Now ( 08:33AM )


Recent News Headlines for MSFT:
Amazon, Apple, and Other Hot Names Investors Are Watching ( Nov-05-22 08:34AM )
Market Rally Buckling From Fed, Apple, Tesla, Cloud Stocks; What To Do Now ( 08:33AM )
3 Rock-Solid Dividend Growth Stocks to Buy Right Now ( 05:55AM )
Weekly Roundup ( Nov-04-22 06:17PM )
Activision Blizzard Q3 Preview: What's in Store? ( 05:50PM )


Recent News Headlines for GOOGL:
Apple and Google stocks just had their worst week in more than two years ( Nov-05-22 11:48AM )
2 FAANG Stocks to Buy Like There's No Tomorrow ( 08:40AM )
Amazon, Apple, and Other Hot Names Investors Are Watching ( 08:34AM )
Market Rally Buckling From Fed, Apple, Tesla, Cloud Stocks; What To Do Now ( 08:33AM )
I had to be an overachiever to escape homelessness and land a six-figure tech job. Heres what I think about quiet quitting.
( 08:00AM )


Recent News Headlines for TSLA:
UPDATE 2-Musk's Twitter updates app to start charging $8 for blue checkmark ( Nov-05-22 06:09PM )
UPDATE 1-Musk's Twitter updates app to start charging $8 for blue checkmark ( 06:09PM )
Elon Musk Sends a Message to Tesla Shareholders, Fans ( 05:40PM )
EXPLAINER-Will Twitter layoffs violate U.S. law? ( 09:13AM )
Market Rally Buckling From Fed, Apple, Tesla, Cloud Stocks; What To Do Now ( 08:33AM )


Recent News Headlines for AMZN:
Johnson & Johnson's Big Purchase, Amazon's New Moves, and More ( Nov-05-22 10:45AM )
Don't Overlook This $104 Billion Item Buried In Amazon's Latest Report ( 09:53AM )
3 Perks of Buying Glasses and Contacts at Costco ( 09:00AM )
Amazon stock breaks long-term chart support, closes in on big inflection point ( 09:00AM )
This 1 Division of Northrop Grumman Is Leading the Charge. Is the Stock a Buy? ( 08:53AM )
```

Getting the headlines into the dataframe and using VaderSentiment Analysis on it-

```
news.head(10)
```

	Ticker	Date	Time	Headline	neg	neu	pos	compound
0	AAPL	Nov-06-22	11:58AM	Dow Jones Futures Loom: Market Rally Faces Fed...	0.119	0.881	0.000	-0.2263
1	AAPL	Nov-06-22	09:00AM	Streaming Services Raise Prices Again. Heres H...	0.196	0.804	0.000	-0.2960
2	AAPL	Nov-06-22	08:04AM	Better Big Tech Stock: Apple vs. Alphabet	0.000	0.674	0.326	0.4404
3	AAPL	Nov-06-22	08:00AM	Amazon Sweetens Prime Offerings as Growth Stalls	0.000	0.538	0.462	0.5106
4	AAPL	Nov-06-22	07:12AM	Market Rally Buckling From Fed, Apple, Tesla, ...	0.000	1.000	0.000	0.0000
5	AAPL	Nov-06-22	06:09PM	Market Rally Buckling From Fed, Apple, Tesla, ...	0.000	1.000	0.000	0.0000
6	AAPL	Nov-05-22	06:09PM	UPDATE 3-Musk's Twitter updates app to start c...	0.000	1.000	0.000	0.0000
7	AAPL	Nov-05-22	06:09PM	UPDATE 2-Musk's Twitter updates app to start c...	0.000	1.000	0.000	0.0000
8	AAPL	Nov-05-22	11:52AM	Apples AR glasses could be pushed back to 2025...	0.000	1.000	0.000	0.0000
9	AAPL	Nov-05-22	11:48AM	Apple and Google stocks just had their worst w...	0.240	0.760	0.000	-0.6249

Sorting the stocks-

Ticker	Date	Time	neg	neu	pos	compound
AAPL	2022-11-06	11:58AM	0.119	0.881	0.000	-0.2263
AAPL	2022-11-06	09:00AM	0.196	0.804	0.000	-0.2960
AAPL	2022-11-06	08:04AM	0.000	0.674	0.326	0.4404
AAPL	2022-11-06	08:00AM	0.000	0.538	0.462	0.5106
AAPL	2022-11-06	07:12AM	0.000	1.000	0.000	0.0000

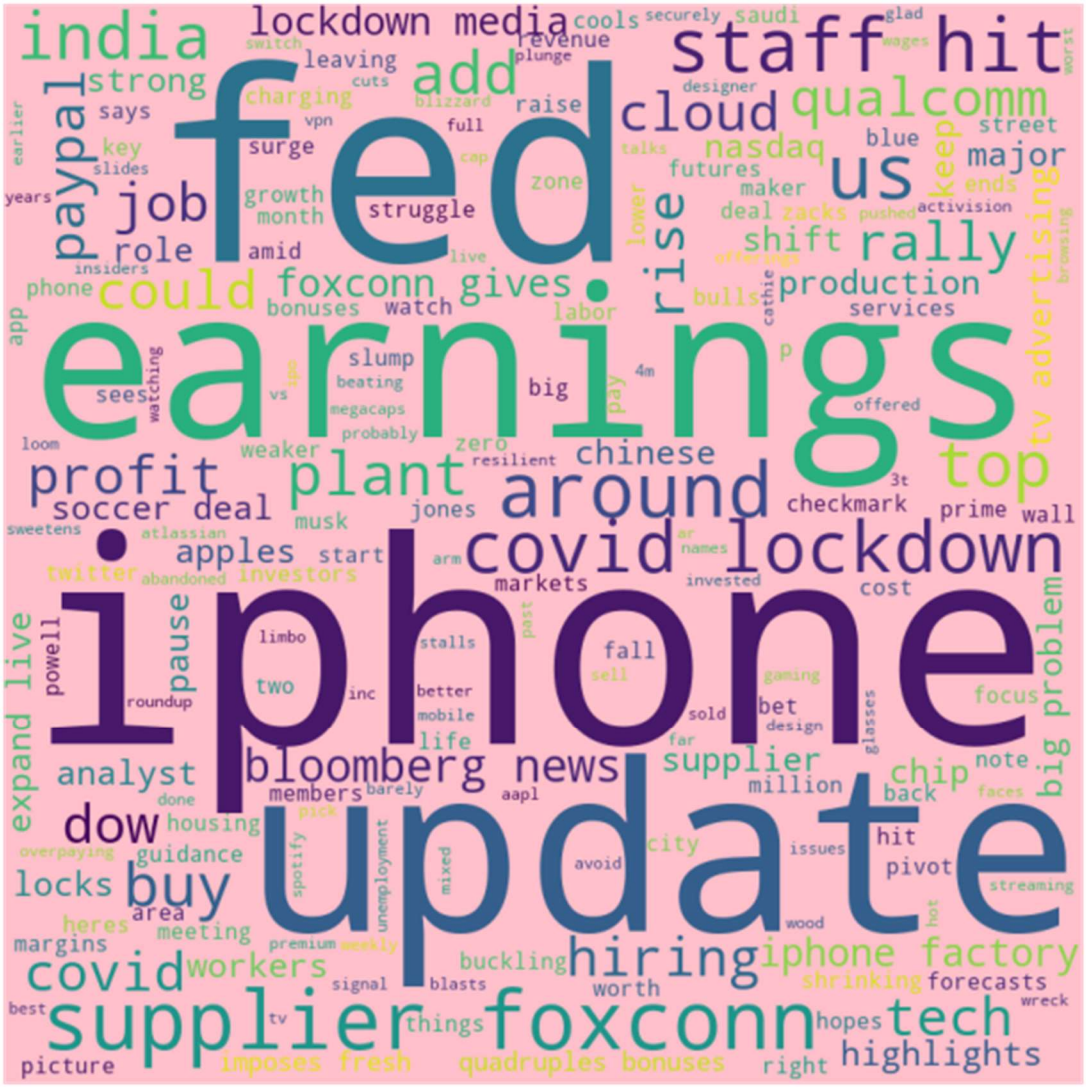
Ticker	Date	Time	neg	neu	pos	compound
MSFT	2022-11-06	11:58AM	0.119	0.881	0.000	-0.2263
MSFT	2022-11-06	07:12AM	0.000	1.000	0.000	0.0000
MSFT	2022-11-06	05:00AM	0.000	1.000	0.000	0.0000
MSFT	2022-11-05	08:34AM	0.000	0.825	0.175	0.1779
MSFT	2022-11-05	05:55AM	0.000	0.755	0.245	0.3818

Ticker	Date	Time	neg	neu	pos	compound
GOOGL	2022-11-06	11:58AM	0.119	0.881	0.000	-0.2263
GOOGL	2022-11-06	08:04AM	0.000	0.674	0.326	0.4404
GOOGL	2022-11-06	07:12AM	0.000	1.000	0.000	0.0000
GOOGL	2022-11-06	05:15AM	0.000	0.682	0.318	0.6808
GOOGL	2022-11-05	11:48AM	0.240	0.760	0.000	-0.6249

Ticker	Mean Sentiment
GOOGL	0.13
MSFT	0.12
AMZN	0.12
AAPL	0.06
TSLA	0.00

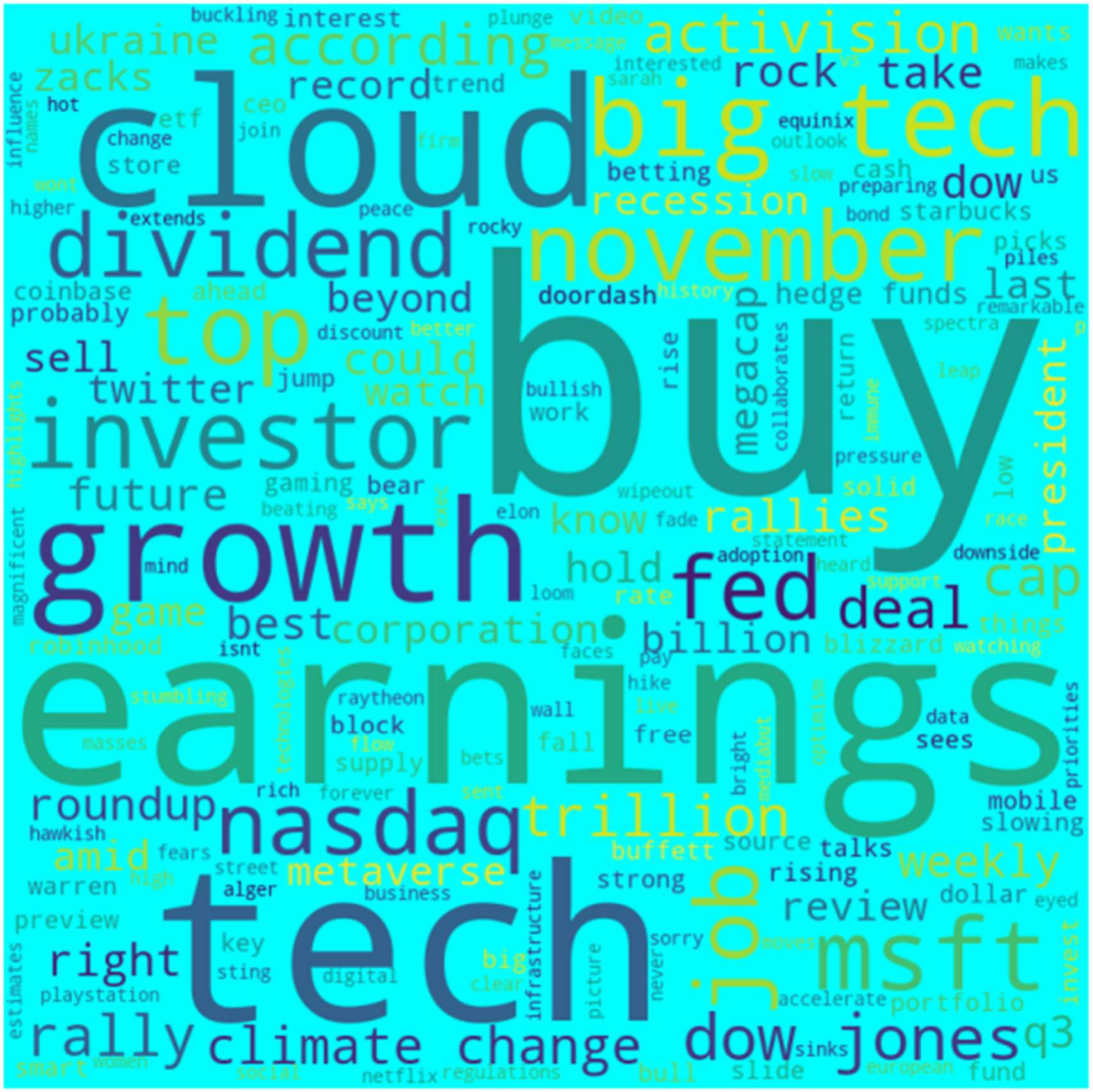
Results-

Apple-



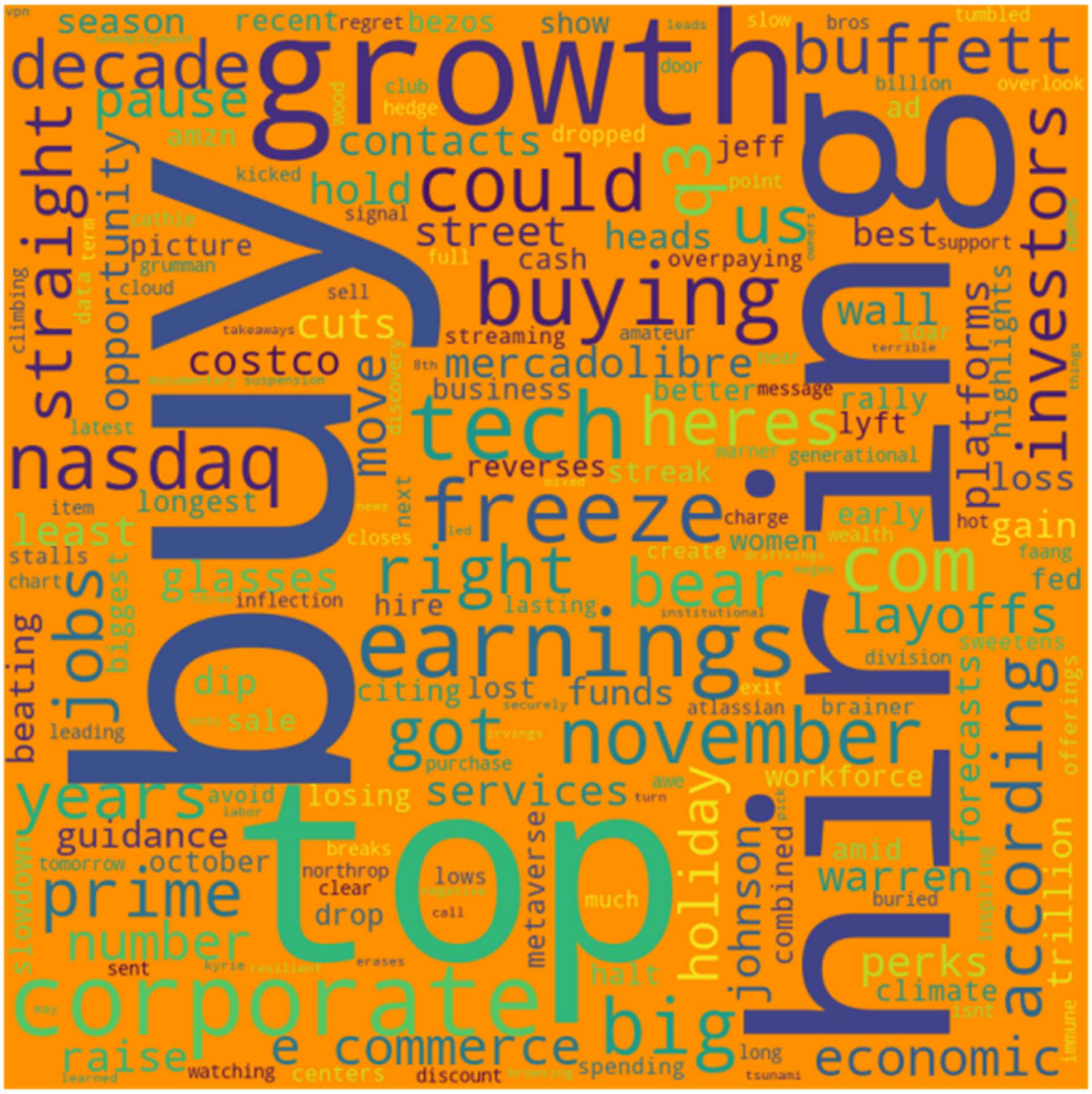
The main news topics for apple are iphone, earnings, fed, covid lockdown (china),supplier and foxconn

Microsoft-



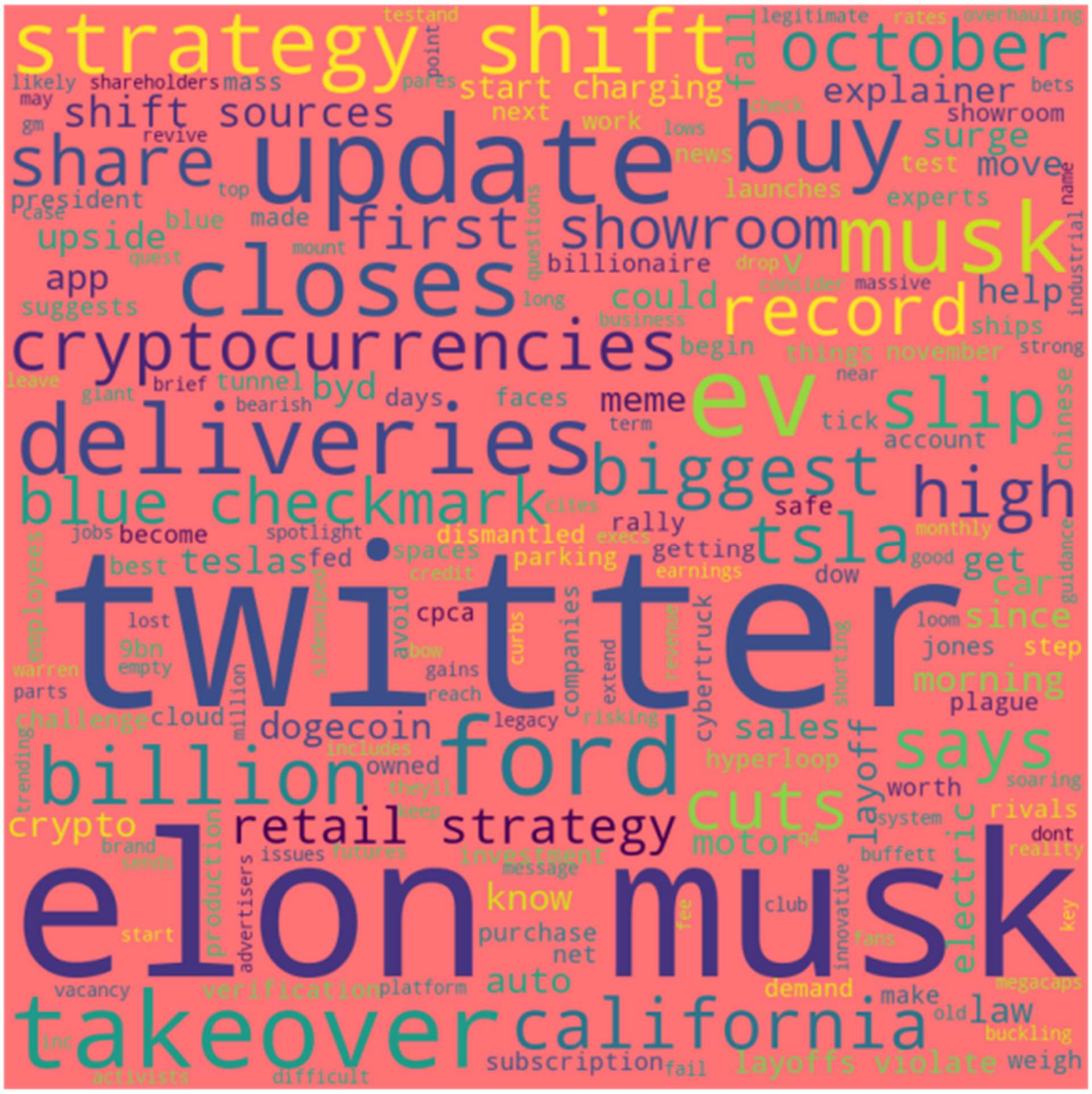
The main news topics for microsoft are earnings, buy (indicating a buy rating from brokerages),tech, growth, cloud, dividend and investor.

Amazon-



The main news topics for amazon are hiring(hiring slowdown maybe), buy, freeze, growth and earnings.

Tesla-



The main news topics for Tesla are Elon musk, twitter takeover, ford, cryptocurrencies, deliveries and shift.

Google-

The main news topics for Google are best, buy, growth, earnings, cloud, big tech, and antitrust.

Next Steps-

I will try to predict NASDAQ using the sentiments of these companies as they make up nearly 40% of NASDAQ. I will include more news sources to get more refined information. Further, a financial dictionary can be created to analyze the sentiments of strict financial vocabulary.

Citations-

Professor Zachary Twitter Code

Twitter accounts of Investing.com, NYTBusiness and Financialjuice and Finviz Website

https://www.geeksforgeeks.org/generating-word-cloud-python/

https://towardsdatascience.com/stock-news-sentiment-analysis-with-python-193d4b4378d4

```
import numpy as np
from sklearn.model_selection import train_test_split
```

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation, NMF
from sklearn.preprocessing import LabelEncoder
```

```
!pip install -q kaggle
!mkdir -~/.kaggle
!cp kaggle.json ~/.kaggle/
!ls ~/.kaggle
!chmod 600 /root/.kaggle/kaggle.json
!kaggle datasets download -d ankurzing/sentiment-analysis-for-financial-news
!unzip -q financial-news-dataset.zip
!ls -lrt
```

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation, NMF
from sklearn.preprocessing import LabelEncoder
```

```
!cp -r /content/drive/MyDrive/Colab\ Notebooks/.kaggle /content/drive/MyDrive/Colab\ Notebooks/data_science_projects/Sentiment_analysis_financial_news/
!ls /content/drive/MyDrive/Colab\ Notebooks/data_science_projects/Sentiment_analysis_financial_news/
```

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation, NMF
from sklearn.preprocessing import LabelEncoder
```

```
!cp /content/drive/MyDrive/Colab\ Notebooks/data_science_projects/Sentiment_analysis_financial_news/financial-news-dataset.zip financial-news-dataset.zip
!unzip -q financial-news-dataset.zip
!ls -lrt
```

```
In [16]: df.to_csv('news_sentiment.csv')
         df.shape
         df.dropna().shape
         df.dropna().shape
         df.head()
         # Initialize variables
         total_words = 0
         positive_count = 0
         negative_count = 0
         neutral_count = 0

         # Loop through each text in the dataset
         for text in df['cleaned_text']:
             # Split the text into words
             words = text.split()
             # Update the total word count
             total_words += len(words)
             # Count positive, negative, and neutral words
             for word in words:
                 if word in positive_words:
                     positive_count += 1
                 elif word in negative_words:
                     negative_count += 1
                 else:
                     neutral_count += 1

         average_length = df['cleaned_text'].str.length.mean()
         print("Average length of cleaned text:", average_length)
         # Calculate the proportions of positive, negative, and neutral words
         positive_ratio = positive_count / total_words
         negative_ratio = negative_count / total_words
         neutral_ratio = neutral_count / total_words
         print("Positive ratio:", positive_ratio)
         print("Negative ratio:", negative_ratio)
         print("Neutral ratio:", neutral_ratio)

         # Calculate the sentiment score
         sentiment_score = positive_ratio - negative_ratio
         print("Sentiment score:", sentiment_score)

         # Create a word cloud for the cleaned text
         text = df['cleaned_text'].sum()
         wordcloud = WordCloud(width=800, height=400).generate(text)
         plt.figure(figure_size=(10, 5))
         plt.imshow(wordcloud, interpolation='bilinear')
         plt.axis('off')
         plt.title('Word Cloud')
         plt.show()

Average length of cleaned text: 1.0
Positive ratio: 0.0
Negative ratio: 0.0
Neutral ratio: 1.0
Sentiment score: 0.0
```

```
In [17]: df.to_csv('news_sentiment.csv')
         df.shape
         df.dropna().shape
         df.dropna().shape
         df.head()
         # Initialize variables
         total_words = 0
         positive_count = 0
         negative_count = 0
         neutral_count = 0

         # Loop through each text in the dataset
         for text in df['cleaned_text']:
             # Split the text into words
             words = text.split()
             # Update the total word count
             total_words += len(words)
             # Count positive, negative, and neutral words
             for word in words:
                 if word in positive_words:
                     positive_count += 1
                 elif word in negative_words:
                     negative_count += 1
                 else:
                     neutral_count += 1

         average_length = df['cleaned_text'].str.length.mean()
         print("Average length of cleaned text:", average_length)
         # Calculate the proportions of positive, negative, and neutral words
         positive_ratio = positive_count / total_words
         negative_ratio = negative_count / total_words
         neutral_ratio = neutral_count / total_words
         print("Positive ratio:", positive_ratio)
         print("Negative ratio:", negative_ratio)
         print("Neutral ratio:", neutral_ratio)

         # Calculate the sentiment score
         sentiment_score = positive_ratio - negative_ratio
         print("Sentiment score:", sentiment_score)

         # Create a word cloud for the cleaned text
         text = df['cleaned_text'].sum()
         wordcloud = WordCloud(width=800, height=400).generate(text)
         plt.figure(figure_size=(10, 5))
         plt.imshow(wordcloud, interpolation='bilinear')
         plt.axis('off')
         plt.title('Word Cloud')
         plt.show()

Average length of cleaned text: 1.0
Positive ratio: 0.0
Negative ratio: 0.0
Neutral ratio: 1.0
Sentiment score: 0.0
```