1. PROBLEM STATEMENT

Presented by: Rolando López

Business Context

The prices of the stocks of companies listed under a global exchange are influenced by a variety of factors, with the company's financial performance, innovations and collaborations, and market sentiment being factors that play a significant role. News and media reports can rapidly affect investor perceptions and, consequently, stock prices in the highly competitive financial industry. With the sheer volume of news and opinions from a wide variety of sources, investors and financial analysts often struggle to stay updated and accurately interpret its impact on the market. As a result, investment firms need sophisticated tools to analyze market sentiment and integrate this information into their investment strategies.

Problem Definition

With an ever-rising number of news articles and opinions, an investment startup aims to leverage artificial intelligence to address the challenge of interpreting stock-related news and its impact on stock prices. They have collected historical daily news for a specific company listed under NASDAQ, along with data on its daily stock price and trade volumes.

As a member of the Data Science and AI team in the startup, you have been tasked with analyzing the data, developing an AI-driven sentiment analysis system that will automatically process and analyze news articles to gauge market sentiment, and summarizing the news at a weekly level to enhance the accuracy of their stock price predictions and optimize investment strategies. This will empower their financial analysts with actionable insights, leading to more informed investment decisions and improved client outcomes.

Data Dictionary

- `Date` : The date the news was released
- `News` : The content of news articles that could potentially affect the company's stock price
- `Open` : The stock price (in $) at the beginning of the day
- `High` : The highest stock price (in $) reached during the day
- `Low` : The lowest stock price (in $) reached during the day
- `Close` : The adjusted stock price (in $) at the end of the day
- `Volume` : The number of shares traded during the day
- `Label` : The sentiment polarity of the news content
  - 1: positive
  - 0: neutral
  - -1: negative

Rolando's interpretation of the problem statement

The main challenge here is the overwhelming amount of stock news out there – it's really hard for analysts to process it all and figure out the market sentiment (positive, neutral, negative) and how it affects stock prices.

Our project is about building an AI system for an investment startup. We're using historical news and stock data to automatically classify the sentiment of news articles and create weekly summaries. The goal is to give financial analysts clearer, actionable insights to help improve stock price predictions and make smarter investment decisions using our models.

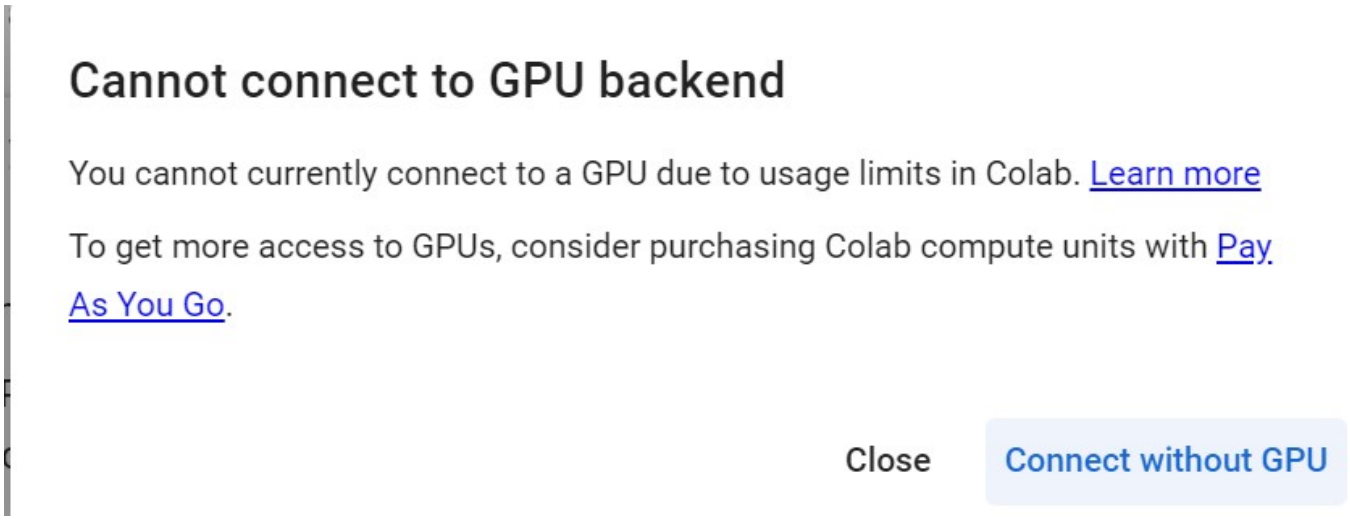
2. ENV PREP

2.1. additional instructions

Note: If the free-tier GPU of Google Colab is not accessible (due to unavailability or exhaustion of daily limit or other reasons), the following steps can be taken:

1. Wait for 12-24 hours until the GPU is accessible again or the daily usage limits are reset.
2. Switch to a different Google account and resume working on the project from there.
3. Try using the CPU runtime:

   - To use the CPU runtime, click on *Runtime => Change runtime type => CPU => Save*
   - One can also click on the *Continue without GPU* option to switch to a CPU runtime (kindly refer to the snapshot below)
   - The instructions for running the code on the CPU are provided in the respective sections of the notebook.



2.2. Libraries

2.2.1 Installing libraries (part 1)

In [ ]:
```
# # installing the sentence-transformers and gensim libraries for word embeddings

!pip install -U --no-cache-dir sentence-transformers gensim transformers tqdm scikit-learn pandas numpy matplotlib seaborn -q
```

2.2.2. Importing libraries (part 1)

In [ ]:
```
# To manipulate and analyze data
import pandas as pd
import numpy as np

# To visualize data
import matplotlib.pyplot as plt
import seaborn as sns

# To used time-related functions
import time

# To parse JSON data
import json

# To build, tune, and evaluate ML models
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, f1_score, pre
cision_score, recall_score

# To load/create word embeddings
from gensim.models import Word2Vec
from gensim.models import KeyedVectors
from gensim.scripts.glove2word2vec import glove2word2vec

# To work with transformer models
import torch
from sentence_transformers import SentenceTransformer

# To implement progress bar related functionalities
from tqdm import tqdm
tqdm.pandas()

# To ignore unnecessary warnings
import warnings
warnings.filterwarnings('ignore')
```

2.2.3. Installing libraries (part 2)

In [ ]:
```
# Installation for GPU llama-cpp-python
# uncomment and run the following code in case GPU is being used
!CMAKE_ARGS="-DLLAMA_CUBLAS=on" FORCE_CMAKE=1 pip install llama-cpp-python==0.2.40 --force-reinstall --no-cache-dir -q

# other versions tested
#0.1.78
#0.2.40

# Installation for CPU llama-cpp-python
# uncomment and run the following code in case GPU is not being used
#!CMAKE_ARGS="-DLLAMA_CUBLAS=off" FORCE_CMAKE=1 pip install llama-cpp-python==0.1.85 --force-reinstall --no-cache-dir -q
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.2/11.2 MB 115.4 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.0/62.0 kB 162.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.5/45.5 kB 280.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.9/134.9 kB 290.1 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.4/16.4 MB 319.7 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.8/45.8 kB 282.4 MB/s eta 0:00:00
  Building wheel for llama-cpp-python (pyproject.toml) ... done
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
gensim 4.3.3 requires numpy<2.0,>=1.18.5, but you have numpy 2.2.5 which is incompatible.
google-colab 1.0.0 requires pandas==2.2.2, but you have pandas 2.2.3 which is incompatible.
tsfresh 0.21.0 requires scipy>=1.14.0; python_version >= "3.10", but you have scipy 1.13.1 which is incompatible.
numba 0.60.0 requires numpy<2.1,>=1.22, but you have numpy 2.2.5 which is incompatible.
tensorflow 2.18.0 requires numpy<2.1.0,>=1.26.0, but you have numpy 2.2.5 which is incompatible.
```

2.2.4. Importing libraries (part 2)

In [ ]:
```
# Function to download the model from the Hugging Face model hub
from huggingface_hub import hf_hub_download

# Importing the Llama class from the llama_cpp module
from llama_cpp import Llama

# Importing the library for data manipulation
import pandas as pd

from tqdm import tqdm # For progress bar related functionalities
tqdm.pandas()
```

2.2.5. Printing libraries' current versions

In [ ]:
```
!pip list | grep gensim
!pip list | grep transformers
!pip list | grep scikit-learn
!pip list | grep numpy
!pip list | grep pandas
!pip list | grep matplotlib
!pip list | grep seaborn
```

```
gensim                          4.3.3
sentence-transformers           4.1.0
transformers                    4.51.3
scikit-learn                    1.6.1
numpy                           2.2.5
geopandas                       1.0.1
pandas                          2.2.3
pandas-datareader               0.10.0
pandas-gbq                      0.28.1
pandas-stubs                    2.2.2.240909
sklearn-pandas                  2.2.0
matplotlib                      3.10.3
matplotlib-inline               0.1.7
matplotlib-venn                 1.1.2
seaborn                         0.13.2
```

2.3. Loading the dataset

In [ ]:
```
# mounting Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [ ]:
```
# reading the csv file
stock_news_orig = pd.read_csv("/content/drive/MyDrive/Education/DBA AI/M06 NLP/project 06_Stock mkt news/stock_news.csv")

#Creating a copy of the dataset
data = stock_news_orig.copy()
```

2.4. Globals

In [ ]:
```
# variables to be used across the notebook
gbl_random_seed = 42
```

3. DATA OVERVIEW

In [ ]:
```
# looking at the first 5 rows
data.head()
```

Out[ ]:

	Date	News	Open	High	Low	Close	Volume	Label
0	2019-01-02	The tech sector experienced a significant dec...	41.740002	42.244999	41.482498	40.246914	130672400	-1
1	2019-01-02	Apple lowered its fiscal Q1 revenue guidance ...	41.740002	42.244999	41.482498	40.246914	130672400	-1
2	2019-01-02	Apple cut its fiscal first quarter revenue fo...	41.740002	42.244999	41.482498	40.246914	130672400	-1
3	2019-01-02	This news article reports that yields on long...	41.740002	42.244999	41.482498	40.246914	130672400	-1
4	2019-01-02	Apple's revenue warning led to a decline in U...	41.740002	42.244999	41.482498	40.246914	130672400	-1

In [ ]:
```
# shape of the dataset
data.shape
```

Out[ ]:
```
(349, 8)
```

In [ ]:
```
# data types of columns
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 349 entries, 0 to 348
Data columns (total 8 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Date    349 non-null    object 
 1   News    349 non-null    object 
 2   Open    349 non-null    float64
 3   High    349 non-null    float64
 4   Low     349 non-null    float64
 5   Close   349 non-null    float64
 6   Volume  349 non-null    int64  
 7   Label   349 non-null    int64  
dtypes: float64(4), int64(2), object(2)
memory usage: 21.9+ KB
```

In [ ]:
```
# changing the type of the column
data['Date'] = pd.to_datetime(data['Date'])
```

In [ ]:
```
# stats of the dataset
data.describe().T
```

Out[ ]:

	count	mean	min	25%	50%	75%	
Date	349	2019-02-16 16:05:30.085959936	2019-01-02 00:00:00	2019-01-14 00:00:00	2019-02-05 00:00:00	2019-03-22 00:00:00	2019-00:
Open	349.0	46.229233	37.567501	41.740002	45.974998	50.7075	66.8
High	349.0	46.700458	37.817501	42.244999	46.025002	50.849998	67
Low	349.0	45.745394	37.305	41.482498	45.639999	49.7775	65.8(
Close	349.0	44.926317	36.254131	40.246914	44.596924	49.11079	64.8(
Volume	349.0	128948236.103152	45448000.0	103272000.0	115627200.0	151125200.0	244439
Label	349.0	-0.054441	-1.0	-1.0	0.0	0.0	

The dataset contains 349 entries covering January to April 2019. Stock prices (Open, High, Low, Close) range roughly from $36 - 67, averagingaround$ 44-46. Trading Volume is highly variable, ranging from about 4.5 million to 244 million shares, with a mean around 128 million. The sentiment labels (-1, 0, 1) are present, with a slight overall bias towards negative or neutral sentiment in the dataset.

In [ ]:
```
# checking for null values
data.isnull().sum()
```

Out[ ]:

	0
Date	0
News	0
Open	0
High	0
Low	0
Close	0
Volume	0
Label	0

dtype: int64

there are no null values

In [ ]:
```
# checking for duplidated values
data.duplicated().sum()
```

Out[ ]: 0

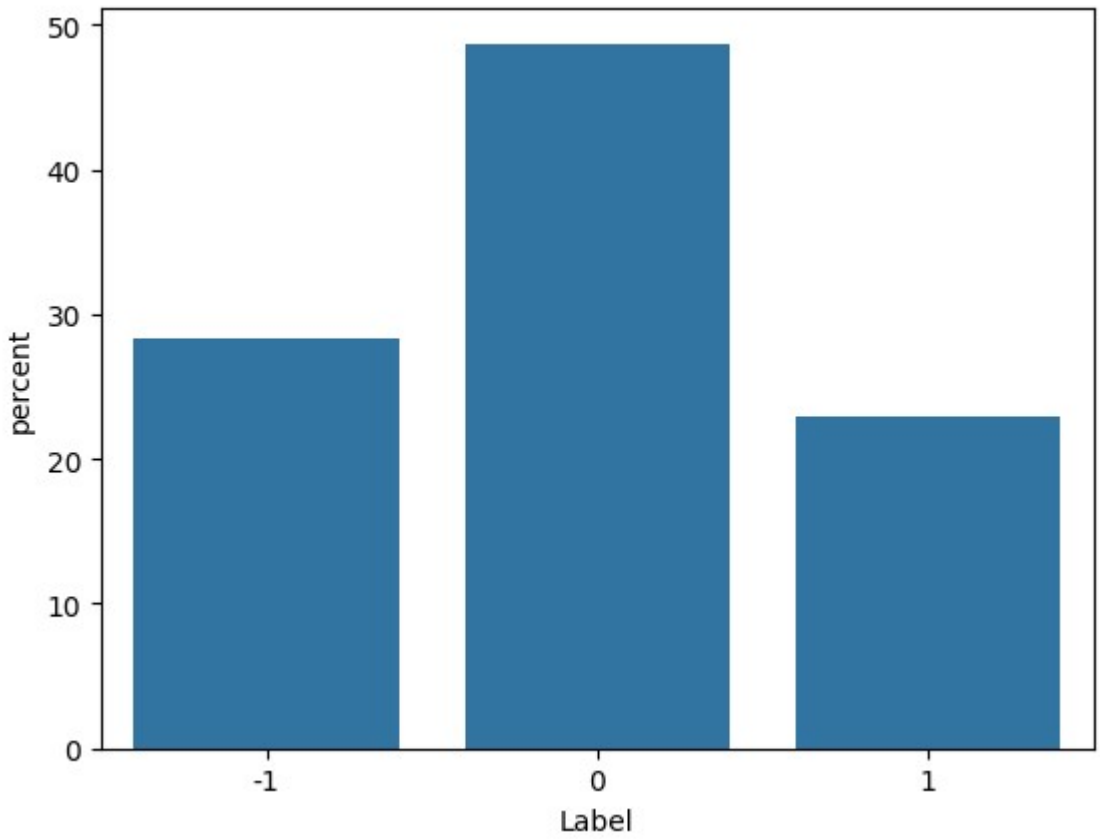
no duplicated values

4. EXPLORATORY DATA ANALYSIS

4.1. Univariate Analysis

Observations on label

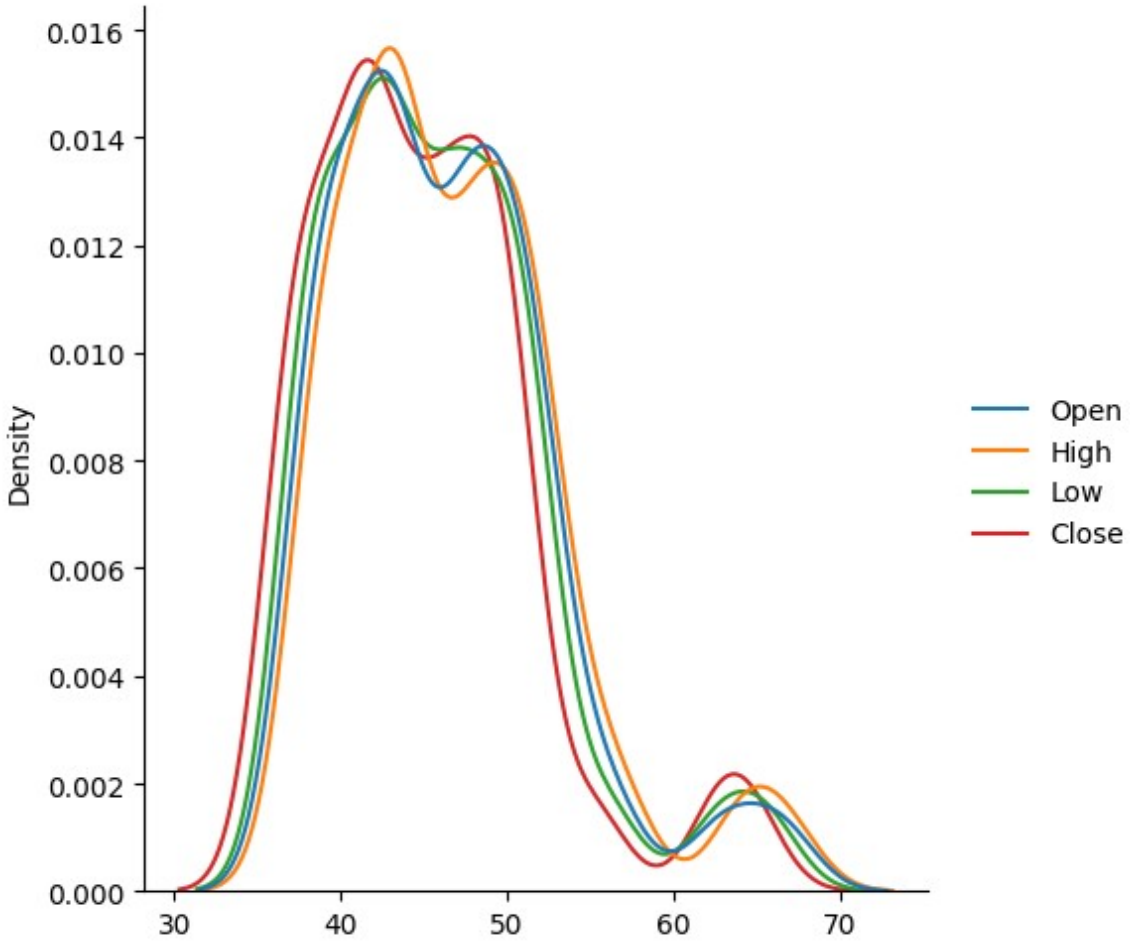
In [ ]:
```
sns.countplot(data=data, x="Label", stat="percent");
```



'Label' column is imbalanced, 0 is dominant at ~48%, while -1 and 1 are between 23-28% each

Observations on Price

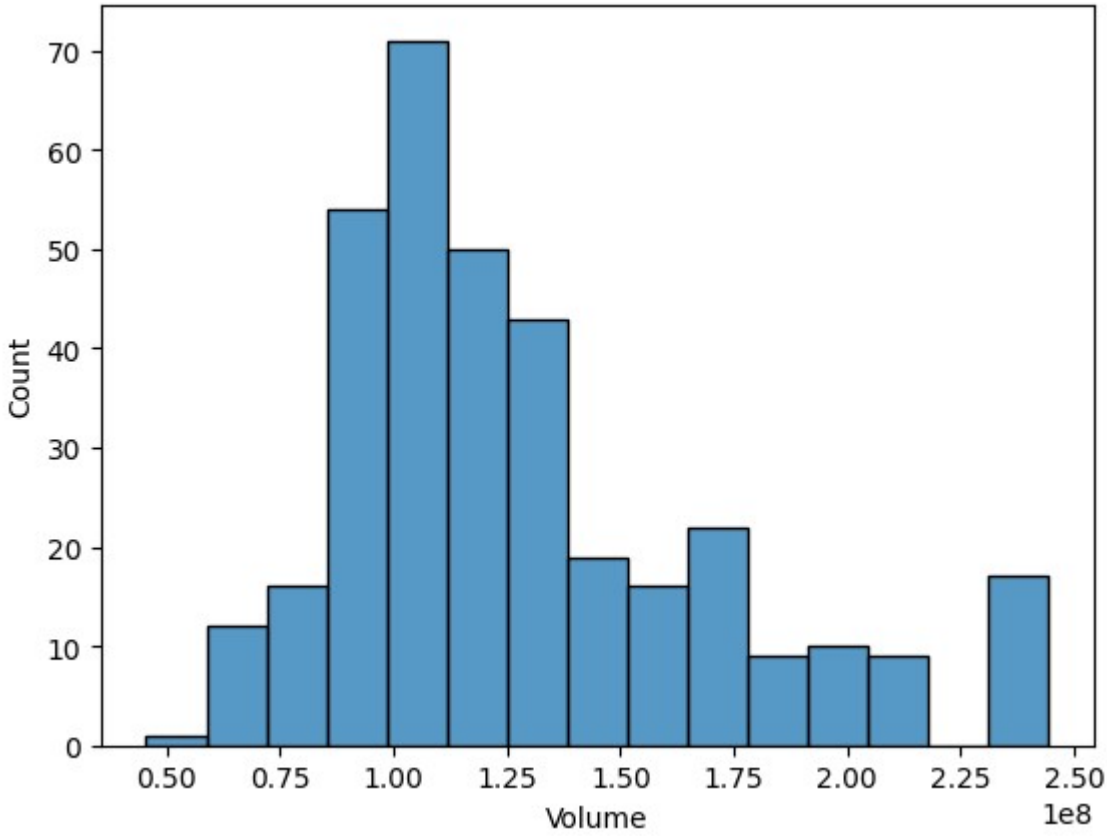
In [ ]:
```
sns.displot(data=data[['Open','High','Low','Close']], kind="kde", palette
="tab10");
```



The distributions of Open, High, Low, and Close prices are very similar and overlap significantly, suggesting relatively consistent and narrow daily price ranges. There's a slight bimodal shape, that indicates the stock spent time trading in two distinct price bands during the dataset's period.

Observations on Volume

In [ ]:
```
sns.histplot(data, x='Volume');
```



The distribution of trading volume is right-skewed, showing that most days have moderate volume, but there are infrequent occurrences of much higher trading activity. These volume spikes likely correspond to specific events driving increased market interest.

Observations on News length

In [ ]:
```
#Calculating the total number of words present in the news content.
data['news_len'] = data['News'].apply(lambda x: len(x.split(' ')))

data['news_len'].describe() # statistical summary for the news content leng
th
```

Out[ ]:

	news_len
count	349.000000
mean	49.312321
std	5.727770
min	19.000000
25%	46.000000
50%	50.000000
75%	53.000000
max	61.000000

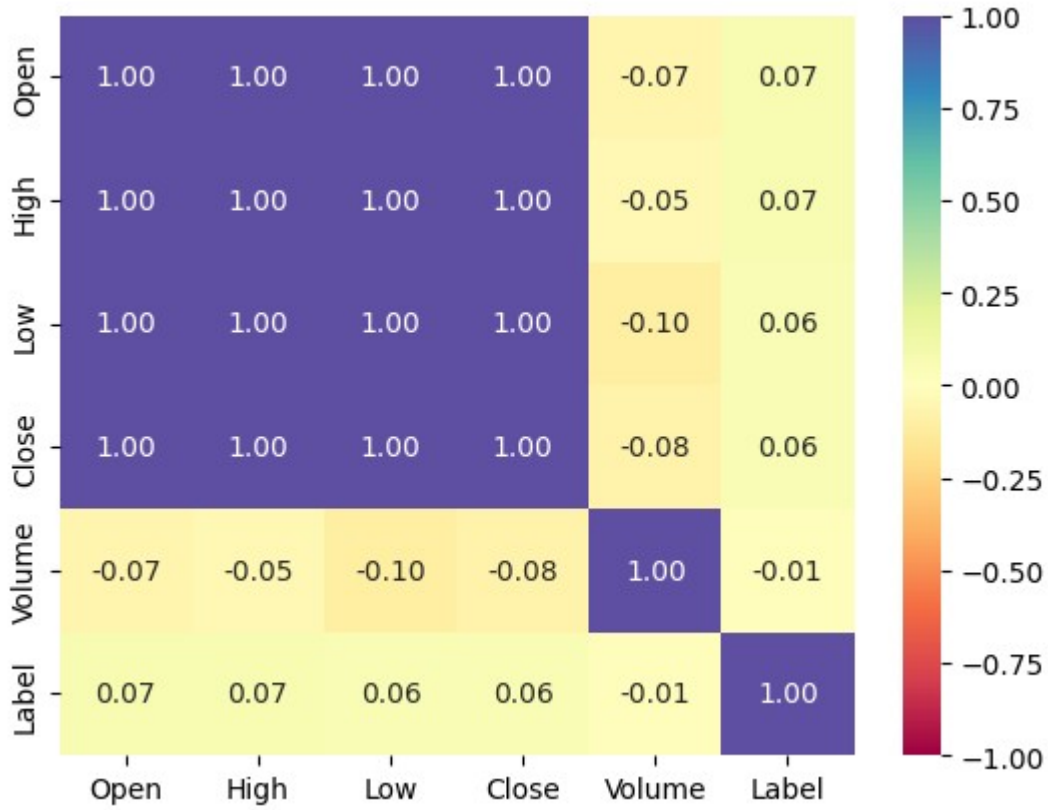
dtype: float64

The stats of the length of the news are: 49 chars mean, with a minimum of 19 and max of 61 chars

4.2. Bivariate Analysis

Correlation

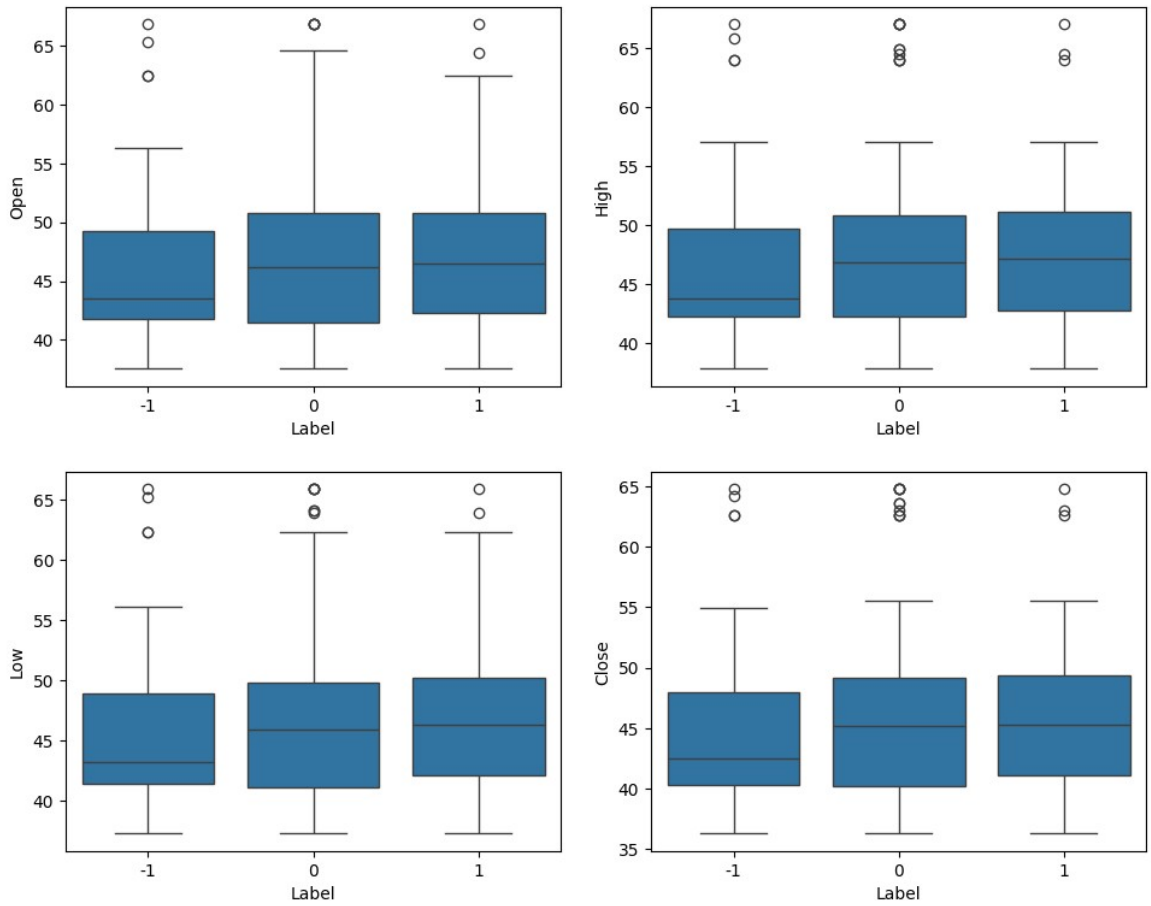
In [ ]:
```
data[['Open', 'High', 'Low', 'Close', 'Volume', 'Label']].corr()

sns.heatmap(
    data[['Open', 'High', 'Low', 'Close', 'Volume', 'Label']].corr(), annot
=True, vmin=-1, vmax=1, fmt=".2f", cmap="Spectral"
);
```



The correlation matrix confirms the strong relationship between the different price points in a day. Crucially, it shows that the sentiment label has very little linear correlation with the absolute price level. It also shows a weak positive linear correlation with volume, suggesting that while not a strong linear relationship, there's a slight tendency for volume to be higher when sentiment moves away from -1 towards 1, but the box plot for volume provides a clearer picture that both positive and negative sentiments drive volume higher than neutral. The matrix indicates the relationships are largely non-linear or weak linearly, particularly concerning the sentiment label's link to price.

Label vs Price

In [ ]:
```
plt.figure(figsize=(10, 8))

for i, variable in enumerate(['Open', 'High', 'Low', 'Close']):
    plt.subplot(2, 2, i + 1)
    sns.boxplot(data=data, x="Label", y=variable)
    plt.tight_layout(pad=2)

plt.show()
```

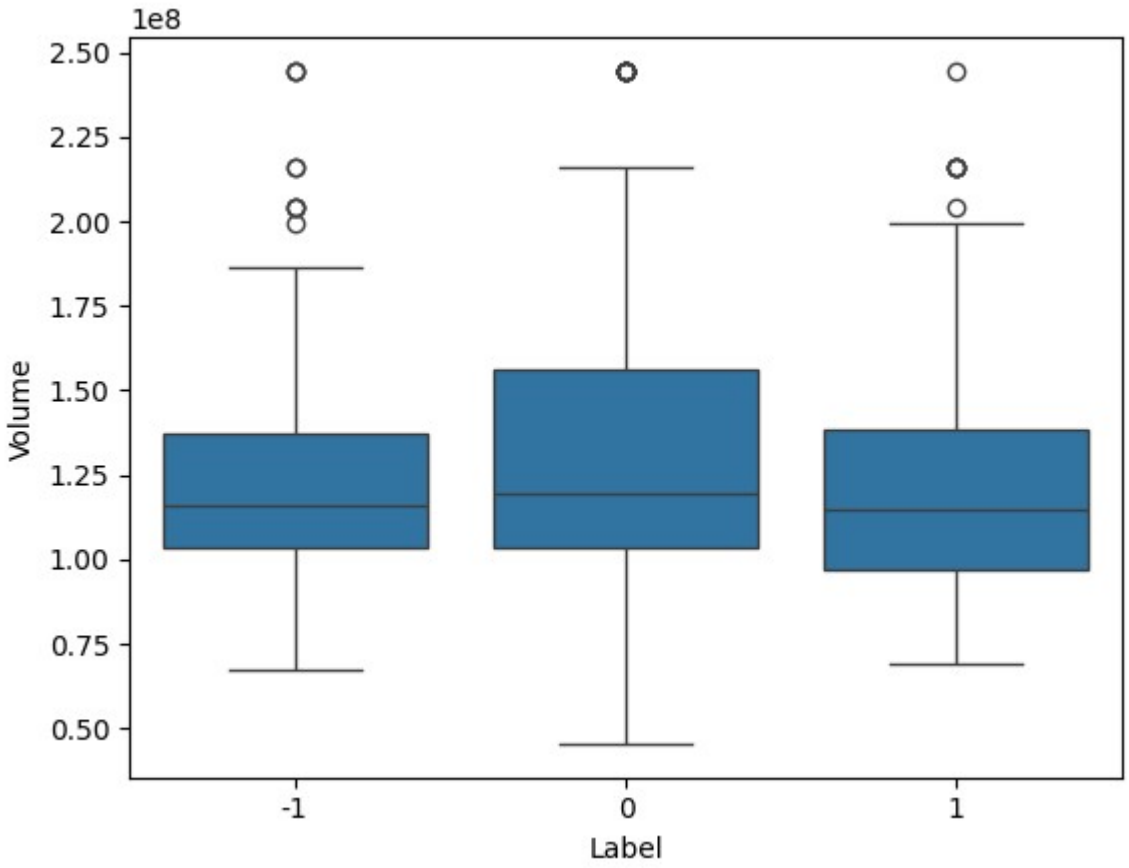


The box plots reveal that while there might be a slight upward trend in median prices from negative to positive sentiment, the overall distributions for each price type (Open, High, Low, Close) are very similar across all three sentiment labels. The interquartile ranges (boxes) and whiskers largely overlap.

This suggests that the absolute price level of the stock is not strongly differentiated by the sentiment of the news on that day. While sentiment might influence price movement (change from the previous day), it doesn't appear to strongly correlate with the absolute price range the stock is trading at when the news is released.

Label vs Volume

In [ ]:
```
sns.boxplot(data=data, x="Label", y="Volume");
```



The boxes and whiskers for positive and negative sentiment show higher volume ranges than neutral days. This strongly suggests that news with a distinct positive or negative sentiment tends to generate higher trading activity (volume) than days with neutral or routine news. Significant news, whether good or bad, appears to drive more investor action than the absence of clear sentiment drivers.

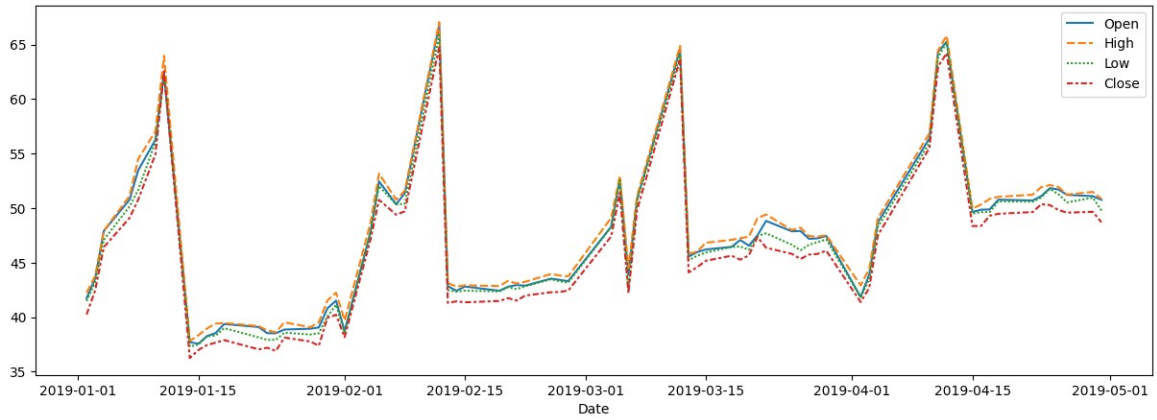
Date vs Price

In [ ]:
```
# Group the 'stocks' DataFrame by the 'Date' column

stock_daily = data.groupby('Date').agg(
    {
        'Open': 'mean',
        'High': 'mean',
        'Low': 'mean',
        'Close': 'mean',
        'Volume': 'mean',
    }
).reset_index()

stock_daily.set_index('Date', inplace=True)
stock_daily.head()
```

Out[ ]:

Date	Open	High	Low	Close	Volume
2019-01-02	41.740002	42.244999	41.482498	40.246914	130672400.0
2019-01-03	43.570000	43.787498	43.222500	42.470604	103544800.0
2019-01-04	47.910000	47.919998	47.095001	46.419842	111448000.0
2019-01-07	50.792500	51.122501	50.162498	49.110790	109012000.0
2019-01-08	53.474998	54.507500	51.685001	50.787209	216071600.0

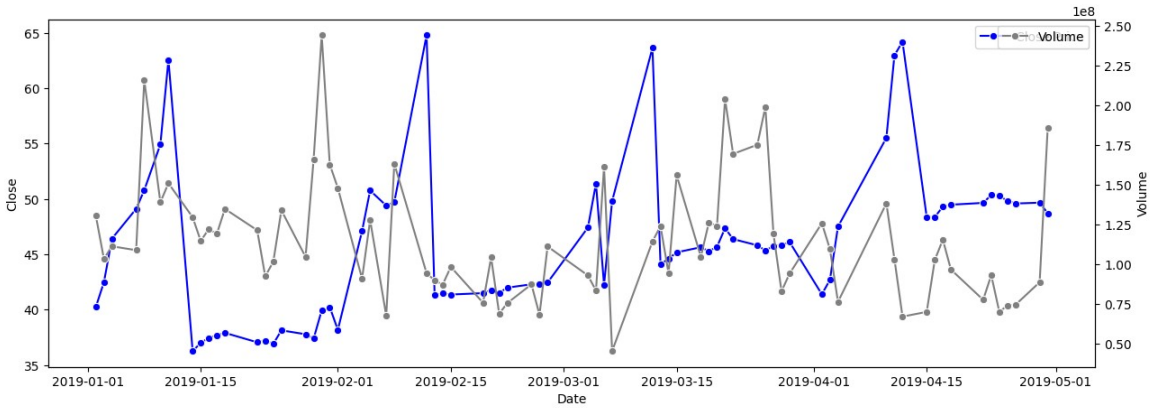
In [ ]:
```
plt.figure(figsize=(15,5))
sns.lineplot(stock_daily.drop("Volume", axis=1));
```



All four price lines track each other extremely closely throughout the period. Insight: This visual confirms the earlier observation from the KDE plot that the difference between the Open, High, Low, and Close prices within a single day is generally small relative to the overall movement of the stock price over several days or weeks. It shows that daily volatility (the range within a day) is relatively contained during this period, despite the price itself undergoing noticeable trends and fluctuations.

Volume vs Close Price

In [ ]:
```
# Create a figure and axis
fig, ax1 = plt.subplots(figsize=(15,5))

# Lineplot on primary y-axis
sns.lineplot(data=stock_daily.reset_index(), x='Date', y='Close', ax=ax1, c
olor='blue', marker='o', label='Close Price')

# Create a secondary y-axis
ax2 = ax1.twinx()

# Lineplot on secondary y-axis
sns.lineplot(data=stock_daily.reset_index(), x='Date', y='Volume', ax=ax2,
color='gray', marker='o', label='Volume')

ax1.legend(bbox_to_anchor=(1,1));
```



By comparing the Close Price (blue line) movements with the Volume (grey line) fluctuations, we can observe their relationship. There appears to be a strong tendency for significant price changes or periods of increased price volatility in the Close Price (sharp movements or turns in the blue line) to coincide with spikes in Volume (peaks in the grey line). This suggests that when the stock price makes notable moves, whether up or down, it's often backed by higher trading activity and market participation, compared to days where the price is more stable and volume is lower.

5. DATA PREPROCESSING

5.1. Train-test-validation split

In [ ]:
```
# separating predictors from label
X = data.drop('Label', axis=1)
y = data['Label']

# Split data into training and temporary sets
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3, sh
uffle=True, random_state=gbl_random_seed, stratify=y)

# Split the temporary set into validation and test sets
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=
0.5, shuffle=True, random_state=gbl_random_seed, stratify=y_temp)

print("Train data shape is: ", X_train.shape, ", Validation data shape is:
",
X_val.shape, ", Test data shape is: ", X_test.shape)

print("Train label shape is: ", y_train.shape, ", Validation label shape is
: ", y_val.shape, ", Test label shape is: ", y_test.shape)
```

```
Train data shape is:  (244, 8) , Validation data shape is:  (52, 8) , Test
data shape is:  (53, 8)
Train label shape is:  (244,) , Validation label shape is:  (52,) , Test la
bel shape is:  (53,)
```

6. WORD EMBEDDINGS

6.1. Word2Vec

In [ ]:
```
# Creating a list of all words in our dataset
words_list = [item.split(" ") for item in data['News'].values]
```

In [ ]:
```
# Creating an instance of Word2Vec
vec_size = 300
model_W2V = Word2Vec(words_list, vector_size = vec_size, min_count = 1, win
dow=5, workers = 6)
```

In [ ]:
```
# Checking the size of the vocabulary
print("Length of the vocabulary is", len(list(model_W2V.wv.key_to_index)))
```

```
Length of the vocabulary is 4682
```

Let's check out a few word embeddings obtained using the model.

In [ ]:
```
# Checking the word embedding of a random word
word = "stock"
model_W2V.wv[word]
```

Out[ ]:
```
array([-3.60552734e-03,  3.75152379e-02,  8.67900904e-03,  1.16152335e-02,
        6.16652193e-04, -5.68401217e-02,  2.95417458e-02,  8.87997746e-02,
        3.70907830e-03, -2.10442599e-02,  6.62047789e-03, -2.11855937e-02,
       -5.00079151e-03,  1.54764205e-02, -2.58149300e-02, -2.55615730e-02,
        2.20414177e-02, -3.41607723e-03,  5.76064130e-03, -2.61103716e-02,
       -2.28942856e-02,  7.79024651e-03,  2.99761817e-02,  1.53650576e-02,
        2.43012439e-02,  5.26128453e-04, -3.56147662e-02,  9.38795879e-03,
       -2.87450999e-02, -4.04495075e-02,  1.04256403e-02, -2.73104962e-02,
        3.91019043e-03, -9.68637317e-03, -1.94812252e-03,  2.32211873e-02,
        1.30461836e-02, -3.39261964e-02, -2.43379851e-03, -1.26176728e-02,
       -1.96498167e-02,  4.24254220e-03,  5.80444233e-04, -2.09361408e-02,
        2.00255159e-02,  3.78683582e-02,  8.48372839e-03,  1.69585627e-02,
       -3.05563939e-04,  2.32116692e-02,  1.16562890e-02, -1.11516826e-02,
       -2.21364684e-02,  1.43441400e-02, -7.34119862e-03,  2.68976148e-02,
        1.86079536e-02,  1.07444543e-03,  7.64503656e-03,  1.94091420e-03,
       -1.33173559e-02, -1.01526426e-02, -3.79140579e-06,  1.29501987e-02,
       -1.56615814e-03,  1.70756802e-02,  6.79301936e-03,  1.21196536e-02,
       -2.75654420e-02, -9.37538128e-03, -1.01732113e-03,  2.37016939e-02,
        3.52227762e-02, -2.08093487e-02,  8.20961781e-03,  1.34834591e-02,
       -3.05302907e-02,  4.09937138e-03, -1.62403844e-02,  2.58945543e-02,
       -1.32562891e-02, -2.57009901e-02,  3.43833375e-03,  6.38053641e-02,
        5.05309599e-03, -2.40067905e-03, -1.00890426e-02,  7.21329544e-03,
        3.94586548e-02,  1.27624385e-02,  3.26938666e-02, -1.43356752e-02,
        1.87290609e-02, -5.79607382e-04,  3.65770608e-02,  3.51790190e-02,
        2.79466212e-02, -1.53981317e-02, -1.72301419e-02,  2.12487616e-02,
        4.47804527e-03,  3.88767989e-03,  2.68626381e-02,  1.51910866e-02,
        4.98746987e-03, -1.97122488e-02, -1.40835578e-02,  1.92850139e-02,
       -2.04761140e-02,  5.78050921e-03, -4.74286675e-02, -2.64757518e-02,
       -2.91006011e-03,  2.65973154e-02,  1.66025683e-02,  2.56713089e-02,
       -4.33653640e-03, -2.63485359e-03,  5.29135019e-02, -4.41830270e-02,
        9.07831918e-03,  2.54847836e-02,  2.50082798e-02, -5.26682613e-03,
       -2.01151781e-02,  2.01529283e-02,  1.49909928e-02, -3.74549888e-02,
       -6.75684132e-04,  1.76620260e-02,  1.36093674e-02,  4.22651246e-02,
        2.12977808e-02, -4.04870883e-02,  7.17986841e-03,  2.20364314e-02,
       -9.92983393e-03, -3.09813339e-02, -4.22047004e-02, -4.23801094e-02,
        1.53171718e-02, -3.83907631e-02, -1.52169382e-02,  1.52582619e-02,
        1.38227688e-02, -1.48958443e-02, -5.98044544e-02, -9.22312681e-03,
        2.90968735e-02, -2.03759447e-02,  7.38509698e-03, -4.85306680e-02,
       -1.52151110e-02, -1.25422971e-02,  1.11489706e-02,  2.69066487e-02,
       -3.78967077e-02, -2.03512069e-02, -2.50532175e-03,  3.09885442e-02,
        9.14506987e-03,  2.63574217e-02, -4.42229770e-02,  2.96819191e-02,
       -1.51964668e-02,  5.24787093e-03, -1.05262138e-02,  6.15055615e-04,
        3.09669878e-03,  5.83194718e-02, -3.23513127e-03,  6.93271123e-03,
        2.80782245e-02,  2.38506845e-03,  6.23628125e-03,  5.35528129e-03,
        2.49547535e-03, -2.28531063e-02, -5.68327494e-03, -9.04380064e-03,
       -2.99849492e-02,  1.94951966e-02, -3.96043360e-02, -1.92067232e-02,
       -1.24395471e-02, -3.84943327e-04,  4.41150181e-02,  3.76603119e-02,
        2.02457346e-02, -4.21996340e-02,  1.48030585e-02, -6.36193156e-03,
       -4.13958803e-02,  7.35470885e-03,  5.62790083e-03, -3.01651172e-02,
       -1.66558067e-03, -3.10401414e-02,  1.66894738e-02, -5.09596523e-03,
       -2.84480974e-02,  1.74091253e-02,  2.52480945e-03, -1.73683688e-02,
        5.76480012e-03, -1.07892537e-02, -1.04831122e-02,  2.01896224e-02,
       -1.77712701e-02, -1.08257141e-02, -7.04803364e-03, -4.44901437e-02,
       -8.00743699e-03, -1.03478534e-02,  3.58930752e-02, -3.51857170e-02,
       -2.98447497e-02, -5.57355918e-02, -3.90006378e-02, -3.08549386e-02,
        1.70591772e-02,  1.20237535e-02, -1.11254491e-02, -2.50633694e-02,
```

```
 -1.73405893e-02, -1.57492347e-02,  9.97961033e-03, -8.18408385e-04,
 -2.47097034e-02,  1.22414930e-02,  3.56831029e-02, -1.90272331e-02,
 -2.84696650e-02,  3.15025412e-02, -1.87502671e-02,  2.18273532e-02,
  3.28830630e-03,  8.63721408e-03,  7.05289841e-03, -5.80557436e-02,
  9.42753442e-03, -2.15269979e-02, -2.50588376e-02,  4.39940579e-03,
 -9.89176799e-04, -4.49002385e-02, -9.66655277e-03,  1.19777285e-02,
  3.99110233e-03,  2.31667440e-02,  1.11493971e-02,  5.51586854e-04,
  2.85945665e-02, -4.84872935e-03, -3.61216106e-02, -2.04755627e-02,
  5.24215102e-02,  1.16809765e-02, -5.01320772e-02, -3.09325736e-02,
  2.42666546e-02,  2.03265063e-02,  1.76762566e-02, -5.67058176e-02,
 -4.02253456e-02,  5.57844061e-03,  1.70723666e-02,  1.89292282e-02,
 -3.17625180e-02,  1.29134497e-02, -1.96342487e-02,  1.69314502e-03,
 -5.16943401e-03, -6.83990214e-03,  3.09725367e-02,  1.72189306e-02,
  1.89149305e-02,  1.50486212e-02, -3.41522880e-02, -5.96719701e-03,
  1.54460547e-02, -2.21023336e-03, -1.82248037e-02,  2.01048013e-02,
 -3.30413668e-03, -1.19062606e-03, -4.66542467e-02,  2.36671548e-02,
  3.48865846e-03,  4.01100107e-02,  1.43842017e-02,  5.03031276e-02,
  3.47766429e-02,  3.09619051e-03,  4.56034206e-02,  5.78150190e-02,
  7.30892643e-03, -2.58739479e-02,  3.12350597e-02, -1.84775311e-02],
dtype=float32)
```

In [ ]:
```
# Checking the word embedding of a random word
word = "economy"
model_W2V.wv[word]
```

Out[ ]:
```
array([-1.08419312e-03,  5.17156720e-03,  3.28308856e-03,  1.89649127e-03,
       -1.21265720e-03, -1.12755271e-02,  6.77801808e-03,  1.57075226e-02,
       -1.98244979e-03, -2.02365522e-03, -2.34696874e-03, -3.50481109e-03,
        7.45038269e-04, -2.20140704e-04, -4.24796343e-03, -1.51149696e-03,
        7.02976715e-03,  2.00308673e-03,  3.59008764e-03, -2.87491502e-03,
       -5.78742707e-03,  2.76061730e-03,  3.99027439e-03, -7.82807590e-04,
        1.22053165e-03, -1.23992085e-03, -7.10784551e-03,  2.39096070e-03,
       -6.61150273e-03, -9.52598546e-03, -1.93554803e-03, -6.54514413e-03,
        2.79662828e-03, -2.82211788e-03, -2.70032161e-03,  5.88063523e-03,
       -8.68487987e-05, -5.29299490e-03, -4.61042255e-05, -2.98520806e-03,
       -5.66843478e-03,  8.53706209e-04,  1.50856772e-03, -3.91782448e-03,
        3.92284337e-03,  3.75915458e-03, -6.76601136e-04,  9.89668886e-04,
       -1.10036402e-04,  9.53599229e-04,  4.61896835e-03, -4.42899158e-03,
       -4.27390123e-03,  5.10202372e-04, -2.92728702e-03,  6.95579266e-03,
        1.25226658e-03,  3.45412758e-03,  1.34345086e-03, -1.31606823e-03,
       -5.42952027e-03,  2.02926947e-03,  1.68163655e-03, -7.25036778e-04,
       -5.13370731e-04,  2.34112702e-03,  2.02620844e-03,  2.15927325e-03,
       -5.92576200e-03, -1.99224893e-03, -1.97876198e-03,  5.93876513e-03,
        6.97574625e-03, -1.20612048e-03,  4.11124201e-03, -1.66791677e-03,
       -2.89696781e-03,  7.19012809e-04, -5.53972379e-04,  5.63051505e-03,
       -7.52324006e-04, -8.59855092e-04, -6.39757840e-04,  1.04987202e-02,
        2.82057398e-03, -1.25715183e-03, -3.14995879e-03,  1.46175164e-03,
        8.90484173e-03, -7.42069387e-04,  2.87533575e-03, -4.96240053e-03,
        1.98325282e-03, -6.50878792e-05,  3.74076748e-03,  4.76949988e-03,
        2.88725225e-03, -2.06878409e-03, -5.67899970e-03,  6.29442395e-04,
        1.46410393e-03, -2.25066673e-03,  6.87537994e-03,  4.55183536e-03,
        1.55076932e-03, -5.23392856e-03, -2.17393157e-03,  4.58651502e-03,
       -3.72766866e-03,  2.39178888e-03, -7.83663616e-03, -2.29699840e-03,
       -1.62038463e-03,  5.63741801e-03,  4.90199821e-03,  9.62540857e-04,
       -2.46309349e-03,  2.78929854e-03,  8.34968872e-03, -8.82410258e-03,
        1.05466193e-03,  2.25758716e-03,  3.84094380e-03,  1.52156776e-04,
       -2.36839685e-03,  2.45029898e-03,  2.38295877e-03, -3.22403316e-03,
        2.08765222e-03,  1.87165895e-03, -1.19771587e-03,  8.89557879e-03,
        3.56847490e-03, -5.77066652e-03,  2.50073755e-03,  5.05304663e-03,
       -1.24589936e-03, -2.77643511e-03, -6.76833140e-03, -9.79828835e-03,
        1.34279532e-03, -5.58049697e-03, -3.05972039e-03,  4.70921397e-03,
       -9.72984053e-05,  9.68401204e-04, -1.05664711e-02, -4.87981102e-04,
        6.07431820e-03, -1.73549794e-04,  4.39030235e-04, -9.17815417e-03,
        4.76796413e-04, -1.95033604e-03, -5.11424732e-05,  4.53358889e-03,
       -9.40628629e-03, -2.92240409e-03,  3.47616477e-03,  1.90483127e-03,
        2.56262312e-04,  1.06775609e-03, -7.73609662e-03,  1.58753747e-03,
       -1.11372070e-03,  2.52175494e-04, -4.55308659e-03,  1.79687142e-03,
       -1.64697028e-03,  8.34537111e-03, -2.08022585e-03,  4.04206105e-03,
        1.78758043e-03, -4.04029677e-04,  2.03672936e-03, -2.48942268e-03,
       -1.65797432e-03, -7.07320942e-06, -3.35397222e-03,  1.79501250e-03,
       -1.02491106e-03,  5.94531826e-04, -7.77538912e-03, -2.23498652e-03,
        3.52331524e-04, -2.58509582e-03,  7.73836765e-03,  6.67189248e-03,
        5.28452452e-03, -8.87056068e-03,  2.56668020e-04,  2.03693140e-04,
       -6.26146188e-03, -1.39077124e-03, -1.18931755e-03, -6.15742570e-03,
        8.28915276e-04, -6.82335021e-03,  3.29932460e-04, -9.60932754e-04,
       -4.78433957e-03,  3.84278852e-03,  8.90688854e-04, -3.32871056e-03,
       -3.83910810e-05, -2.19620578e-03, -2.44432013e-03, -9.22783584e-05,
       -4.90058679e-03, -2.34226417e-03, -1.23763108e-03, -4.09611408e-03,
        1.17615378e-03,  5.20173111e-04,  4.57247067e-03, -2.45795352e-03,
       -7.04743713e-03, -1.08509557e-02, -4.81142057e-03, -4.22720844e-03,
        4.96021798e-03,  3.19734681e-04, -5.05023869e-03, -3.32275778e-03,
```

```
       -3.25318822e-03, -1.50224904e-03, -1.80203712e-03,  9.76127572e-04,
       -3.35600181e-03,  3.42251896e-03,  6.53074170e-03, -4.67028283e-03,
       -6.29469613e-03,  6.89996965e-03, -5.59856044e-03,  1.05838035e-03,
       -2.71864259e-03,  2.77111633e-03, -1.31289638e-03, -9.23692342e-03,
        4.96807043e-03, -2.74565304e-03, -5.26434323e-03, -6.64293766e-04,
       -1.66161754e-03, -7.71095371e-03, -1.83527020e-03,  1.57145457e-03,
        2.89926585e-03,  1.60788608e-04, -4.56353795e-04, -3.84830229e-04,
        5.24909189e-03, -2.37963075e-04, -5.42141497e-03, -4.68290970e-03,
        6.63563190e-03,  4.06016689e-03, -1.02035925e-02, -2.74562952e-03,
        2.08629342e-03,  3.28054791e-03,  3.33326840e-04, -5.76964999e-03,
       -5.21820271e-03,  2.90663075e-03,  3.90373124e-03,  2.75929575e-03,
       -1.76110421e-03,  3.31403245e-03, -5.41434716e-03,  3.76288244e-03,
       -1.20692409e-03,  6.35819277e-04,  2.17075041e-03,  4.21312544e-03,
        4.34716232e-03,  4.95331921e-03, -2.23135925e-03, -3.95078119e-03,
        4.66080615e-03,  1.00176039e-04, -2.81787082e-03,  4.11627628e-03,
       -9.87631734e-04, -1.92296656e-03, -6.38894318e-03,  2.83913710e-03,
        4.22375160e-04,  8.83853715e-03,  1.29271776e-03,  6.05648104e-03,
        2.88042589e-03, -1.69926172e-03,  7.51767354e-03,  9.60577559e-03,
        3.32430191e-03, -6.80895709e-03,  1.82524393e-03, -7.97792571e-04],
      dtype=float32)
```

In [ ]:
```
# Retrieving the words present in the Word2Vec model's vocabulary
words = list(model_W2V.wv.key_to_index.keys())

# Retrieving word vectors for all the words present in the model's vocabula
ry
wvs = model_W2V.wv[words].tolist()

# Creating a dictionary of words and their corresponding vectors
word_vector_dict = dict(zip(words, wvs))
```

In [ ]:
```
def average_vectorizer_Word2Vec(doc):
    # Initializing a feature vector for the sentence
    feature_vector = np.zeros((vec_size,), dtype="float64")

    # Creating a list of words in the sentence that are present in the mode
l vocabulary
    words_in_vocab = [word for word in doc.split() if word in words]

    # adding the vector representations of the words
    for word in words_in_vocab:
        feature_vector += np.array(word_vector_dict[word])

    # Dividing by the number of words to get the average vector
    if len(words_in_vocab) != 0:
        feature_vector /= len(words_in_vocab)

    return feature_vector
```

In [ ]:
```
# creating a dataframe of the vectorized documents
start = time.time()

# dataset to be used for word2vec models
X_train_wv = pd.DataFrame(X_train["News"].apply(average_vectorizer_Word2Ve
c).tolist(), columns=['Feature '+str(i) for i in range(vec_size)])
X_val_wv = pd.DataFrame(X_val["News"].apply(average_vectorizer_Word2Vec).to
list(), columns=['Feature '+str(i) for i in range(vec_size)])
X_test_wv = pd.DataFrame(X_test["News"].apply(average_vectorizer_Word2Ve
c).tolist(), columns=['Feature '+str(i) for i in range(vec_size)])

end = time.time()
print('Time taken ', (end-start))
```

```
Time taken  2.0359890460968018
```

In [ ]:
```
print("train shape:", X_train_wv.shape, "; val shape:", X_val_wv.shape, ";
test shape:", X_test_wv.shape)
```

```
train shape: (244, 300) ; val shape: (52, 300) ; test shape: (53, 300)
```

6.2. GloVe

In [ ]:
```
# load the Stanford GloVe model
filename = '/content/drive/MyDrive/Education/DBA AI/M06 NLP/project 06_Stoc
k mkt news/glove.6B.100d.txt.word2vec'
glove_model = KeyedVectors.load_word2vec_format(filename, binary=False)
```

In [ ]:
```
# Checking the size of the vocabulary
print("Length of the vocabulary is", len(glove_model.index_to_key))
```

```
Length of the vocabulary is 400000
```

Let's check out a few word embeddings.

In [ ]:
```
# Checking the word embedding of a random word
word = "stock"
glove_model[word]
```

Out[ ]:
```
array([ 8.6341e-01,  6.9648e-01,  4.5794e-02, -9.5708e-03, -2.5498e-01,
       -7.4666e-01, -2.2086e-01, -4.4615e-01, -1.0423e-01, -9.9931e-01,
        7.2550e-02,  4.5049e-01, -5.9912e-02, -5.7837e-01, -4.6540e-01,
        4.3429e-02, -5.0570e-01, -1.5442e-01,  9.8250e-01, -8.1571e-02,
        2.6523e-01, -2.3734e-01,  9.7675e-02,  5.8588e-01, -1.2948e-01,
       -6.8956e-01, -1.2811e-01, -5.2265e-02, -6.7719e-01,  3.0190e-02,
        1.8058e-01,  8.6121e-01, -8.3206e-01, -5.6887e-02, -2.9578e-01,
        4.7180e-01,  1.2811e+00, -2.5228e-01,  4.9557e-02, -7.2455e-01,
        6.6758e-01, -1.1091e+00, -2.0493e-01, -5.8669e-01, -2.5375e-03,
        8.2777e-01, -4.9102e-01, -2.6475e-01,  4.3015e-01, -2.0516e+00,
       -3.3208e-01,  5.1845e-02,  5.2646e-01,  8.7452e-01, -9.0237e-01,
       -1.7366e+00, -3.4727e-01,  1.6590e-01,  2.7727e+00,  6.5756e-02,
       -4.0363e-01,  3.8252e-01, -3.0787e-01,  5.9202e-01,  1.3468e-01,
       -3.3851e-01,  3.3646e-01,  2.0950e-01,  8.5905e-01,  5.1865e-01,
       -1.0657e+00, -2.6371e-02, -3.1349e-01,  2.3231e-01, -7.0192e-01,
       -5.5737e-01, -2.3418e-01,  1.3563e-01, -1.0016e+00, -1.4221e-01,
        1.0372e+00,  3.5880e-01, -4.2608e-01, -1.9386e-01, -3.7867e-01,
       -6.9646e-01, -3.9989e-01, -5.7782e-01,  1.0132e-01,  2.0123e-01,
       -3.7153e-01,  5.0837e-01, -3.7758e-01, -2.6205e-01, -9.3676e-01,
        1.0053e+00,  8.4393e-01, -2.4698e-01,  1.7339e-01,  9.4473e-01],
      dtype=float32)
```

In [ ]:
```
# Checking the word embedding of a random word
word = "economy"
glove_model[word]
```

Out[ ]:
```
array([-0.19382  ,  1.017    ,  1.076    ,  0.02954  , -0.39192  ,
       -1.3891   , -0.87873  , -0.63162  ,  0.9643   , -0.43035  ,
       -0.34868  ,  0.22736  , -0.40296  ,  0.15641  , -0.16813  ,
       -0.15343  , -0.15799  , -0.27612  ,  0.18088  , -0.28386  ,
        0.49847  ,  0.29864  ,  0.32353  ,  0.18108  , -0.59623  ,
       -0.54165  , -0.70019  , -0.64956  , -0.69063  ,  0.18084  ,
       -0.38581  ,  0.56086  , -0.40313  , -0.38777  , -0.70615  ,
        0.20657  ,  0.34171  , -0.23393  , -0.35882  , -0.2201   ,
       -0.76182  , -1.2047   ,  0.4339   ,  1.1656   ,  0.1836   ,
       -0.21601  ,  0.93198  , -0.059616 , -0.11624  , -1.3259   ,
       -0.79772  , -0.0074957, -0.0889   ,  1.4749   ,  0.31157  ,
       -2.2952   , -0.058351 ,  0.39353  ,  1.4983   ,  0.74023  ,
       -0.20109  ,  0.098124 , -0.73081  , -0.32294  ,  0.16703  ,
        0.87431  , -0.041624 , -0.51022  ,  1.0737   , -0.4257   ,
        1.0581   ,  0.19859  , -0.60087  , -0.33906  ,  0.60243  ,
       -0.091581 , -0.47201  ,  0.74933  , -0.60168  , -0.44178  ,
        0.77391  ,  0.81114  , -1.2889   ,  0.32055  , -0.36117  ,
       -0.88078  ,  0.055524 , -0.26837  , -0.33688  , -1.4359   ,
        0.85666  ,  0.32025  , -0.15361  , -0.30208  , -0.38208  ,
        0.30508  ,  0.75374  , -0.68041  ,  0.98619  , -0.19628  ],
      dtype=float32)
```

In [ ]:
```
# Retrieving the words present in the GloVe model's vocabulary
glove_words = glove_model.index_to_key

# Creating a dictionary of words and their corresponding vectors
glove_word_vector_dict = dict(zip(glove_model.index_to_key,list(glove_mode
l.vectors)))
```

In [ ]:
```
vec_size=100
```

In [ ]:
```
def average_vectorizer_GloVe(doc):
    # Initializing a feature vector for the sentence
    feature_vector = np.zeros((vec_size,), dtype="float64")

    # Creating a list of words in the sentence that are present in the mode
l vocabulary
    words_in_vocab = [word for word in doc.split() if word in glove_words]

    # adding the vector representations of the words
    for word in words_in_vocab:
        feature_vector += np.array(glove_word_vector_dict[word])

    # Dividing by the number of words to get the average vector
    if len(words_in_vocab) != 0:
        feature_vector /= len(words_in_vocab)

    return feature_vector
```

In [ ]:
```
# creating a dataframe of the vectorized documents
start = time.time()

# dataset to be used for GloVe models
X_train_gl = pd.DataFrame(X_train["News"].apply(average_vectorizer_GloVe).t
olist(), columns=['Feature '+str(i) for i in range(vec_size)]) #Complete th
e code to apply GloVe on 'News' column
X_val_gl = pd.DataFrame(X_val["News"].apply(average_vectorizer_GloVe).tolis
t(), columns=['Feature '+str(i) for i in range(vec_size)]) #Complete the co
de to apply GloVe on 'News' column
X_test_gl = pd.DataFrame(X_test["News"].apply(average_vectorizer_GloVe).tol
ist(), columns=['Feature '+str(i) for i in range(vec_size)]) #Complete the
code to apply GloVe on 'News' column

end = time.time()
print('Time taken ', (end-start))
```

```
Time taken  26.344531297683716
```

In [ ]:
```
# printing the shapes of the final dataframes
print("train shape:", X_train_gl.shape, "; val shape:", X_val_gl.shape, ";
test shape:", X_test_gl.shape)
```

```
train shape: (244, 100) ; val shape: (52, 100) ; test shape: (53, 100)
```

6.3. Sentence transformer

6.3.1. Defining the model

In [ ]:
```
#Defining the model
model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')
```

```
Xet Storage is enabled for this repo, but the 'hf_xet' package is not insta
lled. Falling back to regular HTTP download. For better performance, instal
l the package with: `pip install huggingface_hub[hf_xet]` or `pip install h
f_xet`
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo,
but the 'hf_xet' package is not installed. Falling back to regular HTTP dow
nload. For better performance, install the package with: `pip install huggi
ngface_hub[hf_xet]` or `pip install hf_xet`
```

6.3.2. Encoding the dataset

In [ ]:
```
# setting the device to GPU if available, else CPU
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

In [ ]:
```
# encoding the dataset
start = time.time()

# datasets for sentence transformer models
X_train_st = model.encode(X_train["News"].values, show_progress_bar=True, d
evice=device) # apply Sentence Transformer on 'News' column
X_val_st = model.encode(X_val["News"].values, show_progress_bar=True, devic
e=device) # apply Sentence Transformer on 'News' column
X_test_st = model.encode(X_test["News"].values, show_progress_bar=True, dev
ice=device) # apply Sentence Transformer on 'News' column

end = time.time()
print("Time taken ",(end-start))
```

```
Time taken  1.085937738418579
```

In [ ]:
```
print("train shape:", X_train_st.shape, "; val shape:", X_val_st.shape, "; test shape:", X_test_st.shape) # print the shapes of the final dataframes
```

```
train shape: (244, 384) ; val shape: (52, 384) ; test shape: (53, 384)
```

- Each news content has been converted to a 384-dimensional vector.

7. SENTIMENT ANALYSIS

F1-score will be used to select the best model because it balances Precision and Recall, which are both crucial. Precision measures the accuracy of positive predictions, while Recall measures how well the model finds all actual positives. Balancing these is essential for reliable sentiment classification, where both false positives and false negatives can be costly.

7.1. Utility functions

In [ ]:
```
def plot_confusion_matrix(model, predictors, target, dataset_name):
    """
    Plot a confusion matrix to visualize the performance of a classificatio
n model.

    Parameters:
    actual (array-like): The true labels.
    predicted (array-like): The predicted labels from the model.

    Returns:
    None: Displays the confusion matrix plot.
    """
    pred = model.predict(predictors)  # Make predictions using the classifi
er.

    cm = confusion_matrix(target, pred)  # Compute the confusion matrix.

    plt.figure(figsize=(5, 4))  # Create a new figure with a specified siz
e.
    label_list = [0, 1,-1]  # Define the labels for the confusion matrix.
    sns.heatmap(cm, annot=True, fmt='.0f', cmap='Blues', xticklabels=label_
list, yticklabels=label_list)
    # Plot the confusion matrix using a heatmap with annotations.

    plt.ylabel('Actual')  # Label for the y-axis.
    plt.xlabel('Predicted')  # Label for the x-axis.
    plt.title("Confusion Matrix for dataset: " + dataset_name)  # Title of
the plot.
    plt.show()  # Display the plot.
```

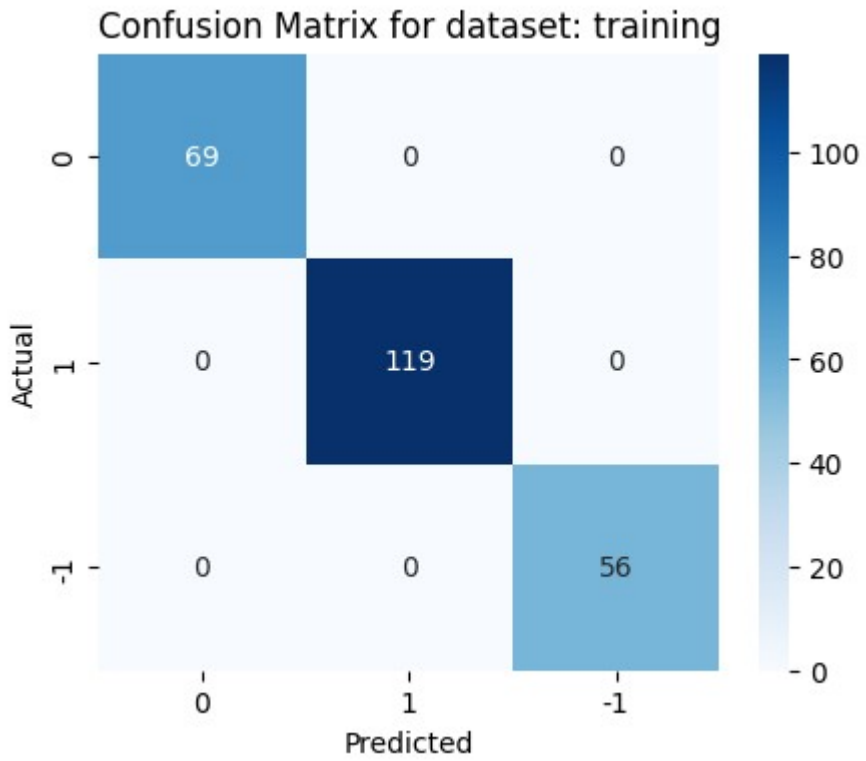
In [ ]:
```
def model_performance_classification_sklearn(model, predictors, target):
    """
    Compute various performance metrics for a classification model using sk
learn.

    Parameters:
    model (sklearn classifier): The classification model to evaluate.
    predictors (array-like): The independent variables used for prediction
s.
    target (array-like): The true labels for the dependent variable.

    Returns:
    pandas.DataFrame: A DataFrame containing the computed metrics (Accurac
y, Recall, Precision, F1-score).
    """
    pred = model.predict(predictors)  # Make predictions using the classifi
er.

    acc = accuracy_score(target, pred)  # Compute Accuracy.
    recall = recall_score(target, pred,average='weighted')  # Compute Recal
l.
    precision = precision_score(target, pred,average='weighted')  # Compute
Precision.
    f1 = f1_score(target, pred,average='weighted')  # Compute F1-score.

    # Create a DataFrame to store the computed metrics.
    df_perf = pd.DataFrame(
        {
            "Accuracy": [acc],
            "Recall": [recall],
            "Precision": [precision],
            "F1": [f1],
        }
    )

    return df_perf  # Return the DataFrame with the metrics.
```

7.2. Base Model - Word2Vec

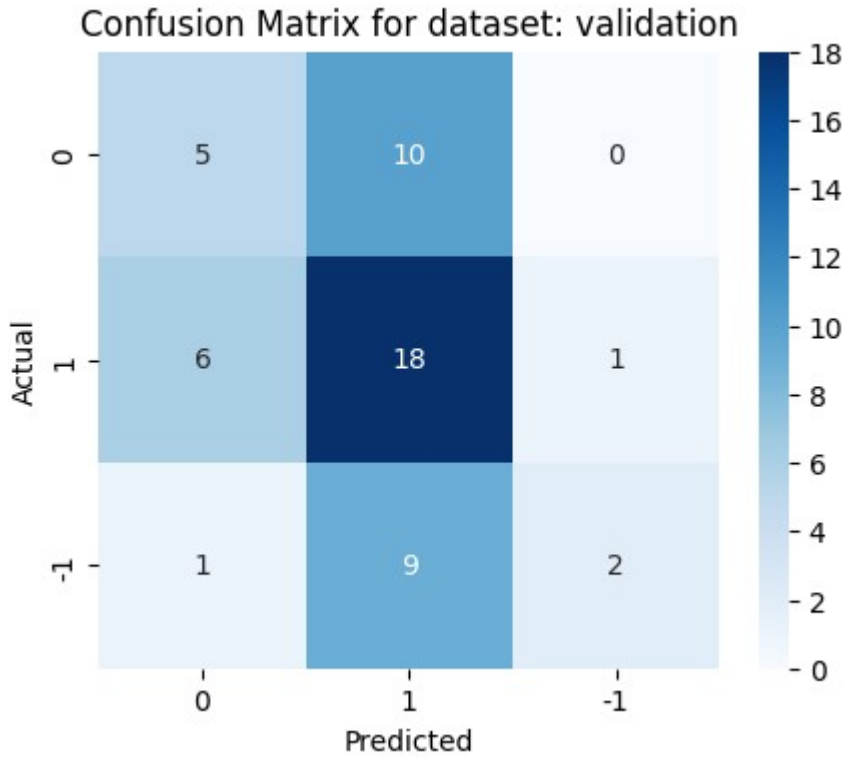
In [ ]:
```
# Building the model

#base_wv = GradientBoostingClassifier(random_state = 42)
base_wv = RandomForestClassifier(random_state= gbl_random_seed)
# base_wv = DecisionTreeClassifier(random_state=42)

# Fitting on train data
base_wv.fit(X_train_wv, y_train)
```

Out[ ]:
```
RandomForestClassifier(random_state=42)
```

In [ ]:
```
plot_confusion_matrix(base_wv,X_train_wv,y_train, "training")
```



In [ ]:
```
plot_confusion_matrix(base_wv,X_val_wv,y_val, "validation")
```



In [ ]:
```
#Calculating different metrics on training data
base_train_wv = model_performance_classification_sklearn(base_wv,X_train_w
v,y_train)
print("Training performance:\n", base_train_wv)
```

```
Training performance:
    Accuracy  Recall  Precision   F1
0       1.0     1.0        1.0  1.0
```

In [ ]:
```
#Calculating different metrics on validation data
base_val_wv = model_performance_classification_sklearn(base_wv,X_val_wv,y_v
al)
print("Validation performance:\n",base_val_wv)
```

```
Validation performance:
    Accuracy    Recall  Precision        F1
0  0.480769  0.480769   0.507926  0.447532
```

Model performance drops significantly on the validation data for all metrics (below 50%). The validation confusion matrix clearly shows many misclassifications across all classes. This indicates that the model is overfitting the training data.

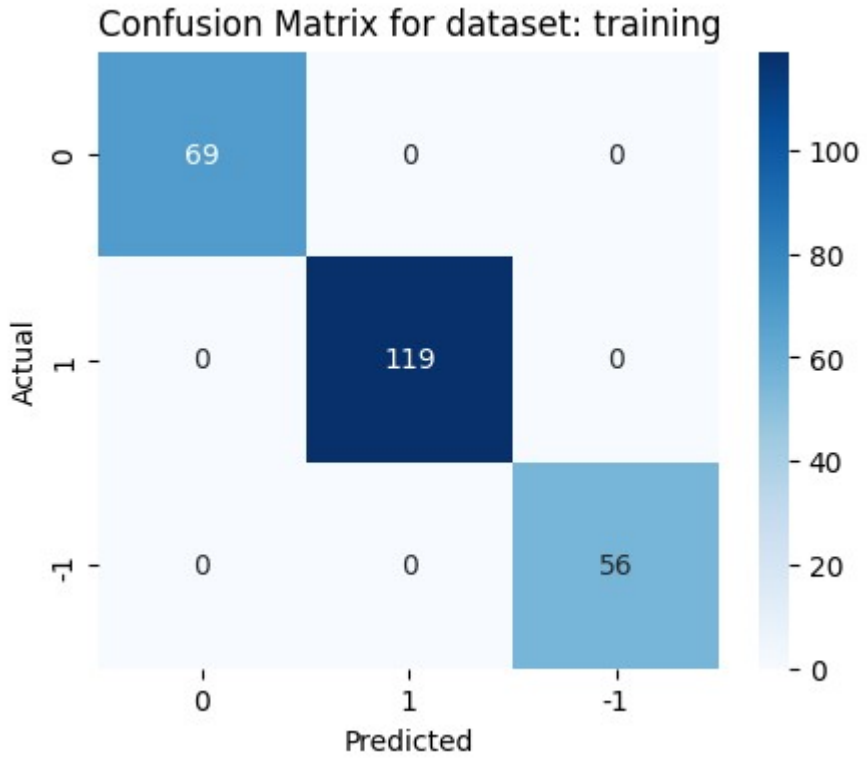
This model is over

7.3. Base Model - GloVe

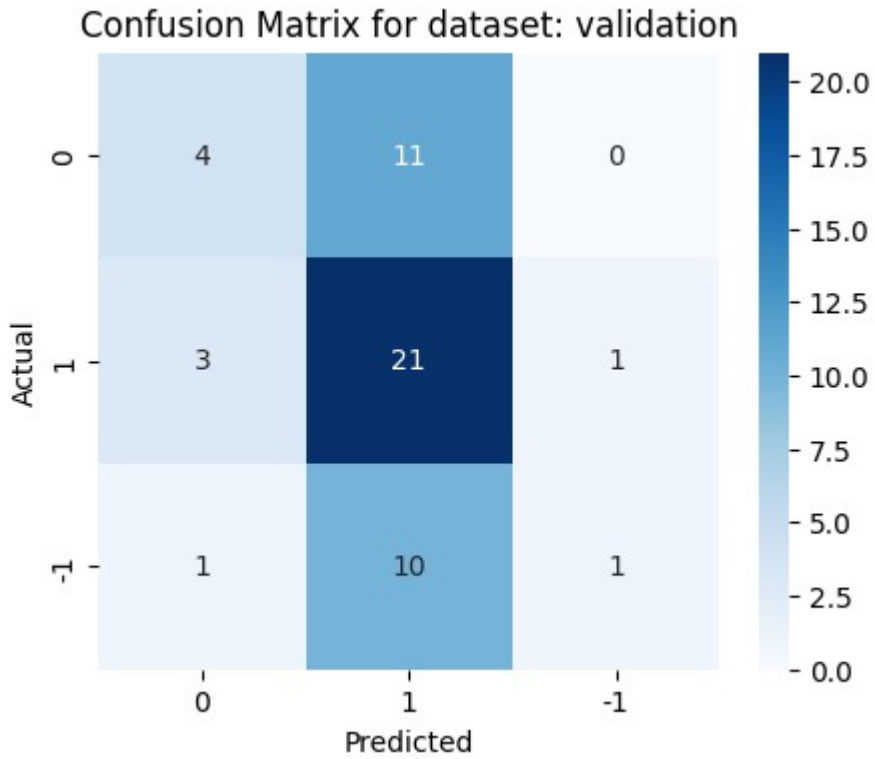
In [ ]:
```
#Building the model

#base_wv = GradientBoostingClassifier(random_state = 42)
base_wv = RandomForestClassifier(random_state=gbl_random_seed)
#base_wv = DecisionTreeClassifier(random_state=42)

# Fitting on train data
base_wv.fit(X_train_gl, y_train) # fit the chosen model on the train data
```

Out[ ]:
```
RandomForestClassifier(random_state=42)
```

In [ ]:
```
plot_confusion_matrix(base_wv, X_train_gl, y_train, "training") # plot the confusion matrix for the train data
```



In [ ]:
```
plot_confusion_matrix(base_wv, X_val_gl, y_val, "validation") #plot the confusion matrix for the validation data
```



In [ ]:
```
#Calculating different metrics on training data
base_train_gl=model_performance_classification_sklearn(base_wv, X_train_gl, y_train) # compute the model performance for the training data
print("Training performance:\n", base_train_gl)
```

```
Training performance:
    Accuracy  Recall  Precision   F1
0       1.0     1.0        1.0  1.0
```

In [ ]:
```
#Calculating different metrics on validation data
base_val_gl = model_performance_classification_sklearn(base_wv, X_val_gl, y_val) # compute the model performance for the validation data
print("Validation performance:\n",base_val_gl)
```

```
Validation performance:
    Accuracy  Recall  Precision        F1
0       0.5     0.5        0.5  0.434679
```

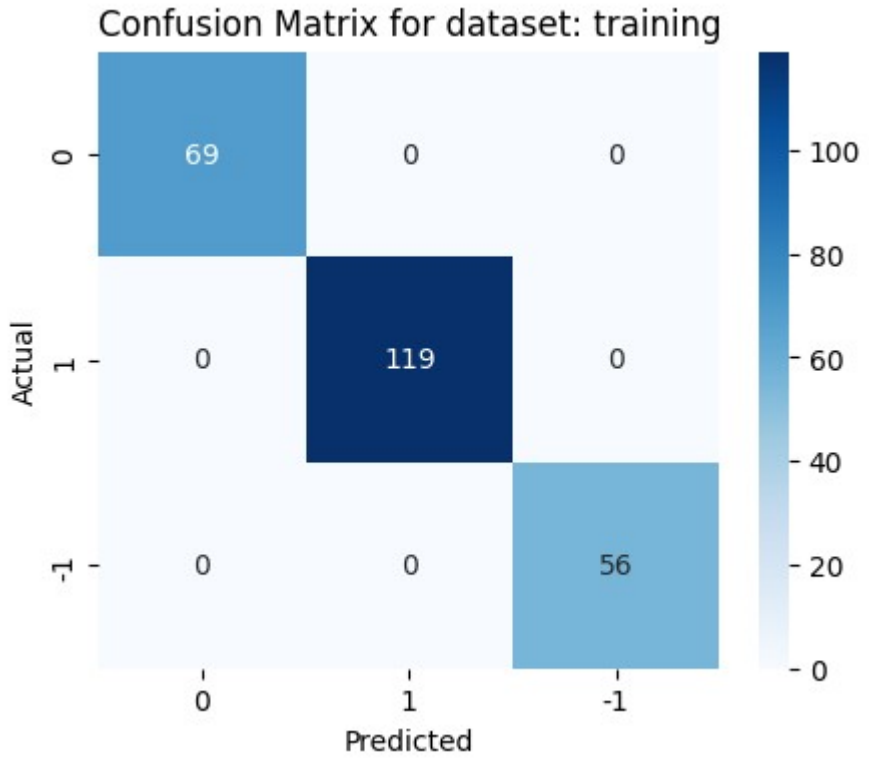
This model's performance is not good either, it drops significantly on the validation data for all metrics (50% or less). The validation confusion matrix clearly shows many misclassifications across all classes. This indicates that the model is overfitting the training data.

7.4. Base Model - Sentence Transformer

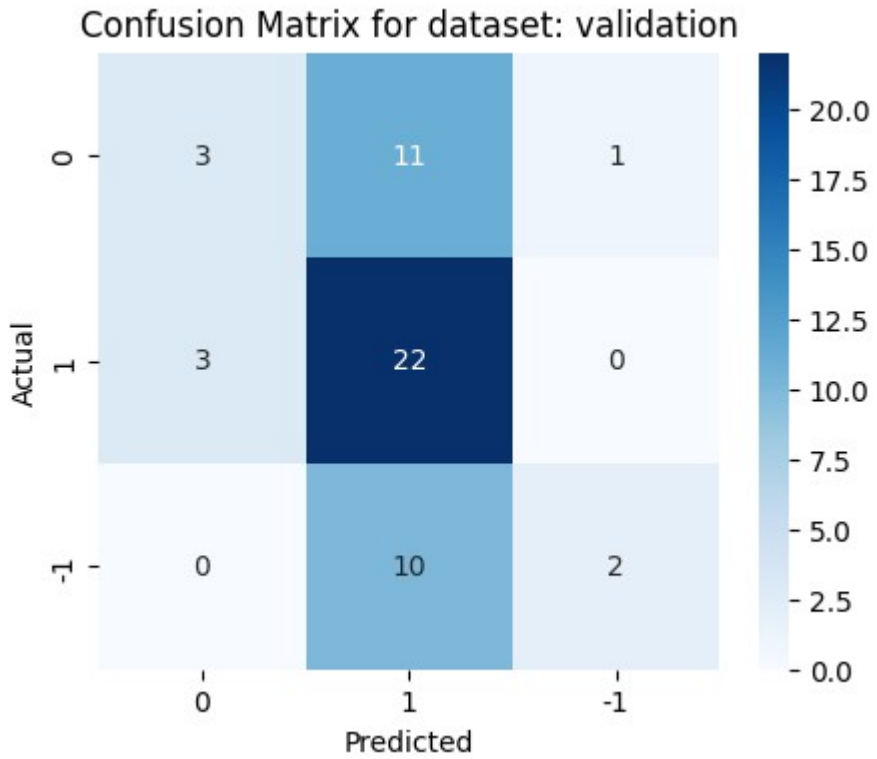
In [ ]:
```
# Building the model

#base_wv = GradientBoostingClassifier(random_state = 42)
base_wv = RandomForestClassifier(random_state= gbl_random_seed)
#base_wv = DecisionTreeClassifier(random_state=42)

# Fitting on train data
base_wv.fit(X_train_st, y_train) # fit the chosen model on the train data
```

Out[ ]:
```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

In [ ]:
```
plot_confusion_matrix(base_wv, X_train_st, y_train, "training") # plot the
confusion matrix for the train data
```



In [ ]:
```
plot_confusion_matrix(base_wv, X_val_st, y_val, "validation") # plot the co
nfusion matrix for the validation data
```



In [ ]:
```
#Calculating different metrics on training data
base_train_st=model_performance_classification_sklearn(base_wv, X_train_st,
y_train) # compute the model performance for the training data
print("Training performance:\n", base_train_st)
```

```
Training performance:
    Accuracy  Recall  Precision   F1
0       1.0     1.0        1.0  1.0
```

In [ ]:
```
#Calculating different metrics on validation data
base_val_st = model_performance_classification_sklearn(base_wv, X_val_st, y
_val) # compute the model performance for the validation data
print("Validation performance:\n",base_val_st)
```

```
Validation performance:
    Accuracy    Recall  Precision        F1
0  0.519231  0.519231   0.544052  0.455042
```

This model did a little better than previous ones. However, performance still drops significantly on the validation data (45-51%). The validation confusion matrix clearly shows many misclassifications across all classes.

7.5. Tuned Model - Word2Vec

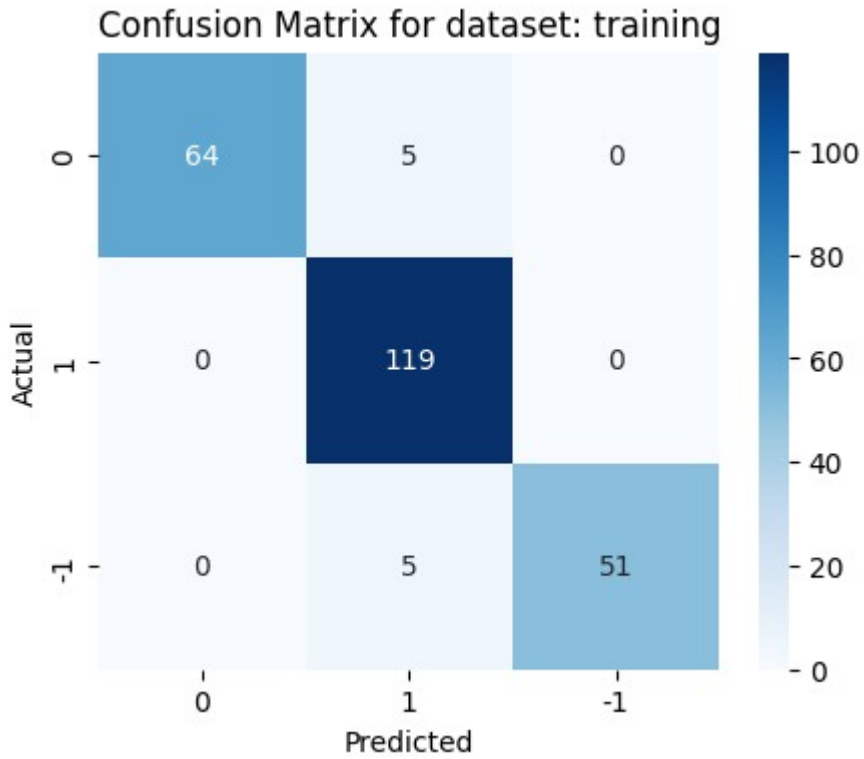
In [ ]:
```
start = time.time()

#tuned_wv = GradientBoostingClassifier(random_state = 42)
tuned_wv = RandomForestClassifier(random_state= gbl_random_seed)
#tuned_wv = DecisionTreeClassifier(random_state=42)


parameters = {
    'max_depth': np.arange(3,7),
    'min_samples_split': np.arange(5,12,2),
    'max_features': ['log2', 'sqrt', 0.2, 0.4]
}

# Run the grid search
grid_obj = GridSearchCV(tuned_wv, parameters, scoring='f1_weighted',cv=5,n_
jobs=-1)
grid_obj = grid_obj.fit(X_train_wv, y_train)

end = time.time()
print("Time taken ",(end-start))

# Set the clf to the best combination of parameters
tuned_wv = grid_obj.best_estimator_
```

```
Time taken  123.45890092849731
```

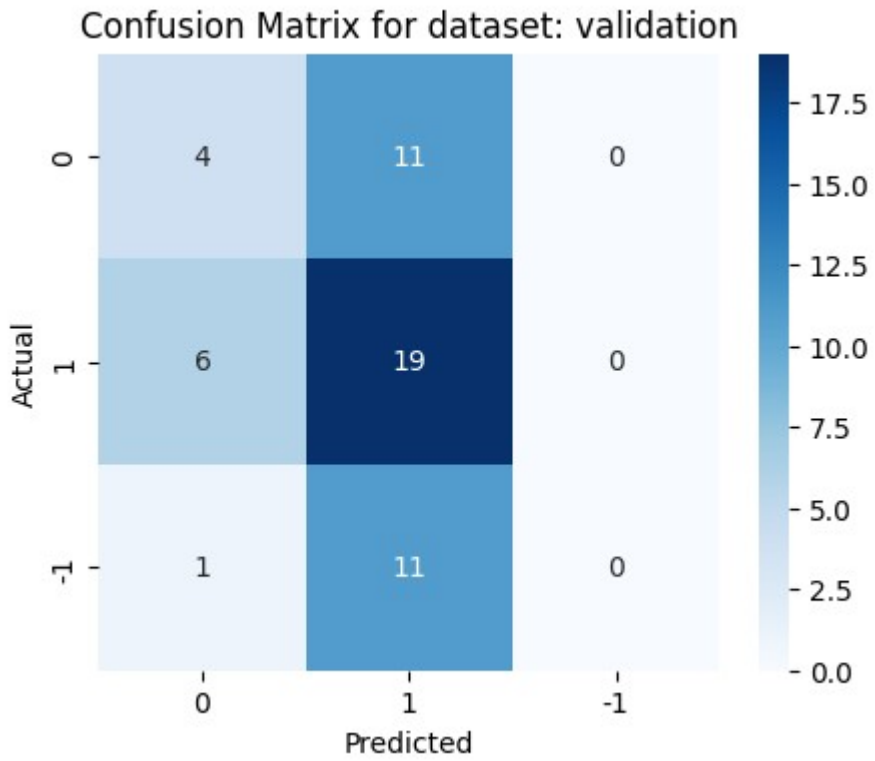
In [ ]:
```
# Fit the best algorithm to the data.
tuned_wv.fit(X_train_wv, y_train)
```

Out[ ]:
```
RandomForestClassifier(max_depth=5, max_features=0.4, min_samples_split=5,
                       random_state=42)
```

In [ ]:
```
plot_confusion_matrix(tuned_wv, X_train_wv, y_train, "training")
```



In [ ]:
```
plot_confusion_matrix(tuned_wv,X_val_wv,y_val, "validation")
```



In [ ]:
```
#Calculating different metrics on training data
tuned_train_wv=model_performance_classification_sklearn(tuned_wv,X_train_w
v,y_train)
print("Training performance:\n",tuned_train_wv)
```

```
Training performance:
    Accuracy    Recall  Precision        F1
0  0.959016  0.959016   0.962193  0.958979
```

In [ ]:
```
#Calculating different metrics on validation data
tuned_val_wv = model_performance_classification_sklearn(tuned_wv,X_val_wv,y
_val)
print("Validation performance:\n",tuned_val_wv)
```

```
Validation performance:
    Accuracy    Recall  Precision        F1
0  0.442308  0.442308   0.327691  0.365564
```

This model is overfitting and, on unseen data, is effectively failing to distinguish between negative, neutral, and positive sentiment, often defaulting to predicting 'positive'. This makes its performance on real-world, unseen data very poor, especially for identifying negative and neutral sentiment.

7.6. Tuned Model - GloVe

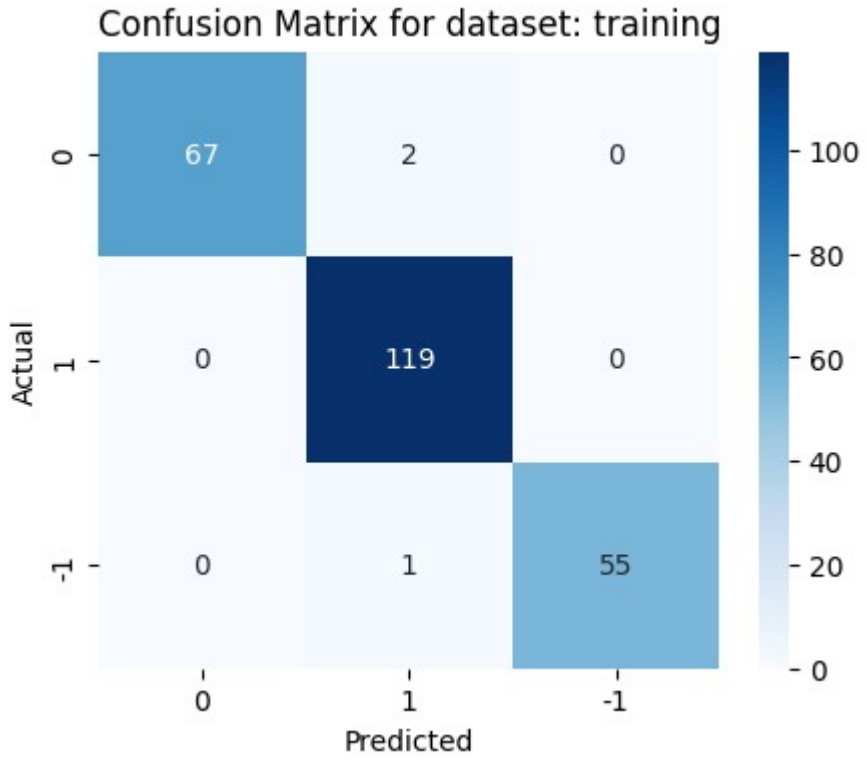
In [ ]:
```
start = time.time()

#tuned_wv = GradientBoostingClassifier(random_state = 42)
tuned_wv = RandomForestClassifier(random_state=gbl_random_seed)
#tuned_wv = DecisionTreeClassifier(random_state=42)

parameters = {
    'max_depth': np.arange(3,7),
    'min_samples_split': np.arange(5,12,2),
    'max_features': ['log2', 'sqrt', 0.2, 0.4]
}

# Run the grid search
grid_obj = GridSearchCV(tuned_wv, parameters, scoring='f1_weighted',cv=5,n_
jobs=-1)
grid_obj = grid_obj.fit(X_train_gl, y_train)

end = time.time()
print("Time taken ",(end-start))

# Set the clf to the best combination of parameters
tuned_gl = grid_obj.best_estimator_
```

```
Time taken  71.72246098518372
```

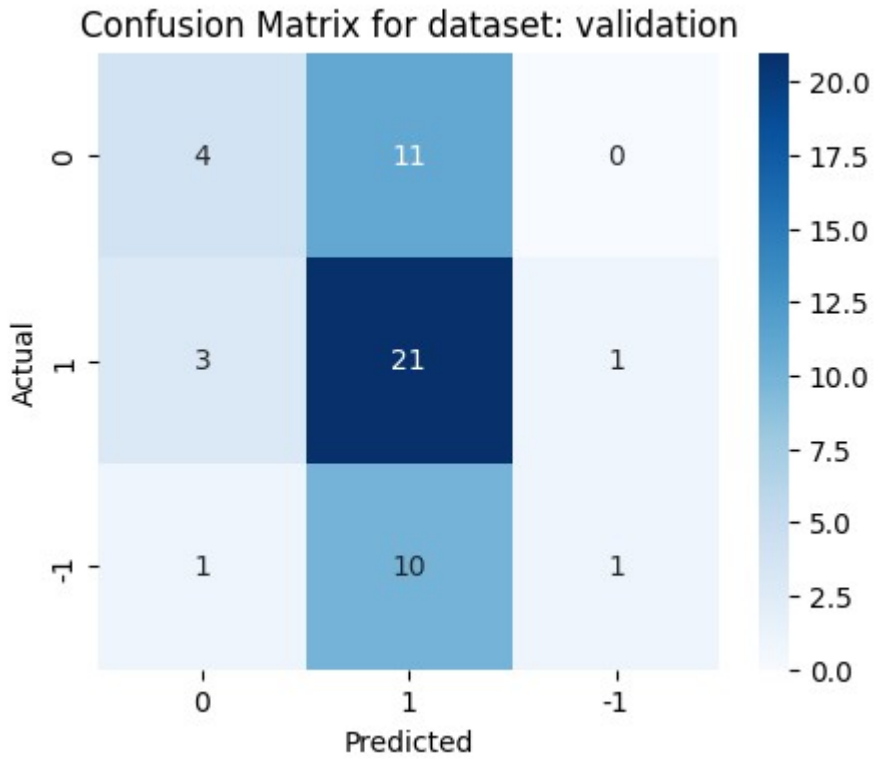
In [ ]:
```
# Fit the best algorithm to the data.
tuned_gl.fit(X_train_gl, y_train)
```

Out[ ]:
```
RandomForestClassifier(max_depth=6, min_samples_split=5, random_state=42)
```

In [ ]:
```
plot_confusion_matrix(tuned_gl, X_train_gl, y_train, "training") # plot the
confusion matrix for the train data
```



In [ ]:
```
plot_confusion_matrix(tuned_gl, X_val_gl, y_val, "validation") # plot the c
onfusion matrix for the validation data
```



In [ ]:
```
#Calculating different metrics on training data
tuned_train_gl = model_performance_classification_sklearn(tuned_gl, X_train
_gl, y_train)
print("Training performance:\n",tuned_train_gl)
```

```
Training performance:
    Accuracy    Recall  Precision        F1
0  0.987705  0.987705   0.988007  0.987703
```

In [ ]:
```
#Calculating different metrics on validation data
tuned_val_gl = model_performance_classification_sklearn(tuned_gl, X_val_gl,
y_val)
print("Validation performance:\n",tuned_val_gl)
```

```
Validation performance:
    Accuracy  Recall  Precision        F1
0       0.5     0.5        0.5  0.434679
```

With metrics between 43-50%, the performance of this model is slightly better than the previous one, however, it's still overfitting the training data. Model often defaults to predicting 'positive'. This makes its performance on real-world, unseen data very poor, especially for identifying negative and neutral sentiment.

7.7. Tuned Model - Sentence Transformer

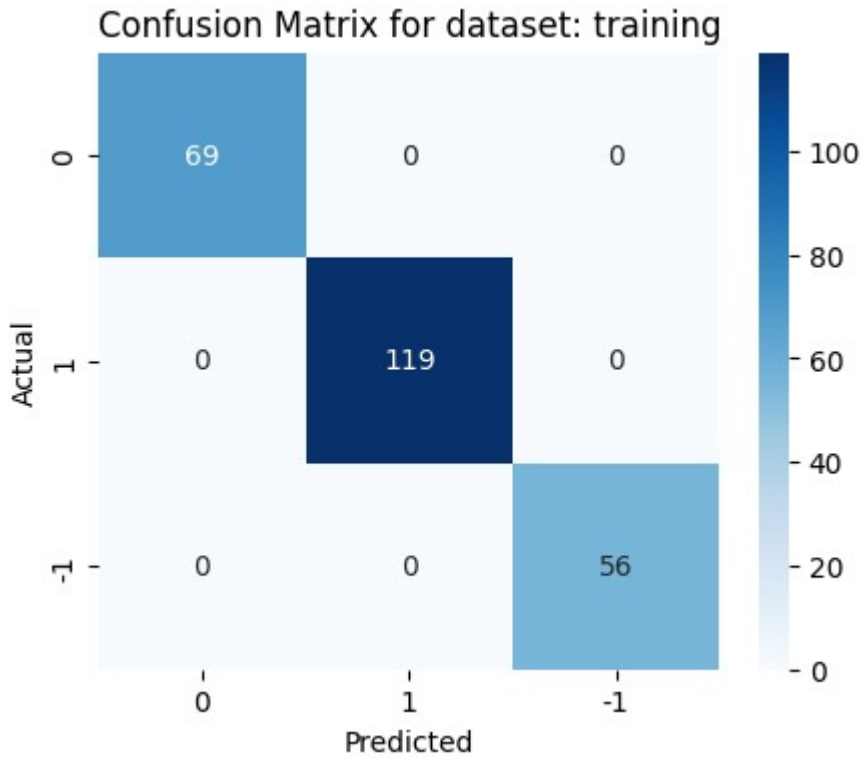
In [ ]:
```
start = time.time()

#tuned_wv = GradientBoostingClassifier(random_state = 42)
tuned_wv = RandomForestClassifier(random_state = gbl_random_seed)
#tuned_wv = DecisionTreeClassifier(random_state=42)

parameters = {
    'max_depth': np.arange(3,7),
    'min_samples_split': np.arange(5,12,2),
    'max_features': ['log2', 'sqrt', 0.2, 0.4]
}

# Run the grid search
grid_obj = GridSearchCV(tuned_wv, parameters, scoring='f1_weighted',cv=5,n_
jobs=-1)
grid_obj = grid_obj.fit(X_train_st, y_train)

end = time.time()
print("Time taken ",(end-start))

# Set the clf to the best combination of parameters
tuned_st = grid_obj.best_estimator_
```

```
Time taken  146.39013934135437
```

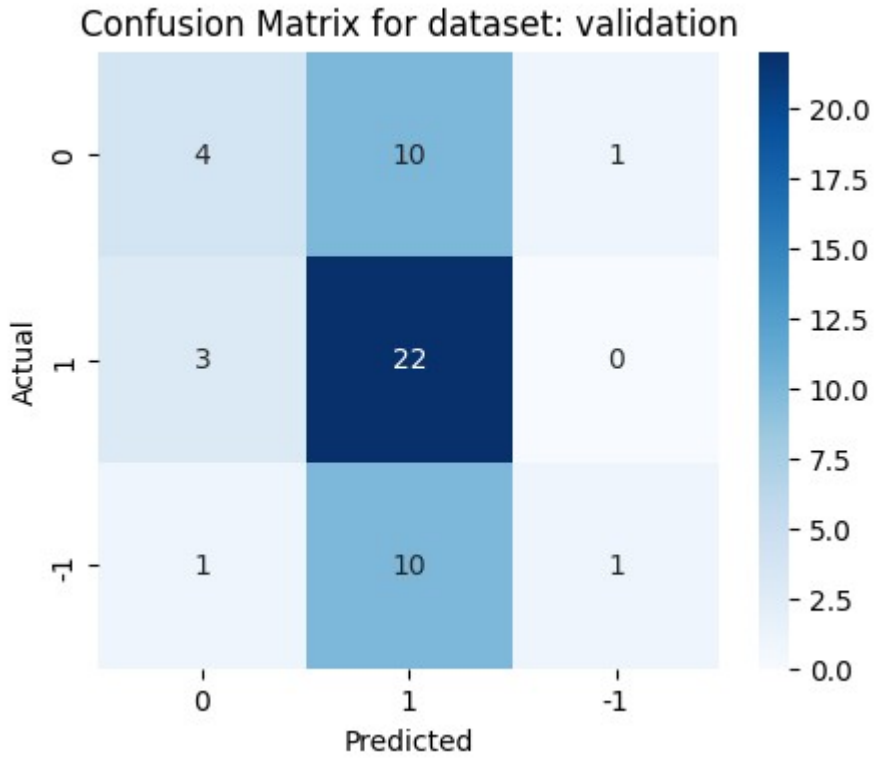
In [ ]:
```
# Fit the best algorithm to the data.
tuned_st.fit(X_train_st, y_train)
```

Out[ ]:
```
RandomForestClassifier(max_depth=6, max_features=0.4, min_samples_split=7,
                       random_state=42)
```

In [ ]:
```
plot_confusion_matrix(tuned_st, X_train_st, y_train, "training") #plot the confusion matrix for the train data
```



In [ ]:
```
plot_confusion_matrix(tuned_st, X_val_st, y_val, "validation") # plot the c
onfusion matrix for the validation data
```



In [ ]:
```
#Calculating different metrics on training data
tuned_train_st=model_performance_classification_sklearn(tuned_st, X_train_s
t, y_train)
print("Training performance:\n",tuned_train_st)
```

```
Training performance:
    Accuracy  Recall  Precision   F1
0       1.0     1.0        1.0  1.0
```

In [ ]:
```
#Calculating different metrics on validation data
tuned_val_st = model_performance_classification_sklearn(tuned_st, X_val_st,
y_val)
print("Validation performance:\n",tuned_val_st)
```

```
Validation performance:
    Accuracy    Recall  Precision        F1
0  0.519231  0.519231   0.511447  0.449031
```

Similar to previous models, this model is overfitting the training data to a 100%, while the validation metrics are between 44-51%, which again shows poor generalization to unseen data.

7.8. Model Performance Summary and Final Model Selection

In [ ]:
```
#training performance comparison

models_train_comp_df = pd.concat(
    [base_train_wv.T,
     base_train_gl.T,
     base_train_st.T,
     tuned_train_wv.T,
     tuned_train_gl.T,
     tuned_train_st.T,
    ],axis=1
)

models_train_comp_df.columns = [
    "Base Model (Word2Vec)",
    "Base Model (GloVe)",
    "Base Model (Sentence Transformer)",
    "Tuned Model (Word2Vec)",
    "Tuned Model (GloVe)",
    "Tuned Model (Sentence Transformer)",
]

print("Training performance comparison:")
models_train_comp_df.T
```

```
Training performance comparison:
```

Out[ ]:

	Accuracy	Recall	Precision	F1
Base Model (Word2Vec)	1.000000	1.000000	1.000000	1.000000
Base Model (GloVe)	1.000000	1.000000	1.000000	1.000000
Base Model (Sentence Transformer)	1.000000	1.000000	1.000000	1.000000
Tuned Model (Word2Vec)	0.959016	0.959016	0.962193	0.958979
Tuned Model (GloVe)	0.987705	0.987705	0.988007	0.987703
Tuned Model (Sentence Transformer)	1.000000	1.000000	1.000000	1.000000

In [ ]:
```
#validation performance comparison

models_val_comp_df = pd.concat(
    [base_val_wv.T,
     base_val_gl.T,
     base_val_st.T,
     tuned_val_wv.T,
     tuned_val_gl.T,
     tuned_val_st.T,
     ],axis=1
)

models_val_comp_df.columns = [
    "Base Model (Word2Vec)",
    "Base Model (GloVe)",
    "Base Model (Sentence Transformer)",
    "Tuned Model (Word2Vec)",
    "Tuned Model (GloVe)",
    "Tuned Model (Sentence Transformer)",
]

print("Validation performance comparison:")
models_val_comp_df.T
```

```
Validation performance comparison:
```

Out[ ]:

	Accuracy	Recall	Precision	F1
Base Model (Word2Vec)	0.480769	0.480769	0.507926	0.447532
Base Model (GloVe)	0.500000	0.500000	0.500000	0.434679
Base Model (Sentence Transformer)	0.519231	0.519231	0.544052	0.455042
Tuned Model (Word2Vec)	0.442308	0.442308	0.327691	0.365564
Tuned Model (GloVe)	0.500000	0.500000	0.500000	0.434679
Tuned Model (Sentence Transformer)	0.519231	0.519231	0.511447	0.449031

7.8.1. Model Performance Check on Test Data

Model selected as final model is 'Tuned Model (GloVe)' = tuned_gl

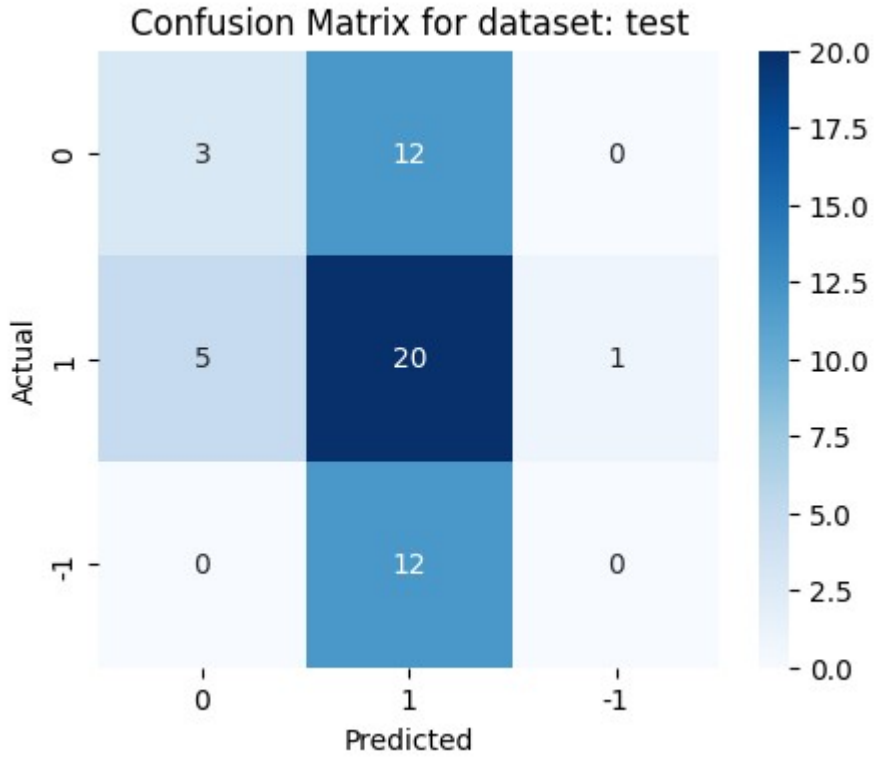
We evaluated six models based on training and validation performance, prioritizing generalization to unseen data. Models achieving perfect training scores but poor validation results were excluded as severe overfitters.

Among the remaining models, 'Tuned Model (GloVe)' was selected as it achieved the highest F1-score (43%) on the validation dataset. While 43% validation F1 is a modest score, it represented the best generalization performance observed across all six evaluated models.

It's also interesing to note that this model (as well as others), have the exact metric values for 'accuracy' and 'recall', and a close value for 'precision'; which may be explained because the metrics are 'weighted'.

In [ ]:
```
final_model = tuned_gl
final_model_X_test = X_test_gl
```

In [ ]:
```
plot_confusion_matrix(final_model, final_model_X_test, y_test, "test") #plo
t the confusion matrix for the final model and test data
```



In [ ]:
```
#Calculating different metrics on test data
final_model_per = model_performance_classification_sklearn(final_model, fin
al_model_X_test, y_test)
print("Test performance:\n",final_model_per)
```

```
Test performance:
    Accuracy    Recall  Precision        F1
0  0.433962  0.433962   0.329117  0.354154
```

Accuracy of **43.4%** and an F1-score of **35.4%**. This indicates the model's current ability to classify sentiment reliably in real-world scenarios is limited.

Analysis of the confusion matrix reveals a significant challenge: the model exhibits a strong bias towards predicting **'Positive' sentiment**. It frequently misclassifies neutral news and critically, completely failed to identify **any actual negative news** in the test set, misclassifying all of it as positive.

This low performance, particularly the inability to detect negative sentiment, means the model is not yet reliable for informing investment decisions, and it's very clear that more effort is needed to improve its accuracy and generalization capability.

8. WEEKLY NEWS SUMMARIZATION

Important Note: It is recommended to run this section of the project independently from the previous sections in order to avoid runtime crashes due to RAM overload.

8.1. Installing and Importing the necessary libraries

...The code from these sections were moved to section 2

8.2. Loading the data

In [ ]:
```
# new dataset copy
data2 = stock_news_orig.copy()
```

8.3. Loading the model

In [ ]:
```
# loading the model from Hugging Face

model_name_or_path = "TheBloke/Mistral-7B-Instruct-v0.2-GGUF"
model_basename = "mistral-7b-instruct-v0.2.Q6_K.gguf"


model_path = hf_hub_download(
    repo_id = model_name_or_path,
    filename = model_basename,
)
```

In [ ]:
```
#uncomment the below snippet of code if the runtime is connected to GPU.
llm = Llama(
    model_path=model_path, # Path to the model
    n_gpu_layers=100, #Number of layers transferred to GPU
    n_ctx=4500, #Context window
)
```

```
llama_model_loader: loaded meta data with 24 key-value pairs and 291 tensor
s from /root/.cache/huggingface/hub/models--TheBloke--Mistral-7B-Instruct-v
0.2-GGUF/snapshots/3a6fbf4a41a1d52e415a4958cde6856d34b2db93/mistral-7b-inst
ruct-v0.2.Q6_K.gguf (version GGUF V3 (latest))
llama_model_loader: Dumping metadata keys/values. Note: KV overrides do not
apply in this output.
llama_model_loader: - kv   0:                       general.architecture st
r              = llama
llama_model_loader: - kv   1:                               general.name st
r              = mistralai_mistral-7b-instruct-v0.2
llama_model_loader: - kv   2:                       llama.context_length u3
2              = 32768
llama_model_loader: - kv   3:                     llama.embedding_length u3
2              = 4096
llama_model_loader: - kv   4:                          llama.block_count u3
2              = 32
llama_model_loader: - kv   5:                  llama.feed_forward_length u3
2              = 14336
llama_model_loader: - kv   6:                 llama.rope.dimension_count u3
2              = 128
llama_model_loader: - kv   7:                 llama.attention.head_count u3
2              = 32
llama_model_loader: - kv   8:              llama.attention.head_count_kv u3
2              = 8
llama_model_loader: - kv   9:     llama.attention.layer_norm_rms_epsilon f3
2              = 0.000010
llama_model_loader: - kv  10:                       llama.rope.freq_base f3
2              = 1000000.000000
llama_model_loader: - kv  11:                          general.file_type u3
2              = 18
llama_model_loader: - kv  12:                       tokenizer.ggml.model st
r              = llama
llama_model_loader: - kv  13:                      tokenizer.ggml.tokens ar
r[str,32000]   = ["<unk>", "<s>", "</s>", "<0x00>", "<...
llama_model_loader: - kv  14:                      tokenizer.ggml.scores ar
r[f32,32000]   = [0.000000, 0.000000, 0.000000, 0.0000...
llama_model_loader: - kv  15:                  tokenizer.ggml.token_type ar
r[i32,32000]   = [2, 3, 3, 6, 6, 6, 6, 6, 6, 6, 6, 6, ...
llama_model_loader: - kv  16:                tokenizer.ggml.bos_token_id u3
2              = 1
llama_model_loader: - kv  17:                tokenizer.ggml.eos_token_id u3
2              = 2
llama_model_loader: - kv  18:            tokenizer.ggml.unknown_token_id u3
2              = 0
llama_model_loader: - kv  19:            tokenizer.ggml.padding_token_id u3
2              = 0
llama_model_loader: - kv  20:               tokenizer.ggml.add_bos_token bo
ol             = true
llama_model_loader: - kv  21:               tokenizer.ggml.add_eos_token bo
ol             = false
llama_model_loader: - kv  22:                    tokenizer.chat_template st
r              = {{ bos_token }}{% for message in mess...
llama_model_loader: - kv  23:               general.quantization_version u3
2              = 2
llama_model_loader: - type  f32:   65 tensors
llama_model_loader: - type q6_K:  226 tensors
```

```
llm_load_vocab: special tokens definition check successful ( 259/32000 ).
llm_load_print_meta: format           = GGUF V3 (latest)
llm_load_print_meta: arch             = llama
llm_load_print_meta: vocab type       = SPM
llm_load_print_meta: n_vocab          = 32000
llm_load_print_meta: n_merges         = 0
llm_load_print_meta: n_ctx_train      = 32768
llm_load_print_meta: n_embd           = 4096
llm_load_print_meta: n_head           = 32
llm_load_print_meta: n_head_kv        = 8
llm_load_print_meta: n_layer          = 32
llm_load_print_meta: n_rot            = 128
llm_load_print_meta: n_embd_head_k    = 128
llm_load_print_meta: n_embd_head_v    = 128
llm_load_print_meta: n_gqa            = 4
llm_load_print_meta: n_embd_k_gqa     = 1024
llm_load_print_meta: n_embd_v_gqa     = 1024
llm_load_print_meta: f_norm_eps       = 0.0e+00
llm_load_print_meta: f_norm_rms_eps   = 1.0e-05
llm_load_print_meta: f_clamp_kqv      = 0.0e+00
llm_load_print_meta: f_max_alibi_bias = 0.0e+00
llm_load_print_meta: n_ff             = 14336
llm_load_print_meta: n_expert         = 0
llm_load_print_meta: n_expert_used    = 0
llm_load_print_meta: rope scaling     = linear
llm_load_print_meta: freq_base_train  = 1000000.0
llm_load_print_meta: freq_scale_train = 1
llm_load_print_meta: n_yarn_orig_ctx  = 32768
llm_load_print_meta: rope_finetuned   = unknown
llm_load_print_meta: model type       = 7B
llm_load_print_meta: model ftype      = Q6_K
llm_load_print_meta: model params     = 7.24 B
llm_load_print_meta: model size       = 5.53 GiB (6.56 BPW)
llm_load_print_meta: general.name     = mistralai_mistral-7b-instruct-v0.2
llm_load_print_meta: BOS token        = 1 '<s>'
llm_load_print_meta: EOS token        = 2 '</s>'
llm_load_print_meta: UNK token        = 0 '<unk>'
llm_load_print_meta: PAD token        = 0 '<unk>'
llm_load_print_meta: LF token         = 13 '<0x0A>'
llm_load_tensors: ggml ctx size =    0.22 MiB
llm_load_tensors: offloading 32 repeating layers to GPU
llm_load_tensors: offloading non-repeating layers to GPU
llm_load_tensors: offloaded 33/33 layers to GPU
llm_load_tensors:        CPU buffer size =   102.54 MiB
llm_load_tensors:      CUDA0 buffer size =  5563.55 MiB
.....................................................................
.....................
llama_new_context_with_model: n_ctx      = 4500
llama_new_context_with_model: freq_base  = 1000000.0
llama_new_context_with_model: freq_scale = 1
llama_kv_cache_init:      CUDA0 KV buffer size =   562.50 MiB
llama_new_context_with_model: KV self size  =  562.50 MiB, K (f16):  281.25
MiB, V (f16):  281.25 MiB
llama_new_context_with_model:  CUDA_Host input buffer size   =    16.81 MiB
llama_new_context_with_model:      CUDA0 compute buffer size =   345.45 MiB
llama_new_context_with_model:  CUDA_Host compute buffer size =     8.80 MiB
```

```
llama_new_context_with_model: graph splits (measure): 3
AVX = 1 | AVX_VNNI = 0 | AVX2 = 1 | AVX512 = 1 | AVX512_VBMI = 0 | AVX512_V
NNI = 0 | FMA = 1 | NEON = 0 | ARM_FMA = 0 | F16C = 1 | FP16_VA = 0 | WASM_
SIMD = 0 | BLAS = 1 | SSE3 = 1 | SSSE3 = 1 | VSX = 0 | MATMUL_INT8 = 0 |
Model metadata: {'tokenizer.chat_template': "{{ bos_token }}{% for message
in messages %}{% if (message['role'] == 'user') != (loop.index0 % 2 == 0)
%}{{ raise_exception('Conversation roles must alternate user/assistant/use
r/assistant/...') }}{% endif %}{% if message['role'] == 'user' %}{{ '[INST]
' + message['content'] + ' [/INST]' }}{% elif message['role'] == 'assistant
' %}{{ message['content'] + eos_token}}{% else %}{{ raise_exception('Only u
ser and assistant roles are supported!') }}{% endif %}{% endfor %}", 'token
izer.ggml.add_eos_token': 'false', 'tokenizer.ggml.padding_token_id': '0',
'tokenizer.ggml.unknown_token_id': '0', 'tokenizer.ggml.eos_token_id': '2',
'general.architecture': 'llama', 'llama.rope.freq_base': '1000000.000000',
'llama.context_length': '32768', 'general.name': 'mistralai_mistral-7b-inst
ruct-v0.2', 'tokenizer.ggml.add_bos_token': 'true', 'llama.embedding_length
': '4096', 'llama.feed_forward_length': '14336', 'llama.attention.layer_nor
m_rms_epsilon': '0.000010', 'llama.rope.dimension_count': '128', 'tokenizer
.ggml.bos_token_id': '1', 'llama.attention.head_count': '32', 'llama.block_
count': '32', 'llama.attention.head_count_kv': '8', 'general.quantization_v
ersion': '2', 'tokenizer.ggml.model': 'llama', 'general.file_type': '18'}
Using chat template: {{ bos_token }}{% for message in messages %}{% if (mes
sage['role'] == 'user') != (loop.index0 % 2 == 0) %}{{ raise_exception('Con
versation roles must alternate user/assistant/user/assistant/...') }}{% end
if %}{% if message['role'] == 'user' %}{{ '[INST] ' + message['content'] +
' [/INST]' }}{% elif message['role'] == 'assistant' %}{{ message['content']
+ eos_token}}{% else %}{{ raise_exception('Only user and assistant roles ar
e supported!') }}{% endif %}{% endfor %}
Using chat eos_token:
Using chat bos_token:
```

8.4. Aggregating the data weekly

In [ ]:
```
data2["Date"] = pd.to_datetime(data2['Date'])  # Convert the 'Date' column
to datetime format.
```

In [ ]:
```
# Group the data by week using the 'Date' column.
weekly_grouped = data2.groupby(pd.Grouper(key='Date', freq='W'))
```

In [ ]:
```
# Join the news values with ' || ' separator.

weekly_grouped = weekly_grouped.agg(
    {
        'News': lambda x: ' || '.join(x)
    }
).reset_index()

print(weekly_grouped.shape)
```

```
(18, 2)
```

In [ ]:
```
weekly_grouped
```

Out[ ]:

	Date	News
0	2019-01-06	The tech sector experienced a significant dec...
1	2019-01-13	Sprint and Samsung plan to release 5G smartph...
2	2019-01-20	The U.S. stock market declined on Monday as c...
3	2019-01-27	The Swiss National Bank (SNB) governor, Andre...
4	2019-02-03	Caterpillar Inc reported lower-than-expected ...
5	2019-02-10	The Dow Jones Industrial Average, S&P 500, an...
6	2019-02-17	This week, the European Union's second highes...
7	2019-02-24	This news article discusses progress towards ...
8	2019-03-03	The Dow Jones Industrial Average and other ma...
9	2019-03-10	Spotify, the world's largest paid music strea...
10	2019-03-17	The United States opposes France's digital se...
11	2019-03-24	Facebook's stock price dropped more than 3% o...
12	2019-03-31	This news article reports that the S&P 500 In...
13	2019-04-07	Apple and other consumer brands, including LV...
14	2019-04-14	In March, mobile phone shipments to China dro...
15	2019-04-21	The chairman of Taiwan's Foxconn, Terry Gou, ...
16	2019-04-28	Taiwan's export orders continued to decline f...
17	2019-05-05	Spotify reported better-than-expected Q1 reve...

In [ ]:
```
# creating a copy of the data
data3 = weekly_grouped.copy()
```

8.5. Summarization

Note:

- The model is expected to summarize the news from the week by identifying the top three positive and negative events that are most likely to impact the price of the stock.
- As an output, the model is expected to return a JSON containing two keys, one for Positive Events and one for Negative Events.

For the project, we need to define the prompt to be fed to the LLM to help it understand the task to perform. The following should be the components of the prompt:

1. **Role**: Specifies the role the LLM will be taking up to perform the specified task, along with any specific details regarding the role
    - **Example**: `You are an expert data analyst specializing in news content analysis.`
2. **Task**: Specifies the task to be performed and outlines what needs to be accomplished, clearly defining the objective
    - **Example**: `Analyze the provided news headline and return the main topics contained within it.`
3. **Instructions**: Provides detailed guidelines on how to perform the task, which includes steps, rules, and criteria to ensure the task is executed correctly
    - **Example**:

```
Instructions:
1. Read the news headline carefully.
2. Identify the main subjects or entities mentioned in the headline.
3. Determine the key events or actions described in the headline.
4. Extract relevant keywords that represent the topics.
5. List the topics in a concise manner.
```

1. **Output Format**: Specifies the format in which the final response should be structured, ensuring consistency and clarity in the generated output
    - **Example**: `Return the output in JSON format with keys as the topic number and values as the actual topic.`

Full Prompt Example:

```
You are an expert data analyst specializing in news content analysis.

Task: Analyze the provided news headline and return the main topics contained wit
hin it.

Instructions:
1. Read the news headline carefully.
2. Identify the main subjects or entities mentioned in the headline.
3. Determine the key events or actions described in the headline.
4. Extract relevant keywords that represent the topics.
5. List the topics in a concise manner.

Return the output in JSON format with keys as the topic number and values as the
actual topic.
```

Sample Output:

```
{"1": "Politics", "2": "Economy", "3": "Health" }
```

8.5.1. Utility Functions

In [ ]:
```python
# defining a function to parse the JSON output from the model
def extract_json_data(json_str):
    import json
    try:
        # Find the indices of the opening and closing curly braces
        json_start = json_str.find('{')
        json_end = json_str.rfind('}')

        if json_start != -1 and json_end != -1:
            extracted_category = json_str[json_start:json_end + 1]  # Extra
ct the JSON object
            data_dict = json.loads(extracted_category)
            return data_dict
        else:
            print(f"Warning: JSON object not found in response: {json_st
r}")
            return {}
    except json.JSONDecodeError as e:
        print(f"Error parsing JSON: {e}")
        return {}
```

8.5.2. Defining the response function

In [ ]:
```
#Defining the response function
def response_mistral_1(prompt, news):
    model_output = llm(
      f"""
      [INST]
      {prompt}
      News Articles: {news}
      [/INST]
      """,
      max_tokens=500,
      temperature=0.1,
      top_p=0.5, #Complete the code to set the value for top_p
      top_k=5,
      echo=False,
    )

    final_output = model_output["choices"][0]["text"]

    return final_output
```

8.5.3. Checking the model output on a sample

Note: Use this section to test out the prompt with one instance before using it for the entire weekly data.

In [ ]:
```
news = data3.loc[0, 'News']
```

In [ ]:
```
print(len(news.split(' ')))
news
```

```
2611
```

Out[ ]: ' The tech sector experienced a significant decline in the aftermarket foll
owing Apple\'s Q1 revenue warning. Notable suppliers, including Skyworks, B
roadcom, Lumentum, Qorvo, and TSMC, saw their stocks drop in response to Ap
ple\'s downward revision of its revenue expectations for the quarter, previ
ously announced in January. || Apple lowered its fiscal Q1 revenue guidanc
e to $84 billion from earlier estimates of $89-$93 billion due to weaker th
an expected iPhone sales. The announcement caused a significant drop in App
le\'s stock price and negatively impacted related suppliers, leading to bro
ader market declines for tech indices such as Nasdaq 10 || Apple cut its f
iscal first quarter revenue forecast from $89-$93 billion to $84 billion du
e to weaker demand in China and fewer iPhone upgrades. CEO Tim Cook also me
ntioned constrained sales of Airpods and Macbooks. Apple\'s shares fell
8.5% in post market trading, while Asian suppliers like Hon || This news a
rticle reports that yields on long-dated U.S. Treasury securities hit their
lowest levels in nearly a year on January 2, 2019, due to concerns about th
e health of the global economy following weak economic data from China and
Europe, as well as the partial U.S. government shutdown. Apple || Apple\'s
revenue warning led to a decline in USD JPY pair and a gain in Japanese yen
, as investors sought safety in the highly liquid currency. Apple\'s underp
erformance in Q1, with forecasted revenue of $84 billion compared to analys
t expectations of $91.5 billion, triggered risk aversion mood in markets ||
Apple CEO Tim Cook discussed the company\'s Q1 warning on CNBC, attributing
US-China trade tensions as a factor. Despite not mentioning iPhone unit sal
es specifically, Cook indicated Apple may comment on them again. Services r
evenue is projected to exceed $10.8 billion in Q1. Cook also addressed the
lack of || Roku Inc has announced plans to offer premium video channels on
a subscription basis through its free streaming service, The Roku Channel.
Partners include CBS Corp\'s Showtime, Lionsgate\'s Starz, and Viacom Inc
\'s Noggin. This model follows Amazon\'s successful Channels business, whic
h generated an estimated || Wall Street saw modest gains on Wednesday but
were threatened by fears of a global economic slowdown following Apple\'s s
hocking revenue forecast cut, blaming weak demand in China. The tech giant
\'s suppliers and S&P 500 futures also suffered losses. Reports of decelera
ting factory activity in China and the euro zone || Apple\'s fiscal first
quarter revenue came in below analysts\' estimates at around $84 billion, a
significant drop from the forecasted range of $89-$93 billion. The tech gia
nt attributed the shortfall to lower iPhone revenue and upgrades, as well a
s weakness in emerging markets. Several brokerages had already reduced thei
r production estimates || Apple Inc. lowered its quarterly sales forecast
for the fiscal first quarter, underperforming analysts\' expectations due t
o slowing Chinese economy and trade tensions. The news sent Apple shares tu
mbling and affected Asia-listed suppliers like Hon Hai Precision Industry C
o Ltd, Taiwan Semiconductor Manufacturing Company, and LG Innot || The Aus
tralian dollar experienced significant volatility on Thursday, plunging to
multi-year lows against major currencies due to automated selling, liquidit
y issues, and a drought of trades. The largest intra-day falls in the Aussi
e\'s history occurred amid violent movements in AUD/JPY and AUD/ || In ear
ly Asian trading on Thursday, the Japanese yen surged as the U.S. dollar an
d Australian dollar collapsed in thin markets due to massive stop loss sale
s triggered by Apple\'s earnings warning of sluggish iPhone sales in China
and risk aversion. The yen reached its lowest levels against the U.S. dolla
r since March || The dollar fell from above 109 to 106.67 after Apple\'s
revenue warning, while the 10-year Treasury yield also dropped to 2.61%. Th
is followed money flowing into US government paper. Apple\'s shares and U.S
. stock index futures declined, with the NAS || RBC Capital maintains its
bullish stance on Apple, keeping its Outperform rating and $220 price targe

t. However, analyst Amit Daryanani warns of ongoing iPhone demand concerns, which could impact pricing power and segmentation efforts if severe. He sug gests potential capital allocation adjustments if the stock underperforms f or several quarters || Oil prices dropped on Thursday as investor sentimen t remained affected by China\'s economic slowdown and turmoil in stock and currency markets. US WTI Crude Oil fell by $2.10 to $45.56 a barrel, while International Brent Oil was down $1.20 at $54.26 || In this news article, investors\' concerns about a slowing Chinese and global economy, amplified by Apple\'s revenue warning, led to a significant surge in the Japanese yen . The yen reached its biggest one-day rise in 20 months, with gains of over 4% versus the dollar. This trend was driven by automated || In Asia, gold prices rose to over six-month highs on concerns of a global economic slowdo wn and stock market volatility. Apple lowered its revenue forecast for the first quarter, leading Asian stocks to decline and safe haven assets like g old and Japanese yen to gain. Data showed weakened factory activity in Asia , particularly China, adding to || Fears of a global economic slowdown led to a decline in the US dollar on Thursday, as the yen gained ground due to its status as a safe haven currency. The USD index slipped below 96, and US D JPY dropped to 107.61, while the yen strengthened by 4.4%. || In Thursda y trading, long-term US Treasury yields dropped significantly below 2.6%, r eaching levels not seen in over a year, as investors shifted funds from sto cks to bonds following Apple\'s warning of decreased revenue due to emergin g markets and China\'s impact on corporate profits, with the White House ad visor adding to concerns of earnings down || Gold prices have reached thei r highest level since mid-June, with the yellow metal hitting $1,291.40 per ounce due to investor concerns over a slowing economy and Apple\'s bearish revenue outlook. Saxo Bank analyst Ole Hansen predicts gold may reach $1,30 0 sooner || Wedbush analyst Daniel Ives lowered his price target for Apple from $275 to $200 due to concerns over potential iPhone sales stagnation, w ith an estimated 750 million active iPhones worldwide that could cease grow ing or even decline. He maintains an Outperform rating and remains bullish on the long || Oil prices rebounded on Thursday due to dollar weakness, si gns of output cuts by Saudi Arabia, and weaker fuel oil margins leading Riy adh to lower February prices for heavier crude grades sold to Asia. The Org anization of the Petroleum Exporting Countries (OPEC) led by Saudi Arabia a nd other producers || This news article reports on the impact of Apple\'s Q1 revenue warning on several tech and biotech stocks. Sesen Bio (SESN) and Prana Biotechnology (PRAN) saw their stock prices drop by 28% and 11%, resp ectively, following the announcement. Mellanox Technologies (ML || Gold pr ices reached within $5 of $1,300 on Thursday as weak stock markets and a sl umping dollar drove investors towards safe-haven assets. The U.S. stock mar ket fell about 2%, with Apple\'s rare profit warning adding to investor une ase. COMEX gold futures settled at $1 || The FDIC Chair, Jelena McWilliams , expressed no concern over market volatility affecting the U.S banking sys tem due to banks\' ample capital. She also mentioned a review of the CAMELS rating system used to evaluate bank health for potential inconsistencies an d concerns regarding forum shopping. This review comes from industry || Ap ple cut its quarterly revenue forecast for the first time in over 15 years due to weak iPhone sales in China, representing around 20% of Apple\'s reve nue. This marks a significant downturn during Tim Cook\'s tenure and reflec ts broader economic concerns in China exacerbated by trade tensions with th e US. U || Goldman analyst Rod Hall lowered his price target for Apple fro m $182 to $140, citing potential risks to the tech giant\'s 2019 numbers du e to uncertainties in Chinese demand. He reduced his revenue estimate for t he year by $6 billion and EPS forecast by $1.54 || Delta Air Lines lowered its fourth-quarter revenue growth forecast to a range of 3% from the previo

us estimate of 3% to 5%. Earnings per share are now expected to be $1.25 to $1.30. The slower pace of improvement in late December was unexpected, and Delta cited this as || Apple\'s profit warning has significantly impacted the stock market and changed the outlook for interest rates. The chance of a rate cut in May has increased to 15-16% from just 3%, according to Investing com\'s Fed Rate Monitor Tool. There is even a 1% chance of two cuts in May. || The White House advisor, Kevin Hassett, stated that a decline in Chinese economic growth would negatively impact U.S. firm profits but recover once a trade deal is reached between Washington and Beijing. He also noted that Asian economies, including China, have been experiencing significant slowdowns since last spring due to U.S. tariffs || The White House economic adviser, Kevin Hassett, warned that more companies could face earnings downgrades due to ongoing trade negotiations between the U.S. and China, leading to a decline in oil prices on Thursday. WTI crude fell 44 cents to $44.97 a barrel, while Brent crude inched || Japanese stocks suffered significant losses on the first trading day of 2019, with the Nikkei 225 and Topix indices both falling over 3 percent. Apple\'s revenue forecast cut, citing weak iPhone sales in China, triggered global growth concerns and sent technology shares tumbling. The S&P 50 || Investors withdrew a record $98 billion from U.S. stock funds in December, with fears of aggressive monetary policy and an economic slowdown driving risk reduction. The S&P 500 fell 9% last month, with some seeing declines as a buying opportunity. Apple\'s warning of weak iPhone sales added || Apple\'s Q1 revenue guidance cut, resulting from weaker demand in China, led to an estimated $3.8 billion paper loss for Berkshire Hathaway due to its $252 million stake in Apple. This news, coupled with broad market declines, caused a significant $21.4 billion decrease in Berk || This news article reports that a cybersecurity researcher, Wish Wu, planned to present at the Black Hat Asia hacking conference on how to bypass Apple\'s Face ID biometric security on iPhones. However, his employer, Ant Financial, which operates Alipay and uses facial recognition technologies including Face ID, asked him to withdraw || OPEC\'s production cuts faced uncertainty as oil prices were influenced by volatile stock markets, specifically due to Apple\'s lowered revenue forecast and global economic slowdown fears. US WTI and Brent crude both saw gains, but these were checked by stock market declines. Shale production is expected to continue impacting the oil market in || Warren Buffett\'s Berkshire Hathaway suffered significant losses in the fourth quarter due to declines in Apple, its largest common stock investment. Apple cut its revenue forecast, causing a 5-6% decrease in Berkshire\'s Class A shares. The decline resulted in potential unrealized investment losses and could push Berk || This news article reports that on Thursday, the two-year Treasury note yield dropped below the Federal Reserve\'s effective rate for the first time since 2008. The market move suggests investors believe the Fed will not be able to continue tightening monetary policy. The drop in yields was attributed to a significant decline in U.S || The U.S. and China will hold their first face-to-face trade talks since agreeing to a 90-day truce in their trade war last month. Deputy U.S. Trade Representative Jeffrey Gerrish will lead the U.S. delegation for negotiations on Jan. 7 and 8, || Investors bought gold in large quantities due to concerns over a global economic slowdown, increased uncertainty in the stock market, and potential Fed rate hikes. The precious metal reached its highest price since June, with gold ETF holdings also seeing significant increases. Factors contributing to this demand include economic downturn, central bank policy mistakes, and || Delta Air Lines Inc reported lower-than-expected fourth quarter unit revenue growth, citing weaker than anticipated late bookings and increased competition. The carrier now expects total revenue per available seat mile to rise about 3 percent in the peri

```
od, down from its earlier forecast of 3.5 percent growth. Fuel prices are a
lso expected to ||  U.S. stocks experienced significant declines on Thursda
y as the S&P 500 dropped over 2%, the Dow Jones Industrial Average fell nea
rly 3%, and the Nasdaq Composite lost approximately 3% following a warning
of weak revenue from Apple and indications of slowing U.S. factory activity
, raising concerns ||  President Trump expressed optimism over potential tr
ade talks with China, citing China\'s current economic weakness as a potent
ial advantage for the US. This sentiment was echoed by recent reports of we
akened demand for Apple iPhones in China, raising concerns about the overal
l health of the Chinese economy. The White House is expected to take a stro
ng stance in ||  Qualcomm secured a court order in Germany banning the sale
of some iPhone models due to patent infringement, leading Apple to potentia
lly remove these devices from its stores. However, third-party resellers li
ke Gravis continue selling the affected iPhones. This is the third major ef
fort by Qualcomm to ban Apple\'s iPhones glob ||  Oil prices rose on Friday
in Asia as China confirmed trade talks with the U.S., with WTI gaining 0.7%
to $47.48 and Brent increasing 0.7% to $56.38 a barrel. The gains came afte
r China\'s Commerce Ministry announced that deputy U.S. Trade ||  Gold pric
es surged past the psychologically significant level of $1,300 per ounce in
Asia on Friday due to growing concerns over a potential global economic dow
nturn. The rise in gold was attributed to weak PMI data from China and Appl
e\'s reduced quarterly sales forecast. Investors viewed gold as a safe have
n asset amidst ||  In an internal memo, Huawei\'s Chen Lifang reprimanded t
wo employees for sending a New Year greeting on the company\'s official Twi
tter account using an iPhone instead of a Huawei device. The incident cause
d damage to the brand and was described as a "blunder" in the memo. The mis
take occurred due to ||  This news article reports on the positive impact o
f trade war talks between Beijing and Washington on European stock markets,
specifically sectors sensitive to the trade war such as carmakers, industri
als, mining companies, and banking. Stocks rallied with mining companies le
ading the gains due to copper price recovery. Bayer shares climbed despite
a potential ruling restricting || Amazon has sold over 100 million devices
with its Alexa digital assistant, according to The Verge. The company is ca
utious about releasing hardware sales figures and did not disclose holiday
numbers for the Echo Dot. Over 150 products feature Alexa, and more than 28
,000 smart home || The Supreme Court will review Broadcom\'s appeal in a sh
areholder lawsuit over the 2015 acquisition of Emulex. The case hinges on w
hether intent to defraud is required for such lawsuits, and the decision co
uld extend beyond the Broadcom suit. An Emulex investor filed a class actio
n lawsuit ||  The Chinese central bank announced a fifth reduction in the r
equired reserve ratio (RRR) for banks, freeing up approximately 116.5 billi
on yuan for new lending. This follows mounting concerns about China\'s econ
omic health amid slowing domestic demand and U.S. tariffs on exports. Premi
er Li Keqiang || The stock market rebounded strongly on Friday following po
sitive news about US-China trade talks, a better-than-expected jobs report,
and dovish comments from Federal Reserve Chairman Jerome Powell. The Dow Jo
nes Industrial Average rose over 746 points, with the S&P 500 and Nasdaq Co
m'
```

In [ ]:
```
# defining the prompt for the task "the task is to identify the top three p
ositive and negative events from the week that are likely to impact the sto
ck price"

prompt = f"""
You are a financial expert tasked with analyzing news articles and identify
ing key events that could significantly impact a company's stock price.

**Instructions:**

1. Analyze the provided news text for the week.
2. Identify the top 3 positive events and the top 3 negative events that ar
e most likely to affect the stock price.
3. Focus on events with clear implications for the company's financial perf
ormance, strategic direction, or market position.


**Desired Output Format:**
Return a JSON object with the following structure:
{{ "positive_events": ["[Event 1]", "[Event 2]", "[Event 3]"], "negative_ev
ents": ["[Event 1]", "[Event 2]", "[Event 3]"] }}

"""
```

In [ ]:
```
data3
news
```

Out[ ]: ' The tech sector experienced a significant decline in the aftermarket foll
owing Apple\'s Q1 revenue warning. Notable suppliers, including Skyworks, B
roadcom, Lumentum, Qorvo, and TSMC, saw their stocks drop in response to Ap
ple\'s downward revision of its revenue expectations for the quarter, previ
ously announced in January. || Apple lowered its fiscal Q1 revenue guidanc
e to $84 billion from earlier estimates of $89-$93 billion due to weaker th
an expected iPhone sales. The announcement caused a significant drop in App
le\'s stock price and negatively impacted related suppliers, leading to bro
ader market declines for tech indices such as Nasdaq 10 || Apple cut its f
iscal first quarter revenue forecast from $89-$93 billion to $84 billion du
e to weaker demand in China and fewer iPhone upgrades. CEO Tim Cook also me
ntioned constrained sales of Airpods and Macbooks. Apple\'s shares fell
8.5% in post market trading, while Asian suppliers like Hon || This news a
rticle reports that yields on long-dated U.S. Treasury securities hit their
lowest levels in nearly a year on January 2, 2019, due to concerns about th
e health of the global economy following weak economic data from China and
Europe, as well as the partial U.S. government shutdown. Apple || Apple\'s
revenue warning led to a decline in USD JPY pair and a gain in Japanese yen
, as investors sought safety in the highly liquid currency. Apple\'s underp
erformance in Q1, with forecasted revenue of $84 billion compared to analys
t expectations of $91.5 billion, triggered risk aversion mood in markets ||
Apple CEO Tim Cook discussed the company\'s Q1 warning on CNBC, attributing
US-China trade tensions as a factor. Despite not mentioning iPhone unit sal
es specifically, Cook indicated Apple may comment on them again. Services r
evenue is projected to exceed $10.8 billion in Q1. Cook also addressed the
lack of || Roku Inc has announced plans to offer premium video channels on
a subscription basis through its free streaming service, The Roku Channel.
Partners include CBS Corp\'s Showtime, Lionsgate\'s Starz, and Viacom Inc
\'s Noggin. This model follows Amazon\'s successful Channels business, whic
h generated an estimated || Wall Street saw modest gains on Wednesday but
were threatened by fears of a global economic slowdown following Apple\'s s
hocking revenue forecast cut, blaming weak demand in China. The tech giant
\'s suppliers and S&P 500 futures also suffered losses. Reports of decelera
ting factory activity in China and the euro zone || Apple\'s fiscal first
quarter revenue came in below analysts\' estimates at around $84 billion, a
significant drop from the forecasted range of $89-$93 billion. The tech gia
nt attributed the shortfall to lower iPhone revenue and upgrades, as well a
s weakness in emerging markets. Several brokerages had already reduced thei
r production estimates || Apple Inc. lowered its quarterly sales forecast
for the fiscal first quarter, underperforming analysts\' expectations due t
o slowing Chinese economy and trade tensions. The news sent Apple shares tu
mbling and affected Asia-listed suppliers like Hon Hai Precision Industry C
o Ltd, Taiwan Semiconductor Manufacturing Company, and LG Innot || The Aus
tralian dollar experienced significant volatility on Thursday, plunging to
multi-year lows against major currencies due to automated selling, liquidit
y issues, and a drought of trades. The largest intra-day falls in the Aussi
e\'s history occurred amid violent movements in AUD/JPY and AUD/ || In ear
ly Asian trading on Thursday, the Japanese yen surged as the U.S. dollar an
d Australian dollar collapsed in thin markets due to massive stop loss sale
s triggered by Apple\'s earnings warning of sluggish iPhone sales in China
and risk aversion. The yen reached its lowest levels against the U.S. dolla
r since March || The dollar fell from above 109 to 106.67 after Apple\'s
revenue warning, while the 10-year Treasury yield also dropped to 2.61%. Th
is followed money flowing into US government paper. Apple\'s shares and U.S
. stock index futures declined, with the NAS || RBC Capital maintains its
bullish stance on Apple, keeping its Outperform rating and $220 price targe

t. However, analyst Amit Daryanani warns of ongoing iPhone demand concerns, which could impact pricing power and segmentation efforts if severe. He sug gests potential capital allocation adjustments if the stock underperforms f or several quarters || Oil prices dropped on Thursday as investor sentimen t remained affected by China\'s economic slowdown and turmoil in stock and currency markets. US WTI Crude Oil fell by $2.10 to $45.56 a barrel, while International Brent Oil was down $1.20 at $54.26 || In this news article, investors\' concerns about a slowing Chinese and global economy, amplified by Apple\'s revenue warning, led to a significant surge in the Japanese yen . The yen reached its biggest one-day rise in 20 months, with gains of over 4% versus the dollar. This trend was driven by automated || In Asia, gold prices rose to over six-month highs on concerns of a global economic slowdo wn and stock market volatility. Apple lowered its revenue forecast for the first quarter, leading Asian stocks to decline and safe haven assets like g old and Japanese yen to gain. Data showed weakened factory activity in Asia , particularly China, adding to || Fears of a global economic slowdown led to a decline in the US dollar on Thursday, as the yen gained ground due to its status as a safe haven currency. The USD index slipped below 96, and US D JPY dropped to 107.61, while the yen strengthened by 4.4%. || In Thursda y trading, long-term US Treasury yields dropped significantly below 2.6%, r eaching levels not seen in over a year, as investors shifted funds from sto cks to bonds following Apple\'s warning of decreased revenue due to emergin g markets and China\'s impact on corporate profits, with the White House ad visor adding to concerns of earnings down || Gold prices have reached thei r highest level since mid-June, with the yellow metal hitting $1,291.40 per ounce due to investor concerns over a slowing economy and Apple\'s bearish revenue outlook. Saxo Bank analyst Ole Hansen predicts gold may reach $1,30 0 sooner || Wedbush analyst Daniel Ives lowered his price target for Apple from $275 to $200 due to concerns over potential iPhone sales stagnation, w ith an estimated 750 million active iPhones worldwide that could cease grow ing or even decline. He maintains an Outperform rating and remains bullish on the long || Oil prices rebounded on Thursday due to dollar weakness, si gns of output cuts by Saudi Arabia, and weaker fuel oil margins leading Riy adh to lower February prices for heavier crude grades sold to Asia. The Org anization of the Petroleum Exporting Countries (OPEC) led by Saudi Arabia a nd other producers || This news article reports on the impact of Apple\'s Q1 revenue warning on several tech and biotech stocks. Sesen Bio (SESN) and Prana Biotechnology (PRAN) saw their stock prices drop by 28% and 11%, resp ectively, following the announcement. Mellanox Technologies (ML || Gold pr ices reached within $5 of $1,300 on Thursday as weak stock markets and a sl umping dollar drove investors towards safe-haven assets. The U.S. stock mar ket fell about 2%, with Apple\'s rare profit warning adding to investor une ase. COMEX gold futures settled at $1 || The FDIC Chair, Jelena McWilliams , expressed no concern over market volatility affecting the U.S banking sys tem due to banks\' ample capital. She also mentioned a review of the CAMELS rating system used to evaluate bank health for potential inconsistencies an d concerns regarding forum shopping. This review comes from industry || Ap ple cut its quarterly revenue forecast for the first time in over 15 years due to weak iPhone sales in China, representing around 20% of Apple\'s reve nue. This marks a significant downturn during Tim Cook\'s tenure and reflec ts broader economic concerns in China exacerbated by trade tensions with th e US. U || Goldman analyst Rod Hall lowered his price target for Apple fro m $182 to $140, citing potential risks to the tech giant\'s 2019 numbers du e to uncertainties in Chinese demand. He reduced his revenue estimate for t he year by $6 billion and EPS forecast by $1.54 || Delta Air Lines lowered its fourth-quarter revenue growth forecast to a range of 3% from the previo

us estimate of 3% to 5%. Earnings per share are now expected to be $1.25 to $1.30. The slower pace of improvement in late December was unexpected, and Delta cited this as || Apple\'s profit warning has significantly impacted the stock market and changed the outlook for interest rates. The chance of a rate cut in May has increased to 15-16% from just 3%, according to Investing com\'s Fed Rate Monitor Tool. There is even a 1% chance of two cuts in May. || The White House advisor, Kevin Hassett, stated that a decline in Chinese economic growth would negatively impact U.S. firm profits but recover once a trade deal is reached between Washington and Beijing. He also noted that Asian economies, including China, have been experiencing significant slowdowns since last spring due to U.S. tariffs || The White House economic adviser, Kevin Hassett, warned that more companies could face earnings downgrades due to ongoing trade negotiations between the U.S. and China, leading to a decline in oil prices on Thursday. WTI crude fell 44 cents to $44.97 a barrel, while Brent crude inched || Japanese stocks suffered significant losses on the first trading day of 2019, with the Nikkei 225 and Topix indices both falling over 3 percent. Apple\'s revenue forecast cut, citing weak iPhone sales in China, triggered global growth concerns and sent technology shares tumbling. The S&P 50 || Investors withdrew a record $98 billion from U.S. stock funds in December, with fears of aggressive monetary policy and an economic slowdown driving risk reduction. The S&P 500 fell 9% last month, with some seeing declines as a buying opportunity. Apple\'s warning of weak iPhone sales added || Apple\'s Q1 revenue guidance cut, resulting from weaker demand in China, led to an estimated $3.8 billion paper loss for Berkshire Hathaway due to its $252 million stake in Apple. This news, coupled with broad market declines, caused a significant $21.4 billion decrease in Berk || This news article reports that a cybersecurity researcher, Wish Wu, planned to present at the Black Hat Asia hacking conference on how to bypass Apple\'s Face ID biometric security on iPhones. However, his employer, Ant Financial, which operates Alipay and uses facial recognition technologies including Face ID, asked him to withdraw || OPEC\'s production cuts faced uncertainty as oil prices were influenced by volatile stock markets, specifically due to Apple\'s lowered revenue forecast and global economic slowdown fears. US WTI and Brent crude both saw gains, but these were checked by stock market declines. Shale production is expected to continue impacting the oil market in || Warren Buffett\'s Berkshire Hathaway suffered significant losses in the fourth quarter due to declines in Apple, its largest common stock investment. Apple cut its revenue forecast, causing a 5-6% decrease in Berkshire\'s Class A shares. The decline resulted in potential unrealized investment losses and could push Berk || This news article reports that on Thursday, the two-year Treasury note yield dropped below the Federal Reserve\'s effective rate for the first time since 2008. The market move suggests investors believe the Fed will not be able to continue tightening monetary policy. The drop in yields was attributed to a significant decline in U.S || The U.S. and China will hold their first face-to-face trade talks since agreeing to a 90-day truce in their trade war last month. Deputy U.S. Trade Representative Jeffrey Gerrish will lead the U.S. delegation for negotiations on Jan. 7 and 8, || Investors bought gold in large quantities due to concerns over a global economic slowdown, increased uncertainty in the stock market, and potential Fed rate hikes. The precious metal reached its highest price since June, with gold ETF holdings also seeing significant increases. Factors contributing to this demand include economic downturn, central bank policy mistakes, and || Delta Air Lines Inc reported lower-than-expected fourth quarter unit revenue growth, citing weaker than anticipated late bookings and increased competition. The carrier now expects total revenue per available seat mile to rise about 3 percent in the peri

od, down from its earlier forecast of 3.5 percent growth. Fuel prices are a lso expected to || U.S. stocks experienced significant declines on Thursda y as the S&P 500 dropped over 2%, the Dow Jones Industrial Average fell nea rly 3%, and the Nasdaq Composite lost approximately 3% following a warning of weak revenue from Apple and indications of slowing U.S. factory activity , raising concerns || President Trump expressed optimism over potential tr ade talks with China, citing China\'s current economic weakness as a potent ial advantage for the US. This sentiment was echoed by recent reports of we akened demand for Apple iPhones in China, raising concerns about the overal l health of the Chinese economy. The White House is expected to take a stro ng stance in || Qualcomm secured a court order in Germany banning the sale of some iPhone models due to patent infringement, leading Apple to potentia lly remove these devices from its stores. However, third-party resellers li ke Gravis continue selling the affected iPhones. This is the third major ef fort by Qualcomm to ban Apple\'s iPhones glob || Oil prices rose on Friday in Asia as China confirmed trade talks with the U.S., with WTI gaining 0.7% to $47.48 and Brent increasing 0.7% to $56.38 a barrel. The gains came afte r China\'s Commerce Ministry announced that deputy U.S. Trade || Gold pric es surged past the psychologically significant level of $1,300 per ounce in Asia on Friday due to growing concerns over a potential global economic dow nturn. The rise in gold was attributed to weak PMI data from China and Appl e\'s reduced quarterly sales forecast. Investors viewed gold as a safe have n asset amidst || In an internal memo, Huawei\'s Chen Lifang reprimanded t wo employees for sending a New Year greeting on the company\'s official Twi tter account using an iPhone instead of a Huawei device. The incident cause d damage to the brand and was described as a "blunder" in the memo. The mis take occurred due to || This news article reports on the positive impact o f trade war talks between Beijing and Washington on European stock markets, specifically sectors sensitive to the trade war such as carmakers, industri als, mining companies, and banking. Stocks rallied with mining companies le ading the gains due to copper price recovery. Bayer shares climbed despite a potential ruling restricting || Amazon has sold over 100 million devices with its Alexa digital assistant, according to The Verge. The company is ca utious about releasing hardware sales figures and did not disclose holiday numbers for the Echo Dot. Over 150 products feature Alexa, and more than 28 ,000 smart home || The Supreme Court will review Broadcom\'s appeal in a sh areholder lawsuit over the 2015 acquisition of Emulex. The case hinges on w hether intent to defraud is required for such lawsuits, and the decision co uld extend beyond the Broadcom suit. An Emulex investor filed a class actio n lawsuit || The Chinese central bank announced a fifth reduction in the r equired reserve ratio (RRR) for banks, freeing up approximately 116.5 billi on yuan for new lending. This follows mounting concerns about China\'s econ omic health amid slowing domestic demand and U.S. tariffs on exports. Premi er Li Keqiang || The stock market rebounded strongly on Friday following po sitive news about US-China trade talks, a better-than-expected jobs report, and dovish comments from Federal Reserve Chairman Jerome Powell. The Dow Jo nes Industrial Average rose over 746 points, with the S&P 500 and Nasdaq Co m'

In [ ]:
```
# getting response from Mistral

%%time
summary = response_mistral_1(prompt, news)
print(summary)
```

```
llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     241.26 ms /   242 runs   (    1
.00 ms per token,  1003.05 tokens per second)
llama_print_timings: prompt eval time =    4954.54 ms /  3931 tokens (    1
.26 ms per token,   793.41 tokens per second)
llama_print_timings:        eval time =   10037.25 ms /   241 runs   (   41
.65 ms per token,    24.01 tokens per second)
llama_print_timings:       total time =   16935.10 ms /  4172 tokens

 {
          "positive_events": [
            "Roku Inc announced plans to offer premium video channels on a
subscription basis through its free streaming service, The Roku Channel.",
            "The U.S. and China will hold their first face-to-face trade ta
lks since agreeing to a 90-day truce in their trade war last month.",
            "The Chinese central bank announced a fifth reduction in the re
quired reserve ratio (RRR) for banks, freeing up approximately 116.5 billio
n yuan for new lending."
          ],
          "negative_events": [
            "Apple cut its quarterly revenue forecast for the first time in
over 15 years due to weak iPhone sales in China.",
            "Oil prices dropped on Thursday as investor sentiment remained
affected by China's economic slowdown and turmoil in stock and currency mar
kets.",
            "Apple's profit warning led to significant declines in U.S. sto
ck indices, including the S&P 500, Dow Jones Industrial Average, and Nasdaq
Composite."
          ]
        }
CPU times: user 13.5 s, sys: 919 ms, total: 14.4 s
Wall time: 17 s
```

8.5.4. Checking the model output on the weekly data

In [ ]:
```
# applying the function to get the Key news events

%%time
data3['Key Events'] = data3['News'].progress_apply(lambda x: response_mistr
al_1(prompt,x))
```

```
  0%|          | 0/18 [00:00<?, ?it/s]Llama.generate: prefix-match hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     146.77 ms /   242 runs   (    0
.61 ms per token,  1648.87 tokens per second)
llama_print_timings: prompt eval time =       0.00 ms /     1 tokens (    0
.00 ms per token,      inf tokens per second)
llama_print_timings:        eval time =    9695.15 ms /   242 runs   (   40
.06 ms per token,    24.96 tokens per second)
llama_print_timings:       total time =   11034.90 ms /   243 tokens
 11%|█         | 2/18 [00:11<01:28,  5.53s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =      87.56 ms /   170 runs   (    0
.52 ms per token,  1941.59 tokens per second)
llama_print_timings: prompt eval time =    2603.28 ms /  2234 tokens (    1
.17 ms per token,   858.15 tokens per second)
llama_print_timings:        eval time =    6237.61 ms /   169 runs   (   36
.91 ms per token,    27.09 tokens per second)
llama_print_timings:       total time =    9498.09 ms /  2403 tokens
 17%|██        | 3/18 [00:20<01:47,  7.19s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     205.14 ms /   341 runs   (    0
.60 ms per token,  1662.30 tokens per second)
llama_print_timings: prompt eval time =    2450.51 ms /  2089 tokens (    1
.17 ms per token,   852.47 tokens per second)
llama_print_timings:        eval time =   12430.39 ms /   340 runs   (   36
.56 ms per token,    27.35 tokens per second)
llama_print_timings:       total time =   16608.34 ms /  2429 tokens
 22%|██       | 4/18 [00:37<02:29, 10.71s/it]Llama.generate: prefix-match h
it

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     167.11 ms /   302 runs   (    0
.55 ms per token,  1807.17 tokens per second)
llama_print_timings: prompt eval time =    1608.44 ms /  1475 tokens (    1
.09 ms per token,   917.04 tokens per second)
llama_print_timings:        eval time =   11016.67 ms /   301 runs   (   36
.60 ms per token,    27.32 tokens per second)
llama_print_timings:       total time =   13857.06 ms /  1776 tokens
 28%|██▊       | 5/18 [00:51<02:33, 11.81s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     150.49 ms /   268 runs   (    0
.56 ms per token,  1780.90 tokens per second)
llama_print_timings: prompt eval time =    3368.22 ms /  2587 tokens (    1
.30 ms per token,   768.06 tokens per second)
llama_print_timings:        eval time =   10634.70 ms /   267 runs   (   39
.83 ms per token,    25.11 tokens per second)
llama_print_timings:       total time =   15177.23 ms /  2854 tokens
 33%|███       | 6/18 [01:06<02:35, 12.94s/it]Llama.generate: prefix-match
hit
```

```
llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     114.81 ms /   208 runs   (    0
.55 ms per token,  1811.74 tokens per second)
llama_print_timings: prompt eval time =    1320.38 ms /  1083 tokens (    1
.22 ms per token,   820.22 tokens per second)
llama_print_timings:        eval time =    8113.74 ms /   207 runs   (   39
.20 ms per token,    25.51 tokens per second)
llama_print_timings:       total time =   10229.22 ms /  1290 tokens
 39%|███▉      | 7/18 [01:16<02:12, 12.07s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     152.65 ms /   278 runs   (    0
.55 ms per token,  1821.18 tokens per second)
llama_print_timings: prompt eval time =    1496.11 ms /  1271 tokens (    1
.18 ms per token,   849.54 tokens per second)
llama_print_timings:        eval time =   11230.23 ms /   277 runs   (   40
.54 ms per token,    24.67 tokens per second)
llama_print_timings:       total time =   13814.08 ms /  1548 tokens
 44%|████      | 8/18 [01:30<02:06, 12.63s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =      95.39 ms /   185 runs   (    0
.52 ms per token,  1939.51 tokens per second)
llama_print_timings: prompt eval time =     725.78 ms /   570 tokens (    1
.27 ms per token,   785.36 tokens per second)
llama_print_timings:        eval time =    6914.83 ms /   184 runs   (   37
.58 ms per token,    26.61 tokens per second)
llama_print_timings:       total time =    8277.92 ms /   754 tokens
 50%|████▌     | 9/18 [01:38<01:41, 11.28s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     175.32 ms /   228 runs   (    0
.77 ms per token,  1300.46 tokens per second)
llama_print_timings: prompt eval time =     813.14 ms /   654 tokens (    1
.24 ms per token,   804.29 tokens per second)
llama_print_timings:        eval time =    8625.57 ms /   227 runs   (   38
.00 ms per token,    26.32 tokens per second)
llama_print_timings:       total time =   10561.78 ms /   881 tokens
 56%|█████     | 10/18 [01:49<01:28, 11.06s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     101.35 ms /   175 runs   (    0
.58 ms per token,  1726.62 tokens per second)
llama_print_timings: prompt eval time =     849.48 ms /   726 tokens (    1
.17 ms per token,   854.64 tokens per second)
llama_print_timings:        eval time =    6387.44 ms /   174 runs   (   36
.71 ms per token,    27.24 tokens per second)
llama_print_timings:       total time =    7940.52 ms /   900 tokens
 61%|█████▌    | 11/18 [01:57<01:10, 10.12s/it]Llama.generate: prefix-match
hit
```

```
llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     174.16 ms /   305 runs   (    0
.57 ms per token,  1751.23 tokens per second)
llama_print_timings: prompt eval time =    1326.94 ms /  1138 tokens (    1
.17 ms per token,   857.61 tokens per second)
llama_print_timings:        eval time =   11141.23 ms /   304 runs   (   36
.65 ms per token,    27.29 tokens per second)
llama_print_timings:       total time =   14203.58 ms /  1442 tokens
 67%|██████▋   | 12/18 [02:11<01:08, 11.36s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     151.42 ms /   266 runs   (    0
.57 ms per token,  1756.69 tokens per second)
llama_print_timings: prompt eval time =    1422.73 ms /  1263 tokens (    1
.13 ms per token,   887.73 tokens per second)
llama_print_timings:        eval time =    9689.14 ms /   265 runs   (   36
.56 ms per token,    27.35 tokens per second)
llama_print_timings:       total time =   12183.01 ms /  1528 tokens
 72%|███████▏  | 13/18 [02:23<00:58, 11.62s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     168.91 ms /   294 runs   (    0
.57 ms per token,  1740.57 tokens per second)
llama_print_timings: prompt eval time =    1936.36 ms /  1656 tokens (    1
.17 ms per token,   855.21 tokens per second)
llama_print_timings:        eval time =   10927.66 ms /   293 runs   (   37
.30 ms per token,    26.81 tokens per second)
llama_print_timings:       total time =   14109.55 ms /  1949 tokens
 78%|███████▊  | 14/18 [02:37<00:49, 12.37s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     148.31 ms /   259 runs   (    0
.57 ms per token,  1746.32 tokens per second)
llama_print_timings: prompt eval time =     994.93 ms /   870 tokens (    1
.14 ms per token,   874.44 tokens per second)
llama_print_timings:        eval time =    9451.27 ms /   258 runs   (   36
.63 ms per token,    27.30 tokens per second)
llama_print_timings:       total time =   11482.42 ms /  1128 tokens
 83%|████████▎ | 15/18 [02:49<00:36, 12.11s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     101.90 ms /   195 runs   (    0
.52 ms per token,  1913.72 tokens per second)
llama_print_timings: prompt eval time =     523.28 ms /   507 tokens (    1
.03 ms per token,   968.88 tokens per second)
llama_print_timings:        eval time =    7241.29 ms /   194 runs   (   37
.33 ms per token,    26.79 tokens per second)
llama_print_timings:       total time =    8404.49 ms /   701 tokens
 89%|████████▉ | 16/18 [02:57<00:21, 11.00s/it]Llama.generate: prefix-matc
h hit

llama_print_timings:        load time =     733.29 ms
```

```
llama_print_timings:      sample time =     126.37 ms /   215 runs   (    0
.59 ms per token,  1701.33 tokens per second)
llama_print_timings: prompt eval time =    1661.79 ms /  1446 tokens (    1
.15 ms per token,   870.15 tokens per second)
llama_print_timings:        eval time =    7985.04 ms /   214 runs   (   37
.31 ms per token,    26.80 tokens per second)
llama_print_timings:       total time =   10566.50 ms /  1660 tokens
 94%|█████████▍| 17/18 [03:08<00:10, 10.87s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =     129.33 ms /   224 runs   (    0
.58 ms per token,  1731.98 tokens per second)
llama_print_timings: prompt eval time =    1001.50 ms /   870 tokens (    1
.15 ms per token,   868.69 tokens per second)
llama_print_timings:        eval time =    8171.79 ms /   223 runs   (   36
.64 ms per token,    27.29 tokens per second)
llama_print_timings:       total time =   10084.22 ms /  1093 tokens
100%|██████████| 18/18 [03:18<00:00, 10.64s/it]Llama.generate: prefix-match
hit

llama_print_timings:        load time =     733.29 ms
llama_print_timings:      sample time =      98.03 ms /   184 runs   (    0
.53 ms per token,  1877.00 tokens per second)
llama_print_timings: prompt eval time =     995.80 ms /   870 tokens (    1
.14 ms per token,   873.67 tokens per second)
llama_print_timings:        eval time =    6806.19 ms /   183 runs   (   37
.19 ms per token,    26.89 tokens per second)
llama_print_timings:       total time =    8441.72 ms /  1053 tokens
100%|██████████| 18/18 [03:26<00:00, 11.49s/it]

CPU times: user 3min 19s, sys: 4.89 s, total: 3min 24s
Wall time: 3min 26s
```

In [ ]:
```
# printing the first 5 rows
data3.head()
```

Out[ ]:

	Date	News	Key Events
0	2019-01-06	The tech sector experienced a significant dec...	{\n "positive_events": [\n ...
1	2019-01-13	Sprint and Samsung plan to release 5G smartph...	{\n "positive_events": [\n ...
2	2019-01-20	The U.S. stock market declined on Monday as c...	{\n "positive_events": [\n ...
3	2019-01-27	The Swiss National Bank (SNB) governor, Andre...	{\n "positive_events": [\n ...
4	2019-02-03	Caterpillar Inc reported lower-than-expected ...	{ "positive_events": ["Apple reported spendin...

In [ ]:
```
# printing the first 5 rows of the 'Key_Events' column
data3["Key Events"].head()
```

Out[ ]:

	Key Events
0	{\n "positive_events": [\n ...
1	{\n "positive_events": [\n ...
2	{\n "positive_events": [\n ...
3	{\n "positive_events": [\n ...
4	{ "positive_events": ["Apple reported spendin...

dtype: object

Formatting the model output

In [ ]:
```
# formatting the data
data3['model_response_parsed'] = data3['Key Events'].apply(extract_json_dat
a)
data3.head()
```

Out[ ]:

	Date	News	Key Events	model_response_parsed
0	2019-01-06	The tech sector experienced a significant dec...	{\n "positive_events": [\n ...	{'positive_events': ['Roku Inc announced plans...
1	2019-01-13	Sprint and Samsung plan to release 5G smartph...	{\n "positive_events": [\n ...	{'positive_events': ['Sprint and Samsung plann...
2	2019-01-20	The U.S. stock market declined on Monday as c...	{\n "positive_events": [\n ...	{'positive_events': ['Dialog Semiconductor rep...
3	2019-01-27	The Swiss National Bank (SNB) governor, Andre...	{\n "positive_events": [\n ...	{'positive_events': ['IBM's stock price increa...
4	2019-02-03	Caterpillar Inc reported lower-than-expected ...	{ "positive_events": ["Apple reported spendin...	{'positive_events': ['Apple reported spending ...

In [ ]:
```
# parsing the positive events from the negative events
model_response_parsed = pd.json_normalize(data3['model_response_parsed'])

# showing the first 5 rows
model_response_parsed.head()
```

Out[ ]:

	positive_events	negative_events
0	[Roku Inc announced plans to offer premium vid...	[Apple cut its quarterly revenue forecast for ...
1	[Sprint and Samsung planning to release 5G sma...	[Geely forecasting flat sales for 2019 due to ...
2	[Dialog Semiconductor reported fourth quarter ...	[The U.S. stock market declined on Monday as c...
3	[IBM's stock price increased after hours due t...	[The Swiss National Bank (SNB) governor, Andre...
4	[Apple reported spending over $60 billion with...	[Caterpillar Inc reported lower-than-expected ...

In [ ]:
```
# getting the final dataframe with the week date, news, positive events and negative events

final_output = pd.concat([data3.reset_index(drop=True), model_response_parsed], axis=1)
final_output.drop(['Key Events','model_response_parsed'], axis=1, inplace=True)
final_output.columns = ['Week End Date', 'News', 'Week Positive Events', 'Week Negative Events']

# showing the first 5 rows
final_output.head()
```

Out[ ]:

	Week End Date	News	Week Positive Events	Week Negative Events
0	2019-01-06	The tech sector experienced a significant dec...	[Roku Inc announced plans to offer premium vid...	[Apple cut its quarterly revenue forecast for ...
1	2019-01-13	Sprint and Samsung plan to release 5G smartph...	[Sprint and Samsung planning to release 5G sma...	[Geely forecasting flat sales for 2019 due to ...
2	2019-01-20	The U.S. stock market declined on Monday as c...	[Dialog Semiconductor reported fourth quarter ...	[The U.S. stock market declined on Monday as c...
3	2019-01-27	The Swiss National Bank (SNB) governor, Andre...	[IBM's stock price increased after hours due t...	[The Swiss National Bank (SNB) governor, Andre...
4	2019-02-03	Caterpillar Inc reported lower-than-expected ...	[Apple reported spending over $60 billion with...	[Caterpillar Inc reported lower-than-expected ...

9. CONCLUSIONS AND RECOMMENDATIONS

- ***A. Executive Summary of Findings***

- Market Dynamics: Stock price fluctuations often align with trading volume spikes, particularly during non-neutral news events. While sentiment shows weak correlation with price levels, it's more closely tied to market activity like volume and volatility.
- Data Characteristics: The dataset includes short news snippets (~49 characters) and corresponding stock data. Sentiment labels are imbalanced, skewed toward 'Neutral.'
- Model Prototyping: Six models were tested using various embeddings (GloVe, Sentence Transformer, Word2Vec), focusing on F1-score performance for generalization.
- Current Performance: The best model (tuned GloVe-based) reached a 43% F1-score on validation. However, it often misclassified neutral/negative news as positive and failed to detect negative sentiment accurately.

- ***B. Implications for Investment Strategy***

The current model isn't reliable enough for use in investment decisions. Its tendency to over-predict positive sentiment undermines its ability to flag market-moving negative news effectively.

- ***C. Proposed Next Steps & Recommendations***
- Improve Data & Signal Extraction:
  - Address class imbalance to better detect Positive and Negative news.
  - Adopt advanced contextual embeddings for richer semantic understanding.
- Boost Model Generalization:
  - Test alternative architectures better suited for sequential data.
- Deepen Error Analysis:
  - Analyze frequent misclassifications to uncover blind spots and refine feature engineering.
- Refine the Link Between Sentiment and Market Outcomes:
  - Based on the EDA, sentiment appears more strongly related to trading volume and price movement rather than absolute price levels. We should focus on using the predicted sentiment as a feature in models designed to predict price changes or volatility, aligning our approach with the observed market dynamics.
- Explore Leveraging Large Language Models (LLMs) as an Alternative: Investigate using powerful LLMs directly via prompting (zero-shot or few-shot learning) for sentiment classification. This approach could potentially be simpler and faster to implement compared to training models from scratch or extensively fine-tuning smaller models, as LLMs already possess vast linguistic understanding and some inherent sentiment analysis capabilities from their pre-training.

- ***D. Conclusion***

Our analysis surfaces key challenges in generalization and sentiment detection. The next phase will apply more advanced techniques in data handling, embeddings, and modeling to create a sentiment tool capable of delivering actionable insights and supporting investment strategy. Or alternatively, classifying sentiment with an LLM is also a viable and straight forward solution.

Power Ahead