In [3]:
```
import json
import requests
from bs4 import BeautifulSoup

def scrape_article_content(url):
    headers = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit
    }
    try:
        response = requests.get(url, headers=headers)
        response.raise_for_status()
        soup = BeautifulSoup(response.text, 'html.parser')

        # Modify the following selector to match the article content structu
        content_paragraphs = soup.find_all('p')
        article_text = ' '.join([p.get_text(strip=True) for p in content_par

        return article_text

    except requests.exceptions.RequestException as e:
        print(f"Error fetching article from {url}: {e}")
        return None

with open('/Users/tiff1101/Downloads/alpha_vantage_financial_news_week.json'
    articles_data = json.load(f)

# Print the first item to verify the data structure
print("First article:", articles_data[0])

# Assuming `articles_data` is a list of article metadata with URLs
for article in articles_data:
    article_url = article['url']
    full_content = scrape_article_content(article_url)
    if full_content:
        article['full_content'] = full_content

print(articles_data[0])
```

```
First article: {'title': 'DATA BREACH ALERT: Edelson Lechtzin LLP Is Investi
gating Claims On Behalf Of PracticeSuite, Inc. Customers Whose Data May Have
Been Compromised', 'url': 'https://www.benzinga.com/pressreleases/24/11/g421
00443/data-breach-alert-edelson-lechtzin-llp-is-investigating-claims-on-beha
lf-of-practicesuite-inc-cust', 'time_published': '20241120T220956', 'author
s': ['Globe Newswire'], 'summary': 'NEWTOWN, Pa., Nov. 20, 2024 ( GLOBE NEWS
WIRE ) -- The law firm of Edelson Lechtzin LLP is investigating claims regar
ding data privacy violations by PracticeSuite, Inc. ( "PracticeSuite" ) . Pr
acticeSuite learned of suspicious activity on its computer network on or abo
ut October 11, 2024.', 'banner_image': 'https://www.benzinga.com/next-asset
s/images/schema-image-default.png', 'source': 'Benzinga', 'category_within_s
```

ource': 'News', 'source_domain': 'www.benzinga.com', 'topics': [{'topic': 'F
inance', 'relevance_score': '1.0'}, {'topic': 'Financial Markets', 'relevanc
e_score': '0.108179'}], 'overall_sentiment_score': -0.026751, 'overall_senti
ment_label': 'Neutral', 'ticker_sentiment': [{'ticker': 'STT', 'relevance_sc
ore': '0.082111', 'ticker_sentiment_score': '0.0', 'ticker_sentiment_label':
'Neutral'}]}
{'title': 'DATA BREACH ALERT: Edelson Lechtzin LLP Is Investigating Claims O
n Behalf Of PracticeSuite, Inc. Customers Whose Data May Have Been Compromis
ed', 'url': 'https://www.benzinga.com/pressreleases/24/11/g42100443/data-bre
ach-alert-edelson-lechtzin-llp-is-investigating-claims-on-behalf-of-practice
suite-inc-cust', 'time_published': '20241120T220956', 'authors': ['Globe New
swire'], 'summary': 'NEWTOWN, Pa., Nov. 20, 2024 (GLOBE NEWSWIRE) -- The l
aw firm of Edelson Lechtzin LLP is investigating claims regarding data priva
cy violations by PracticeSuite, Inc. ("PracticeSuite") . PracticeSuite lea
rned of suspicious activity on its computer network on or about October 11,
2024.', 'banner_image': 'https://www.benzinga.com/next-assets/images/schema-
image-default.png', 'source': 'Benzinga', 'category_within_source': 'News',
'source_domain': 'www.benzinga.com', 'topics': [{'topic': 'Finance', 'releva
nce_score': '1.0'}, {'topic': 'Financial Markets', 'relevance_score': '0.108
179'}], 'overall_sentiment_score': -0.026751, 'overall_sentiment_label': 'Ne
utral', 'ticker_sentiment': [{'ticker': 'STT', 'relevance_score': '0.08211
1', 'ticker_sentiment_score': '0.0', 'ticker_sentiment_label': 'Neutral'}],
'full_content': 'NEWTOWN, Pa., Nov. 20, 2024 (GLOBE NEWSWIRE) -- The law f
irm of Edelson Lechtzin LLP is investigating claims regarding data privacy v
iolations by PracticeSuite, Inc. ("PracticeSuite"). PracticeSuite learned of
suspicious activity on its computer network on or about October 11, 2024. To
join this case, goHERE. About PracticeSuite, Inc. PracticeSuite, Inc., locat
ed in Tampa, Florida, is a cloud-based software company that provides compre
hensive medical practice management solutions. These solutions include medic
al billing, electronic health records, and overall practice management servi
ces. What happened? On or about October 11, 2024, PracticeSuite discovered t
hat an unauthorized hacker had accessed a data file on one of its servers, w
hich contained backup records from one of PracticeSuite\'s clients, Texan EN
T Specialists, PLLC, up to March 19, 2024. After learning about this, Practi
ceSuite began investigating and discovered that certain personal information
had been obtained. The compromised information that the data breach has impa
cted includes names, Social Security numbers, account numbers, demographic i
nformation such as addresses, dates of birth, phone numbers, email addresse
s, and other sensitive personal health information. How can I protect my per
sonal data? If you receive a data breach notification, you must guard agains
t possible misuse of your personal information, including identity theft and
fraud, by regularly reviewing your account statements and monitoring your cr
edit reports for suspicious or unauthorized activity. Additionally, you shou
ld consider legal options for mitigating such risks. Edelson Lechtzin LLP is
investigating a class action lawsuit to seek legal remedies for customers wh
o may have had their sensitive personal and patient data compromised by the
PracticeSuite data breach. For more information, please contact: Marc H. Ede
lson, Esq.EDELSON LECHTZIN LLP411 S. State Street, Suite N-300Newtown, PA 18
940Phone: 844-696-7492Email:medelson@edelson-law.comWeb:www.edelson-law.com
About Edelson Lechtzin LLP Edelson Lechtzin LLPis a national class action la

```
w firm with offices in Pennsylvania and California. In addition to cases inv
olving data breaches, our lawyers focus on class and collective litigation i
n cases alleging securities and investment fraud, violations of the federal
antitrust laws, employee benefit plans under ERISA, wage theft and unpaid ov
ertime, consumer fraud, and dangerous and defective drugs and medical device
s. This press release may be considered Attorney Advertising in some jurisdi
ctions. No class has been certified in this case, so counsel does not repres
ent you unless you retain one. You may select counsel of your choice. You ma
y also remain an absent class member and do nothing now. Your ability to sha
re in any potential future recovery does not depend on serving as lead plain
tiff.  © 2024 Benzinga.com. Benzinga does not provide investment advice. All
rights reserved. Trade confidently with insights and alerts from analyst rat
ings, free reports and breaking news that affects the stocks you care abou
t.'}
```

First article: {'title': 'Goldman Sachs To Spin Out Digital Assets Platform Amid Blockchain Push - Goldman Sachs Gr (NYSE:GS) ', 'url': 'https://www.benzinga.com/markets/cryptocurrency/24/11/42047066/goldman-sachs-to-spin-out-digital-assets-platform-amid-blockchain-push', 'time_published': '20241118T195339', 'authors': ['Murtuza Merchant'], 'summary': 'Goldman Sachs Group Inc. NYSE: GS) is advancing its blockchain ambitions, planning to spin out its digital-assets platform into an independent company within the next 12 to 18 months.', 'banner_image': 'https://cdn.benzinga.com/files/images/story/2024/11/18/Goldman-Sachs-Accepts-Resolution.jpeg?width=1200&height=800&fit=crop', 'source': 'Benzinga', 'category_within_source': 'News', 'source_domain': 'www.benzinga.com', 'topics': [{'topic': 'Finance', 'relevance_score': '1.0'}, {'topic': 'Blockchain', 'relevance_score': '0.838487'}, {'topic': 'Financial Markets', 'relevance_score': '0.161647'}], 'overall_sentiment_score': 0.449601, 'overall_sentiment_label': 'Bullish', 'ticker_sentiment': [{'ticker': 'GS', 'relevance_score': '0.181819', 'ticker_sentiment_score': '0.0', 'ticker_sentiment_label': 'Neutral'}, {'ticker': 'CRYPTO:BTC', 'relevance_score': '0.269776', 'ticker_sentiment_score': '0.443827', 'ticker_sentiment_label': 'Bullish'}, {'ticker': 'CRYPTO:ETH', 'relevance_score': '0.091509', 'ticker_sentiment_score': '0.274239', 'ticker_sentiment_label': 'Somewhat-Bullish'}]} {'title': 'Goldman Sachs To Spin Out Digital Assets Platform Amid Blockchain Push - Goldman Sachs Gr (NYSE:GS) ', 'url': 'https://www.benzinga.com/markets/cryptocurrency/24/11/42047066/goldman-sachs-to-spin-out-digital-assets-platform-amid-blockchain-push', 'time_published': '20241118T195339', 'authors': ['Murtuza Merchant'], 'summary': 'Goldman Sachs Group Inc. NYSE: GS) is advancing its blockchain ambitions, planning to spin out its digital-assets platform into an independent company within the next 12 to 18 months.', 'banner_image': 'https://cdn.benzinga.com/files/images/story/2024/11/18/Goldman-Sachs-Accepts-Resolution.jpeg?width=1200&height=800&fit=crop', 'source':

'Benzinga', 'category_within_source': 'News', 'source_domain': 'www.benzinga.com', 'topics': [{'topic': 'Finance', 'relevance_score': '1.0'}, {'topic': 'Blockchain', 'relevance_score': '0.838487'}, {'topic': 'Financial Markets', 'relevance_score': '0.161647'}], 'overall_sentiment_score': 0.449601, 'overall_sentiment_label': 'Bullish', 'ticker_sentiment': [{'ticker': 'GS', 'relevance_score': '0.181819', 'ticker_sentiment_score': '0.0', 'ticker_sentiment_label': 'Neutral'}, {'ticker': 'CRYPTO:BTC', 'relevance_score': '0.269776', 'ticker_sentiment_score': '0.443827', 'ticker_sentiment_label': 'Bullish'}, {'ticker': 'CRYPTO:ETH', 'relevance_score': '0.091509', 'ticker_sentiment_score': '0.274239', 'ticker_sentiment_label': 'Somewhat-Bullish'}],

'full_content': 'Goldman Sachs Group Inc.NYSE: GS) is advancing its blockchain ambitions, planning to spin out its digital-assets platform into an independent company within the next 12 to 18 months. What Happened:The platform aims to enable large financial institutions to create, trade, and settle financial instruments using blockchain technology, Bloombergreported. Mathew McDermott, Goldman's global head of Digital Assets, revealed the plans, emphasizing the importance of making the platform "industry-owned" to benefit the broader market. The move is contingent on regulatory approval. Goldman is actively engaging with potential partners to enhance the platform's capabilities and explore new commercial applications. The first strategic partner,Tradeweb Markets Inc., has already joined forces with Goldman to develop use cases, marking a significant step toward the initiative's realization. Goldman's spin-out plan reflects a broader trend of financial institutions leveraging blockchain to streamline the issuance, trading, and settlement of traditional assets like cash and bonds. Blockchain technology offers a faster, more efficient alternative to conventional systems, a proposition increasingly attractive to institutional players. Also Read:Bitcoin To $100,000$ 'Seems Around The Corner,' Bulls Are 'On The Right Side Of History' — Ber 93,000 has further bolstered confidence in the transformative potential of digital assets. "Despite market volatility, we see significant long-term opportunities for blockchain and Bitcoin in institutional finance," McDermott said. Goldman's optimism is evident in its investments in blockchain-backed financial products. Earlier this year, the firm partnered withDRW Capitalto allocate $600 million across spot Bitcoin and Ethereum ETFs, signaling a strong commitment to the sector's growth. In addition to its blockchain platform, Goldman is exploring opportunities to facilitate secondary transactions in private digital-asset companies, providing liquidity for clients like family offices while allowing buyers to capitalize on private market discounts. The bank is also preparing to resume its Bitcoin-backed lending activities, highlighting its confidence in Bitcoin's role as a financial instrument. Goldman's efforts align with renewed momentum in the crypto market following Donald Trump's election victory. Investors are betting on favorable

digital asset policies under his administration, drivinginstitutional interest and regulatory clarity. These developments will be central to discussions at Benzinga'sFuture of Digital Assetsevent on Nov. 19. Read Next: Photo: Shutterstock © 2024 Benzinga.com. Benzinga does not provide investment advice. All rights reserved. Trade confidently with insights and alerts from analyst ratings, free reports and breaking news that affects the stocks you care about.'}

In [4]:
```
import yfinance as yf
from datetime import datetime, timedelta

# Expanded list of stock indices from global markets
tickers = [
    '^GSPC', '^DJI', 'QQQ', '^IXIC', '^RUT', '^RUA', '^W5000', '^MID', '^VIX
    '^FTSE', '^DAX', '^CAC40', '^N225', '^HSI', '^ASX200', '^STOXX50E', '^AE
    '^BSE', '^NSEI', '^KOSPI', '^TSEC', '^KLSE', '^JSE', '^BVSP', '^MXX', '^
    '^TA35', '^SENSEX', '^SET50', '^SSEC', '^TWSE', '^IDX', '^CSI300','^NZ50
    '^OMXS30', '^GDAXI', '^SPTSX', '^OSEBX', '^ATHEX', '^KSE', '^IBOV', '^TA
    '^MERV', '^ALLORDS', '^JKSE', '^IBOVESPA', '^IBEX35', '^TADAWUL', '^SAX'
]

# Function to calculate the binary trend (1 for increase, 0 for decrease)
def calculate_binary_trend(stock_data):
    """
    Calculates binary trends based on intraday price movement.

    Parameters:
        stock_data (DataFrame): The stock data with 'Open' and 'Close' colum

    Returns:
        List[int]: Binary trends for each day in the data.
    """
    trends = []
    for _, row in stock_data.iterrows():
        trends.append(int(row['Close'] > row['Open']))
    return trends

def get_stock_trends(tickers, start_date, end_date):
    """
    Fetches intraday trends for a list of stock indices over a given date ra

    Parameters:
        tickers (list): List of stock indices.
        start_date (str): Start date in 'YYYY-MM-DD' format.
        end_date (str): End date in 'YYYY-MM-DD' format.

    Returns:
        dict: Dictionary of binary trends for each ticker.
        list: Flattened list of all binary trends.
```

```
    """
    binary_trends = {}

    # Retrieve data for each ticker for the specified date range and calcula
    for ticker in tickers:
        print(f"Fetching data for {ticker}...")
        try:
            stock_data = yf.download(ticker, start=start_date, end=end_date,
            if not stock_data.empty:
                binary_trends[ticker] = calculate_binary_trend(stock_data[['
            else:
                binary_trends[ticker] = [None] * ((datetime.strptime(end_dat
        except Exception as e:
            print(f"Error fetching data for {ticker}: {e}")
            binary_trends[ticker] = [None] * ((datetime.strptime(end_date, '

    # Flatten all binary trends into a single list (filter out None values)
    binary_feature_list = [trend for trends in binary_trends.values() for tr

    return binary_trends, binary_feature_list

# Set start and end dates for data retrieval
start_date = "2024-11-21"
end_date = "2024-11-24"

# Fetch binary trends
binary_trends, binary_feature_list = get_stock_trends(tickers, start_date, e

# Print the binary trends for each ticker
print("Binary Trends for Each Index:")
print(binary_trends)

# Display the binary feature list
print("Binary Feature List for Classification:")
print(binary_feature_list)
```

```
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
```

```
  trends.append(int(row['Close'] > row['Open']))
[*********************100%***********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[*********************100%***********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[*********************100%***********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[*********************100%***********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[*********************100%***********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[*********************100%***********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[*********************100%***********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[*********************100%***********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[*********************100%***********************]  1 of 1 completed

1 Failed download:
['^DAX']: YFPricesMissingError('$%ticker%: possibly delisted; no price data
found  (1d 2024-11-21 -> 2024-11-24)')
[*********************100%***********************]  1 of 1 completed

1 Failed download:
['^CAC40']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
```

```
d')
Fetching data for ^GSPC...
Fetching data for ^DJI...
Fetching data for QQQ...
Fetching data for ^IXIC...
Fetching data for ^RUT...
Fetching data for ^RUA...
Fetching data for ^W5000...
Fetching data for ^MID...
Fetching data for ^VIX...
Fetching data for ^NDX...
Fetching data for ^FTSE...
Fetching data for ^DAX...
Fetching data for ^CAC40...
Fetching data for ^N225...
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^ASX200']: YFTzMissingError('$%ticker%: possibly delisted; no timezone fou
nd')
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^BEL20']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
d')
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
```

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

1 Failed download:
['^BSE']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
d')
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^KOSPI']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
d')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^TSEC']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
d')
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^JSE']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
d')
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
```

```
Fetching data for ^HSI...
Fetching data for ^ASX200...
Fetching data for ^STOXX50E...
Fetching data for ^AEX...
Fetching data for ^BEL20...
Fetching data for ^IBEX...
Fetching data for ^BSE...
Fetching data for ^NSEI...
Fetching data for ^KOSPI...
Fetching data for ^TSEC...
Fetching data for ^KLSE...
Fetching data for ^JSE...
Fetching data for ^BVSP...
Fetching data for ^MXX...
Fetching data for ^IPSA...
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^IPSA']: YFPricesMissingError('$%ticker%: possibly delisted; no price data
found  (1d 2024-11-21 -> 2024-11-24)')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^RTS']: YFPricesMissingError('$%ticker%: possibly delisted; no price data
found  (1d 2024-11-21 -> 2024-11-24)')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^TA35']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
d')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^SENSEX']: YFTzMissingError('$%ticker%: possibly delisted; no timezone fou
nd')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^SET50']: YFPricesMissingError('$%ticker%: possibly delisted; no price dat
a found  (1d 2024-11-21 -> 2024-11-24)')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^SSEC']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
d')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^TWSE']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
d')
```

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

1 Failed download:
['^IDX']: YFPricesMissingError('$%ticker%: possibly delisted; no price data
found  (1d 2024-11-21 -> 2024-11-24)')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^CSI300']: YFTzMissingError('$%ticker%: possibly delisted; no timezone fou
nd')
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^PSI20']: YFPricesMissingError('$%ticker%: possibly delisted; no price dat
a found  (1d 2024-11-21 -> 2024-11-24)')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^OMXHPI']: YFPricesMissingError('$%ticker%: possibly delisted; no price da
ta found  (1d 2024-11-21 -> 2024-11-24)')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^OMXS30']: YFPricesMissingError('$%ticker%: possibly delisted; no price da
ta found  (1d 2024-11-21 -> 2024-11-24)')
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^SPTSX']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
d')
```

```
Fetching data for ^RTS...
Fetching data for ^TA35...
Fetching data for ^SENSEX...
Fetching data for ^SET50...
Fetching data for ^SSEC...
Fetching data for ^TWSE...
Fetching data for ^IDX...
Fetching data for ^CSI300...
Fetching data for ^NZ50...
Fetching data for ^PSI20...
Fetching data for ^OMXHPI...
Fetching data for ^OMXS30...
Fetching data for ^GDAXI...
Fetching data for ^SPTSX...
Fetching data for ^OSEBX...
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^OSEBX']: YFPricesMissingError('$%ticker%: possibly delisted; no price dat
a found  (1d 2024-11-21 -> 2024-11-24)')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^ATHEX']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
d')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^KSE']: YFPricesMissingError('$%ticker%: possibly delisted; no price data
found  (1d 2024-11-21 -> 2024-11-24)')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^IBOV']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
d')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^TAIEX']: YFTzMissingError('$%ticker%: possibly delisted; no timezone foun
d')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^PX']: YFPricesMissingError('$%ticker%: possibly delisted; no price data f
ound  (1d 2024-11-21 -> 2024-11-24)')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^BIST100']: YFTzMissingError('$%ticker%: possibly delisted; no timezone fo
und')
```

```
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^ALLORDS']: YFTzMissingError('$%ticker%: possibly delisted; no timezone fo
und')
[**********************100%**********************]  1 of 1 completed
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^IBOVESPA']: YFTzMissingError('$%ticker%: possibly delisted; no timezone f
ound')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^IBEX35']: YFTzMissingError('$%ticker%: possibly delisted; no timezone fou
nd')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^TADAWUL']: YFTzMissingError('$%ticker%: possibly delisted; no timezone fo
und')
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['^SAX']: YFPricesMissingError('$%ticker%: possibly delisted; no price data
found  (1d 2024-11-21 -> 2024-11-24)')
Fetching data for ^ATHEX...
Fetching data for ^KSE...
Fetching data for ^IBOV...
Fetching data for ^TAIEX...
Fetching data for ^PX...
Fetching data for ^BIST100...
Fetching data for ^MERV...
Fetching data for ^ALLORDS...
Fetching data for ^JKSE...
Fetching data for ^IBOVESPA...
Fetching data for ^IBEX35...
Fetching data for ^TADAWUL...
Fetching data for ^SAX...
Fetching data for ^NYA...
[**********************100%**********************]  1 of 1 completed
```

```
Binary Trends for Each Index:
{'^GSPC': [1, 1], '^DJI': [1, 1], 'QQQ': [0, 1], '^IXIC': [0, 1], '^RUT': [
1, 1], '^RUA': [1, 1], '^W5000': [1, 1], '^MID': [1, 1], '^VIX': [0, 0], '^N
DX': [0, 1], '^FTSE': [1, 1], '^DAX': [None, None, None], '^CAC40': [None, N
one, None], '^N225': [0, 1], '^HSI': [0, 0], '^ASX200': [None, None, None],
'^STOXX50E': [1, 1], '^AEX': [1, 1], '^BEL20': [None, None, None], '^IBEX':
[1, 1], '^BSE': [None, None, None], '^NSEI': [0, 1], '^KOSPI': [None, None,
None], '^TSEC': [None, None, None], '^KLSE': [0, 0], '^JSE': [None, None, No
ne], '^BVSP': [0, 1], '^MXX': [1, 1], '^IPSA': [None, None, None], '^RTS': [
None, None, None], '^TA35': [None, None, None], '^SENSEX': [None, None, Non
e], '^SET50': [None, None, None], '^SSEC': [None, None, None], '^TWSE': [Non
e, None, None], '^IDX': [None, None, None], '^CSI300': [None, None, None], '
^NZ50': [1, 1], '^PSI20': [None, None, None], '^OMXHPI': [None, None, None],
'^OMXS30': [None, None, None], '^GDAXI': [1, 1], '^SPTSX': [None, None, Non
e], '^OSEBX': [None, None, None], '^ATHEX': [None, None, None], '^KSE': [Non
e, None, None], '^IBOV': [None, None, None], '^TAIEX': [None, None, None], '
^PX': [None, None, None], '^BIST100': [None, None, None], '^MERV': [0, 1], '
^ALLORDS': [None, None, None], '^JKSE': [0, 1], '^IBOVESPA': [None, None, No
ne], '^IBEX35': [None, None, None], '^TADAWUL': [None, None, None], '^SAX':
[None, None, None], '^NYA': [1, 1]}
Binary Feature List for Classification:
[1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0,
0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1]
/var/folders/qj/lwff4ygj6r728t_ytn73v3sw0000gn/T/ipykernel_35179/1758677038.
py:27: FutureWarning: Calling int on a single element Series is deprecated a
nd will raise a TypeError in the future. Use int(ser.iloc[0]) instead
  trends.append(int(row['Close'] > row['Open']))
```

In [5]:
```python
#text cleaning
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

# Download necessary NLTK data files
nltk.download('stopwords')
nltk.download('wordnet')

# Initialize lemmatizer and stopwords
lemmatizer = WordNetLemmatizer()
stop_words = set(stopwords.words('english'))

def clean_text(text):
    # Remove punctuation and non-alphabetic characters
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    # Convert text to lowercase
    text = text.lower()
    # Tokenize and remove stopwords, then lemmatize
    words = text.split()
    words = [lemmatizer.lemmatize(word) for word in words if word not in stc
```

```
    return ' '.join(words)

# Apply text cleaning to the content in articles_data
for article in articles_data:
    article['cleaned_content'] = clean_text(article['full_content'])
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/tiff1101/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/tiff1101/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

In [6]:
```
from sklearn.feature_extraction.text import TfidfVectorizer

# Extract cleaned content from articles for TF-IDF transformation
corpus = [article['cleaned_content'] for article in articles_data]

# Initialize TF-IDF Vectorizer
tfidf_vectorizer = TfidfVectorizer(max_features=1000)  # Adjust max_features
tfidf_matrix = tfidf_vectorizer.fit_transform(corpus)

# Convert the TF-IDF matrix to a DataFrame for better handling
import pandas as pd
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=tfidf_vectorizer.get
```

In [7]:
```
from transformers import BertTokenizer, BertModel
import torch

# Initialize BERT tokenizer and model
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertModel.from_pretrained('bert-base-uncased')

def get_bert_embeddings(text):
    # Tokenize and convert to tensor
    inputs = tokenizer(text, return_tensors='pt', truncation=True, padding=T
    with torch.no_grad():
        outputs = model(**inputs)
    # Use the mean of the last hidden state as the sentence embedding
    embeddings = torch.mean(outputs.last_hidden_state, dim=1)
    return embeddings.squeeze().numpy()

# Apply BERT embeddings to the cleaned content
for article in articles_data:
    article['bert_embedding'] = get_bert_embeddings(article['cleaned_content
```

```
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/tqdm/auto.py:21: TqdmW
arning: IProgress not found. Please update jupyter and ipywidgets. See http
s://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

In [8]:
```
print(article['cleaned_content'])
```

```
founded motley fool financial service company dedicated making world smarter
happier richer motley fool reach million people every month premium investin
g solution free guidance market analysis foolcom personal finance education
toprated podcasts nonprofit motley fool foundation founded motley fool finan
cial service company dedicated making world smarter happier richer motley fo
ol reach million people every month premium investing solution free guidance
market analysis foolcom personal finance education toprated podcasts nonprof
it motley fool foundation youre reading free article opinion may differ motl
ey fool premium investing service become motley fool member today toget inst
ant access top analyst recommendation indepth research investing resourcesan
d morelearn youre reading free article opinion may differ motley fool premiu
m investing service become motley fool member today toget instant access top
analyst recommendation indepth research investing resource morelearn glbe ea
rnings call period ending september image source motley fool globale onlineg
lbeq earnings callnov et operator welcome globale third quarter earnings ann
ouncement conference call call simultaneously webcast company website invest
or relation section news event opening remark introduction turn call erica m
annion sapphire investor relation please go ahead erica mannioninvestor rela
tion thank good morning today globale amir schlachet cofounder chief executi
ve officer ofer koren chief financial officer nir debbie cofounder president
amir begin review business result third quarter ofer review financial result
third quarter followed company outlook fourth quarter full year open call qu
estion certain statement make today may constitute forwardlooking statement
information within meaning section security act section e security exchange
act safe harbor provision u private security litigation reform act relate cu
rrent expectation view future event forwardlooking statement subject risk un
certainty assumption beyond control addition forwardlooking statement reflec
t current view respect future event guarantee future performance actual outc
ome may differ materially information contained forwardlooking statement res
ult number factor including set forth section titled risk factor prospectus
filed sec september document filed furnished sec statement reflect managemen
t current expectation regarding future event operating performance speak dat
e call put undue reliance forwardlooking statement although believe expectat
ion reflected forwardlooking statement reasonable cannot guarantee future re
sult level activity performance event circumstance reflected forwardlooking
statement achieved occur except required applicable law make obligation upda
te revise publicly forwardlooking statement whether result new information f
uture event otherwise date statement made reflect occurrence unanticipated e
vent please refer press release dated november additional information additi
on certain metric discus today nongaap metric presentation financial informa
tion intended considered isolation substitute superior financial information
prepared presented accordance gaap use nongaap financial measure financial o
perating decisionmaking mean evaluate periodtoperiod comparison believe meas
ure provide useful information operating result enhance overall understandin
g past financial performance future prospect allow greater transparency resp
ect key metric used management financial operating decisionmaking informatio
n nongaap financial measure please see reconciliation table provided press r
elease dated november throughout call provide number key performance indicat
```

or used management often used competitor industry key performance indicator discussed detail press release dated november turn call amir cofounder ceo amir schlachetcofounder chief executive officer thank erica welcome everyone q earnings call delivered strong third quarter thanks growing contribution new merchant launch within quarter well growth existing merchant q saw gmv revenue growth accelerating respectively well top range guidance strong topline growth accompanied continued efficiency measure cost control yielding adjusted ebitda growth rate almost midpoint guidance range given successful launch new merchant trading well positive consumer sentiment september expect growth continue accelerating q rate year year raising fullyear guidance accordingly expect annual gmv full year grow revenue grow adjusted ebitda grow almost year year moreover strong performance across board combined new merchant booking alltime high strengthen conviction ability hit mark gmv growth beyond financial metric would also like give update strategic poster main initiative know globales mission power better global ecommerce clear leader global ecommerce service nevertheless invest adding new feature new functionality time continuously transform way merchant shopper across world engage one another example q went live new clickandcollect option bopis buy online pick store enabling relevant merchant enhance multichannel reach global level merchant using bopis utilize physical store different country around world local fulfillment hub global ecommerce well convenient pickup location shopper thereby enabling better shopper experience efficient global stock utilization rise physical store footfall first u merchant utilize capability saw around ecommerce sale canada managed incountry branded store first month offering option shopper operational side started beta testing several additional direct injection lane outbound inbound several new consolidated return lane strive provide merchant bestinclass logistics offering around globe enhancing efficiency offering better diverse set option shopper duty tax front comprehensive duty drawback platform support additional option standard shipping mail granted early access several merchant new global duty drawback program gddp new program enables merchant provide better customer service simplified transparent hasslefree duty tax refund process leveling global shopper experience domestic one last least continue invest aibased capability evolving landscape large language model llm enables u make incredible leap service efficiency example q rolled new iteration ai product classification tool leverage cuttingedge llm generate precise mapping product merchant catalog respective harmonized customer code h code crucial prerequisite accurate duty tax calculation well shipping import restriction management compared previous system utilize firstgeneration machine learning algorithm new llmbased tool yield increased accuracy ability learn improve time constantly bettering performance platform continues evolve merchant sign go live u mentioned earlier third quarter continued onboard many new merchant across different market vertical north america went live leading fashion brand fastgrowing plussize fashion brand torrid emerging elevated basic twin brand nuuds tone iconic footwear brand dr marten name vertical went live raycon fastgrowing consumer electronics brand watchmaker fossil cosmetic brand jones road beauty vitamin brand dr berg quarter also completed successful migration several brand old borderfree platform onto globale including llbean land end oscar de la renta enjoy benefit trading cuttingedge platform service also went live many new brand across uk europe including significant launch sport vertical two top premier league bu

ndesliga football club manchester united bayern munich deepening penetration track record within vertical also went live many brand across various vertic al including german ski lifestyle brand bogner british clothing brand trapst ar london swiss sciencebased sportswear brand xbionic swedish horse riding b rand equestrian stockholm fastgrowing swedish swimwear brand bright ukbased land rover spare part site lr part polish clubwear brand misbhv danish handb ag brand vee collective many many others asia pacific continued exhibit fast growth q gaining share activity apacbased brand choose go global via platfor m example would leading australian fashion brand dissh brand offs site japan hong kong japanese preowned fashion seller ragtag hong kongbased leather bra nd cafune korean sport gear brand dimito q also saw several addition growing portfolio luxury brand notably web store derek rose luxury fashion designer margaret howell uk longchamp paco rabanne france famous watch jewelry brand chopard switzerland biggest news merchant front recent launch harrod probabl y world iconic luxury department store last list planned large merchant laun ched planned launch year integration scayle platform chosen harrod support o nline sale delivered fully localized topoftheline shopping experience every harrod customer around globe including domestic customer uk well time tradit ional holiday sale extremely excited supporting online business harrod much look forward growing together coming year apart launch new merchant always a lso continue expand activity existing brand brand group part landandexpand m otion quarter added remaining lane victoria secret thereby completing phased launch planned also went live disney australia new zealand added support mar ket assoulines apac store france went live luxury eyewear brand vuarnet expa nding list brand work within lvmh group also went live swiss design fashion brand strellson second brand go live u three brand form holy fashion group w ell iconic swimmer brand billabong third brand go live liberated brand group new merchant booking year alltime high continue see strong demand service ac ross geography vertical serve moving additional element road map continue de velop roll new capability shopify managed market continues grow planned gene rate great value thousand merchant using among new feature released past qua rter order editing provides merchant ability add international order prior f ulfillment support several new alternative payment method upgrade automated catalog analysis algorithm allowing merchant sell product market q also expa nded strategic partnership digital transformation leader transcosmos followi ng successful joint work japan extended partnership transcosmos also cover s outh korea work together support international online growth korean client f inally im happy announce planned recently launched demand generation offerin g based revamped borderfreecom brand discovery portal hundred merchant alrea dy signed launch displaying product portal tapping fastgrowing mass internat ional shopper wear world starting fashion journey borderfreecom launch suppo rted outofhome medium campaign online campaign across multiple channel shopp er cashback reward offering many participating merchant already signed well still early inning already see increased level engagement activity portal be lieve time demand generation offering spearheaded borderfreecom grow provide value merchant hand ofer take quarterly figure detail present updated guidan ce remainder year ofer korenchief financial officer thank amir thanks joinin g u today earnings call pleased performance third quarter q experienced acce lerated growth combined healthy margin strong growth driven contribution new merchant successfully onboarded launched planned improvement consumer sentim

ent september id like point addition gaap result ill also discussing certain nongaap result gaap financial result along reconciliation gaap nongaap resul t found earnings release gmv continued grow rapidly q generated billion gmv increase year year midpoint guidance quarter strong gmv growth fueled positi ve consumer sentiment september trading volume generated new merchant launch ed upper side expectation q generated total revenue million year year servic e fee revenue million fulfillment service revenue million quarter characteri zed balanced growth service fee take rate slightly decreased q due loss ted baker account provided demand generation service high service fee take rate fulfillment service take rate increased compared q driven favorable mix nong aap gross profit outpaced topline growth q nongaap gross profit million year year representing nongaap gross margin compared period last year driven oper ational efficiency gaap gross profit million representing margin moving oper ational expense continue invest development enhancement platform rd expense q excluding stockbased compensation million revenue compared million revenue period last year total rd spend q million also continued invest sale marketi ng continue see strong pipeline ahead sale marketing expense excluding shopi fyrelated amortization expense stockbased compensation acquisitionrelated in tangible amortization million revenue compared million revenue period last y ear shopify warrantrelated amortization expense million total sale marketing expense quarter million general administrative expense excluding stockbased compensation acquisitionrelated contingent consideration million revenue com pared million revenue period last year total ga spend q million adjusted ebi tda totaled million representing adjusted ebitda margin increase million mar gin period last year net loss million compared net loss million yearago peri od net loss driven mainly amortization expense related shopify warrant trans actionrelated intangible believe despite impact item mentioned close gaap br eakeven q thanks growth peak trading period amortization expense related sho pify warrant expected decrease significantly april end december believe trac k turn gaap profitable mid making firstever gaap profitable year switching g ear turning balance sheet cash flow statement continue generate cash quarter cash flow generated operating activity million compared million year ago end ed quarter million cash cash equivalent including shortterm deposit marketab le security moving financial outlook introducing guidance q raising fullyear guidance q expecting gmv range billion billion midpoint range represents gro wth rate versus q expect q revenue range million million midpoint range repr esents growth rate versus q adjusted ebitda expecting profit range million m illion midpoint range represents growth rate versus q full year raising guid ance anticipate gmv range billion billion representing annual growth middle point range revenue expected range million million representing growth rate midpoint range adjusted ebitda expecting profit million million representing growth rate midpoint range conclusion opportunity u immense continue enable merchant around globe unlock potential global directtoconsumer business rema in focused execution believe continue grow rapidly generating growing free c ash flow year ahead amir nir happy answer question might operator operator t hank operator instruction first question come william nance goldman sachs li ne open william nanceanalyst hey guy nice result morning appreciate taking q uestion great hear commentary profitability outlook wondering note ebitda ma rgin coming strong wondering could talk around outlook free cash flow conver sion ebitda think historically youve elevated free cash flow conversion four

th quarter dip first quarter wondering could speak seasonality there still s
omething expect thank ofer korenchief financial officer sure thank ofer yes
mentioned seasonality cash flow free cash flow particular typically q q quit
e standard due peak season q positive working capital typically around q str
ong quarter q way around typically slowest quarter term cash flow generally
speaking look full year put aside seasonality free cash flow tends slightly
adjusted ebitda level expect strong q term free cash flow william nanceanaly
st great thats great hear fullyear number helpful announced product enhancem
ent buy online pick store thats kind operationally intensive get product rol
led im wondering could talk kind client reception product enhancement youre
thinking pricing pricing power next couple year thank nir debbicofounder pre
sident hi nir thank question continue invest product enhance solution amir m
entioned bopis weve seen great traction first large enterprise client rolled
shop canada shop canada aiming provide service many enterprise client global
network physical retail order enable global multichannel great customer expe
rience also field lot product enhancement optimize cost trade merchant duty
drawback globale duty drawback program recently released market intend conti
nue build capability inherently increase slightly take rate example duty dra
wback new service merchant adopt basically outofpocket money save money recl
aim duty expense return however take cut yes able increase take rate however
merchant see increase fee contrary save cost create efficiency place happy e
njoy extra margin general dont expect foreseeable future increase price unde
rstand merchant going difficult time interest rate cost medium buying etc tr
ying make efficient trade trying utilizing pricing power william nanceanalys
t thats great hear pricing value appreciate taking question amir schlachetco
founder chief executive officer thanks lot operator next question come andre
w bauch well fargo line open andrew bauchanalyst hey guy thanks taking quest
ion wanted hit demand gen platform hundred merchant said pipeline give u sen
se big merchant anticipate activity level would could reconfirm believe youv
e mentioned past demand gen would gross margin uplift maybe could kind put w
ay dollar frame opportunity nir debbicofounder president hi andrew thank nir
mentioned yes weve recently launched borderfreecom cornerstone demand genera
tion initiative partner brand still sorry still early inning excited longter
m potential contribution brand coming year grows start adopt believe coming
year affect effect also ability generate additional take think short term co
ming quarter focus would penetration product making sure create value brand
seeing increase traffic couple hundred brand came already registered midsize
largest joined already think gain traction able demonstrate value bring merc
hant see client coming stage dont think impact take rate top line believe bu
ilding coming quarter see adoption rate value create might start see tractio
n andrew bauchanalyst great followup would want talk managed market last tim
e spoke believe mentioned tracking ahead plan guess kind think product persp
ective also tracking ahead plan think could mean far adoption managed market
year ahead amir schlachetcofounder chief executive officer yes hi amir said
managed market indeed tracking well go adding new feature new capability tha
nks continuing gain traction within shopify merchant base expect trend conti
nue next year product gradually improves expect merchant adoption curve impr
ove well andrew bauchanalyst great thank amir amir schlachetcofounder chief
executive officer thanks andrew operator next question come koji ikeda bank
america line open koji ikedaanalyst yes hey guy thanks taking question wante

d ask question gmv guide fourth quarter kind looking midpoint guide implies gmv adding million sequentially thats pretty big absolute number year year look q last year sequential net add gmv million know growth wanted kind dig little bit definitely hear launch good news seeing transaction trend quarter date thats giving confidence guide ofer korenchief financial officer yes thank koji ofer basically seen two driver behind basically build confidence around q guide first one mentioned merchant successful merchant launch throughout second half year obviously call many name including larger one victoria secret harrod manchester united others basically launched plan mentioned they re trading well theyre trading higher side expectation definitely one strong driver second one improved consumer sentiment first unlike first half surprisingly stable second half continues volatile term consumer sentiment saw mentioned previous quarter softness july deep going august however september picked nicely continued october beginning november know control may may continue volatile havent incorporated highest number seen past month two also much encouraged compared saw august also driving growth operator next question come scott berg needham line open scott berganalyst hi everyone really nice quarter guess two well start booking revenue growth year date dont know going firmer near think composition booking looked like year date know get net revenue retention fourth quarter number shifted landed really large company last year know launch went well launch larger typical didnt know youre seeing customer year try land really big market versus historical cadence might little bit small strong expansion activity time nir debbicofounder president yes hi scott nir happy yeartodate result booking record level booking seen great momentum market large enterprise enterprise signing going live victoria secret actually gave u global ecommerce outside specific distribution market local distribution market go harrod took global market also domestic market uk part multilocal strategy seen also manchester united weve taken global expansion also local market part multilocal strategy see positive effect large merchant actually launching virtually almost everything give global ecommerce even though domestic ecommerce part think see robust pipeline also looking optimistic advanced stage funnel see conversion lead deal staying strong quite optimistic going continue also going scott berganalyst helpful thank ofer kind noted ga expense quarter quarter seasonal nature much weve seen historically anything led im really trying think modeling q next year might lower rate sustainably going forward ofer korenchief financial officer yes oneoff expense related timing really oneoffs q basically look q andor maybe somethin g middle two quarter representing pace going forward obviously growing expect expense grow time base think look midpoint quarter operator next question come samad samana jefferies line open samad samanaanalyst hi good morning thanks taking question maybe first going back borderfreecom guess think youre embedding q expectation launch anything think maybe take rate implication youre thinking fourth quarter next year ofer korenchief financial officer hi samad thanks question ofer q embedding mainly expense related boderfreecom launched product expense associated dont expect see revenue yet nir mentioned currently main objective create value merchant create confidence service would say short term going least first half next year nothing significant top line expense related service already embedded q guide samad samanaanalyst great know mentioned booking prior question well guess comment harrod last major launch year imply large merchant youve booked point well wait go live gues

s case think maybe seasonality youre thinking based golives looking like sim
ilar first half nir debbicofounder president yeah understood robust pipeline
already signed advanced stage project motion would launch year dont think wo
uld see material change timing versus weve seen expect would say couple dece
nt merchant launch first half dont think q would see significant contributio
n havent year new merchant think q would start see rise operator next questi
on come james faucette morgan stanley line open james faucetteanalyst great
thank much wanted ask quickly take rate little bit clarification mentioned t
ed baker one driver service fee take rate decel strip would service fee stab
le maybe slightly versus last quarter similarly fulfillment alluded broadbas
ed aov increase last quarter compressing fulfillment take rate factor quarte
r think trajectory near medium term ofer korenchief financial officer hi jam
es ofer thank question term service fee take rate yes mentioned main impact
ted baker account basically put aside quite stable quarter expect see stabil
ity side going forward term fulfillment take rate always much volatile servi
ce fee different parameter impacting aov mix multi local forth quarter posit
ive mix positive impact actually expected expectation met reality reality me
t expectation term aov actually slightly decreased q havent seen trend q con
tinuing look forward try think fulfillment take rate would think q take rate
bit lower embedded guidance would good base next year quarter slightly highe
r see ongoing take rate operator next question come chris zhang ubs line ope
n chris zhangubs analyst hey good morning thanks taking question question wh
ats embedded q guidance term merchant launch told youve talked completing la
st merchant launch year also number smaller merchant midsized merchant assum
e better visibility part spotify managed market wondered whats rampup patter
n throughout fourth quarter last year since turnkey solution based understan
ding lot easier turn expect continued onboarding new shopify managed market
merchant throughout rest fourth quarter even holiday shopping season ofer ko
renchief financial officer yes thank question ofer think generally speaking
every year see less significant launch calendar typically around october som
etimes see going november smaller merchant also onboarding launching general
ly speaking larger merchant decentsized merchant wouldnt go october everybod
y preparing peak season peak behind u typically everybody go holiday season
see launch december nothing significant would impact result typically start
seeing merchant getting back work sort prioritizing project midjanuary begin
ning february typically see additional launch coming toward end q q would st
andard cadence seeing well operator next question come brian peterson raymon
d james line open unknown speaker analyst hi thanks taking question john bri
an wanted ask sale cycle enterprise customer realize customer journey going
different there certainly seasonality holiday wondering could speak timing s
ale cycle trended versus earlier year attracted larger larger brand platform
maybe helping pace sale cycle quick followup nir debbicofounder president hi
nir thank question enterprise client think last two year three year see impr
ovement sale cycle timing come multiple factor u gaining much experience onb
oarding launching enterprise merchant getting would say clarification financ
ial stability etc much quicker due positioning publicly traded company stabl
e company generates cash every quarter going multiple factor sharpened year
see additional potential improvement dont think material difference take sel
l enterprise large enterprise client sale cycle typically run around six mon
th launching would realm three month four month depending phase depending co

mplexity ask client enterprise client would launch impact already sale funne
l said earlier already signed early advanced stage back see converting launc
hing throughout year unknown speaker analyst great thanks thats really good
color maybe followup earlier commentary managed market wondering could share
update youve added feature youre seeing maybe merchant size changing also co
lor pace merchandise managed market thank amir schlachetcofounder chief exec
utive officer yes sure amir yes expected see gradual increase pace somewhat
also size number slightly larger merchant joining gradual process dependent
collection various feature applicable various merchant dont expect kind majo
r stepups continuous gradual improvement roll feature operator next question
come patrick walravens citizen jmp line open patrick walravensanalyst great
thank congratulation yeah amir think possible impact would business end gett
ing higher tariff potential universal tariff united state amir schlachetcofo
under chief executive officer yes hi pat good question think two side positi
on one hand obviously higher tariff make harder u shopper buy outoftheus mer
chant sell believe long extremely high unreasonable tariff impact probably g
rander scale detrimental let say type good price point mainly deal hand impo
rtant remember business helping merchant overcome trade barrier way addition
al trade barrier case form tariff actually make offering valuable merchant w
ant serve u biggest ecommerce market world take prior example like introduct
ion brexit rule market historically weve seen change like brexit good busine
ss yes uk merchant took hit hand made service much valuable european merchan
t selling uk everyone overcome additional export import challenge depending
side look general thats assumption going forward patrick walravensanalyst aw
esome thank operator operator instruction next question come maddie schrage
keybanc capital market line open maddie schragekeybanc capital market analys
t hey guy wanted go back demand generation portion wondering expense guy inc
urring right going recurring expense kind onetime expense baked guidance sec
ond question geo callouts obviously u seems strong goal additional geo penet
ration thanks ofer korenchief financial officer hi maddie ofer thanks questi
on regarding expense related demand gen theyre onetime expense obviously hig
her expense around launch ongoing expense promoting product traffic borderfr
eecom expense going forward term second question nir debbicofounder presiden
t yes clear expect going forward rate investment demand generation create ma
terial difference level spend sale marketing see today ofer korenchief finan
cial officer regarding second part question mentioned seen better consumer s
entiment around globe call europe continent saw highest rise time would ment
ion really year year look ladenburg base pretty low seen good trade europe d
estination market low base europe least currently well term trading maddie s
chragekeybanc capital market analyst great thanks question operator next que
stion come mark zgutowicz benchmark line open mark zgutowiczanalyst thank ap
ology asked im jumping another earnings call term shopify managed market im
curious target roughly million million year curious gmv contribution coming
lower versus higher end range maybe talk related gmv pickup expect see perha
ps next year perhaps there copromotion shopify ofer korenchief financial off
icer yes thank question think looking year top range broad range within rang
e mentioned already mentioned call going pretty well planned regarding going
forward expect see continuous add merchant coming dont expect see step chang
e expect offering continue grow operator question time turn call amir closin
g remark amir schlachetcofounder chief executive officer thank everyone join

```
ing u call today conclude would like take opportunity thank amazing globale
staff member worked tirelessly past month onboard new merchant support exist
ing merchant make necessary preparation ensure fully ready serve global clie
nt allimportant peak trading period would also like thank ongoing support sh
ared belief vision transform world global directtoconsumer ecommerce head ho
liday period behalf u globale would like wish happy successful holiday seaso
n much look forward updating call continue exciting rapid path capture immen
se opportunity lie ahead u next time goodbye take care operator operator sig
noff duration minute erica mannioninvestor relation amir schlachetcofounder
chief executive officer ofer korenchief financial officer william nanceanaly
st nanceanalyst nir debbicofounder president andrew bauchanalyst koji ikedaa
nalyst scott berganalyst samad samanaanalyst james faucetteanalyst chris zha
ngubs analyst unknown speaker analyst patrick walravensanalyst pat walravens
analyst maddie schragekeybanc capital market analyst mark zgutowiczanalyst g
lbe analysis earnings call transcript article transcript conference call pro
duced motley fool strive foolish best may error omission inaccuracy transcri
pt article motley fool assume responsibility use content strongly encourage
research including listening call reading company sec filing please see ourt
erms conditionsfor additional detail including obligatory capitalized discla
imer liability motley fool position recommends globale online motley fool ad
isclosure policy stock mentioned average return recommendation since incepti
on cost basis return based previous market day close related article invest
better motley fool get stock recommendation portfolio guidance motley fool p
remium service making world smarter happier richer motley fool right reserve
d market data powered byxigniteandpolygonio motley fool service around globe
free tool affiliate friend
```

In [9]:
```python
#part 2
import pandas as pd
import numpy as np
import re
from sklearn.feature_extraction.text import TfidfVectorizer
from transformers import BertTokenizer, BertModel
import torch
from sklearn.preprocessing import StandardScaler

# Assuming articles_data is already loaded and contains 'cleaned_content'
# Extract cleaned content from articles for TF-IDF transformation
corpus = [article['cleaned_content'] for article in articles_data]
print("Corpus prepared for transformation.")


### Preprocess the Corpus (if needed)
def preprocess_text(text):
    """
    Preprocesses the text data by removing special characters, lowercasing,
    """
    text = re.sub(r'\W', ' ', text)  # Remove special characters and punctua
    text = text.lower()  # Convert to lowercase
    text = re.sub(r'\s+', ' ', text)  # Replace multiple spaces with a singl
```

```
    text = text.strip()  # Remove trailing spaces
    return text

# Apply preprocessing (uncomment if corpus needs further processing)
# corpus = [preprocess_text(text) for text in corpus]
# print("Text preprocessing completed.")

### Convert Text to Numerical Features

# Option 1: Using TF-IDF Vectorization
tfidf_vectorizer = TfidfVectorizer(max_features=500, stop_words='english')
tfidf_features = tfidf_vectorizer.fit_transform(corpus).toarray()
tfidf_feature_names = tfidf_vectorizer.get_feature_names_out()

# Add TF-IDF features to articles_data
for i, article in enumerate(articles_data):
    article['tfidf_features'] = tfidf_features[i].tolist()  # Convert each r

print("TF-IDF feature extraction completed.")

# Option 2: Using BERT Embeddings (advanced NLP technique)
# Initialize BERT tokenizer and model
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
bert_model = BertModel.from_pretrained('bert-base-uncased')

def get_bert_embeddings(text):
    """
    Generates BERT embeddings for a given text.
    """
    with torch.no_grad():  # Disable gradient calculation for faster process
        inputs = tokenizer(text, return_tensors='pt', truncation=True, paddi
        outputs = bert_model(**inputs)
        # Pooling method: Using the mean of the last hidden state for senter
        return outputs.last_hidden_state.mean(dim=1).squeeze().numpy()

# Apply BERT embeddings in batches to optimize for large datasets
bert_features = [get_bert_embeddings(text) for text in corpus]

# Add BERT features to articles_data
for i, article in enumerate(articles_data):
    article['bert_features'] = bert_features[i].tolist()  # Convert numpy ar

print("BERT embedding extraction completed.")

import requests

# # Example API call (use your actual API key and modify parameters as neede
# def get_stock_price(symbol, date):
#     api_key = 'G9EC8UT11P0EG92N'
#     url = f'https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_A
```

```
#     response = requests.get(url)
#     data = response.json()

#     # Extract stock price for the given date
#     try:
#         return data['Time Series (Daily)'][date]['5. adjusted close']
#     except KeyError:
#         return None

# # Merge stock prices with article data
# for article in articles_data:

#     try:
#         raw_date = article['time_published'].split('T')[0]  # Extract the
#         formatted_date = datetime.strptime(raw_date, '%Y%m%d').strftime('%
#         article['formatted_date'] = formatted_date
#         article['stock_price'] = get_stock_price('^GSPC', formatted_date)
#     except ValueError as e:
#         print(f"Error processing date for article {article}: {e}")
```

```
Corpus prepared for transformation.
TF-IDF feature extraction completed.
BERT embedding extraction completed.
```

In [10]:
```
#LDA Topic Modeling
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
articles_df = pd.DataFrame(articles_data)
# Vectorize the text
count_vect = CountVectorizer(max_df=0.95, min_df=2, stop_words='english')
dtm = count_vect.fit_transform(articles_df['cleaned_content'])

# Apply LDA
lda = LatentDirichletAllocation(n_components=5, random_state=42)
lda.fit(dtm)

# Print top words per topic
for idx, topic in enumerate(lda.components_):
    print(f"Topic {idx + 1}:")
    print([count_vect.get_feature_names_out()[i] for i in topic.argsort()[-1
```

```
Topic 1:
['free', 'price', 'etf', 'rank', 'data', 'company', 'report', 'zacks', 'shar
e', 'stock']
Topic 2:
['fashion', 'growth', 'experience', 'walmart', 'quarter', 'market', 'novembe
r', 'analyst', 'share', 'th']
Topic 3:
['trading', 'benzinga', 'buy', 'stock', 'target', 'rating', 'analyst', 'pric
e', 'trade', 'option']
Topic 4:
['service', 'market', 'new', 'merchant', 'best', 'business', 'question', 'br
and', 'quarter', 'year']
Topic 5:
['global', 'statement', 'transaction', 'investor', 'stock', 'share', 'millio
n', 'market', 'financial', 'company']
```

In [11]:

```python
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Assume `lda` is your trained LDA model and `count_vect` is your CountVecto
# Create a word cloud for each topic
def plot_topic_wordclouds(lda_model, feature_names, num_words=10):
    """
    Generate and display word clouds for each topic in the LDA model.

    Parameters:
        lda_model: Trained LDA model
        feature_names: List of feature names from CountVectorizer
        num_words: Number of words to include in the word cloud
    """
    num_topics = lda_model.n_components
    plt.figure(figsize=(15, 10))

    for topic_idx, topic in enumerate(lda_model.components_):
        # Get the top words for the topic
        top_words = {feature_names[i]: topic[i] for i in topic.argsort()[-nu

        # Generate the word cloud
        wordcloud = WordCloud(
            background_color='white',
            width=800,
            height=400,
            max_words=num_words,
            colormap='viridis'
        ).generate_from_frequencies(top_words)

        # Plot the word cloud
        plt.subplot((num_topics + 2) // 3, 3, topic_idx + 1)  # Adjust subpl
        plt.imshow(wordcloud, interpolation='bilinear')
        plt.axis('off')
```

```
        plt.title(f"Topic {topic_idx + 1}", fontsize=14)

    plt.tight_layout()
    plt.show()

# Use feature names from the CountVectorizer
feature_names = count_vect.get_feature_names_out()

# Call the function to plot word clouds
plot_topic_wordclouds(lda, feature_names, num_words=10)
```

In [12]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.ensemble import RandomForestClassifier
import torch
import torch.nn as nn
import joblib

# Assuming articles_data is your original data and binary_feature_list conta
# Convert articles_data to a DataFrame
articles_df = pd.DataFrame(articles_data)

# Ensure binary_feature_list has the correct length
if len(binary_feature_list) != len(articles_df):
    raise ValueError("The length of binary_feature_list does not match the r

# Add the binary_feature_list to the DataFrame as a new column
articles_df['binary_trend'] = binary_feature_list

# Drop rows with missing values in the 'binary_trend' or 'overall_sentiment_
```

```
articles_df.dropna(subset=['binary_trend', 'overall_sentiment_score'], inpla

# Extract features from TF-IDF and/or BERT embeddings
tfidf_features = np.array(articles_df['tfidf_features'].tolist())
bert_features = np.array(articles_df['bert_features'].tolist())

# Add overall_sentiment_score as a feature
sentiment_scores = articles_df['overall_sentiment_score'].values.reshape(-1,

# Combine TF-IDF, BERT features, and sentiment scores
X = np.hstack([tfidf_features, bert_features, sentiment_scores])

# Use the binary trends as the target variable
y = articles_df['binary_trend']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rar

# Train a Random Forest Classifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Save the trained Random Forest model
joblib.dump(rf_model, "/Users/tiff1101/Downloads/trained_rf_classifier_model

# Make predictions using the Random Forest model
y_pred_rf = rf_model.predict(X_test)

# Evaluate the model's performance
accuracy_rf = accuracy_score(y_test, y_pred_rf)
print(f"Random Forest Accuracy: {accuracy_rf}")
print(f"Classification Report:\n{classification_report(y_test, y_pred_rf)}")

# Prepare data for LSTM (reshaping to [samples, time steps, features])
X_train_lstm = X_train.reshape(X_train.shape[0], 1, X_train.shape[1])
X_test_lstm = X_test.reshape(X_test.shape[0], 1, X_test.shape[1])

# Define LSTM model class
class StockPredictorLSTM(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(StockPredictorLSTM, self).__init__()
        self.lstm = nn.LSTM(input_size, hidden_size, batch_first=True)
        self.fc = nn.Linear(hidden_size, output_size)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        _, (hn, _) = self.lstm(x)
        out = self.fc(hn[-1])
        return self.sigmoid(out)
```

```
# Model parameters
input_size = X_train.shape[1]
hidden_size = 50
output_size = 1  # For binary classification

# Initialize LSTM model, loss function, and optimizer
model = StockPredictorLSTM(input_size, hidden_size, output_size)
criterion = nn.BCELoss()  # Binary Cross-Entropy Loss for classification
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)

# Train the LSTM model
epochs = 10
for epoch in range(epochs):
    model.train()
    X_train_tensor = torch.tensor(X_train_lstm, dtype=torch.float32)
    y_train_tensor = torch.tensor(y_train.values, dtype=torch.float32).view(

    optimizer.zero_grad()
    outputs = model(X_train_tensor)
    loss = criterion(outputs, y_train_tensor)
    loss.backward()
    optimizer.step()

    print(f"Epoch {epoch+1}/{epochs}, Loss: {loss.item()}")

# Save the trained LSTM model
torch.save(model.state_dict(), "/Users/tiff1101/Downloads/trained_lstm_class

# Evaluate the LSTM model
model.eval()
with torch.no_grad():
    X_test_tensor = torch.tensor(X_test_lstm, dtype=torch.float32)
    y_test_tensor = torch.tensor(y_test.values, dtype=torch.float32).view(-1
    y_pred_lstm = model(X_test_tensor)
    y_pred_lstm_binary = (y_pred_lstm.numpy() > 0.5).astype(int)
    accuracy_lstm = accuracy_score(y_test, y_pred_lstm_binary)
    print(f"LSTM Accuracy: {accuracy_lstm}")
    print(f"Classification Report:\n{classification_report(y_test, y_pred_ls
```

```
Random Forest Accuracy: 0.6
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         3
           1       0.67      0.86      0.75         7

    accuracy                           0.60        10
   macro avg       0.33      0.43      0.38        10
weighted avg       0.47      0.60      0.53        10

Epoch 1/10, Loss: 0.6986969709396362
Epoch 2/10, Loss: 0.6650489568710327
Epoch 3/10, Loss: 0.6381494998931885
Epoch 4/10, Loss: 0.6174789667129517
Epoch 5/10, Loss: 0.6022161245346069
Epoch 6/10, Loss: 0.5914400219917297
Epoch 7/10, Loss: 0.5842071175575256
Epoch 8/10, Loss: 0.5796263217926025
Epoch 9/10, Loss: 0.5769017934799194
Epoch 10/10, Loss: 0.5753427743911743
LSTM Accuracy: 0.7
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         3
           1       0.70      1.00      0.82         7

    accuracy                           0.70        10
   macro avg       0.35      0.50      0.41        10
weighted avg       0.49      0.70      0.58        10
```

```
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

In [13]:

```
#support vector machine
import pandas as pd
```

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.svm import SVC
import joblib

# Assuming articles_data is your original data and binary_feature_list conta
# Convert articles_data to a DataFrame
articles_df = pd.DataFrame(articles_data)

# Ensure binary_feature_list has the correct length
if len(binary_feature_list) != len(articles_df):
    raise ValueError("The length of binary_feature_list does not match the r

# Add the binary_feature_list to the DataFrame as a new column
articles_df['binary_trend'] = binary_feature_list

# Drop rows with missing values in the 'binary_trend' or 'overall_sentiment_
articles_df.dropna(subset=['binary_trend', 'overall_sentiment_score'], inpla

# Extract features from TF-IDF and/or BERT embeddings
tfidf_features = np.array(articles_df['tfidf_features'].tolist())
bert_features = np.array(articles_df['bert_features'].tolist())

# Add overall_sentiment_score as a feature
sentiment_scores = articles_df['overall_sentiment_score'].values.reshape(-1,

# Combine TF-IDF, BERT features, and sentiment scores
X = np.hstack([tfidf_features, bert_features, sentiment_scores])

# Use the binary trends as the target variable
y = articles_df['binary_trend']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rar

# Train a Support Vector Machine Classifier with RBF kernel
svm_model = SVC(kernel='rbf', probability=True, random_state=42)
svm_model.fit(X_train, y_train)

# Save the trained SVM model
joblib.dump(svm_model, "/Users/tiff1101/Downloads/trained_svm_classifier_moc

# Make predictions using the SVM model
y_pred_svm = svm_model.predict(X_test)

# Evaluate the model's performance
accuracy_svm = accuracy_score(y_test, y_pred_svm)
print(f"SVM Accuracy: {accuracy_svm}")
print(f"Classification Report:\n{classification_report(y_test, y_pred_svm)}"
```

```
# For confidence/probability scores of predictions
y_pred_prob_svm = svm_model.predict_proba(X_test)[:, 1]  # Probability for c
print("Prediction Probabilities for Class 1:")
print(y_pred_prob_svm)
#fine tuning later
```

```
SVM Accuracy: 0.7
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         3
           1       0.70      1.00      0.82         7

    accuracy                           0.70        10
   macro avg       0.35      0.50      0.41        10
weighted avg       0.49      0.70      0.58        10

Prediction Probabilities for Class 1:
[0.73143276 0.77904661 0.7688115  0.69297022 0.71724597 0.79211234
 0.65718464 0.64383917 0.70209638 0.05726466]
```

```
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

In [14]:

```
#feedforward neural network
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.utils import to_categorical

# Assuming articles_data is your original data and binary_feature_list conta
```

```
# Convert articles_data to a DataFrame
articles_df = pd.DataFrame(articles_data)

# Ensure binary_feature_list has the correct length
if len(binary_feature_list) != len(articles_df):
    raise ValueError("The length of binary_feature_list does not match the r

# Add the binary_feature_list to the DataFrame as a new column
articles_df['binary_trend'] = binary_feature_list

# Drop rows with missing values in the 'binary_trend' or 'overall_sentiment_
articles_df.dropna(subset=['binary_trend', 'overall_sentiment_score'], inpla

# Extract features from TF-IDF and/or BERT embeddings
tfidf_features = np.array(articles_df['tfidf_features'].tolist())
bert_features = np.array(articles_df['bert_features'].tolist())

# Add overall_sentiment_score as a feature
sentiment_scores = articles_df['overall_sentiment_score'].values.reshape(-1,

# Combine TF-IDF, BERT features, and sentiment scores
X = np.hstack([tfidf_features, bert_features, sentiment_scores])

# Use the binary trends as the target variable
y = articles_df['binary_trend']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran

# Standardize the feature data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Convert target variable to categorical (one-hot encoding)
y_train_categorical = to_categorical(y_train, num_classes=2)
y_test_categorical = to_categorical(y_test, num_classes=2)

# Build the Feedforward Neural Network
model = Sequential([
    Dense(128, input_dim=X_train.shape[1], activation='relu'),  # Input laye
    Dropout(0.3),  # Dropout for regularization
    Dense(64, activation='relu'),  # Hidden layer
    Dropout(0.3),
    Dense(32, activation='relu'),  # Hidden layer
    Dense(2, activation='softmax')  # Output layer (binary classification)
])

# Compile the model
model.compile(optimizer=Adam(learning_rate=0.001),
```

```
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Print model summary
model.summary()

# Define early stopping
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_

# Train the model
history = model.fit(X_train, y_train_categorical,
                    validation_split=0.2,
                    epochs=50,
                    batch_size=16,
                    callbacks=[early_stopping],
                    verbose=1)

# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test_categorical, verbose=0)
print(f"Neural Network Accuracy: {accuracy:.4f}")

# Make predictions
y_pred_nn = model.predict(X_test)
y_pred_classes = np.argmax(y_pred_nn, axis=1)

# Evaluate predictions
print(f"Classification Report:\n{classification_report(y_test, y_pred_classe

# Save the trained model
model.save('/Users/tiff1101/Downloads/trained_dense_nn_with_sentiment.h5')

# Optional: Visualize training history
import matplotlib.pyplot as plt
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Model Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
#number of layers and nodes
```

```
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/keras/src/layers/core/
dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argument
to a layer. When using Sequential models, prefer using an `Input(shape)` obj
ect as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

Model: "sequential"

Layer (type)	Output Shape	Par
dense (Dense)	(None, 128)	162
dropout (Dropout)	(None, 128)	
dense_1 (Dense)	(None, 64)	8
dropout_1 (Dropout)	(None, 64)	
dense_2 (Dense)	(None, 32)	2
dense_3 (Dense)	(None, 2)	

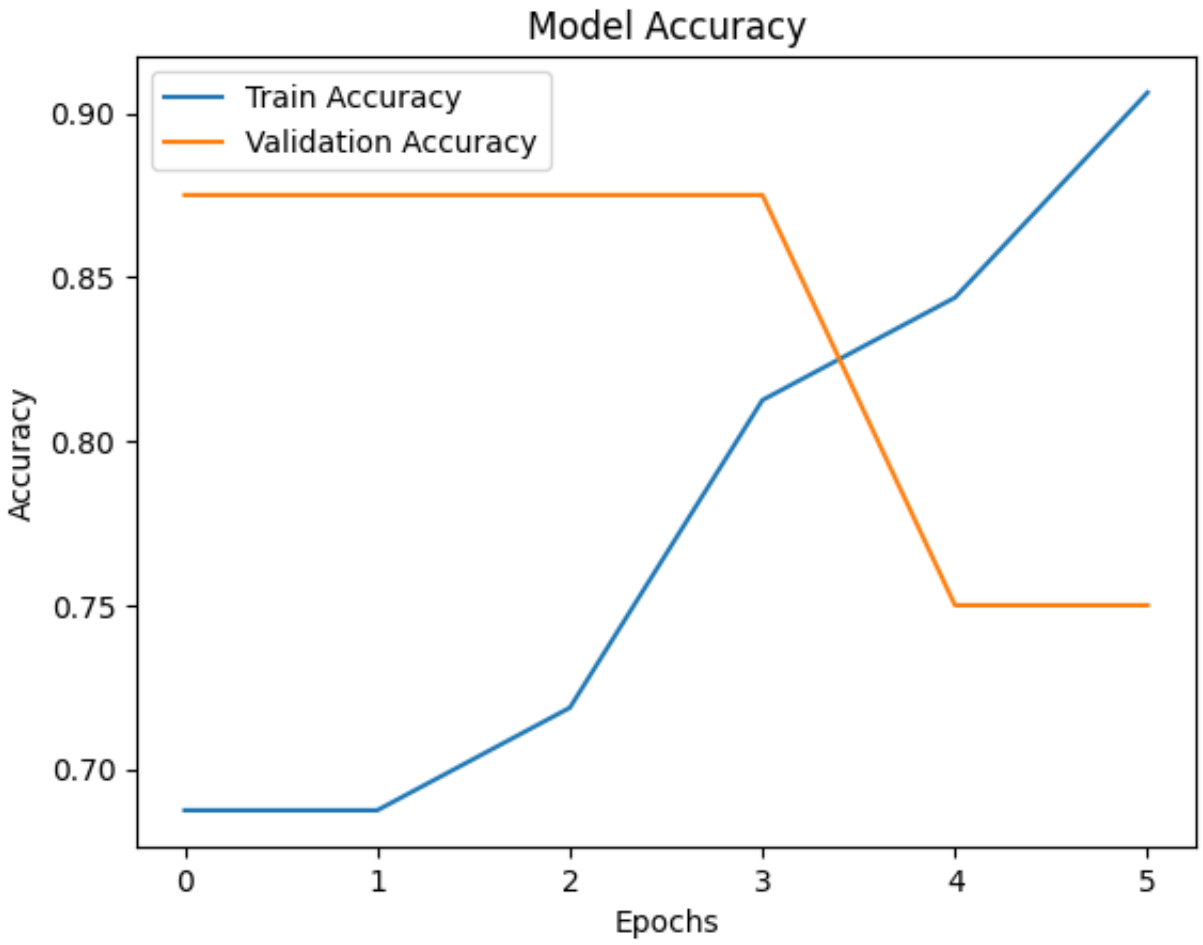
Total params: 172,962 (675.63 KB)

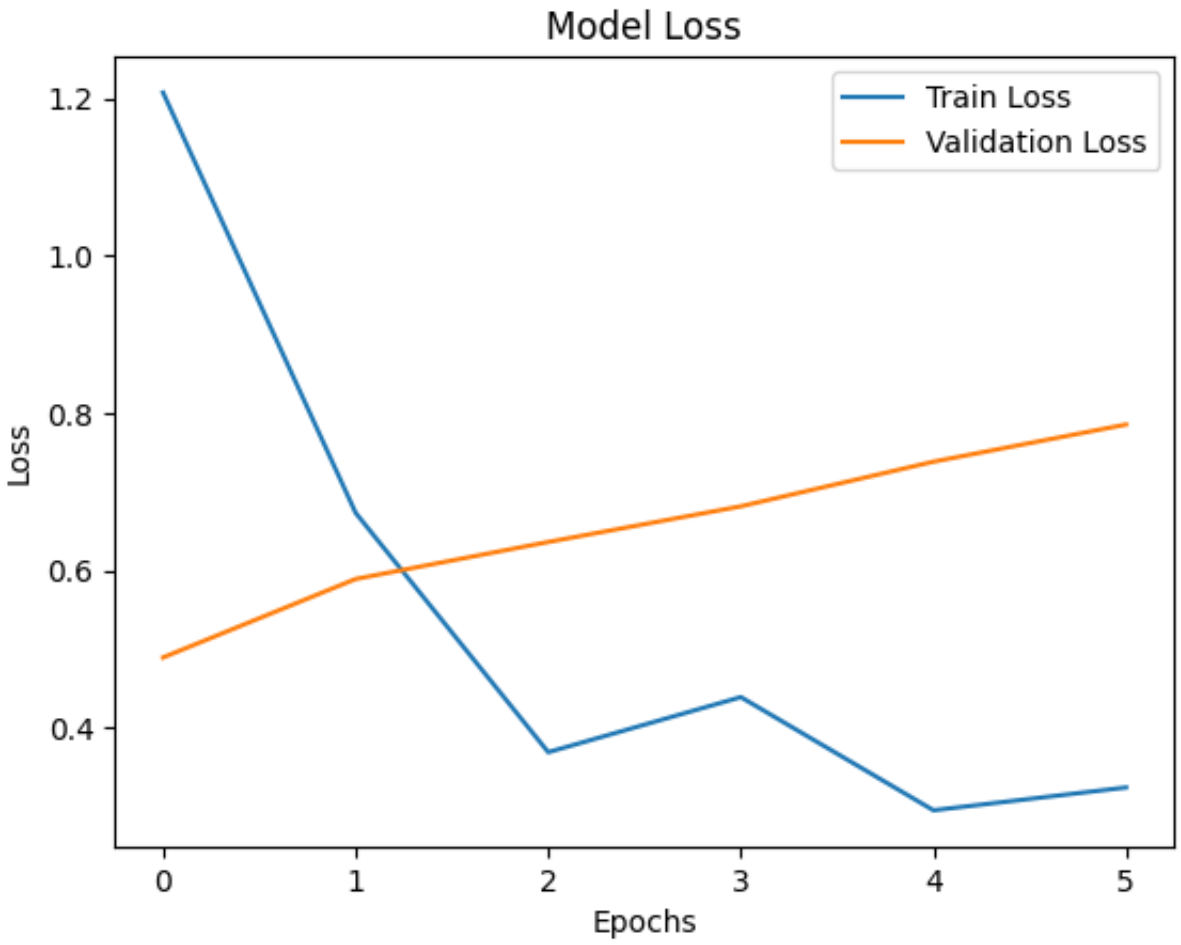
Trainable params: 172,962 (675.63 KB)

Non-trainable params: 0 (0.00 B)

```
Epoch 1/50
2/2 ━━━━━━━━━━━━━━━━━━━━ 1s 70ms/step - accuracy: 0.6458 - loss: 1.4508 - va
l_accuracy: 0.8750 - val_loss: 0.4889
Epoch 2/50
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.6875 - loss: 0.7286 - va
l_accuracy: 0.8750 - val_loss: 0.5888
Epoch 3/50
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.7083 - loss: 0.3620 - va
l_accuracy: 0.8750 - val_loss: 0.6359
Epoch 4/50
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.7917 - loss: 0.4834 - val
_accuracy: 0.8750 - val_loss: 0.6812
Epoch 5/50
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.8333 - loss: 0.3209 - va
l_accuracy: 0.7500 - val_loss: 0.7380
Epoch 6/50
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.9167 - loss: 0.2989 - va
l_accuracy: 0.7500 - val_loss: 0.7853
Neural Network Accuracy: 0.7000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step
```

```
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or
`keras.saving.save_model(model)`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g. `model.save('my_model.
keras')` or `keras.saving.save_model(model, 'my_model.keras')`.
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         3
           1       0.70      1.00      0.82         7

    accuracy                           0.70        10
   macro avg       0.35      0.50      0.41        10
weighted avg       0.49      0.70      0.58        10
```


Model Accuracy
Train Accuracy
Validation Accuracy
Accuracy
0.90
0.85
0.80
0.75
0.70
0
1
2
3
4
5
Epochs



In [15]:
```
#naive bayes

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, classification_report
from sklearn.naive_bayes import MultinomialNB
import joblib

# Assuming articles_data is your original data and binary_feature_list conta
# Convert articles_data to a DataFrame
articles_df = pd.DataFrame(articles_data)

# Ensure binary_feature_list has the correct length
if len(binary_feature_list) != len(articles_df):
    raise ValueError("The length of binary_feature_list does not match the r

# Add the binary_feature_list to the DataFrame as a new column
articles_df['binary_trend'] = binary_feature_list

# Drop rows with missing values in the 'binary_trend' or 'overall_sentiment_
```

```
articles_df.dropna(subset=['binary_trend', 'overall_sentiment_score'], inpla

# Preprocess text data (assuming articles_data has a 'cleaned_content' colum
corpus = articles_df['cleaned_content'].tolist()

# Extract features using TF-IDF
tfidf_vectorizer = TfidfVectorizer(max_features=1000, stop_words='english')
X_tfidf = tfidf_vectorizer.fit_transform(corpus).toarray()

# Normalize sentiment scores to be non-negative
sentiment_scores = articles_df['overall_sentiment_score'].values
min_sentiment_score = sentiment_scores.min()
sentiment_scores_normalized = sentiment_scores - min_sentiment_score  # Shif
sentiment_scores_normalized = sentiment_scores_normalized.reshape(-1, 1)

# Combine TF-IDF features and normalized sentiment scores
X = np.hstack([X_tfidf, sentiment_scores_normalized])

# Use the binary trends as the target variable
y = articles_df['binary_trend']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran

# Train a Naive Bayes Classifier
nb_model = MultinomialNB()
nb_model.fit(X_train, y_train)

# Save the trained Naive Bayes model
joblib.dump(nb_model, "/Users/tiff1101/Downloads/trained_nb_classifier_model

# Make predictions using the Naive Bayes model
y_pred_nb = nb_model.predict(X_test)

# Evaluate the model's performance
accuracy_nb = accuracy_score(y_test, y_pred_nb)
print(f"Naive Bayes Accuracy: {accuracy_nb}")
print(f"Classification Report:\n{classification_report(y_test, y_pred_nb)}")

# Save TF-IDF vectorizer for future use
joblib.dump(tfidf_vectorizer, "/Users/tiff1101/Downloads/tfidf_vectorizer_wi

# Feature analysis
feature_names = tfidf_vectorizer.get_feature_names_out()
top_n = min(20, len(feature_names))  # Ensure we don't request more features
class_labels = nb_model.classes_

print("\nTop features for each class:")
for i, class_label in enumerate(class_labels):
    # Extract the top features for the current class
```

```
    top_features_idx = np.argsort(nb_model.feature_log_prob_[i])[-top_n:]
    top_features = [(feature_names[j], nb_model.feature_log_prob_[i][j]) for
    top_features = sorted(top_features, key=lambda x: x[1], reverse=True)
    print(f"Class {class_label}:")
    for feature, weight in top_features:
        print(f"  {feature}: {weight:.4f}")
```

```
Naive Bayes Accuracy: 0.7
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         3
           1       0.70      1.00      0.82         7

    accuracy                           0.70        10
   macro avg       0.35      0.50      0.41        10
weighted avg       0.49      0.70      0.58        10


Top features for each class:
Class 0:
  option: -6.2374
  bank: -6.2914
  trade: -6.3940
  price: -6.4075
  kinder: -6.4398
  target: -6.4400
  parcel: -6.4673
  archegos: -6.4727
  ftc: -6.4820
  analyst: -6.4842
  astera: -6.4895
  best: -6.4938
  pbms: -6.5266
  morgan: -6.5303
  institution: -6.5337
  rating: -6.5505
  hope: -6.5615
  financial: -6.5640
  lab: -6.5668
Class 1:
  option: -5.8631
  price: -6.0442
  share: -6.1116
  trade: -6.1888
  company: -6.1971
  stock: -6.2122
  analyst: -6.2432
  rating: -6.3423
  market: -6.3426
```

```
trump: -6.3490
allbirds: -6.3542
investor: -6.3579
target: -6.3763
buy: -6.3965
better: -6.3988
trading: -6.4307
data: -6.4405
technology: -6.4409
million: -6.4588
```

```
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

In [16]:

```
#transformer-based NLP model (BERT)
import pandas as pd
import numpy as np
from transformers import BertTokenizer, BertForSequenceClassification
from transformers import AdamW
from torch.utils.data import Dataset, DataLoader
import torch
import torch.nn as nn
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.ensemble import RandomForestClassifier

# Step 1: Dataset Preparation
# Assuming `articles_data` contains your articles and `binary_feature_list`
articles_df = pd.DataFrame(articles_data)

# Ensure binary_feature_list matches the articles
if len(binary_feature_list) != len(articles_df):
    raise ValueError("The length of binary_feature_list does not match the r

# Add the binary_feature_list to the DataFrame
articles_df['binary_trend'] = binary_feature_list

# Drop rows with missing values in text or target
```

```
articles_df.dropna(subset=['cleaned_content', 'binary_trend'], inplace=True)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(
    articles_df['cleaned_content'], articles_df['binary_trend'], test_size=0
)

# Step 2: Custom Dataset for PyTorch
class StockPredictionDataset(Dataset):
    def __init__(self, texts, labels, tokenizer, max_len):
        self.texts = texts
        self.labels = labels
        self.tokenizer = tokenizer
        self.max_len = max_len

    def __len__(self):
        return len(self.texts)

    def __getitem__(self, idx):
        text = str(self.texts.iloc[idx])
        label = self.labels.iloc[idx]

        encoding = self.tokenizer(
            text,
            max_length=self.max_len,
            padding="max_length",
            truncation=True,
            return_tensors="pt"
        )

        return {
            'input_ids': encoding['input_ids'].squeeze(0),
            'attention_mask': encoding['attention_mask'].squeeze(0),
            'label': torch.tensor(label, dtype=torch.long)
        }

# Step 3: Tokenizer and Dataset Preparation
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
max_len = 128

train_dataset = StockPredictionDataset(X_train, y_train, tokenizer, max_len)
test_dataset = StockPredictionDataset(X_test, y_test, tokenizer, max_len)

train_loader = DataLoader(train_dataset, batch_size=16, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=16)

# Step 4: Model Initialization
model = BertForSequenceClassification.from_pretrained('bert-base-uncased', r
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)
```

```
# Step 5: Optimizer and Loss Function
optimizer = AdamW(model.parameters(), lr=2e-5)
criterion = nn.CrossEntropyLoss()

# Step 6: Training Loop
epochs = 3
for epoch in range(epochs):
    model.train()
    total_loss = 0

    for batch in train_loader:
        input_ids = batch['input_ids'].to(device)
        attention_mask = batch['attention_mask'].to(device)
        labels = batch['label'].to(device)

        optimizer.zero_grad()

        outputs = model(input_ids, attention_mask=attention_mask, labels=lab
        loss = outputs.loss
        logits = outputs.logits

        total_loss += loss.item()
        loss.backward()
        optimizer.step()

    avg_loss = total_loss / len(train_loader)
    print(f"Epoch {epoch + 1}/{epochs}, Loss: {avg_loss:.4f}")

# Step 7: Evaluation
model.eval()
y_pred = []
y_true = []

with torch.no_grad():
    for batch in test_loader:
        input_ids = batch['input_ids'].to(device)
        attention_mask = batch['attention_mask'].to(device)
        labels = batch['label'].to(device)

        outputs = model(input_ids, attention_mask=attention_mask)
        logits = outputs.logits
        predictions = torch.argmax(logits, dim=1)

        y_pred.extend(predictions.cpu().numpy())
        y_true.extend(labels.cpu().numpy())

# Evaluation Metrics
accuracy = accuracy_score(y_true, y_pred)
print(f"Accuracy: {accuracy}")
```

```
print("Classification Report:")
print(classification_report(y_true, y_pred))

# Save the trained model
model.save_pretrained("/Users/tiff1101/Downloads/bert_stock_classifier/")
tokenizer.save_pretrained("/Users/tiff1101/Downloads/bert_stock_classifier/"
```

```
Some weights of BertForSequenceClassification were not initialized from the
model checkpoint at bert-base-uncased and are newly initialized: ['classifie
r.bias', 'classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use
it for predictions and inference.
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/transformers/optimizat
ion.py:591: FutureWarning: This implementation of AdamW is deprecated and wi
ll be removed in a future version. Use the PyTorch implementation torch.opti
m.AdamW instead, or set `no_deprecation_warning=True` to disable this warnin
g
  warnings.warn(
Epoch 1/3, Loss: 0.5954
Epoch 2/3, Loss: 0.5366
Epoch 3/3, Loss: 0.5039
Accuracy: 0.7
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         3
           1       0.70      1.00      0.82         7

    accuracy                           0.70        10
   macro avg       0.35      0.50      0.41        10
weighted avg       0.49      0.70      0.58        10

/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

Out[16]:
```
('/Users/tiff1101/Downloads/bert_stock_classifier/tokenizer_config.json',
 '/Users/tiff1101/Downloads/bert_stock_classifier/special_tokens_map.json',
 '/Users/tiff1101/Downloads/bert_stock_classifier/vocab.txt',
 '/Users/tiff1101/Downloads/bert_stock_classifier/added_tokens.json')
```

In [17]:
```
#ensemble method/model stacking
import pandas as pd
import numpy as np
from sklearn.ensemble import StackingClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassif
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
import joblib

# Step 1: Data Preparation
# Assuming `articles_data` contains your articles and `binary_feature_list`
articles_df = pd.DataFrame(articles_data)

# Ensure binary_feature_list matches the number of articles
if len(binary_feature_list) != len(articles_df):
    raise ValueError("The length of binary_feature_list does not match the r

# Add the binary_feature_list to the DataFrame
articles_df['binary_trend'] = binary_feature_list

# Drop rows with missing values in required columns
articles_df.dropna(subset=['tfidf_features', 'bert_features', 'overall_senti

# Step 2: Feature Extraction
# Extract features from TF-IDF and BERT embeddings
tfidf_features = np.array(articles_df['tfidf_features'].tolist())
bert_features = np.array(articles_df['bert_features'].tolist())

# Add overall sentiment score as a feature
sentiment_scores = articles_df['overall_sentiment_score'].values.reshape(-1,

# Combine all features into a single feature matrix
X = np.hstack([tfidf_features, bert_features, sentiment_scores])

# Target variable
y = articles_df['binary_trend']

# Step 3: Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran

# Step 4: Define Base Models
# Define individual models to be used in stacking
base_models = [
```

```
    ('random_forest', RandomForestClassifier(n_estimators=100, random_state=
    ('gradient_boosting', GradientBoostingClassifier(n_estimators=100, randc
    ('svc', SVC(kernel='rbf', probability=True, random_state=42))
]

# Step 5: Define the Meta Model
# Logistic Regression as the meta-classifier
meta_model = LogisticRegression()

# Step 6: Create the Stacking Ensemble
stacking_model = StackingClassifier(
    estimators=base_models,
    final_estimator=meta_model,
    cv=5,  # 5-fold cross-validation for training the meta-model
    stack_method='predict_proba'  # Use probabilities from base models
)

# Step 7: Train the Stacking Model
stacking_model.fit(X_train, y_train)

# Step 8: Evaluate the Stacking Model
y_pred = stacking_model.predict(X_test)

# Accuracy and Classification Report
accuracy = accuracy_score(y_test, y_pred)
print(f"Stacking Ensemble Accuracy: {accuracy}")
print("Classification Report:")
print(classification_report(y_test, y_pred))

# Step 9: Save the Stacking Model
joblib.dump(stacking_model, "/Users/tiff1101/Downloads/trained_stacking_clas

# Step 10: Feature Importance (Optional)
# Extract feature importances from Random Forest (if applicable)
for name, model in stacking_model.named_estimators_.items():
    if hasattr(model, "feature_importances_"):
        print(f"Feature importances for {name}:")
        print(model.feature_importances_)

# Step 11: Load and Predict (Example)
# To load the model later and make predictions:
loaded_model = joblib.load("/Users/tiff1101/Downloads/trained_stacking_class
y_loaded_pred = loaded_model.predict(X_test)
print(f"Loaded Model Accuracy: {accuracy_score(y_test, y_loaded_pred)}")
```

```
Stacking Ensemble Accuracy: 0.7
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         3
           1       0.70      1.00      0.82         7

    accuracy                           0.70        10
   macro avg       0.35      0.50      0.41        10
weighted avg       0.49      0.70      0.58        10

Feature importances for random_forest:
[0.         0.         0.         ... 0.         0.         0.01426744]
Feature importances for gradient_boosting:
[2.49206264e-07 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
 0.00000000e+00 0.00000000e+00]
Loaded Model Accuracy: 0.7
```

```
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/opt/anaconda3/envs/myenv/lib/python3.9/site-packages/sklearn/metrics/_class
ification.py:1531: UndefinedMetricWarning: Precision is ill-defined and bein
g set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

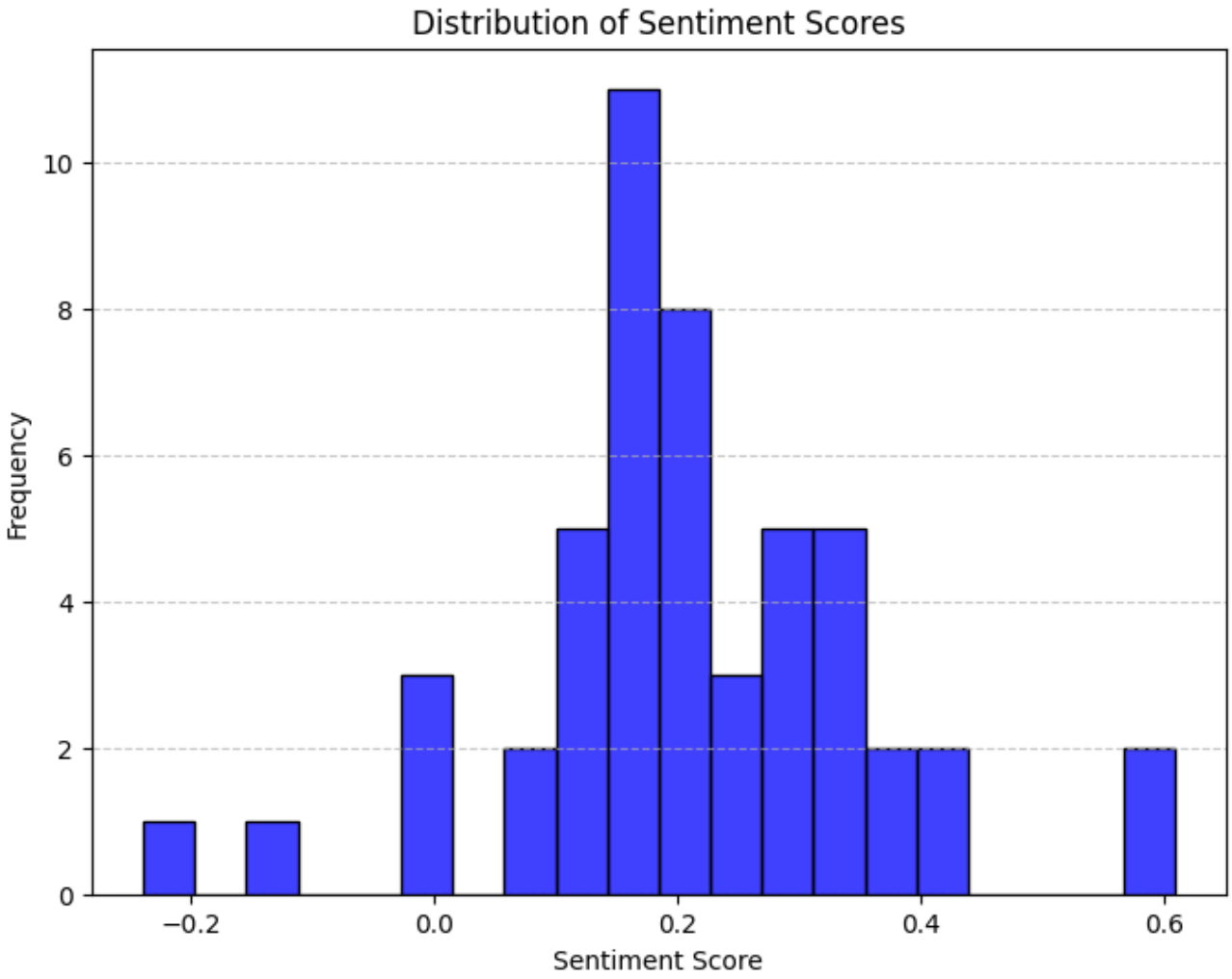
In [18]:

```
#Visualizations
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
import nltk
from nltk.corpus import stopwords
from nltk import FreqDist
from nltk.util import bigrams
import pandas as pd
import networkx as nx
from collections import Counter

articles_df = pd.DataFrame(articles_data)

# Download NLTK stopwords
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
```

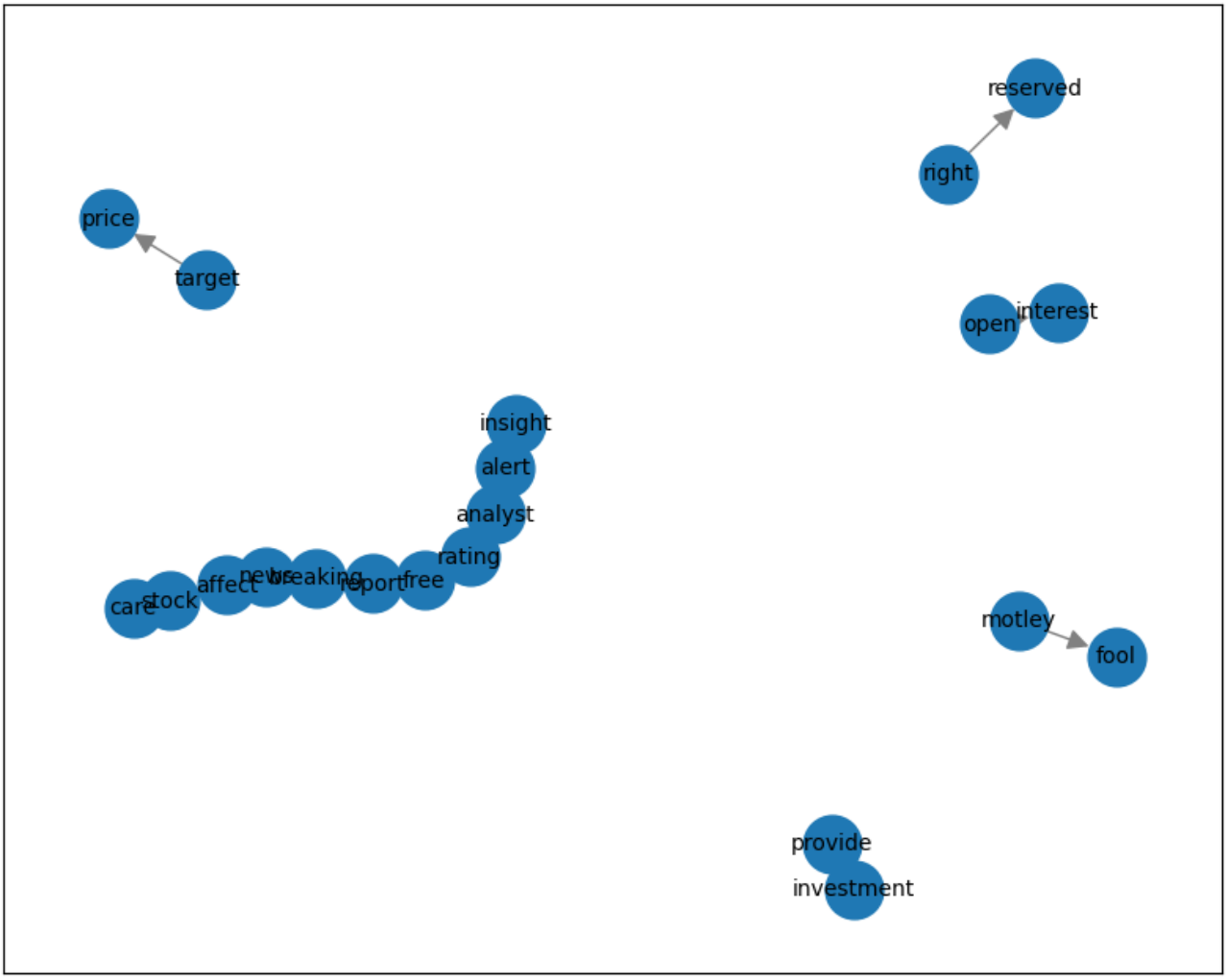
```
# Ensure you have a `cleaned_content` and `overall_sentiment_score` column i
if 'cleaned_content' not in articles_df.columns or 'overall_sentiment_score'
    raise ValueError("The DataFrame must contain 'cleaned_content' and 'over

# Step 1: Sentiment Bar Chart
def plot_sentiment_bar_chart(df):
    """a
    Plots a bar chart of sentiment scores.
    """
    plt.figure(figsize=(8, 6))
    sns.histplot(df['overall_sentiment_score'], bins=20, kde=False, color='b
    plt.title('Distribution of Sentiment Scores')
    plt.xlabel('Sentiment Score')
    plt.ylabel('Frequency')
    plt.grid(axis='y', linestyle='--', alpha=0.7)
    plt.show()

plot_sentiment_bar_chart(articles_df)


# Step 2: Directed Bigram Network Graph
def plot_directed_bigram_network(df, top_n=20, min_count=2):
    """
    Plots a directed bigram network from the cleaned text with control over

    Parameters:
        df (DataFrame): The dataframe containing the cleaned text data.
        top_n (int): The maximum number of bigrams to include in the network
        min_count (int): Minimum frequency of bigrams to include.
    """
    # Preprocess text and extract bigrams
    all_words = ' '.join(df['cleaned_content']).lower().split()
    filtered_words = [word for word in all_words if word not in stop_words a
    bigram_list = list(bigrams(filtered_words))

    # Count bigrams
    bigram_counts = Counter(bigram_list)

    # Select top N bigrams by frequency
    bigram_df = pd.DataFrame(bigram_counts.items(), columns=['bigram', 'cour
    bigram_df = bigram_df[bigram_df['count'] >= min_count].sort_values(by='c

    # Create a directed graph
    directed_bigram_graph = nx.DiGraph()
    for bigram, count in bigram_df.itertuples(index=False):  # Use itertuple
        word1, word2 = bigram  # Unpack the bigram tuple
        directed_bigram_graph.add_edge(word1, word2, weight=count)

    # Draw the directed graph
```

```
    plt.figure(figsize=(10, 8))
    pos = nx.spring_layout(directed_bigram_graph, k=0.5)
    nx.draw_networkx(
        directed_bigram_graph, pos, with_labels=True,
        node_size=700, font_size=10,
        edge_color='gray', arrows=True, arrowsize=20
    )
    plt.title(f"Directed Bigram Network Graph (Top {top_n} Bigrams)")
    plt.show()

# Example usage
plot_directed_bigram_network(articles_df, top_n=15, min_count=3)


# Step 3: Word Cloud
def plot_word_cloud(df):
    """
    Plots a word cloud from the cleaned text.
    """
    text = ' '.join(df['cleaned_content'])
    wordcloud = WordCloud(stopwords=stop_words, background_color='white', wi

    plt.figure(figsize=(10, 6))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title("Word Cloud of Article Content")
    plt.show()

plot_word_cloud(articles_df)
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/tiff1101/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```


Distribution of Sentiment Scores
Frequency
Sentiment Score
0
2
4
6
8
10
−0.2
0.0
0.2
0.4
0.6

Directed Bigram Network Graph (Top 15 Bigrams)



Word Cloud of Article Content