TRADING USING SENTIMENTAL ANALYSIS

DISCLAIMER:

This project is solely for research and education purposes, aimed at exploring alternate trading strategies and assessing their efficiency . It is explicitly not intended for investment advice .

TEAM MEMBERS

G.ABHINAV

P.ABHISHEK

G.SREEKER

P.VAMSHI

INTRODUCTION

Sentiment analysis, a branch of natural language processing, involves the use of computational techniques to determine the sentiment expressed in textual data. When applied to the realm of financial markets, sentiment analysis becomes a powerful tool for traders and investors seeking to gauge market sentiment towards specific assets. Trading using sentimental analysis leverages the insights gained from analyzing social media, news articles, and other textual data to make informed investment decisions. In this context, a project focused on trading using sentiment analysis utilizes a dataset that includes the number of likes and comments associated with specific stocks across various online platforms. By harnessing this information, the project aims to quantify market sentiment and predict potential price movements. The dataset's incorporation of social media interactions provides a unique dimension to the analysis, as it reflects public perception and sentiment towards particular stocks.

The project goes beyond traditional financial indicators by employing machine learning algorithms to process the sentiment data, calculating returns based on the observed sentiment trends. This approach enables investors to not only assess the sentiment-driven dynamics of specific stocks but also to compare the returns generated by sentiment-based trading strategies with those from other conventional forms of investment. Ultimately, such an endeavor contributes to a more comprehensive understanding of the role sentiment plays in financial markets and its impact on investment outcomes

PURPOSE

The primary objective of this project is to leverage Twitter impressions as a novel method for evaluating potential stock portfolios, aiming to generate profitable returns. The approach involves analyzing the impact of social media sentiment, particularly on Twitter, to inform investment decisions. The project goes beyond traditional financial metrics by incorporating the power of public opinion and discourse on social media platforms.

To assess the effectiveness of these investment strategies, the project conducts a comprehensive comparison of the returns obtained from the Twitter-informed portfolios with those of various other assets. These include but are not limited to Gold, Indices, individual Stocks, Exchange-Traded Funds (ETFs), and Commodities. By juxtaposing the performance of the Twitter-informed portfolios with these diverse asset classes, the project aims to evaluate the viability and competitiveness of this unconventional investment approach.

In summary, the purpose of this project is to develop and assess a unique investment strategy that leverages Twitter impressions, compares returns across various asset classes, and explores the synergy between social media sentiment and news-based information. The ultimate aim is to provide investors with a more comprehensive and informed perspective to guide their investment decisions.

SCOPE

Sentiment analysis in trading involves assessing market sentiment, often derived from sources like news articles, social media, and financial reports, to inform investment decisions. This practice has gained prominence due to its potential impact on market dynamics. Here's a brief overview of its key aspects:

1. Information Synthesis:

Sentiment analysis helps traders aggregate and interpret vast amounts of information from diverse sources, providing a quick overview of the prevailing sentiment in the market.

2. Market Response to News:

Financial markets are sensitive to news and events. Sentiment analysis aids in understanding how markets might react to specific news, allowing traders to respond promptly.

3. Algorithmic Trading:

Automated trading systems leverage sentiment analysis to make rapid decisions based on predefined criteria, enhancing the efficiency of trading strategies.

4. Risk Management:

By gauging sentiment, traders can adjust their positions or implement risk management strategies, reducing exposure to potential market fluctuations.

5. Contrarian Investing:

Sentiment analysis identifies situations where market sentiment is excessively positive or negative. Contrarian investors may leverage this information to anticipate reversals in market direction.

6. Event-driven Trading:

Sentiment analysis is crucial in event-driven trading, helping traders assess sentiment changes around specific events and adjust their positions accordingly.

7. Macro Trends and Indicators:

Sentiment analysis can identify broader market trends and sentiment indicators, assisting traders in making strategic, longer-term investment decisions.

8. Machine Learning Applications:

Advances in machine learning enable the development of sophisticated sentiment analysis models that can analyze large datasets and provide more accurate insights into market sentiment.

Project was done with a static dataset but powerful web scraping tools can be used for a dynamic dataset to get data from various sources like Google news, twitter, Moneycontrol etc..

We used only No. of Likes and No. of comments to determine which stocks to invest in but looking for specific words such as bull, bear, sell, buy, 52-week high, Golden crossover, etc... Can be used in order to determine stocks more accurately

Returns can be compared on a wider range of assets such as:

stocks, indices, ETFs, commodities, and more.

SOFTWARE REQUIREMENTS

The software requirements for a sentiment analysis project can vary based on the project's specific goals, technology stack, and implementation preferences. However, here is a general set of software requirements for a sentiment analysis project:

1. Programming Languages:

   - Choose a programming language suitable for implementing the sentiment analysis algorithms and models. Common choices include Python, Java, or R.

2. Development Frameworks:

   - Utilize machine learning and natural language processing frameworks such as TensorFlow, PyTorch, scikit-learn, or NLTK (Natural Language Toolkit) to facilitate the development and training of sentiment analysis models.

3. Web Development Framework (if applicable):

   - If the sentiment analysis project involves a web-based user interface, use a web development framework such as Django, Flask (for Python), or Node.js (JavaScript) for backend development. Frontend frameworks like React, Angular, or Vue.js may be used for the user interface.

4. Database Management System:

   - Choose a database management system (DBMS) to store and manage the data. Options include PostgreSQL, MySQL, MongoDB, or SQLite, depending on the nature of the data and project requirements.

5. Text Processing Libraries:

   - Utilize libraries for text processing and analysis, such as spaCy, NLTK, or TextBlob, to preprocess textual data and extract relevant features for sentiment analysis.

6. Machine Learning Libraries:

- Depending on the chosen machine learning approach, incorporate libraries like scikit-learn, TensorFlow, or PyTorch for building and training sentiment analysis models.

7. Version Control:

- Use version control systems like Git to track changes in the source code, collaborate with a team, and maintain a history of project modifications.

8. Documentation Tools:

- Employ documentation tools such as Sphinx or Jupyter Notebooks to document code, algorithms, and project details comprehensively.

These software requirements provide a foundation for building a sentiment analysis project, and the specific tools and technologies chosen may vary based on project constraints and objectives.

CODE

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
import datetime as dt
import yfinance as yf
import os

senti_data = pd.read_csv('sentiment_data.csv')
senti_data['date'] = pd.to_datetime(senti_data['date'])
senti_data = senti_data.set_index(['date', 'symbol'])
senti_data['engagement_ratio'] = senti_data['Comments']/senti_data['Likes']
senti_data = senti_data[(senti_data['Likes']>200)&(senti_data['Comments']>100)]
senti_data
grouped_df = (senti_data.reset_index('symbol').groupby([pd.Grouper(freq='M'),
'symbol'])
                [['engagement_ratio']].mean())
grouped_df['rank'] = (grouped_df.groupby(level=0)['engagement_ratio']
                     .transform(lambda x: x.rank(ascending=False)))
grouped_df

fil_df = grouped_df[grouped_df['rank']<6].copy()
fil_df = fil_df.reset_index(level=1)
fil_df.index = fil_df.index+pd.DateOffset(1)
fil_df = fil_df.reset_index().set_index(['date', 'symbol'])
fil_df.head(20)
dates = fil_df.index.get_level_values('date').unique().tolist()
fd = {}

for d in dates:
   fd[d.strftime('%Y-%m-%d')] = fil_df.xs(d, level=0).index.tolist()
fd
list_of_stocks = senti_data.index.get_level_values('symbol').unique().tolist()

prices_df = yf.download(tickers=list_of_stocks,
                    start='2021-01-01',
                    end='2023-03-01')

returns_df = np.log(prices_df['Adj Close']).diff().dropna()
```

```
new_df = pd.DataFrame()

for start_date in fd.keys():
    end_date =
(pd.to_datetime(start_date)+pd.offsets.MonthEnd()).strftime('%Y-%m-%d')

    cols = fd[start_date]
    t_df = returns_df[start_date:end_date][cols].mean(axis=1).to_frame('portfolio_return')

    new_df = pd.concat([new_df, t_df], axis=0)
new_df

qqq_df = yf.download(tickers='QQQ',
                    start='2021-01-01',
                    end='2023-03-01')

qqq_ret = np.log(qqq_df['Adj Close']).diff().to_frame('nasdaq_return')

new_df = new_df.merge(qqq_ret,
                      left_index=True,
                      right_index=True)

new_df

portfolios_cumulative_return = np.exp(np.log1p(new_df).cumsum()).sub(1)

portfolios_cumulative_return.plot(figsize=(16,6))

plt.title(' Return Over Time')

plt.gca().yaxis.set_major_formatter(mtick.PercentFormatter(1))

plt.ylabel('Return')

plt.show()
```

ANALYSIS REPORT OF DATASET

month

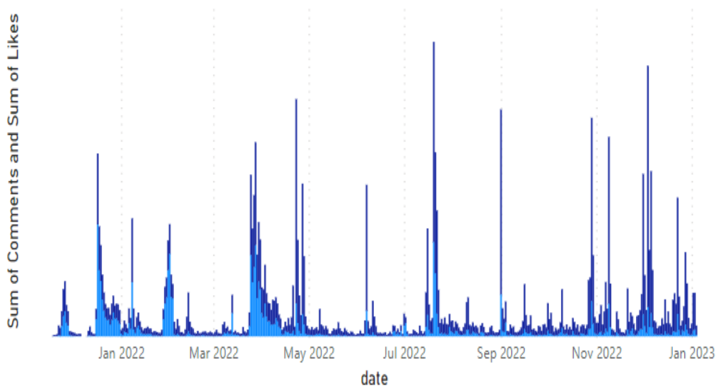
All

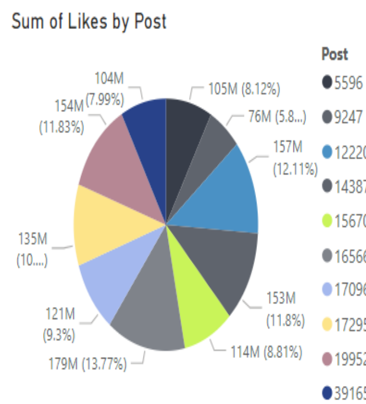
15M

Sum of Post

3bn

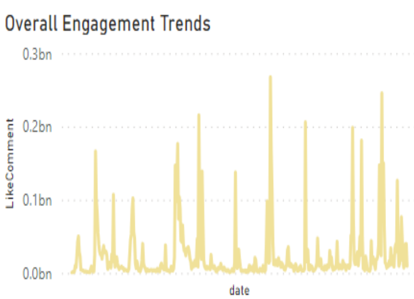
Sum of Comments

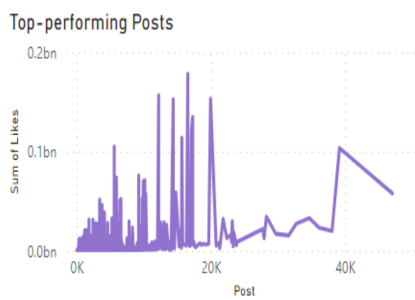




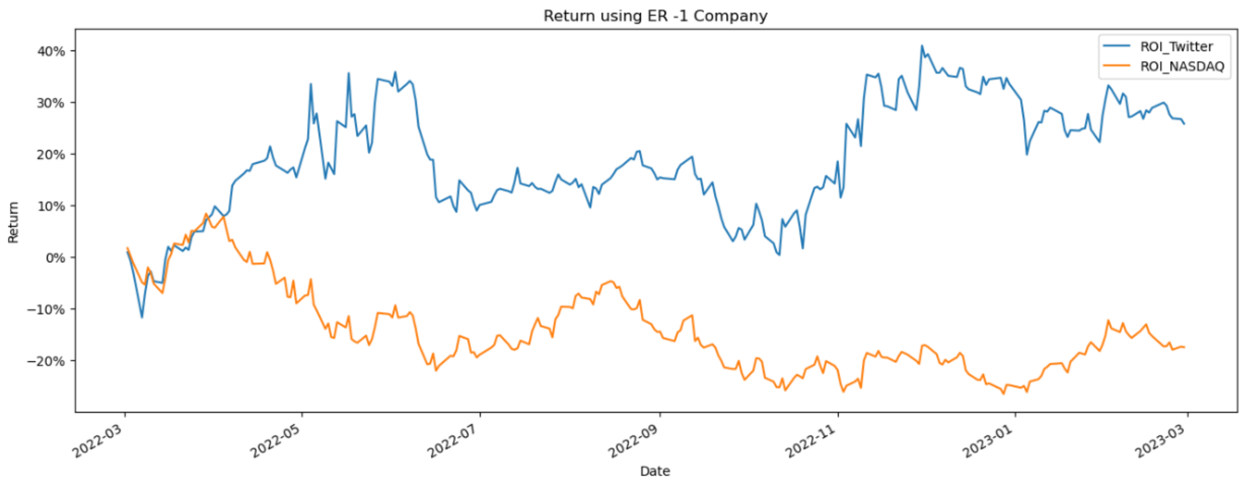
Post	Sum of Likes	Sum of Comments
1148	695	138
1037	1112	321
1725	1317	326
1658	2404	347
1063	2083	383
1369	2423	462
1927	2784	901
1246	4376	934
2539	5327	1007
1844	5808	1046
1847	4796	1061
1746	6333	1179
Total	**6623489199**	**2735360954**

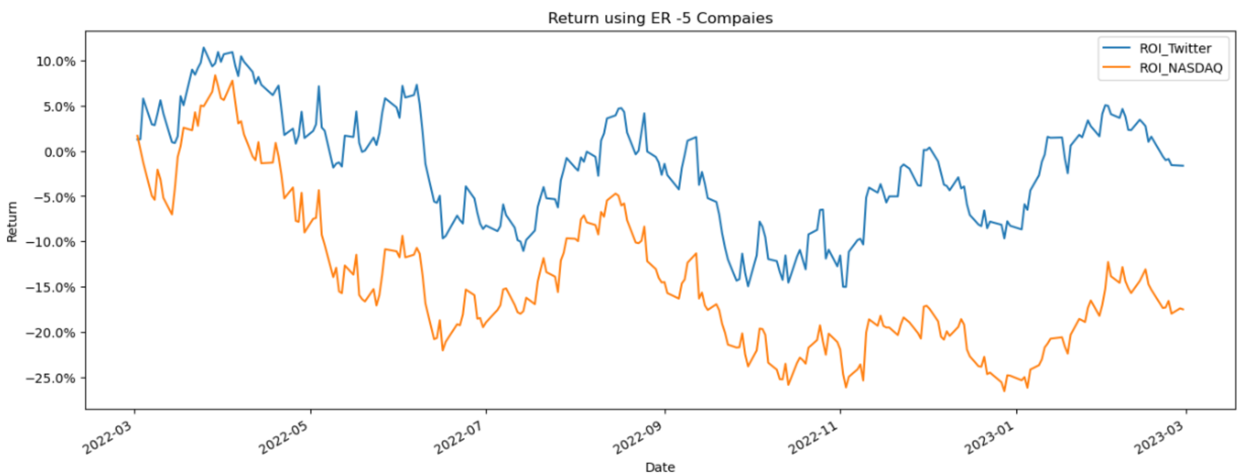
Filters

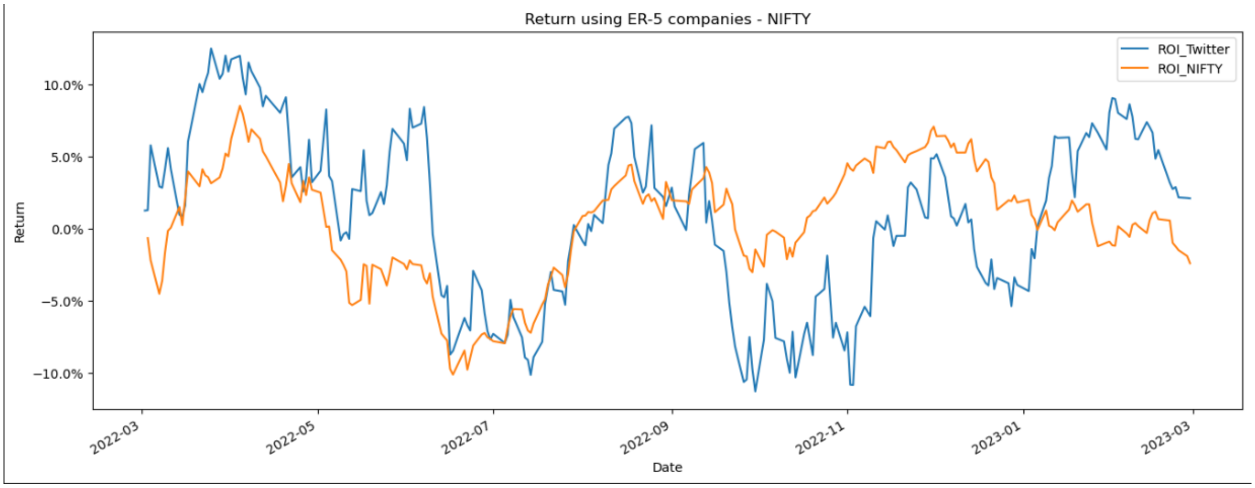


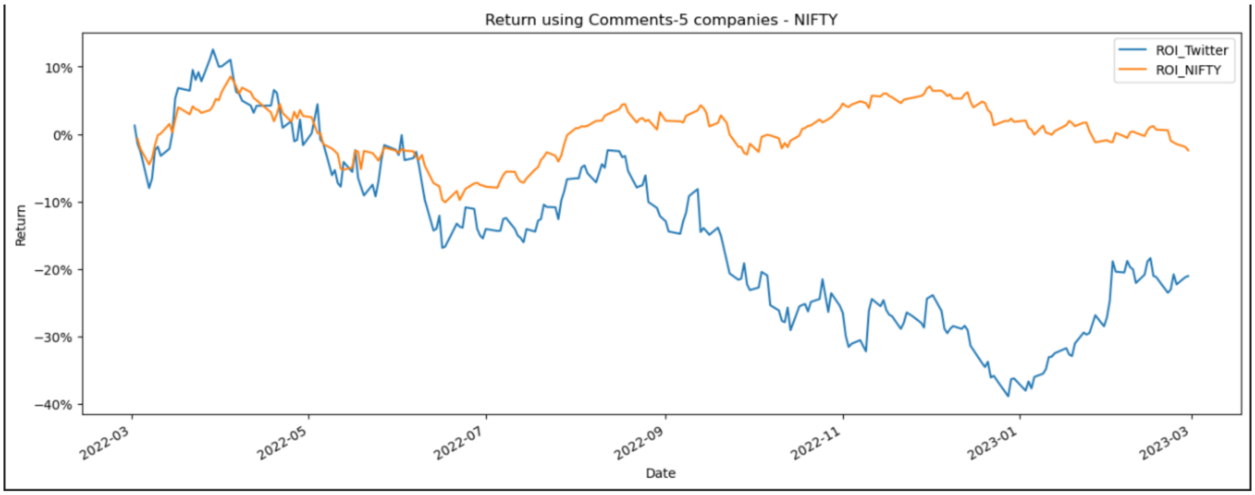


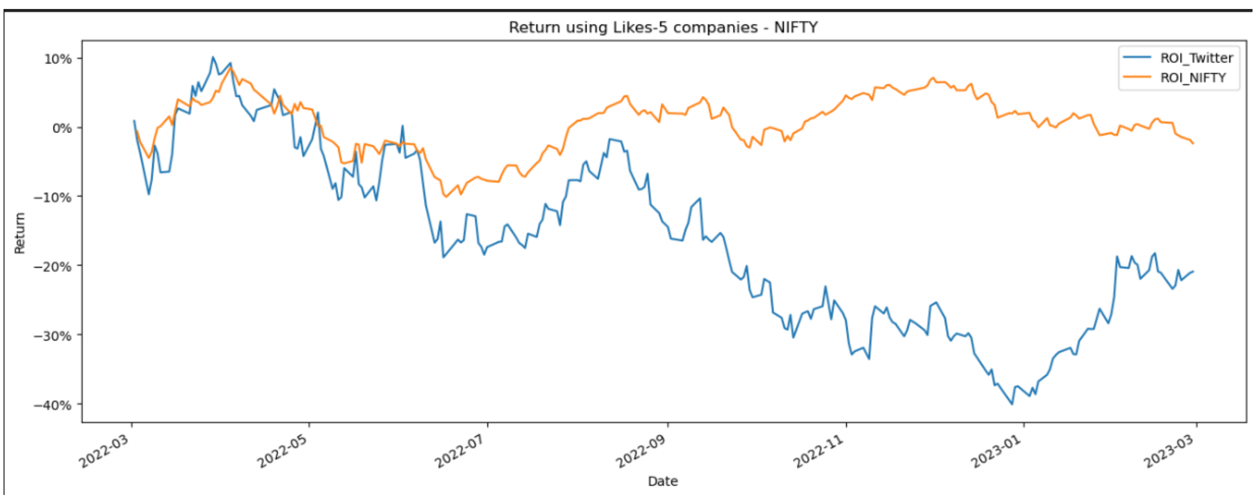
RESULTS

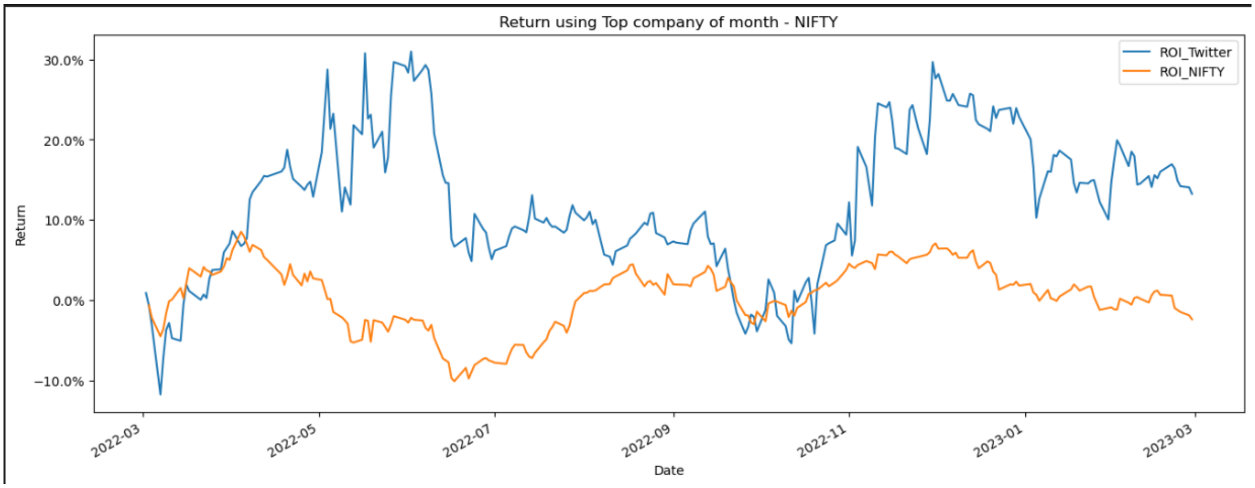

Return using ER -1 Company
ROI_Twitter
ROI_NASDAQ
40%
30%
20%
10%
0%
−10%
−20%
Return
2022-03
2022-05
2022-07
2022-09
2022-11
2023-01
2023-03
Date

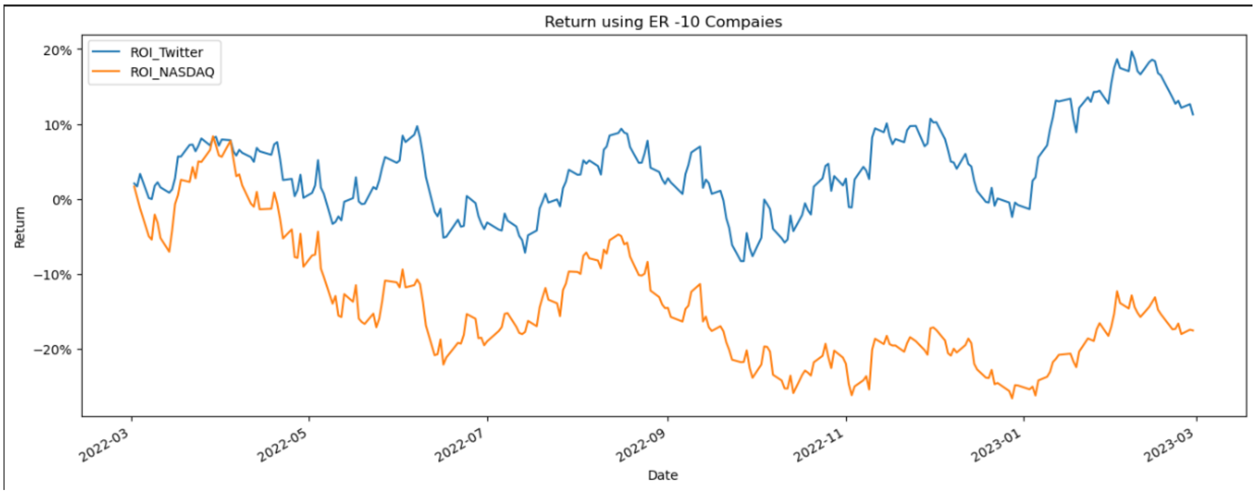

Return using ER -5 Compaies
ROI_Twitter
ROI_NASDAQ
10.0%
5.0%
0.0%
−5.0%
−10.0%
−15.0%
−20.0%
−25.0%
Return
2022-03
2022-05
2022-07
2022-09
2022-11
2023-01
2023-03
Date

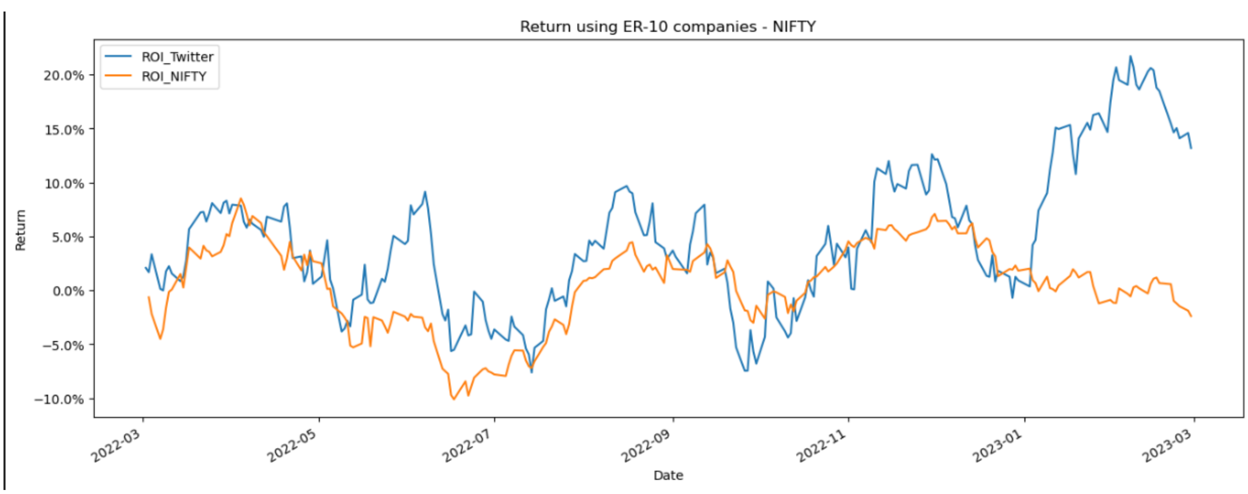

Return using ER-5 companies - NIFTY
ROI_Twitter
ROI_NIFTY
Return
10.0%
5.0%
0.0%
−5.0%
−10.0%
2022-03
2022-05
2022-07
2022-09
2022-11
2023-01
2023-03
Date

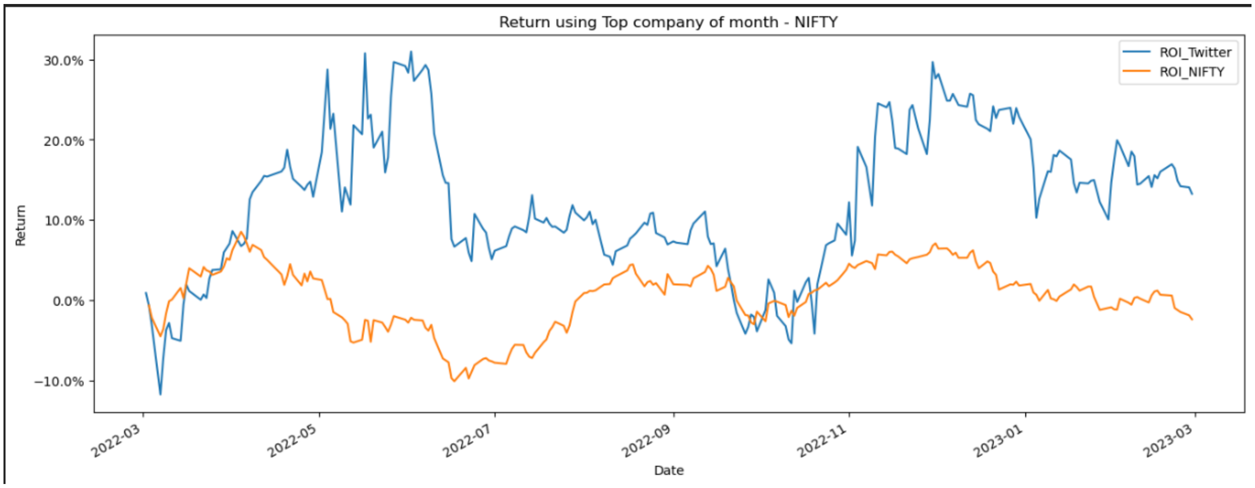

Return using Comments-5 companies - NIFTY
ROI_Twitter
ROI_NIFTY
Return
10%
0%
−10%
−20%
−30%
−40%
2022-03
2022-05
2022-07
2022-09
2022-11
2023-01
2023-03
Date

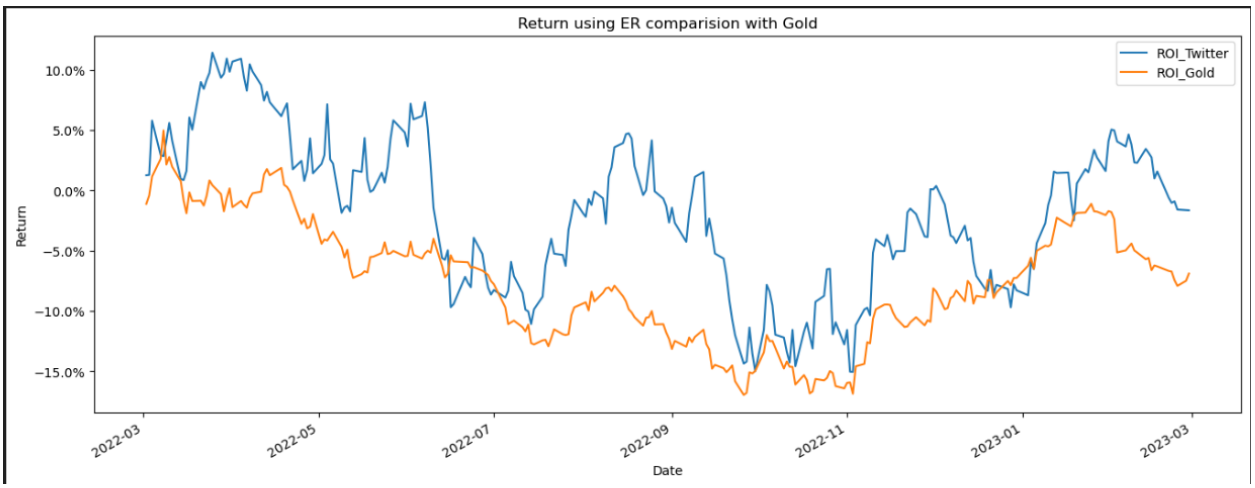

Return using Likes-5 companies - NIFTY
ROI_Twitter
ROI_NIFTY
Return
10%
0%
−10%
−20%
−30%
−40%
2022-03
2022-05
2022-07
2022-09
2022-11
2023-01
2023-03
Date


Return using Top company of month - NIFTY
ROI_Twitter
ROI_NIFTY
30.0%
20.0%
10.0%
0.0%
−10.0%
Return
2022-03
2022-05
2022-07
2022-09
2022-11
2023-01
2023-03
Date


Return using ER -10 Compaies
ROI_Twitter
ROI_NASDAQ
20%
10%
0%
−10%
−20%
Return
2022-03
2022-05
2022-07
2022-09
2022-11
2023-01
2023-03
Date


Return using ER-10 companies - NIFTY
ROI_Twitter
ROI_NIFTY
20.0%
15.0%
10.0%
5.0%
0.0%
−5.0%
−10.0%
Return
2022-03
2022-05
2022-07
2022-09
2022-11
2023-01
2023-03
Date


Return using Top company of month - NIFTY
ROI_Twitter
ROI_NIFTY
30.0%
20.0%
10.0%
0.0%
−10.0%
Return
2022-03
2022-05
2022-07
2022-09
2022-11
2023-01
2023-03
Date


Return using ER comparision with Gold
ROI_Twitter
ROI_Gold
10.0%
5.0%
0.0%
−5.0%
−10.0%
−15.0%
Return
2022-03
2022-05
2022-07
2022-09
2022-11
2023-01
2023-03
Date

INFERENCES FROM THE CHART ANALYSIS

- This method was giving better results as compared to investment in traditional methods such as gold.
- Although investing in the top company for a month gave better results. It is always recommended to diversify your portfolio in order to reduce risk and hence is not advisable .
- Investment using Engagement ratio always provided higher returns as compared to Likes or comments.
- Return using ER > Return using Likes > Return using Comments.

CONCLUSION

Sentiment Analysis plays a crucial role in deciphering market trends and investor behavior in the realm of trading and investments. By analyzing various reports, one can gauge the overall sentiment towards a particular asset, company, or market. Additionally, the correlation between social media metrics such as likes, comments, and engagement ratio can provide valuable insights into the broader sentiment landscape. The depth of engagement on these platforms often mirrors the intensity of investor interest and can serve as an important supplementary tool for decision-making in the dynamic world of financial markets. Integrating Sentiment Analysis with social media metrics offers a holistic approach to understanding market sentiment, empowering investors with additional tools to make informed decisions in the complex and rapidly changing landscape of trading and investments.