STOCK SENTIMENT ANALYSIS REPORT

Author: Kushagra Agrawal
Enrollment No: 23116051

OVERVIEW

What is Stock Sentiment Analysis?

Stock Sentiment Analysis involves analyzing the sentiments expressed in various sources such as news articles, social media, and financial reports to predict stock market movements. It leverages natural language processing (NLP) and machine learning to interpret and quantify the sentiment behind textual data, providing insights into investor behavior and market trends.

Purpose of the Report

This document outlines the methodology, analysis, and findings of a stock sentiment analysis project aimed at predicting stock price movements and aiding investment decisions.

Flow of Project:

1. Define the Scope:

- **Identify Target Stocks:** GOOG (Alphabet Inc.) and NVDA (Nvidia Corp.)
- **Data Sources:** New York Times - For scraping news articles and Yahoo Finance - For historical stock price data.

I focused on ***GOOG (Alphabet Inc.)*** stocks and ***NVDA (Nvidia Corp.)*** and used the *S&P 500* as a benchmark. Data was collected over the maximum period available using ***yfinance*** library and *Ticker* function. This is done to superset the period for the news headlines collected to prevent any errors while combining both the data.

2. Data Collection:

- **Sources**: Financial news websites – NYT.
- **Tools**: Python Requests Library and NYT API Token.

I used the ***NYT API Token*** to access the archives of the New York Times, one of the leading newspapers. Using the ***requests*** library in Python, I retrieved data from January 1, 2003, to December 31, 2023. Since the API has a restriction of 5 requests per minute, I implemented a *time.sleep(12)* function to stay within this limit. The retrieved data is written in a CSV file named *nyt_articles_io.csv*, containing HEADLINES and DATE columns. I also utilized the API's feature to filter specific sections of news that most affect the stock market on a day-to-day basis. For example, business, USA, Your Money and many more sections.

3. Data Preprocessing:

- **Cleaning**: Removal of HTML tags, punctuation, and stopwords.
- **Tokenization**: Splitting text into individual words or phrases.
- **Stemming & Lemmatization**: Reducing words to their base or root form.

In my project, the *preprocess_text* function prepares text for analysis. I start by removing punctuation using Python's re library. Then, *nltk* tokenization breaks the text into words, removing unnecessary words for clarity. Stemming with *PorterStemmer* reduces words to their basic forms, while lemmatization with *WordNetLemmatizer* converts words to their dictionary forms. This makes sure the text is consistent and helps improve accuracy in tasks like sentiment analysis and information retrieval in natural language processing (NLP).

4. Data Labeling:

- **Stock Price Movement**: Label data as 'increase', 'decrease', or 'no change' based on historical stock price movements.

There is very little chance that stocks prices will remain unchanged on trading days, hence we have used **1** to represent *increase* and **0** to represent *decrease*. If the closing of next day is greater than that day's closing, it would indicate 1 and vice-versa. We also have a *cumulative column* with determines the streak of increase or decrease which would be required in future to make predictions.

5. Feature Extraction and Sentiment Analysis:

- **Sentiment Scores**: Calculating sentiment scores using libraries such as *VADER*, *TextBlob* and *NLTK*.
- **Aggregate Scores:** Compute average stock prices over different time periods.

I have employed VADER, TextBlob, and NLTK libraries to calculate sentiment scores for news headlines. VADER is used for its efficiency in handling social media texts, TextBlob for its simplicity and ease of use, and NLTK for its comprehensive natural language processing tools. These stock prices are then aggregated over different time periods i.e. [2,5,60,250,500] days, providing an in-depth sentiment analysis helping in more accurate market predictions.

6. Model Training and Evaluation:

- **Models**: Linear Discriminant Analysis, Logistic Regression, Support Vector Machines (SVM), Random Forests, Neural Networks.
- **Metrics**: Accuracy, Precision, Recall, F1-score, ROC curve analysis.

Firstly, we defined the parameters that will be used for predicting the stocks and target i.e. Label (Up/Down). I have used the following 5 models to gain ground to find out the ones and zeroes we require. I have imported the models from *sklearn* library. We divided the initial data into 80:20 train and test and checked for the metrics - Accuracy, Precision, Recall, F1-score, ROC curve analysis, precision score and accuracy score.

7. Visualization and Correlation with Stock Prices

- **Correlation Plots:** Visualize the relationship between sentiment scores and stock prices.

8. Predictions

- **Cumulative Profit (in ratio to the starting price of stock):** Plot the net profit with respect to starting price.
- **Buy/Sell Points Plots:** Plot the buy and sell points on the stock price graph.
- **Period of Prediction:** 105 days

9. Analysis and Risk Calculations:

- **Sharpe Ratio**: Measures the risk-adjusted return of the portfolio.
- **Maximum Drawdowns**: Indicates the largest single drop from peak to trough in the portfolio value.
- **Number of Trades Executed**: Counts the number of buy/sell trades made.
- **Win Ratio**: The proportion of profitable trades to total trades.

Findings:

1. **The stock prices of GOOG and NVDA during their maximum period.**

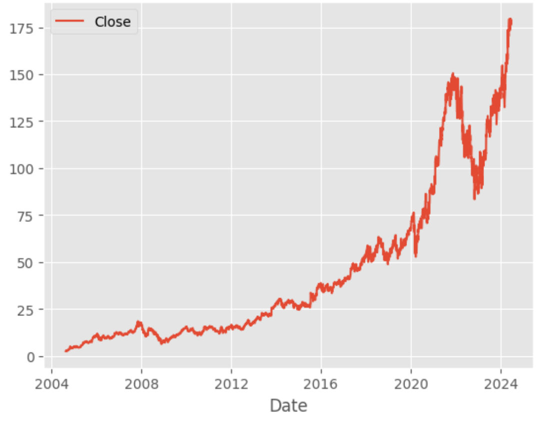

Figure 1 – GOOG

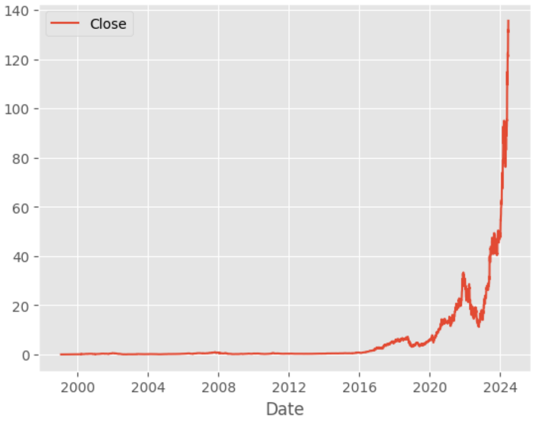

Figure 2 - NVDA

2. Correlation between Sentiment Scores and Stock prices changes from 2003 to 2004:

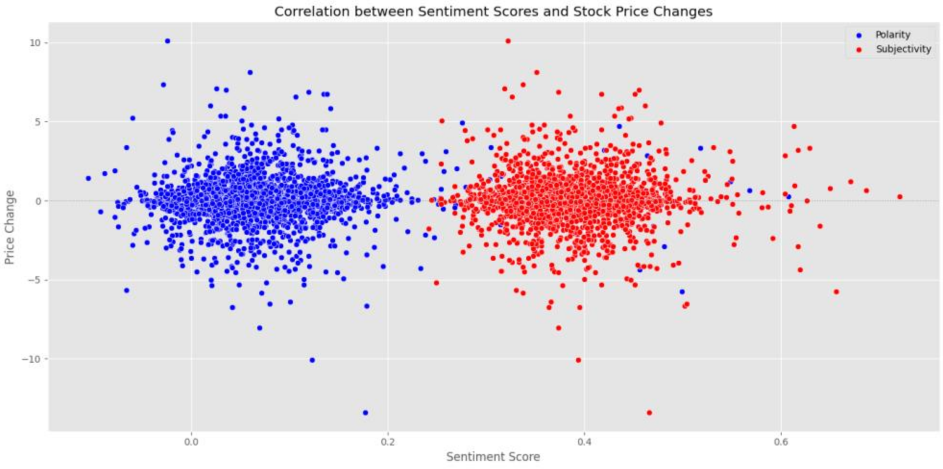

Figure 3 - GOOG

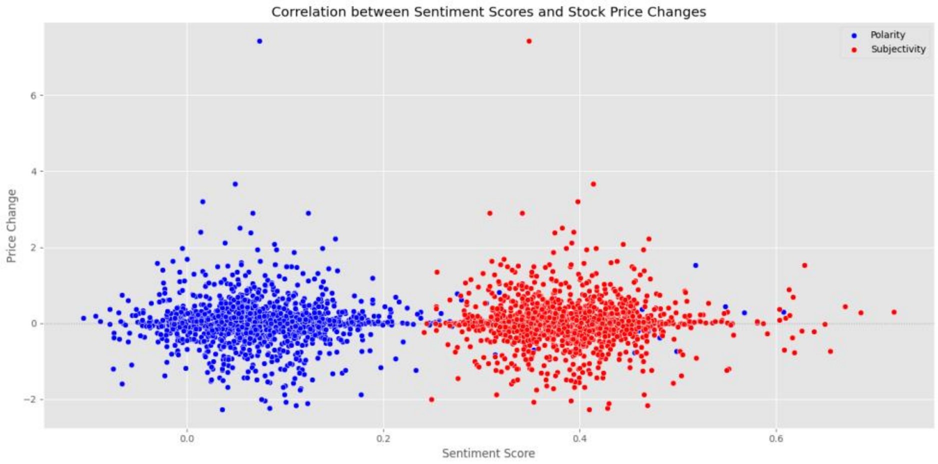

Figure 4 - NVDA

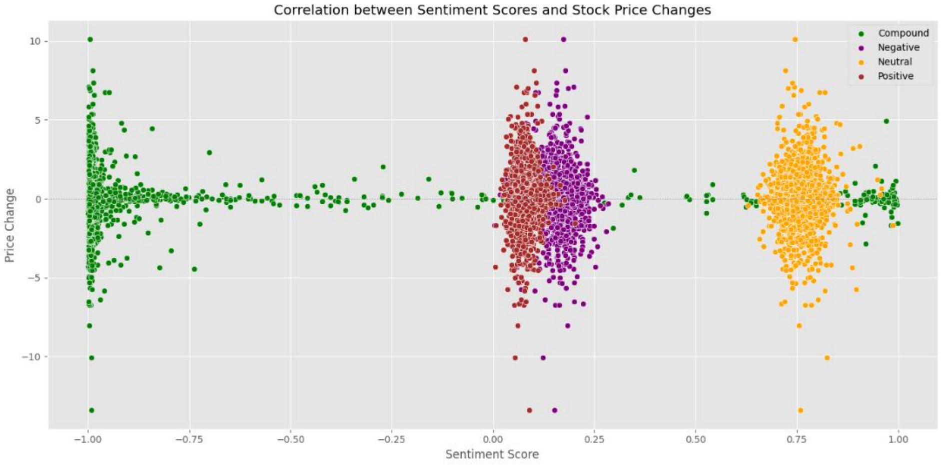

Figure 5 - GOOG

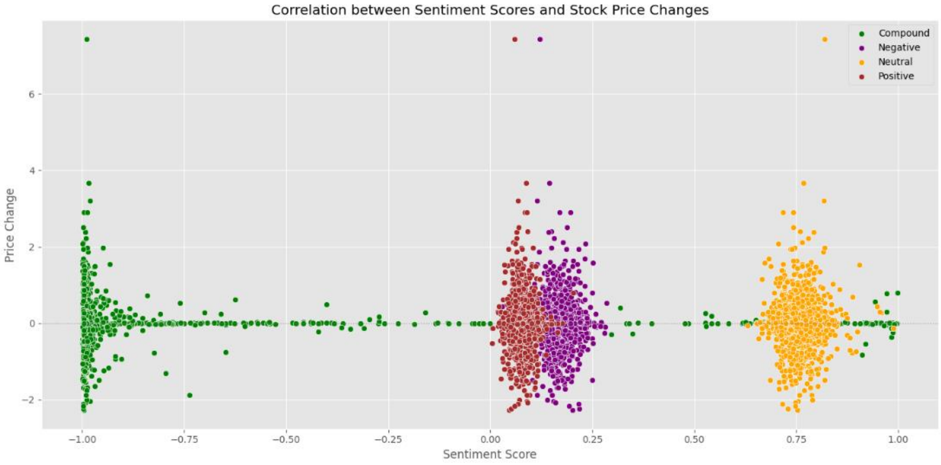

Figure 6 - NVDA

3. **Model Performance during Testing**:
   - Almost all models gave around 50% accuracy in predicting the zeroes and ones during training.

Model	Precision Score	
	GOOG	NVDA
Linear Regression	0.50	0.52
Logistic Regression	0.50	0.50
SVM	0.51	0.49
Random Forest	0.51	0.49
Neural Network	0.52	0.49
Average of All Models	0.51	0.50

4. **ROC Curve Analysis of all models:**

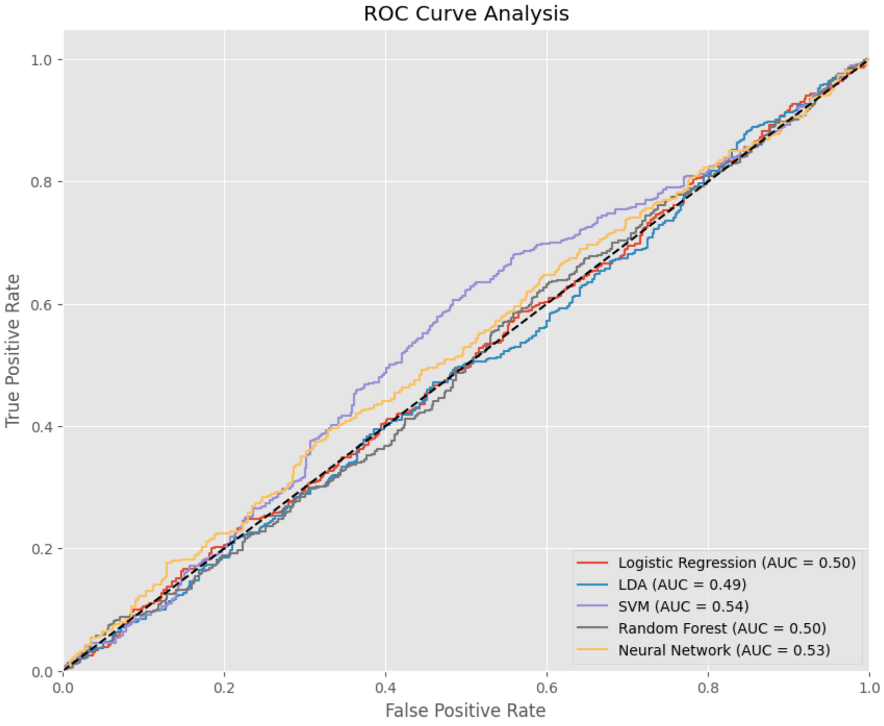

Figure 7 - GOOG

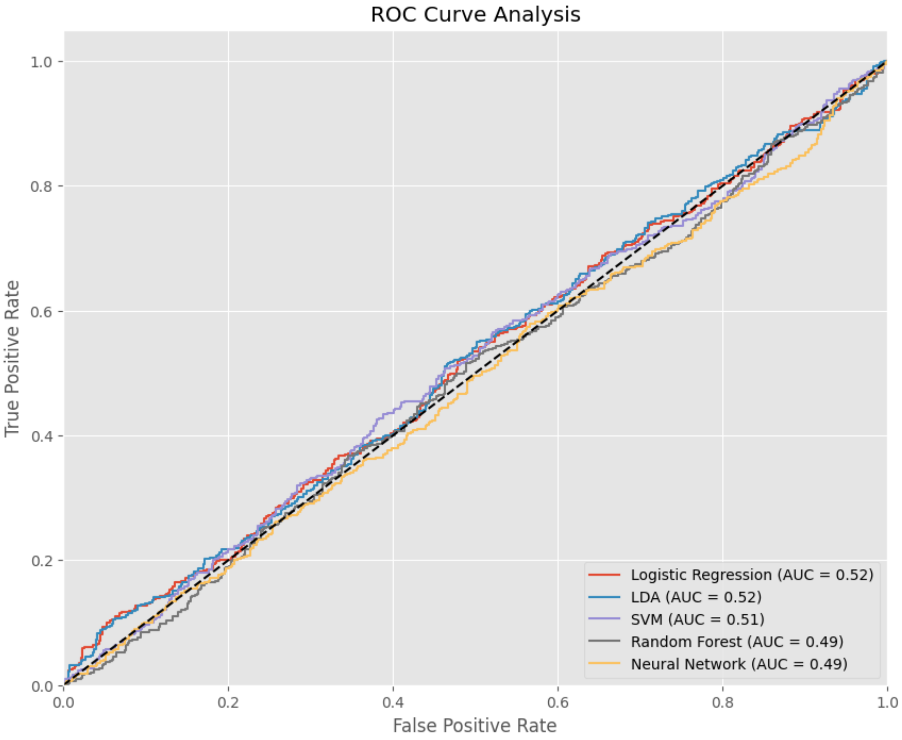

Figure 8 - NVDA

5. Model Performance during Prediction:

Model	Precision Score	
	GOOG	NVDA
Linear Regression	0.49	0.42
Logistic Regression	0.52	0.53
SVM	0.57	0.52
Random Forest	0.64	0.48
Neural Network	0.57	0.52
Weighted Average of All Models	0.60	0.51

6. Buy/Sell Points Plot:

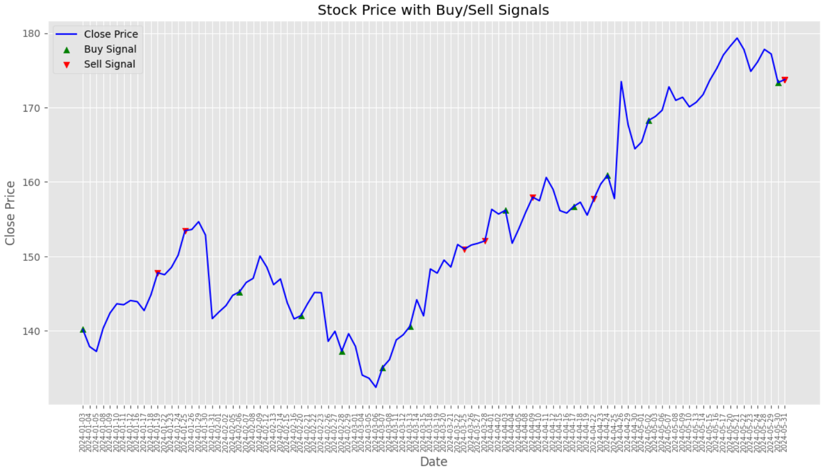

Figure 9 - GOOG

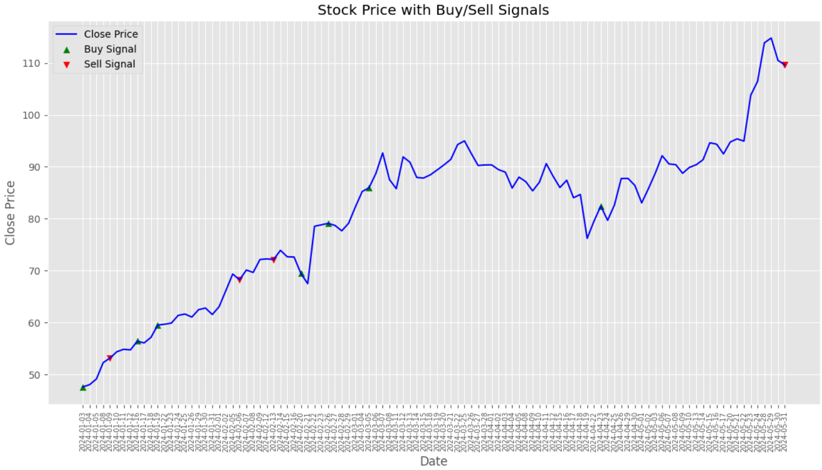

Figure 10 - NVDA

7. **Cumulative Profits Over Time** (In Ratio with the Starting Stock Value):

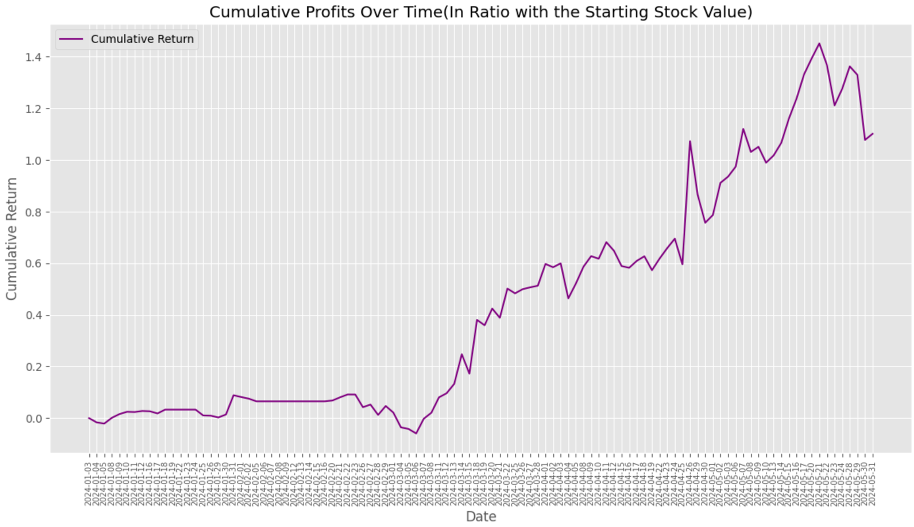

Figure 11 - GOOG

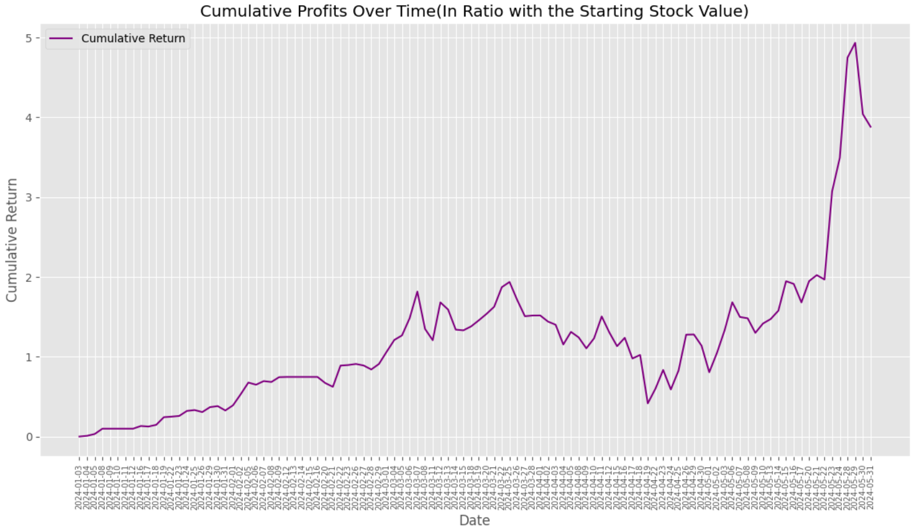

Figure 12 - NVDA

8. **Portfolio Performance**:

Deliverables	GOOG	NVDA
Sharpe Ratio	1.76	2.00
Maximum Drawdown	-15.26%	-51.80%
Number of Trades	11	7
Win Ratio	100.00%	100%
Portfolio Total Return	8.03%	31.57%
GOOG/NVDA Total Return	23.94%	130.48%
SP500 Total Return	11.27%	11.27%

9. **Trades**:

GOOG	Buy Date	Sell Date	Buy Price	Sell Price	Profit/ Loss	Return
1	2024-01-03	2024-01-19	140.200455	147.801804	7.601349	0.0542177
2	2024-02-06	2024-03-25	145.244720	150.978195	5.733475	0.0394746
3	2024-02-20	2024-03-25	142.038361	150.978195	8.939834	0.06294
4	2024-02-28	2024-03-28	137.273788	152.086929	14.81314	0.107909
5	2024-03-07	2024-04-09	135.086288	157.960251	22.87396	0.1693285
6	2024-03-13	2024-04-22	140.610001	157.770462	17.16046	0.122043
7	2024-04-03	2024-05-31	156.192261	173.762268	17.57001	0.1124896
8	2024-04-17	2024-05-31	156.701691	173.762268	17.06058	0.108873
9	2024-04-24	2024-05-31	160.916885	173.762268	12.84538	0.0798262
10	2024-05-02	2024-05-31	168.268524	173.762268	5.493744	0.0326487
11	2024-05-30	2024-05-31	173.362717	173.762268	0.399551	0.0023047

NVDA	Buy Date	Sell Date	Buy Price	Sell Price	Profit/ Loss	Return
1	2024-01-03	2024-01-19	47.56286	53.1331	5.570284	0.117114
2	2024-01-16	2024-02-06	56.3747	68.2142	11.83948	0.210014
3	2024-01-19	2024-02-13	59.4833	72.1187	12.63538	0.212419
4	2024-02-20	2024-05-31	69.443	109.624	40.18096	0.578618
5	2024-02-26	2024-05-31	79.0818	109.624	30.5422	0.38621021
6	2024-03-05	2024-05-31	85.95694	109.624	23.66706	0.27533624
7	2024-04-23	2024-05-31	82.41623	109.624	27.20777	0.33012639

Results:

- A Sharpe ratio of 1.76 of GOOG indicates that the portfolio has performed well on a risk-adjusted basis. Generally, a Sharpe ratio above 1 is considered good, above 2 is very good, and above 3 is excellent. Therefore, a ratio of 1.74 suggests that the portfolio has provided decent returns relative to its risk.
- NVDA has a Sharpe ratio of 2 slightly higher than GOOG, indicating even better risk-adjusted performance (very good result).
- The maximum drawdown of -15.26% of GOOG shows the largest peak-to-through decline in the portfolio's value during the analysis period. This indicates a significant drop at some point, suggesting moderate risk exposure.
- The maximum drawdown of NVDA is extremely high, indicating a substantial drop at some point. This suggests very high-risk exposure and potential volatility.
- A win ratio of 100.00% is very impressive, indicating that all trades made were profitable.

- The portfolio's total return of GOOG i.e. 8.03% is positive, indicating overall growth. However, when compared to the benchmarks, this return appears less impressive.
- NVDA, on the other hand with 31.57%, significantly outperformed the S&P 500 benchmarks. This impressive return highlights NVDA's strong performance.

Conclusion:

While both GOOG and NVDA exhibit good Sharpe ratios and impressive win ratios, indicating a high level of profitable trades with decent risk-adjusted returns, their total returns present a contrasting picture. Our GOOG's total return of 8.03% is relatively lower than both GOOG and the S&P 500 benchmarks, despite its favorable maximum drawdown of -15.26%, suggesting moderate risk exposure. In contrast, NVDA's total return of 31.57% significantly outperforms the S&P 500, though it comes with a much higher maximum drawdown of -51.60%, indicating substantial risk exposure. This analysis suggests that while GOOG has been managed in a way that minimizes losses and ensures consistent trade success, NVDA has delivered exceptional returns at the cost of higher volatility.

Drawbacks:

- **Data Quality**: The accuracy of the model depends on the quality and consistency of the textual data.
- **Ambiguity in Language**: Our sentiment analysis struggles with sarcasm, irony, and context-dependent meanings.
- **Market Changes**: The model requires regular updates and recalibration to adapt to changing market conditions and investor behavior.

Appendices

Appendix A: Sentiment Analysis Methodologies

- **NLP Techniques:** I have used 5 different NLP Techniques which are - Linear Discriminant Analysis, Logistic Regression, Support Vector Machines (SVM), Random Forests, Neural Networks. Due to less precision on each model, I decided to take the ones and zeroes as an average of all the models as it returned me with better scope for predictions than any other model i.e. 1s and 0s.
- **Sentiment Scoring Algorithms:** I have used TextBlob and VADER which give us Polarity, Subjectivity, Negative, Positive, Neutral and Compound Scores.
- **Buy/Sell Strategy:** My strategy has been to buy my stocks on those days when the stock has increased for at least two days and selling them when they decrease for two consecutive days. This strategy has worked pretty good on the stocks I applied on.

Appendix B: Predictive Model Details

- **Model Training Parameters:** Parameters and settings used for training predictive models are [*'Index', 'Open', 'Close', 'Volume', 'Subjectivity', 'Polarity', 'Compound', 'Negative', 'Neutral', Positive', 'Yesterday', 'RollingMean_2', 'RollingStd_2', 'Momentum_2', 'RollingMean_5', RollingStd_5', 'Momentum_5', 'RollingMean_60', 'RollingStd_60', 'Momentum_60', RollingMean_250', 'RollingStd_250', 'Momentum_250'*] whereas the target is *Label (Up/ Down)*.