Machine Learning

Project

Sentiment Analysis on Reddit

Authors:
Edouard Bueche 319539
Yassine Ben Said 385157
Pierre Porchet

Instructor:
Elise Gourier

Contents

Abstract

This study investigates whether a trading strategy can be developed based on sentiment analysis of Reddit comments. We extracted and cleaned data from different Reddit forums over the year 2021, analyzed the sentiment of comments, and examined the predictive power of this sentiment on future stock returns. We tried various Natural Language Processing (NLP) techniques to asses a score to a comment, we further implemented models to predict the future stock returns using this sentiment scores.[Our results indicate a poor relationship between Reddit sentiment (as we compute them) and stock returns leading us to question more the methodology rather than the idea itself.]

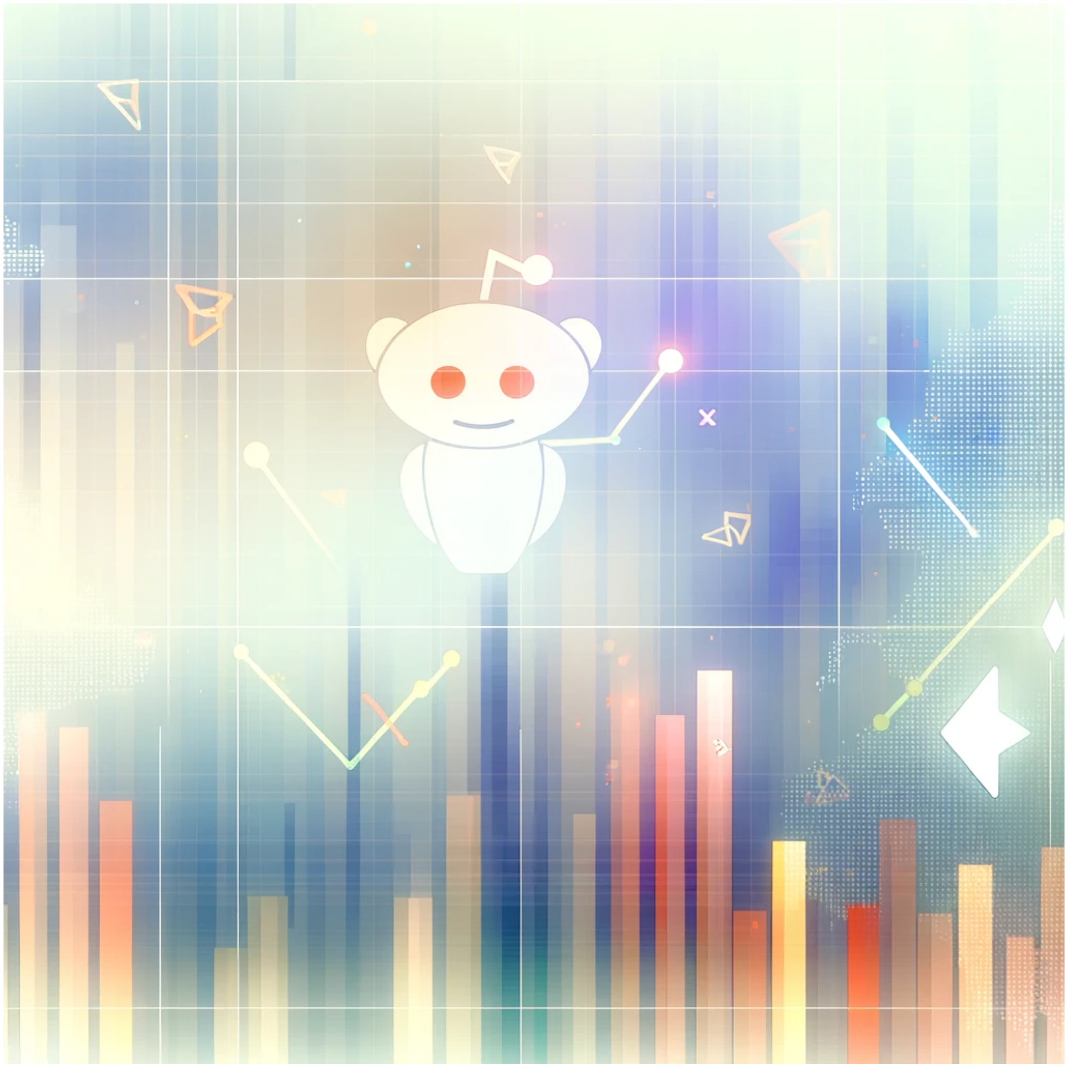

Figure 1: Abstract Illustration made by Dalle

1 Introduction

This project aims to explore the feasibility of developing a trading strategy based on sentiment analysis of comments from Reddit forums. The main goal is to determine whether sentiment extracted from Reddit can help us predict future stock returns. We will examine various NLP techniques to analyze sentiment and other ML methods to predict returns.

2 Literature Review

The literature on sentiment analysis in financial markets is extensive. Studies have shown that sentiment extracted from social media platforms like Twitter and Reddit can impact stock prices. For instance, Bollen et al. (2011) demonstrated that Twitter mood predicts the stock market. Similarly, studies have explored the predictive power of sentiment from financial news and its implications for trading strategies.

Sautner et al. (2023) developed a method to identify the attention paid by earnings call participants to firms' climate change exposures [7]. Their method captures exposures related to opportunity, physical, and regulatory shocks associated with climate change. They demonstrated that these measures are useful in predicting important real outcomes related to the net-zero transition, such as job creation in disruptive green technologies and green patenting. This study underscores the importance of understanding firm-level exposures to climate change and its implications for market behavior and resource allocation.

Vaswani et al. (2017) introduced the Transformer model [8], a network architecture based solely on attention mechanisms, which has proven superior in quality to previous models using recurrent or convolutional networks for sequence transduction tasks. The Transformer model significantly improved performance in machine translation tasks and has shown great potential in other NLP applications.

Jakubik et al. (2023) used Long Short-Term Memory (LSTM) networks to forecast Bitcoin prices by incorporating sentiment scores extracted from financial news [2] .
Their approach demonstrated the effectiveness of combining sentiment analysis with traditional financial data to improve forecasting accuracy.

We also gathered ideas for our project in other sources like the "When Is a Liability Not a Liability?" by Loughran and McDonald [4] or "Transformers: State-of-the-art Natural Language Processing" by Hugging Face [1] famous paper on transformers.

3 Data Extraction and Cleaning

We took our dataset from Kaggle [3], a popular data science platform. The dataset included posts and comments from 13 subreddits known for their discussions on stock market trading and investment strategies throughout 2021.

The selected subreddits are:

- `finance`, `stockmarket`, `forex`, `options`, `investing`, `stocks`
- `wallstreetbets`, `pennystocks`, `robinhood`, `gamestop`, `securityanalysis`
- `robinhoodpennystock`, `financialdependence`

Clean and Pre-Process

Due to the presence of missing data (NaNs), especially posts with only titles and no body text, we performed extensive data cleaning to ensure the quality of our dataset.
We also applied several preprocessing steps to clean and prepare the text data for analysis, including removing non-string entries and special characters, tokenizing and lemmatizing words, and removing stopwords and unwanted small words.

Bit of Context

The year 2021 was an unusual year for financial markets due to the ongoing COVID-19 pandemic and significant market events such as the GameStop short squeeze driven by discussions on the `wallstreetbets` subreddit. These unique circumstances resulted in a highly volatile market with intense discussions online, particularly regarding certain stocks.
We also knew that a lot of people online consider Elon Musk as sort of Messy, hence we expected a ”big” number of comments talking about Tesla.

Choice of Stocks to predict

The stocks we expected to find the most comments about were the following :
Tesla, S&P500, Apple, Amazon, Facebook, Microsoft, Google and Nvidia.

But how may we know if one comment we have is indeed talking about one of the stock of interest? Which are those comments that contain information, about the stock we are interested in, that we may use? The ”easy” solution was to label few hundred comments by hand and then try to run some algorithm to ”duplicate” the pattern he noticed were present into the manually-labelled data. We had an idea which would allow us to label our entire dataset in few hours...
We used the OpenAI API [6], that allows us to send request to different version of GPT through the python code, in order to use that we had to buy tokens that cost us not more than 30$ in total.

4 Data Labeling

We decided to use the GPT-3.5-turbo model to analyze the sentiment of comments, providing a more nuanced and context-aware sentiment score. The GPT model was chosen because it is capable of understanding and interpreting complex and context-specific language, which is crucial for accurately assessing sentiment in Reddit comments. We hope that a Language Model that large would make a good job selecting the data we will use later. This method allowed us to classify each comment into categories: Apple, Facebook, Microsoft, Amazon, Nvidia, Tesla, Google, S&P500, or None.

To achieve this, we designed a prompt that we send to GPT through the python code to ensure accurate and consistent classification. The system message used was:

> ”You are a text classification assistant. Classify each of the following texts into one of these categories: Apple, Facebook, Microsoft, Amazon, Nvidia, Tesla, Google, S&P 500, None. Return the classifications as a numbered list corresponding to the input texts, with each classification on a new line, in the following format: ’1. [Classification]’, ’2. [Classification]’, and so on. Give me nothing more than one of these categories for each text, please.”

This prompt ensures that the model focuses on categorizing each comment accurately into one

of the predefined categories, which is essential for our analysis of sentiment trends related to specific stocks. When we executed that code for the first time (a version where we sent 1 Reddit comment at a time), it was taking 60 hours just for one forum... in order to do make it last shorter we tried give it 25 texts at each request and use multi-thread to send several requests at the same time. At that time, we noticed that there were also a bound for the maximum number of request you may do per day...
With those few adaptations, we had been able to classify all the 13 forums in 3 days.

Summary Statistics and Labeled-Data Analysis

Year-Month	Amazon	Apple	Facebook	Google	Microsoft	Nvidia	S&P 500	Tesla
2021-01	43	60	15	8	8	33	358	337
2021-02	27	39	2	11	3	32	178	147
2021-03	10	29	3	13	6	24	165	92
2021-04	13	19	10	2	11	12	96	54
2021-05	15	8	3	3	0	31	86	44
2021-06	9	15	5	5	8	18	119	49
2021-07	11	11	9	12	9	25	105	43
2021-08	17	10	4	5	7	17	114	31
2021-09	7	25	12	10	9	14	114	19
2021-10	15	14	14	4	7	40	100	56
2021-11	9	10	6	5	11	18	90	54
2021-12	7	7	4	4	5	17	101	50

Table 1: Number of Each Label for Each Month

So we end up with more than 3500 labeled comments , out of the half a million comments from the beginning, it is a hit-rate of less than a pourcent.
Even if it's not a huge amount of data, it correspond to our expectations: a lot of unuseable info on the web...

Label	Amazon	Apple	Facebook	Google	Microsoft	Nvidia	S&P 500	Tesla
Count	183	247	107	112	84	281	1626	1056

Table 2: Number of Each Label Over the Year

The following graph displays the source of the comments, indicating the specific forum from which they are extracted, for each month.

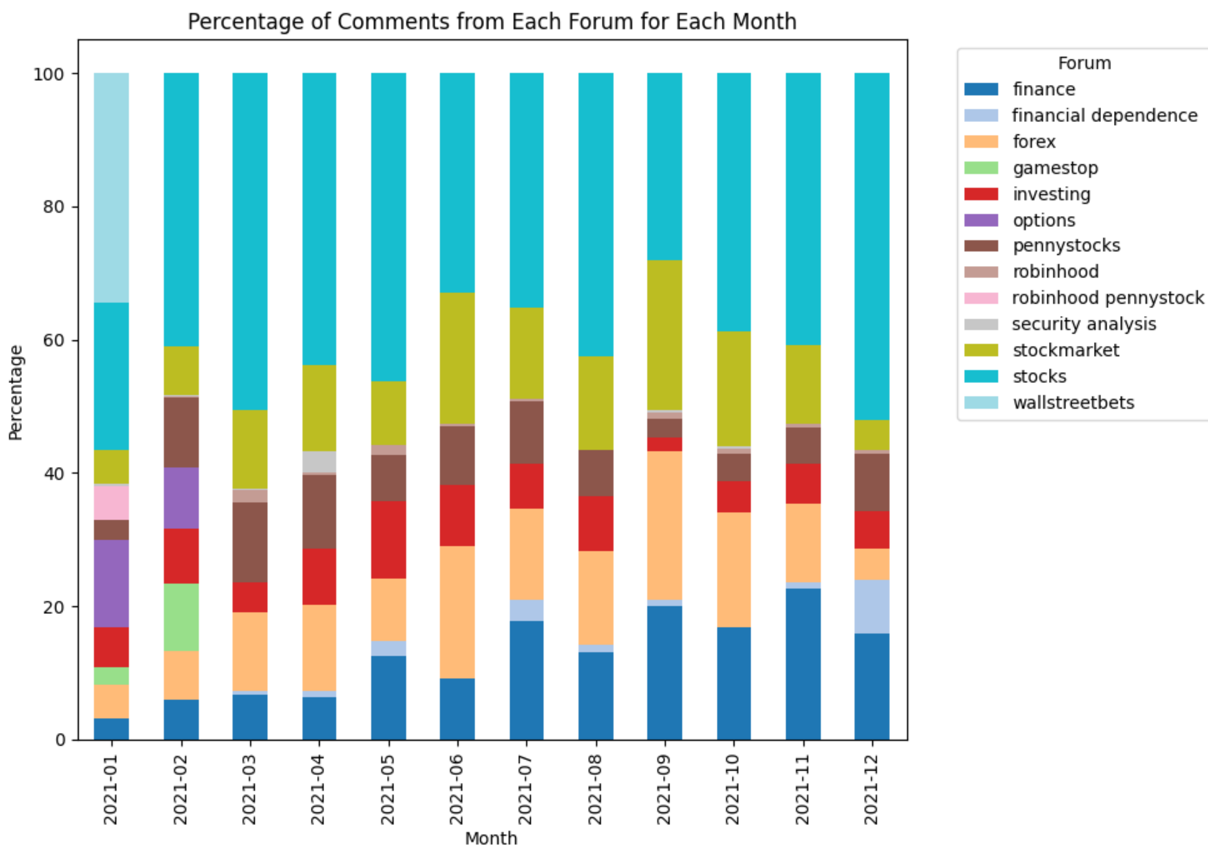

Figure 2: Caption describing the image.

the Figure 3 confirms exactly the doubts we had: there are a lot of comments on the S&P500 and also on Tesla, for the rest, there is clearly not enough data to try to predict anything.
In an academic paper, you might reformulate it as: When attempting to predict returns, we will limit our analysis to comments labeled with either S&P 500 or Tesla, to ensure the significance of our results."

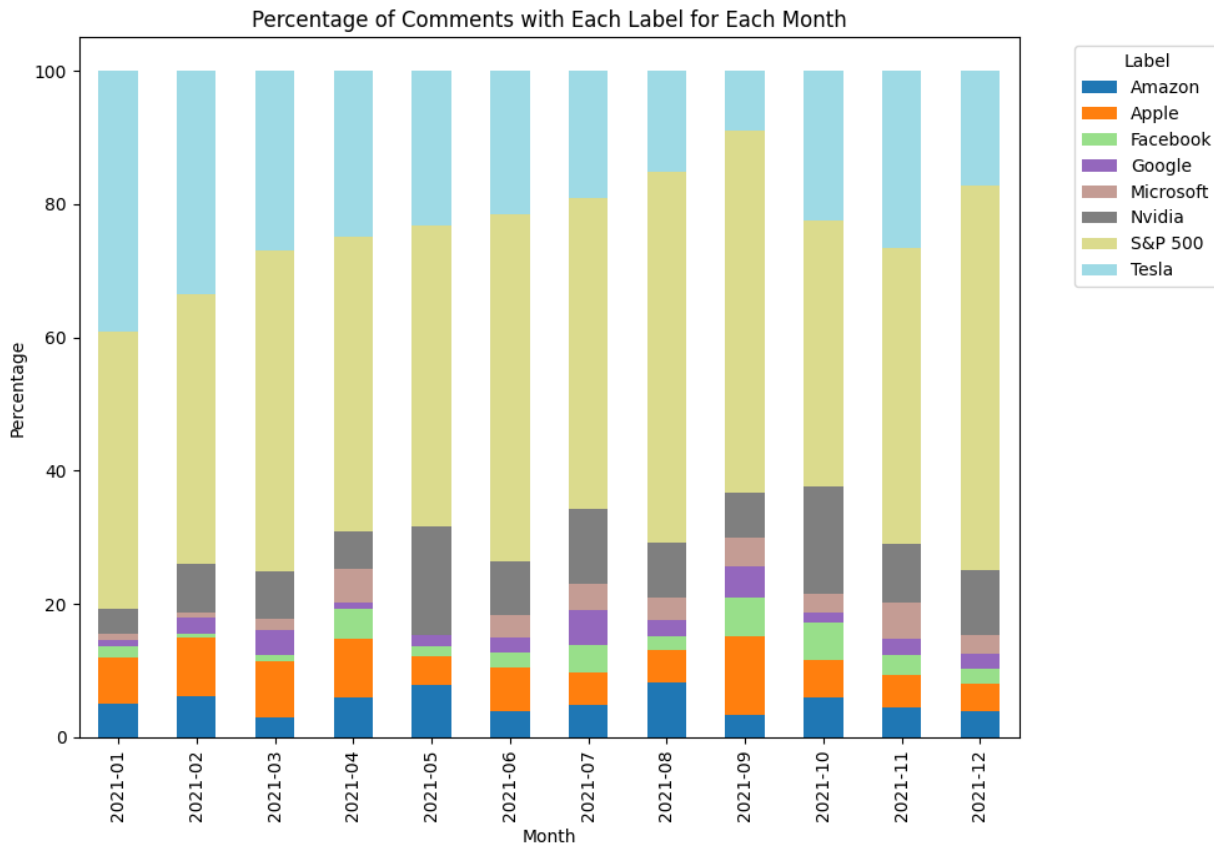

Figure 3: Caption describing the image.

Conclusion on Data Volume

Given the limited volume of labeled data (around 3500 comments) and the complexity of the language used in Reddit comments, we concluded that training a robust model on our own data would be challenging. The small dataset size would likely lead to overfitting and poor generalization to new, unseen comments.

5 Sentiment Score

Exploring Bag of Words and Dictionary Keyword Search

Initially, we attempted to use simple Bag of Words (BoW) and dictionary-based keyword search methods to analyze the sentiment and predict stock returns. These methods involved counting the frequency of specific words or phrases associated with positive or negative sentiment.

However, the results from these methods were poor. The BoW approach lacked the ability to capture the context and nuances of the comments. Dictionary keyword searches were also limited, as they failed to account for the varying ways sentiment can be expressed. The dictionnary we ended up with at the end contained the following words for example: ”Musk” , ”million”, ”Tech”, ”Industry” . . .

Given the complexities and nuances of natural language, especially in the context of social media, we decided to use pre-trained sentiment analysis tools to ensure comprehensive sentiment assessment. The tools we used are VADER, TextBlob, and GPT-3.5-turbo. Each of these tools was chosen for its strengths.

VADER Sentiment Analysis

VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool specifically attuned to sentiments expressed in social media (from Natural Language Tools Processing NLTP [5]). It is designed to handle the informal language often found in social media posts, making it suitable for our dataset of Reddit comments. VADER provides a compound score between -1 (very negative) and 1 (very positive), which captures the overall sentiment of the text. However it’s not possible to ask him the sentiment of the comment relativ to a stock for example, so it gives an overall sentiment, not necessarly link to the stock we are talking about, it is one weakness.

TextBlob Sentiment Analysis

TextBlob is a simple yet effective NLP tool that provides a sentiment polarity score between -1 and 1. It uses a combination of predefined rules and a lexicon-based approach to determine the sentiment of the text. TextBlob provides a complementary perspective to VADER as their structure are similars, enhancing the robustness of our sentiment analysis.

GPT-3.5-turbo Sentiment Analysis

We decided to use the GPT-3.5-turbo model for sentiment analysis because of its advanced capabilities in understanding and interpreting complex and context-specific language. Given the unique context of the 2021 data, including the COVID-19 pandemic and significant market events like the GameStop short squeeze, GPT-3.5-turbo’s ability to provide nuanced and context-aware sentiment scores was invaluable. This model was particularly chosen for its proficiency in capturing the subtleties of sentiment in Reddit comments.

The system message used for the GPT-3.5-turbo model was:

> ”You are a sentiment analysis assistant. Analyze the sentiment of the following text with respect to the given stock label and give a score between -1 and 1, where -1 is very negative and 1 is very positive with respect to the stock. Return the score in the following example format: ’Sentiment score: 0.24, if you feel ambivalent between negative and positive you can put 0.”

This prompt ensures that the model focuses on assessing each comment accurately one sentiment score related to specific stocks.

High and Low Score Differences

To further analyze the sentiment differences, we selected comments with high and low differences in sentiment scores across different sentiment analysis methods (between two of them), they are display below in table 3 and table 4

Title	Forum	Vader	Textblob	GPT
How to get Free $500 Amazon GiftCard	finance	0.6124	0.400000	-0.80
Facebook lost a bunch of talent in 2021	finance	0.1280	0.500000	-0.60
REASSURANCE	pennystocks	0.7902	-1.000000	0.75
The S&P 500 jumps above 4000 points for the first time.	stocks	0.8964	0.002083	0.75

Table 3: High score differences between our three models: Vader, Textblob and GPT

Title	Forum	Vader	Textblob	GPT
Microsoft Surpasses Apple as Most Valuable Company	finance	0.5256	0.500000	0.90
Tesla Semi electric truck is finally about to go into production	stocks	0.0314	0.032500	0.62
S&P 500 drop last month	stocks	0.0900	0.083333	0.10

Table 4: Low score differences between our three models: Vader, Textblob and GPT

As one can see, our different implemented models assign the wrong sentiment scores to the different labeled comments, and they are also inconsistent with each other. For example, in table 3, the comment ”How to get Free $500 Amazon GiftCard” is rated as 0.6124 with Vader and -0.80 with GPT, altough the comment is neutral and should have value 0. The last comment, ”The S&P 500 jumps above 4000 points for the first time”, if the score is attributed to SP500, it should have a high score. However, there is a value of nearly 0 for textblob.

For the low score differences, the same phenomena appears.

Daily Mean Sentiment Calculation

To capture the overall sentiment trend for each stock, we calculated the daily mean sentiment scores using VADER, TextBlob, and GPT-3. This approach helps to smooth out daily fluctuations and provides a clearer picture of the sentiment trends.

The idea behind using the daily average sentiment for a stock is the following. When many singers sing together, individual voices might be slightly off-key, but the collective sound may harmonize beautifully. Similarly, while each comment have its biases or errors, computing the mean of all sentiment score of the day might result in a more accurate and reliable overall sentiment score.

Correlation Analysis with Stock Returns

We examined the correlation between the sentiment scores and the daily stock returns to evaluate the predictive power of the sentiment analysis. The correlation heat-map, as shown in

Figure 6, illustrates the strength and direction of the relationships between different sentiment scores and stock returns.

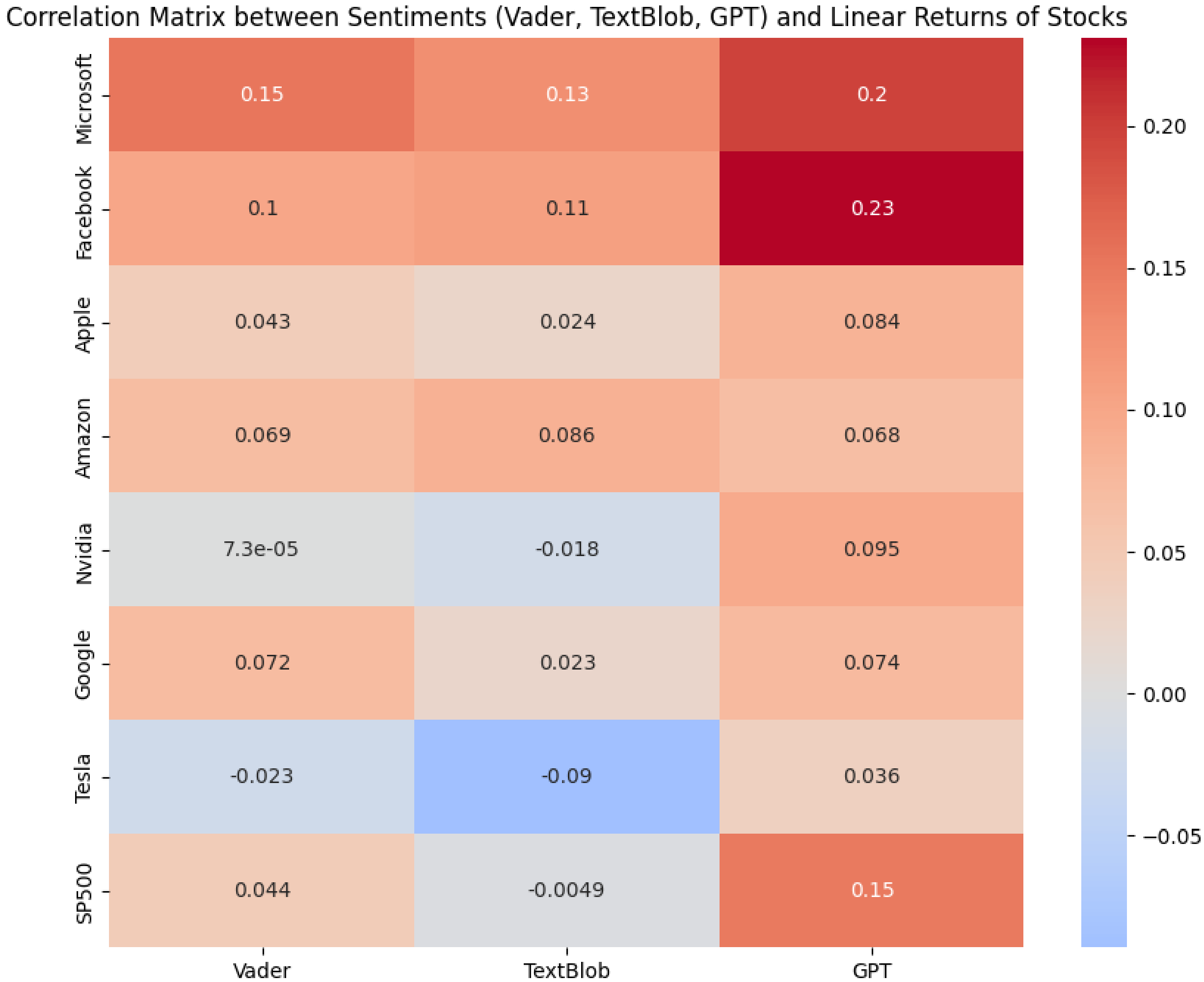

Figure 4: Heatmap of correlation coefficients between sentiment scores and stock returns.

What we can see is that the best score of 0.23 correlation is for Facebook, hence not significant because too few comments on it, it's quite the same for Microsoft. Contrarily, the score of 0.15 for GPT sentiment analysis on S&P500 is a bit more encouraging because there are a lot more data.

6 Linear Regressions

In this section, we shift our focus to modeling, beginning with linear regression. The model is formalized as follows:

$$R_{t+1} = \mu + \sum_{i=0}^{P} \lambda_{t-i} \cdot R_{t-i} + \sum_{i=0}^{Q} \beta_{t-i} \cdot \text{Score}_{t-i} + \epsilon \tag{1}$$

In this model, t represents a day, R denotes the returns, and Score corresponds to the sentiment score. The coefficients μ, λ, and β are components of the linear regression model, while P and Q are considered hyperparameters.. Initially, we employed ordinary least squares regression, progressing to ridge and lasso regression methods, and ultimately adopting the elastic net approach due to its superior performance. This approach introduces two additional hyperparameters: the $l1$ ratio and α for the elastic net.

The model was trained using data from the first nine months of 2021 and tested on the last quarter of the year. Through K-fold cross-validation incorporating sentiment scores from GPT, VADER, and TextBlob, we determined that the optimal values for P and Q are both 1. This indicates that the model primarily utilizes data from the preceding day, suggesting that earlier days' data do not significantly influence the prediction.

The figures below compare the performance of different sentiment scoring methods against the actual returns in our dataset for the last three months of 2021.

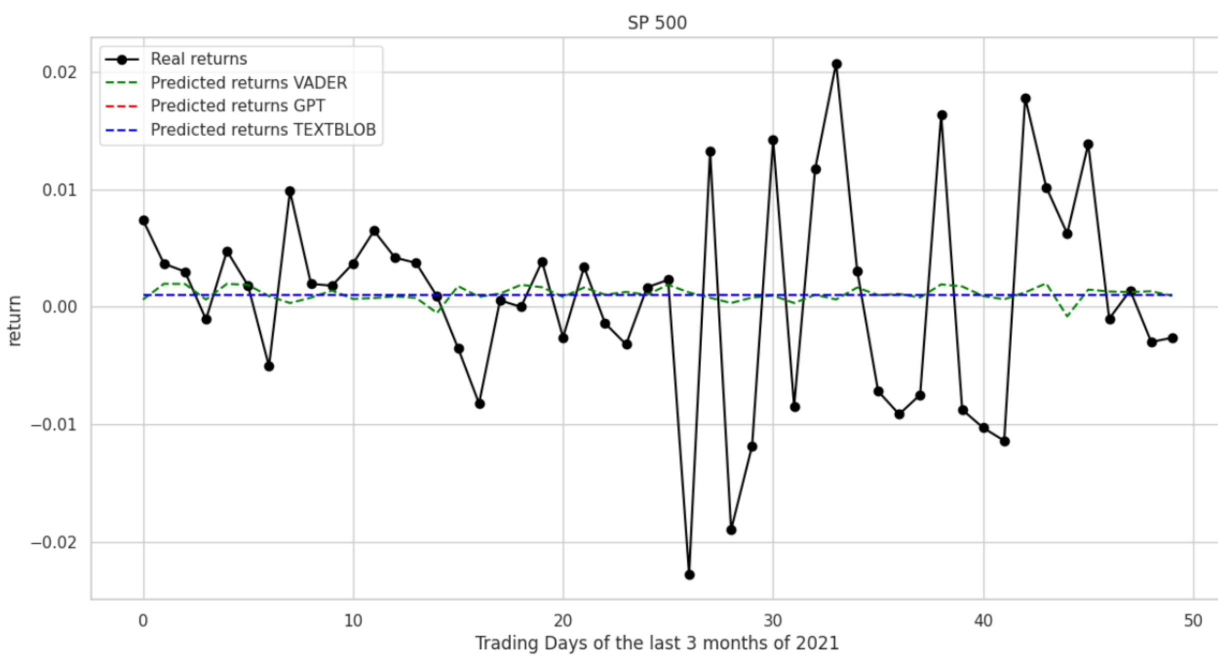

Figure 5: Returns for SP500

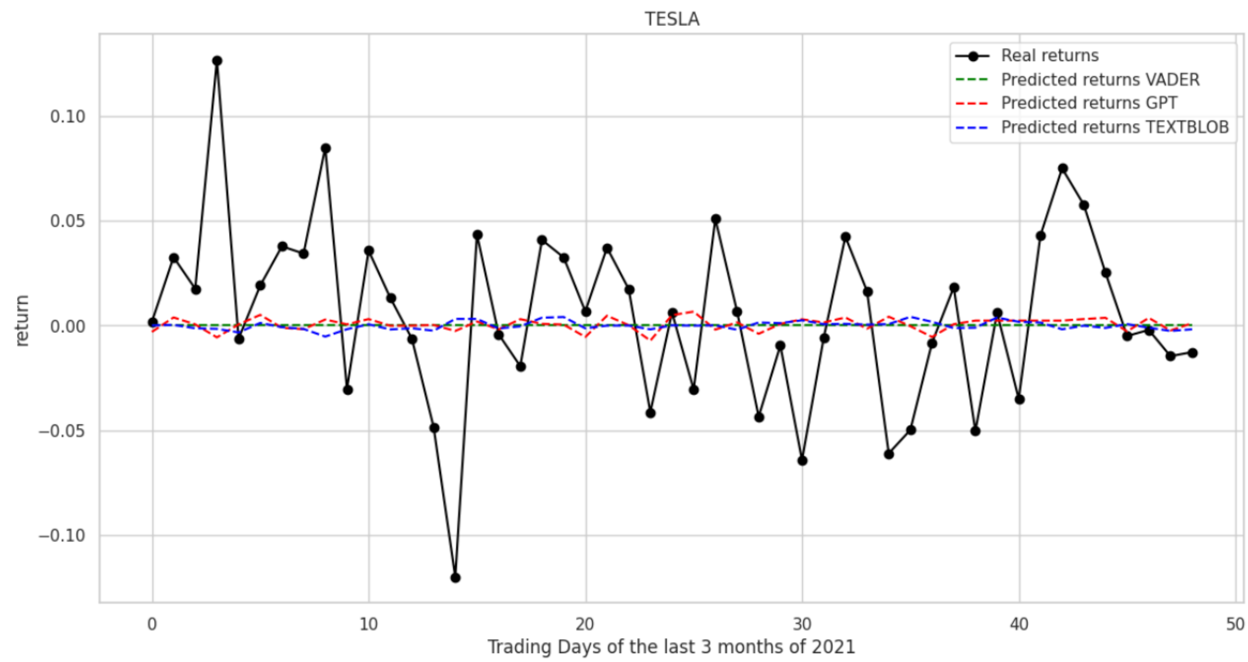

Figure 6: Returns for TESLA

In our initial analysis, the predicted returns are observed to be close to zero, which at first glance suggests limited informational value from the graph alone. However, a deeper examination into whether the predicted returns align in direction with the actual returns reveals a more nuanced insight. Across all models, the directional accuracy consistently exceeds 50% for all the models for both SP500 and TESLA (Detailed information is displayed in the table below). This outcome indicates that, while not exceptionally precise, the model offers a probability greater than one in two of correctly predicting the direction of returns. Such performance suggests a moderately effective model, although one that could benefit from enhanced risk management strategies to improve reliability and predictive accuracy.

Method	Match Rate (%)
VADER	54.0
TEXTBLOB	58.0
GPT	58.0

Table 5: Comparison of Predicted Returns Direction Matching with Real Returns for SP500

Method	Match Rate (%)
VADER	55.102
TEXTBLOB	53.612
GPT	57.143

Table 6: Comparison of Predicted Returns Direction Matching with Real Returns for TESLA

7 Machine Learning Models

In this section, we describe the machine learning models used to predict stock returns based on the sentiment analysis scores obtained from Reddit comments. The chosen models are Random Forest, Gradient Boosting, and Long Short-Term Memory (LSTM) networks. These models were selected based on their performance in related studies and their ability to handle the complexities of financial data and sentiment analysis. For each algorithm, we plot only one graph for either Tesla or S&P500 and the rest is in the appendix.

Gradient Boosting for S&P 500

Sentiment Model	MSE	Directional Accuracy
GPT Sentiment	0.0001	50.82%
VADER Sentiment	0.0001	49.18%
TextBlob Sentiment	0.0001	57.38%

Table 7: Evaluation Metrics for Gradient Boosting on S&P 500

The results from the Gradient Boosting model show a relatively close performance among the different sentiment analysis methods, with TextBlob achieving the highest directional accuracy at 57.38%.

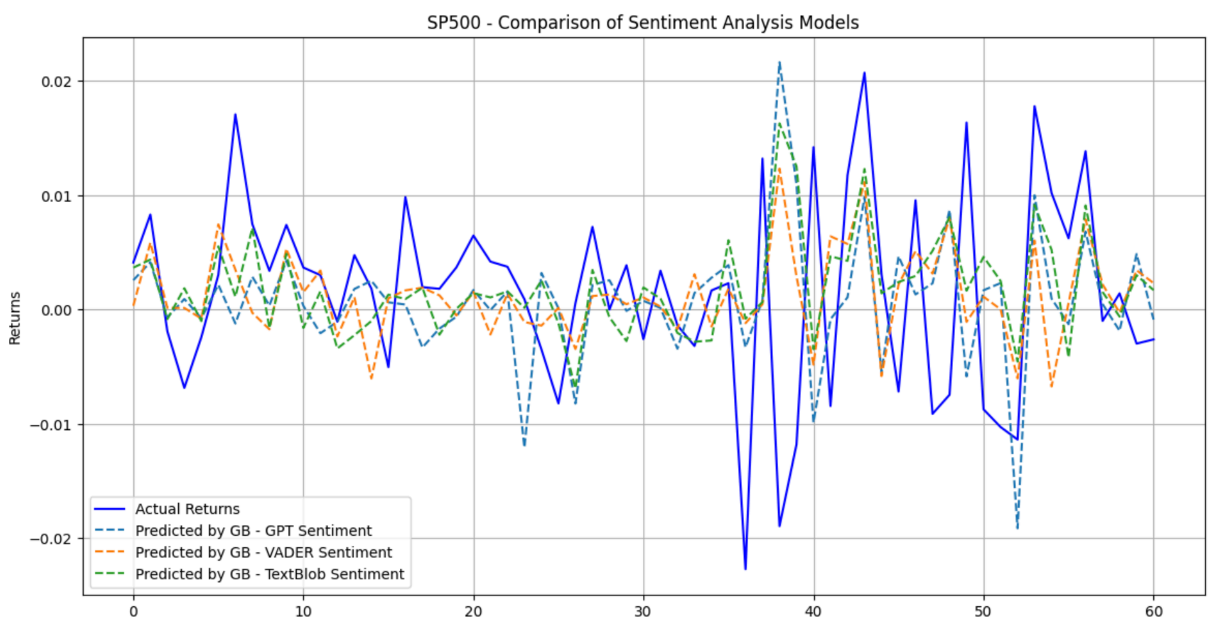

Figure 7: Gradient Boosting, test dataset on S&P500

Long Short-Term Memory (LSTM) for Tesla

For LSTM, the number of epochs was set to 50, aiming to sufficiently train the model without causing overfitting. The architecture included 50 neurons in the LSTM layer, which provided a good balance between model complexity and performance.

Sentiment Model	MSE	Directional Accuracy
GPT Sentiment	0.0017	52.46%
VADER Sentiment	0.0016	57.38%
TextBlob Sentiment	0.0015	57.38%

Table 8: Evaluation Metrics for LSTM on Tesla

LSTM models showed a good performance on Tesla data, with directional accuracies noticeably higher than those of Gradient Boosting for the S&P 500. The close MSE values indicate that all sentiment analysis methods are similarly effective at capturing the quantitative aspect of market sentiment.

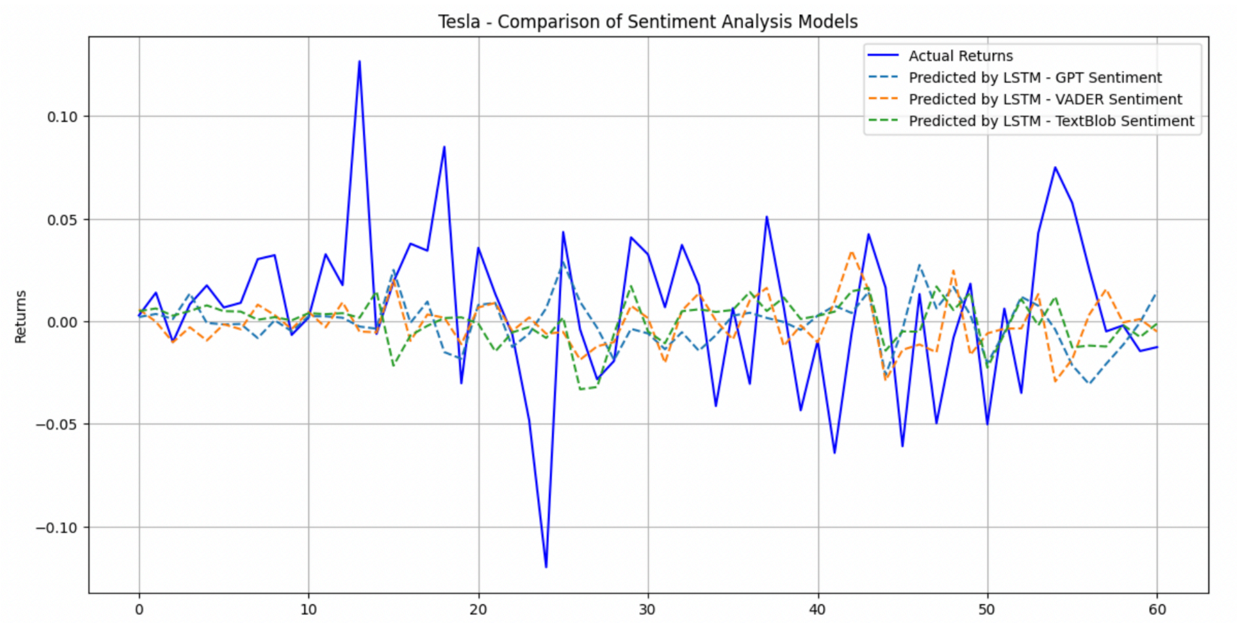

Figure 8: LSTM, test dataset on Tesla

Random Forest for Both Tesla and S&P 500

The number of trees in the Random Forest and the number of boosting stages in Gradient Boosting were both set to 100. This parameter was chosen based on preliminary tests that balanced computational efficiency with predictive performance. The depth of each tree and the learning rate in Gradient Boosting were adjusted to prevent overfitting while still capturing sufficient complexity in the data.

Random Forest models provided a mix of results. For Tesla, GPT sentiment analysis led with the highest directional accuracy at 67.21%, significantly outperforming VADER and TextBlob. This could be due to GPT's deep learning-based nature, which may capture complex patterns in data more effectively.

Stock/Index	Sentiment Model	MSE	Directional Accuracy
Tesla	GPT Sentiment	0.0001	67.21%
Tesla	VADER Sentiment	0.0001	54.10%
Tesla	TextBlob Sentiment	0.0001	62.30%
S&P 500	GPT Sentiment	0.0016	52.46%
S&P 500	VADER Sentiment	0.0017	45.90%
S&P 500	TextBlob Sentiment	0.0016	55.74%

Table 9: Comparison of MSE and Directional Accuracy using Random Forest on Tesla and S&P 500

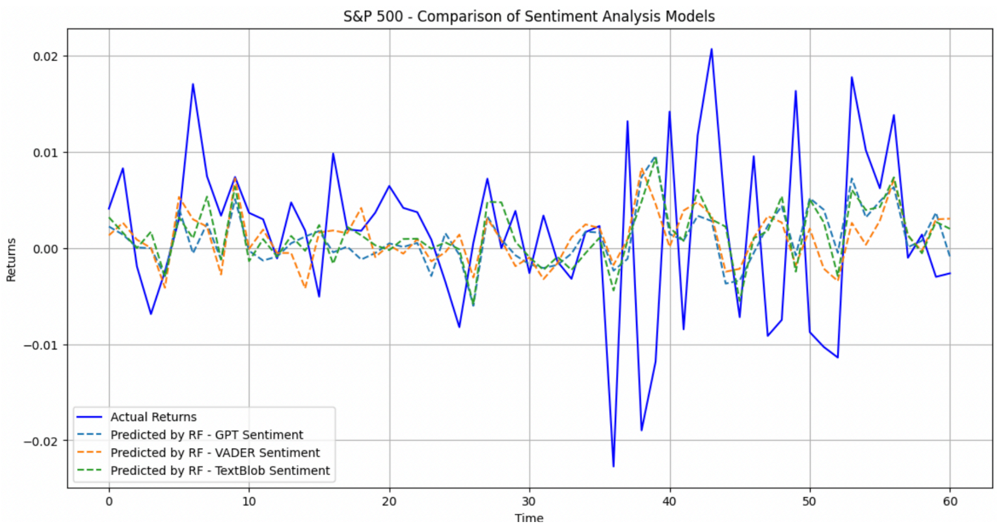

Figure 9: Random Forest on test dataset for S&P500

Interpretation of Results

We have to be realistic with these numbers, directional accuracy is a good metric to analyse performance of a model but it doesn't contain any information about the magnitude of the predictions relative to the actual return. There could be cases , where you can achieve 70% and still lose money depending on the trading startegy you implement.
Using the same sentiment analysis method across all models, we were able to directly compare the performances of our 3 algorithms.

For instance, using the sentiment scores derived from GPT-3.5-turbo, we observed that the Random Forest model achieved the highest directional accuracy for Tesla, significantly outperforming Gradient Boosting and LSTM.
The graphs showing which is the best algorithm on the test dataset for each sentimental analyser and stocks are in the appendix. What we ended up with is Random Forest being the best in 4 cases out of the 6 and LSTM to steal the last 2 cases, it may be explained by the ability to capture time series structure for the latter and being extremly powerfull for RF.

These comparisons highlight that the choice of algorithm can significantly impact the prediction accuracy, and no single model consistently outperforms others across different sentiment analysis methods and datasets. Gradient Boosting was not as good as the others, it still has correct predictions. Therefore, it is crucial to evaluate multiple models to identify the most suitable one for a given

Conclusions

The choice of sentiment analysis method and the tuning of model parameters have significant impacts on the performance of stock return predictions. This analysis demonstrates the importance of model selection and parameter optimization in financial forecasting, especially when dealing with complex data sources like sentiment scores derived from social media.

Further research could explore the combination of these models or the integration of additional data sources to enhance prediction accuracy and reliability.

8 Discussion

The analysis conducted in this study underscores the nuanced relationship between sentiment scores derived from Reddit comments and stock market returns. Despite employing advanced sentiment analysis techniques and various machine learning models, the results highlight several key observations and challenges:

- Contextual Relevance: The effectiveness of sentiment analysis tools like VADER, TextBlob, and GPT-3.5-turbo varied, with each tool demonstrating strengths in different contexts. This variability underscores the importance of selecting the right tool based on the specific nature of the data and the required depth of analysis.
- Model Performance Variability: The models exhibited different levels of accuracy, with the Random Forest model showing particularly strong performance in directional accuracy for Tesla when using GPT-derived sentiment scores. This suggests that combining deep learning-based sentiment analysis with ensemble machine learning models may capture complex patterns more effectively, although the success of such combinations can vary significantly across different stocks and market conditions.
- Data Volume and Quality: The limited volume of contextually rich comments poses a significant challenge, as seen from the analysis where only a small fraction of the comments could be reliably labeled and used for training predictive models. This limitation highlights the critical role of data quality and quantity in developing robust financial prediction models.
- Implications for Trading Strategies: The moderate success in directional accuracy suggests that while sentiment analysis can provide valuable insights, it may not suffice on its own to form a reliable trading strategy. The findings suggest the need for incorporating comprehensive risk management strategies to account for the probabilistic nature of prediction models based on social media sentiment.

Conclusion

This study aimed to assess the feasibility of generating actionable trading signals from Reddit comment sentiment. Despite the sophisticated tools and techniques employed, the findings suggest that while there is potential in using sentiment analysis as part of a broader investment strategy, reliance solely on sentiment data from platforms like Reddit is insufficient. The mixed performance across different models and sentiment tools indicates that sentiment analysis should be integrated with other forms of financial analysis to enhance its predictive power and reliability.

Future research should focus on expanding the dataset, integrating diverse sentiment indicators, and exploring hybrid models that combine traditional financial indicators with sentiment data. Additionally, further exploration into the optimization of model parameters and the use of real-time data could enhance the applicability and effectiveness of sentiment-based trading strategies.

By advancing in these areas, researchers and practitioners can better harness the potential of big data and machine learning in financial markets, moving closer to developing robust, real-world applications that capitalize on the vast amounts of unstructured data generated by social media platforms.

References

[1] Hugging Face. *Transformers: State-of-the-art Natural Language Processing*. 2021. URL: https://huggingface.co/transformers/.

[2] J. Jakubik et al. “Incorporating financial news for forecasting Bitcoin prices based on long short-term memory networks”. In: *Quantitative Finance* (2023).

[3] Kaggle. *Reddit Comments Dataset*. 2021. URL: https://www.kaggle.com/datasets.

[4] Tim Loughran and Bill McDonald. “When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks”. In: *The Journal of Finance* 66 (2011), pp. 35–65.

[5] NLTK. *Natural Language Toolkit*. 2021. URL: https://www.nltk.org.

[6] OpenAI. *GPT-3 API*. 2021. URL: https://beta.openai.com/.

[7] Zacharias Sautner et al. “Firm-Level Climate Change Exposure”. In: *The Journal of Finance* (2023). URL: https://doi.org/10.1111/jofi.13219.

[8] Ashish Vaswani et al. “Attention Is All You Need”. In: *arXiv preprint arXiv:1706.03762* (2017). URL: https://doi.org/10.48550/arXiv.1706.03762.

Appendix

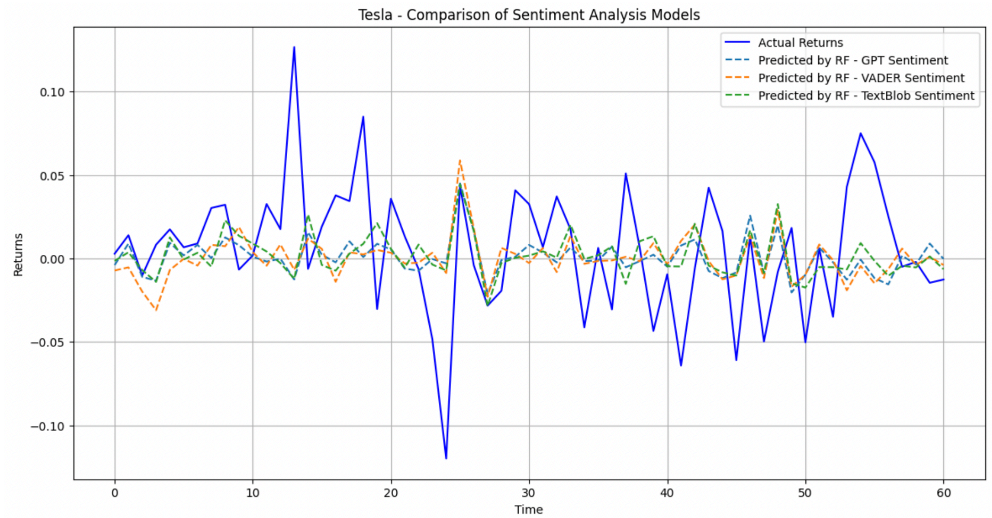

Figure 10: Gradient Boosting on test dataset for Tesla

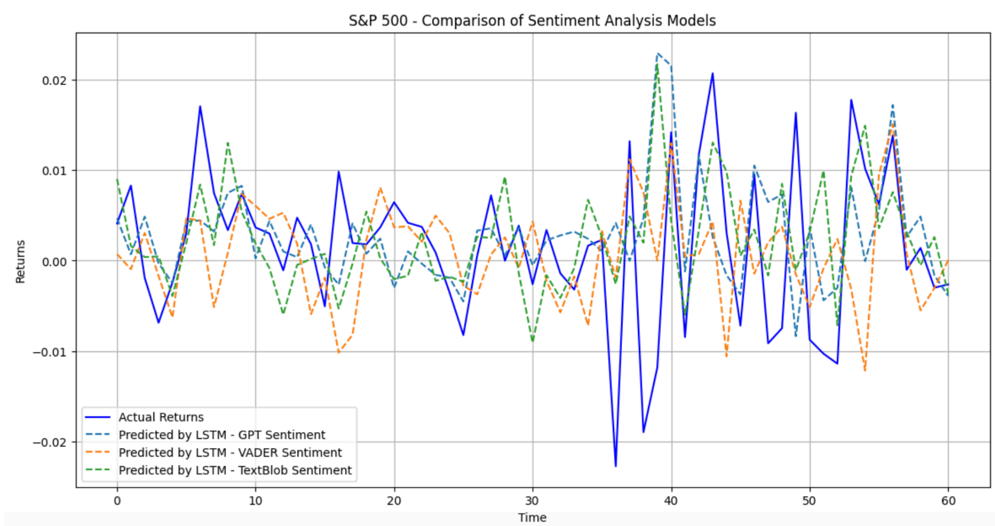

Figure 11: LSTM on test dataset for S&P 500

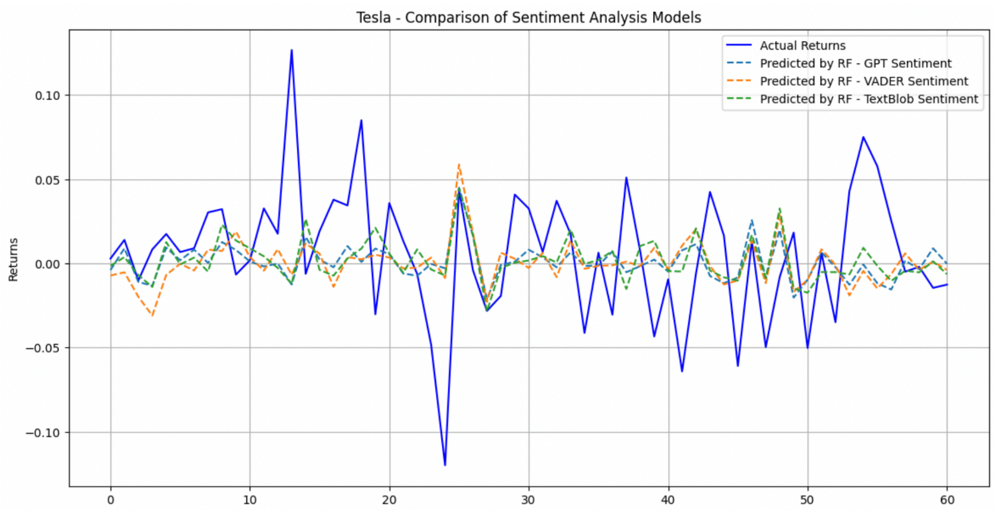

Figure 12: Random Forest on test dataset for Tesla

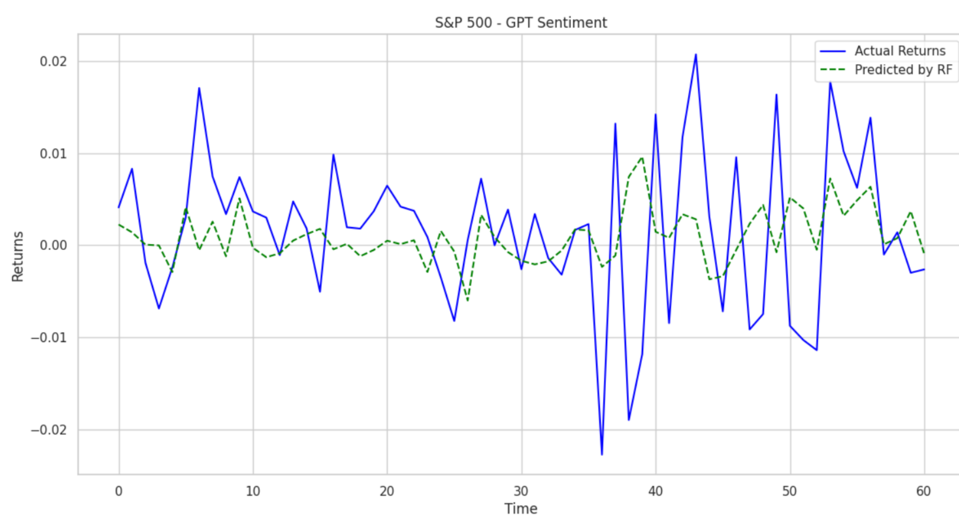

Figure 13: Best Model using GPT for SP500

Figure 14: Best Model using GPT for TESLA

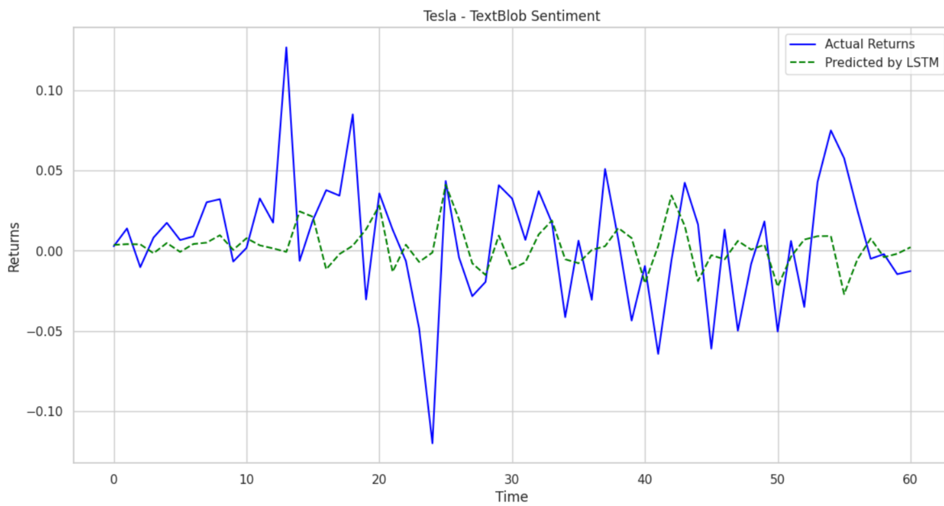

Figure 15: Best Model using TEXTBLOB for TESLA

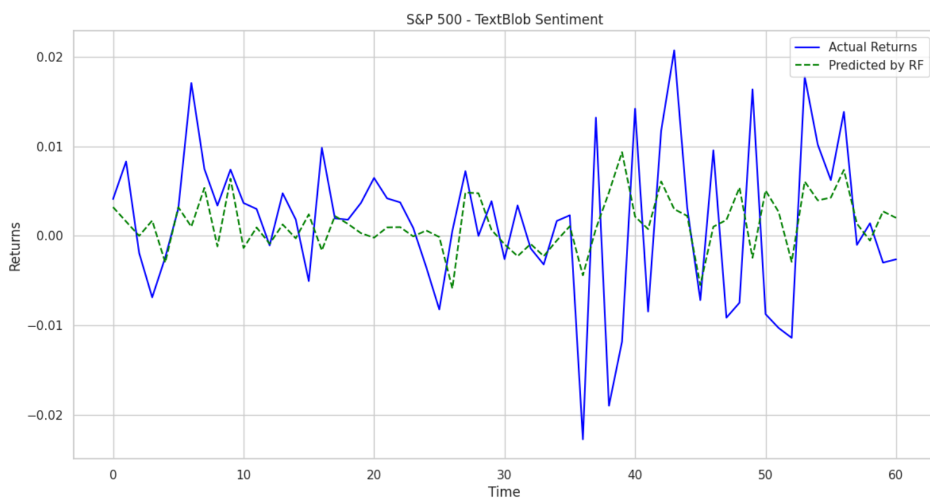

Figure 16: Best Model using TEXTBLOB for SP500

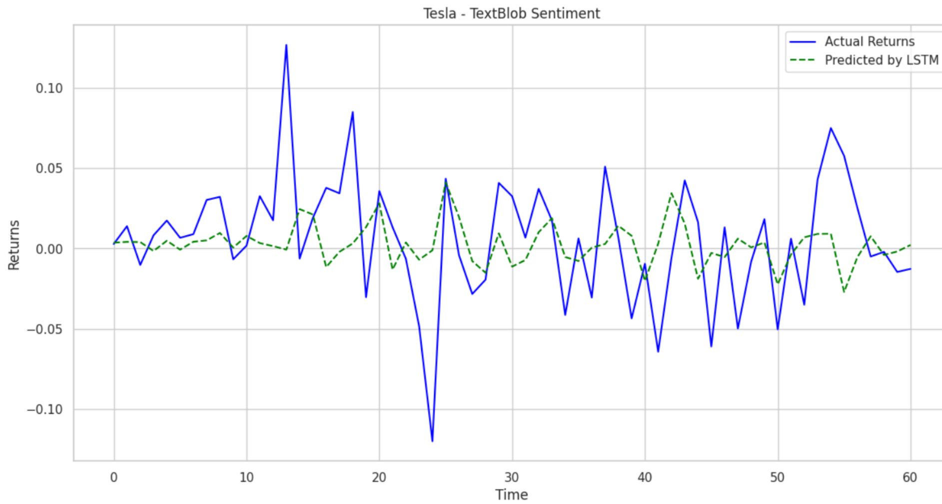

Figure 17: Best Model using TEXTBLOB for TESLA

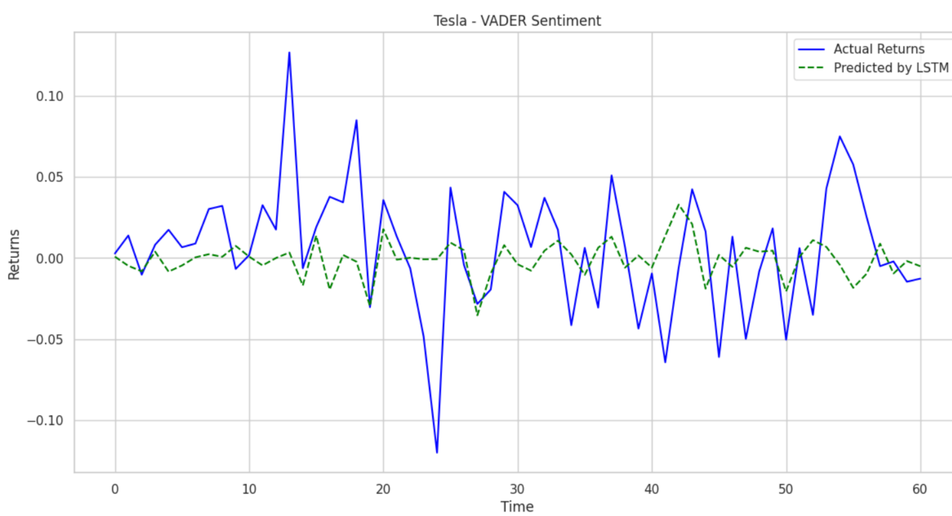

Figure 18: Best Model using VADER for TESLA

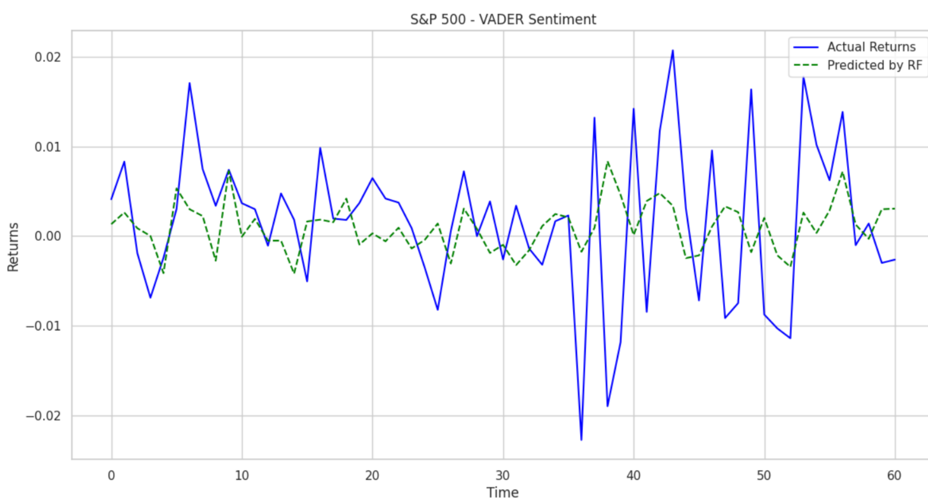

Figure 19: Best Model using VADER for SP500