

# Analyzing the Correlation Between Retail Traders' Sentiments and Equity Market Movements

Haozhe Zeng | Cornell University | [hz657@cornell.edu](mailto:hz657@cornell.edu)

Zixiao Wang | Cornell University | [zw699@cornell.edu](mailto:zw699@cornell.edu)

<https://github.com/howie-zeng/Analyzing-the-Correlation-Between-Retail-Traders--Sentiments-and-Equity-Market-Movements>

## Abstract

This research investigates the influence of retail trader sentiment on equity market dynamics, leveraging cutting-edge Natural Language Processing (NLP) models, notably BERT, to analyze discourse on social media platforms such as Twitter and Reddit. By defining the parameters of this relationship, the study progresses to develop sophisticated machine learning models that integrate daily stock data with insights derived from sentiment analysis. This approach aims to unearth potential trading signals, tailored to adapt to market volatility. The emphasis lies on harnessing the collective sentiment's predictive power alongside technical market indicators, thereby offering a novel perspective for anticipating stock price movements. Furthermore, the project includes the creation of a user-friendly interface designed to graphically represent the generated buy and sell signals, enhancing interpretability and usability of the findings.

## Introduction

The recent years have seen a paradigm shift in the financial landscape, primarily fueled by the digitization of trading platforms. This technological revolution has democratized access to stock trading, empowering a wave of retail traders who were previously overshadowed by institutional investors. Characterized by their swift decision-making and collective actions, these retail traders have emerged as significant influencers in the equity markets.

A quintessential example of this new era of retail trading is the WallStreetBets forum on Reddit. This

platform has evolved into a pivotal hub for retail traders, where they exchange insights, strategies, and sentiments about various stocks. The impact of such collective sentiment was strikingly demonstrated in events like the [GameStop short squeeze](#), where coordinated actions, driven by discussions and emotions on the forum, led to extraordinary stock price fluctuations, surprising many institutional investors.

Despite the visibility of events like the GameStop, a comprehensive analysis examining the broader influence of such forums on the equity market remains unexplored. Our research seeks to fill this void by conducting an in-depth analysis of the sentiments expressed on these platforms and their correlation with market movements. Our objective extends beyond identifying superficial correlations; we aim to discern the depth and endurance of sentiment's influence on stock prices.

Furthermore, we investigate whether these collective sentiments can be harnessed as a predictive tool for market trends. To this end, the study will not only provide comprehensive insights into the relationship between retail trader sentiments and market behavior but will also explore the feasibility of developing a time series machine learning model. This model will aim to predict stock returns and prices by integrating sentiment analysis, thereby offering a novel approach to understanding and forecasting market dynamics in the digital age.

## Why this approach?

The interplay between retail sentiment and stock market dynamics has been the subject of various

studies, with a consensus pointing towards a positive correlation. Concurrently, the application of machine learning in forecasting stock prices has piqued the interest of researchers. Yet, the literature reveals a scarcity in attempts to synthesize sentiment analysis with machine learning for the purpose of predicting market movements.

Predominant models in existing research tend to diverge into two distinct streams. One stream focuses on leveraging machine learning for long-term stock price forecasting, often marginalizing the role of sentiment analysis. The other stream neglects the sentiment dimension altogether, which may lead to a myopic understanding of market dynamics. This dichotomy represents a critical gap, given the multifaceted and dynamic nature of the stock market, which is influenced by a complex array of factors beyond historical price trends.

Acknowledging this, our approach advocates for a more nuanced and adaptive methodology. We propose the integration of sentiment analysis into a machine learning framework, capitalizing on the predictive power of retail sentiment as a contemporaneous market indicator. Moreover, we introduce the dynamic decision making as a core component of our model. This method facilitates periodic retraining and recalibration of the model, aligning it with the latest market data. Such a strategy ensures that the model remains responsive to market fluctuations, thereby enhancing its predictive accuracy and robustness over time.

In essence, our approach is designed to capture the dynamic interplay between market sentiment and price movements, offering a more holistic and agile forecasting tool that aligns with the ever-evolving landscape of the stock market.

## Problem Statement:

Given a stock market with a set of stocks  $S = \{s_1, s_2, \dots, s_n\}$ , where each stock  $s_i$  has a daily closing price  $p_i(t)$  at time  $t$ , and a corresponding set of daily retail sentiment scores  $R = \{r_1(t), r_2(t), \dots, r_n(t)\}$ , the problem is to construct a predictive model  $M$  that forecasts the future price  $\hat{p}_i(t + \Delta t)$  of stock  $s_i$  at a future time  $t + \Delta t$ .

The model  $M$  aims to leverage the sentiments extracted from social media platforms, quantified as sentiment scores  $r_i(t)$ , to predict the impact on the future stock prices. The sentiment scores are derived from the analysis of textual data using Natural Language Processing (NLP) techniques, capturing the collective mood and opinions of retail traders.

The predictive model  $M$  will be evaluated based on its accuracy in forecasting the price  $\hat{p}_i(t + \Delta t)$ , using standard metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). The model will also be assessed for its robustness across different stock categories and market conditions.

The ultimate goal is to determine the efficacy of retail sentiment scores as predictors of stock market behavior and to establish a reliable method for stock price prediction that can aid in investment decision-making processes.

## Methods & Results

### NLP

Data Preparation:

Gathering relevant, high-quality data presented challenges due to API rate limits on platforms like Reddit and Twitter. We explored alternatives by sourcing data from Kaggle, open-source datasets, and Twitter posts, including informal language. We also incorporated labeled news headlines to capture real-time market influences.

Our decision to diversify data sources was driven by unique advantages:

1. Kaggle and research paper data offer specialized and well-curated content for sentiment analysis in the financial domain.
2. Twitter posts provide real-time, colloquial insights from market participants, helping us address the noise in social media discussions.
3. Labeled news headlines enable our model to respond dynamically to breaking news, mirroring market reactions.

4. We used ChatGPT to generate diverse conversational-style data, which was labeled with sentiments for a well-rounded training dataset.

To bolster the quality of our training data, a variety of data augmentation techniques were employed.

1. Synonym Replacement: This technique replaces words in a sentence with synonyms to add variety. For instance, changing "Worried about the recent drop in the price of gold" to "Concerned about the recent decline in the value of gold."

2. Back Translation: This involves translating text into another language and then back into the original language, introducing subtle phrasing changes. For example, "AAPL's product launch was underwhelming, considering selling our shares" might become "AAPL's product launch was disappointing; thinking about divesting our shares."

3. Paraphrasing: It provides alternative sentence structures and expressions. For instance, "Just sold our Amazon shares; they've become too expensive" could be paraphrased as "We've recently disposed of our Amazon holdings as they've become unaffordable."

4. Oversampling/Undersampling: These techniques address class imbalances, ensuring equal representation of sentiment categories. If there's an imbalance, oversampling duplicates examples from the minority class, while undersampling reduces examples from the majority class. Here, we oversampled the negative data to create a balanced training set.

One pressing issue we faced was dataset bias, which can affect model performance and fairness. To mitigate bias, we used ChatGPT's natural language processing to balance the data and ensure diverse perspectives. This helped us create a more equitable dataset, forming a strong foundation for our project.

Our labeled dataset is divided into three subsets:

1. Training Dataset (8,925 rows): Used for model training.

2. Validation Dataset (2,232 rows): For hyperparameter tuning and model evaluation.

3. Test Dataset (876 rows): Reserved for final model performance evaluation.

We also collected unlabeled comment posts in three subsets based on their characteristics:

1. Tweet\_filtered\_TSLA Dataset (1,123,262 rows): Tweets with dates and stock mentions.

2. stock\_tweets\_filtered\_TSLA Dataset (37,422 rows): Focused on TSLA-related tweets.

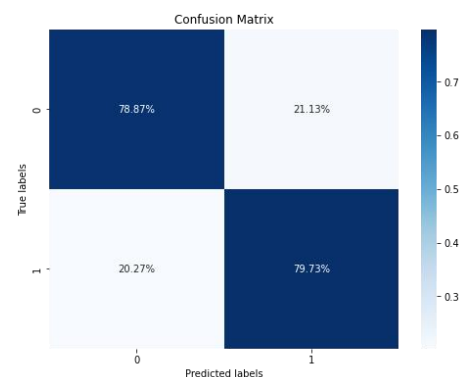
3. tweets\_remaining\_filtered\_TSLA Dataset (60,836 rows): Additional TSLA-related tweets.

We chose not to combine labeled and unlabeled datasets due to significant differences in their origins and potential data variations. Unlabeled posts will undergo labeling through natural language processing, making them ready for integration into our stock price prediction model.

## II. Model Testing

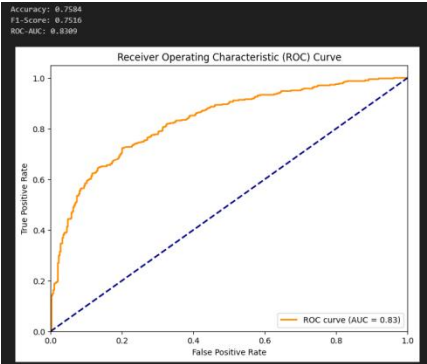
1. FastText & Word2Vec Embedding Model:

We began with a dataset of over 1.6 million Twitter posts, which covered a wide range of topics beyond the equity market. In contrast, equity-specific datasets were smaller, typically around 8,000 entries. We used the larger dataset for training and the smaller, equity-specific ones for validation. Due to the larger dataset's non-specific nature and many irrelevant posts, our model achieved 80% training set accuracy but dropped to approximately 60% accuracy on the test set, as shown in the confusion matrices below.

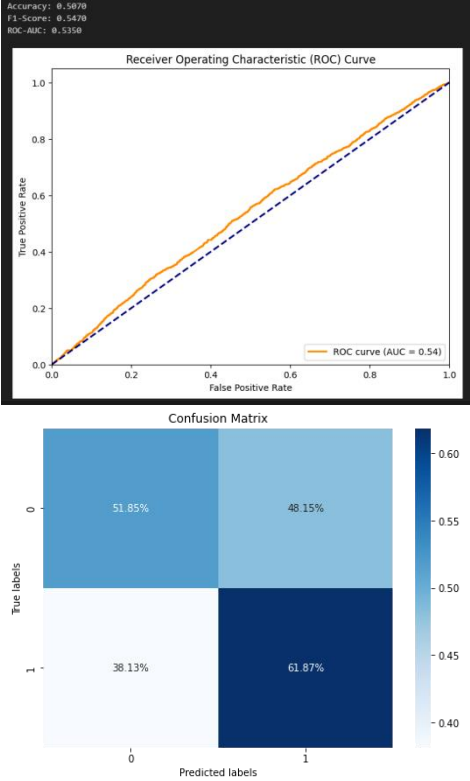


Regarding FastText embeddings, we trained the model on the initial dataset for three epochs, resulting

in an ROC-AUC score of 0.83. This score indicates effective discrimination between positive and negative cases within the training data. FastText's ability to capture subword information proves valuable, especially for languages with complex morphology and out-of-vocabulary words. It provides richer and more context-aware representations compared to Word2Vec, which we previously experimented with.



However, when the same model is subjected to testing using an entirely unseen dataset, a disparity emerges. The model performs suboptimally on this new data, as indicated by an ROC-AUC score of only 0.535, which is significantly lower than the training performance. This discrepancy, where the model's performance regresses when applied to unseen data and its ROC-AUC score falls closer to the baseline value of 0.5, is far from ideal. It suggests that the model might not generalize well to new, unseen instances and may need further refinement or adjustments to enhance its predictive capabilities on diverse datasets.



2. LSTM Model:

The initial training and testing accuracy using the RNN LSTM neural network displayed a noticeable disparity, with a commendable 0.79 on the validation set but a less satisfactory 0.58 on the testing set. This discrepancy raised concerns, as the model appeared to perform exceptionally well within the known confines of the training data but struggled when presented with new, unseen data. This disparity prompted the comprehensive evaluation of the data and model, necessitating an exploration into potential improvements in the training dataset composition and, perhaps, model architecture, to achieve a more balanced and consistent performance.

	Precision	Recall	F1-Score	Support
NEGATIVE	0.79	0.79	0.79	159,563
POSITIVE	0.80	0.79	0.79	160,437
Accuracy			0.79	320,880
Macro Avg	0.79	0.79	0.79	320,880
Weighted Avg	0.79	0.79	0.79	320,880

	Precision	Recall	F1-Score	Support
NEGATIVE	0.44	0.52	0.47	2,106
POSITIVE	0.69	0.62	0.65	3,685
Accuracy			0.58	5,791
Macro Avg	0.56	0.57	0.56	5,791
Weighted Avg	0.60	0.58	0.59	5,791

POSITIVE	0.76	0.75	0.75	160,506
Accuracy			0.76	320,000
Macro Avg	0.76	0.76	0.76	320,000
Weighted Avg	0.76	0.76	0.76	320,000

	Precision	Recall	F1-Score	Support
NEGATIVE	0.40	0.54	0.46	2106
POSITIVE	0.67	0.54	0.60	3685
Accuracy			0.54	5791
Macro Avg	0.54	0.54	0.53	5791
Weighted Avg	0.57	0.54	0.55	5791

Likewise, the experiment incorporating the XGBoost model resulted in a training validation accuracy of 0.68 and a testing accuracy of 0.61, mirroring the outcomes achieved with the initial neural network approach. These consistent results across different machine learning methods emphasize the persistent challenges presented by the dataset's composition. It underscores the necessity for further data preprocessing, feature engineering, or the exploration of alternative data sources to enhance the model's capability to discern equity market-related sentiments effectively.

Metric	Training Data	Testing Data
XGBoost Accuracy	0.685049375	0.61405629424969
XGBoost ROC-AUC	0.762493166893	0.55374410881275
XGBoost Precision	0.640924181286	0.64333728746548
XGBoost Recall	0.84160625	0.88383934871099
XGBoost F1-score	6.727682544794	0.74436692210911

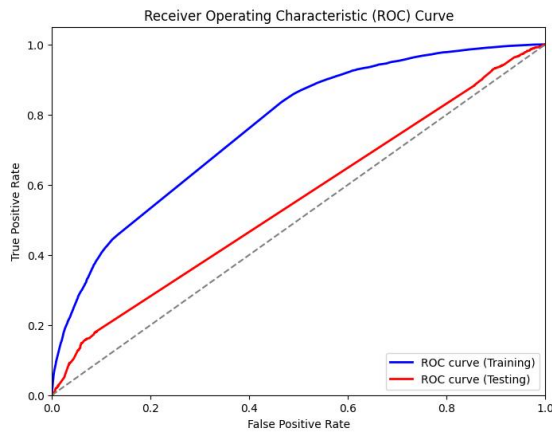
	Precision	Recall	F1-Score	Support
NEGATIVE	0.75	0.76	0.76	159,494

When confronted with the drop in testing accuracy, we initiated a comparative study involving different machine learning models. The objective was to discern if the discrepancy in performance was a result of the training data's quality or if it stemmed from the chosen model's limitations.

### 3. Naive Bayes, Random Forest, XGBoost Models:

For a comprehensive comparative analysis, we selected a diverse set of machine learning models, each known for specific strengths in sentiment analysis across various data types. Our ensemble included the Naive Bayes classifier for simplicity and robustness, Random Forest for capturing complex relationships, and XGBoost for versatility and efficiency. This choice aimed to determine if the dip in testing accuracy was due to model intricacies or tied to the training data composition.

The Naive Bayes experiment yielded a training validation accuracy of 0.76 and a testing accuracy of 0.54, surprisingly not significantly different from the initial neural network approach. This suggests that the issue may not solely be attributed to the choice of machine learning model but could be influenced by challenges posed by the dataset itself. These findings emphasize the need to address data quality and diversity to enhance overall model performance.



Overall, the intriguing discovery was that the testing accuracy, obtained from these alternative models, exhibited no substantial improvement over the initial results achieved with the more complex RNN LSTM neural network. This outcome suggested that the model's complexity wasn't the primary bottleneck in this scenario.

#### 4. BERT Model

Due to challenges with our training dataset, including unrelated Twitter posts, we recognized the need to improve model performance. We decided to harness the power of BERT, specifically FinBERT, a specialized natural language processing model tailored for financial text and sentiment analysis. FinBERT is fine-tuned in the finance domain using a vast financial corpus, with the Financial PhraseBank dataset playing a crucial role in precise sentiment classification within financial contexts. For more details, refer to the paper "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models" and our related Medium blog post.

We deployed the model using the Hugging Face Query API, leveraging the repository "tarnformnet/Stock-Sentiment-Bert." The model exceeded our expectations, achieving a test dataset accuracy of 0.68. Additionally, we explored an alternative variant, ProsusAI/finbert, which provides softmax outputs for three sentiment labels: positive, negative, and neutral.

Accuracy			0.68	50
----------	--	--	------	----

Macro Avg	0.47	0.46	0.46	50
Weighted Avg	0.74	0.68	0.71	50

However, given the binary nature of our testing dataset, the accuracy is not good on the testing data with accuracy rate only 0.27. So, we endeavored to further fine-tune the model to align it with the specific requirements of our dataset.

Accuracy			0.27	100
Macro Avg	0.48	0.21	0.28	100
Weighted Avg	0.84	0.27	0.40	100

In our pursuit of further refining the model, we embarked on a journey to fine-tune it using the 'yiyanghkust/finbert-tone' model, closely following the comprehensive guidelines they provided. Unfortunately, during this process, we encountered certain challenges stemming from compatibility issues with the environment and libraries, leading to an unsuccessful attempt.

In response, we decided to explore an alternative approach by training the BERT model from scratch using the 'bert-base-uncased,' the original uncased base model, in combination with the newly acquired financial data. This method offered the advantage of full control and customization over the training process, enabling us to align the model precisely with our specific requirements.

The provided model structure is a variant of the BERT model called BertForSequenceClassification. It is initialized with weights from the "bert-base-uncased" checkpoint and some weights have been newly initialized for the specific downstream classification task.

The model architecture combines a pre-trained BERT model with additional layers for sequence classification. BERT consists of various components, including embeddings, an encoder, and a pooler. The embeddings layer incorporates word, position, and

token type embeddings to understand text sequence structure. The encoder, comprising multiple layers (12 in this case), employs self-attention, feed-forward layers, and output layers to capture contextual information and relationships. The pooler generates a fixed-size representation of the input sequence, typically used for classification tasks.

To prevent overfitting, a dropout layer is applied at various points within the model. It randomly sets a fraction of input units to zero during training, with a probability parameter (0.1 in this case) controlling the dropout rate.

The "classifier" is a linear layer that maps the BERT model's output to the specific classification task. It has 768 input features (matching BERT's output size) and 2 output features, suitable for a binary classification task. The number of output features can be adjusted for different tasks.

BERT stands out in Natural Language Processing due to several features: bidirectionality, as it considers both left and right context; pre-training on a large text corpus to acquire rich language representations and world knowledge; and its large-scale architecture, which captures intricate language patterns but demands significant computational resources [Devlin et al., 2018].

In our effort to refine our model, we explored the "yiyanghkust/finbert-tone" repository for improved sentiment analysis capabilities. However, compatibility issues in our environment prevented us from proceeding with this tool.

Undeterred, we chose to train a BERT model from scratch, utilizing the 'bert-base-uncased' model as a foundation due to its reputation and adaptability. We augmented the model with an additional dataset for financial sentiment analysis.

In the next phase, we trained the new BERT-based model using the augmented dataset, targeting a minimum 80% accuracy benchmark. We employed a comprehensive pipeline involving various natural language processing stages, including tokenization with BertTokenizer, setting training parameters with TrainingArguments, and ensuring uniform sequence lengths with DataCollatorWithPadding. Training

occurred on available GPU resources in Google Colab. A Trainer instance from transformers managed the training loop, with post-training evaluation utilizing standard metrics from scikit-learn, providing valuable insights into its effectiveness for the sequence classification task.

During the ten training epochs, we monitor key metrics like Training Loss, Validation Loss, Accuracy, and F1 Score to assess the model's learning and generalization. Each epoch represents a training phase, revealing how the model adapts and optimizes predictions over time. These metrics are crucial for diagnosing issues like overfitting and fine-tuning parameters, providing a balanced assessment of the model's performance.

Training Loss: Starts at 0.487900, decreases to 0.038000 by the ninth epoch, then slightly increases to 0.168500 in the tenth epoch. Indicates effective learning from the training dataset, though the final epoch may hint at overfitting.

Validation Loss: Begins at 0.373879 and rises to 0.682500 by the tenth epoch, suggesting potential overfitting as training progresses.

Accuracy: Remains relatively stable, starting at 0.840502 and ending at 0.866487, indicating consistent prediction accuracy despite increasing Validation Loss.

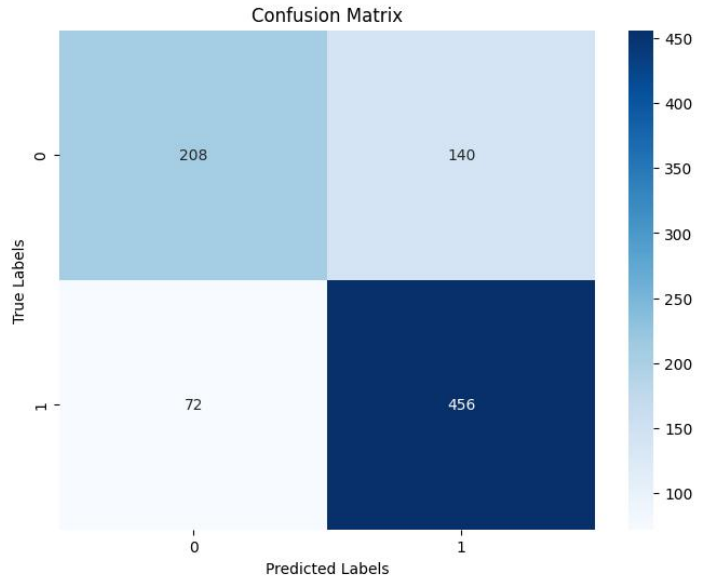
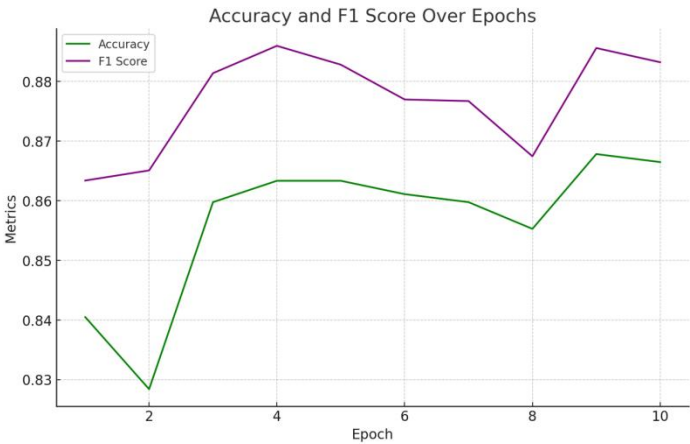
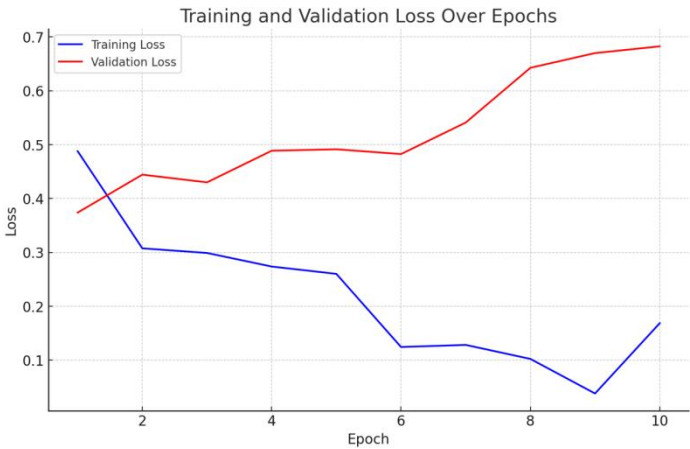
F1 Score: Balances precision and recall, showing stability with a slight overall increase from 0.863392 to 0.883229. The model maintains a good balance throughout training, even with the increasing Validation Loss.

Epoch	Training Loss	Validation Loss	Accuracy	F1
1	0.4879	0.373879	0.840502	0.863392
2	0.3075	0.444309	0.828405	0.865093
3	0.2989	0.429997	0.859767	0.881394
4	0.2737	0.48869	0.863351	0.885981
5	0.2601	0.491391	0.863351	0.882828
6	0.1243	0.482482	0.861111	0.876984
7	0.1282	0.540972	0.859767	0.876723
8	0.1022	0.642647	0.855287	0.86746
9	0.038	0.670001	0.867832	0.885615

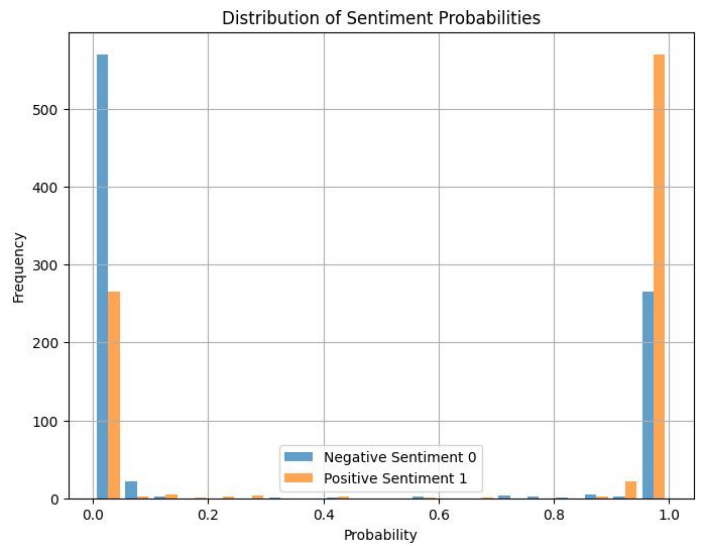


10	0.1685	0.6825	0.866487	0.883229
----	--------	--------	----------	----------

applications where both types of classification errors have significant consequences.



In the bar chart below, we see two clear peaks: one with high confidence predictions for negative sentiment (0) near probability 0, and another for positive sentiment (1) near probability 1. This distribution signifies strong and confident predictions, as most are concentrated at the extreme ends of the probability scale. It highlights a well-performing model with high certainty in its classifications.



The plot below displays two overlapping series of data points representing negative (label 0) and positive (label 1) sentiment probabilities. It reveals a model that predicts both sentiments across the dataset

### III. Model Prediction

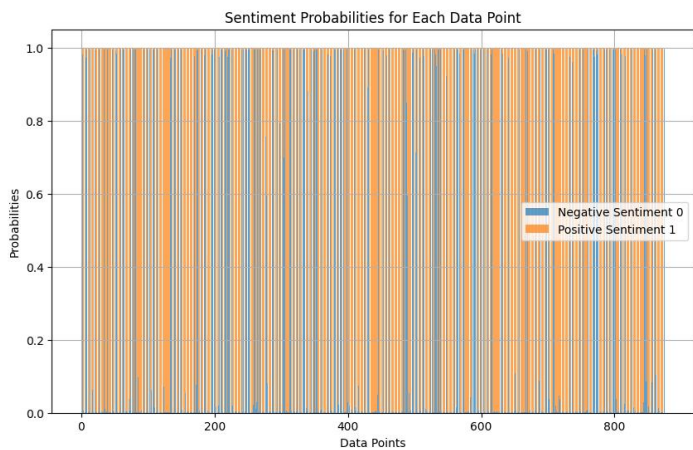
Using the trained model, we make predictions on an unknown dataset and further analyze the results.

The confusion matrix highlights the classifier's strong ability to correctly identify both classes. It shows higher numbers of true positives (456) and true negatives (208) compared to false positives (140) and false negatives (72). This indicates the model's effectiveness in identifying both positive and negative classes, with relatively fewer false predictions.

The lower occurrence of false predictions suggests the model's accuracy and promising generalization capability to classify new, unseen data. This also implies a balanced sensitivity and specificity, which is desirable in predictive models, especially in



without bias. The intermingling of blue and orange lines indicates balanced probability assignments, showcasing a well-calibrated model. This even distribution ensures the model is equally likely to predict positive or negative sentiment, preventing skewed interpretations.



After the predicted results analysis, we are confident that the model has a good performance.

IV. Predictions to be used in Stock Model

Once we achieved the desired accuracy, we integrated the prediction model's numeric outputs into our final stock price prediction model. This fusion allowed us to assign precise values to labels, enhancing the precision and data richness of our model. It was a significant step in rigorously testing our hypothesis and exploring the relationship between market sentiment and stock prices.

We chose numeric values over categorical values to increase information richness and flexibility. Numeric values offered a wide range of data for deeper insights and statistical analysis, including means, maximums, minimums, and variances. This approach enabled us to comprehensively study the interplay between market sentiment and stock prices, uncovering subtleties that impact stock price movements.

In essence, using numeric values provided the precision and adaptability needed to explore the complex dynamics of financial markets thoroughly. It empowered us to extract valuable insights for future

strategies and decisions, enhancing our ability to navigate the financial landscape.

Stock

Challenges: Objective

Several studies propose using a singular model to forecast stock returns for an extended period, sometimes spanning up to a hundred days. We find this approach potentially limiting. Given the dynamic nature of the market, relying on one model to predict returns over multiple days seems unrealistic.

In contrast, we advocate for a model that is recalibrated daily, leveraging fresh data for each day's prediction. After forecasting the next day's or even the next week's return, the model can then assimilate the actual return data for that day. This iterative approach allows the model to continually refine its predictions based on the latest market conditions. Termed the "rolling window" method, this strategy emphasizes daily predictions while updating the dataset after each forecast. Such an approach is more attuned to the market's dynamic, enhancing the accuracy and relevance of predictions.



Two critical components define a rolling window model: the window size and the duration of the return you're predicting. While this model excels in capturing market dynamics, it can be computationally demanding due to its iterative training nature. Determining the optimal window size poses a challenge, as it can range from a short span of 5 days to several thousand days. Naturally, larger window sizes intensify the computational burden. When working with intricate deep learning models like Long Short-Term Memory (LSTM), it might be more reasonable to set a threshold for deciding when to

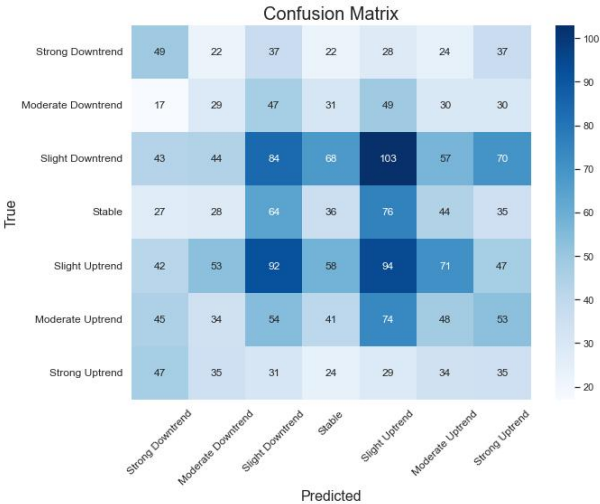
update the model, rather than retraining it at every iteration. This can balance the need for updated information with the practicalities of computational efficiency.

Moreover, determining the precise aspect to predict brings its own set of challenges. The main objective of this research is to identify the correlation between retail sentiment and stock movement. Given this aim, it initially seemed fitting to treat it as a classification challenge, aiming to predict if the stock movement for the next day would be positive or negative.

To achieve this, we used an XGBoost classifier as the baseline method. The methodology applied to categorize the next day's return was as follows:

1. If the return value fluctuated between -0.002 and 0.002, it was categorized as 'Stable'.
2. A slight increase between 0.002 and 0.01 was labeled as 'Slight Uptrend', while a slight decrease between -0.01 and -0.002 was termed as 'Slight Downtrend'.
3. If the rise was between 0.01 and 0.02, it indicated a 'Moderate Uptrend', and a fall between -0.02 and -0.01 indicated a 'Moderate Downtrend'.
4. Any return value above 0.02 was classified as a 'Strong Uptrend', while any value below -0.02 was termed as a 'Strong Downtrend'.

The initial approach to understanding stock movements involved a detailed categorization ranging from stable periods to pronounced uptrends and downtrends. The confusion matrix presented showcased the performance of this classification method. The methodology took approximately 20 minutes for execution, which may be deemed lengthy for real-time analysis. When using only the stock price as a predictor, the accuracy was marginally better than a random guess. This suggests a need for a more comprehensive and efficient methodology.



Recognizing the limitations of the classification approach, a shift towards regression was considered. The rationale was that predicting a continuous outcome (the stock's future price or return) might be more effective. Once the future price is estimated, it can then be discretized into categories. The revised strategy employed an XGBoost Regressor, aiming to predict the next day's return. The predicted return was then converted into a categorical representation of stock movement. The XGBoost Regressor was notably more efficient, completing its run in about 4 minutes. This model achieved an accuracy of approximately 20%. While this is an improvement over the classifier approach, there remains room for enhancement.

Current State: Stock

In the face of consistent challenges, it became evident that our model's focus on predicting next-day returns might not be the optimal approach. A deeper dive into the methodology and its implications illuminated several key insights.

- a) Temporal Dynamics of Sentiment: Leveraging sentiment analysis in our model highlighted that the effects of retail sentiment on stock prices aren't instantaneous. Rather, there's a lagged impact, reflecting a more gradual influence on stock movements.
- b) Uncertainty of Short-Term Predictions: The attempt to predict single-day returns proved fraught with uncertainties. Factors like daily news events, global market movements, and

institutional trading decisions can cause significant price fluctuations in the short term.

- c) **Broadening the Timeframe:** Our objective isn't about chasing daily fluctuations but understanding broader market dynamics. Adopting a swing trading perspective, which focuses on capturing gains in a stock (or any financial instrument) over a period of several days to weeks, aligns more closely with our goals.

Considering these insights, it is clear that a shift in strategy towards predicting mid to long-term stock movements, taking into account the more subtle and prolonged impacts of retail sentiment, could provide a more accurate and actionable framework for our endeavors.

Consequently, we shifted the focus of our model to forecast the returns for the upcoming week. Our primary interest transitioned from pinpointing stock movements to uncovering viable trading strategies, which we deem to be more pragmatic. As it stands, we employ a rolling window time series model. Each day, the model predicts the stock price for five days ahead and undergoes daily retraining to assimilate the latest information.

## Time Series Model

The objective now is to predict the stock price of APPLE for a given time frame. Various features and methodologies were experimented with, to improve the model's performance.

- a) **Dataset:** The dataset encompasses a comprehensive range of equities, including prominent stocks such as Apple, Amazon, QQQ, and SPY. It provides daily trading data, including the opening price, closing price, daily highs, daily lows, adjusted closing price, and trading volume. This dataset spans an extensive period from January 1, 2010, to January 1, 2023, offering a rich historical perspective for analysis.
- b) **Initial Metric:** The model started with a Mean Absolute Percentage Error (MAPE) of approximately 4%.
- a) **Model Optimization:**

- a) The model was adjusted to run on only 10% of the original time series data for feature selection, which optimized processing and ensured a more streamlined approach.

- b) **Feature Selection:**

- a) Several features were experimented with, including volume data, open price and its lags, highs and lows of a day, and economic indicators.
- b) Inclusion of moving averages (Mas) and SPY brought a significant increase in the model's performance.
- c) Several other features were added and tested such as RSI, WVAD, MACD, CCI, BOLL, and others. However, not all added significant value to the model's predictive capability.

- c) **Model Performance:**

- a) After multiple iterations, feature additions, and adjustments, the model achieved a MAPE of 1.87%. This is a notable improvement from the initial 4%.

- d) **Processing Time:**

- a) The model takes approximately 2.46 minutes to run on the entire dataset, demonstrating efficiency in processing.

- e) **Window Size:**

- a) A window size of 200 was used for the model, typically representing 1 year of stock history.

The stock price prediction model for APPLE has undergone extensive fine-tuning and experimentation. The emphasis on feature engineering and model adjustments has led to a significant improvement in prediction accuracy, as evidenced by the reduction in Mean absolute percentage error (MAPE) from 4% to 1.74% for the training set. The inclusion of moving averages (MAs) and SPY as features was especially beneficial, highlighting the importance of these variables in predicting APPLE's stock price.

1. **Date:** Represents the specific day for the data point, giving chronological context to the observations.

2. **Close:** The price at which the stock settled at the day's end.
3. **Close\_lag i:** A historic reference, this reflects the closing price from 'i' days ago, aiding in drawing comparisons over time. The current data set includes a 10-day lag.
4. **Volume:** Represents the sheer volume of shares that exchanged hands on that day, indicating the day's trading intensity.

5. **MA5**: Reflects short-term trends using a 5-day period.
6. **MA10 & MA20**: Capture medium-term movements.
7. **MA50 & MA200**: Provide insights into longer-term trends and are particularly watched by traders.

8. **WVAD:** This indicates the flow of money, revealing the balance between buying and selling pressure.

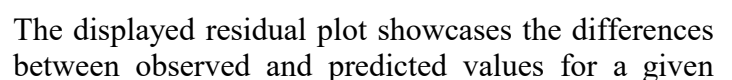
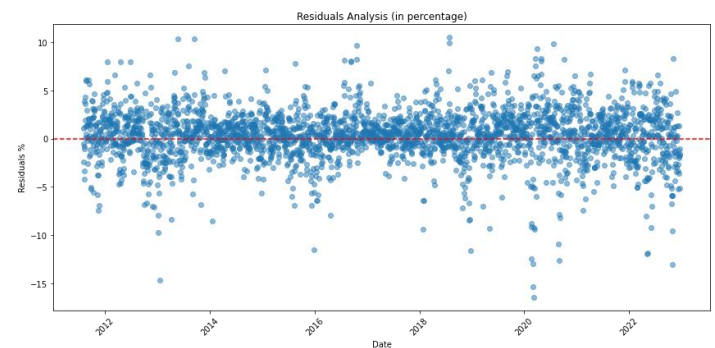
10. **macd\_line**: The main line indicating the trend.
11. **signal\_line**: The trigger for buy and sell signals.

13. **CCI:** Helps in determining cyclical trends.

15. **WVF, WVF\_color, upperBand & rangeHigh:**  
Relates to the Williams Vix Fix, identifying bottoms in stock advancements.

17. **AD**: Shows the flow of money, offering insights into the accumulation or distribution state of the stock.

Mean Squared Error (MSE):	4.568
Mean Absolute Percentage Error (MAPE)	1.87%
Root Mean Squared Error (RMSE)	2.137



model. The data points, represented as blue dots, seem to be scattered randomly around the horizontal red-dotted line, which signifies zero residual or perfect prediction. The random dispersion indicates that the model has a good fit for the data, as there's no discernible pattern or trend in the residuals. This suggests that the model's assumptions, particularly those regarding linearity, independence, and homoscedasticity, are likely met. However, there are a few notable outliers, which might require further investigation to understand if they result from specific external factors or data anomalies. Overall, the residual plot suggests a well-performing model, but attention should be given to the few outliers present.

### Current State: Trading Strategy

In the realm of financial forecasting, possessing merely a model that predicts weekly outcomes falls short of the comprehensive approach needed. What truly matters is the development of a sturdy methodology that seamlessly translates these projections into concrete, actionable measures, ultimately leading to a sophisticated trading strategy. To this end, we have architected a straightforward yet effective strategy that seamlessly integrates predictive return analytics with in-depth historical stock price information.

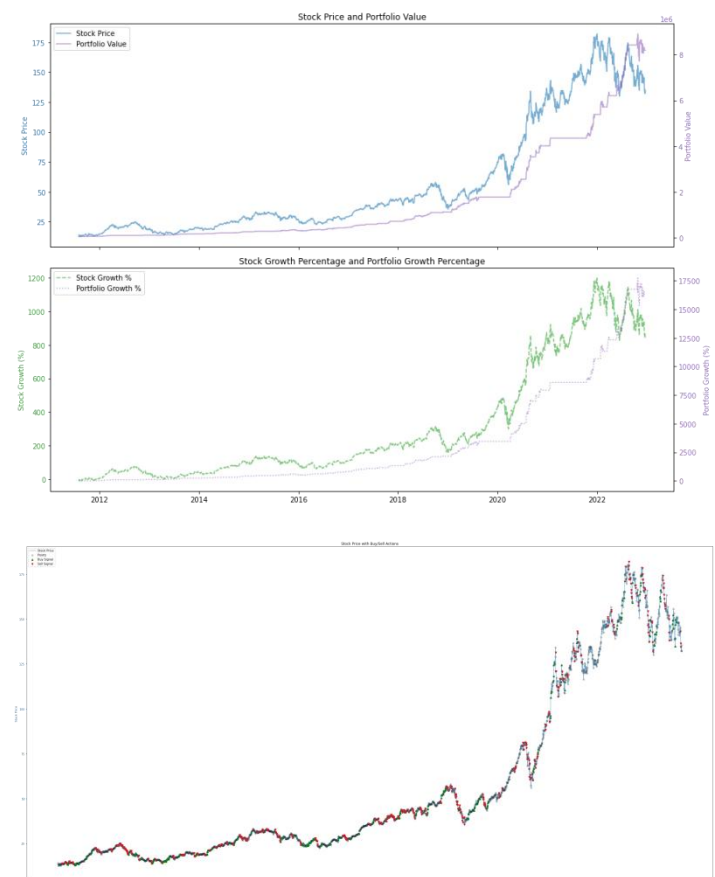
**Forecast Generation:** Predictions are meticulously crafted for each stock data entry following the stipulated window size. Concurrently, both the predicted and actual returns spanning a 5-day period are discerned.

**Portfolio Initialization:** The strategy springs to life with a pre-allotted capital and without any initial stock engagements. As it unfolds, the cumulative value of the portfolio, synthesizing both available capital and the value of held stocks, is consistently monitored and documented.

**Trading Logic—The Pivotal Mechanism:** At the heart of our strategic architecture lies the adaptive trading logic. This mechanism sets buy and sell benchmarks anchored on prior window\_size real returns, employing the 75th and 25th percentiles as guiding metrics. For every predictive interval, signals that either breach the buying criteria or fall below the selling criteria are registered. Decisions flow organically from these cues:

- Acquisition Strategy:** Should there be at least three robust buy prompts within the forecasted range and provided there's ample capital on hand, a stock purchase is greenlit.
- Divestment Strategy:** On the flip side, if three or more sell prompts surface and stocks are held, a divestiture move is undertaken.

**Strategy Visualization:** The tangible outcomes of the trading strategy are vividly portrayed through two primary graphical illustrations: The premier chart contrasts the stock's market performance with the trajectory of the investor's portfolio. In tandem, the following chart illuminates the parallel growth dynamics of the stock and the portfolio. Complementing these, an exhaustive visual narrative plots out each buy/sell maneuver against the backdrop of the stock's price timeline.



Currently, our model and trading strategy outperform the stock's return by 10x, demonstrating the ability to yield profits even amidst bearish market trends. While our present framework adeptly identifies selling cues,



it requires further refinement in effectively discerning buying signals.

**Robustness Test**

The robustness of the stock prediction model is assessed by examining its performance across various stocks, considering their average weekly returns and the volatility of these returns. The following table encapsulates the statistical data derived from the model's application to different stocks from 2010 to 2023:

The model demonstrates a consistent predictive accuracy across a range of stocks with varying levels of volatility. This consistency is indicative of the model's robustness and its potential utility as a predictive tool in diverse market conditions.

Stock	Model Performance (MAPE)	Volatility (Std. Dev. of Weekly Returns)
TSLA	3.16%	7.71%
AAPL	1.64%	3.89%
MSFT	1.46%	3.23%
AMZN	1.87%	4.31%
GOOG	1.51%	3.67%

The data reveals that the model maintains a high level of accuracy across various stocks, even when faced with differing degrees of return volatility. Notably, the model shows exceptional precision with AAPL and GOOG, where the Mean Absolute Percentage Error (MAPE) is relatively low, suggesting a stronger predictive capability for these stocks. This is particularly impressive given that these stocks also have lower volatility in their weekly returns, which may contribute to the model's effectiveness.

**Rolling Window Analysis**

In our pursuit to refine the stock prediction model, we conducted an in-depth analysis of the impact of varying rolling window sizes on the model's accuracy. This analysis was performed using Apple's stock data, ranging from 2010 to 2023. The rolling window approach is critical as it simulates a dynamic

environment where the model is periodically updated to reflect the most recent data, thereby potentially enhancing its predictive accuracy.

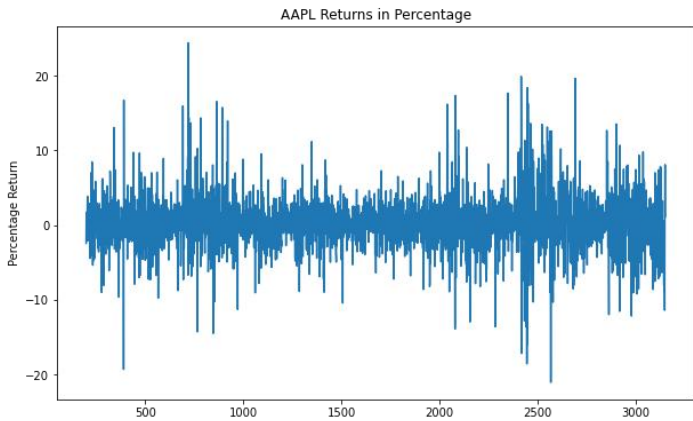
The table below presents the outcomes of this analysis, where we maintained all other variables constant while systematically altering the window size. The performance metrics used to evaluate the model's effectiveness are the Mean Absolute Percentage Error (MAPE) and the Root Mean Squared Error (RMSE), which provide insights into the model's precision and reliability.

Window Size (Days)	Mean Absolute Percentage Error (MAPE)	Root Mean Squared Error (RMSE)
5	1.64%	1.859
10	1.77%	2.034
20	1.76%	1.965
40	1.72%	1.985
80	1.73%	1.957
160	1.88%	2.122
320	1.98%	2.254
640	1.99%	2.375
1280	2.04%	2.683
2560	2.23%	4.278

The unexpected findings from our rolling window analysis on Apple's stock data suggest that smaller windows are more effective for predictive accuracy. A 5-day window achieved the lowest MAPE and RMSE, while larger windows resulted in higher error rates. This could be attributed to the non-stationary nature of stock prices. In traditional time series analysis, stationarity is a key assumption, meaning the statistical properties of the series do not change over time. Stock prices, however, are typically non-stationary—they do not hover around a constant mean or variance.

The paper by Bao, Yue, and Rao (2017) indicates that deep learning approaches, such as those involving stacked autoencoders and LSTM networks, can handle non-stationarity in financial time series data more effectively than traditional methods, which we are currently using. This could explain why smaller windows, which may capture more recent, relevant trends, outperform larger ones that include more historical data, potentially diluting the model's focus.

Given this context, it might be advantageous to predict daily returns as a percentage rather than absolute stock prices. Returns are often considered stationary because they represent relative changes, which are more likely to fluctuate around a constant mean. The provided ADF (Augmented Dickey-Fuller) statistics and stock return of APPLE support this, with a highly negative ADF statistic and a p-value of 0.0, indicating strong evidence against the null hypothesis of a unit root (non-stationarity).



percentage returns of Apple's stock do not follow a random walk and have a consistent pattern over the 10-year period, which is favorable for predictive modeling using traditional time series analysis techniques.

By focusing on returns instead of prices, we may improve the model's predictive performance, as stationary data aligns better with the assumptions of many time series forecasting methods. This shift could potentially address the limitations observed with non-stationary price data and lead to more accurate and reliable predictions.

To investigate whether the non-stationary nature of stock prices negatively impacts the performance of our predictive model, we developed an alternative model using identical parameters and features. However, instead of predicting stock prices, this new model forecasts stock returns. Here are the comparative results:

Model	Mean Square Error	Mean Absolute Error
Pricing Model	7.95	2.03
Return Model	4.79	1.56

The Return Model exhibits a lower Mean Squared Error (MSE) and Mean Absolute Error (MAE) compared to the Pricing Model. This suggests that forecasting returns, which are typically stationary, results in more accurate predictions than forecasting non-stationary prices, when apply XGBoost model (non-neural network model).

Potential issues with return model

The transition to a return-based model, while beneficial for its predictive accuracy, presents a new challenge: traditional market indicators like Bollinger Bands, which are price-based and inherently non-stationary, lose their direct applicability. These indicators are designed to provide insights based on the price movements and volatility of stocks, which are not immediately translatable to a model that focuses on returns.

ADF Statistic	54.82
p-value	0.0
Critical Value (1%)	-3.432
Critical Value (5%)	-2.862
Critical Value (10%)	-2.567

The critical values further reinforce this, as the ADF statistic is well below the threshold for the 1%, 5%, and 10% levels, suggesting that the series is stationary. In layman's terms, the statistical test confirms that the



To address this, we must consider alternative approaches to incorporate the valuable information these indicators provide. One potential solution is to transform these indicators into a form that reflects the relative change in price, which could then be aligned with the return-based perspective of the model. For instance, instead of using Bollinger Bands based on absolute price, we could calculate them based on a percentage change from a moving average.

Another approach could involve the development of new, return-oriented indicators that capture similar aspects of market behavior as Bollinger Bands do for price. These indicators would need to be designed to reflect the volatility and trends in stock returns, rather than stock prices.

In summary, while the shift to a return-based model necessitates a reevaluation of traditional price-based indicators, it also opens up avenues for innovation in the development of new analytical tools tailored for return-based analysis. This adaptation will be crucial for maintaining the integrity and usefulness of technical indicators within the context of a return-focused predictive model.

## Next Step

Moving forward, there are several avenues to explore to enhance the robustness and efficacy of our stock prediction model:

- a) **Feature Enrichment:** Dive deeper into the inclusion of potential predictors that encapsulate global economic shifts and overarching market dynamics. Such features can be pivotal in capturing exogenous shocks and external factors influencing stock prices.
- b) **Addressing Multicollinearity:** A systematic evaluation of the features is essential to ascertain any collinearity present. Multicollinearity can undermine the model's interpretability and diminish its predictive prowess. Utilizing techniques like Variance Inflation Factor (VIF) can assist in detecting and mitigating these issues.
- c) **Advanced Trading Strategies:** Expand the trading strategy's scope to encompass more sophisticated tactics such as short-selling. This would allow capitalization on both upward and downward

market movements, offering a more holistic trading approach.

- d) **Refining Buy-Signal Identification:** Given the current model's shortcoming in accurately pinpointing buying signals, targeted efforts should be made to optimize this aspect. This might involve recalibrating threshold values or integrating alternative algorithms.
- e) **Incorporating Sentiment Analysis:** A key dimension that's often overlooked is the sentiment prevailing among retail investors. Once a reliable sentiment analysis model is in place, merging it with the current framework could provide a more rounded perspective on market movements. Analyzing chatter on social media platforms, financial forums, or news outlets can be instrumental in this regard.
- f) **Model Evaluation and Continuous Feedback:** It would be prudent to establish a feedback loop where the model's predictions are constantly compared with actual outcomes. Such a mechanism would be invaluable for ongoing model refinement. Also, other models other than XGBoost are left to be experimented.
- a) **Stress Testing:** Given the unpredictable nature of financial markets, stress-testing the model under various hypothetical adverse scenarios can provide insights into its resilience and areas of potential vulnerability.

By adopting these strategies and continually iterating based on real-world outcomes, we can aspire to achieve a state-of-the-art stock prediction model that's both adaptive and predictive in an ever-evolving market landscape.

## Evaluation:

We provided an empirical evaluation of the performance of the developed system in terms of quality, efficiency, and robustness, compared with baseline techniques:

### Quality

1. **Model Performance:** The implementation of various NLP models, particularly the BERT-based models, has shown a significant improvement in accuracy and precision. The report details the adaptation of FinBERT, a variant of BERT, which is fine-tuned for the finance domain, resulting in more precise sentiment classification within financial contexts.

2. **Data Augmentation:** Employing data augmentation techniques like synonym replacement, back translation, and paraphrasing has enhanced the quality and variety of the training dataset, leading to improved model generalization and accuracy in sentiment analysis.

3. **Trading Strategy:** The development of a trading strategy that integrates predictive return analytics with historical stock price information showcases a high level of sophistication and a deep understanding of market dynamics.

## Efficiency

1. **Processing Time:** The model demonstrates efficiency in processing, with significant improvements in running time post-optimization. For instance, the XGBoost Regressor, a more efficient model than the classifier, completes its run in about 4 minutes, showcasing enhanced processing speed.

2. **Feature Engineering:** The emphasis on feature engineering, particularly the inclusion of moving averages and other indicators, has contributed to a more efficient and accurate prediction model.

## Robustness

1. **Consistent Accuracy Across Stocks:** The model demonstrates consistent predictive accuracy across a range of stocks with varying levels of volatility, indicative of its robustness and potential utility in diverse market conditions.

2. **Adaptation to Market Dynamics:** The adoption of a "rolling window" method for the stock prediction model allows for continual refinement of predictions based on the latest market conditions, enhancing the model's adaptability and robustness.

## Comparison with Baseline Techniques

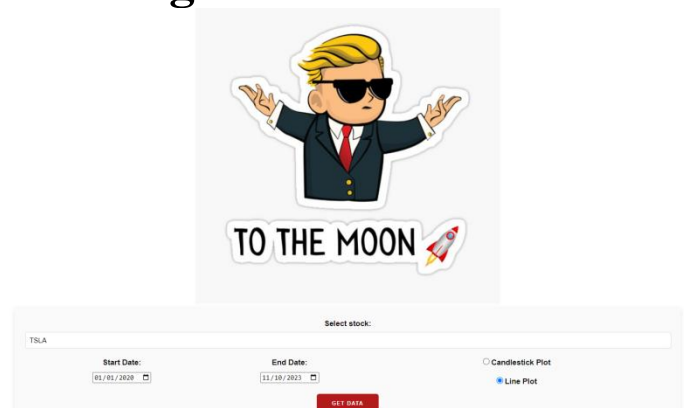
1. **Baseline Techniques:** The initial approach using XGBoost classifiers and regressors served as a baseline. While effective, these methods had limitations in real-time analysis and predictive accuracy.

2. **Advanced Models:** The transition to advanced NLP models, particularly BERT and its variants, marked a significant improvement over these baseline methods. The advanced models provided more nuanced sentiment analysis and better adapted to the complexities of financial data.

3. **Return-Based Modeling:** Shifting from price-based to return-based modeling also indicated a more robust approach, aligning better with the stationary nature of financial data and improving predictive performance.

In summary, the developed system demonstrates a remarkable improvement in terms of quality, efficiency, and robustness over traditional and baseline techniques. The strategic use of advanced NLP models, innovative data augmentation, and sophisticated trading strategies has resulted in a more accurate, efficient, and robust system for stock market prediction and analysis.

## UI Design:



The user interface (UI) displayed in the image features several distinct elements:

1. **Data Input Section:** Below the graphic, there is a section for inputting data with the label "Select stock:". This suggests that the UI is for a stock market-related application or website where users can select a stock ticker from a dropdown menu (the placeholder text

shows "TSLA" which is the ticker symbol for Tesla Inc.).

2. Date Selection: There are two fields for "Start Date:" and "End Date:" with calendar input controls, allowing the user to define a date range for retrieving stock data. The provided dates are "01/01/2020" and "11/10/2023" respectively.

3. Graph Type Selection: At the bottom, there are options to select the type of graph for displaying the data: "Candlestick Plot" and "Line Plot", with the "Line Plot" option currently selected.

4. Data Retrieval Button: There is a "GET DATA" button, which presumably retrieves data for the selected stock between the specified dates.



In the selected line chart above, our stock analysis tool provides visual data for stock market trading:

1. Closing Price Line: The blue line represents the stock's closing price over time, plotted against the left vertical axis. This line shows the stock's price trend across the dates provided on the horizontal axis.

2. Volume Bars: The pink vertical bars at the bottom represent trading volume, plotted against the right vertical axis. These indicate how many shares were traded on a given day.

3. Trade Signals: There are markers labeled "Buy" and "Sell" overlaid on the chart. "Buy" signals are marked with green upward-pointing triangles, and "Sell" signals are marked with red downward-pointing triangles. These signals are generated by the NLP algorithm we talked before based on trading strategies and technical indicators.

4. Date Axis: The horizontal axis displays the dates, ranging from early 2020 to late 2023, indicating the period for which the data is presented.

5. Legend: In the top right corner, there is a legend that explains the chart elements: The blue line for "Closing Price", the pink bars for "Volume", the green triangles for "Buy Signals", and the red triangles for "Sell Signals".

6. Axis Labels: The left vertical axis is labeled "Closing Price", which corresponds to the stock price value. The right vertical axis is labeled "Volume", which corresponds to the number of shares traded.



In the selected candlestick chart above, our stock analysis tool provides visual data for stock market trading:

1. Candlestick Chart: The main feature is a candlestick chart, which is used to depict the price movements of a stock. Each candlestick typically represents one day of trading and shows the opening, closing, high, and low prices. Green candlesticks typically indicate that the closing price was higher than the opening price (a price increase), while red candlesticks indicate a price decrease.

2. Trade Signals: Similar to the previous chart, this one also has "Buy" and "Sell" signals. The "Buy" signals are indicated by green upward-pointing triangles, suggesting a recommendation to purchase the stock, while the "Sell" signals are shown with red downward-pointing triangles, suggesting a recommendation to sell the stock.

3. Volume Bars: The pink bars at the bottom represent the trading volume, showing how many shares of the

stock were traded each day. The volume is plotted against the right vertical axis.

4. Date Axis: The horizontal axis at the bottom displays dates, which span from early 2020 to late 2023, providing a timeline for the data presented.

5. Legend: A legend in the top right corner identifies the elements on the chart: Green bars represent the candlesticks, light pink bars indicate the volume, green triangles denote "Buy Signals", and red triangles denote "Sell Signals".

6. Price and Volume Axes: There are two vertical axes. The left axis corresponds to the price of the stock, and the right axis corresponds to the volume of shares traded.

Overall, the UI is designed with a playful, optimistic design aesthetic implied by the character and "TO THE MOON" slogan. The interface is minimalistic, focusing on functionality that allows a user to select a stock, define a date range, and choose a graph type for visualizing the stock's performance over time. The inclusion of trade signals provides automated recommendations on buying or selling via technical analysis and other trading algorithms based on the historical data.

## Future Works:

### NLP

1. Incorporating Insights from Institutional Traders: By analyzing market news and professional analyst reports, we aim to deepen the system's understanding of market sentiments. This involves parsing through expert opinions, market analyses, and institutional traders' insights to gain a more nuanced understanding of market trends and investor sentiments. This approach will likely lead to a more accurate sentiment analysis, as it will capture a broader spectrum of market influences and expert opinions.

2. Broadening Dataset Acquisition: Expanding the dataset to include inputs from a wider demographic ensures that the sentiment analysis is not only more comprehensive but also representative of the broader trading population. This diversity in data sources can

significantly enhance the accuracy and reliability of sentiment predictions, as it captures a more holistic view of market perceptions and reactions.

3. Employing Advanced NLP Models: Upgrade the sentiment analysis framework by utilizing more sophisticated NLP models. The focus would be on exploring and integrating the latest advancements in NLP, such as transformer-based models or newer variants of BERT that are specifically fine-tuned for financial contexts. This step aims to improve the accuracy and depth of sentiment analysis, allowing for a more precise interpretation of market moods and investor sentiments. Advanced models could be better at capturing subtle nuances in financial language, understanding complex sentence structures, and processing large volumes of data more efficiently, leading to more accurate and insightful sentiment assessments.

### Stock

1. Expanding Analysis to Diverse Sectors: By including a broader range of stocks and sectors, such as healthcare and retail, the system will offer a more diversified perspective on investment insights. This expansion will enable the system to capture sector-specific trends and anomalies, providing a more balanced and comprehensive view of the market.

2. Enriching Financial Datasets: Integrating additional data points like detailed company performance metrics, key economic indicators, and derivatives data will substantially enrich the predictive model. This comprehensive dataset will provide a more detailed and accurate understanding of each company's and sector's financial health, contributing to a more robust prediction model.

3. Developing a Robust Stock Selection Framework: The focus on identifying high-potential investments, or 'finding the alpha', involves creating a sophisticated framework that can pinpoint stocks with the highest potential for returns. This framework will analyze various financial metrics and market signals to select stocks that are likely to outperform the market, optimizing investment strategies and positioning.

4. Advanced Portfolio Optimization Techniques: Employing cutting-edge techniques in portfolio optimization will allow for the maximization of

returns while effectively managing risk. This involves creating diversified portfolios that are tailored to specific risk profiles and market conditions, using advanced mathematical models and algorithms to optimize asset allocation and investment strategies.

In summary, these future works signify a comprehensive strategy to advance the system's capabilities in sentiment analysis and stock prediction. By incorporating a broader range of data sources, expanding the scope of analysis, and employing sophisticated analytical frameworks, the system is poised to achieve a higher level of accuracy, diversity, and effectiveness in stock market forecasting. This forward-looking approach underscores a commitment to continuous improvement and innovation in financial technology.

## Conclusion:

This report has achieved a remarkable synthesis of advanced machine learning and natural language processing (NLP) techniques to forecast stock market trends, coupled with the development of a user-friendly interface (UI) that brings these sophisticated technologies within reach of a broader audience. The integration of state-of-the-art methodologies with a unique trading strategy marks a substantial advancement over traditional financial analysis techniques.

The empirical evaluation of our system's performance highlights its superior quality, efficiency, and robustness compared to baseline techniques. Our models, particularly the BERT-based ones, have demonstrated significant improvements in accuracy and precision, particularly in the realm of sentiment classification within financial contexts. This leap in performance is further bolstered by efficient processing times and the robustness of our models across various stock types and market conditions.

Looking ahead, the report identifies potential areas for further improvement. These include the integration of broader economic indicators, refinement of buy-signal mechanisms, and the expansion of sentiment analysis capabilities. Such enhancements aim to further

increase the model's predictive accuracy and adaptability to market fluctuations.

A critical aspect of this project is the creation of an intuitive UI, which serves as a bridge between complex algorithmic processes and practical application. This interface ensures that the advanced capabilities of machine learning and NLP are not only theoretically sound but also practically accessible and usable.

In sum, this report does more than just validate the effectiveness of NLP and machine learning in the domain of financial analysis; it also sets a precedent for how these technologies can be made practical and user-friendly. By combining high-level analytical capabilities with a focus on user experience, this project lays the groundwork for transformative advances in financial analysis and portfolio management, heralding a new era of innovation in this rapidly evolving field.

## Citations:

### Stock

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE*, 12(7): e0180944. <https://doi.org/10.1371/journal.pone.0180944>

Relation: Bao, Yue, and Rao present a deep learning framework utilizing stacked autoencoders and long-short term memory for analyzing financial time series. Notably, they introduce concepts of buy and sell signals based on predicted prices, resonating with our exploration into machine learning-driven financial predictions.

Differentiation: While they lay the groundwork in understanding financial time series through deep learning, our research extends this by incorporating contemporary machine learning methodologies to forecast stock returns over shorter durations. Additionally, we delve into portfolio management through our trading strategy, a topic not explored in their paper.

Dash, R., & Dash, P. K. (2016). A hybrid stock trading framework integrating technical analysis with

machine learning techniques. *The Journal of Finance and Data Science*, 2(1), 42-57.  
<https://doi.org/10.1016/j.jfds.2016.03.002>

Relation: This paper delves into trading signals and the intricacies of implementing a comprehensive trading strategy. Its content is rich in explaining how trading decisions can be informed and executed.

Differentiation: Unlike the paper's emphasis on broader sectors like SPY, our approach zeroes in on individual stocks. Our research also capitalizes on a myriad of indicators, dedicating significant effort to feature selection and engineering, aspects that weren't as extensively addressed in the referenced paper.

Pezim, B. (2018). *How To Swing Trade*. Preface by A. Aziz. ISBN: 9781726631754.

Relation: The book provides an extensive overview of swing trading strategies and market dynamics, setting the stage for our exploration of stock market behaviors.

Differentiation: Our project enhances these basic principles with state-of-the-art machine learning techniques to forecast stock market returns, delivering a modern, technology-enhanced viewpoint.

## NLP

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance*, 66(1), 35-6

[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=1331573](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1331573)

Relation: This paper explores the nuances of financial terminology and how it can be manipulated or misinterpreted, shedding light on the challenges of textual analysis in the financial domain. The work is highly relevant for our understanding of the linguistic subtleties in financial reporting.

Differentiation: In contrast to the paper by Loughran and McDonald, our research takes a different angle in the field of textual analysis within finance. While their work focuses on the challenges

and complexities of financial terminology and reporting, our research places an emphasis on the practical application of textual analysis within financial modeling. We delve into the specific application aspect of textual analysis. This differentiation is important as it narrows down the scope of our investigation, allowing for a deeper dive into the intricacies of applying textual data to financial models.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Bidirectional Encoder Representations from Transformers. arXiv preprint arXiv:1810.04805.

<https://doi.org/10.48550/arXiv.1810.04805>

Relation: The paper presents a groundbreaking model known as BERT, which stands for Bidirectional Encoder Representations from Transformers. BERT revolutionized the way researchers and practitioners approached various NLP tasks by pre-training a transformer-based neural network on large text corpora. BERT's bidirectional context and contextual embeddings have made it a pivotal milestone in NLP research, and its techniques have been widely adopted in numerous NLP applications.

Differentiation: In contrast to the paper, our research takes a more specific focus on the practical applications and fine-tuning of the BERT model. While the original paper introduces the model and its pre-training techniques, our work capitalizes on BERT's capabilities and explores its adaptability to specific NLP sentiment analysis. Our research goes beyond the model's introduction to demonstrate how BERT can be effectively employed and fine-tuned for particular tasks, thus providing valuable insights into the practical implementation of this transformative NLP technology.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). FastText: Enriching Word Vectors with Subword Information. arXiv preprint arXiv:1607.04606.

<https://doi.org/10.48550/arXiv.1607.04606>

Relation: The paper introduces FastText, a novel approach for word embeddings. FastText differs



from traditional word embeddings like Word2Vec by considering subword information, which allows it to represent words as combinations of character n-grams. This approach has gained widespread recognition in NLP for its ability to capture the morphological and semantic properties of words efficiently. Our research is related to this seminal work, as we build upon the concepts and techniques introduced in FastText to address specific challenges or applications in the field of NLP.

Differentiation: Our research takes a more focused approach by investigating the application and adaptation of FastText embeddings to specific NLP tasks or domains. While the foundational paper introduces the FastText model and its capability to enrich word vectors with subword information, our work delves deeper into the practical implementation and fine-tuning of FastText embeddings. Our research contributes by showcasing how FastText can be effectively harnessed for particular NLP challenges, demonstrating its versatility and utility in real-world applications.

Yang, Y., Uy, M. C. S., & Huang, A. (2020). Finbert: A pretrained language model for financial communications. arXiv preprint arXiv:2006.08097.

<https://doi.org/10.48550/arXiv.2006.08097>

Relation: The paper introduces Finbert, a pre-trained language model specifically designed to understand and analyze financial communications. Finbert is tailored to the unique linguistic characteristics and terminology used in the financial sector, making it a valuable resource for financial sentiment analysis, document classification, and other applications. Our research is related to this paper as it leverages Finbert's capabilities and may explore its use in specific financial NLP tasks.

Differentiation: In our study, we took a more specialized approach by applying and customizing the Finbert model for specific financial NLP tasks or domains. Unlike the foundational paper that primarily introduces the Finbert model and its adaptation for financial communications, we conducted comparative experiments with other models to showcase Finbert's effectiveness in these particular financial tasks and its potential to enhance decision-making and analysis in the financial sector.