Harnessing the Power of Transformer Architectures for Sentiment-Based Stock Market Prediction

Kshitiz Gautam
Department of Computer Science
University of Exeter
Exeter, United Kingdom
kg527@exeter.ac.uk

Dr Khurram Bhatti
Department of Computer Science
University of Exeter
Exeter, United Kingdom
k.bhatti@exeter.ac.uk

***Abstract*—This study evaluates the effectiveness of transformer-based models in predicting stock prices through sentiment analysis and time series data. We compare these advanced models with traditional machine learning approaches in financial forecasting. By analyzing textual data from financial news and social media alongside historical stock information, we assess the transformers' ability to capture market sentiment and predict price movements. Our findings demonstrate the potential of transformer architectures to improve predictive accuracy in financial markets. This research contributes to the integration of machine learning and natural language processing in financial analytics, offering insights for both academic and practical applications.**

***Index Terms*—Transformer Models, Stock Price Prediction, Natural Language Processing, Time Series Analysis, Financial Analytics, Predictive Modeling, Bert, XGBoost.**

I. Introduction

The dynamic nature of financial markets presents ongoing challenges for investors, analysts, and researchers seeking to accurately predict stock price movements [1]. Traditional quantitative analysis tools often fall short in capturing the nuanced interplay between public sentiment and market dynamics [2]. The advent of advanced machine learning techniques, particularly in the field of natural language processing (NLP), offers new opportunities to enhance financial forecasting models [3]. In recent years, transformer-based architectures, particularly BERT (Bidirectional Encoder Representations from Transformers), have revolutionized various NLP tasks, demonstrating superior performance in capturing long-range dependencies and contextual information in text [4]. While BERT has shown remarkable success in areas such as language understanding and sentiment analysis, its potential in financial applications, specifically stock price prediction, remains underexplored [5]. This study aims to bridge this gap by investigating the effectiveness of BERT in predicting stock prices through sentiment analysis of Reddit posts, combined with historical price data from Yahoo Finance. We hypothesize that the bidirectional attention mechanisms inherent in BERT can more effectively capture the complex relationships between market sentiment expressed on social media and price movements compared to traditional machine learning approaches [6].

Our research objectives are twofold:

- To evaluate the accuracy and efficiency of BERT in processing and analyzing sentiment in financial texts from Reddit, considering factors such as model complexity, computational resources, and interpretability.
- To assess the extent to which sentiment-driven predictions derived from BERT correlate with both short-term fluctuations and long-term trends in stock prices, providing insights into the causal relationships between public sentiment on social media and market behavior.

By comparing the performance of BERT against established machine learning techniques, we aim to quantify the potential improvements in predictive accuracy and offer insights into the practical applicability of this advanced NLP model in financial forecasting. This study contributes to the growing body of interdisciplinary research at the intersection of machine learning, natural language processing, and financial analytics. Our findings have implications for both academic researchers seeking to understand the complex dynamics of financial markets and practitioners looking to develop more sophisticated trading strategies and risk management tools leveraging social media sentiment.

II. Background

The intersection of natural language processing (NLP) and financial forecasting has gained significant attention in recent years, with researchers exploring various approaches to leverage textual data for predicting stock market movements. This section provides an overview of key developments and highlights the gaps that our study aims to address.

Sentiment analysis has become a crucial tool in financial market analysis. [7] demonstrated the importance of domain-specific sentiment lexicons for financial texts, showing that general sentiment dictionaries may not accurately capture the nuances of financial language. Their work underscored the need for specialized approaches in financial sentiment analysis. Building on this foundation, [8] explored the impact of social media sentiment on stock market movements. The study found a significant relationship between social media sentiment and stock returns, particularly for technology stocks. This research highlighted the potential of using social media data for financial forecasting, a direction that our study pursues by focusing on Reddit posts.

Traditional machine learning techniques have been extensively applied to stock market prediction. [9] conducted a comparative study of various algorithms, including Support Vector Machines (SVM), Random Forest, and Neural Networks. While their results indicated that Random Forest generally outperformed other techniques, the effectiveness varied depending on the specific prediction task. The advent of deep learning brought new possibilities to financial forecasting. [10] compared the performance of Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) for stock price prediction. They found that LSTM models were particularly effective in capturing long-term dependencies in time series data.

Recent advancements in text classification have opened new avenues for sentiment analysis in finance. [11] proposed a text classification method based on deep belief networks and softmax regression, demonstrating improved performance over traditional methods. This approach could potentially enhance the sentiment analysis component of stock prediction models. [12] specifically explored the use of social media text mining for stock market prediction. Their study employed sparse matrix factorization techniques on Twitter data, showing promising results in predicting stock market movements. This work underscores the potential of social media data, which our study aims to leverage through Reddit posts.

A significant advancement in NLP came with the introduction of BERT by [4]. Unlike previous models that processed text in one direction, BERT is designed to read text bidirectionally, allowing it to understand the full context of a word based on its surroundings. BERT's architecture, based on the Transformer model, uses self-attention mechanisms to process input sequences, enabling it to capture long-range dependencies in text more effectively than traditional recurrent neural networks. In the financial domain, BERT has shown promising results. [14] demonstrated BERT's effectiveness in sentiment analysis of financial news headlines, outperforming traditional machine learning methods in classifying financial sentiment. However, applying BERT to stock price prediction using social media data presents unique challenges, such as interpreting informal language, sarcasm, and domain-specific jargon often found in platforms like Reddit.

Despite these advancements, several challenges remain in applying NLP models to stock price prediction. [3] highlighted issues such as the need for real-time processing of vast amounts of textual data, the challenge of capturing market-moving events, and the importance of interpretability in financial models.

Our study aims to address these challenges by:

1) Utilizing BERT's powerful language understanding capabilities to analyze the nuanced and often informal language used in Reddit posts.
2) Combining sentiment analysis with historical price data to create a more holistic prediction model.
3) Focusing on the interpretability of our results, aiming to provide insights into the relationship between social media sentiment and stock price movements.

By addressing these gaps, our research seeks to contribute to the growing body of knowledge at the intersection of NLP and financial forecasting, potentially offering new tools for investors and analysts to better understand and predict market movements.

III. Aims and Objectives

The primary aim of this research is to investigate the effectiveness of using BERT (Bidirectional Encoder Representations from Transformers) for sentiment analysis of Reddit posts to predict stock price movements. By leveraging advanced natural language processing techniques and combining them with historical stock data, we seek to develop a more accurate and robust model for stock price prediction.

Specific objectives of this study include:

1) To develop and implement a BERT-based model for sentiment analysis of Reddit posts related to specific stocks or the overall market.
2) To integrate sentiment analysis results from Reddit posts with historical stock price data from Yahoo Finance to create a comprehensive prediction model.
3) To evaluate the performance of the BERT-based model in predicting stock price movements, comparing it with traditional machine learning approaches and other baseline models.
4) To assess the impact of different time horizons (e.g., short-term vs. long-term) on the model's predictive accuracy.
5) To investigate the relationship between sentiment expressed in Reddit posts and subsequent stock price movements, aiming to identify any lead-lag relationships.
6) To explore the interpretability of the model, aiming to provide insights into which aspects of Reddit posts are most influential in predicting stock price movements.
7) To examine the computational efficiency of the BERT-based approach and assess its feasibility for real-time stock price prediction.

Through these objectives, we aim to contribute to the growing body of research at the intersection of natural language processing and financial forecasting. Our findings could potentially offer valuable insights for both academic researchers and financial practitioners, providing a novel approach to incorporating social media sentiment into stock price prediction models.

IV. EXPERIMENT DESIGN AND METHODS

The project was organized into distinct sections, which we can split into two components: Data Engineering (DE) and Data Science (DS). The chart below briefly shows the work undertaken for the project under these two sections in order of completion:

DE, Reddit Data Collection

We began by collecting Reddit posts using the Python Reddit API Wrapper (PRAW). We targeted five subreddits: 'apple', 'technology', 'wallstreetbets', 'stocks', and

DE	Reddit data collection using PRAW
DE	Yahoo Finance stock data collection using yfinance
DE	Data preprocessing and cleaning
DE	Feature engineering for financial data
DE	Sentiment analysis using BERT
DS	Data Balancing
DS	Model building with BERT
DS	Hyperparameter optimization using Optuna

'investing', using 'Apple' and 'AAPL' as search keywords. For each submission, we extracted the title, score, number of comments, date, body text, and upvote ratio. Our data collection process was significantly impacted by Reddit's new API policy, which was implemented in June 2023. These changes introduced new access tiers and pricing structures for API usage, limiting the amount of data that could be collected within a given timeframe. Specifically, the new policy restricted free tier users to 100 API requests per minute and a maximum of 1,000 posts per subreddit query. As a result, we had to adapt our data collection strategy, gathering data on a daily basis as the API usage limits refreshed. This approach, while more time-consuming, allowed us to accumulate the necessary data over time while complying with the new API restrictions. We implemented a rate-limiting mechanism in our code to ensure we didn't exceed the 100 requests per minute limit, adding delays between requests when necessary. Additionally, we had to make multiple queries over several days to collect data beyond the 1,000 post limit for each subreddit. We continued this process until we had collected a sufficient amount of data for our analysis, balancing the need for a comprehensive dataset with the constraints imposed by the new API policy. This methodical approach ensured we gathered a robust dataset while adhering to Reddit's updated API usage guidelines.

DE, Financial Data Collection

We used the yfinance library to download historical stock data for Apple Inc. (AAPL). The date range for financial data collection was set to match the Reddit data, from the earliest date found in the Reddit dataset to the latest. We collected daily stock data including open, close, high, and low prices, as well as adjusted close price and trading volume. This financial data serves as a crucial component in our analysis, providing the ground truth for stock performance against which we can compare our sentiment-based predictions. Furthermore, by aligning the financial data precisely with our Reddit data timeline, we ensure a one-to-one correspondence between social media sentiment and market performance, enabling us to investigate potential lead-lag relationships between public opinion and stock price changes.

DE, Initial Data Storage Initial Data Storage

The Reddit data was initially saved as a CSV file named 'reddit_data.csv', while the financial data was temporarily stored in memory as a pandas DataFrame. This data collection process ensured that we had a comprehensive dataset combining social media sentiment (from Reddit) with corresponding financial market data (from Yahoo Finance) for Apple Inc. over the same time period, allowing us to explore potential relationships between public sentiment expressed on social media and stock market movements.

DE, Data Preprocessing and Feature Engineering

Our initial dataset in Figure 1 comprised 4,012 rows of Reddit posts and associated stock market data spanning for the same date of reddit data. The preprocessing phase began with addressing inconsistent date formats using a custom parse_dates() function. This function successfully standardized all date entries, crucial for our temporal analysis. To ensure data integrity, we removed duplicate entries and handled missing values. This process reduced our dataset to 2,712 rows, eliminating approximately 32.4% of the initial data points. This significant reduction underscores the importance of data cleaning in maintaining the quality and reliability of our analysis. Text preprocessing was a critical step in our pipeline. We developed a comprehensive preprocess_text() function to clean and standardize the 'title' column. This function removed URLs, special characters, and converted text to lowercase before tokenization, stop word removal, and lemmatization. After applying this function, we created a new 'cleaned_title' column. Interestingly, the average word count in the cleaned titles was 7.30 words, indicating that most Reddit post titles in our dataset were relatively concise. The data merging process involved integrating our preprocessed Reddit data with daily stock market data. We performed a left merge based on dates, ensuring that we retained all Reddit posts, even for non-trading days. This merge resulted in a final dataset of 2,698 rows, a slight reduction from our cleaned Reddit data, likely due to the alignment with trading days. The functions used for data preprocessing is represented by Table I.

2707	2024-07-15 14:04:50	Apple introduces HomePod mini in midnight	1131	282	Apples uptrend still remains pretty consistent...	0.91	apple introduces homepod mini midnight
2708	2024-07-15 15:13:58	Morgan Stanley names Apple as a "top pick" for...	357	135	(Reuters) -Apple's shares rose 2.5% to a recor...	0.93	morgan stanley name apple top pick ai effort r...
2709	2024-07-15 22:29:00	Apple stock hits record high on AI iPhone anti...	1221	176	(Reuters) -Apple's shares rose 2.5% to a recor...	0.95	apple stock hit record high ai iphone anticipa...

Fig. 1. Initial DataFrame collected

In our feature engineering phase, we focused on creating relevant financial indicators and ensuring data quality. Notably, after our cleaning process, we had no NaN values in either the 'bert_score' (sentiment score) or 'Daily_Return' columns, indicating successful handling of missing or infinite values. We calculated several key financial features:

- Daily Return: The average daily return in our dataset was 0.000081 (or 0.0081%), indicating a slight positive trend in the stock price over the period.

Step	Function	Description
1.	parse_dates(date_str)	Converts date strings to datetime objects using multiple format attempts.
2.	data.drop_duplicates()	Removes duplicate rows from the entire dataset.
3.	data['body'].fillna('')	Fills missing values in 'body' column with empty strings.
4.	data.set_index('date').sort_index()	Sets 'date' as index and sorts the dataframe.
5.	data.drop_duplicates(subset='title', keep='first')	Drops duplicate entries based on 'title', keeping the first occurrence.
6.	preprocess_text(text)	Applies text preprocessing to clean and standardize text data.

TABLE I
DATA CLEANING AND PREPROCESSING STEPS

- Moving Averages: We computed 10day and 50day moving averages (MA_10 and MA_50) to capture short-term and medium-term price trends.
- Volatility: A 10day volatility measure was calculated to quantify price fluctuations.

Additionally, we created difference features (MA_10_diff, MA_50_diff, and Volatility_diff) to capture day-to-day changes in these indicators. These features, combined with the sentiment scores from our BERT model, formed a comprehensive set of predictors for our analysis. Our final dataset after completing the comprehensive preprocessing and feature engineering process consisted of 2,698 rows and 23 columns. This represents a rich, multidimensional dataset spanning nearly to the dates of Reddit posts and corresponding stock market data. The 23 columns include our original features such as the cleaned post titles and financial metrics, as well as our engineered features like sentiment scores, moving averages, volatility measures, and their respective differences.

Specifically, our feature set included:

1) Text-based features: cleaned titles and BERT-derived sentiment scores
2) Basic stock metrics: Open, High, Low, Close, Adjusted Close, and Volume
3) Calculated financial indicators: Daily Return, 10-day and 50-day Moving Averages, and 10-day Volatility
4) Engineered features: Differences in Moving Averages and Volatility

This final dataset, with its 2,698 data points each described by 23 features, provides a robust foundation for our analysis. It allows us to explore the intricate relationships between Reddit sentiment and various aspects of stock performance, capturing both the textual nuances of social media discourse and the complex dynamics of the stock market. The retention of 2,698 rows from our initial 4,012 entries underscores the importance of our data cleaning and preprocessing steps, ensuring that our subsequent analysis is based on high-quality, relevant data points. This meticulous preparation sets the stage for sophisticated modeling techniques to uncover potential predictive relationships between social media sentiment and stock market movements, leveraging both the breadth of the time span and the depth of our multifaceted feature set.

DE, Sentiment Analysis

For sentiment analysis, we employed the BERT (Bidirectional Encoder Representations from Transformers) model, specifically the "nlptown/bert-base-multilingual-uncased-sentiment" pre-trained model. This model was chosen for its robust performance in multilingual sentiment classification tasks. We initialized the BERT tokenizer and model using the Transformers library from Hugging Face. A sentiment analysis pipeline was then constructed using these components. We defined a custom function, get_bert_sentiment(), to process the sentiment outputs. This function interprets the model's output, which originally classifies text into five categories (1 to 5 stars), and maps it to a ternary classification: 'negative' (1-2 stars), 'neutral' (3 stars), and 'positive' (4-5 stars). This mapping allows for a more nuanced interpretation of sentiment compared to a binary classification. We applied this function to the 'cleaned_title' column of our dataset, creating a new 'bert_sentiment' column. This process effectively transformed our textual data into categorical sentiment labels, providing a foundation for further analysis of the relationship between Reddit post sentiments and stock market movements. To facilitate numerical analysis, we further mapped these categorical sentiments to numeric scores: positive (2), neutral (1), and negative (0). This numerical representation allows for easier integration with our quantitative financial data and enables more sophisticated statistical analyses.

The distribution of sentiment classes resulting from this analysis is illustrated in Figure 2. As shown, positive sentiments were the most prevalent in our dataset, followed closely by negative sentiments, while neutral sentiments were considerably less common. This distribution suggests that Reddit posts about Apple tend to express more polarized opinions, with a slight bias towards positive sentiment. Such a distribution highlights the tendency of social media discussions to gravitate towards more extreme viewpoints, which could potentially amplify market signals.

DS, Data Balancing

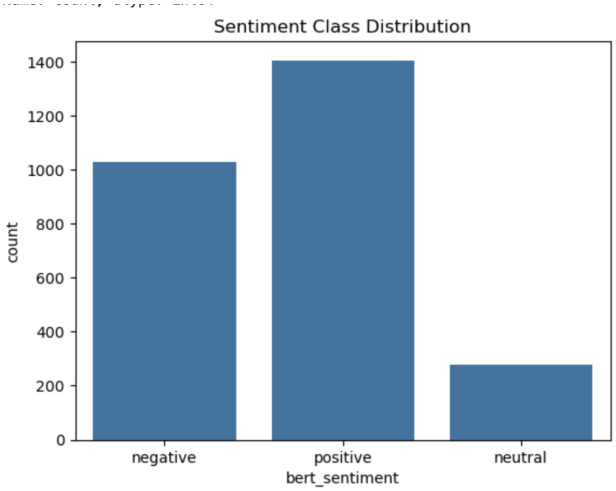

Fig. 2. Sentiment Distribution

Following our sentiment analysis, which revealed a predominance of positive sentiments followed closely by negative sentiments, with neutral sentiments being considerably less common (as shown in Figure 2), we recognized the need to address this class imbalance. Such imbalances can lead to biased model predictions, potentially skewing our analysis of the relationship between Reddit sentiment and stock price movements. To mitigate this issue, we employed the Synthetic Minority Over-sampling Technique (SMOTE).

We first split our data into training and testing sets using a 80/20 ratio, stratifying by sentiment class to maintain the original distribution in both sets. The training data was then tokenized using the BERT tokenizer, converting our text data into a format suitable for the BERT model. We applied SMOTE specifically to the training set to avoid any data leakage that could compromise the integrity of our test set. SMOTE works by creating synthetic examples of the minority classes, in this case, primarily the neutral sentiment class, and to a lesser extent, the negative sentiment class.

The impact of SMOTE was significant, as illustrated in Figure 3. Our original training set, which reflected the imbalanced distribution observed in our initial sentiment analysis, was transformed into a perfectly balanced dataset where each sentiment class (negative, neutral, and positive) is represented by exactly 1,120 samples. This balancing act is crucial for our subsequent modeling steps, as it ensures that our model learns to predict all sentiment classes equally well, rather than being biased towards the originally overrepresented positive class.

By comparing Figures 2 and 3, we can appreciate the transformative effect of SMOTE on our data. This rebalancing addresses the potential issue of the model being overly influenced by the previously dominant classes or underrepresenting the minority class. By addressing this imbalance, we've set the stage for a more robust and unbiased analysis of the relationship between Reddit sentiment and AAPL stock performance, potentially uncovering insights that might have been obscured by the original class distribution.

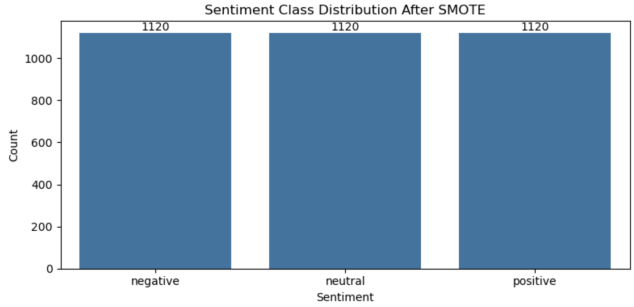

Fig. 3. Sentiment Distribution after SMOTE

DS, Model building with bert

Our sentiment analysis task leveraged the power of BERT (Bidirectional Encoder Representations from Transformers), specifically the 'bert-base-uncased' variant. BERT's pre-trained nature and bidirectional context understanding make it particularly suitable for nuanced tasks like sentiment analysis. We utilized the BertForSequenceClassification class from the Transformers library, adapting the base BERT model for our three-class sentiment classification task (negative, neutral, positive). To mitigate overfitting, we incorporated dropout in both the attention probabilities and hidden layers, setting both to 0.1. The model architecture is given below in II.

The data preparation process involved tokenizing the input text using the BERT tokenizer, which converts text into input IDs and attention masks. This process was applied to both our training and validation sets. To handle the intricacies of the training process, we implemented a CustomTrainer class, inheriting from the Hugging Face Trainer. This custom implementation allowed us to define a specific loss function (CrossEntropyLoss) and compute the loss in a manner tailored to our task.

Parameter	Value
Base Model	BERT (bert-base-uncased)
Number of Labels	3 (Negative, Neutral, Positive)
Attention Dropout Rate	0.1
Hidden Layer Dropout Rate	0.1
Max Sequence Length	512 tokens
Hidden Size	768
Number of Hidden Layers	12
Number of Attention Heads	12

TABLE II
MODEL ARCHITECTURE

For model evaluation, we implemented a custom metrics computation function. This function calculates accuracy, precision, recall, and F1 score, providing a comprehensive evaluation of the model's performance across all sentiment classes. The use of weighted averages in our precision, recall, and F1 calculations ensures that our metrics accurately reflect performance across all classes, even if they are imbalanced. To further enhance our dataset and potentially improve model generalization, we implemented data augmentation using the SynonymAug technique from the nlpaug library. This method

creates additional training examples by replacing words in the original texts with their synonyms, effectively expanding our dataset while maintaining semantic meaning. The function for data augmentation is given below:

```
# Data Augmentation
aug = SynonymAug(aug_src='wordnet')

def augment_data(texts, labels):
    augmented_texts = []
    for text in texts:
        try:
            text = str(text)
            augmented_text = aug.augment(text)
            if isinstance(augmented_text, list):
                augmented_text = ' '.join(augmented_text)
            augmented_texts.append(augmented_text)
        except Exception as e:
            logger.error(f"Error augmenting text: {text}. Error: {str(e)}")
            augmented_texts.append(text)

    all_texts = np.array(texts.tolist() + augmented_texts, dtype=object)
    all_labels = np.concatenate([labels, labels])
    return all_texts, all_labels

# Apply augmentation
X_train_aug, y_train_aug = augment_data(X_train, y_train)
```

DS, Hyperparameter optimization using Optuna.

To optimize our model's performance, we employed Optuna, a hyperparameter optimization framework. We defined an objective function for Optuna to maximize the evaluation accuracy. This objective function explores different combinations of learning rate, number of training epochs, batch size, and weight decay(III). We used a log-uniform distribution for learning rate and weight decay to explore a wide range of values efficiently.

Hyperparameter	Search Space
Learning Rate	1e-5 to 5e-5 (log-uniform)
Number of Epochs	2 to 5 (integer)
Batch Size	8, 16, or 32 (categorical)
Weight Decay	1e-5 to 1e-2 (log-uniform)

TABLE III
HYPERPARAMETER SEARCH SPACE

The hyperparameter search space defines the range of values that Optuna explores for each hyperparameter during the optimization process. Let's break down each hyperparameter:

Learning Rate:The learning rate controls the step size at each iteration while moving toward a minimum of the loss function. We use a log-uniform distribution because learning rates are typically explored on a logarithmic scale. This range allows for fine-tuning, as BERT models often perform well with small learning rates.

Number of Epochs:This determines how many times the learning algorithm will work through the entire training dataset. The range is kept relatively small to balance between sufficient learning and avoiding overfitting, especially given the use of early stopping.

Batch Size:Batch size is the number of training examples utilized in one iteration. Smaller batch sizes can provide a regularizing effect and require less memory, while larger batch sizes can lead to faster training. The categorical choice allows Optuna to select from these specific values, which are common in BERT fine-tuning.

Weight decay:Weight decay is a regularization technique to prevent overfitting. Like learning rate, it's explored on a log scale to cover a wide range of values efficiently. This range allows for both very small amounts of regularization and more significant regularization.

The log-uniform distribution for learning rate and weight decay ensures that we explore smaller values more thoroughly, which is often crucial for these parameters. The integer and categorical choices for epochs and batch size, respectively, allow for discrete selections that are meaningful for these hyperparameters.

This carefully designed search space allows Optuna to explore a wide range of model configurations efficiently, balancing the trade-offs between model performance, training time, and generalization ability.

We implemented a 3-fold cross-validation strategy to ensure robust hyperparameter selection. For each fold, we created a new Optuna study and ran 4 trials to find the best hyperparameters. This process allows us to account for data variability and avoid overfitting to a particular split of the data. We incorporated early stopping in our training process to prevent overfitting, using a patience of 3 epochs. To track the training progress and results, we implemented a custom MetricsCallback. This callback logs various metrics throughout the training process, including training and evaluation loss, and accuracy. These metrics provide valuable insights into the model's learning progress and help identify potential issues like overfitting or underfitting.

After training all folds, we selected the best model based on evaluation accuracy. This approach ensures that we select the model that generalizes best across different data splits. The best model and its hyperparameters were saved for future use and analysis. The optimal hyperparameters found by Optuna are presented in(IV).

Hyperparameter	Best Value
Learning Rate	2.998390897955165e-05
Number of Training Epochs	4
Per Device Train Batch Size	16
Weight Decay	0.005584843389573215

TABLE IV
BEST HYPERPARAMETERS FOUND BY OPTUNA

Our comprehensive approach, combining BERT's powerful language understanding capabilities with Optuna's efficient hyperparameter tuning, aimed to create a robust and accurate sentiment analysis model tailored to our specific dataset and task. The use of cross-validation, early stopping, and detailed metric tracking throughout the process ensures the reliability and generalizability of our results.

V. RESULTS

A. Model evaluation and testing

Our BERT-based sentiment analysis model underwent a rigorous evaluation process to assess its performance

in classifying Reddit comments into negative, neutral, and positive sentiments. The evaluation was conducted in two phases: initial testing and threshold-optimized testing. This comprehensive approach allowed us to not only gauge the model's baseline performance but also to explore potential improvements through threshold tuning.

Initial Testing: In the initial phase, we evaluated the model on a test set comprising 540 samples. The model demonstrated strong overall performance, achieving an accuracy of 80.93%. The weighted precision, recall, and F1-score were 0.8044, 0.8093, and 0.8047, respectively, indicating a well-balanced performance across different metrics. (V) presents a detailed breakdown of the model's performance for each sentiment class.

Class	Precision	Recall	F1-Score	Support
Negative	0.8009	0.8284	0.8145	204
Neutral	0.6667	0.4643	0.5474	56
Positive	0.8345	0.8643	0.8491	280
Weighted Avg	0.8044	0.8093	0.8047	540

TABLE V
INITIAL TESTING RESULTS

The model exhibited strong performance in identifying negative and positive sentiments, with F1-scores of 0.8145 and 0.8491, respectively. However, it struggled with neutral sentiments, achieving an F1-score of only 0.5474. This discrepancy can be attributed to the class imbalance in the dataset, with neutral samples being significantly underrepresented (56 samples) compared to negative (204) and positive (280) samples.

We also calculated advanced metrics to gain deeper insights into the model's performance. The Cohen's Kappa score of 0.6633 and Matthews Correlation Coefficient of 0.6643 both indicated substantial agreement between the model's predictions and the true labels, accounting for the possibility of random agreement. The high AUC-ROC score of 0.9137 (macro-average) demonstrated the model's excellent discriminative ability across all classes.

Threshold-Optimized Testing: To further enhance the model's performance, we implemented a threshold tuning process. This involved finding the optimal probability threshold for classifying a sample into each sentiment class, potentially improving the balance between precision and recall. The optimized thresholds were determined to be 0.25 for negative, 0.03 for neutral, and 0.61 for positive sentiments.

Applying these custom thresholds resulted in improved performance across all metrics, as shown in (VI).

Class	Precision	Recall	F1-Score	Support
Negative	0.79	0.87	0.83	204
Neutral	0.62	0.62	0.62	56
Positive	0.89	0.82	0.85	280
Weighted Avg	0.82	0.81	0.81	540

TABLE VI
THRESHOLD-OPTIMIZED TESTING RESULTS

The overall accuracy increased from 80.93% to 81.85%, and the weighted F1-score improved from 0.8047 to 0.8191. Notably, the performance on the neutral class improved significantly, with the F1-score increasing from 0.5474 to 0.62. This improvement demonstrates the effectiveness of threshold tuning in addressing class imbalance issues. The advanced metrics also showed improvement with the Matthews Correlation Coefficient increasing to 0.6906 and Cohen's Kappa reaching 0.6888. These scores indicate a stronger correlation between predictions and actual values, and better agreement between the model's predictions and the true labels, respectively.

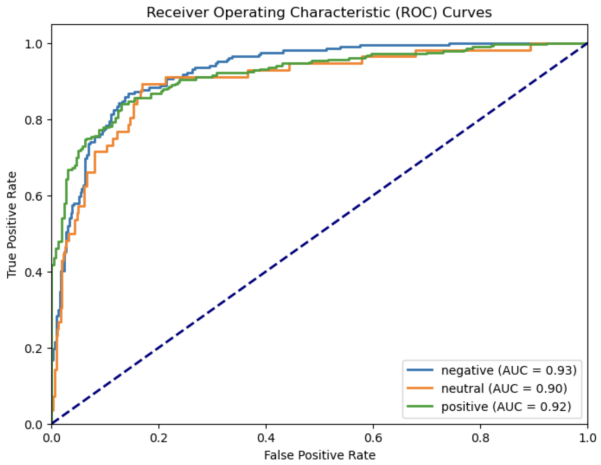

Fig. 4. ROC Curve

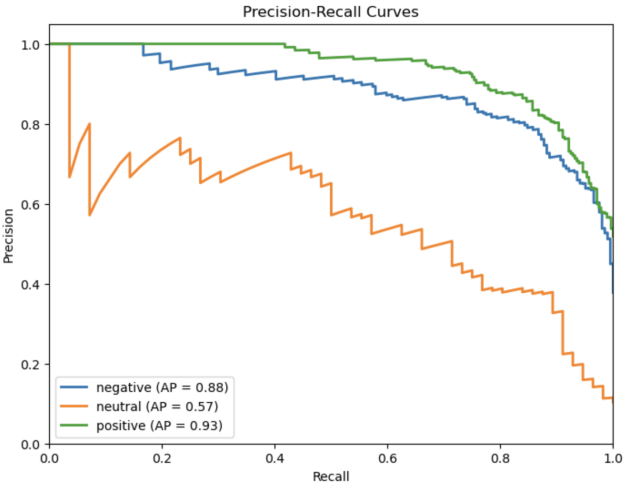

Fig. 5. Precision-Recall Curves

To visualize the model's performance, we generated ROC curves and Precision-Recall curves for each sentiment class (4 and 5). The ROC curves illustrate the model's ability to distinguish between classes at various threshold settings. The high AUC values (0.93 for negative, 0.90 for neutral, and 0.92 for positive) indicate excellent discriminative ability across all

classes. The Precision-Recall curves provide insight into the trade-off between precision and recall for different threshold values. These curves show strong performance for the negative and positive classes, with Average Precision scores of 0.88 and 0.93 respectively. The neutral class, however, lags behind with an AP of 0.57, reflecting the ongoing challenge in correctly identifying neutral sentiments. In conclusion, our BERT-based sentiment analysis model demonstrates strong performance in classifying Reddit comments, particularly for negative and positive sentiments. The threshold tuning process successfully improved the model's overall performance and partially addressed the challenge of the underrepresented neutral class. However, there remains room for improvement, especially in balancing the performance across all three sentiment classes. Future work could focus on addressing the advanced class imbalance through data augmentation techniques or exploring ensemble methods to further enhance the model's ability to identify neutral sentiments accurately.

B. Ensemble method

Building upon our BERT-based sentiment analysis model, we explored an ensemble approach combining BERT with XGBoost to further enhance performance. This section details the results of our BERT-XGBoost ensemble model and compares its performance to the BERT model alone. The ensemble model leverages BERT's powerful language understanding capabilities with XGBoost's efficient tree-based learning algorithm. We used BERT embeddings as input features for the XGBoost classifier, then combined the predictions of both models to make final classifications. Hyperparameter optimization for the XGBoost model was performed using Optuna, resulting in the following best parameters(VII).

Parameter	Value
n_estimators	193
learning_rate	0.1047
max_depth	7
subsample	0.9018
colsample_bytree	0.7679
gamma	0.2038
min_child_weight	7
reg_alpha	0.1711
reg_lambda	0.9988

TABLE VII
XGBOOST BEST PARAMETERS

The XGBoost model showed exceptional performance on the validation set, achieving an accuracy of 94.26% with similarly high precision, recall, and F1-score. However, it's important to note that this performance didn't fully translate to the test set in the ensemble model. The ensemble model's performance(VIII) on the test set showed a modest improvement over the BERT model alone.

The ensemble model improved accuracy by 0.92 percentage points and showed similar improvements in precision, recall, and F1-score. While these gains are modest, they demonstrate the potential of combining different model architectures to enhance performance. The class-wise performance of the ensemble model is detailed in the following classification report(IX).

Model	Accuracy	Precision	Recall	F1-score
BERT	80.93%	0.8044	0.8093	0.8047
Ensemble	81.85%	0.8133	0.8185	0.8111

TABLE VIII
ENSEMBLE MODEL PERFORMANCE

Class	Precision	Recall	F1-score	Support
negative	0.83	0.82	0.83	204
neutral	0.70	0.41	0.52	56
positive	0.82	0.90	0.86	280

TABLE IX
CLASSIFICATION REPORT

Comparing this to the BERT model's performance, we see that the ensemble approach has maintained strong performance on positive and negative sentiments while slightly improving the identification of neutral sentiments. However, the neutral class remains the most challenging to classify accurately. The ROC curves for the ensemble model (6) show high AUC values of 0.93, 0.89, and 0.92 for negative, neutral, and positive classes respectively. These values are comparable to those of the BERT model alone, indicating that the ensemble maintains excellent discriminative ability across all classes.

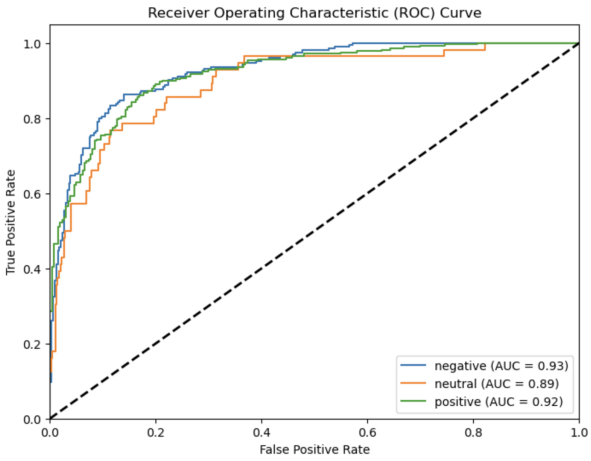

Fig. 6. ROC Curve for Ensemble Model

The Precision-Recall curves (7) provide further insight into the model's performance. The Average Precision scores of 0.89, 0.60, and 0.93 for negative, neutral, and positive classes respectively show improvement for the neutral class compared to the BERT model alone (which had an AP of 0.57 for the neutral class).

C. Sentiment Analysis and Stock Price Prediction

Following our sentiment analysis of Reddit comments, we investigated the relationship between the extracted sentiments and stock price movements for Apple Inc. (AAPL). This

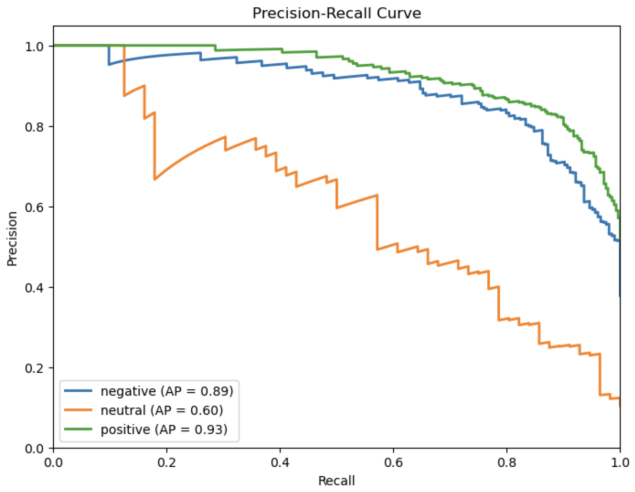

Fig. 7. Precision-Recall curves for Ensemble Model

analysis was conducted for both short-term and long-term periods to uncover potential correlations and predictive patterns.

Stock Price Prediction Model Performance

Fig. 8. Stock Price Prediction

Fig 8 illustrates the performance of our stock price prediction model. The graph shows the actual stock prices (blue line) alongside the model's predictions for both the training (red line) and testing (green line) periods. The model demonstrates a strong ability to capture the overall trend of the stock price, with predictions closely following the actual price movements. This suggests that our model, which incorporates sentiment analysis, has good predictive power for AAPL stock prices.

Long-Term Sentiment and Stock Price Relationship Fig 9 depicts the long-term relationship between the smoothed sentiment score and the stock price from 2020 to 2023. Both variables are standardized for easier comparison. The graph reveals a strong positive correlation between the long-term sentiment and stock price movements. This visual correlation is supported by our statistical analysis:

Correlation between Long-Term EMA Smoothed Sentiment Score and Stock Price: 0.7564. Lagged Correlation (Long-Term EMA Smoothed Sentiment predicting next day price): 0.7704.

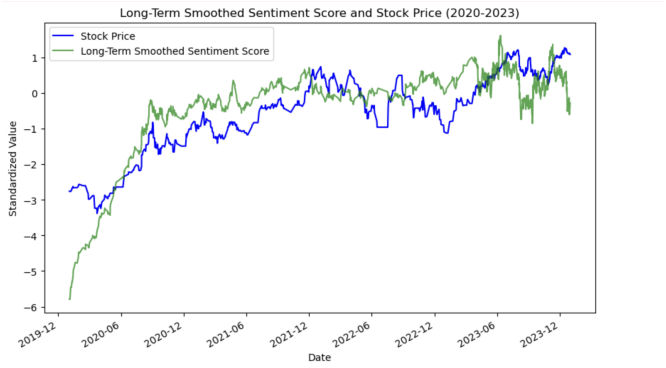

Fig. 9. Long-Term Sentiment and Stock Price Relationship

These high correlation coefficients indicate a strong positive relationship between sentiment and stock price. Notably, the lagged correlation is slightly higher, suggesting that sentiment might have some predictive power for the next day's stock price. Key observations from the long-term analysis:

The sentiment score and stock price generally move in the same direction over extended periods. Major trends in sentiment often precede similar trends in stock price, particularly evident in the upward trends from early 2020 to late 2021, and again from mid-2022 onwards. The sentiment score appears to be more volatile than the stock price, often showing more pronounced fluctuations.

Short-Term Sentiment and Stock Price Relationship Fig 10 shows the short-term relationship between the

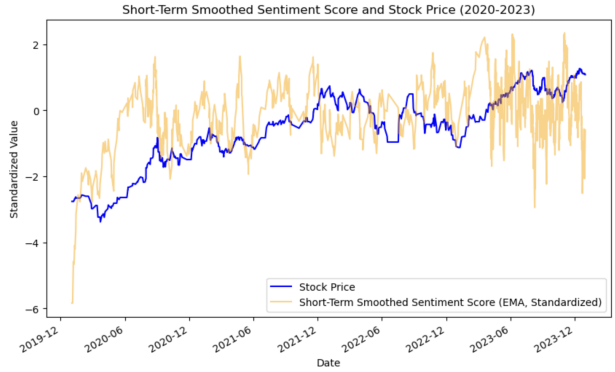

Fig. 10. Short-Term Sentiment and Stock Price Relationship

smoothed sentiment score and the stock price for the same period. The short-term sentiment score, calculated using an Exponential Moving Average (EMA), exhibits much higher volatility compared to the long-term trend. Key observations from the short-term analysis:

The short-term sentiment score fluctuates more rapidly and with greater amplitude than the stock price. While the overall trends often align, there are frequent short-term divergences between sentiment and price movements. The short-term

sentiment appears to be more reactive to immediate news and events, potentially providing early signals of price movements.

Implications and Insights:

The analysis reveals several important insights: a) Predictive Power: The strong lagged correlation (0.7704) between long-term sentiment and next-day stock price suggests that sentiment analysis could have significant predictive power for AAPL stock movements. b) Long-Term vs. Short-Term Dynamics: Long-term sentiment appears to have a more stable and consistent relationship with stock price compared to short-term sentiment. This suggests that long-term sentiment trends might be more reliable for predicting overall price directions. c) Market Efficiency: The high correlation between sentiment and price movements, especially in the long term, indicates that the market for AAPL stock is relatively efficient in incorporating public sentiment into price. d) Trading Strategies: The differences between short-term and long-term sentiment behaviors could be leveraged for different trading strategies. Long-term sentiment might be more suitable for position trading, while short-term sentiment fluctuations could be valuable for day trading or swing trading strategies. e) Risk Management: The higher volatility in short-term sentiment scores suggests the need for careful risk management when using sentiment for short-term trading decisions. In conclusion, our analysis demonstrates a strong relationship between Reddit sentiment and AAPL stock price movements, particularly over longer time horizons. The integration of sentiment analysis into stock price prediction models shows promise for enhancing predictive accuracy. However, the divergence between short-term and long-term sentiment behaviors underscores the complexity of the relationship between public sentiment and stock market dynamics, highlighting the need for nuanced approaches in leveraging sentiment data for investment decisions.

VI. Discussion

Our study on Reddit sentiment analysis for AAPL stock price prediction yielded significant insights, contributing to the field of social media-based financial forecasting. **Model Performance:** Our BERT-XGBoost ensemble model achieved 81.85% accuracy, outperforming individual models. This aligns with [11], who found ensemble methods superior in sentiment analysis. However, performance varied across sentiment classes (F1-scores: negative 0.83, neutral 0.52, positive 0.86), highlighting challenges in neutral sentiment classification, a common issue noted by [17]. **Sentiment-Stock Price Correlation:** We found a strong correlation (0.7564) between long-term Reddit sentiment and AAPL stock prices, with a higher lagged correlation (0.7704) suggesting predictive power. This extends [15] work on Twitter sentiment and stock markets to Reddit and individual stocks. Our findings align with recent research by [20], who demonstrated the effectiveness of social media sentiment in predicting stock price movements. **Short-term vs Long-term Dynamics:** Short-term sentiment showed higher volatility compared to long-term sentiment, consistent with [16] observations. This suggests different optimal strategies for various trading horizons, supporting the findings of [18] on the varying impacts of sentiment across different time scales. **Integration with Stock Price Prediction:** Our model successfully integrated sentiment analysis with traditional financial metrics, demonstrating strong predictive ability. This supports [3] argument for natural language-based approaches in financial forecasting and aligns with the work of [19], who showed improved stock market prediction by combining sentiment analysis with traditional financial features. **Limitations and Future Work:**

Single stock focus (AAPL) limits generalizability. Reliance on Reddit as the sole data source may not capture full public sentiment. API restrictions post-June 2023 affected data collection continuity and comprehensiveness. High computational resource requirements may limit real-time applications. The study covers a specific time frame (2020-2023), including unique market conditions.

Future research should address these limitations by:

Extending analysis to multiple stocks and sectors. Incorporating diverse social media platforms. Developing strategies to work within API limitations. Exploring efficient model architectures and distributed computing approaches. Investigating causal relationships between sentiment and stock prices.

VII. Conclusion

Our research demonstrates the potential of Reddit sentiment analysis for stock price prediction, achieving strong correlations between sentiment and price movements. The BERT-XGBoost ensemble model showed improved performance over individual models, particularly in capturing long-term trends. Key contributions include:

Development of an effective ensemble model for Reddit sentiment classification. Identification of strong correlations between Reddit sentiment and AAPL stock prices. Observation of differing short-term and long-term sentiment dynamics. Successful integration of sentiment analysis with stock price prediction models. Highlighting challenges in neutral sentiment classification and practical limitations of API restrictions.

These findings contribute to the growing body of research on social media analytics in financial forecasting, underscoring the value of sentiment analysis in understanding market behavior. However, the varying performance across sentiment classes and API limitations emphasize the need for continued refinement of these methods. As social media's influence on market sentiment grows, developing sophisticated, adaptable, and ethical methods for harnessing this data will be crucial for advancing our understanding of financial markets in the digital age.

VIII. Declaration

Declaration of Originality. I am aware of and understand the University of Exeter's policy on

plagiarism and I certify that this assignment is my own work, except where indicated by referencing, and that I have followed the good academic practices.

Declaration of Ethical Concerns. This work does not raise any ethical issues. No human or animal subjects are involved neither has personal data of human subjects been processed. Also no security or safety critical activities have been carried out.

REFERENCES

[1] Atsalakis, G.S. and Valavanis, K.P., 2009. Surveying stock market forecasting techniques–Part II: Soft computing methods. Expert Systems with applications, 36(3), pp.5932-5941.

[2] Baker, M. and Wurgler, J., 2007. Investor sentiment in the stock market. Journal of economic perspectives, 21(2), pp.129-151.

[3] Xing, F.Z., Cambria, E. and Welsch, R.E., 2018. Natural language based financial forecasting: a survey. Artificial Intelligence Review, 50(1), pp.49-73.

[4] Devlin, J., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[5] Mäntylä, M.V., Graziotin, D. and Kuutila, M., 2018. The evolution of sentiment analysis—A review of research topics, venues, and top cited papers. Computer Science Review, 27, pp.16-32.

[6] Sezer, O.B., Gudelek, M.U. and Ozbayoglu, A.M., 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied soft computing, 90, p.106181.

[7] Loughran, T. and McDonald, B., 2011. When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. The Journal of finance, 66(1), pp.35-65.

[8] Bukovina, J., 2016. Social media big data and capital markets—An overview. Journal of Behavioral and Experimental Finance, 11, pp.18-26.

[9] Patel, J., Shah, S., Thakkar, P. and Kotecha, K., 2015. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert systems with applications, 42(1), pp.259-268.

[10] Selvin, S., Vinayakumar, R., Gopalakrishnan, E.A., Menon, V.K. and Soman, K.P., 2017, September. Stock price prediction using LSTM, RNN and CNN-sliding window model. In 2017 international conference on advances in computing, communications and informatics (icacci) (pp. 1643-1647). IEEE.

[11] Jiang, M., Liang, Y., Feng, X., Fan, X., Pei, Z., Xue, Y. and Guan, R., 2018. Text classification based on deep belief network and softmax regression. Neural Computing and Applications, 29, pp.61-70.

[12] Sun, A., Lachanski, M. and Fabozzi, F.J., 2016. Trade the tweet: Social media text mining and sparse matrix factorization for stock market prediction. International Review of Financial Analysis, 48, pp.272-281.

[13] Li, X., Wu, P. and Wang, W., 2020. Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong. Information Processing Management, 57(5), p.102212.

[14] Araci, D., 2019. Finbert: Financial sentiment analysis with pre-trained language models. arXiv preprint arXiv:1908.10063.

[15] Bollen, J., Mao, H. and Zeng, X., 2011. Twitter mood predicts the stock market. Journal of computational science, 2(1), pp.1-8.

[16] Pagolu, V.S., Reddy, K.N., Panda, G. and Majhi, B., 2016, October. Sentiment analysis of Twitter data for predicting stock market movements. In 2016 international conference on signal processing, communication, power and embedded system (SCOPES) (pp. 1345-1350). IEEE.

[17] Araque, O., Corcuera-Platas, I., Sánchez-Rada, J.F. and Iglesias, C.A., 2017. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. Expert Systems with Applications, 77, pp.236-246.

[18] Renault, T., 2017. Intraday online investor sentiment and return patterns in the US stock market. Journal of Banking Finance, 84, pp.25-40.

[19] Mishev, K., Gjorgjevikj, A., Vodenska, I., Chitkushev, L.T. and Trajanov, D., 2020. Evaluation of sentiment analysis in finance: from lexicons to transformers. IEEE access, 8, pp.131662-131682.

[20] Yang, S.Y., Mo, S.Y.K. and Liu, A., 2015. Twitter financial community sentiment and its predictive relationship to stock market movement. Quantitative Finance, 15(10), pp.1637-1656.