Comparative Analysis of TSLA Stock Price Prediction Models with Twitter Text Data

Team: Andrew Krumenacker, Rafiz Sadique, Evan Qiang
Mentor: Jiahao Huang

Contents

1 Abstract

For this project, we study the problem of predicting Tesla stock price movements using Twitter data, which is important for investors and financial analysts seeking to leverage social media insights for market predictions and its effects on the economy. Our primary contribution is to develop a model that directly uses raw text data from tweets to predict stock price increases or decreases, bypassing traditional sentiment labeling. We aim to create a more robust prediction model by incorporating the actual dates of tweets and aligning them with stock price movements over specified periods (weekly, monthly, or quarterly). To do this, we evaluate the performance of various NLP techniques and supervised learning models to determine the most effective combination for this task.

2 Introduction

In the digital age, it has become increasingly important to consider social media as a medium that can affect stock performance. Traditionally, researchers have assigned sentiment labels to text data and assumed a relationship between sentiment and stock price movement. However, this approach can be problematic due to the noisy and ambiguous nature of textual data, especially on social media platforms like Twitter. Sentiment analysis models often fail to capture more nuanced language, and condensing tweets into labels can overlook critical factors like geopolitical events and investor emotions. Stock prices are inherently volatile, so having a relatively accurate model that can predict them can be advantageous in the financial sector.

3 Background

As mentioned, sentiment analysis is a popular method to model text data due to its interpretability and accuracy. Research groups that have tackled the same problem as us incorporate sentiment into models such as LSTM and BERT to enhance performance. However, it is still difficult to predict the closing price of a stock compared to a simpler model or random chance due to the variability of the stock market and the source of the data. We aim to address both of these issues by focusing on long-term changes in stock price via a custom output label and aggregating data from Twitter across multiple years.

Our key prior works are:

- Weng, Xiaojian, Xudong Lin, and Shuaibin Zhao. ”Stock price prediction based on LSTM and Bert.” 2022 International Conference on Machine Learning and Cybernetics (ICMLC). IEEE, 2022.

- Swathi, T., N. Kasiviswanath, and A. Ananda Rao. ”An optimal deep learning-based LSTM for stock price prediction using twitter sentiment analysis.” Applied Intelligence 52.12 (2022): 13675-13688.

4 Summary of Our Contributions

4.1 Data

We created a new dataset by combining extracted from tweets from both Twitter's API and Kaggle with the percent change in actual stock price data during different periods. There has been much discussion and controversy regarding Elon Musk's takeover of Twitter, including the influx of bot accounts and spam tweets. Because our models depend on the quality of the data, we wanted to see if results would differ between the datasets. For all datasets, we defined and calculated a new output label that tracked price changes over a specified time period.

4.2 Analysis

Our project uses a range of NLP techniques, including Bag of Words, Word2Vec, and sentiment analysis. We wanted to assess the performance of these methods against various models such as logistic regression (our baseline), LSTM, and Google's DistilBERT. Ultimately, our goal was to evaluate and qualitatively justify which NLP approach and machine learning model combination would yield the most predictive power.

5 Detailed Description of Contributions

5.1 Methods

5.1.1 Project Overview - Data Gathering

For this project, our training data was obtained from a dataset freely available on Kaggle consisting of 3 million tweets from 2015-2019 mentioning the top technology companies. The dataset can be found here. After honing this dataset down to tweets only regarding Tesla, it contained 1 million tweets. During training, we utilized 40,000 tweets that were randomly sampled, and 10,000 for testing given our computational restraints using Google Colab.

After this, we used the yfinance library to gather the stock price data for Tesla during this time frame and map each tweet to its respective label. As mentioned above, we created three different training datasets consisting of the same tweets with different labels depending on the desired time period for measuring stock price movement (weekly, monthly, or quarterly). For each batch, we determined stock price movements by comparing the price of the current batch dates to the price of the subsequent batch dates. For instance, for our training dataset measuring quarterly percent change in Tesla's stock price, if in Q2: 2015 Tesla had a negative quarter, then we labeled all tweets in the Q1:2015 with a 0. On the other hand, if Tesla's stock price had increased during this period, then the tweets of the previous period would have been labeled with a 1. Ultimately, the goal of this is to measure the potential impact of tweets at different time frequencies on the stock price movement. Looking at the data distribution provided in the Appendix, it is clear that these three different labeling schemes result in training datasets with different labeling distributions, which is what we set out to measure.

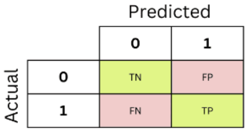
We also gathered 7579 tweets posted from May 3rd to May 10th via Twitter's API; the API does not provide widely available access to tweets more than a week old. The API also limits the amount of tweets one can retrieve, so we tried to reduce the chance of getting low-quality tweets (such as bot tweets) by being as specific with our keywords as possible. Tweets in the Kaggle dataset were included if they contained a company's ticker symbol; thus, we selected for English, non-retweet tweets that contained "TSLA" along with one of Tesla's products (a car model, Cybertruck, or Powerwall). This resulted in approximately 750 tweets. We then expanded our search by searching for tweets containing both "TSLA" and "Tesla." Due to the short timespan of the tweets, we took the average of the stock price in that week and assigned 1 to tweets that were posted on days with a higher than average stock price and 0 otherwise.

5.1.2 Data Preprocessing

After obtaining the tweets that we planned to use for the model training and evaluation with evenly distributed labels, we performed several cleaning processes on each tweet to optimize the performance of the logistic regression, LSTM, and DistilBERT models. Since tweets are often very messy, this process is imperative for each of the models as it removes noise from the training data, allowing the models to focus on the essential linguistic features. By standardizing the text, eliminating irrelevant elements like @mentions, URLs, and non-alphanumeric characters, and ensuring consistent formatting through lowercasing and space removal, we enhance the models' ability to accurately capture and learn from the underlying patterns in the data. This preparation step is crucial for improving the overall effectiveness and accuracy of our predictive models.

5.2 Experiments and Results

Our project aims to analyze the relationship between Twitter data and Tesla's stock price movements by comparing logistic regression, LSTM, and DistilBERT models. We hypothesize that more sophisticated models (LSTM, DistilBERT) will outperform logistic regression in predicting stock price changes without relying on sentiment analysis, and that different labeling schemes (weekly, monthly, quarterly) will yield varying predictive accuracies. Using a dataset of 40,000 tweets, we evaluate these models based on binary cross entropy loss, accuracy, confusion matrix, and F1 score. Our design decisions include using Bag of Words for logistic regression and leveraging word embeddings for LSTM and DistilBERT. By directly relating tweets to stock price changes, we aim to determine the most predictive time frame and the best-performing model for this task.



		Predicted	
		0	1
Actual	0	TN	FP
	1	FN	TP

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$L_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(Y_i \cdot \log \hat{Y}_i + (1 - Y_i) \cdot \log(1 - \hat{Y}_i))$$

5.2.1 Logistic Regression (Baseline Model) and Sentiment Analysis

To compare the performance of our LSTMs and DistilBERT models, we used logistic regression as a baseline and analyzed two different text representations: bag-of-words and sentiment analysis. We chose bag-of-words since it offered the simplest view of our dataset, and we chose sentiment analysis to compare our results with the work of other researchers.

For the Kaggle dataset, the model was not that accurate for any time period, and the plots of the most significant words do not seem to provide much additional information. For instance, "january" and "private" are strongly negatively correlated to stock price in the monthly case but are strongly positively correlated to stock price in the quarterly case. This result is not surprising considering the large size of the dataset, the fact that it encompasses multiple years worth of tweets, and the difficulty of predicting stock prices in general. On the other hand, logistic regression performed surprisingly well on the API dataset with F1 scores of both labels above 0.65. We believe that the main difference is the shorter timespan; since the dataset is restricted to tweets posted in a weeklong interval, current events will heavily affect both stock price and text information. For instance, bp is probably negatively correlated due to recent news that BP will take over Tesla's supercharging sites, while optimus is probably positively correlated due to a recent video posted about the robot. It is important to note that in our analysis below, we do not train our model on 2024 tweets, so we should not necessarily expect such a high accuracy for BERT and LSTM.

For sentiment analysis, we used a pre-trained model from Hugging Face, twitter-roberta-base-sentiment-latest, which returns a label "positive", "negative", or "neutral" and a confidence score for every tweet. To get a 0 or 1 label, we used the sentiment data as input to logistic regression, where our variables were I(positive), I(neutral), I(negative), I(positive)*score, I(neutral)*score, and I(negative)*score, where I is an indicator variable. Unlike the bag-of-words case, this method was not successful on any dataset or time period. We believe this is a result of the assignment of one label to an entire tweet, especially for the API dataset, which might miss out on keywords such as "optimus" and "bp" that the bag-of-words model captures.

5.2.2 LSTM Model

The second model we employed in our analysis of the Twitter data for Tesla stock price prediction was the Long Short-Term Memory (LSTM) neural network. The LSTM model's design featured a dual-layer architecture with 64 units each and incorporated dropout layers to mitigate overfitting. We employed Word2Vec-generated embeddings to convert tweets into numerical vectors, providing a more nuanced input than traditional text representation methods.

The LSTM model demonstrated accuracies of around 50% across all labeling schemes—weekly, monthly, and quarterly—suggesting performance close to random guessing. However, a notable aspect of the model's behavior was its significantly high recall, particularly in the weekly dataset where it reached 84.38%. This high recall indicates the model's strong ability to identify tweets that correlate with downward movements in Tesla's stock price, which is especially useful for investors looking to avoid potential losses. While this sensitivity to negative price movements is advantageous in risk management, it also led to a high rate of false positives, as reflected in the lower precision scores. The performance on the `curr_tweets` dataset mirrored these trends, with the weekly model showing a high recall of 80.81%.

5.2.3 DistilBERT Model

The third model that we decided to analyze on our Twitter data was Google's DistilBERT model (66 million parameters; 40% less than base BERT). Here, we utilized the pre-trained embedding library for tokenizing with a max token length of 128. Additionally, due to the limited computational abilities of Colab and the long training time of large models like DistilBERT, we performed our analysis after 5 epochs of fine-tuning, even though this was likely prior to the model converging. Finally, for this model, we used a training set of size 30,000, validation of size 10,000, and testing (2019 and May 2024, separately) of 10,000 each.

Overall, the results that we received from the DistilBERT were quite interesting, but still subpar to what we had hoped to achieve prior to experimentation. In the weekly model, the validation accuracy that we achieved after 5 epochs of fine-tuning was 54.55 %, which is not much better than our baseline models. The best performance achieved on the validation set occurred with the monthly percent change labeling model, in which we had achieved an accuracy score of 56.19 % once training was stopped. Finally, with the quarterly model, our results were consistently the lowest across several different initialization points, with the validation and testing accuracy hovering right around 50 %. While these results were not exactly as we had hoped prior to our experimental analysis, it is interesting to see the difference between the confusion matrices of each of the different models. For example, within the weekly DistilBERT model, we see that despite having uniformly distributed labels, the model tends to predict "0" labels at a much higher frequency than "1" labels, resulting in a higher recall and lower precision. One hypothesis that we had for this was that significant events (e.g., earnings reports, product launches) could have different impacts depending on the time frame. Weekly labeling might capture the immediate impact of such events and subsequently output "0" labels as investors might be more quick to voice opinions on social media when a stock underperforms as opposed to overperforming. On the other hand, quarterly labeling might smooth out these effects, which is why it would then show a much more evenly distributed confusion matrix with close precision and recall measures.

6 Compute/Other Resources Used

In order to complete this NLP project, we utilized a variety of resources for things like data gathering, data preprocessing, and model training/evaluation. First, our primary dataset containing tweets from 2015-2019 was obtained from Kaggle, and our test set of tweets from May 2024 was obtained using Twitter's API. Additionally, once we had the text data, we deployed the yfinance library to map the changes in stock prices to the tweets at different specific dates. Finally, for model training and evaluation, we utilized scikit-learn's libraries as well as Google's DistilBERT model, which is a smaller, more refined version of the DistilBERT model. All training was complete on Google Colab's free-to-access T4 GPU.

7 Conclusions

Our investigation revealed that while traditional sentiment analysis may fall short due to the complex and ambiguous nature of social media text, advanced NLP models like LSTM and DistilBERT can provide insights into stock price trends. One of the major takeaways was that traditional accuracy metrics are not the sole indicator of a model's value in real-world applications. While some of our models showed accuracies around 50 %, similar to random guessing, they provided valuable insights into potential downturns in stock prices, particularly useful for risk-averse strategies. This underlines the importance of balancing sensitivity and specificity in predictive models, especially in dynamic and noisy environments like social media.

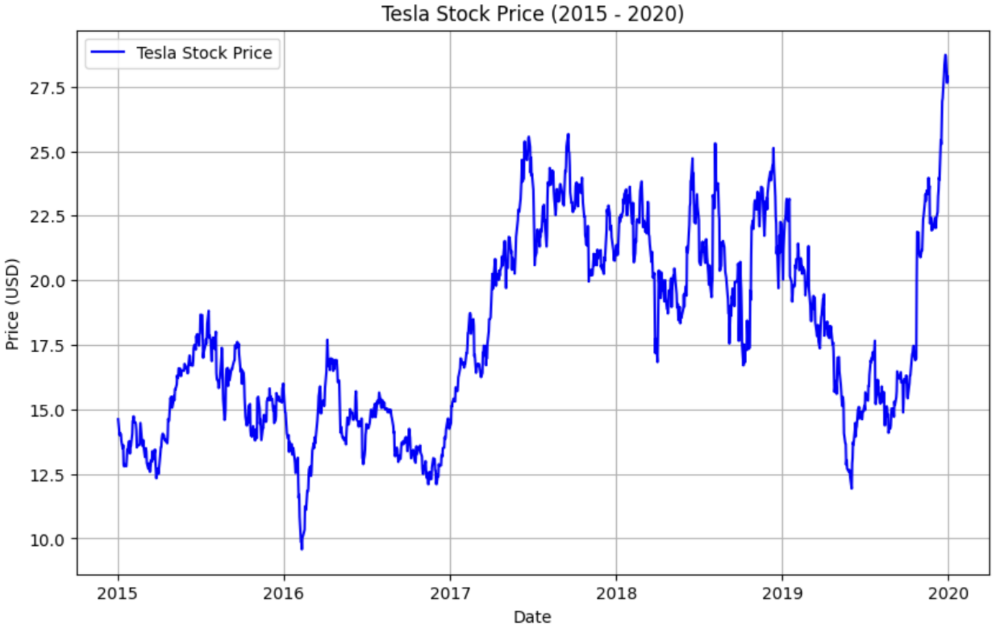
Throughout the project, we encountered several challenges, including the inherent messiness of Twitter data and the computational limitations of using Google Colab. These obstacles required us to streamline our data preprocessing and adapt our model training processes. Feedback from our mentor and peers was invaluable in refining our approach, particularly in enhancing the data cleaning steps and adjusting model parameters for better performance.

Looking to the future, there is significant room for improvement in this area of research. Enhancing precision without compromising recall could involve integrating more sophisticated NLP techniques or exploring hybrid models that combine the strengths of ML approaches. Ethical considerations include ensuring that our models do not inadvertently amplify biases present in social media data and avoiding excessive profits for companies that can increase the growing wealth gap in the US. As we continue to refine these predictive models, their potential to influence investment strategies and broader economic understanding will only grow, making this a promising field for ongoing exploration and development.

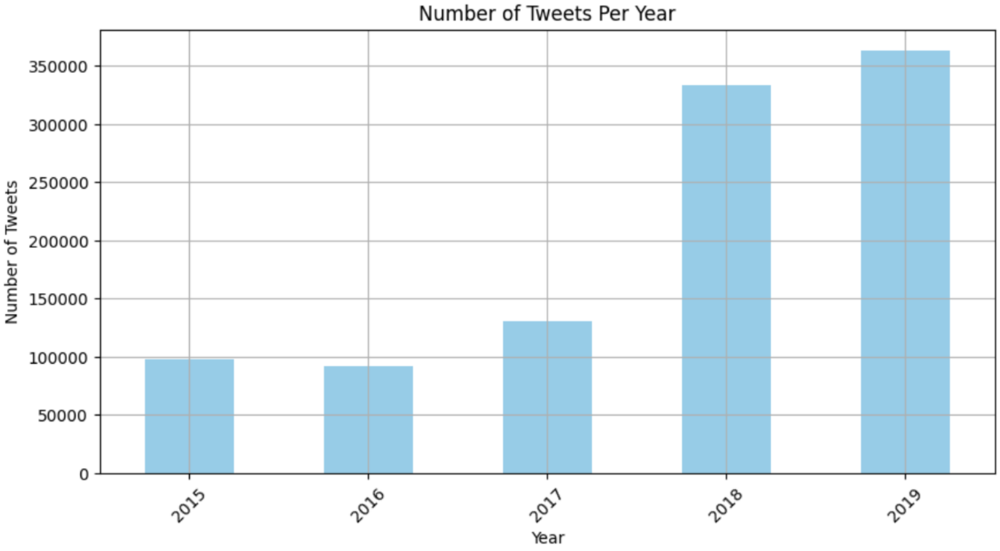
8 Appendix

8.1 Kaggle Data Preprocessing and Background Information

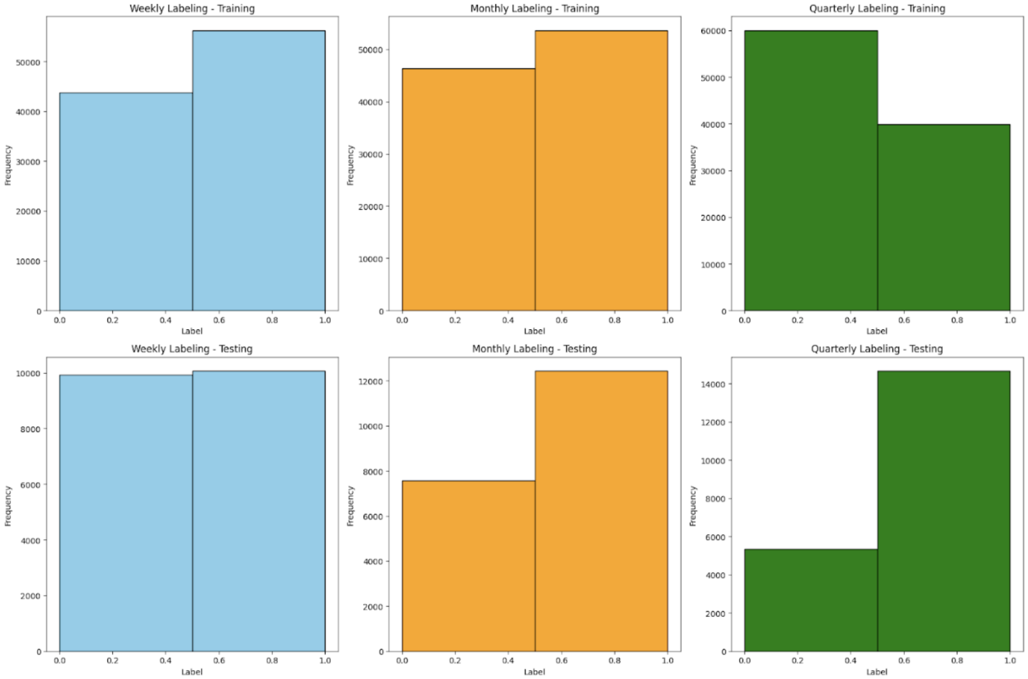
8.1.1 Tesla Stock Price (2015-2020)



8.1.2 Raw Tweets Per Year (Training Data)

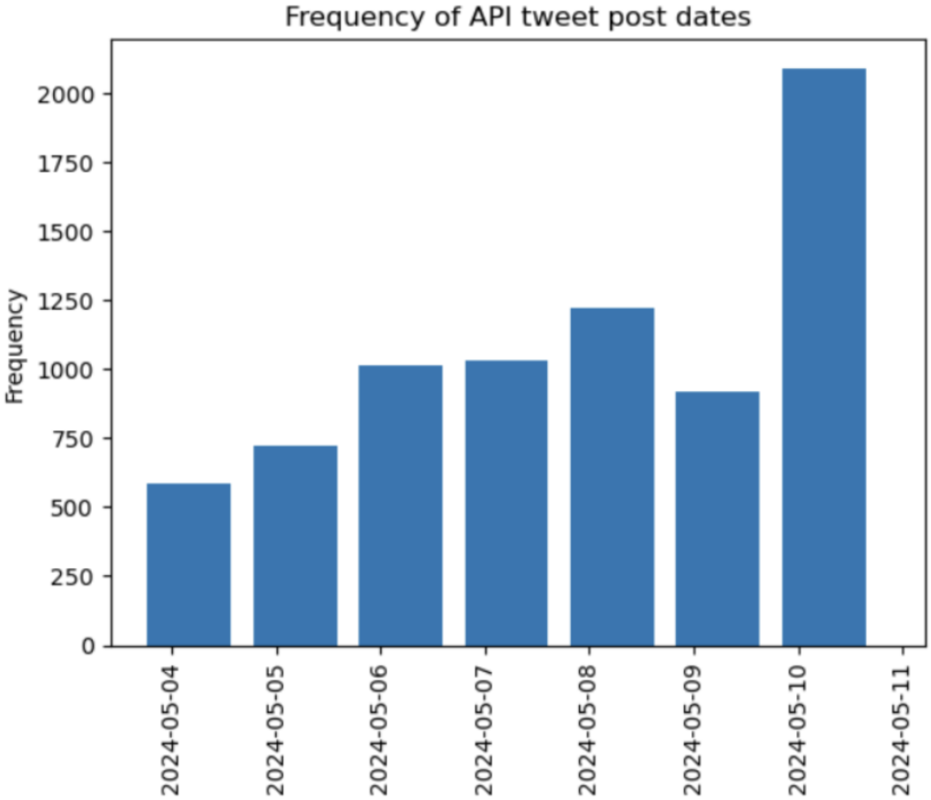


8.1.3 Training Data Labeling Distribution

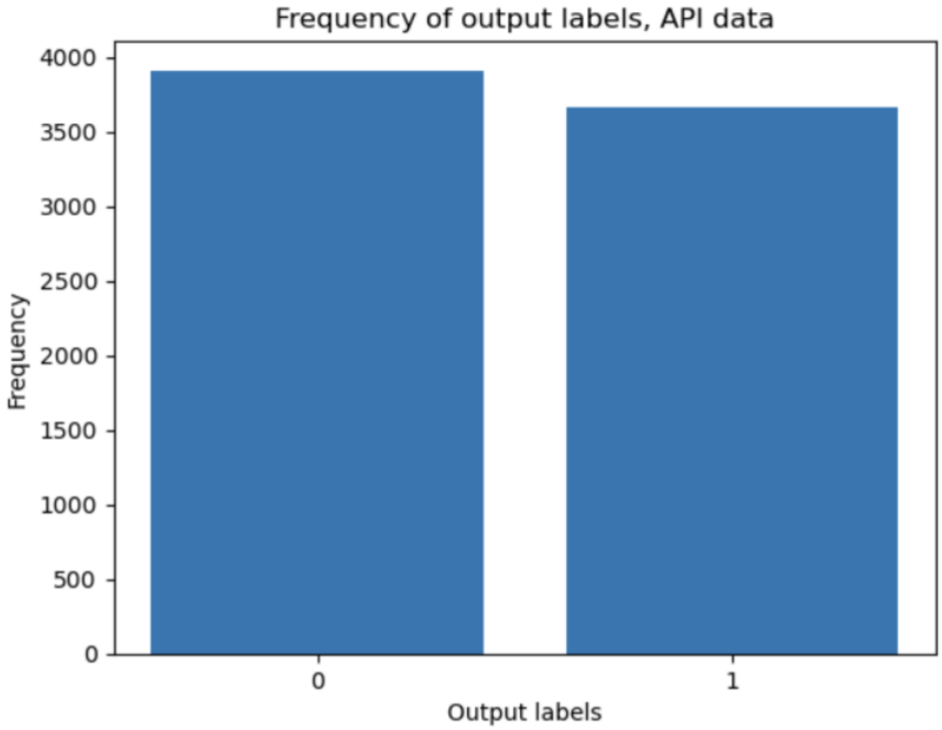


8.2 API Data EDA and Comparison with Kaggle

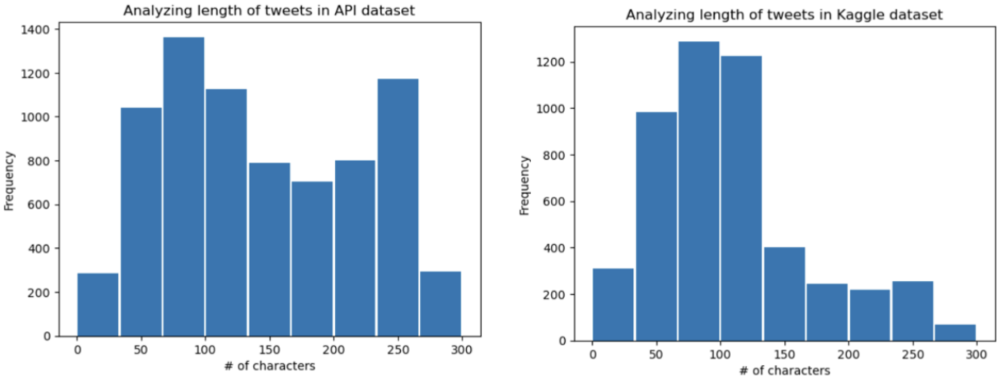
8.2.1 Frequency of API tweet post dates



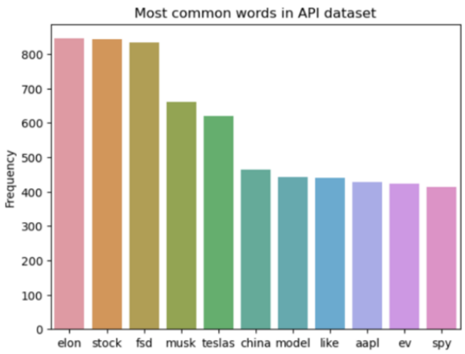
8.2.2 Frequency of API tweet output labels

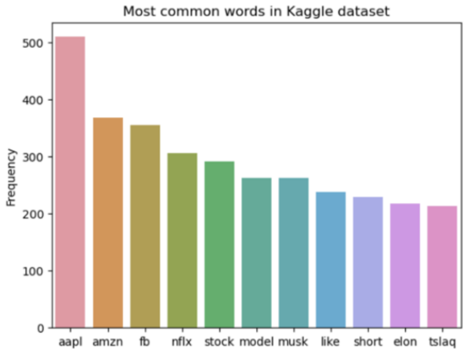


8.2.3 Length of tweets

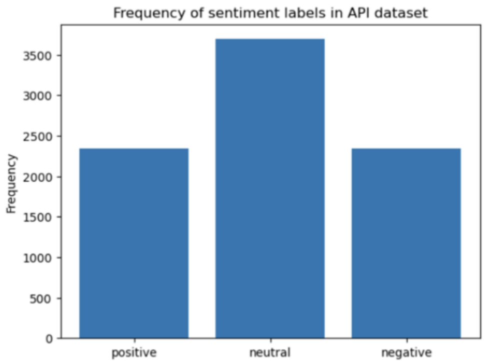


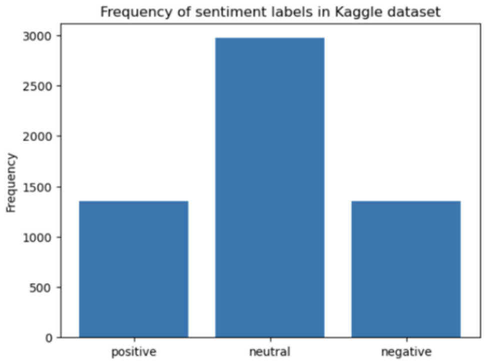
8.2.4 Most common words



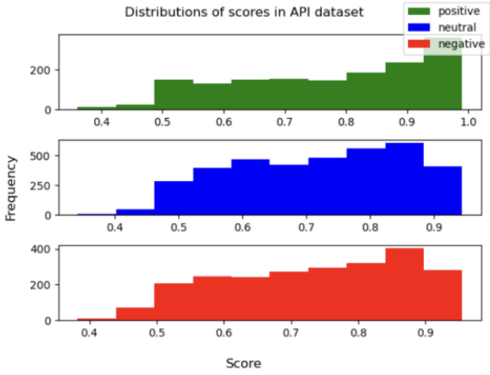


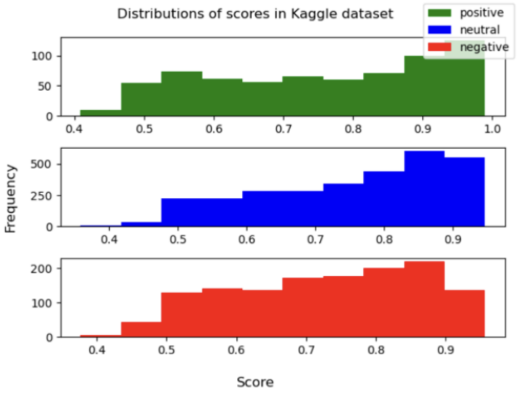
8.2.5 Sentiment labels





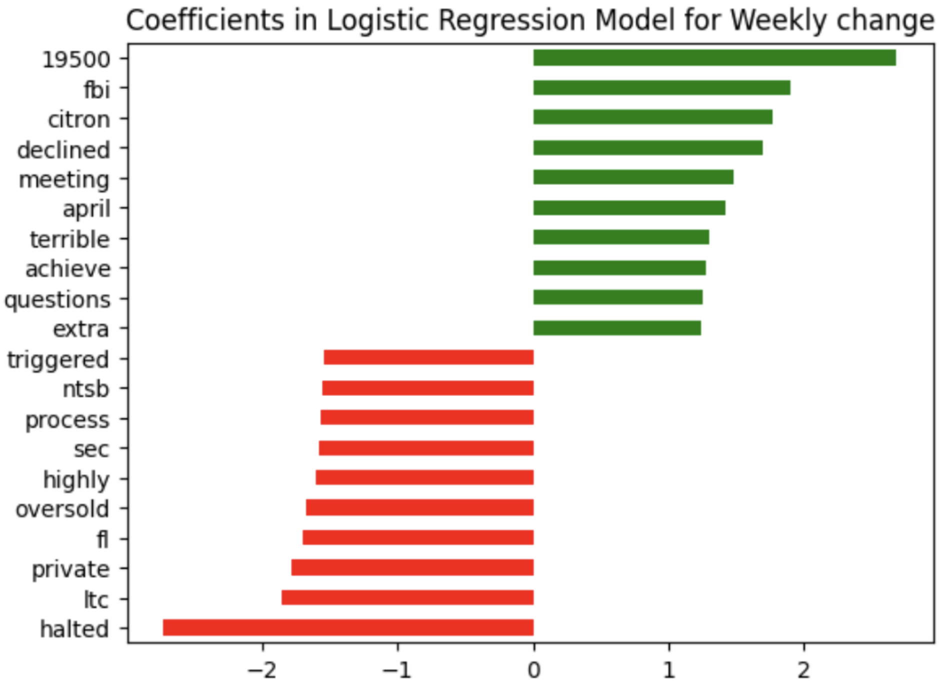
8.2.6 Distributions of sentiment confidence scores



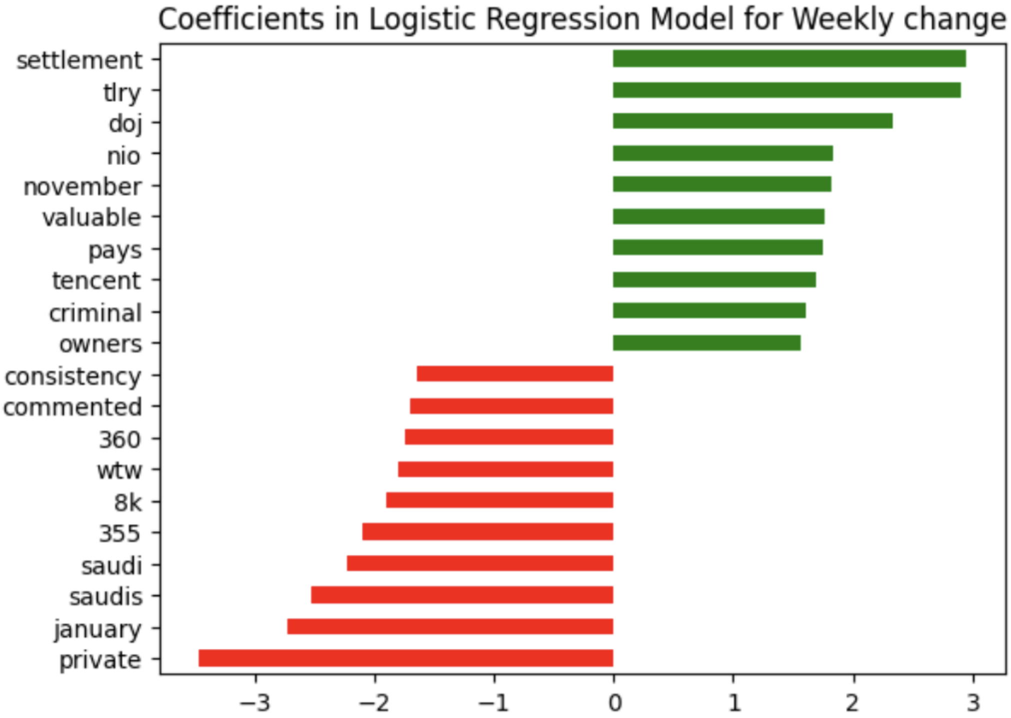


8.3 Logistic Regression Model with Bag of Words (BOW)

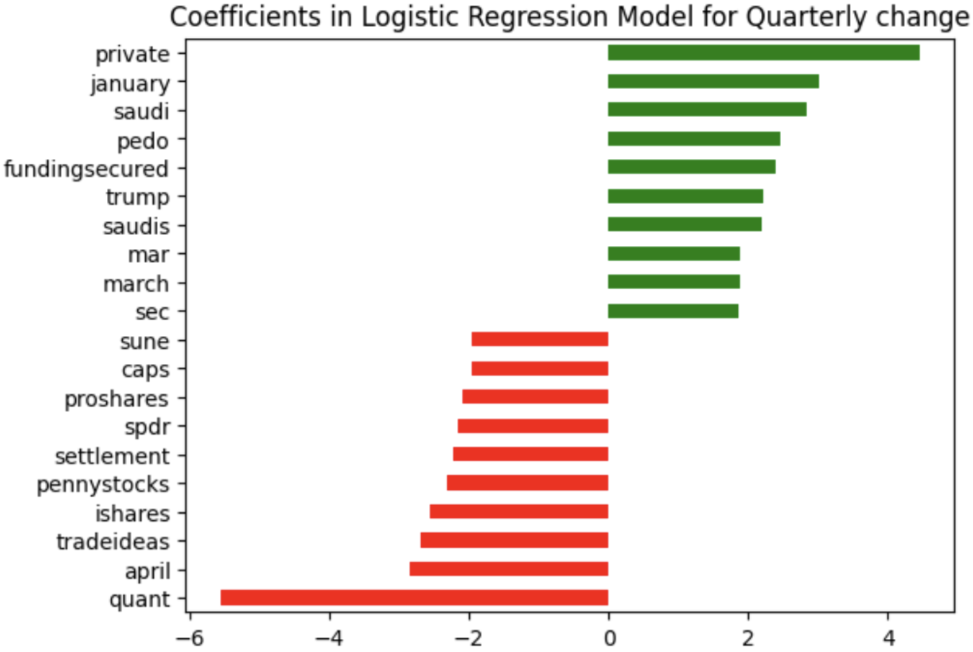
8.3.1 Top coefficients in logistic regression model utilizing weekly percent change labels



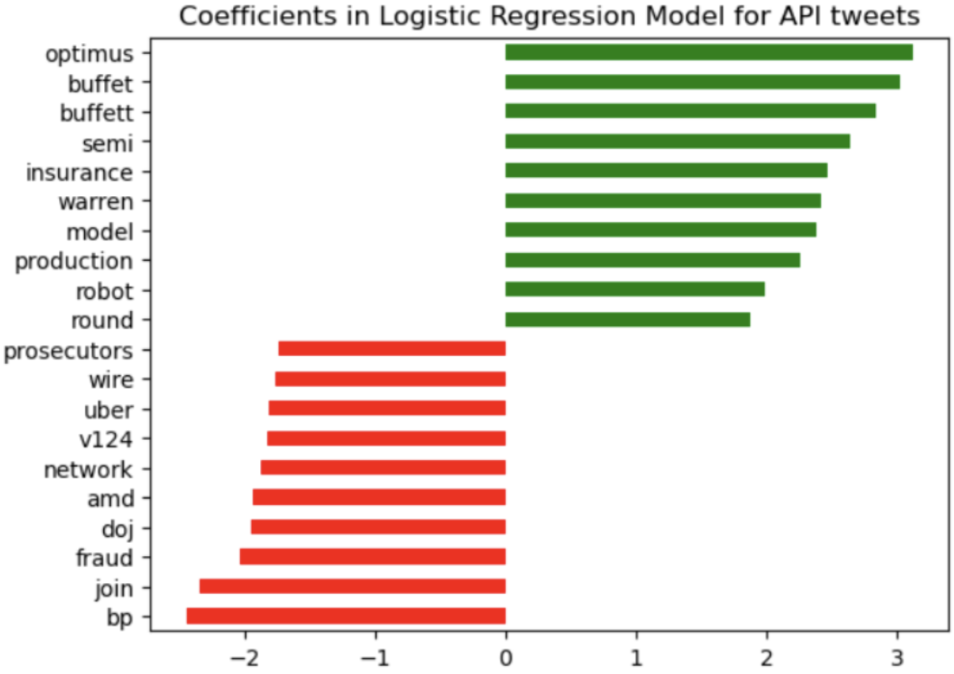
8.3.2 Top coefficients in logistic regression model utilizing monthly percent change labels



8.3.3 Top coefficients in logistic regression model utilizing quarterly percent change labels



8.3.4 Top coefficients in logistic regression model for API tweets



8.3.5 Logistic regression evaluation metrics

Classification Report (API Data)

```
Precision   Recall   F1
0 (Decrease)  0.69      0.71     0.70
1 (Increase)  0.67      0.66     0.66
```

Classification Report (Weekly Model)

```
Precision   Recall   F1
0 (Decrease)  0.51      0.50     0.50
1 (Increase)  0.51      0.53     0.52
```

Classification Report (Monthly Model)

```
Precision   Recall   F1
0 (Decrease)  0.49      0.37     0.42
1 (Increase)  0.49      0.60     0.54
```

Classification Report (Quarterly Model)

```
Precision   Recall   F1
0 (Decrease)  0.48      0.47     0.48
1 (Increase)  0.48      0.50     0.49
```

8.4 Logistic Regression with Sentiment Analysis

Classification Report (API Data)

```
Precision   Recall   F1
0 (Decrease)  0.53      0.70     0.60
1 (Increase)  0.49      0.32     0.38
```

Confusion Matrix (API Data)

```
552   241
494   229
```

Classification Report (Weekly Model)

```
Precision   Recall   F1
0 (Decrease)  0.52      0.77     0.62
1 (Increase)  0.57      0.29     0.38
```

Confusion Matrix (Weekly Model)

```
383   112
358   147
```

Classification Report (Monthly Model)

```
Precision   Recall   F1
0 (Decrease)  0.49      0.68     0.57
1 (Increase)  0.50      0.32     0.39
```

Confusion Matrix (Monthly Model)

```
335   160
345   160
```

Classification Report (Quarterly Model)

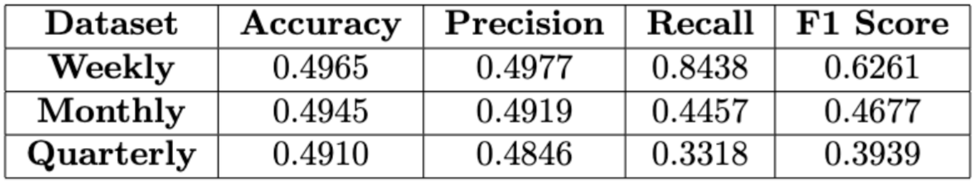
```
Precision   Recall   F1
0 (Decrease)  0.49      0.63     0.55
1 (Increase)  0.50      0.36     0.41
```

Confusion Matrix (Quarterly Model)

```
312   183
325   180
```

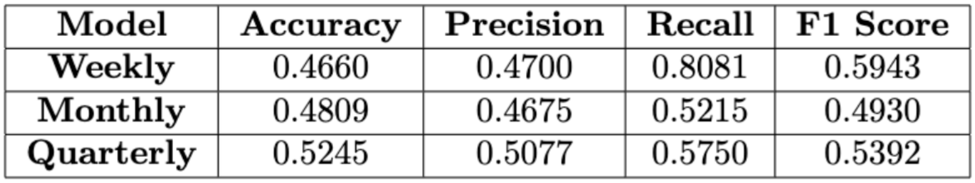
8.5 LSTM Model with Word2Vec

8.5.1 Performance Metrics



Dataset	Accuracy	Precision	Recall	F1 Score
Weekly	0.4965	0.4977	0.8438	0.6261
Monthly	0.4945	0.4919	0.4457	0.4677
Quarterly	0.4910	0.4846	0.3318	0.3939

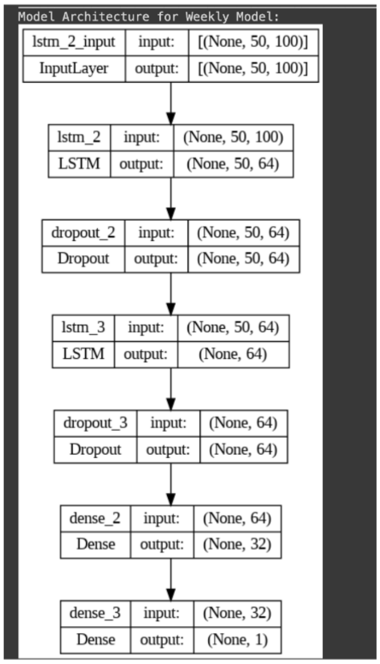
Table 1: Optimized performance metrics for test data



Model	Accuracy	Precision	Recall	F1 Score
Weekly	0.4660	0.4700	0.8081	0.5943
Monthly	0.4809	0.4675	0.5215	0.4930
Quarterly	0.5245	0.5077	0.5750	0.5392

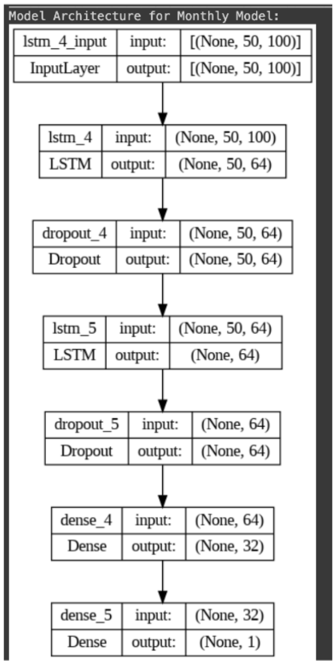
Table 2: Performance metrics for curr_tweets dataset

8.5.2 Model Architecture(s)



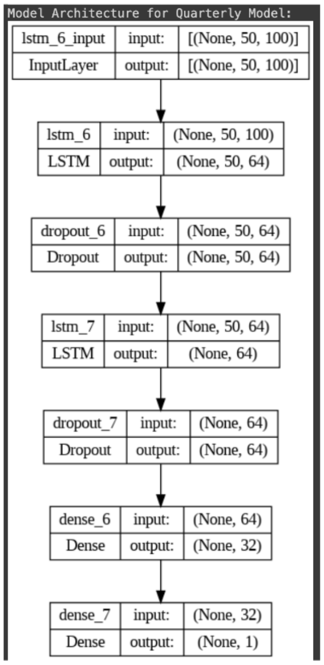
Model Architecture for Weekly Model:

lstm_2_input | input: | [(None, 50, 100)]
InputLayer | output: | [(None, 50, 100)]
lstm_2 | input: | (None, 50, 100)
LSTM | output: | (None, 50, 64)
dropout_2 | input: | (None, 50, 64)
Dropout | output: | (None, 50, 64)
lstm_3 | input: | (None, 50, 64)
LSTM | output: | (None, 64)
dropout_3 | input: | (None, 64)
Dropout | output: | (None, 64)
dense_2 | input: | (None, 64)
Dense | output: | (None, 32)
dense_3 | input: | (None, 32)
Dense | output: | (None, 1)



Model Architecture for Monthly Model:

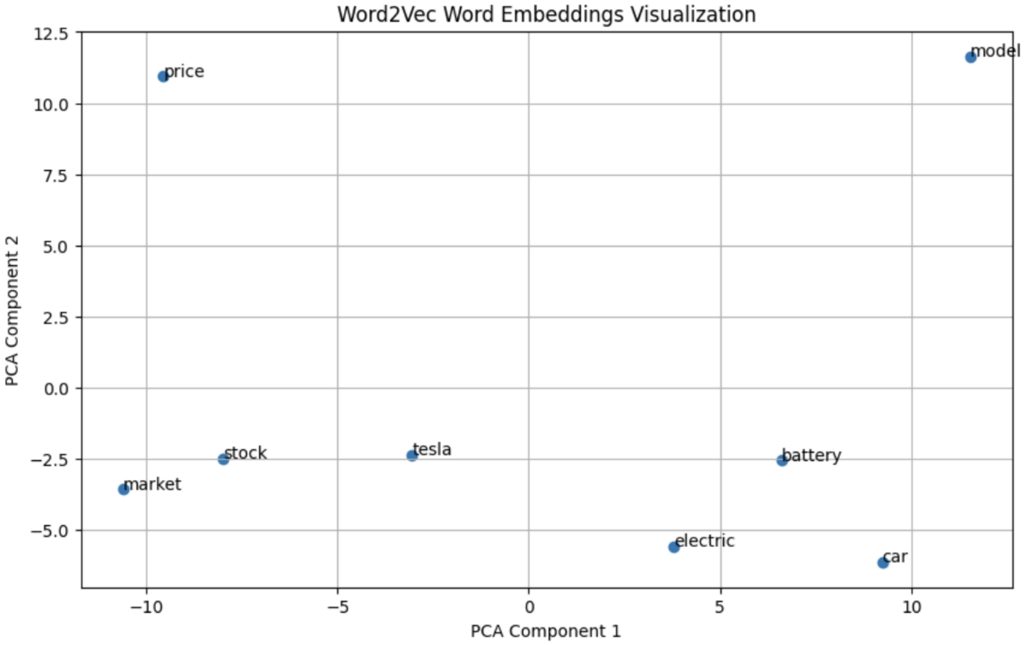
lstm_4_input | input: | [(None, 50, 100)]
InputLayer | output: | [(None, 50, 100)]
lstm_4 | input: | (None, 50, 100)
LSTM | output: | (None, 50, 64)
dropout_4 | input: | (None, 50, 64)
Dropout | output: | (None, 50, 64)
lstm_5 | input: | (None, 50, 64)
LSTM | output: | (None, 64)
dropout_5 | input: | (None, 64)
Dropout | output: | (None, 64)
dense_4 | input: | (None, 64)
Dense | output: | (None, 32)
dense_5 | input: | (None, 32)
Dense | output: | (None, 1)



Model Architecture for Quarterly Model:

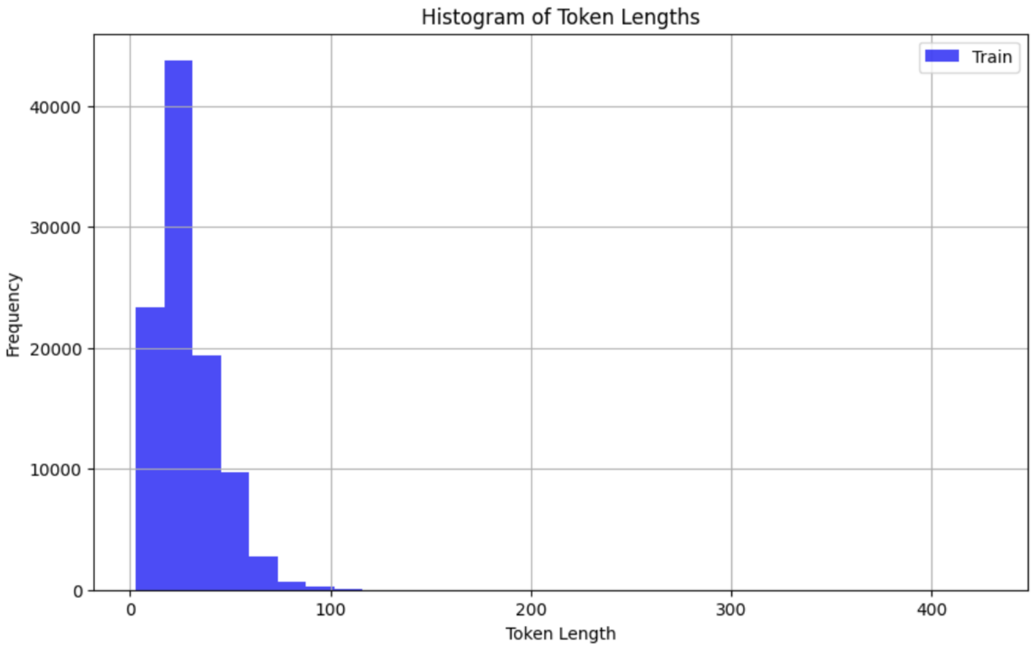
lstm_6_input | input: | [(None, 50, 100)]
InputLayer | output: | [(None, 50, 100)]
lstm_6 | input: | (None, 50, 100)
LSTM | output: | (None, 50, 64)
dropout_6 | input: | (None, 50, 64)
Dropout | output: | (None, 50, 64)
lstm_7 | input: | (None, 50, 64)
LSTM | output: | (None, 64)
dropout_7 | input: | (None, 64)
Dropout | output: | (None, 64)
dense_6 | input: | (None, 64)
Dense | output: | (None, 32)
dense_7 | input: | (None, 32)
Dense | output: | (None, 1)

8.5.3 Word2Vec embeddings

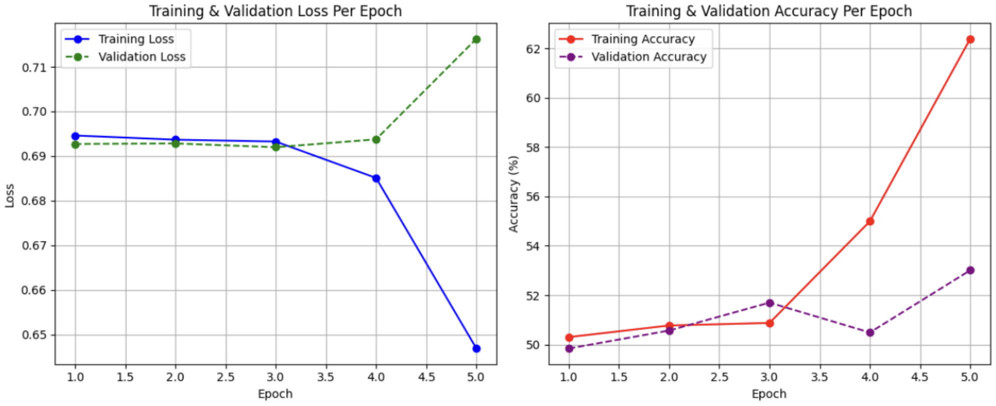


8.6 Fine-tuning DistilBERT model with pre-trained word embeddings

8.6.1 Analysis of DistilBERT tokenization lengths



8.6.2 DistilBERT results using weekly percent change labels



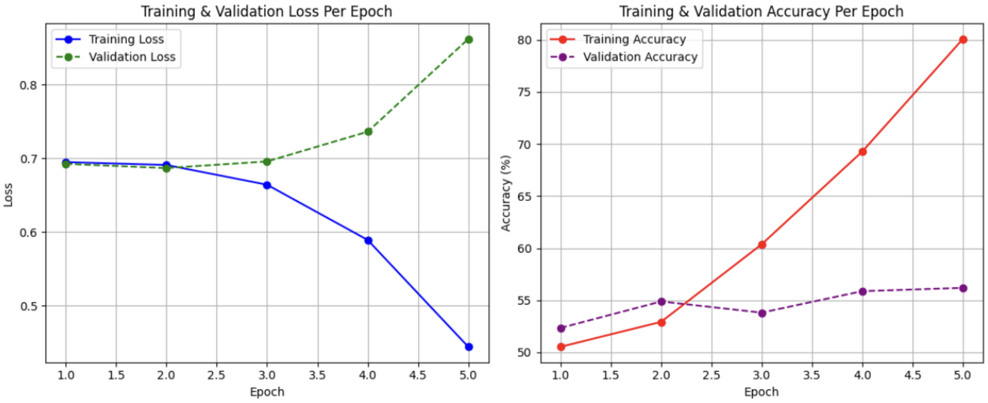
Classification Report (Weekly Model)

```
Precision   Recall   F1
0 (Decrease)  0.50      0.71     0.58
1 (Increase)  0.50      0.29     0.37
```

Confusion Matrix (Weekly Model)

```
3510  1490
3527  1473
```

8.6.3 DistilBERT results using monthly percent change labels



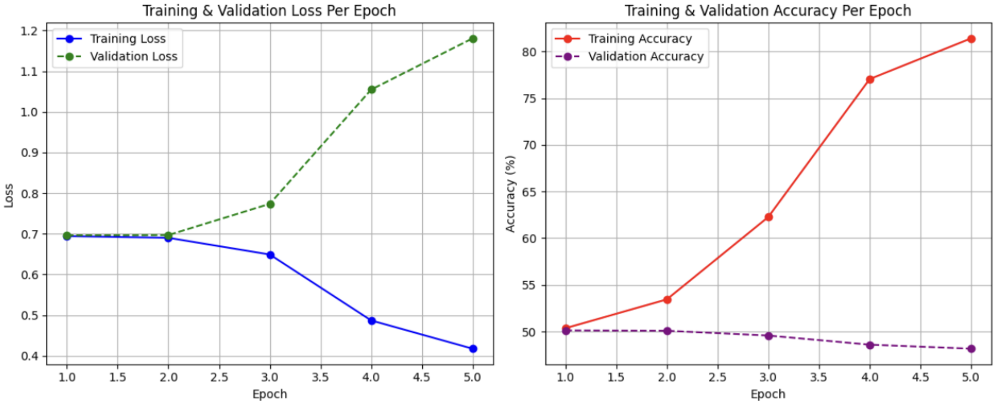
Classification Report (Monthly Model)

	Precision	Recall	F1
0 (Decrease)	0.48	0.37	0.42
1 (Increase)	0.49	0.60	0.54

Confusion Matrix (Monthly Model)

1858	3142
1997	3003

8.6.4 DistilBERT results using quarterly percent change labels



Classification Report (Quarterly Model)

	Precision	Recall	F1
0 (Decrease)	0.49	0.40	0.44
1 (Increase)	0.49	0.58	0.53

Confusion Matrix (Quarterly Model)

2020	2980
2122	2878