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.