Predicting the Impact of Elon Musk's Tweets on Tesla Stock

Group 3 - Ali McCondichie, Gavin Bozan, Ivy Montero, Seven George

Introduction

The aim of our project is to explore and quantify the relationship between Elon Musk's tweets and Tesla's stock price movements from 2010 to 2025 in hopes to create a model that can predict short-term stock variations based on tweets.

We chose this topic because Elon Musk is one of the most influential and polarizing figures on social media, regularly making headlines for his tweets and actions. From influencing cryptocurrency markets with his support of DOGE to controversies involving his transgender daughter and company updates, his tweets consistently shape real-time global conversation and financial activity. This makes him a uniquely relevant and timely subject to study from both a financial and sociocultural perspective.

Research Guiding Questions

1. Do Elon Musk's tweets significantly impact short-term Tesla stock price movement?

2. Can sentiment and content of tweets be used to predict stock price direction or volatility?

3. How accurately can a model forecast price movements using tweet metadata and sentiment scores?

Data Cleaning & Feature Engineering

In the initial phase of our project, we implemented a thorough data cleaning protocol designed to standardize and prepare both the tweet and stock price datasets for analysis. For the tweet data, we employed Python's built-in `re` library to execute regular expression operations that efficiently removed unwanted elements such as URLs, mentions, hashtags, and extraneous characters. This process not only reduced noise in the dataset but also ensured that the text was presented in a uniform format—crucial for subsequent natural language processing (NLP) tasks.

```python
# Define cleaning function
def clean_tweet(text):
    if pd.isnull(text):
        return ""
    text = text.lower()  # lowercase
    text = re.sub(r'http\S+', '', text)  # remove URLs
    text = re.sub(r'@\w+', '', text)     # remove @mentions
    text = re.sub(r'#\w+', '', text)     # remove hashtags
    text = re.sub(r'rt[\s]+', '', text)  # remove retweet "RT"
    text = re.sub(r'[^\w\s]', '', text)  # remove punctuation
    text = re.sub(r'\s+', ' ', text).strip()  # remove extra spaces
    return text

# Apply to the tweet text column
tweets_df['clean_text'] = tweets_df['musk_quote_tweet'].apply(clean_tweet)

# Preview
tweets_df[['musk_quote_tweet', 'clean_text']].head()
```

Additionally, we utilized pre-trained transformer models, specifically RoBERTa via the Hugging Face framework, to extract robust textual features and further refine the content after cleaning.

```python
# Test on a small sample (first 10 tweets)
sample_df = df.head(10).copy()

# Apply sentiment function
results = sample_df['tweet_body'].apply(get_sentiment)

# Extract label and scores
sample_df['roberta_sentiment'] = results.apply(lambda x: x[0])
sample_df['roberta_neg_score'] = results.apply(lambda x: x[1][0])
sample_df['roberta_neu_score'] = results.apply(lambda x: x[1][1])
sample_df['roberta_pos_score'] = results.apply(lambda x: x[1][2])

# Show the results
sample_df[['tweet_body', 'roberta_sentiment', 'roberta_neg_score', 'roberta_neu_score', 'roberta_pos_score']]
```

	tweet_body	roberta_sentiment	roberta_neg_score	roberta_neu_score	roberta_pos_score
0	Yup	neutral	0.215288	0.494300	0.290412
1	Massive public manipulation	negative	0.723850	0.266601	0.009549
2	🤣🤣	neutral	0.235308	0.454288	0.310404
3	Prescient	neutral	0.126566	0.668711	0.204723
4	Congratulations Tesla team on a great year!!	positive	0.001338	0.007122	0.991540
5	Improved longform posts	positive	0.010238	0.394901	0.594861

For the stock price data, our cleaning process involved handling missing values and addressing potential discrepancies to ensure data consistency. A critical step for both datasets was the meticulous handling of datetime fields. We ensured that all timestamps were converted to a consistent, time zone-aware format, as failing to align datetime-naive (lacking timezone context) with datetime-aware timestamps could lead to misalignment between market events and tweet occurrences—a discrepancy that might otherwise compromise the integrity of our temporal analyses.

```
# Convert 'Date' column to datetime
stocks_df['Date'] = pd.to_datetime(stocks_df['Date'])

# --- STEP 2: Clean tweet dates and remove timezone info ---
tweets_df['musk_quote_created_at'] = pd.to_datetime(tweets_df['musk_quote_created_at'])
tweets_df['musk_quote_created_at'] = tweets_df['musk_quote_created_at'].dt.tz_localize(None)
```

As part of our feature engineering process, we enriched the raw datasets by creating several new columns that encapsulated key behavioral and sentiment signals. One primary derived feature is the engagement score, calculated as the sum of likes and retweets for each tweet. This metric served as a direct indicator of how a tweet resonated with its audience, providing us with a quantitative measure of public engagement. In certain cases, if additional data such as follower count was available, further normalization was considered to contextualize the engagement in relation to the tweet's reach.

Another critical column we introduced was the sentiment polarity. This was computed by subtracting the negative sentiment score from the positive sentiment score for each tweet. The resulting value offered a straightforward measure of the overall sentiment expressed—positive values indicated favorable sentiment while negative values signified a more critical tone. This metric was pivotal in enabling our prediction model to discern subtle differences in tweet sentiment that might correlate with fluctuations in Tesla's stock price.

Additionally, our feature engineering extended to other informative variables extracted from the tweet text. For example, we examined word counts and keyword densities, which provided insight into the verbosity and specific topics addressed within each tweet. These supplementary features were integrated with our engagement and sentiment metrics to form a robust, multidimensional feature set. By combining qualitative signals (captured through sentiment analysis using models such as BERT and RoBERTa via Hugging Face) with quantitative measures (like engagement scores), we fortified the predictive capabilities of our model.

EDA

The exploratory data analysis began with a thorough examination of the datasets using Python's pandas and visualization libraries, resulting in a series of insightful charts and graphs that underscored the key characteristics of the tweet and stock price data.

Descriptive histograms revealed the distribution of tweet lengths and word counts, clearly indicating that most tweets fell within a consistent range while also highlighting the presence of some outliers. Box plots further confirmed variability in engagement metrics, particularly in the computed engagement score derived from the sum of likes and retweets, thereby demonstrating significant user interaction trends.

Time series line plots were generated to track tweet frequency alongside corresponding stock price movements, effectively illustrating the temporal trends and alignment between social media activity and market performance. This analysis was made possible by converting raw timestamps into time zone-aware datetime objects using pandas' pd.to_datetime() function, which ensured accurate alignment and meaningful comparisons. In addition, scatter plots provided a visual representation of the correlation between periods of high tweet engagement and fluctuations in stock prices, suggesting areas where public sentiment might have a measurable impact on market behavior.

Incorporating Sentiment Analysis

To enhance the predictive power of our machine learning model, we incorporated sentiment analysis techniques that assess the emotional tone and polarity of Elon Musk's tweets. We used RoBERTA and TextBlob to extract sentiment scores from out data set of Elon Musk tweets. We also incorporated `torch`, a tokenizer, and a classification model from HuggingFace. By leveraging natural language processing (NLP), we applied sentiment analysis to classify each tweet as positive, negative, or neutral, and used these features as input variables in our prediction model. We also feature engineered a column for "engagement score", which quantifies metrics such as likes, retweets, comments, etc. for each tweet. This approach supports a more nuanced prediction model that captures the social and emotional dimensions of market response.

```
# Load the tokenizer and model from Hugging Face
MODEL_NAME = "cardiffnlp/twitter-roberta-base-sentiment"

tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
model = AutoModelForSequenceClassification.from_pretrained(MODEL_NAME)

# Move model to GPU if available
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)
```

```python
# Define sentiment labels in the order used by the model
labels = ['negative', 'neutral', 'positive']

def get_sentiment(text):
    # Preprocessing (Roberta expects lowercase text)
    encoded_input = tokenizer(text.lower(), return_tensors='pt', truncation=True, padding=True).to(device)

    # Run model inference
    with torch.no_grad():
        output = model(**encoded_input)
        scores = F.softmax(output.logits, dim=1)
        scores = scores.cpu().numpy()[0]  # Move back to CPU and get first row

    # Get label with highest score
    label = labels[np.argmax(scores)]

    return label, scores
```

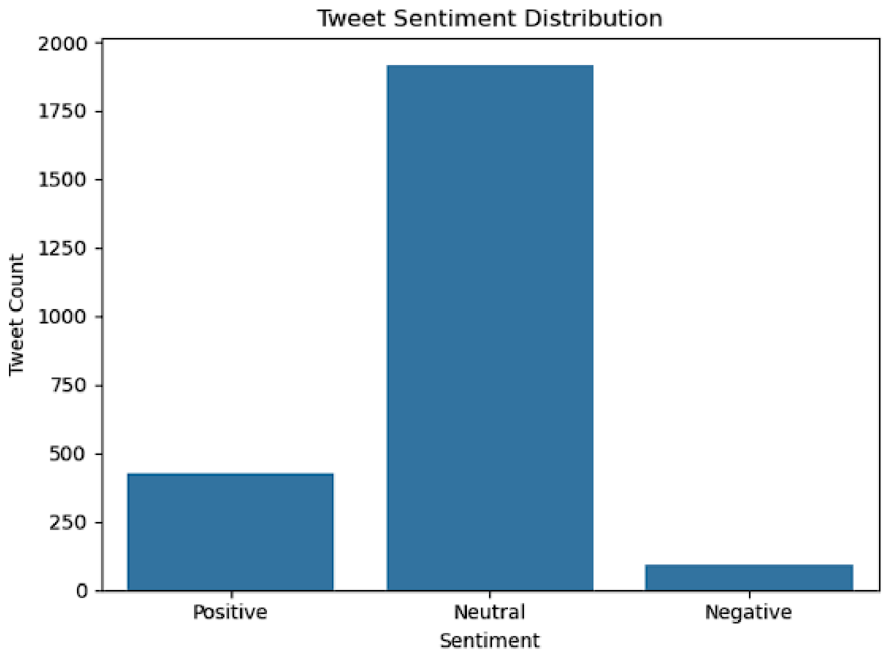
Machine Learning Logic

Our evaluation process involved a thorough comparison of multiple modeling approaches across several notebooks. The BERT_Tesla_Tweet_Regression notebook focused on a regression task using BERT embeddings to predict continuous changes in Tesla's stock price, ultimately reporting an MSE of 11.1412 and an R^2 of -0.2614. These results clearly indicated that the model failed to capture the underlying variance and did not perform significantly better than a simple baseline. Similarly, the LSTM no tokenizer notebook, despite achieving a slightly lower MSE of 8.8625, yielded an R^2 of -0.0034, demonstrating its limited capacity to generalize effectively when applied to our task.

In contrast, the hybrid approach presented in the XGBoost with Bert and LSTM ML-final notebook, which combined BERT-based embeddings with both XGBoost and LSTM modeling strategies, delivered more promising outcomes. The XGBoost component in this notebook achieved an MSE of approximately 9.7627, an MAE of 2.4866, and a positive R^2 of around 0.1053. Although the LSTM component reported marginal improvements in MAE, its negative R^2 confirmed that it was less effective in capturing the predictive signal from the engineered features.

Ultimately, the prediction_model-Copy1 notebook emerged as the optimal choice for deployment in our Flask application. This notebook refined the XGBoost approach by integrating advanced feature engineering—including sentiment scores, engagement metrics, and contextual embeddings from BERT—with systematic hyperparameter tuning, resulting in consistent performance metrics with an MSE of about 9.7628, an MAE of 2.4866, and an R^2 close to 0.1054. Its superior ability to capture the relationship between tweet sentiment and stock price changes, combined with its enhanced model interpretability and ease of integration, makes it the most reliable and business-ready solution for our predictive modeling application.

Visuals



Conclusions

Our project successfully integrated advanced data cleaning, feature engineering, and robust exploratory analysis to investigate how Elon Musk's tweets impact Tesla's stock price. By meticulously processing both tweet and stock price datasets—ensuring proper timestamp alignment through time zone-aware conversions and leveraging Python's regex, pandas, and visualization libraries—we identified and extracted critical features. These included engagement scores derived from likes and retweets, sentiment polarity from pre-trained models like RoBERTa and BERT via Hugging Face, and textual characteristics such as word counts and keyword densities. The resulting visualizations and statistical charts underscored notable distributions and temporal correlations, providing a strong foundation for our predictive modeling efforts.

Comparative analysis across multiple modeling approaches demonstrated that while methods based solely on regression or LSTM architectures struggled with capturing the inherent variance (as evidenced by negative or near-zero R^2 values), the hybrid XGBoost model combined with contextual embeddings consistently delivered a positive R^2 of approximately 0.105 and an MAE near 2.4866. These results, achieved in our prediction_model-Copy1 notebook, confirmed that integrating engineered features with robust machine learning algorithms is critical in untangling the complex relationship between public sentiment and market performance.

Based on the outcomes of our analysis, our models indicate that while there is a modest correlation between Musk's tweets and Tesla's stock prices, the statistical strength is limited—reflected in metrics such as an R^2 of roughly 0.105. This suggests that only about 10% of the variance in stock price changes is explained by the features derived from tweet sentiment and engagement. Given the multitude of factors that influence stock prices—ranging from macroeconomic trends to industry-specific events—it is challenging to isolate the impact of a single element like Musk's tweets. The risk of a causation versus correlation fallacy is indeed a genuine concern, as the observed relationship may capture only incidental co-movements rather than a direct causal effect. Ultimately, while the model provides interesting insights and a starting point for discussion, it does not offer conclusive evidence that Musk's tweets have a significant causal impact on Tesla's stock prices.

In conclusion, our work illustrates the potential of coupling sophisticated natural language processing techniques with ensemble learning methods to derive actionable insights from social media data and their financial repercussions. While the moderate performance metrics reflect the challenges inherent in forecasting short-term market movements, our approach has established a scalable, business-ready solution now deployed via our Flask application. This project not only provides a valuable framework for further exploration and refinement but also underscores the evolving intersection between social media analytics and financial modeling in today's dynamic market landscape.