# Financial Forecasting with AI: Transformer Models' Role in Analyzing Twitter and Stock Correlations

| | | |
|---|---|---|
| Prepared by | : | **GÖKAY AKÇAY** |
| Student No | : | **090200147** |
| Submission Date | : | **January 25, 2024** |

| | | |
|---|---|---|
| Course | : | **FIZ 4901E** |
| Supervisor | : | **PROF DR. EMRE ONUR KAHYA** |

## Table of Contents

# 1. Definition and Purpose of Design

This study aims to establish a compelling correlation between sentiment scores derived from financial text data and the values of stock prices. This will be achieved by demonstrating the fluctuations in stock prices over various time periods in relation to the emotional scores found from relevant financial texts. To validate the result, the price of the stock will be examined in different time scales with different validation metrics. In addition, the effectiveness of sentiment score points over different time intervals, the influence of specific scores on prices, and the utilization of sentimental scores in various algorithms for price prediction will also be examined.

# 2. Scope of Design and Areas of Usage

In this research, tweets from Kaggle were used to predict changes in prices. Two methods were tried: RoBERTa, a transformer model, and VADER, a common sentiment analysis tool. The impact of neutral and positive sentiment scores on price changes was also examined. To ensure prediction accuracy, various measures such as MSE, RMSE, R2 scores, and correlation scores (including the Pearson Coefficient and P-values) were employed. These were tested across different machine learning models and various time intervals to determine which performed best. Besides the price prediction problem, the price data also categorized by price changes for performing the model results in classification problems. Therefore, by detecting the price movement signals and creating price vs sentimental score patterns, this aims to offer a solution in computational finance and quantitative analysis fields.

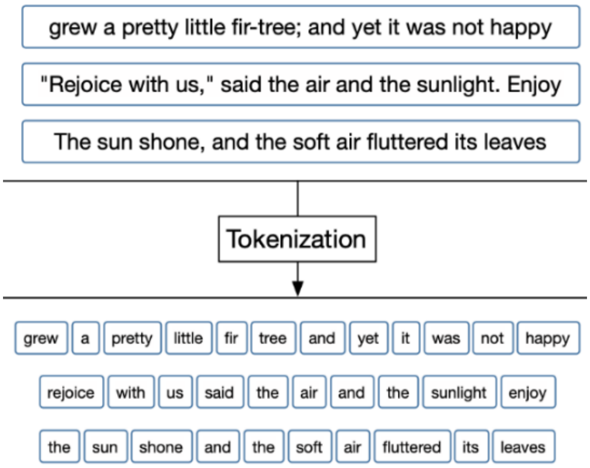
# 3. Conducted Studies

## 3.1 Pre-Processing of a Text

In NLP studies, it is typical to focus on either pure analysis or generation by isolating the fundamental units known as words. It is evident that without separating these units, it is impossible to conduct any meaningful analysis. **[1]** When preparing a text for processing, steps such as removing links (URLs), eliminating stop words, and converting the text to lowercase can be listed. Subsequently, three important stages, namely Tokenization, Lemmatization, and Stemming. These processes collectively transform the raw text into a format suitable for vectorization, where each tokenized, lemmatized, or stemmed term can be represented as a feature in a numerical vector space, ready for use in machine learning algorithms.

### 3.1.1 Tokenization

This process involves dividing sentences into a bag of words, enabling easy vectorization of the words.



grew a pretty little fir-tree; and yet it was not happy

"Rejoice with us," said the air and the sunlight. Enjoy

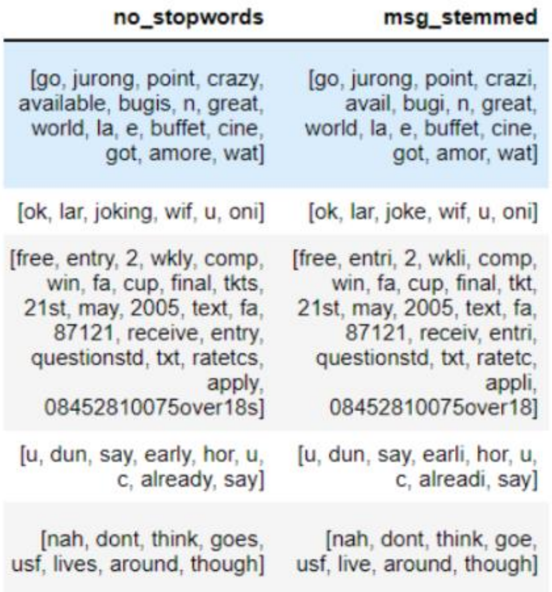
The sun shone, and the soft air fluttered its leaves

Tokenization
grew a pretty little fir tree and yet it was not happy
rejoice with us said the air and the sunlight enjoy
the sun shone and the soft air fluttered its leaves

Figure 1: Tokenization Process

### 3.1.2 Stemmization

Stemming is the process of reducing words to their root form by removing prefixes such as adjectives or adverbs. For example, "available" becomes "avail," "crazy" becomes "crazi," and "entry" becomes "entri."



| no_stopwords | msg_stemmed |
|---|---|
| [go, jurong, point, crazy, available, bugis, n, great, world, la, e, buffet, cine, got, amore, wat] | [go, jurong, point, crazi, avail, bugi, n, great, world, la, e, buffet, cine, got, amor, wat] |
| [ok, lar, joking, wif, u, oni] | [ok, lar, joke, wif, u, oni] |
| [free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receive, entry, questionstd, txt, ratetcs, apply, 08452810075over18s] | [free, entri, 2, wkli, comp, win, fa, cup, final, tkt, 21st, may, 2005, text, fa, 87121, receiv, entri, questionstd, txt, ratetc, appli, 08452810075over18] |
| [u, dun, say, early, hor, u, c, already, say] | [u, dun, say, earli, hor, u, c, alreadi, say] |
| [nah, dont, think, goes, usf, lives, around, though] | [nah, dont, think, goe, usf, live, around, though] |

Figure 2: Stemming

### 3.1.3 Lemmatization

After stemming, some of the words whose roots that being derived do not convey a meaning on their own. For example, words like "entri" and "crazi" have no equivalent in the language. Lemmatization aims to prevent the loss of meaning by reverting these words to their nearest meaningful form, such as "crazi" to "crazy" and "entri" to "entry."

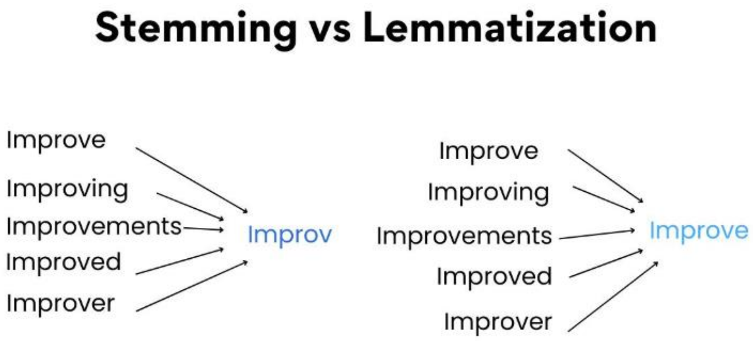


Figure 3: Lemmatization

### 3.1.4 Tagging

Tagging is created to prevent the loss of the actual meanings of words in a sentence after tokenization. It adds tags indicating the role of each word in the sentence, such as noun, verb, adjective, pronoun, etc., alongside each word.

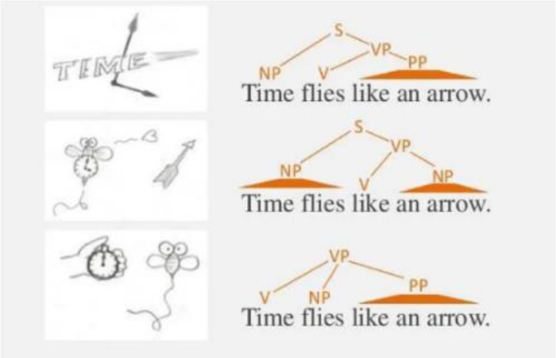

TIME
S
VP
PP
NP
V
Time flies like an arrow.
S
VP
NP
V
NP
Time flies like an arrow.
VP
V
NP
PP
Time flies like an arrow.

Figure 4: Tagging

### 3.1.5 Chunking

In some cases, words should be considered as groups. As shown in the figure above, an adjective can modify a noun, or an adverb can modify a verb and an adjective. For example, "Time flies" (Noun Phrase) and "Like an arrow" (Prepositional phrase) are chunked accordingly. In some cases, we also want to examine aspects like whether verbs are in past or future tense, or if a word is singular or plural. Chunking achieves all of this by dividing words into specific chunks.

## 3.2 Transformer Architecture

A transformer is a self-attention and a deep learning model which are primarily used in Natural Language Processing model. Attention can be described as what humans focus specific words in text. **[2]**

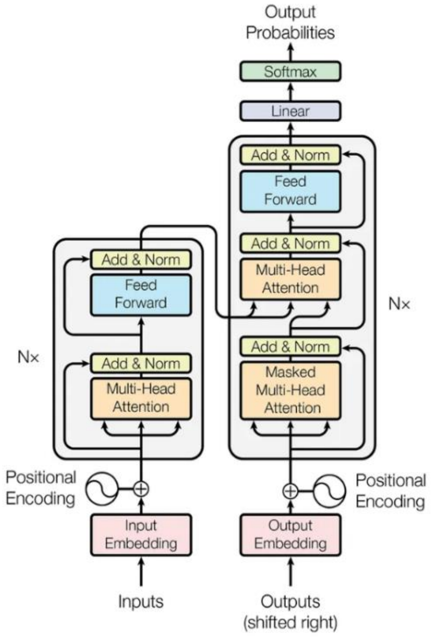


Figure 5: Model Architecture of Transformers

One of the most important aspects of this model compared to others is its use of a structure called the attention mechanism. The Transformer model divides a sequence into two parts, the encoder and the decoder, and utilizes self-attention mechanisms instead of recurrent neural networks (RNNs). What sets Transformers apart from architectures like LSTM, GRU, and RNN is primarily their heavy reliance on multi-head attention and layers called the Feedforward Network (FFN). **[3]**

### 3.2.1 Positional Encoders:

Transformers have two major components: self-attention and a position-wise feed forward layer. Both are permutation equivariant and are not sensitive to the order of input tokens

Before, we attempted to vectorize our sentences by subjecting them to processes like tokenization and tagging. Now, we aim to examine the relationships between the columns of these vectors, which are processed in a sequence to sequence manner. However, Self-attention and position-wise feed forward layers are both permutation equivariant and are not sensitive to the order of input tokens.**[3]** There the arrangement of words within a sentence can influence the accuracy of the model. For instance, the positioning of homonymous words can significantly alter the meaning of a sentence. Therefore, after the input encoding stage, it is important to develop a method to convey word order to the machine. In order to make these models position-aware, the position information must be held by embedded into tokens **[3]** There are various approaches to achieve this, but common ones include associating vectors with the positions of words, normalizing the positional vector, and applying frequency-based embedding like using sinosodial functions. The equations below represent one of the approaches for positional embedding: frequency-based embedding.

$$\mathrm{PE}_{(\mathrm{pos},\,2\mathrm{i})} = \sin\left(\frac{\mathrm{pos}}{10000^{2\mathrm{i}/\mathrm{d}_{\mathrm{model}}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\mathrm{model}}}}\right)$$

### 3.2.2 Multi-Head-Attention

The attention mechanism processes the input sequence to determine the key segments that are significant. In scope of this study , for example “Prices are increased.” A trained self-attention layer will weight the association “Price” to “increase” than the word “are”. This is what we called as attention mechanism **[4]**

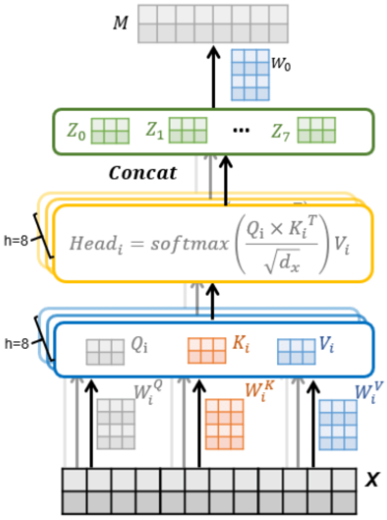


Figure 6: Multi-Head-Attention Mechanism

One way to do this is to examine our word vectors in three different vectors called q, k, and v, where each is of size nx1. The q vector, or query vector, represents the current word. We can think of it as the query word in search engines. The k vector, or key vector, represents the other words in the sequence, which will be compared to the query vector. The v, or value vector, represents the multiplication of the key vectors with certain attention weights. Instead of using single attention mechanism, projecting the vectors on multiple times is much beneficial**. [3]** It is called as "Multi-Head Attention". In the "Attention Is All You Need" paper, the number of a linear projection hyper-parameter,h is set to 8 based on experimental results but it can vary on the purpose. Therefore, the 3 separate q,k and v vectors are examined in 8 different head-attention mechanisms. As a result, the output weights for each vector correspond to a 512xT matrix (where T is the number of tokenized words). The goal here is to create an attention-head matrix to find a distribution of weights between the words. Briefly, a multi-head attention mechanism calculates the scaled dot-product attention multiple times in parallel. The independent outputs are concatenated and linearly transformed into expected dimensions. Multi-head attention is obtained by using the equation [9]

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^O$$

### 3.2.3 BERT Model

BERT (Bidirectional Encoder Representations from Transformers) is a groundbreaking language representation model developed by Google in 2018. Its impact on various natural language processing (NLP) tasks has been profound, as it has set new benchmarks in areas such as text classification, question answering, and language understanding.

In its vanilla form, BERT consists of two main components: an input encoder, which processes the input data. However, unlike traditional directional models that read text input sequentially from left to right or right to left, the Transformer encoder in BERT processes the entire word sequence simultaneously. This bidirectional processing allows BERT to understand the context of a word by considering both its left and right surroundings **[10]**

One of the key techniques used in BERT's pretraining process is Masked Language Modeling (MLM). In MLM, a percentage of the input tokens in each training example are randomly replaced with a special [MASK] token, and the model is trained to predict the original identities of these masked tokens based on the context provided by the surrounding tokens. This bidirectional approach enables BERT to learn and understand the contextual relationships between words in a sentence, resulting in contextually rich word and sentence representations.

BERT also includes a secondary task called "Next Sentence Prediction" (NSP). During the training process, the model randomly selects two sentences and tries to predict whether they follow each other or not. This helps the model understand relationships within the text.

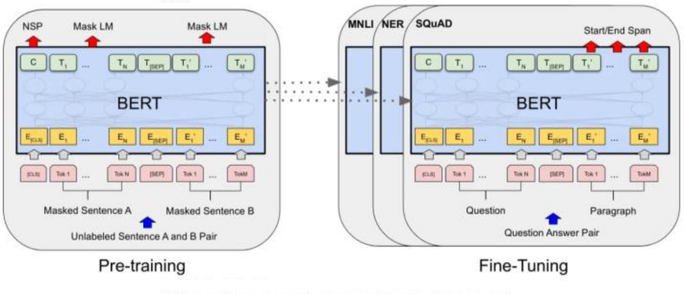


Figure 7: BERT-Model and Masking

### 3.2.4 RoBERTa

In this study, RoBERTa (Robustly Optimized BERT Model) is used due to its advantages on other self-training methods such as being more careful on evaluation of hyperparameter tunings, and sensitiveness to train set size by offering various modifications. There modifications are can be listed as training the model longer, with bigger batches over more data, removing the next sentence prediction objectives, and, dynamically changing masks. **[11]** Besides, RoBERTa is trained on a larged datasets which have been collected from libraries like Book-Corpus including more than 7000 book dataset, CC-new including over 63 million English articles etc. In RoBERTa model next sentence prediction is removed, the relation between words is done by making dynamically masking on dataset.

## 3.3 Lexicon and Rule Based Models

Lexicon-based models, such as VADER (Valence Aware Dictionary and Sentiment Reasoner), are a type of sentiment analysis tool that relies on predefined sets of words and their associated sentiment scores to determine the sentiment of a piece of text. These models are based on the idea that the sentiment of a text can be inferred by analyzing the sentiment of its individual words and combining them to derive an overall sentiment score.

### 3.3.1 VADER (Valance Aware Dictionary and Sentiment Reasoner)

VADER is a sentiment analysis tool known as a lexical database, designed specifically for analyzing sentiments expressed in social media content. This tool is optimized to work effectively with sentiments found in social media text. It employs a diverse range of techniques to accomplish its task.

A key component of this sentiment analysis tool is the sentiment lexicon. This lexicon consists of various lexical features, such as words, that are categorized as either positive or negative based on their sentiment polarity. It does not only provides scores for positivity and negativity but also quantifies the extent to which a sentiment is positive or negative

### 3.4 Conducted Studies on Sentimental Scores Applied on Financial Data

A lot of research has focused on using Transformers and Lexicon based models for predicting prices. Some of these studies have discovered significant insights, suggesting that incorporating sentiment scores could improve predictive models in finance and related fields.

One study aiming to predict prices using time series models like LSTM shows that incorporating sentiment scores improves price prediction. While adding more hidden layers to the LSTM model did not increase its accuracy, using tweets with high polarity to support the model was found to have a significant impact on price predictions **[12]**.

In another study, significant correlations were found between headlines in financial news and prices. When financial data was used with ARIMA, RNN, and Facebook Prophet algorithms, it was observed that RNN outperformed the others and showed higher correlations. **[13]**

Although there appear to be positive correlations between price and sentiment scores, adding sentiment scores obtained from social media and financial news to models resulted in decreased in the highest accuracy in regression problems. However, it was observed that in many classifiers, the model accuracy rates increased when these scores were included. **[14]**

Therefore, in addition to examining the quantative impact of these sentimental scores on prices, it would be useful to consider this study as classification problem. Towards the end of the study, all the tweet and price numerical features converted to class features and reviewed as a classification problem.

## 4. Review of Study Outputs

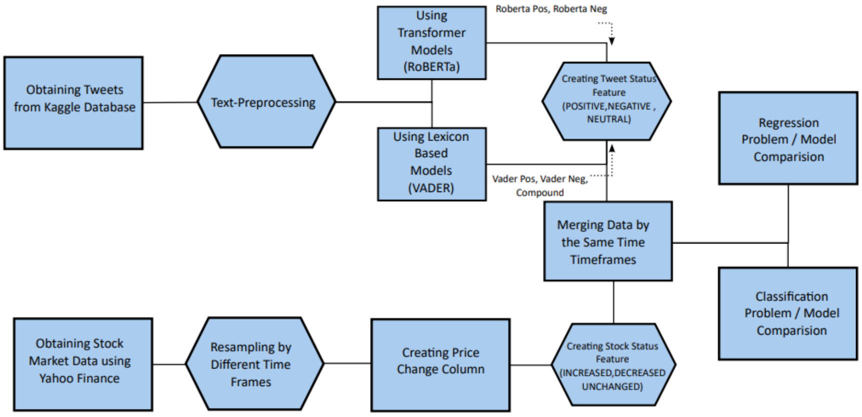
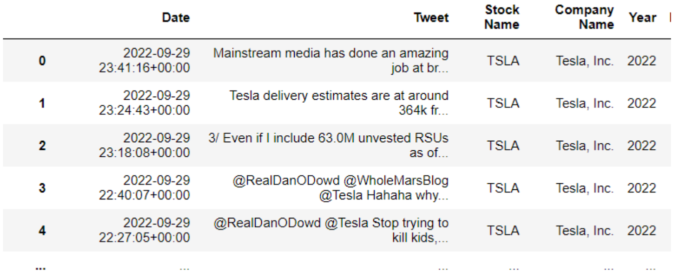

Obtaining Tweets from Kaggle Database
Text-Preprocessing
Using Transformer Models (RoBERTa)
Using Lexicon Based Models (VADER)
Roberta Pos, Roberta Neg
Vader Pos, Vader Neg, Compound
Creating Tweet Status Feature (POSITIVE,NEGATIVE , NEUTRAL)
Merging Data by the Same Time Timeframes
Regression Problem / Model Comparision
Classification Problem / Model Comparision
Obtaining Stock Market Data using Yahoo Finance
Resampling by Different Time Frames
Creating Price Change Column
Creating Stock Status Feature (INCREASED,DECREASED UNCHANGED)

Figure 8: Flow-Chart of the Study

The figure illustrates the steps involved in our project. It begins with data acquisition, where we gather information from Kaggle and Yahoofinance databases. After the preprocessing and labeling, data undergoes to regression and classification models

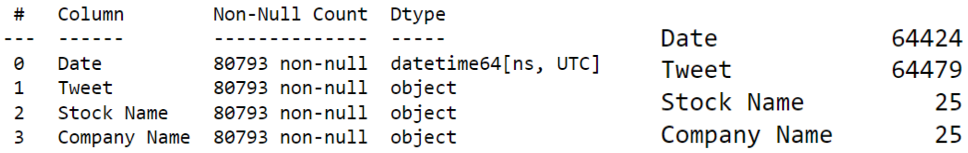
## 4.1 Inspecting Financial Text Data & Pre-Processing



| | Date | Tweet | Stock Name | Company Name | Year |
|---|---|---|---|---|---|
| 0 | 2022-09-29 23:41:16+00:00 | Mainstream media has done an amazing job at br... | TSLA | Tesla, Inc. | 2022 |
| 1 | 2022-09-29 23:24:43+00:00 | Tesla delivery estimates are at around 364k fr... | TSLA | Tesla, Inc. | 2022 |
| 2 | 2022-09-29 23:18:08+00:00 | 3/ Even if I include 63.0M unvested RSUs as of... | TSLA | Tesla, Inc. | 2022 |
| 3 | 2022-09-29 22:40:07+00:00 | @RealDanODowd @WholeMarsBlog @Tesla Hahaha why... | TSLA | Tesla, Inc. | 2022 |
| 4 | 2022-09-29 22:27:05+00:00 | @RealDanODowd @Tesla Stop trying to kill kids,... | TSLA | Tesla, Inc. | 2022 |
| ... | ... | ... | ... | ... | ... |

Figure 9: Tweet Data

In this project, 80793 tweet data from 25 companies are used.



```
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Date          80793 non-null  datetime64[ns, UTC]
 1   Tweet         80793 non-null  object
 2   Stock Name    80793 non-null  object
 3   Company Name  80793 non-null  object
```

```
Date            64424
Tweet           64479
Stock Name         25
Company Name       25
```

Figure 10: Tweet Data Info

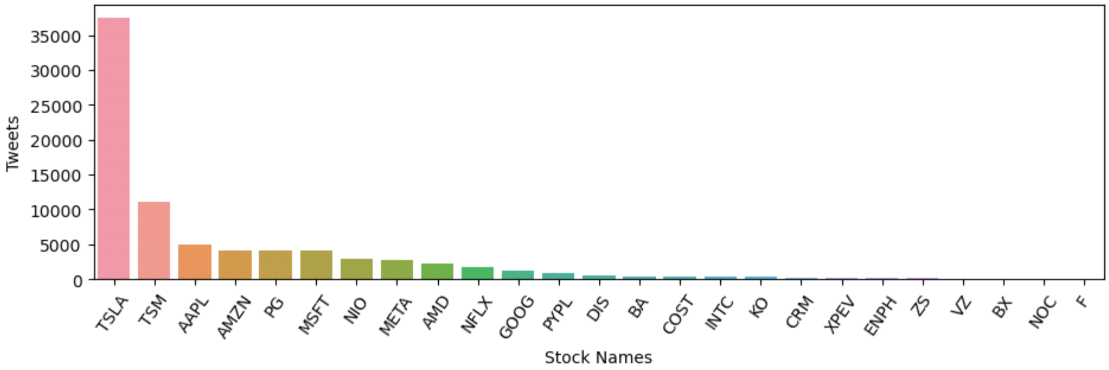
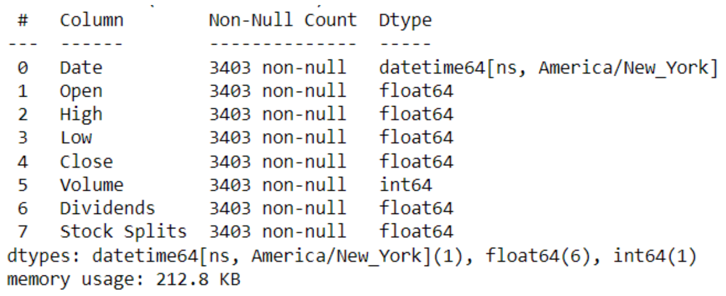


Figure 11: Tweet Amount by Stocks

Data augmentation is an essential element for obtaining better results in Deep Learning models, while using large size of data are used, it is highly possible to get a better prediction results. **[5]** Therefore, it might be reasonable to select the stock with the largest dataset to achieve more accurate results in prediction algorithms. In this case it will be Tesla (TSLA) stock.



```
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Date          3403 non-null   datetime64[ns, America/New_York]
 1   Open          3403 non-null   float64
 2   High          3403 non-null   float64
 3   Low           3403 non-null   float64
 4   Close         3403 non-null   float64
 5   Volume        3403 non-null   int64
 6   Dividends     3403 non-null   float64
 7   Stock Splits  3403 non-null   float64
dtypes: datetime64[ns, America/New_York](1), float64(6), int64(1)
memory usage: 212.8 KB
```

Figure 12: Stock Data Info

For stock data, Yahoo Finance library is used. The columns are shown above are Date, Open, High, Low and Close prices, Volume and Dividends and Stock splits. The only 5 of these columns used in scope of this study. These are Date, Open, High, Low and Close.

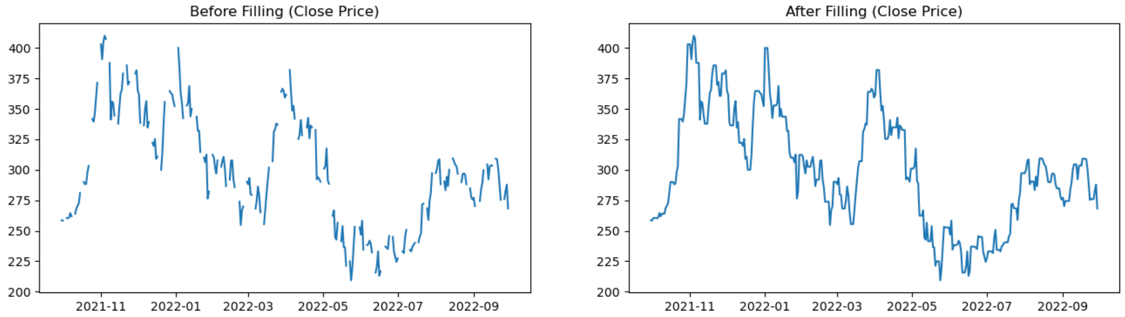


Figure 13: Stock Prices for TSLA & backward filling

Before moving on to natural language processing, we can quantitatively analyze the data and compare their temporal distributions. This way, we can see if there are any increases or decreases in significant time intervals. Due to the smaller size of this dataset compared to other text datasets, this study will be focus on longer-term predictions such as daily, weekly, and monthly instead of testing on minute or hourly prediction models. The graph below shows the daily increase or decrease of a stock and the number of tweets related to it.

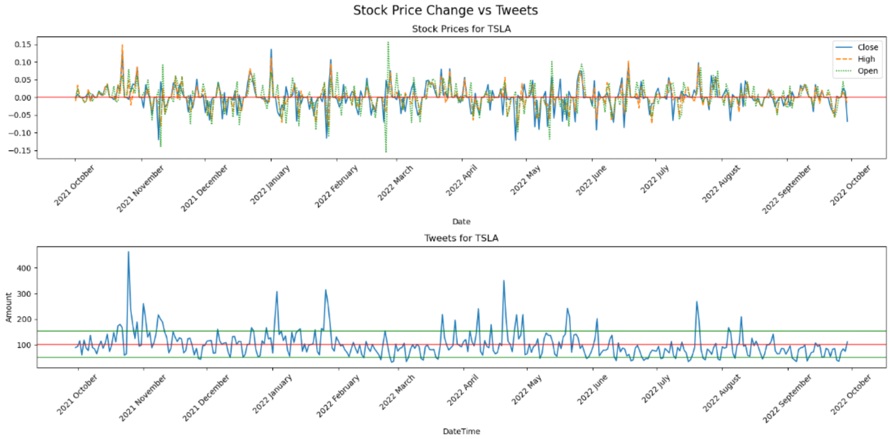


Figure 14: Stock Price Change vs Tweets

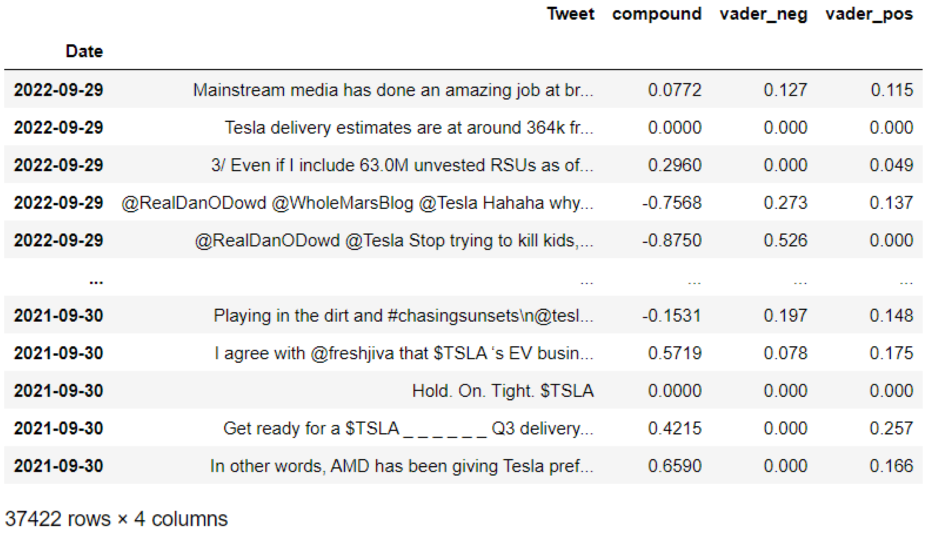
In Amount of tweets figure, red line is used to show mean of tweet amounts while green is the standard deviation. While the stock prices are highly increased or decreased, it can be observed that the amount of tweets in that time range have deviated from the mean.

## 4.2 Inspecting The Sentimental Scores of Tweets

### 4.2.1 Using Vader (Valance Aware Dictionary and Sentiment Reasoner) Model

The Vader algorithm is a specialized approach designed for sentiment analysis of textual data. VADER sentiment analysis is primarily based on a lexicon that pairs lexical features with emotion intensity measures called sentiment scores. The sentiment score of a piece of text can be obtained by summing the intensities of each word in the text.

Due to its focus on the English language and its reliance on specific rule-based approaches, VADER has limitations compared to algorithms like RoBERTa, which are multilingual and can be used across a wide range of languages. However, research shows that in terms of classification, VADER is much stronger than human competitors. While VADER achieves an F1 score of 0.96, human predictions only achieve an F1 score of 0.84. **[6]** Therefore, due to its strength and speed, using the VADER algorithm as an initial model might be more appropriate.



| | Tweet | compound | vader_neg | vader_pos |
|---|---|---|---|---|
| **Date** | | | | |
| **2022-09-29** | Mainstream media has done an amazing job at br... | 0.0772 | 0.127 | 0.115 |
| **2022-09-29** | Tesla delivery estimates are at around 364k fr... | 0.0000 | 0.000 | 0.000 |
| **2022-09-29** | 3/ Even if I include 63.0M unvested RSUs as of... | 0.2960 | 0.000 | 0.049 |
| **2022-09-29** | @RealDanODowd @WholeMarsBlog @Tesla Hahaha why... | -0.7568 | 0.273 | 0.137 |
| **2022-09-29** | @RealDanODowd @Tesla Stop trying to kill kids,... | -0.8750 | 0.526 | 0.000 |
| **...** | ... | ... | ... | ... |
| **2021-09-30** | Playing in the dirt and #chasingsunsets\n@tesl... | -0.1531 | 0.197 | 0.148 |
| **2021-09-30** | I agree with @freshjiva that $TSLA 's EV busin... | 0.5719 | 0.078 | 0.175 |
| **2021-09-30** | Hold. On. Tight. $TSLA | 0.0000 | 0.000 | 0.000 |
| **2021-09-30** | Get ready for a $TSLA _ _ _ _ _ _ Q3 delivery... | 0.4215 | 0.000 | 0.257 |
| **2021-09-30** | In other words, AMD has been giving Tesla pref... | 0.6590 | 0.000 | 0.166 |

37422 rows × 4 columns

Figure 15: Vader Sentimental Scores

The Vader model provides several scores that can be used to analyze the sentiment of a piece of text

**neg_score:** The neg score represents the negativity or pessimism of the text.

**neu_score:** The neu score represents the neutrality or lack of sentiment in the text.

**pos_score:** The pos score indicates the positivity or optimism of the text. It indicates the strength or intensity of positive sentiment expressed in the text.

**compound_score:** This score represents the balance between negative, neutral, and positive scores in the text. A score greater than 0 suggests a positive sentiment, while a score less than 0 indicates a negative sentiment

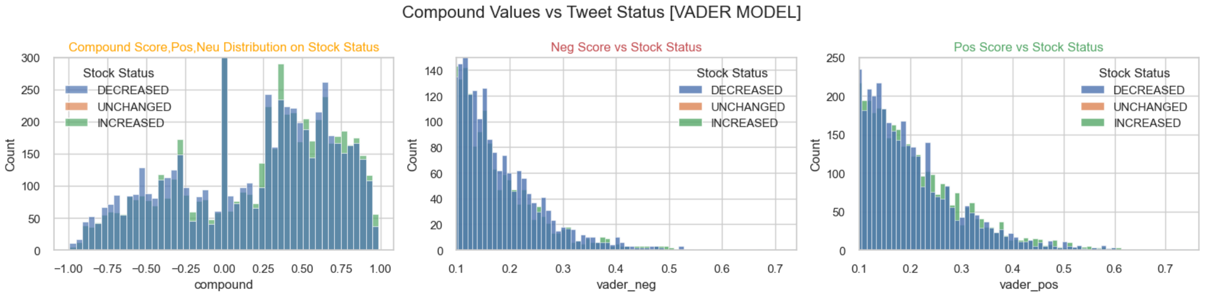


Figure 16: Tweet Status and Stock Distribution for VADER

In cases where the compound score is high, there is often a parallel increase in stock prices, while in cases where the compound score is low, parallel decrease in stock prices.

Analyzing the Vader model scores in relation to price changes:

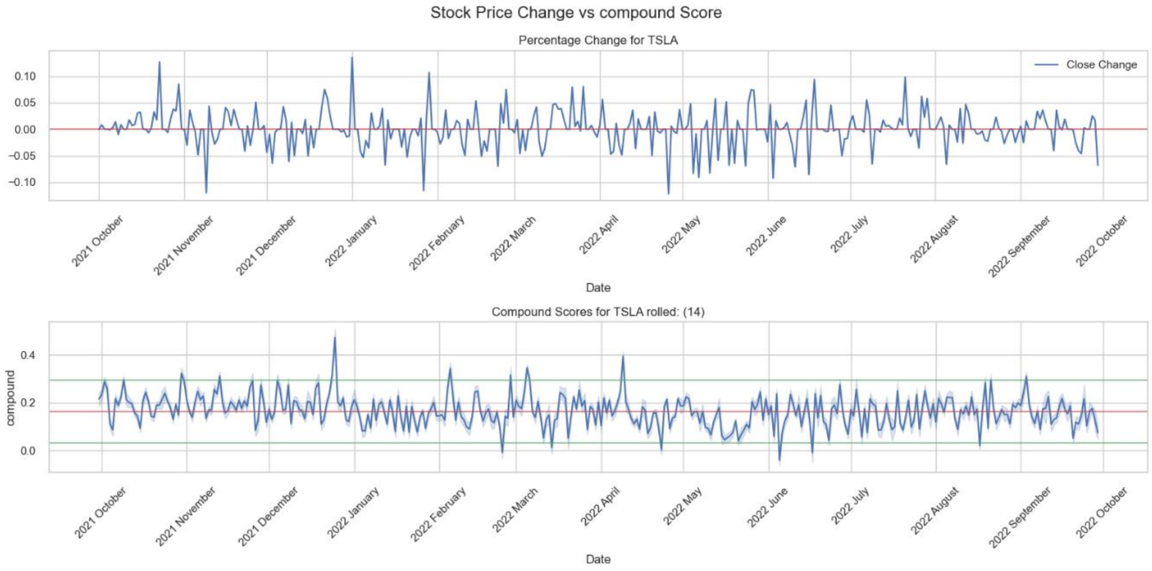
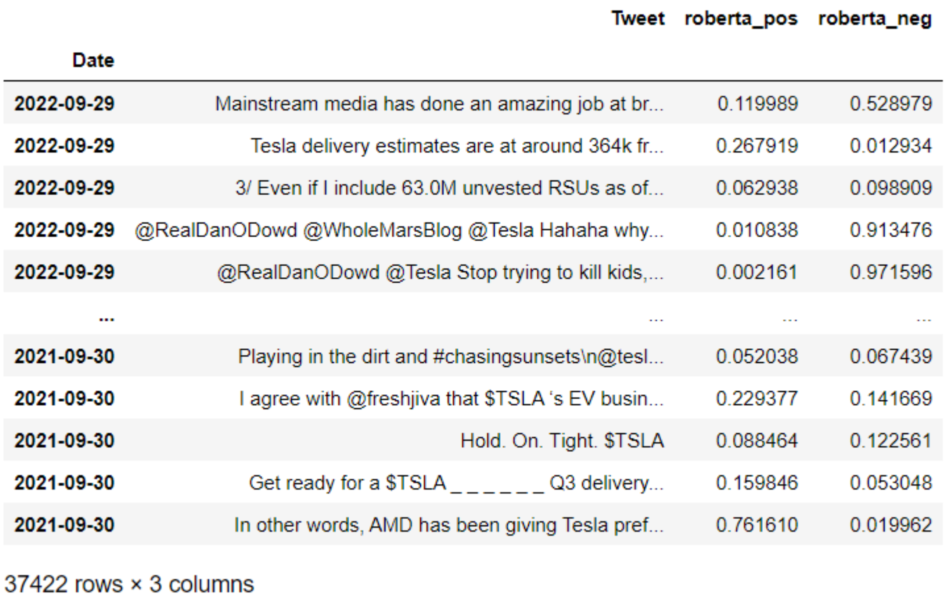


Figure 17: Rolled Price and Compound Scores

There appears to be a higher correlation between the weekly 14 days tweet scores and closing price data. By subjecting these data sets to a Pearson correlation test, we can obtain a more realistic result.

### 4.2.2 RoBERTa (Robustly Optimized BERT Approach) Model

The RoBERTa algorithm uses a multi-head attention mechanism and is pre-trained on large text datasets. One of the most significant differences from the BERT (Bidirectional Encoder Representations from Transformers) model is that in BERT, predictions are made on randomly masked data, whereas in the RoBERTa model, the data is trained in a more straightforward manner on larger datasets where the data can be repeated. Analyzing the RoBERTa model scores in relation to price changes:



| | Tweet | roberta_pos | roberta_neg |
|---|---|---|---|
| **Date** | | | |
| **2022-09-29** | Mainstream media has done an amazing job at br... | 0.119989 | 0.528979 |
| **2022-09-29** | Tesla delivery estimates are at around 364k fr... | 0.267919 | 0.012934 |
| **2022-09-29** | 3/ Even if I include 63.0M unvested RSUs as of... | 0.062938 | 0.098909 |
| **2022-09-29** | @RealDanODowd @WholeMarsBlog @Tesla Hahaha why... | 0.010838 | 0.913476 |
| **2022-09-29** | @RealDanODowd @Tesla Stop trying to kill kids,... | 0.002161 | 0.971596 |
| **...** | ... | ... | ... |
| **2021-09-30** | Playing in the dirt and #chasingsunsets\n@tesl... | 0.052038 | 0.067439 |
| **2021-09-30** | I agree with @freshjiva that $TSLA 's EV busin... | 0.229377 | 0.141669 |
| **2021-09-30** | Hold. On. Tight. $TSLA | 0.088464 | 0.122561 |
| **2021-09-30** | Get ready for a $TSLA _ _ _ _ _ _ Q3 delivery... | 0.159846 | 0.053048 |
| **2021-09-30** | In other words, AMD has been giving Tesla pref... | 0.761610 | 0.019962 |

37422 rows × 3 columns

Figure 18: RoBERTa Sentimental Scores

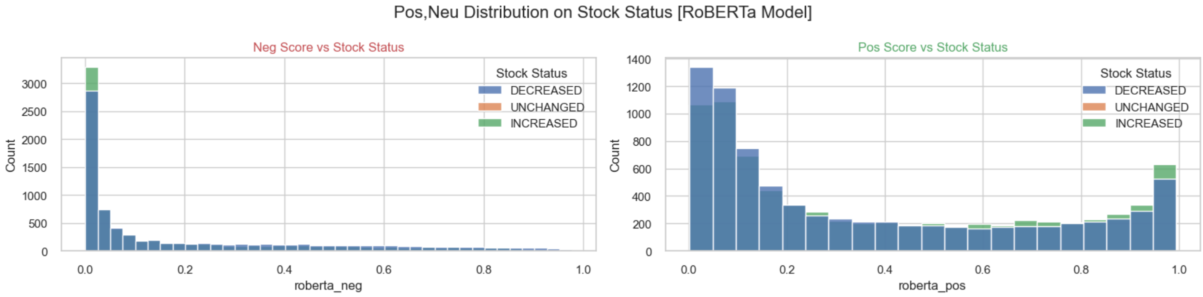


Figure 19: Tweet Status and Stock Distribution for RoBERTa

When using VADER's compound score feature for distributions, it's more decisive. Similarly, in positive and negative RoBERTa scores, we can see a parallel with prices, like in VADER. For example, in the figure on the right, higher positive scores correlate with greater price increases, while in the figure on the left, higher negative scores correspond to larger decreases in prices.

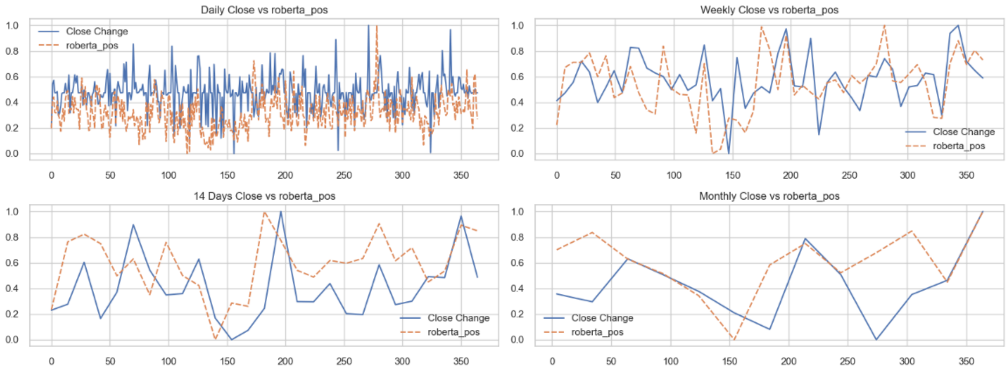


Figure 20: Average Price and Robert Pos Score

Compared to the compound metric, it appears that the positive metrics in Roberta are more correlated with weekly and 14-day price changes. In addition to using 7-day and 14-day timeframes in our prediction algorithms, incorporating positive Roberta scores may also be more effective.

## 4.3 Pearson Correlation Test

A statistic which detects the correlation between features is the Pearson correlation coefficient. As such, it is related with certain distributional assumptions. For the Pearson, a bivariate normal distribution—that is, a linear relationship between X and Y—is specifically assumed. The robustness of Pearson correlation is compromised when linearity is violated. **[7]** The Pearson correlation is not robust, especially when linearity is violated. A single outlier can significantly alter the value of the correlation coefficient (ρ), which is the typical estimate used.

The metrics obtained from the Pearson correlation can indicate how strong the correlation is between the metrics and price changes. A high value of the Pearson coefficient indicates a positive correlation, while a value close to zero suggests no correlation. The p-value indicates whether this relationship is due to randomness. When the p-value is less than 0.05, the distribution is statistically significant, and the null hypothesis can be rejected. In this case, distributions with a high Pearson coefficient and a low p-value are more likely to provide consistent results.

### 4.3.1 Correlation for Vader Compound Score and Price Change:

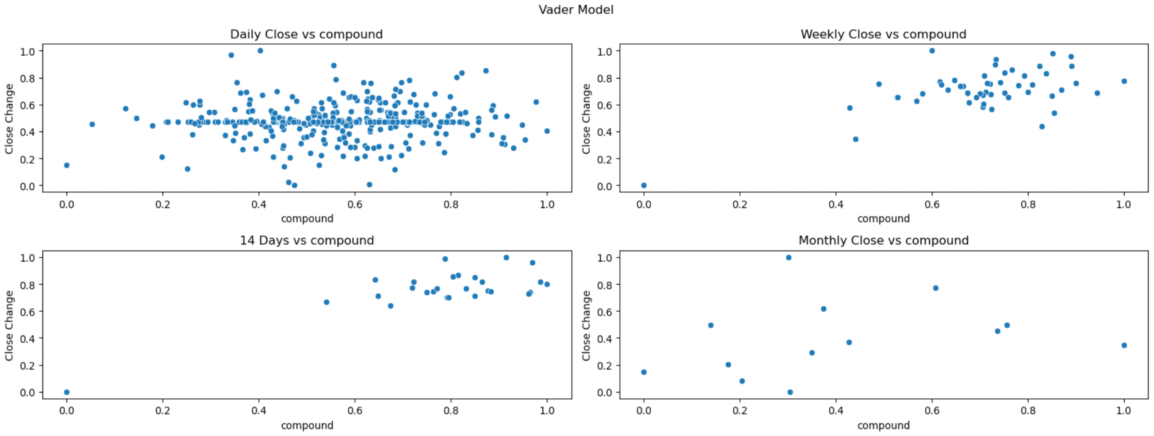
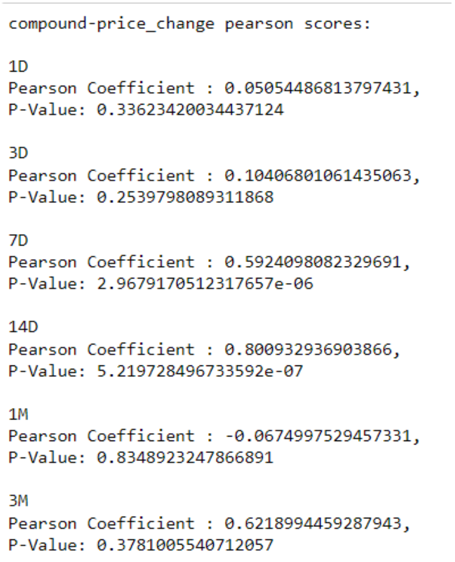


Figure 21: Close Change by Compound Score



```
compound-price_change pearson scores:

1D
Pearson Coefficient : 0.05054486813797431,
P-Value: 0.33623420034437124

3D
Pearson Coefficient : 0.10406801061435063,
P-Value: 0.2539798089311868

7D
Pearson Coefficient : 0.5924098082329691,
P-Value: 2.9679170512317657e-06

14D
Pearson Coefficient : 0.800932936903866,
P-Value: 5.219728496733592e-07

1M
Pearson Coefficient : -0.0674997529457331,
P-Value: 0.8348923247866891

3M
Pearson Coefficient : 0.6218994459287943,
P-Value: 0.3781005540712057
```

Figure 22: Pearson Scores for Vader

## 4.3.2 Correlation for RoBERTa Positive Score and Price Change:

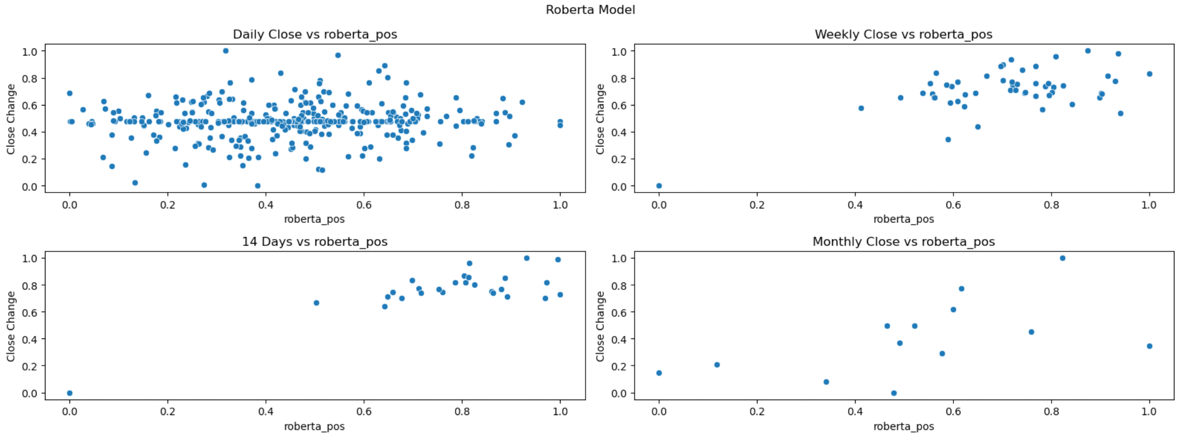
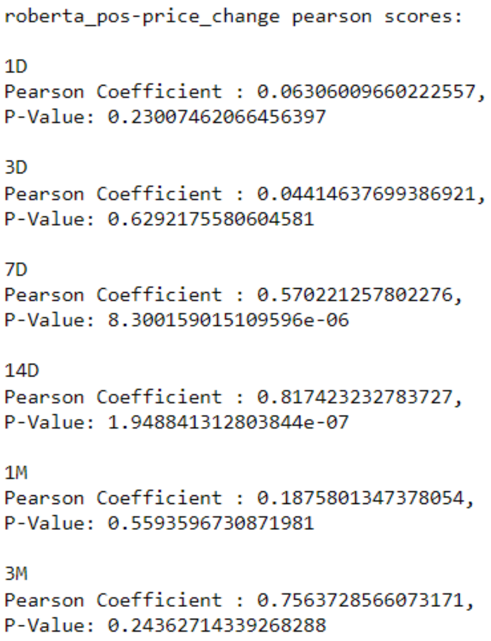


Figure 23: Close Change by Roberta Pos Score



```
roberta_pos-price_change pearson scores:

1D
Pearson Coefficient : 0.06306009660222557,
P-Value: 0.23007462066456397

3D
Pearson Coefficient : 0.04414637699386921,
P-Value: 0.6292175580604581

7D
Pearson Coefficient : 0.570221257802276,
P-Value: 8.300159015109596e-06

14D
Pearson Coefficient : 0.817423232783727,
P-Value: 1.948841312803844e-07

1M
Pearson Coefficient : 0.1875801347378054,
P-Value: 0.5593596730871981

3M
Pearson Coefficient : 0.7563728566073171,
P-Value: 0.24362714339268288
```

Figure 24: Pearson Scores for RoBERTa

As shown by the results, higher Pearson Coefficient and lower p-value is observed in weekly and 14 day up-sampled price data. The prediction on weekly time ranges could be statistically meaningful. The Pearson coefficient for pos_score in RoBERTa is slightly higher than the compound score in Vader

## 4.4 Most Frequent Keywords

In contrast to general compliments, the higher count of words like "growth," "safety," "profits," and "progress" compared to words such as "lost," "fraud," and "debt" shows more positive feelings towards a company. Also, there are more positive financial terms, with words like 'bull,', 'long', 'earnings,' 'bullish,' 'green,' and 'profits' leading, while negative terms start with 'bear,' 'short,' 'loss,' 'bearish,' 'broke,' 'risk,' 'red,' 'inflation.' as shown in the figures 25 & 26.These words can be examined in more detail according to monthly distributions and score averages to predict the company's future prices.

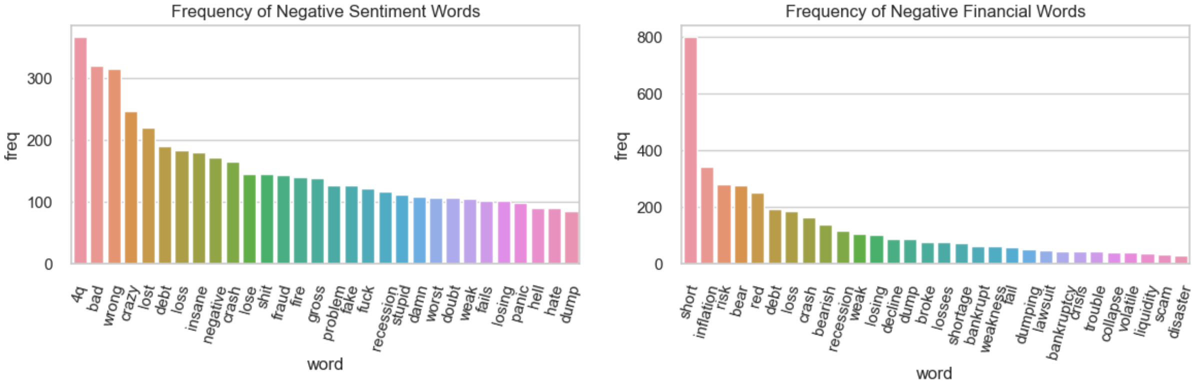


Figure 25: Negative Keyword Distributions

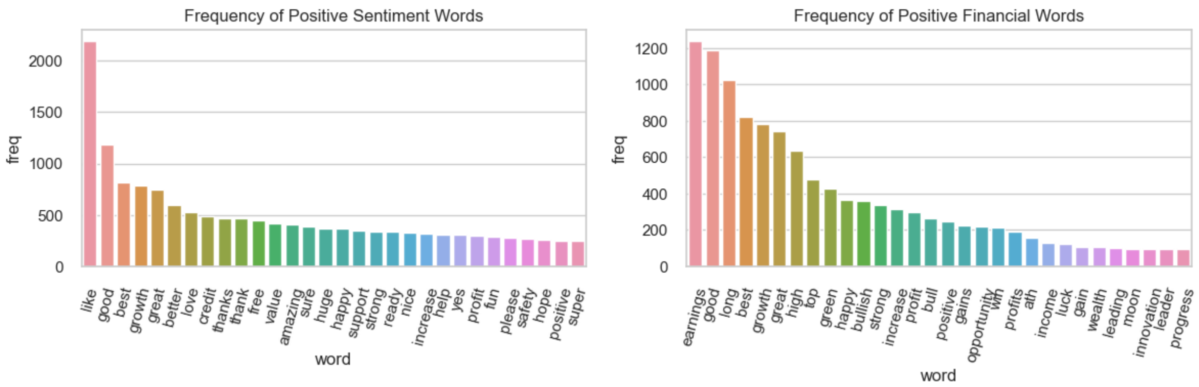


Figure 26: Positive Keyword Distributions

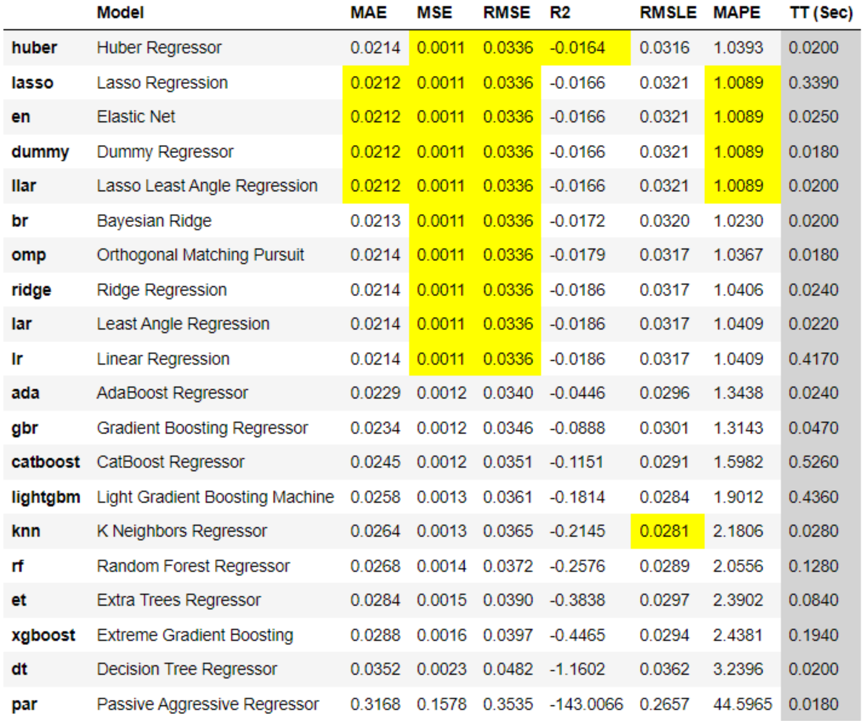
# 5. Predicting Stock Price Change using Sentimental Scores

## 5.1 Approaching as a Numerical Problem

Previously, we discovered a connection between prices and sentiment scores. However, the ultimate goal of sentiment scores is to catching signals before price increases or decreases occur. Therefore, instead of examining existing data, we aim to develop a predictive model based on sentiment scores. In this section, we will compare sentiment scores in various regression models and examine their forward-looking predictions.

## 5.2 Price Prediction Using RoBERTa Positive and Negative Scores

### 5.1.1.1 Model Scores



| | Model | MAE | MSE | RMSE | R2 | RMSLE | MAPE | TT (Sec) |
|---|---|---|---|---|---|---|---|---|
| **huber** | Huber Regressor | 0.0214 | 0.0011 | 0.0336 | -0.0164 | 0.0316 | 1.0393 | 0.0200 |
| **lasso** | Lasso Regression | 0.0212 | 0.0011 | 0.0336 | -0.0166 | 0.0321 | 1.0089 | 0.3390 |
| **en** | Elastic Net | 0.0212 | 0.0011 | 0.0336 | -0.0166 | 0.0321 | 1.0089 | 0.0250 |
| **dummy** | Dummy Regressor | 0.0212 | 0.0011 | 0.0336 | -0.0166 | 0.0321 | 1.0089 | 0.0180 |
| **llar** | Lasso Least Angle Regression | 0.0212 | 0.0011 | 0.0336 | -0.0166 | 0.0321 | 1.0089 | 0.0200 |
| **br** | Bayesian Ridge | 0.0213 | 0.0011 | 0.0336 | -0.0172 | 0.0320 | 1.0230 | 0.0200 |
| **omp** | Orthogonal Matching Pursuit | 0.0214 | 0.0011 | 0.0336 | -0.0179 | 0.0317 | 1.0367 | 0.0180 |
| **ridge** | Ridge Regression | 0.0214 | 0.0011 | 0.0336 | -0.0186 | 0.0317 | 1.0406 | 0.0240 |
| **lar** | Least Angle Regression | 0.0214 | 0.0011 | 0.0336 | -0.0186 | 0.0317 | 1.0409 | 0.0220 |
| **lr** | Linear Regression | 0.0214 | 0.0011 | 0.0336 | -0.0186 | 0.0317 | 1.0409 | 0.4170 |
| **ada** | AdaBoost Regressor | 0.0229 | 0.0012 | 0.0340 | -0.0446 | 0.0296 | 1.3438 | 0.0240 |
| **gbr** | Gradient Boosting Regressor | 0.0234 | 0.0012 | 0.0346 | -0.0888 | 0.0301 | 1.3143 | 0.0470 |
| **catboost** | CatBoost Regressor | 0.0245 | 0.0012 | 0.0351 | -0.1151 | 0.0291 | 1.5982 | 0.5260 |
| **lightgbm** | Light Gradient Boosting Machine | 0.0258 | 0.0013 | 0.0361 | -0.1814 | 0.0284 | 1.9012 | 0.4360 |
| **knn** | K Neighbors Regressor | 0.0264 | 0.0013 | 0.0365 | -0.2145 | 0.0281 | 2.1806 | 0.0280 |
| **rf** | Random Forest Regressor | 0.0268 | 0.0014 | 0.0372 | -0.2576 | 0.0289 | 2.0556 | 0.1280 |
| **et** | Extra Trees Regressor | 0.0284 | 0.0015 | 0.0390 | -0.3838 | 0.0297 | 2.3902 | 0.0840 |
| **xgboost** | Extreme Gradient Boosting | 0.0288 | 0.0016 | 0.0397 | -0.4465 | 0.0294 | 2.4381 | 0.1940 |
| **dt** | Decision Tree Regressor | 0.0352 | 0.0023 | 0.0482 | -1.1602 | 0.0362 | 3.2396 | 0.0200 |
| **par** | Passive Aggressive Regressor | 0.3168 | 0.1578 | 0.3535 | -143.0066 | 0.2657 | 44.5965 | 0.0180 |

Figure 27: RoBERTa Regression Model Scores

### *5.1.1.2 Feature Importance*

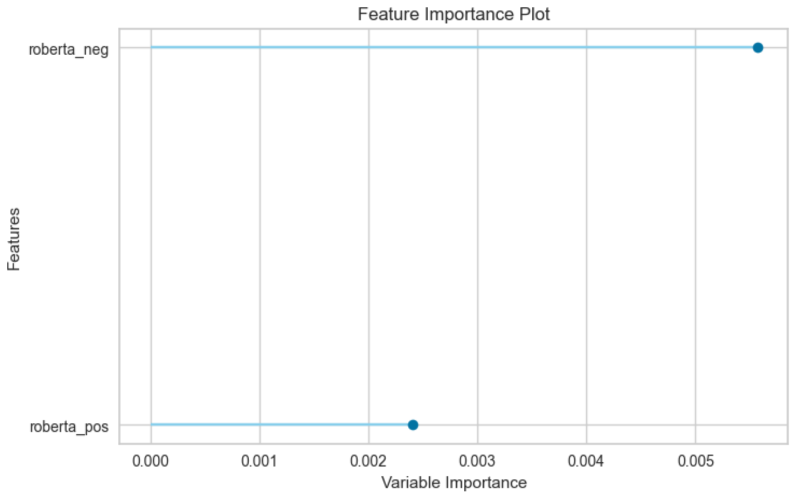


Figure 28: RoBERTa Feature Importance

Negative sentiments have a high impact on price prediction. This is parallel with the studies on stock market psychology. The results on social media platform StockTwits show that user sentiment largely fluctuates with stock returns. The effect is much stronger on negative sentiment than on positive sentiment. That is, negative sentiment becomes stronger after negative returns than positive sentiment does after positive returns. 1% increase in negative sentiment leads to (0.03%) decrease in stock returns. Additionally, in the article by Tetlock et al. [8], which explores possible correlations between the media and the stock market using information from the Wall Street Journal, it is found that high pessimism causes downward pressure on market prices

### *5.1.1.3 Train and Test R2 Scores and Distributions*

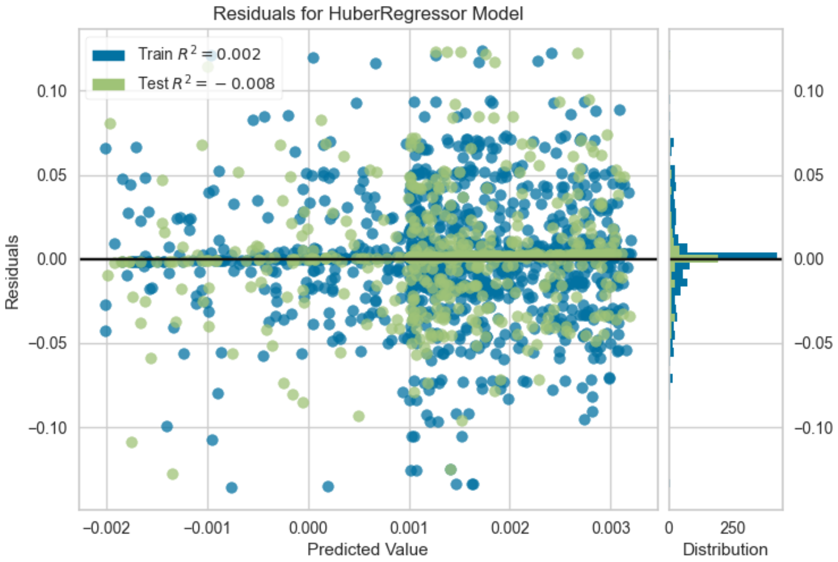


Figure 29: RoBERTa Model Residuals

### *5.1.1.4 Price Pattern on Test Data*

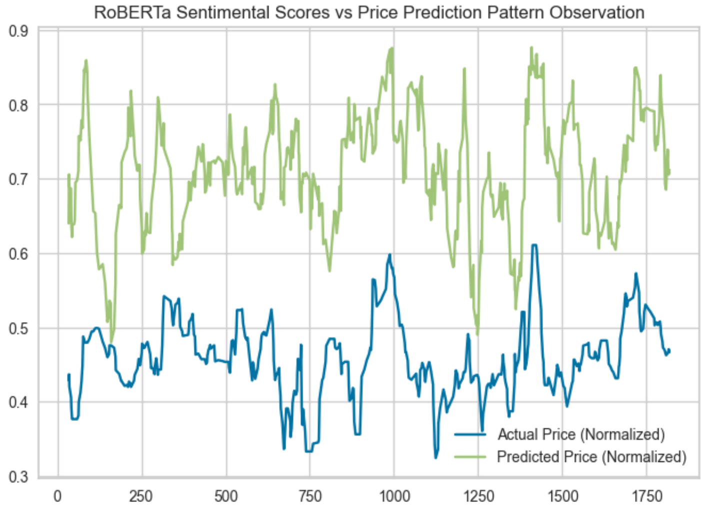
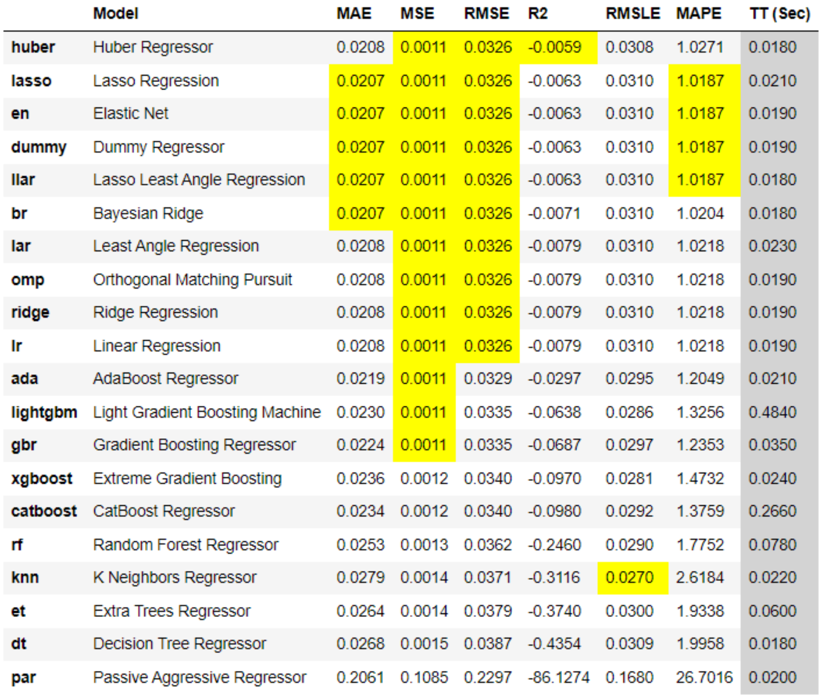


Figure 30: Price Pattern using RoBERTa

## 5.1.2 Price Prediction Using VADER Compound Score

### *5.1.2.1 Model Scores*



| | Model | MAE | MSE | RMSE | R2 | RMSLE | MAPE | TT (Sec) |
|---|---|---|---|---|---|---|---|---|
| **huber** | Huber Regressor | 0.0208 | 0.0011 | 0.0326 | -0.0059 | 0.0308 | 1.0271 | 0.0180 |
| **lasso** | Lasso Regression | 0.0207 | 0.0011 | 0.0326 | -0.0063 | 0.0310 | 1.0187 | 0.0210 |
| **en** | Elastic Net | 0.0207 | 0.0011 | 0.0326 | -0.0063 | 0.0310 | 1.0187 | 0.0190 |
| **dummy** | Dummy Regressor | 0.0207 | 0.0011 | 0.0326 | -0.0063 | 0.0310 | 1.0187 | 0.0190 |
| **llar** | Lasso Least Angle Regression | 0.0207 | 0.0011 | 0.0326 | -0.0063 | 0.0310 | 1.0187 | 0.0180 |
| **br** | Bayesian Ridge | 0.0207 | 0.0011 | 0.0326 | -0.0071 | 0.0310 | 1.0204 | 0.0180 |
| **lar** | Least Angle Regression | 0.0208 | 0.0011 | 0.0326 | -0.0079 | 0.0310 | 1.0218 | 0.0230 |
| **omp** | Orthogonal Matching Pursuit | 0.0208 | 0.0011 | 0.0326 | -0.0079 | 0.0310 | 1.0218 | 0.0190 |
| **ridge** | Ridge Regression | 0.0208 | 0.0011 | 0.0326 | -0.0079 | 0.0310 | 1.0218 | 0.0190 |
| **lr** | Linear Regression | 0.0208 | 0.0011 | 0.0326 | -0.0079 | 0.0310 | 1.0218 | 0.0190 |
| **ada** | AdaBoost Regressor | 0.0219 | 0.0011 | 0.0329 | -0.0297 | 0.0295 | 1.2049 | 0.0210 |
| **lightgbm** | Light Gradient Boosting Machine | 0.0230 | 0.0011 | 0.0335 | -0.0638 | 0.0286 | 1.3256 | 0.4840 |
| **gbr** | Gradient Boosting Regressor | 0.0224 | 0.0011 | 0.0335 | -0.0687 | 0.0297 | 1.2353 | 0.0350 |
| **xgboost** | Extreme Gradient Boosting | 0.0236 | 0.0012 | 0.0340 | -0.0970 | 0.0281 | 1.4732 | 0.0240 |
| **catboost** | CatBoost Regressor | 0.0234 | 0.0012 | 0.0340 | -0.0980 | 0.0292 | 1.3759 | 0.2660 |
| **rf** | Random Forest Regressor | 0.0253 | 0.0013 | 0.0362 | -0.2460 | 0.0290 | 1.7752 | 0.0780 |
| **knn** | K Neighbors Regressor | 0.0279 | 0.0014 | 0.0371 | -0.3116 | 0.0270 | 2.6184 | 0.0220 |
| **et** | Extra Trees Regressor | 0.0264 | 0.0014 | 0.0379 | -0.3740 | 0.0300 | 1.9338 | 0.0600 |
| **dt** | Decision Tree Regressor | 0.0268 | 0.0015 | 0.0387 | -0.4354 | 0.0309 | 1.9958 | 0.0180 |
| **par** | Passive Aggressive Regressor | 0.2061 | 0.1085 | 0.2297 | -86.1274 | 0.1680 | 26.7016 | 0.0200 |

Figure 31: Vader Regression Model Scores

### *5.1.2.2 Train and Test R2 Scores and Distributions*

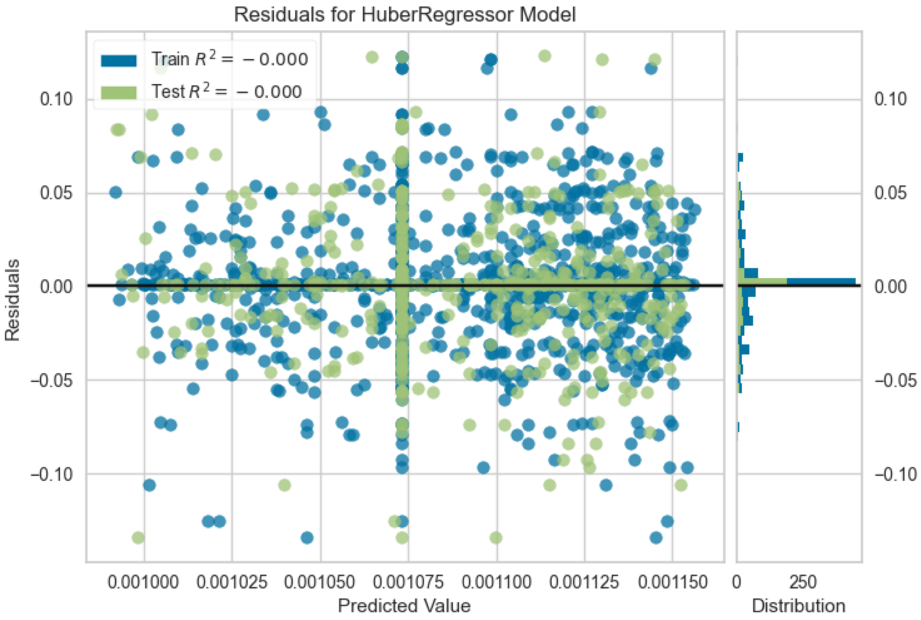


Figure 32: Vader Model Residues

### *5.1.2.3 Price Pattern on Test Data*

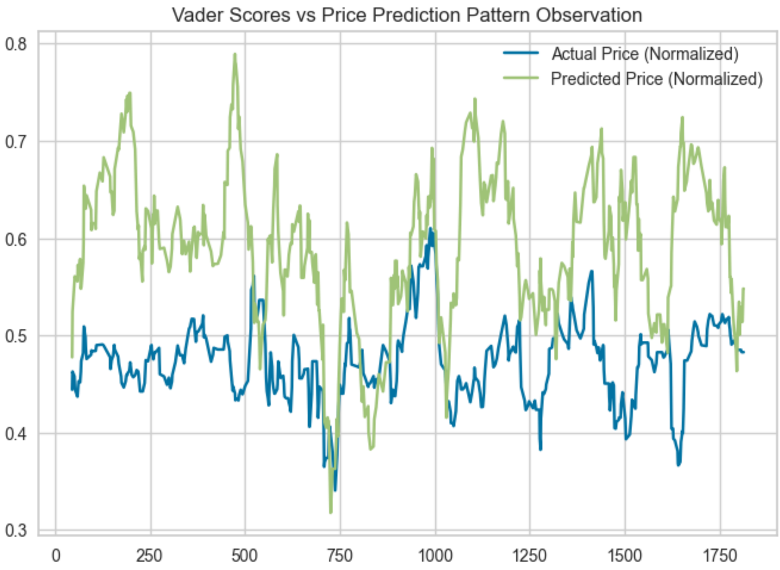


Figure 33: Price Pattern using RoBERTa

## 5.2 Approaching as a Classification Problem

### 5.2.1 Inspecting Classified Price and Tweet Patterns

Neural networks, particularly Long Short-Term Memory (LSTM) networks, still are powerful tools for stock price prediction.Therefore, instead of only focusing on price changes, transforming the problem into a classification task can be more informative, helping identify signals of stock price increase and decrease. To gain deeper insights, the data is visualized in a heatmap, considering various time frames. As previously mentioned, time scales of 7 days and 14 days prove to be particularly effective for prediction and classification. In the heatmap below, the left section presents classified stock prices, distinguishing between increases and decreases. On the right, we examine tweet status alongside the average of compound scores, Roberta positive scores, and Roberta negative scores over the specified time frames. This approach enhances our understanding of the relationship between stock and tweet statuses.

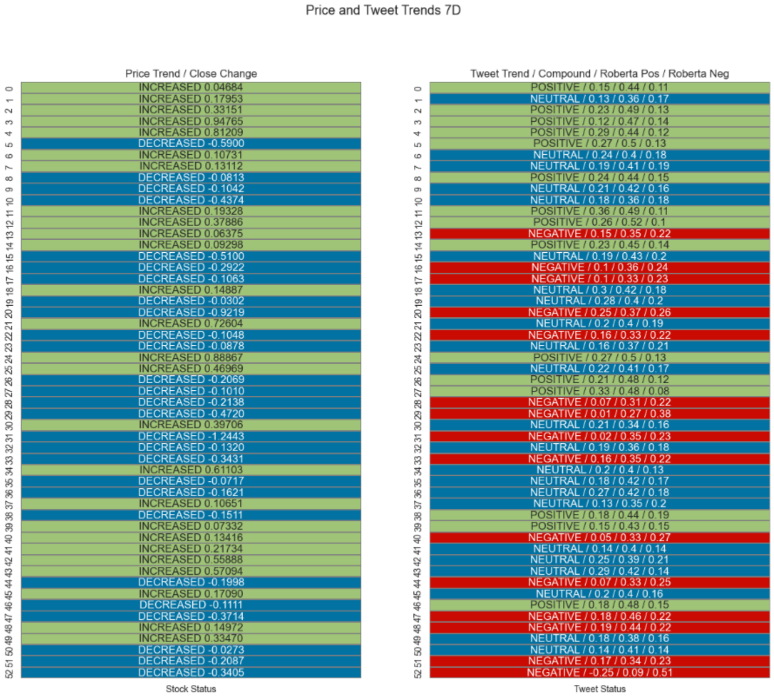
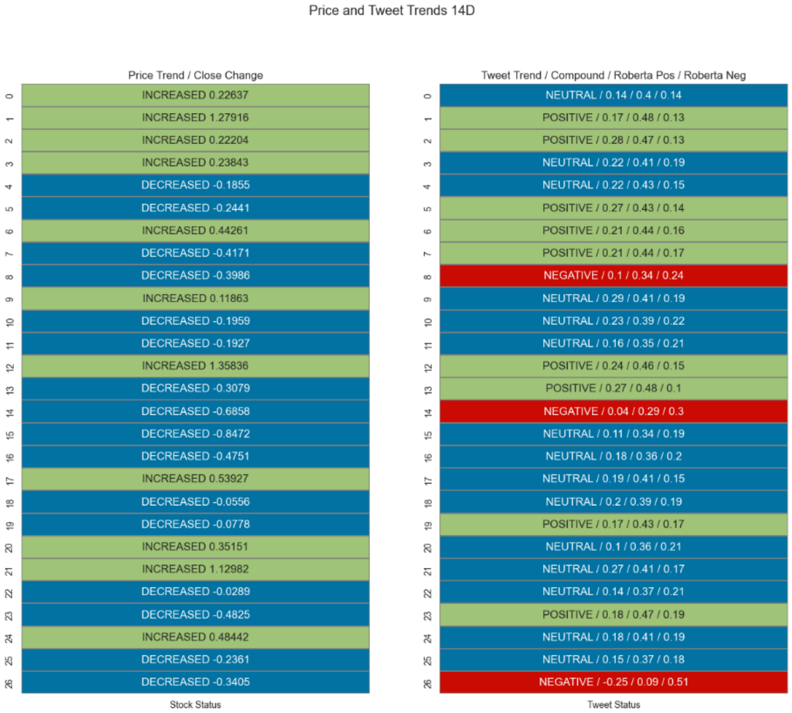


Figure 34: 7 Days Price Vs Stock Pattern



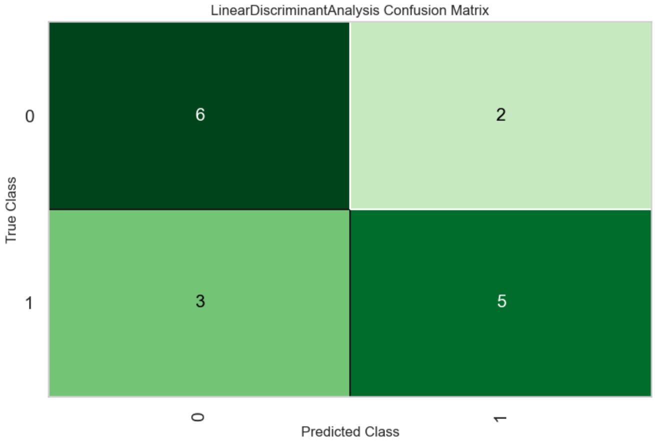
Price and Tweet Trends 14D

|  | Price Trend / Close Change | Tweet Trend / Compound / Roberta Pos / Roberta Neg |
|---|---|---|
| 0 | INCREASED 0.22637 | NEUTRAL / 0.14 / 0.4 / 0.14 |
| 1 | INCREASED 1.27916 | POSITIVE / 0.17 / 0.48 / 0.13 |
| 2 | INCREASED 0.22204 | POSITIVE / 0.28 / 0.47 / 0.13 |
| 3 | INCREASED 0.23843 | NEUTRAL / 0.22 / 0.41 / 0.19 |
| 4 | DECREASED -0.1855 | NEUTRAL / 0.22 / 0.43 / 0.15 |
| 5 | DECREASED -0.2441 | POSITIVE / 0.27 / 0.43 / 0.14 |
| 6 | INCREASED 0.44261 | POSITIVE / 0.21 / 0.44 / 0.16 |
| 7 | DECREASED -0.4171 | POSITIVE / 0.21 / 0.44 / 0.17 |
| 8 | DECREASED -0.3986 | NEGATIVE / 0.1 / 0.34 / 0.24 |
| 9 | INCREASED 0.11863 | NEUTRAL / 0.29 / 0.41 / 0.19 |
| 10 | DECREASED -0.1959 | NEUTRAL / 0.23 / 0.39 / 0.22 |
| 11 | DECREASED -0.1927 | NEUTRAL / 0.16 / 0.35 / 0.21 |
| 12 | INCREASED 1.35836 | POSITIVE / 0.24 / 0.46 / 0.15 |
| 13 | DECREASED -0.3079 | POSITIVE / 0.27 / 0.48 / 0.1 |
| 14 | DECREASED -0.6858 | NEGATIVE / 0.04 / 0.29 / 0.3 |
| 15 | DECREASED -0.8472 | NEUTRAL / 0.11 / 0.34 / 0.19 |
| 16 | DECREASED -0.4751 | NEUTRAL / 0.18 / 0.36 / 0.2 |
| 17 | INCREASED 0.53927 | NEUTRAL / 0.19 / 0.41 / 0.15 |
| 18 | DECREASED -0.0556 | NEUTRAL / 0.2 / 0.39 / 0.19 |
| 19 | DECREASED -0.0778 | POSITIVE / 0.17 / 0.43 / 0.17 |
| 20 | INCREASED 0.35151 | NEUTRAL / 0.1 / 0.36 / 0.21 |
| 21 | INCREASED 1.12982 | NEUTRAL / 0.27 / 0.41 / 0.17 |
| 22 | DECREASED -0.0289 | NEUTRAL / 0.14 / 0.37 / 0.21 |
| 23 | DECREASED -0.4825 | POSITIVE / 0.18 / 0.47 / 0.19 |
| 24 | INCREASED 0.48442 | NEUTRAL / 0.18 / 0.41 / 0.19 |
| 25 | DECREASED -0.2361 | NEUTRAL / 0.15 / 0.37 / 0.18 |
| 26 | DECREASED -0.3405 | NEGATIVE / -0.25 / 0.09 / 0.51 |

Figure 35: 14 Days Price Vs Stock Pattern

### 5.2.2 Stock Status Prediction using Compound, Roberta Pos, Roberta Neg and Tweet Status Scores

As previously mentioned, the significant impact of Roberta's negative sentiment scores and the 7-14 days average score on price changes led us to use them in related models. Here 0 represent decreasing days and 1 represents increasing days. Unchanged days are eliminated from the data since it gives false positives in our model.



LinearDiscriminantAnalysis Confusion Matrix

| True Class \ Predicted Class | 0 | 1 |
|---|---|---|
| 0 | 6 | 2 |
| 1 | 3 | 5 |

Figure 36: 7-Days Stock Status Confusion Matrix using Roberta Neg

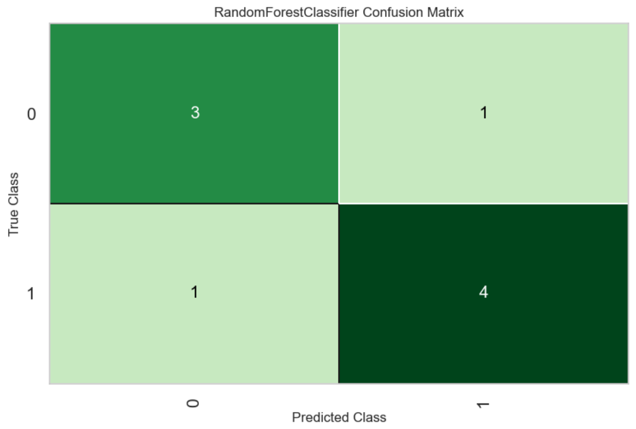


Figure 37: 14-Days Stock Status Confusion Matrix using Roberta Neg

## 6. Conclusion

This case study explored the relationship between Twitter sentiment and stock market price movements. It focused on analyzing financial text data and sentiment scores from tweets, revealing a significant correlation. Sentiment analysis models such as the Valance Aware Dictionary and Sentiment Reasoner model and the Robustly Optimized BERT Approach were pivotal in this investigation. The Pearson Correlation Test applied in the study confirmed the link between Twitter sentiment and stock market prices, highlighting the potential of sentiment analysis in financial forecasting.

However, the correlation observed does not imply causation, and the complexity of stock market movements, influenced by a wide range of factors beyond social media sentiment, must be considered. The findings suggest that Twitter sentiment is a significant factor but is just one aspect among many that affect the stock market. This study creates opportunities for future investigation into the precise relationship between social media sentiment and stock prices as well as how to incorporate it into more thorough market analysis and prediction techniques.

In summary, the case study underscores the growing importance of social media in financial markets and the need for a multifaceted approach in market analysis and prediction. It points towards the promising use of sentiment analysis in forecasting stock price changes, yet calls for a balanced strategy that encompasses diverse data sources and analytical methods.

Study Link: https://github.com/Gokay1904/Financial-Forecasting-with-AI-Transformer-Models

## 7. References

[1] Webster, J., & Kit, C. (1992). Tokenization as the initial phase in NLP. In *Proceedings of the 14th conference on Computational linguistics - Volume 4* (pp. 1106-1110). https://doi.org/10.3115/992424.992434

[2] Shazmeen, S. F. (2022). Sentiment Analysis of Financial News with Supervised Learning.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762v7*.

[4] Chen, P.-C., Tsai, H., Bhojanapalli, S., Chung, H. W., Chang, Y.-W., & Ferng, C.-S. (2021). A simple and effective positional encoding for transformers.

[5] Moreno-Barea, F. J., Jerez, J. M., & Franco, L. (2019). Improving classification accuracy using data augmentation on small data sets. *Expert Systems with Applications*, *160*, 113696. https://doi.org/10.1016/j.eswa.2020.113696

[6] Hutto, C. J., & Gilbert, E. (2015). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*.

[7] Wilcox, R. R. (2013). *Introduction to robust estimation and hypothesis testing* (3rd ed.). Academic Press. https://doi.org/10.1016/C2010-0-67044-1

[8] Tetlock, P. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, *62*(3), 1139-1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x

[9] Mishev, K., Gjorgjevikj, A., Vodenska, I., Chitkushev, L., & Trajanov, D. (2020). Evaluation of sentiment analysis in finance: From lexicons to transformers. *IEEE Access*. https://doi.org/10.1109/ACCESS.2020.3009626

[10] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[11] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

[12] Pimprikar, R., Ramachandra, S., & Senthilkuma, K. (2017). Use of machine learning algorithms and Twitter sentiment analysis for stock market prediction. *International Journal of Pure and Applied Mathematics*, *115*(6), 521-526.

[13] Mohan, S., Mullapudi, S., Sammeta, S., Vijayvergia, P., & Anastasiu, D. C. (2019). Stock price prediction using news sentiment analysis. In *Proceedings of the 5th IEEE International Conference on Big Data Service and Applications (BigDataService 2019) and the Workshops on Big Data in Water Resources, Environmental, and Hydraulic Engineering and on Medical, Health & Big Data Technology* (pp. 205–208). IEEE. https://doi.org/10.1109/BigDataService.2019.00035

[14] Bhardwaj, A., Narayan, Y., Vanraj, Pawan, & Dutta, M. (2015). Sentiment analysis for Indian stock market prediction using Sensex and Nifty. *Procedia Computer Science*, *70*, 85-91. https://doi.org/10.1016/j.procs.2015.10.043