NHH

Tesla Stock Trading Based on Elon Musk's Tweets

FIN11 – Term Paper

Candidate number: 138, 227, 244, 264

Word count: 5461 (excluding abstract, tables, references and appendix)

Abstract

This paper highlights the potential of using Elon Musk's tweets as a trigger for an algorithmic news trading strategy. To answer our research question, we analyzed Elon Musk's Twitter activity and the Tesla Stock price over the last 13 years. Drawing on our statistical analysis and the influential tweets we identified, we formulated and backtested a strategy to ascertain the potential of the concept, before proceeding to develop a classification model.

We proved that it is possible to make a profitable trading strategy based on the information in the tweets and the following market movements. However, developing an accurate text classification model for the tweets proved difficult. Our research identifies and suggests solutions for the classification model. The methodology outlined in this paper can be applied to other similar concepts or scaled up for larger projects.

Table of Contents

1 Introduction 3
2 Theory 5
2.1 News Traders 5
2.2 Automated Algorithmic Trading 5
2.3 Text Classification Model 6
2.4 Backtesting 7
3 Trading Strategy 8
3.1 Implementation 9
3.2 Evaluating the Strategy 9
4 Developing Our Classification Model 11
4.1 Data 11
4.2 Statistics 13
4.3 Classification Model 16
4.4 Training and Testing the Model 17
5 Weaknesses 18
5.1 Solutions 18
6 Optimization 20
7 Conclusion 20
8 Bibliography 22
9 Appendix 26

1 Introduction

Figure 1 Elon Musk's Twitter message on Aug 7. 2018

On August 7, 2018, Elon Musk posted this tweet which sent the Tesla stock jumping a ravishing 11% (The Economic Times, 2021). This was arguably intended as an obtuse joke on 420, a number popular amongst cannabis enthusiasts. The tweet indicated a sell price of $420 per share, which would be a substantial premium compared to the current trading price of $379[1] (CNBC, 2018). Eventually, the tweet found the interest of the U.S. Securities and Exchange Commission (SEC) opening a fraud trial: "Did Musk's tweets affect the firm's stock price?" (The Guardian, 2023). The trial concluded with a $40 million penalty divided between Musk himself and Tesla, alongside Musk stepping down as the chairman of the board (SEC, 2018).

Elon Musk is infamous for his controversial tweets about Tesla. Reaching over 160 million followers, Musk's statements seem to have sent the Tesla Stock both up and down over the last decade. Sometimes drastically. In May 2020, he tweeted "Tesla stock price is too high imo". The tweet sent the Tesla stock price down 10% (The Economic Times, 2021). One could argue if the information was valid or not, but even in a world with increasing fake news (Statista, 2023), it is probable that people are influenced by and act upon Tesla information from Musk's tweets.

Head of quantitative trading at NBIM, Peder Viervoll revealed that professional institutions already are engaged in trading based on signals such as these tweets (Viervoll, 2023). To demonstrate the performance of an algorithm capable of executing trades based on tweets, we conducted a simulation trading the Tesla stock based on Elon Musk's most influential tweets. Here is the result:

[1] Current market value was $64 billion, with a stock price of approximately $379. A stock price per share of $420 would price Tesla at $71 billion, an increase of nearly 11%.

Figure 2 Result from backtesting

The script executed a total of 90 trades. By trading on influential tweets from Elon Musk, the overall profit was 15,75%. The simulation shows that with a perfect engine and timing, it is possible to make a profitable trading algorithm based on Elon Musk's Twitter activity. Throughout the paper, we will detail the development of the trading strategy and how we identified the tweets in the simulation.

As four individuals with limited resources, we are intrigued by the challenge of trading on the signals from Musk's tweets. Hence our research question is:

How difficult is it to create an automated trading algorithm for Tesla stocks based on information from Elon Musk's tweets?

To address our research question, we will begin by presenting the necessary theoretical background and concepts essential for developing a trading algorithm. Next, we will outline our trading strategy and explain the implementation in Python. Subsequently, we will reflect on the influence of transaction costs and propose a metric for assessing the performance of the strategy.

The next phase involves developing our own classification model. In this section, we will describe the process of how we handled our data and conducted statistical analysis. This analysis will be the basis for our training and testing of the classification model's ability to categorize tweets. Next, we will discuss the model's weaknesses, propose potential solutions, and explore ways to optimize the strategy. Finally, we will put context to the results and draw conclusions from our findings.

2 Theory

To address the research question effectively, a solid theoretical foundation is essential. To trade based on tweets, it is crucial to comprehend the dynamics of news trading and understand the various forms of algorithmic trading. Further, a method for assessing the information in the tweets is essential. Therefore, we will provide a detailed explanation of text classification. Finally, we will present backtesting, which plays a very important role in algorithmic trading.

2.1 News Traders

Trading based on information is categorized as informed trading. Informed traders trade on existing, analyzed, and/or new information about an asset. The information might be private or public. There are three main types of informed traders: Value traders, arbitrageurs, and news traders. (Harris, 2003, p. 226)

News traders base their trades on new information about an asset and the estimated impact on the asset price. They most often limit their position to a short period of time. The information can either be firm-specific, industry-related, or based on macroeconomic news (Schenk-Hoppe, 2023). In scenarios where trading is influenced by widely known information, such as tweets, quick action is essential to secure a profit. Traders who can execute trades prior to the impact of their news will be the ones to realize profits (Harris, 2003, p. 228).

2.2 Automated Algorithmic Trading

Algorithmic trading is a trading system that makes quick decisions and transactions in the financial market based on a predefined set of rules (Fuglaas, 2017). There are different types of algorithmic trading; automated, semi-automated, and manual. Viervoll expounded on these diverse methodologies by categorizing them based on the level of risk and the extent of human participation, as depicted in Figure 3 (Viervoll, 2023).

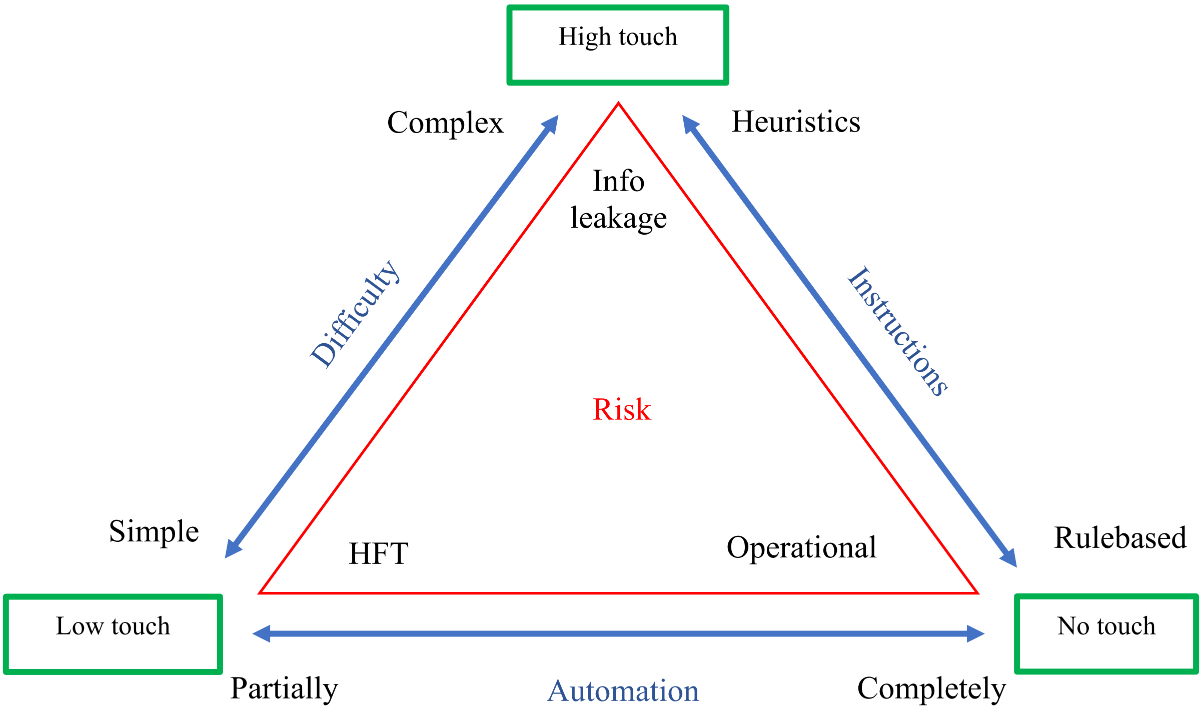

Figure 3 Trading strategies (NBIM)

In the top corner, “High touch” represents manual algorithmic trading, which requires a high level of human involvement due to complexity and heuristics. The foremost risks are information leakage and slow reaction time. The left corner, “Low touch” represents semiautomatic trading. The highest risk is losing to high-frequency traders (HFT) by acting too slowly, thus missing the window of opportunity.

“No touch” represents automatic trading, with complete automation and rule-based instructions (Viervoll, 2023). Buy and sell orders are based on an underlying system, and the orders are executed when the preset trading conditions are met (Mitchell, 2023). This provides a timing advantage. Computers can interpret and decide on binary variables down to 0.3 nanoseconds (Viervoll, 2023). Trading on Elon Musk`s tweets can both be completely automated and rule-based. In addition, “No touch” will eliminate human emotional biases, which might be beneficial while trading based on data and statistics. On the downside, the strategy is dependent on monitoring any changing market conditions and resilience against technical malfunctions (Viervoll, 2023). Autonomous algorithms require operationality, hence identified as the largest risk. Based on requirements of autonomy and speed, we will adopt a "No touch" approach.

2.3 Text Classification Model

A text classification model categorizes text into specific predefined groups (ScienceDirect, 2012). Essential to news trading is the ability to evaluate the impact of news; a text classification model enables this assessment to be automated and instantaneous. Classification models are developed using variables known as features. In a dataset, each feature typically corresponds to a column. In text classification, features might include the individual words found in a text. A feature vector is a numerical representation of an object's characteristics, where each number represents a specific attribute of the object (Shalev-Scwarts S. & Ben-David, 2014). One of the most widely used models for text classification is the Naïve Bayes classifier (Medium, 2023). It is a probabilistic machine-learning algorithm (Bishop, 2006). The classifier employs Bayes' theorem, which describes the probability of an event, based on prior conditions that might be related to the event. The classifier is called naïve because it assumes all features are independent of each other, given the class (Bishop, 2006). There are three main types of Naïve Bayes classifiers, each defined by the distribution of its feature values: Gaussian, Multinomial, and Bernoulli distributions.

Sentiment analysis is also a commonly employed technique in text classification. It is a computational text classification technique used to determine the emotional tone behind words (Amazon, 2023). A sentiment score is a numerical measure indicating the positivity or negativity of a statement (AlphaSense, 2023). Using sentiment scores for classification reduces human biases, allows for tailoring of parameters within an algorithm, and enables large-scale analysis quickly. Nevertheless, the assessment of sentiments could be imprecise when dealing with text that contains irony, conflicting messages, or nuanced feelings.

To effectively categorize text, employing a Naïve Bayes classifier in conjunction with sentiment scoring can be an efficient strategy. Possessing a reliable classification model while trading on tweets is arguably one of the most vital aspects, as it lays the foundation for profitable trading by guiding when to initiate positions.

2.4 Backtesting

Backtesting is a method used to evaluate the performance of a model or strategy by applying it to historical data. This technique allows an ex-post analysis of how well a model would have performed in the past. Strategies implemented by automated trading systems are dependent on backtesting to prove their worth, as they might be too arcane to evaluate otherwise. (Chen, 2021)

A key challenge in backtesting is the assumptions made about trade execution, such as transaction fees, timing, and the availability of assets. These assumptions can sometimes be unrealistic (Chen, 2021). An important consideration is that backtesting does not account for the impact of the new trades as it is impossible to change the history (Lauritzen, 2023). Choosing a liquid stock with high volumes might mitigate this discrepancy.

Another risk is overfitting. Overfitting occurs when a model is repeatedly backtested and tweaked leading the model to capture more noise from the data instead of identifying the underlying pattern. An overfitted model can "explain everything but predict nothing" (Ubøe, 2021). To counteract overfitting, the common practice is to split the dataset. One set for developing a model and one set for validating performance. This approach helps to ensure that the model is robust and capable of making accurate predictions on new data (Twin, 2021). There is no definitive rule for the proportion between training and testing datasets, other than a guideline that the training set should be larger than the test set (Otnheim, 2023).

Backtesting an algorithm requires a program to simulate the market movements and trades. QuantConnect can be utilized to backtest an algorithm. QuantConnect is an open-source, cloud-based algorithmic trading platform for equities, FX, futures, options, derivatives, and cryptocurrencies (QuantConnect, u.d.). The QuantConnect subscription includes access to level 1 data, which comprises open, high, low, and close prices, along with volume information. However, it does not encompass data such as the order book, but the user retains the flexibility to modify the fee structure.

3 Trading Strategy

Our strategy revolves around trading the Tesla stock in response to tweets from Elon Musk. It builds on the principles of news trading, and the underlying premise is: buy the Tesla stock when a tweet from Elon Musk is likely to positively influence the stock price. Conversely, sell the stock if a tweet appears to have a negative connotation that could lower the stock price.

The strategy has been implemented through a "No Touch" trading algorithm with a Naïve Bayes classification model. This gives us the functionality we need to continuously monitor Elon Musk's Twitter activity, assess whether a tweet will positively or negatively influence the stock price, and immediately execute trades based on this assessment.

For rapid and guaranteed trade execution the algorithm will execute market orders. Each position will be held for only one minute, aiming to capitalize on the short-term price fluctuations that follow a tweet. For risk management, we implemented a stop-loss mechanism. The adoption of a take-profit function was evaluated to progressively safeguard our earnings.

However, considering our strategy holds positions for a very short period, we did not find this necessary to implement.

3.1 Implementation

To implement our strategy, we developed the algorithm using Python (Appendix Figure 1). The algorithm is constructed with two distinct classes: one dedicated to trading logic and the other for handling data.

MuskTweet() is the class responsible for data handling, featuring two functions: *GetSource()* and *Reader()*. These functions are responsible for accessing the data source and handling the data. *Reader()* extracts the content of a tweet, its timestamp, and the classification assigned to the tweet.

The class *MyAlgorithm()* has six functions and handles the trading logic. *Initialize()* sets up the algorithm, specifying the start and end dates, the brokerage model, and the initial cash balance. The *OnData()* function is responsible for executing the trades. This function receives new stock price data for every tick. If a tweet is received simultaneously, it assesses its classification and places an order according to the classification. Each order will utilize the entirety of the available cash balance. Once an order is filled, the *ScheduleLiquidation()* function is triggered to liquidate the position after one minute. Each time the *OnData()* function receives new price data *CheckStopLoss()* reviews the algorithm's current holdings. If the price has dropped by one percent from the entry price, it proceeds to liquidate that position.

3.2 Evaluating the Strategy

We have reviewed the outcomes from our simulation, as presented in Figure 4. Our primary focus has been on understanding how transaction costs might affect a live model and on proposing a useful measure for assessing performance.

Figure 4 Result from backtesting

3.2.1 Transaction Costs

In general, transaction costs include all costs associated with trading. There are three main types of transaction costs: explicit, implicit, and missed opportunity costs (Harris, 2003, p. 421). In QuantConnect, only explicit transaction costs are considered in the calculations.

Explicit transaction costs include commission fees to brokers and exchanges (Harris, 2003, p. 421). Initially, higher explicit transaction costs were notable due to QuantConnect's fee model, which charges per share traded. As the price of Tesla stock increased, the number of shares we could trade with our cash holdings decreased. Explicit transaction costs are unlikely to pose a problem in the future unless there is a substantial drop in the stock price, allowing the purchase of more shares with the same capital.

Implicit transaction costs, which include expenses like the bid-ask spread and price impacts from large trades, are not considered in QuantConnect. Liquidity is defined as the ability to trade large sizes quickly and at low cost (Harris, 2003, p. 394). Given Tesla's high trading volume and low spread (Nasdaq, 2023), these costs are anticipated to be minimal for our portfolio, considering its relatively small size and Tesla's high liquidity.

Missed opportunity costs occur when traders fail to fill their orders or miss the window of opportunity. Given that our strategy is "no touch," fully automated, and operates on preset instructions, such costs should not be of concern. However, this assumes the absence of malfunctions.

3.2.2 Performance Measure

The Sortino Ratio is a metric for understanding risk-adjusted returns. It is calculated by subtracting the risk-free rate from the portfolio's return and then dividing this number by the standard deviation of the portfolio's negative asset returns:

$$Sortino\ Ratio: \frac{(R_p - r_f)}{\sigma_d}$$

This ratio is particularly useful for assessing the performance of strategies focused on minimizing losses, as it evaluates a portfolio's performance in relation to its downside risk (Kenton, 2020). Our simulation, covering the period from 2010 to 2023, maintained positions in Tesla stock for a total duration of only 45 minutes. When applying the Sortino Ratio to trading strategies that do not involve long positions, it is recommended to scale the returns to a longer timeframe, such as monthly, before subtracting the risk-free rate (Viervoll, 2023).

Utilizing the Sortino Ratio in a trading strategy, with a classification model prone to errors, can be advantageous. It allows for an evaluation of the model's ability to manage downside risks or errors, thus providing insights into its risk-adjusted return.

4 Developing Our Classification Model

To continuously assess whether tweets will be influential, a classification model is required. In this chapter, we will describe our approach in creating such a classification model. We begin by presenting our data and the preprocessing steps. This is followed by an explanation of the statistical methods we used to identify tweets with significant influence. The tweets we identified were the ones used in the trading simulation shown in the introduction of Figure 2. The insights from the statistics were further used to train and assess our classification model. Our methodology is illustrated in Figure 5.

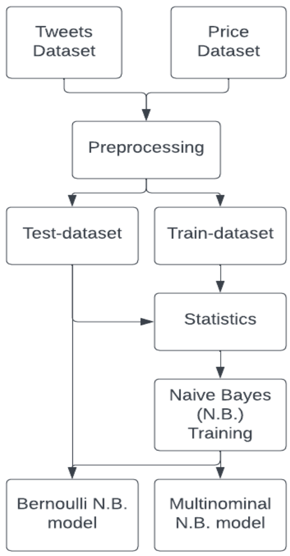

Figure 5 Schematic process of developing the model.

4.1 Data

Our data consisted of two datasets, one with tweets from Elon Musk and one with Tesla stock prices. By using the API of Twitter, we gathered 27.568 tweets in total. The earliest tweet is dated to 2010 and the latest tweet is in June 2023. However, most of the tweets are from 2019 or later. Every tweet is timestamped down to the second. To prepare the data set for analysis we began by isolating the tweets that may have influenced the Tesla stock price, we refined our dataset to include only tweets mentioning Tesla-related terms such as “Tesla”, “TSLA” and “Model 3”. Further, we adjusted the time of the tweets from Coordinated Universal Time (UTC) to Eastern Standard Time (EST) to match the trading hours of Nasdaq. Finally, we eliminated tweets from outside Nasdaq's regular trading hours. This resulted in a total of 950 tweets, with a distribution as shown in Figure 6.

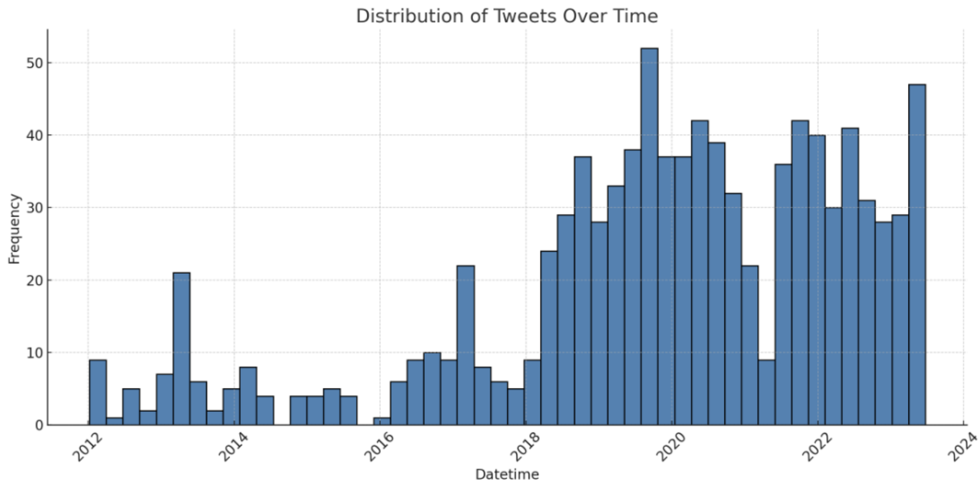

Figure 6 Distribution of tweets over time

The tweets are the basis for building our text-classification model. To reduce the complexity of the text, ensure consistency, and help the model focus on words relevant to the task, we started by cleaning the tweets dataset. First, we lowercased all words, then we identified and removed non-English words and stop words[2], before lastly removing punctuation and special characters. The script performing the text preprocessing can be found in Appendix Figure 2.

The second dataset contained Tesla's stock prices timestamped down to the minute and was acquired from FirstRate Data. We aligned the tweet timestamps to match the stock prices by adjusting them to the minute they were posted. To exemplify: A tweet arriving at 12:15.40 will relate to the price from 12.15. This approach was adopted to ensure that potential price movements were not overlooked. It is important to recognize that staying ahead of the market is a limitation of our analysis.

[2] Stop words examples: 'the', 'is', and 'at' - typically irrelevant for analysis.

For further analysis and to prepare the dataset for backtesting, we divided our data into a training set and a test set. Initially, we shuffled the data to ensure that tweets from all years were evenly distributed across both sets. Subsequently, we divided the dataset using a 70-30 split.

4.2 Statistics

In the training set, we first extracted and analyzed the overall momentum in Tesla's stock price data around the time of Musk's tweets. Second, we visualized the data to determine a representative time horizon during which the stock price is impacted by an influential tweet. Lastly, we used the standard deviation to calculate a confidence interval, enabling us to filter out tweets with significant impact.

In the test set, we implemented a process akin to what we used for the training set. We utilized the standard deviation obtained from the training set to construct a confidence interval. However, the test set was not used in determining a representative time horizon. This specific approach was adopted to prevent potential bias in ascertaining the timeframe of the stock price's reaction.

4.2.1 Momentum

Regarding the test set, the initial step involved calculating the market momentum at the time of the tweets. Only observing the return does not reveal whether fluctuations are due to the tweet or overall momentum in the market. Thus, it was necessary to extract the overall momentum from our return data. To achieve this, we made use of the concept of abnormal return, which compares the actual return against the expected return (Barone, 2021):

$$Abnormalreturn = Actual\,return - Expected\,return$$

When calculating the *expected return*, there is not a predefined limit dictating the duration of the look-back period. The appropriate look-back period varies depending on the strategy being employed (Samuelson, 2023). Strategies focusing on short-term trends require experimentation to determine the most effective look-back period (Samuelson, 2023). Thus, we examined the time horizons 5, 10, 15, 30, and 60 minutes prior to the tweet. The look-back periods from 5 to 15 minutes missed capturing market momentum. Both 30-minute and 60-minute intervals demonstrated comparable results in capturing momentum, leading us to choose the 30-minute interval for its effectiveness in detecting short-term momentum. After calculating the abnormal return of the stock from one to ten minutes following a tweet, we visualized the results in Figure 7:

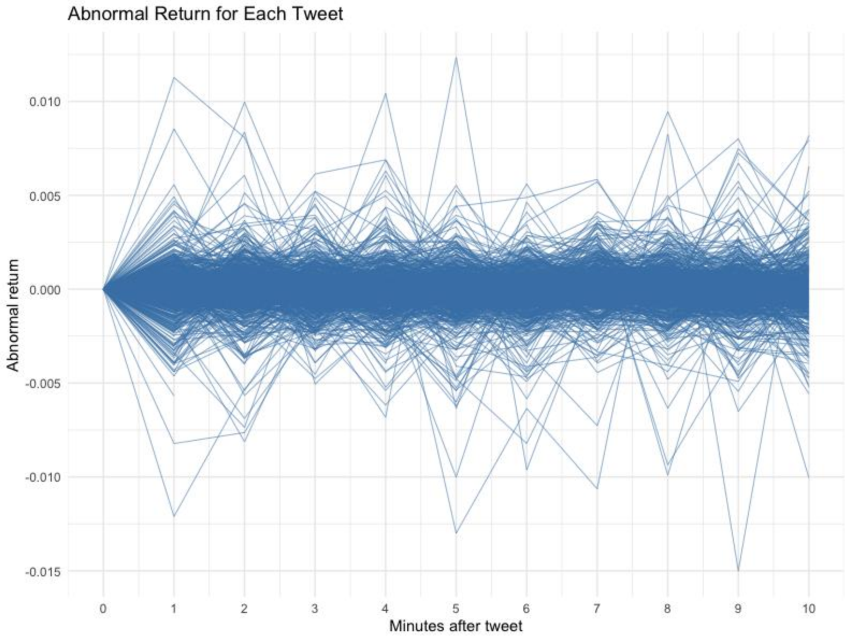

Figure 7 Abnormal return for each tweet

4.2.2 Time Horizon

We analyzed Figure 7 to identify the time horizon in which the stock price reacts to an influential tweet. Next, we computed the standard deviation for each minute following the tweet and discovered that it was at its biggest at the one- and two-minute mark. We suspect this is due to a quick market correction resulting in a diminishing impact after the first minutes.

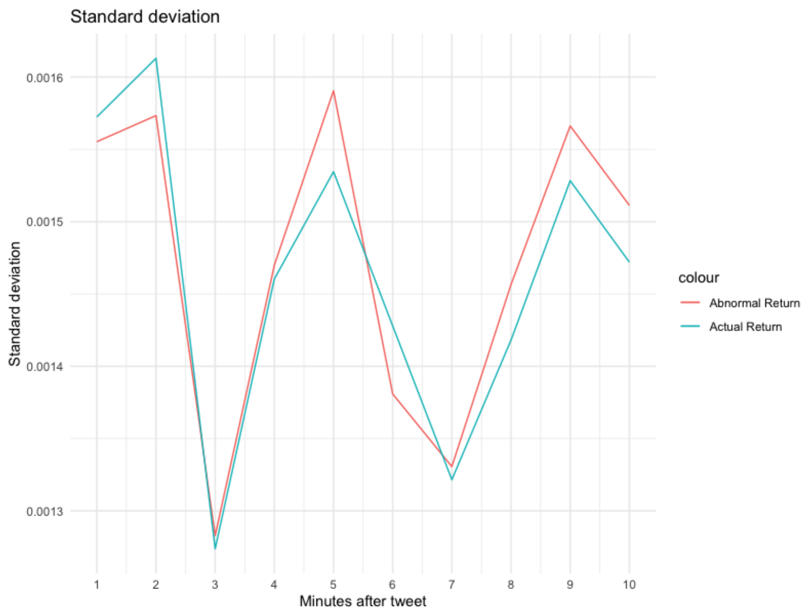

Figure 8 Analysis of return after x minutes

As illustrated in Figure 7, most of the outliers at the one-minute mark gravitate towards zero abnormal return after two minutes have passed. According to Viervoll, a market correction is most likely to occur within seconds, or faster (Viervoll, 2023). Considering this information and examining Figure 8, we chose to focus on the shortest time horizon available in our price data: one minute.

4.2.3 Classification

At the one-minute mark in Figure 7, we observed most tweets do not have an impact on abnormal stock price returns. We wanted to classify the tweets with a significant difference from zero. To classify tweets with positive impact (1), no impact (0), or negative impact (-1), we used the standard deviation from one minute after a tweet and calculated a 95% confidence interval.

Given that each tweet's abnormal return is independent and considering the substantial number of tweets in our dataset, we infer that the distribution of the abnormal returns approximates a normal distribution. This is based on the central limit theorem, which posits that with a sufficiently large sample size, the distribution of sample means will tend toward normality, regardless of the original distribution's shape.

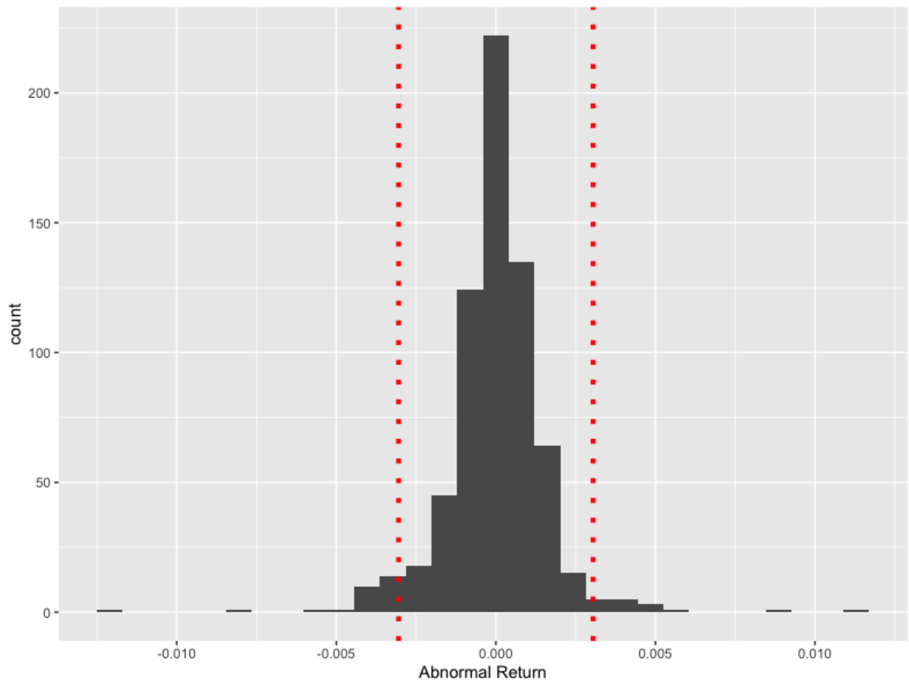

Figure 9 Distribution of abnormal return for tweets

For large samples like ours, the z-score corresponding with a 95% confidence interval is 1,96. The formula for calculating the confidence interval is $\bar{X} \pm 1{,}96 * \hat{s}$, where $\bar{X}$ represents the mean in abnormal return. This resulted in the confidence interval, denoted by the red marks in the distribution presented in Figure 9. From this, we classified the tweets in the training set. (Ganti, 2023)

Regarding the test set, we also calculated the abnormal return. Following this, we used the standard deviation from the training set to calculate the confidence interval and classified the tweets accordingly.

In total, from both the training and the test set, we identified 45 tweets influential tweets, see Appendix Figure 3. The strict significance level was applied to ensure that we do not mislead our classification model. These 45 tweets were the ones used in the trading simulation shown in the introduction.

4.3 Classification Model

To be able to trade Tesla based on future tweets, we needed a text classification model. The model should predict if a tweet will have an impact on the Tesla stock price or not, thereby enabling us to trade based on the predictions. We opted for the Naive Bayes classifier, treating each word in a tweet as an element of a feature vector. For example, suppose that we wish to reconstruct the processed version of the following tweet into a feature vector:

$$Teslas\ financials\ for\ the\ future\ looking\ good$$

Suppose that there are n words in the learning algorithm's vocabulary. An example of a text classification feature vector for this model with each entry $x \in \{0,1\}^n$ is as follows:

$$x = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ exceptional \\ \vdots \\ financials \\ \vdots \\ future \\ \vdots \\ 0 \\ \vdots \\ looking \\ \vdots \\ 0 \end{bmatrix}$$

Figure 10: Feature vector (stop-words excluded)

The appearance of a word is modelled by either the Bernoulli or multinomial distribution. In the Bernoulli Naïve Bayes model, each feature indicates whether words are present or absent. In the Multinomial Naïve Bayes, each feature corresponds to the frequency of a word. The formula for the Naïve Bayes classifier is expressed as follows:

$$argmax_{y_i} P(Y=y_j) \prod_{i=1}^{m} P(x_i \mid Y=y_j)$$

y_i denotes the classification of whether the Tesla stock price is increasing, neutral or decreasing over a predetermined time interval. The variable x_i is the feature vector for tweet i where a total of m tweets are collected. For each observation in the training set, the above product of probabilities is calculated assuming each market trend, and the results are compared. The classification resulting in the higher probability is assumed true and subsequently assigned to that tweet.

As an additional feature, we added a sentiment score for each tweet. This is done by using TextBlob, a popular Python library for processing textual data. Our implementation of the Naive Bayes classifier and sentiment score was carried out in Python using the Scikit library (Pedregosa, 2011). See Appendix Figure 4.

4.4 Training and Testing the Model

After training the model on 70% of the data, we tested the accuracy of the model on the 30% test set. The accuracy of the model is described in the confusion matrices below. The rows marked "Actual" indicate the classifications derived from the statistical analysis, while the columns labeled "Predicted" represent the outcomes generated by our model.

Predicted

Actual	-1	0	1
-1	0	6	0
0	1	271	0
1	0	7	0

Figure 11 Confusion matrix – Multinomial Naive Bayes

Predicted

Actual	-1	0	1
-1	0	6	0
0	0	272	0
1	0	7	0

Figure 12 Confusion matrix – Bernoulli Naive Bayes

In our test set, there were 6 tweets associated with negative impact, 272 with no impact, and 7 tweets with positive impact on the Tesla stock price. As we can observe from the confusion matrices, the models are not able to identify the tweets with either a negative or positive impact. However, instead, our model categorized all tweets as zero, except for one, which was incorrectly labeled as –1. This indicates a significant challenge in distinguishing impactful tweets from neutral ones.

5 Weaknesses

There are multiple reasons for our model's inability to classify tweets correctly. The most critical issue is the combination of data scarcity, low-level features, and class imbalance. The quantity of data required to train text classification models varies and cannot be determined by a universal standard. However, it is reasonable to believe that our training set was not sufficient. Moreover, the scarcity of data amplifies the importance of the features used to train the model. Words are low-level features. It can be ambiguous in how they aggregate to form a classification, thus necessitating a substantial amount of data. A general guideline suggests having ten times more data points than features (Smolic, 2022). In our case, we had 665 data points and 1731 features. Finally, as observed, the model only predicted zeros. This is likely due to class imbalance. When there's a substantial overweight of one class compared to others, a model tends to be biased towards predicting that majority class, as was the situation in our case.

5.1 Solutions

To address the shortage of data points, data augmentation is an effective approach. Data augmentation involves artificially expanding the training set by generating modified versions of existing datapoints (Hastie, 2017). This can be achieved by randomly replacing words in tweets with their synonyms. Another method is back translation, where tweets are translated to another language and then back to English, often leading to minor changes in the phrasing.

To tackle the issue of limited and basic features, one can filter out the less relevant features and include additional, more significant ones. For example, discarding words that appear frequently in all tweets. Additionally, adding new features like the length of the tweet, the inclusion of hashtags, mentions, reposts, URLs, or even emojis.

The model currently employs a standardized sentiment analyzer for the tweets. This method is predominantly focused on text analysis, overlooking vital components such as images, emojis, URLs, and reposts. For reposts our visibility is restricted to the tags, and not the actual content of reposted tweets. This presents a significant risk of inaccuracies in sentiment classification. To address these challenges and improve the accuracy of our sentiment analysis, it would be beneficial to develop and implement a more comprehensive model. This model would need to analyze the entire tweet, encompassing both textual and non-textual elements, to ensure a more precise and reliable sentiment assessment. In fact, Norges Bank Investment Management is actively working on a sentiment model that leverages ChatGPT to predict the impact of news on the stock market (Viervoll, 2023). Applying a generative pre-trained transformer, like ChatGPT, might lead to better outcomes.

Lastly, to reduce the class imbalance, resampling is an effective solution. This involves either oversampling the minority class (-1, 1) or undersampling the majority class (0). However, modifying the original class distribution can be problematic if the distribution does not become representative of real-world scenarios. Oversampling can lead to overfitting, where the model learns the minority class too well and fails to generalize. Undersampling might lead to the loss of potentially important information contained in the majority class. Therefore, it is crucial to balance the need for a representative sample against the risks of introducing bias or losing information.

An alternative approach to address the imbalance problem involves classifying tweets solely as either +1 or -1. To accomplish this, we eliminated the 95% confidence interval, opting to classify tweets that resulted in positive abnormal returns as 1, and those leading to negative abnormal returns as -1. As shown in Appendix Figure 7, this approach was implemented and backtested. The algorithm was now configured to execute trades based on every tweet. The strategy resulted in significant losses due to transaction costs. This derives from the fact that most tweets do not have an impact on stock prices and should not be traded upon. In addition, the classification model classified a substantial number of false positives and false negatives. Overall, the model accurately predicts outcomes about 50% of the time, making it largely ineffective for reliable predictions (see Appendix, Figure 8). The approach resulted in a total of 786 trades, incurring considerable transaction costs in comparison to a non-binary classification approach. In total, we observed a negative total return of 26.51%. Consequently, the initially adopted model, which classifies tweets as -1, 0, or +1, is the preferred choice.

Overall, we have observed that a non-binary classification is the best approach. Going forward, data augmentation, adding additional features, utilizing a more advanced model for sentiment analysis and resampling will be better options to enhance our model.

6 Optimization

To optimize our trading strategy, we propose introducing refined sentiment scoring with decimal precision, such as -0.78, and incorporating a broader range of financial instruments beyond just the Tesla stock. In the following, we will delve into how these adjustments could optimize our algorithm.

An accurate sentiment score could significantly improve our adaptability in terms of the size and duration of our market positions. Currently, our strategy is to invest all our capital in each trade and hold it for precisely one minute. We hypothesize that tweets with strong sentiment values, either close to 1 or -1, have a greater and more lasting effect on the market than those with sentiment values near zero. By precisely gauging tweet sentiment, we could tailor our approach: taking larger positions for a longer period when dealing with tweets of high impact, and opting for smaller, shorter-duration trades in response to tweets with minimal sentiment impact. Additionally, exploring whether our data aligns with Tversky and Kahneman's 1979 study, which suggests that individuals emphasize losses more than gains, would be intriguing (Tversky, 1979). If this theory is confirmed by our analysis, it suggests favoring larger short positions over similar-sized long positions when sentiments are equally removed from a neutral point.

Further optimization could include the involvement of other financial instruments. For example, options trading. This method would entail purchasing call options when tweet sentiment indicates a potential rise in stock prices, and conversely, acquiring put options in anticipation of a decline. Options trading is known for its higher volatility, which may lead to a more substantial return on investment compared to traditional stock trading. The algorithm is straightforward to adopt; however, implementing and developing an alternative strategy necessitates a thorough analysis of the price data associated with options.

7 Conclusion

This research paper aimed to identify how difficult it is to build an (applicable) trading algorithm for Tesla stocks based on information from Elon Musk's tweets. Through the application of statistical analysis on historical data, we have demonstrated that it is indeed possible to develop a profitable trading strategy for the Tesla stock. Our research underscores the potential of leveraging social media data in financial strategies and opens new pathways for innovative approaches in the realm of algorithmic trading.

The main challenge was to apply the results through an applicable text classification model to make the trading strategy operational in the market. Due to data scarcity, low-level features, and class imbalance, the current model has not been able to identify the influential tweets. Nevertheless, we believe there are several possible solutions to these challenges. The most obvious one is arguably an increase in data, though data history is limited. Other solutions such as data augmentation and resampling could also enhance the model. Exploring these possibilities could solve the problems of the model and enable its practical application in real-world scenarios.

Even though it has been proved hard to develop the crucial parts of the operational trading strategy, we are still optimistic about the potential of the model. The fact that professional trading institutions use a lot of resources to develop these models emphasizes that there are potential profits to be made. With the currently increasing development of AI and machine learning, we believe that these models will become more common and hence even more relevant within the field of trading in the future.

8 Bibliography

AlphaSense. (2023, 06 05). *What Are Sentiment Scores and How Are They Calculated?* Retrieved from www.alpha-sense.com: https://www.alpha-sense.com/blog/engineering/sentiment-score/

Amazon. (2023). *What is Sentiment Analysis?* Retrieved from www.aws.amazon.com: https://aws.amazon.com/what-is/sentiment-analysis/

Barone, A. (2021, April 25). *Abnormal Return: Definition, Causes, Example*. Retrieved from Investopedia.com: https://www.investopedia.com/terms/a/abnormalreturn.asp

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. *Springer Science + Business media*.

Chen, J. (2021, August 18). *Backtesting: Definition, How it works, and Downsides*. Retrieved from Investopedia.com: https://www.investopedia.com/terms/b/backtesting.aspAlphaSense. (2023, 06 05). *What Are Sentiment Scores and How Are They Calculated?* Retrieved from www.alpha-sense.com: https://www.alpha-sense.com/blog/engineering/sentiment-score/

Amazon. (2023). *What is Sentiment Analysis?* Retrieved from www.aws.amazon.com: https://aws.amazon.com/what-is/sentiment-analysis/

Barone, A. (2021, April 25). *Abnormal Return: Definition, Causes, Example*. Retrieved from Investopedia.com: https://www.investopedia.com/terms/a/abnormalreturn.asp

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. *Springer Science + Business media*.

Chen, J. (2021, August 18). *Backtesting: Definition, How it works, and Downsides*. Retrieved from Investopedia.com: https://www.investopedia.com/terms/b/backtesting.asp

Chen, J. (2022, January 31). *Algorithmic Trading: Definition, How it works, Pros & cons*. Retrieved from investodepedia.com: https://www.investopedia.com/terms/a/algorithmictrading.asp

CNBC. (2018, August 07). *Tesla shares surge 10% after Elon Musk shocks market with tweet about going private*. Retrieved from CNBC.com:

https://www.cnbc.com/2018/08/07/tesla-says-no-final-decision-has-been-made-to-take-company-private.html

Fernando, J. (2023, 05 11). *Sharpe Ratio: Definition, Formula, and Examples*. Retrieved from www.investopedia.com: https://www.investopedia.com/terms/s/sharperatio.asp

Fuglaas, H. (2017, 05 05). *Algoritmehandel - en innføring*. Retrieved from www.k2trading.no: https://k2trading.no/algoritmehandel-en-innforing/

Ganti, A. (2023, June 21). *Central Limit Theorem (CLT): Definition and Key Characteristics*. Retrieved from Investopedia.com: https://www.investopedia.com/terms/c/central_limit_theorem.asp

Harris, L. (2003). *Trading & Exchanges.* New York: Oxford University Press, Inc .

Hastie, T. &. (2017). The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Second edition. *New York Springer*.

Kenton, W. (2020, 08 09). *Sortino Ratio: Definition, Formula, Calculation, and Example*. Retrieved from www.investopedia.com: https://www.investopedia.com/terms/s/sortinoratio.asp

Lauritzen, J. M. (2023, 09 12). Session 11: Dealer, FX & Derivatives. Bergen, Vestland, Norway: NHH.

McKinney, W. &. (2010). Data Structures for Statistical Computing in Python. *New York Springer*, pp. 51-56.

Medium. (2023, 07 18). *Understanding the Naive Bayes Algorithm: A Powerful Tool for Classification*. Retrieved from www.medium.com: https://medium.com/@data-overload/understanding-the-naive-bayes-algorithm-a-powerful-tool-for-classification-240b3d8b0b6d

Mitchell, C. (2023, June 03). *Autotrading: Meaning, Strategies, Pros and Cons*. Retrieved from Investopedia.com: https://www.investopedia.com/terms/a/autotrading.asp

Nasdaq. (2023, November 23). *TSLA Historical Data*. Retrieved from Nasdaq.com: https://www.nasdaq.com/market-activity/stocks/tsla/historical

Otnheim, H. (2023). MET4: Logistical Regression. *MET4: Logistical Regression*. Bergen, Vestland, Norway: NHH. Retrieved 2023

Pedregosa, F. G. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, pp. 2825-2830.

QuantConnect. (n.d.). *QuantConnect - About us*. Retrieved from QuantConnect.com: https://www.quantconnect.com/about

Samuelson. (2023, September 11). *Lookbak Period in Trading (What is it? - Optimal, Best?)*. Retrieved from Therobusttrader.com: https://therobusttrader.com/lookback-period-in-trading-what-is-it-optimal-best/

Schenk-Hoppe, K. (2023, August 28). Informed traders (overview, information and news, market efficiency, price discovery, thoughts on liquidity and information). Bergen, Vestland, Norway: NHH.

ScienceDirect. (2012). *Text Classification*. Retrieved from www.sciencedirect.com: https://www.sciencedirect.com/topics/mathematics/text-classification

SEC. (2018). *Elon Musk settles SEC fraud charges: Tesla charged with and resolves securities law change.* Washington D.C.: US Securities and Exchange Commision (SEC).

Shalev-Scwarts S. & Ben-David, S. (2014). Understanding Machine Learning: From Theory to Algorithms. *Cambridge University Press*.

Smolic, H. (2022, 12 15). *How much data is needed for machine learning?* Retrieved from www.graphite-note.com: https://graphite-note.com/how-much-data-is-needed-for-machine-learning

Statista. (2023, September 01). *False news worldwide - statistics & facts*. Retrieved from statista.com: https://www.statista.com/topics/6341/fake-news-worldwide/#topicOverview

Stuart Colianni, S. R. (2015). *Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis.* Stanford: Standford University.

The Economic Times. (2021, November 09). *7 Elon Musk tweets that sent Tesla shares on a wild ride*. Retrieved from economictimes.indiatimes.com: https://economictimes.indiatimes.com/markets/stocks/news/7-elon-musk-tweets-that-sent-tesla-shares-on-a-wild-ride/articleshow/87601316.cms

The Guardian. (2023, January 29). *Tesla trial: did Musk's tweet affect the firm's stock price? Experts weigh in*. Retrieved from The Guardian: https://www.theguardian.com/technology/2023/jan/28/tesla-trial-elon-musk-what-you-need-to-know-explainer

Tversky, D. K. (1979). Prospect Theory: An Analysis of Decision under Risk. *Econometrica*.

Twin, A. (2021, 10 22). *Understanding Overfitting and How to Prevent It*. Retrieved from www.investopedia.com: https://www.investopedia.com/terms/o/overfitting.asp

Ubøe, J. (2021). Session 14: Regressionmodel. *NHH: MET2*.

Viervoll, P. (2023, 10 30). Algorithmic trading. (Author, Interviewer)

Viervoll, P. (2023, October 30). FIN11: Session 17: Quantitative Trading (Guest Lecture). Bergen, Vestland, Norway: NHH.

9 Appendix

Figure 1: Trading script

```python
from datetime import datetime, timedelta
from AlgorithmImports import *

class MyAlgorithm(QCAlgorithm):
    def Initialize(self):
        self.SetStartDate(2012, 1, 1)
        self.SetEndDate(2023, 8, 1)
        self.SetCash(10000)

        tsla_security = self.AddEquity("TSLA", Resolution.Tick)
        self.tsla = tsla_security.Symbol
        self.musk = self.AddData(MuskTweet, "MUSKTWTS", Resolution.Tick).Symbol

        self.SetBrokerageModel(BrokerageName.QuantConnectBrokerage, AccountType.Cash)
        tsla_security.SetFeeModel(InteractiveBrokersFeeModel())

        self.stopLossPercentage = 0.99 # 1% stop loss
        self.openingPrices = {}  # To track opening prices of positions

    def OnData(self, data):
        if self.musk in data:
            score = data[self.musk].Value
            quantity = self.CalculateOrderQuantity(self.tsla, score)
            if score == 1:
                self.MarketOrder(self.tsla, quantity)
                self.ScheduleLiquidation(self.Time + timedelta(minutes=1))
            elif score == -1:
                self.MarketOrder(self.tsla, quantity)
                self.ScheduleLiquidation(self.Time + timedelta(minutes=1))

        # Call the function to check for stop loss
        self.CheckStopLoss(data)

    def CheckStopLoss(self, data):
        for holding in self.Portfolio.Values:
            if holding.Invested:
                symbol = holding.Symbol
                if symbol in data and symbol in self.openingPrices:
                    stopLossPrice = self.openingPrices[symbol] * self.stopLossPercentage
                    if data[symbol].Price <= stopLossPrice:
                        self.Liquidate(symbol)
                        self.Log(f"Stop loss triggered for {symbol} at {data[symbol].Price}")

    def ScheduleLiquidation(self, liquidation_time):
        self.Schedule.On(self.DateRules.EveryDay(self.tsla), self.TimeRules.At(liquidation_time.time()), self.ExitPositions)

    def ExitPositions(self):
        self.Liquidate()

    def OnOrderEvent(self, orderEvent):
        if orderEvent.Status == OrderStatus.Filled:
            order = self.Transactions.GetOrderById(orderEvent.OrderId)
            self.Log(f"Order executed: ID: {order.Id}, Type: {order.Type}, Symbol: {order.Symbol}, Quantity: {order.Quantity}. Fees: {orderEvent.OrderFee}")
            if orderEvent.Direction == OrderDirection.Buy:
                self.openingPrices[order.Symbol] = orderEvent.FillPrice

class MuskTweet(PythonData):
    def GetSource(self, config, date, isLive):
        source = "https://www.dropbox.com/scl/fi/4t6szo6q50piauhwjns6v/final_results_sec.csv?rlkey=mtx8ugpkss6y5lq8u6db9a0f3&dl=1"
        return SubscriptionDataSource(source, SubscriptionTransportMedium.RemoteFile)

    def Reader(self, config, line, date, isLive):
        if not (line.strip() and line[0].isdigit()):
            return None
        data = line.split(',')
        tweet = MuskTweet()
        try:
            tweet.Symbol = config.Symbol
            tweet.Time = datetime.strptime(data[0], '%Y-%m-%d %H:%M:%S')
            content = data[1].lower()
            tweet.Value = int(data[2])
            tweet["Tweet"] = str(content)
        except ValueError:
            return None
        return tweet
```

Figure 2: Text preprocessing

```python
import pandas as pd
import re
import pytz
from textblob import TextBlob
import nltk
from nltk.corpus import stopwords, words
from nltk.tokenize import word_tokenize
import string

if not nltk.data.find('corpora/stopwords'):
    nltk.download('stopwords')
if not nltk.data.find('corpora/words'):
    nltk.download('words')
if not nltk.data.find('tokenizers/punkt'):
    nltk.download('punkt')


# English words and stopwords
english_words = set(words.words())
stop_words = set(stopwords.words('english'))

# Function to remove punctuation and URLs from a string
def remove_punctuation_and_url(s):
    no_punctuation = s.translate(str.maketrans('', '', string.punctuation))
    return re.sub(r'http\S+', '', no_punctuation)

# Function to tokenize, remove stopwords, non-English words, and convert to lowercase
def tokenize_and_clean(tweet):
    tokens = word_tokenize(tweet)
    cleaned_tokens = [token.lower() for token in tokens if token.lower() in english_words
                      and token.lower() not in stop_words and token.isalpha()]
    return ' '.join(cleaned_tokens)

# Function to get the sentiment of a tweet
def get_sentiment(text):
    blob = TextBlob(text)
    return blob.sentiment.polarity

# Function to preprocess tweets
def preprocess_tweets(df):
    df['Cleaned_Text'] = df['Text'].apply(lambda x: tokenize_and_clean(remove_punctuation_and_url(x)))
    df['Sentiment'] = df['Cleaned_Text'].apply(get_sentiment)
    return df

# Process Data
df = pd.read_csv('data/tweets/tweets.csv', encoding='latin1')
df = df[["Datetime", "Text"]][::-1].reset_index(drop=True)
df['Datetime'] = pd.to_datetime(df['Datetime'], utc=True).dt.tz_convert(pytz.timezone('America/New_York'))
df.set_index('Datetime', inplace=True)
df = df[df["Text"].str.contains("tesla|tsla|elon|musk|elonmusk|model3|model 3|modely|model y|modelx|model x|" +
                                "model s|models3|models|cybertr")]
df = df.between_time('09:30', '16:00')
df.index = df.index.strftime('%Y-%m-%d %H:%M:%S')
df_tsla = preprocess_tweets(df)

# Output data
df_tsla.to_csv('data/tweets/df_tsla.csv')
```

Figure 3: Tweets

```
Datetime,Text,Rank
2013-04-11 14:03:27,new york judge just ruled in favor tesla dismissing the legal attack by auto dealers to prevent direct sales,1
2013-04-25 11:54:47,announcement of new teslamotors strategy tomorrow tesla owners will like this,1
2013-05-02 13:54:50,time to up the ante and fix some mistakes tesla announcement tomorrow,1
2013-05-07 11:42:59,tesla priority is electrification of cars so priority is model s model x then mass market third gen vehicle amp truck,-1
2015-03-30 12:35:15,major new tesla product line  not a car  will be unveiled at our hawthorne design studio on thurs 8pm april 30,1
2016-04-01 13:23:39,thought it would slow way down today but model 3 order count is now at 198k recommend ordering soon as the wait time is growing rapidly,1
2016-08-23 11:23:03,tesla product announcement at noon california time today,1
2018-04-02 15:03:56,wsj tesla policy is to issue a recall before there are injuries this is absolutely the right thing to do yet there were dozens of recalls by other car comp
2018-07-05 14:02:49,teslabull what a coincidence â€¦,-1
2018-08-07 12:48:13,am considering taking tesla private at 420 funding secured,1
2018-08-07 13:15:03,lizclaman tesla 420,-1
2018-08-07 14:07:17,heydave7 absolutely am super appreciative of tesla shareholders will ensure their prosperity in any scenario,-1
2018-10-08 11:42:18,the physics of how tesla achieved best safety of any cars ever tested note when vehicle weight is taken into account order is more like xs then 3 but they
2018-10-23 14:28:13,to simplify production many tesla model s amp x interior configs will no longer be available after nov 1 order now to be sure of the one you want,-1
2018-10-26 14:02:12,mcjamez tesla clearly youâ€  re not reading my twitter,-1
2018-10-26 15:11:30,bosterogriego tesla mayemusk wsj good question,-1
2019-01-23 09:53:56,teslarati 310s stainless is better for high temp outer skin as it can take 1450 kelvin so active cooling with cryogenic fuel only has to mitigate 300 degre
2020-02-26 12:20:17,erdayastronaut janeidyeve jakethehuman28 teslatruth twitter sure,1
2020-03-24 14:22:17,samtalkstesla mikebloomberg true mikebloomberg this is  accurate you believe in journalistic integrity but if something isnâ€  t done this will continue,1
2020-04-02 15:50:03,28delayslater thirdrowtesla exactly moreover all hospitals were given exact specifications of resmed amp philips ventilators before delivery amp all confir
2020-05-01 11:11:26,tesla stock price is too high imo,-1
2020-05-13 14:00:07,rosileseddie thirdrowtesla acsosheriffs gavinnewsom the custom styling some of the team is doing over their masks is really cool,-1
2020-06-10 10:03:46,flcnhvy gerberkawasaki tesla joerogan good point,-1
2020-07-14 14:48:22,wholemarsblog tesla autopilot was literally named after the term used in aviation also what about autobahn,-1
2020-07-21 13:09:13,samtalkstesla sometimes thatâ€  s how i feel too,-1
2020-07-24 10:12:12,realdonaldtrump tesla thank you on behalf of the tesla team we look forward to building giga texas,-1
2020-11-03 09:53:45,vincent13031925 jonerlichman i put in my last money even though i thought we would still fail but it was either that or certain death for tesla extremely c
2020-11-03 09:59:17,zainraz4 vincent13031925 jonerlichman closest we got was about a month the model 3 ramp was extreme stress amp pain for a long time â€   from mid 2017 to m
2020-11-19 10:14:05,justpaulinelol teslaownerssv unfortunately there are too many rules around liquor distribution even in the us however we will be selling the lightning bott
2021-05-18 09:43:19,aiming for extreme precision with next gen model y â€   microns not millimeters,-1
2021-05-20 13:53:25,heydave7 a longtime tesla supporter gave me the doge dollar sticker at giga berlin,-1
2021-06-21 14:59:34,looking at holding tesla ai day in about a month or so will go over progress with tesla ai software amp hardware both training amp inference purpose is rec
2021-07-30 10:41:34,teslarati residentsponge only thing beyond ludicrous is plaid,1
2021-12-03 11:05:24,teslaownerssv wholemarsblog yes,-1
2022-01-28 14:25:54,tesmaniancom excited for future of solar at tesla,1
2022-02-25 13:20:44,driveteslaca weâ€  ve had about 60 or so cars testing internally in canada hoping to go wider this weekend,-1
2022-04-21 15:12:23,apompliano tesla is really a dozen or more startups if you look at all the things created from scratch that arenâ€  t present in other car companies,1
2022-04-25 09:33:50,ppathole tesla spacex neuralink boringcompany i do love humanity,-1
2022-05-06 11:26:18,mayemusk this rumor originated from a parody account but got picked up by real media amp twitter trends sigh,-1
2022-05-18 12:14:08,factstesla assaadrazzouk zshahan3 johnnacrider1 evafoxu insane,1
2022-09-15 11:50:07,gailalfaratx tesla aelluswamy 106921 coming out in a few days with additional polish 10693 comes out shortly after ai day,-1
2022-11-30 15:53:34,teslaownerssv wow thank you for the kind words reedhastings,1
2022-12-22 13:37:21,congratulations tesla team,1
2023-01-12 10:09:37,muskuniversity exactly,1
2023-01-30 11:57:38,wr4nygov looks like tesla is doing ok,1
```

Figure 4: Classification model

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB, BernoulliNB
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import numpy as np
from scipy.sparse import hstack

dataset = pd.read_csv("last_dataset2.csv", delimiter=",")

tweetsAndSentiment = dataset[['Text', 'Sentiment']]
rank = dataset['Rank']
X_text = tweetsAndSentiment['Text']
X_sentiment = tweetsAndSentiment['Sentiment']
y = rank

# Split data
X_text_train, X_text_test, X_sentiment_train, X_sentiment_test, y_train, y_test = train_test_split(X_text, X_sentiment, y, test_size=0.30, random_state=42)

# Vectorize features
vectorizer = CountVectorizer()
X_text_train_counts = vectorizer.fit_transform(X_text_train)
X_text_test_counts = vectorizer.transform(X_text_test)

X_sentiment_train = (np.array(X_sentiment_train) + 1).reshape(-1, 1)  # scaling to [0, 2]
X_sentiment_test = (np.array(X_sentiment_test) + 1).reshape(-1, 1)    # scaling to [0, 2]
X_train_combined = hstack([X_text_train_counts, X_sentiment_train])
X_test_combined = hstack([X_text_test_counts, X_sentiment_test])

# Train Multinomial N.B.        You, 5 seconds ago • Uncommitted changes
multinomial_clf = MultinomialNB()
multinomial_clf.fit(X_train_combined, y_train)

# Train Bernoulli N.B
bernoulli_clf = BernoulliNB()
bernoulli_clf.fit(X_train_combined, y_train)

# Evaluate
multinomial_pred = multinomial_clf.predict(X_test_combined)
multinomial_accuracy = accuracy_score(y_test, multinomial_pred)
bernoulli_pred = bernoulli_clf.predict(X_test_combined)
bernoulli_accuracy = accuracy_score(y_test, bernoulli_pred)

# Print accuracy scores
print(f"Multinomial Naive Bayes Accuracy: {multinomial_accuracy}")
print(f"Bernoulli Naive Bayes Accuracy: {bernoulli_accuracy}")

# Plot confusion matrix
def plot_confusion_matrix_as_table(cm, title):
    fig, ax = plt.subplots(figsize=(8, 4))
    ax.axis('tight')
    ax.axis('off')
    col_labels = sorted(set(y_test))
    the_table = ax.table(cellText=cm, colLabels=col_labels, rowLabels=col_labels,
                         cellLoc='center', loc='center')
    ax.text(0.5, 0.67, 'Predicted', ha='center', va='center', transform=ax.transAxes, fontsize=10, fontweight='bold')
    ax.text(-0.05, 0.5, 'Actual', ha='center', va='center', rotation='vertical', transform=ax.transAxes, fontsize=10, fontweight='bold')
    plt.title(title)
    plt.show()
multinomial_conf_mat = confusion_matrix(y_test, multinomial_pred)
bernoulli_conf_mat = confusion_matrix(y_test, bernoulli_pred)
plot_confusion_matrix_as_table(multinomial_conf_mat, "Confusion Matrix - Multinomial Naive Bayes")
plot_confusion_matrix_as_table(bernoulli_conf_mat, "Confusion Matrix - Bernoulli Naive Bayes")
```

Figure 5: Example of tradable tweets from backtest: Date, time, tweet, and classification

```
2013-04-11 14:03:27 Tweet: new york judge just ruled in favor tesla dismissing the legal attack by auto dealers to prevent direct sales, Score: 1.0
2013-04-25 11:54:47 Tweet: announcement of new teslamotors strategy tomorrow tesla owners will like this, Score: 1.0
2013-05-02 13:54:50 Tweet: time to up the ante and fix some mistakes tesla announcement tomorrow, Score: 1.0
2013-05-07 11:42:59 Tweet: tesla priority is electrification of cars so priority is model s model x then mass market third gen vehicle amp truck, Score: -1.0
2015-03-30 12:35:15 Tweet: major new tesla product line  not a car  will be unveiled at our hawthorne design studio on thurs 8pm april 30, Score: 1.0
2016-04-01 13:23:39 Tweet: thought it would slow way down today but model 3 order count is now at 198k recommend ordering soon as the wait time is growing
                           rapidly, Score: 1.0
2016-08-23 11:23:03 Tweet: tesla product announcement at noon california time today, Score: 1.0
2018-04-02 15:03:56 Tweet: wsj tesla policy is to issue a recall before there are injuries this is absolutely the right thing to do yet there were dozens
                           of recalls by other car companies last month incl w injuries amp deaths but you only wrote an article about tesla why so, Score: 1.0
2018-07-05 14:02:49 Tweet: teslabull what a coincidence â€¦, Score: -1.0
2018-08-07 12:48:13 Tweet: am considering taking tesla private at 420 funding secured, Score: 1.0
2018-08-07 13:15:03 Tweet: lizclaman tesla 420, Score: -1.0
2018-08-07 14:07:17 Tweet: heydave7 absolutely am super appreciative of tesla shareholders will ensure their prosperity in any scenario, Score: -1.0
2018-10-08 11:42:18 Tweet: the physics of how tesla achieved best safety of any cars ever tested note when vehicle weight is taken into account
                           order is more like xs then 3 but they are all very close, Score: 1.0
2018-10-23 14:28:13 Tweet: to simplify production many tesla model s amp x interior configs will no longer be available after nov 1 order
                           now to be sure of the one you want, Score: -1.0
2018-10-26 14:02:12 Tweet: mcjamez tesla clearly youâ€  re not reading my twitter, Score: -1.0
2018-10-26 15:11:30 Tweet: bosterogriego tesla mayemusk wsj good question, Score: -1.0
```

Figure 6: Example of executed orders in backtest: Date, time, type, price, and quantity

Date Time	Symbol	Type	Price	Quantity	Status	Tag
+ 2013-04-11 14:03:27	TSLA	Buy Market	Fill: $2.92 USD	34071	Filled	
+ 2013-04-11 14:04:27	TSLA	Sell Market	Fill: $2.95 USD	-34071	Filled	Liquidated
+ 2013-04-25 11:54:47	TSLA	Buy Market	Fill: $3.48 USD	28767	Filled	
+ 2013-04-25 11:55:47	TSLA	Sell Market	Fill: $3.48 USD	-28767	Filled	Liquidated
+ 2013-05-02 13:54:50	TSLA	Buy Market	Fill: $3.61 USD	27679	Filled	
+ 2013-05-02 13:55:50	TSLA	Sell Market	Fill: $3.62 USD	-27679	Filled	Liquidated
+ 2013-05-07 11:42:59	TSLA	Sell Market	Fill: $3.85 USD	-25927	Filled	
+ 2013-05-07 11:43:59	TSLA	Buy Market	Fill: $3.84 USD	25927	Filled	Liquidated
+ 2015-03-30 12:35:15	TSLA	Buy Market	Fill: $12.23 USD	8165	Filled	
+ 2015-03-30 12:36:15	TSLA	Sell Market	Fill: $12.36 USD	-8165	Filled	Liquidated
+ 2016-04-01 13:23:39	TSLA	Buy Market	Fill: $15.64 USD	6448	Filled	
+ 2016-04-01 13:24:39	TSLA	Sell Market	Fill: $15.68 USD	-6448	Filled	Liquidated
+ 2016-08-23 11:23:03	TSLA	Buy Market	Fill: $14.88 USD	6791	Filled	
+ 2016-08-23 11:24:03	TSLA	Sell Market	Fill: $14.99 USD	-6791	Filled	Liquidated
+ 2018-04-02 15:03:56	TSLA	Buy Market	Fill: $16.88 USD	6026	Filled	

Figure 7: Backtest with binary classification

Figure 8: Confusion matrix binary classification

Predicted

Actual	-1	1
-1	58	73
1	61	93