HIGH SCHOOL EDITION
Journal of Student Research

The Impact of X (Formerly Twitter) Sentiment on Stock Returns Using Machine Learning Models

Austin Zhang[1] and Paris Zhang[#]

[1]Grandview High School, USA
[#]Advisor

ABSTRACT

The financial world is influenced by numerous factors such as social media. Posts on platforms like X (formerly Twitter) may reflect investors' sentiment and therefore impact the growth of the stock market. In January 2021, Reddit users heavily impacted the Gamestop stock as well as the overall stock market with only a few posts (New York Times, 2021). There exist many similar instances where online statements from individuals or groups changed the direction of a stock (CNBC, 2023). In this study, we analyzed social media data to determine whether different artificial intelligence models can predict the direction of future stock movements. We used Random Forest, K Nearest Neighbors, and Ridge models to analyze 367,666 tweets from X and predict the stocks' direction of change over the course of 1 day or up to 1 week. The performance of these models was assessed by a common metric, F1 score, ranging from 0 to 1 where 0 indicates poor performance and 1 indicates perfect performance. The evaluated machine learning models predicted the direction with F1 Scores of around 0.8, peaking at 0.9, indicating that tweets and social media posts can be used as a tool to guide financial investment. Further studies can investigate the longevity of the impact of specific tweets, incorporating more tweet-related features such as the counts of retweets, followers, and likes.

Introduction

The financial world is inherently unpredictable, only heightened by social media (Suffolk University, 2023). Billionaires' tweets, Reddit posts, and other extraneous factors can suddenly and dramatically impact the stock market. These factors are associated with the term "behavioral economics", the idea that many investors do not invest with the thought of optimal long-term returns or careful calculations but rather the influence of their emotions which are in turn affected by the media (Mullainathan and Thailer, 2000). Mullainathan and Thailer argue that all investors have limited rationality and mostly rely on heuristics in decision making. Social media greatly contributes to these heuristics. According to X Business, in the first 90 days of 2023 alone, there were 498 million tweets regarding finance and stocks, 65% of which were from people of ages 18-34, demonstrating the high level of activity of social media in the modern financial world.

In particular, X users are exposed to opinions on a multitude of matters through short messages, or tweets. Studies including the investigation conducted by Sahayak, Shete, and Pathan in 2015 have demonstrated that artificial intelligence (AI) models can be used to analyze sentiment in tweets. Tellez et. al. extended this idea to non-English text, including emoticons and foreign language, by using models such as Support Vector Machines and Term Frequency-Inverse Document Frequency. These studies combined with the high volume of stock market discussion on X motivated our analysis of the impact of tweets on the stock market, rather than the impacts of other popular social media platforms, such as Reddit or Threads.

A study conducted in April 2023 by researchers at the University of Florida revealed that AI, namely ChatGPT, had the potential to predict the stock market using data that was not directly tied to the stock. Specifically, the researchers ran newspaper headlines through ChatGPT for relevance and asked the model to determine if the headline was positive or negative about the company mentioned. They found that Large Language Models (LLMs) such as ChatGPT-4 had stronger predictability when using either headlines about smaller companies or headlines with bad news. This is consistent with the findings of Mendoza et. al in 2022 that referencing pessimistic news resulted in greater predictability than referencing positive news. These findings motivated our further analysis into the usage of negative news and the predictability of stock market changes for bigger companies.

Another study conducted in April 2021 by Awan et al observed the potential for artificial intelligence models when using big data which are datasets with massive quantities of data. The researchers discovered that models including logistic regression, Naive Bayes, and Random Forest were able to analyze big data and sentiment with 80-90% accuracy. These findings prompt some of our choices regarding model selection in this study. Overall, current literature suggests that the utilization of artificial intelligence in the stock market is a promising area of exploration. In this study, we will evaluate different machine learning models to predict stock market changes for bigger companies and investigate the correlation between bad news and model predictability.

Dataset

Data Overview

We first located data collected from Kaggle.com containing information regarding specific tweets and the stock price/return following the corresponding tweets. Prior to cleaning, the dataset contained 1,395,450 rows and 14 features. In each row, the dataset showed a specific tweet about a company and the 1, 2, 3, and 7-day returns of that company's stock following the date of the tweet. Other features included the 10 and 30-day volatility of the stock in question as well as the volume traded. Overall, there were tweets about 101 different companies. We categorized these as "big" companies due to their strong international brand recognition, high market capitalization, and over $1 billion in annual revenue. The data contained tweets from September 30th, 2021, to September 30th, 2022. Tweets were often 1-2 sentences long, limited to 280 characters long.

To measure how the stock market changed over time, we utilized the stock return value. Stock market returns are usually defined as a percentage gained or lost compared to the original value, calculated by $\frac{E-S}{S} \times 100\%$ where S is the price at the starting point in time and E is the price at the end point in time. The stock return value can be positive or negative or zero, where a positive return indicates a profit, and a negative return indicates a loss. For example, if a stock with an original value of $100 increases to $120 in 2 days, its 2-day return value will be 0.2, or 20%.

Another metric called stock volatility measures the degree of variation in stock price over a certain period. The volatility of stock can be calculated as the standard deviation or variance of the stock returns. For example, volatility across 10 days can indicate the degree to which a stock fluctuates during those 10 days. This value of volatility can range from 0 to positive infinity where high volatility indicates the price of the stock can change rapidly and dramatically, leading to potentially higher returns but also greater risk. Our data contained the 10 day and 30 day volatility of associated stocks.

Data Cleaning

Before we applied the machine learning models, we cleaned and preprocessed the data to remove errors and inconsistencies. In this study, due to the nature of tweets, many data contained links and social media handles in the form of "@" followed by letters. Links, by themselves, do not contain any sentiment or tone and thus were removed to reduce any extra noise. For example, "@CBSi Jamaicans make money with @Payoneer @PayPal, @paxuminc, @ecoPayz and @okpaycom https://t.co/FWzqUqYRyU," is a tweet with both handles and a link, and during cleaning, the link was removed using regular expression.

To visualize the tweet data, we utilized a tool called wordcloud. In a wordcloud, the size of each word is proportional to its frequency. That is, the words that appear larger occur at higher frequencies within the data, and conversely, smaller words do not occur as frequently. Figure 1a is the initial wordcloud prior to thedata cleaning process. Note the word "https" is shown predominantly in Figure 1a but disappears in subsequent wordclouds – we removed this sequence during cleaning due to its lack of sentiment related to stock return prediction. Figure 1b shows the effect of this removal and appears to still contain obvious noise such as the sequence "FWzqUqYRyU". Figure 1c depicts the impact of removing links and handles, and by contrast, Figure 1d shows the impact of only removing links while keeping the handles. Ultimately, we used the data corresponding to the wordcloud shown in Figure 1d since handles can contain important and meaningful information. Additionally, the word "amp" was removed during cleaning, since it is an error resulting from the generation of "&", the ampersand symbol.

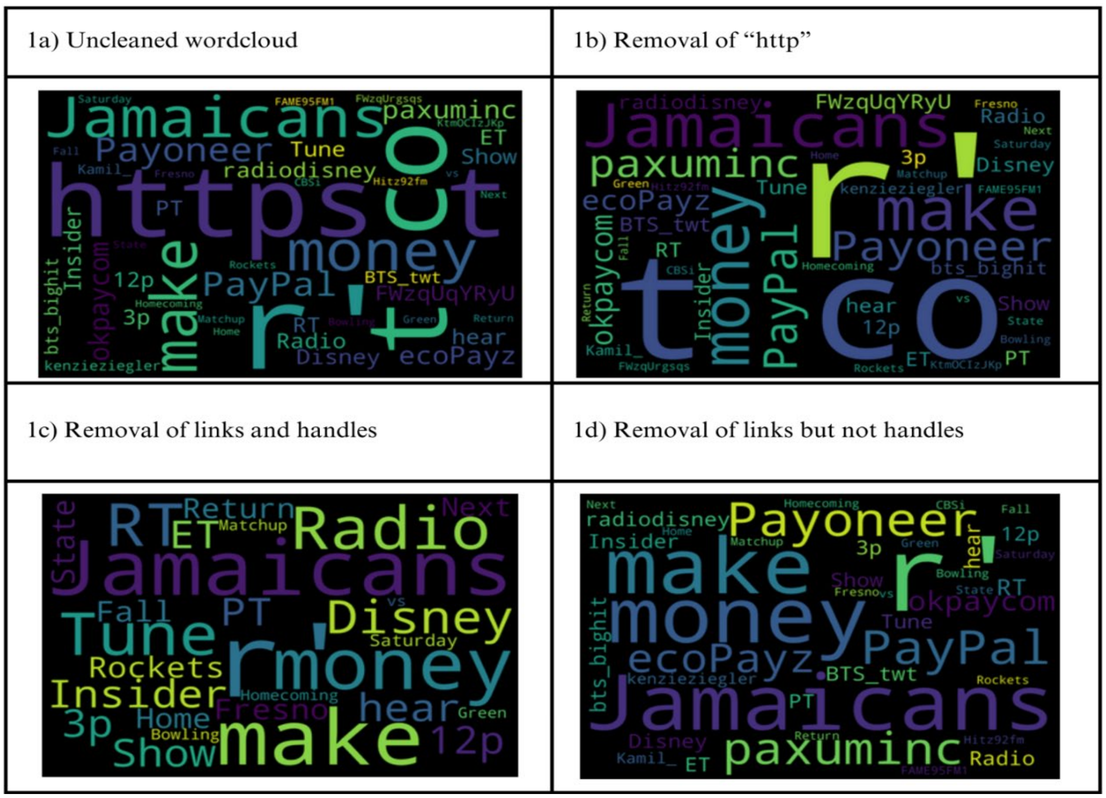


Figure 1. Wordclouds based on the level of data cleaning

Subsequently, the columns corresponding to outcome variables were inspected for abnormal values. We expect the values of the 3-day returns to be relatively similar to those of the 2-day and 7-day returns; however, the range of values for the 3-day returns was much bigger, with some values exceeding 1 million, which is not realistic.

Upon observing rows with outlier 3-day returns values, it became clear that they contained data corresponding to dates rather than tweets.

TWEET	STOCK	DATE
01/10/2018	11.43	0.1469816272965879
01/10/2018	55.76	0.030308464849354462
01/10/2018	32.51	0.01568748077514627
01/10/2018	49.9284	0.007721056552983737
01/10/2018	55.76	0.030308464849354462

Figure 2. Examples of data corresponding to dates

To remove these rows, tweets were converted into a date-time format using Python's built-in function; rows with correct tweets were converted as "NaT", or "Not a Time" while the irrelevant data appeared as an actual date. This filtering allowed the appropriate rows to be removed. Afterwards, the dataset contained 367,666 rows (in comparison to the 1,395,450 starting rows). Figure 3 and Table 1 show the distribution of returns after data cleaning. Table 2 shows the selection criteria for the tweets and number of tweets being selected.

HIGH SCHOOL EDITION
Journal of Student Research
Volume 13 Issue 3 (2024)

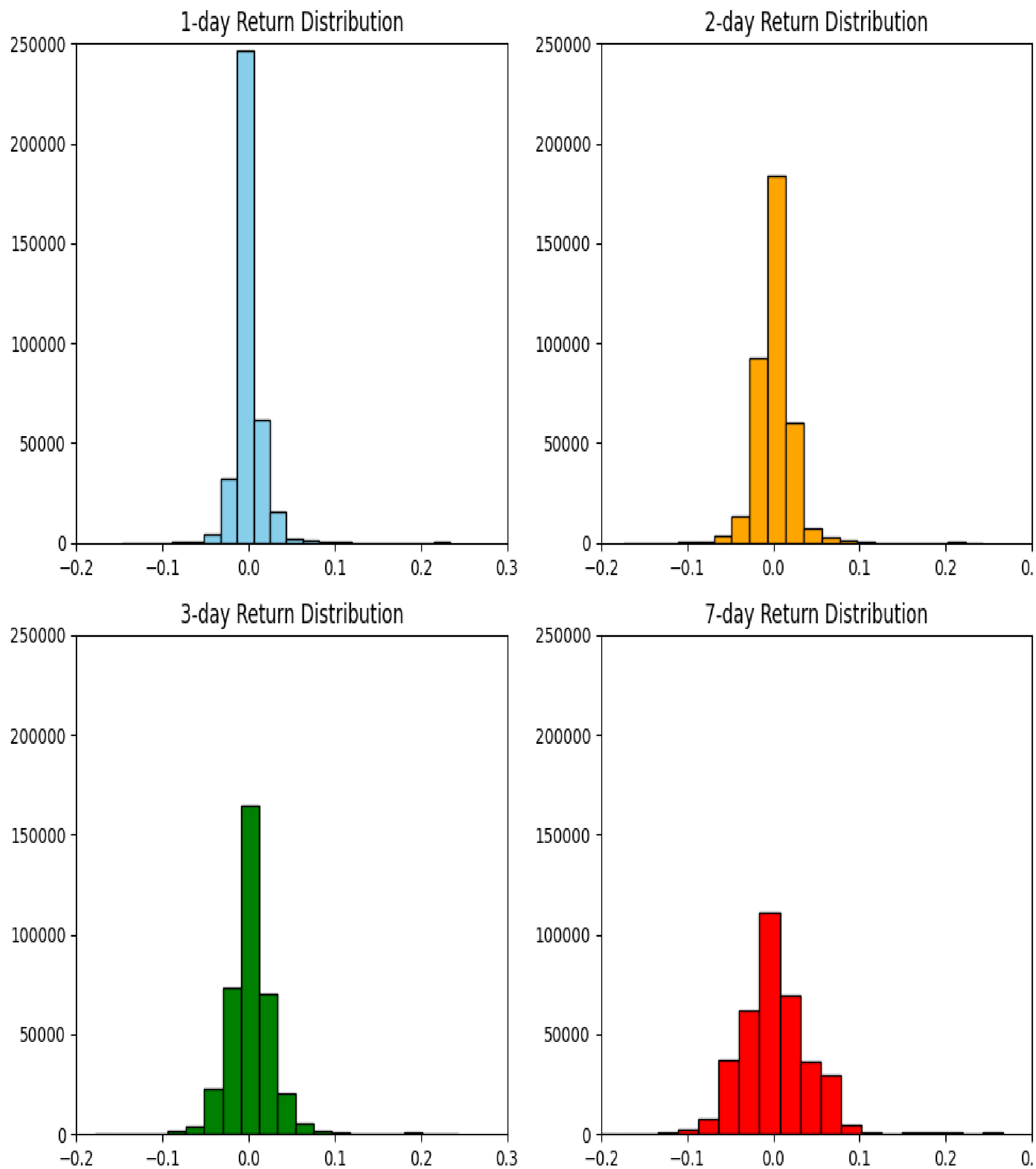


Figure 3. Distribution of each return type.

Table 1. The mean, standard deviation, first quartile (Q1), median, and third quartile (Q3) of each return type.

Category	Mean	Standard Deviation	First Quartile (Q1)	Median	Third Quartile (Q3)
1-day Returns	0.000832	0.018202	-0.005570	0.000000	0.004407
2-day Returns	0.001498	0.022550	-0.009105	0.000000	0.009699
3-day Returns	0.001528	0.026773	-0.011208	-0.000171	0.013464
7-day Returns	0.003422	0.042245	-0.021923	-0.001054	0.026099

Table 2. Selection criteria for data cleaning

Selection Criteria in Data Cleaning	Number of Tweets/Rows Remaining
Removal of rows that contained NaN in tweet and 1, 2, 3, and 7-day return column	568,794
Removal of links in tweet column	568,794
Removal of "amp" in tweet column	568,794
Removal of rows with non-valid data in tweet column	367,666

Tokenization

Tokenization describes the process of splitting text into smaller units called tokens. After cleaning the tweet data, we reduced it to a set of the most important words (i.e., tokens). This was conducted in multiple steps. First, all instances of punctuation from the tweets were identified and removed utilizing Python's built in string.punctuation library since punctuation does not have sentimental value. Next, stop words such as "is", "are", and "the" were removed, since they do not add meaning either. We referenced Natural Language Toolkit (nltk)'s default list for stop words. Finally, using nltk's lemmatizer, we removed "stemming" which means that all tokens with the same stems/prefixes/suffixes were converted into their "base" form, or root (e.g., running → run). We assume that base words and their potential extensions have similar or the same meaning, so the distinction is not useful.

Vectorization

Following tokenization, the tweet data was vectorized using Python package sklearn's CountVectorizer. We considered a total of 1000 unique words, and for each sentence, CountVectorizer returned an integer vector with 1000 entries. Each entry corresponded to the number of occurrences of one of the 1000 words. Finally, after vectorizing the tweet, the 10-day volatility of the stock was also added to the vectors, resulting in vectors with 1,001 features. Figure 4 below depicts the entire process of converting a tweet to a usable vector (including the tokenization and vectorization process).

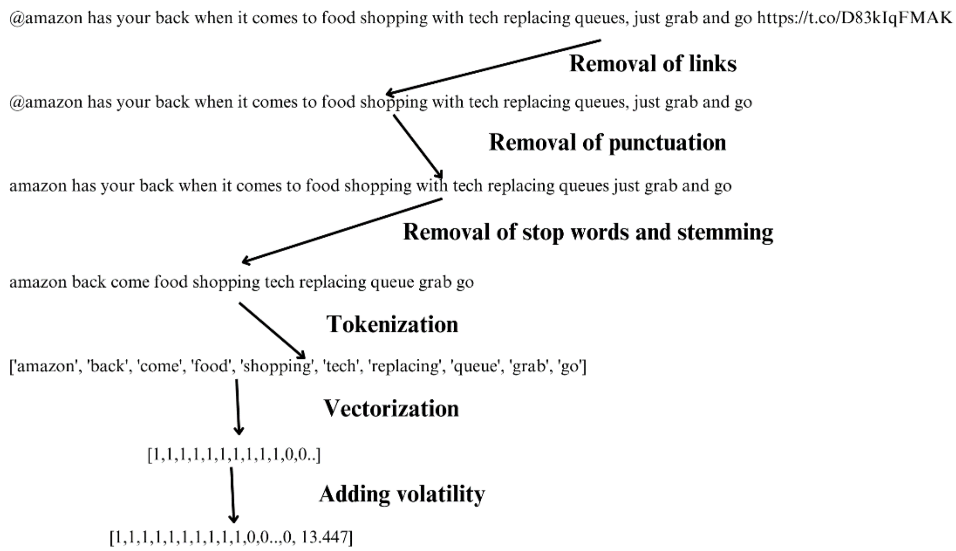


Figure 4. Illustration of data cleaning and processing for an example tweet

Methodology

After converting the tweets into a numerical format, various models were applied to the resulting vectorized data to predict 1, 2, 3, and 7-day stock returns. Binary classification methods, specifically, allowed identification of the overall trend as either negative or non-negative. We aimed to predict the stock direction based on tweet content and market volatility while also identifying underlying trends and patterns.

In particular, we used K-Nearest Neighbors (KNN), Random Forest, and Ridge Regression. Although KNN is a common artificial intelligence model, extensive studies relating to its usage for stock market prediction have not been conducted. We chose Random Forest based on previous studies demonstration of the model's accuracy. Finally, Ridge Regression was chosen due to it being a version of linear regression which is a widely used model. All models were used to predict the direction of each of the 4 types of returns, 1, 2, 3, and 7-day returns.

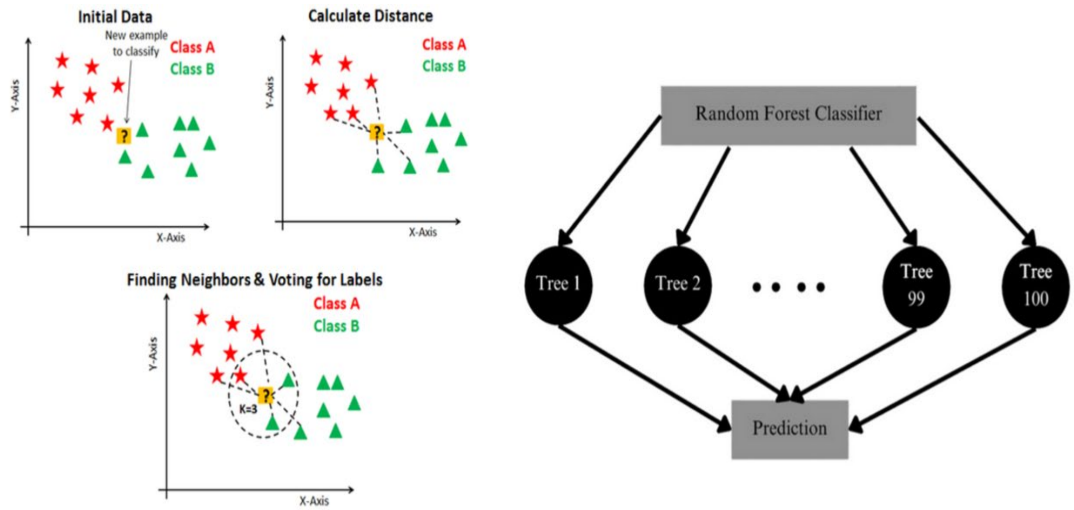
Models

K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a simple, yet powerful machine learning algorithm first developed in 1951. Suppose we are given a dataset with two features and an integer value K. We plot the features on the two-dimensional XY axis (Figure 5a). In this example, the red stars and green triangles represent data points from two different classes in a training dataset. When considering a new data point (yellow question mark), the KNN algorithm computes the distance metrics between the new data point and all other data points in the training dataset. The algorithm then determines the "K" closest data points (i.e., neighbors) and makes a prediction of the point's class based upon the that of of its K nearest neighbors. In Figure 5a, the new data point would be classified as Class B (green triangles) (Deepthi AR, 2019). Furthermore, the algorithm can also predict a numeric value associated with the point, not just its class. For example, if K is set to 3 (that is 3 neighbors), the

KNN algorithm determines the closest 3 neighbors using a distance metric, e.g., Euclidean distance. Then, based on the properties of the 3 nearest neighbors, the algorithm draws a conclusion about the new data point. In this study, we aim to predict a categorical value, that is, the direction of the stock change.

As we can see from our example, the KNN algorithm is easy to understand and interpret. Additionally, it does not require to be linear and has few hyperparameters to tune.



5a) Visualization of KNN

5b) Visualization of Random Forest

Figure 5. Diagrams depicting KNN and Random Forest mechanics.

Random Forest

Random Forest is another commonly used machine learning method. The structure of this model resembles the structure of many decision trees (Figure 5b). A decision tree is a structure that uses step by step decisions and their corresponding consequences, in a tree-like manner. For Random Forest, each decision tree receives a subset of the training data and is trained separately to draw its own conclusions. Figure 6 shows an example of a decision tree that outputs a prediction for the temperature.

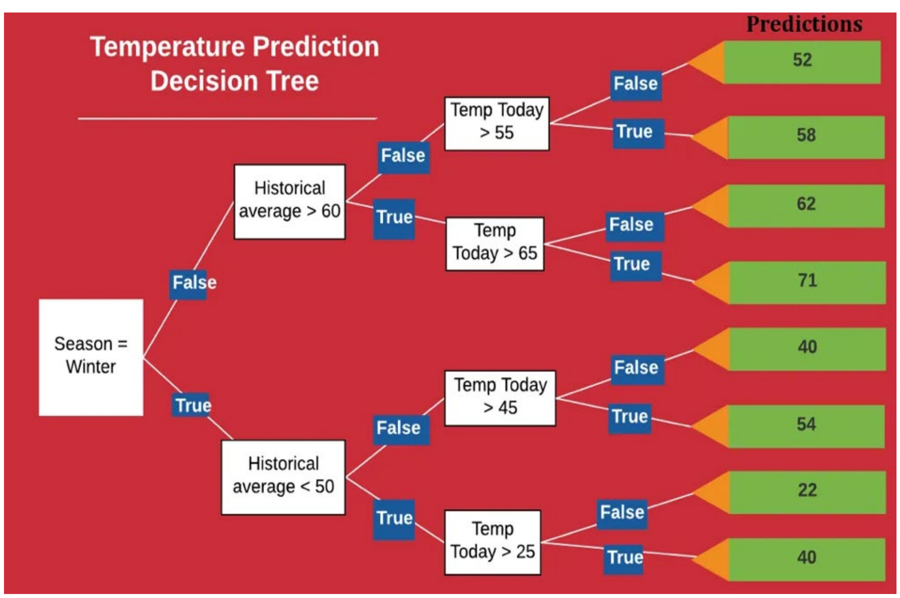


Figure 6. Example of a decision tree that predicts temperature given different related information.

When considering a new data point, each tree makes its own conclusion, and the average of all these individual conclusions is returned as the final prediction for the new data point. This algorithm protects against outliers and noisy data by using the average. This is especially important for the chosen dataset in this study since it contains tweets with an assortment of words, abbreviations, and slang (Koehrsen, 2020).

Ridge Regression

Ridge regression is a variant of Linear Regression that incorporates a method intended to reduce overfitting. Equation 1 shows the loss function for Ridge regression (IBM, 2024). The first term represents the residual sum of squares, which measures the fit of a linear model. The next term, the regularizer, uses the sum of the squares of the models' coefficients to penalize excessively large regression coefficients (indicating overweight of features) and thus reduces the risk of overfitting. When the model has coefficients with large magnitude, it is susceptible to minor changes in the data, and therefore, is not generalizable to all data. λ is a hyperparameter. A larger λ means the loss function penalizes coefficients more.

$$RSS_{L2} = \sum_{i=1}^{n}(Y_i - Y_l)^2 + \lambda \sum_{j=1}^{P} B_j^2$$

Equation 1: Loss function for Ridge regression

We chose Ridge regression because of its ability to prevent overfitting which is crucial since our data has 1001 features. Furthermore, because of its similarity to linear regression, Ridge regression provides a good "baseline" for more complex models.

Classification

To convert the data from numerical stock returns to the direction of stock change, we classified all values into two categories, positive or negative. The label of the negative category also includes the stock returns with no change, or exactly 0 return. Table 3 shows the number of data points and corresponding percentages for each category. Each percentage shown in the table represents the frequency of the category (positive or negative) within the specific time frame (1-day, 2-day, 3-day, or 7-day returns), not the frequency within the entire dataset. For example, the percentage in the 'Positive Return (%)' column for '1-day returns' reflects the proportion of negative returns out of all 1-day returns, rather than out of all returns across all time frames.

Table 3. The number of tweets associated with each category of stock return.

	Positive return	Positive return (%)	Negative return	Negative return (%)
1 Day Returns	133792	36.39%	233874	63.61%
2 Day Returns	162024	44.07%	205642	55.93%
3 Day Returns	177996	48.41%	189670	51.59%

7 Day Returns	177366	48.24%	190300	51.76%

F1 Score

We used F1 score as the metric to measure model performance for binary classification (e.g., positive or negative change in stock return). Ideally, a good predictive model should maximize the prediction of true positives and true negatives and minimize the prediction of false positives and false negatives. Precision measures the accuracy of positive prediction and is defined as the ratio of correctly predicted positive observations to the total number of predicted positives. Mathematically this can be written as $\frac{(TP)}{(TP)+(FP)}$where (TP) is the number of true positives, and (FP) is the number of false positives. Recall, also known as sensitivity, is defined as the ratio of correctly predicted positive observations to all positive observations in the class. Mathematically this can be written as $\frac{(TP)}{(TP)+(FN)}$ where (FN) is the number of false negatives. F1 score is computed as the harmonic mean of precision and recall. In other words, $F1 = 2 \times \frac{P \times R}{P+R}$, where P denotes the precision and R denotes the recall. The minimum F1 score is 0, and the maximum is 1, and a greater F1 score signifies better model performance.

We calculated the F1 score for each category and a weighted average F1 score for all categories, evaluating the model's overall performance. It can be calculated as $F1_{weighted} = \sum_{i=1}^{N} w_i \times F1_i$, where $F1_i$ represents the F1 score for the ith category, and $w_i$is the weight given to that category's F1 score. The weight, in this case, is determined based on size of the category: higher values in the category translates to a higher weight.

Results

F1 scores were calculated for each model (KNN, Random Forest and Ridge Regression) by days of return (1, 2, 3, and 7-days) and category (positive and negative stock return) as shown in Figure 7 and Table 4. Random Forest consistently resulted in the best F1 scores of 0.83, 0.86, 0.89, and 0.93 for the positive category, and 0.91, 0.89, 0.90, and 0.94 for the negative category for 1, 2, 3, and 7-day returns, respectively. KNN followed close behind with F1 scores of 0.78, 0.81, 0.85, and 0.89 for the positive category, and 0.88, 0.85, 0.86, and 0.90 for the negative category for 1, 2, 3, and 7-day returns, respectively. The lowest performing model was Ridge, which consistently had lower F1 scores of 0.34, 0.47, 0.52, and 0.56 for the positive category and 0.65, 0.52, 0.49, and 0.49 for the negative category for 1, 2, 3, and 7-day returns, respectively.

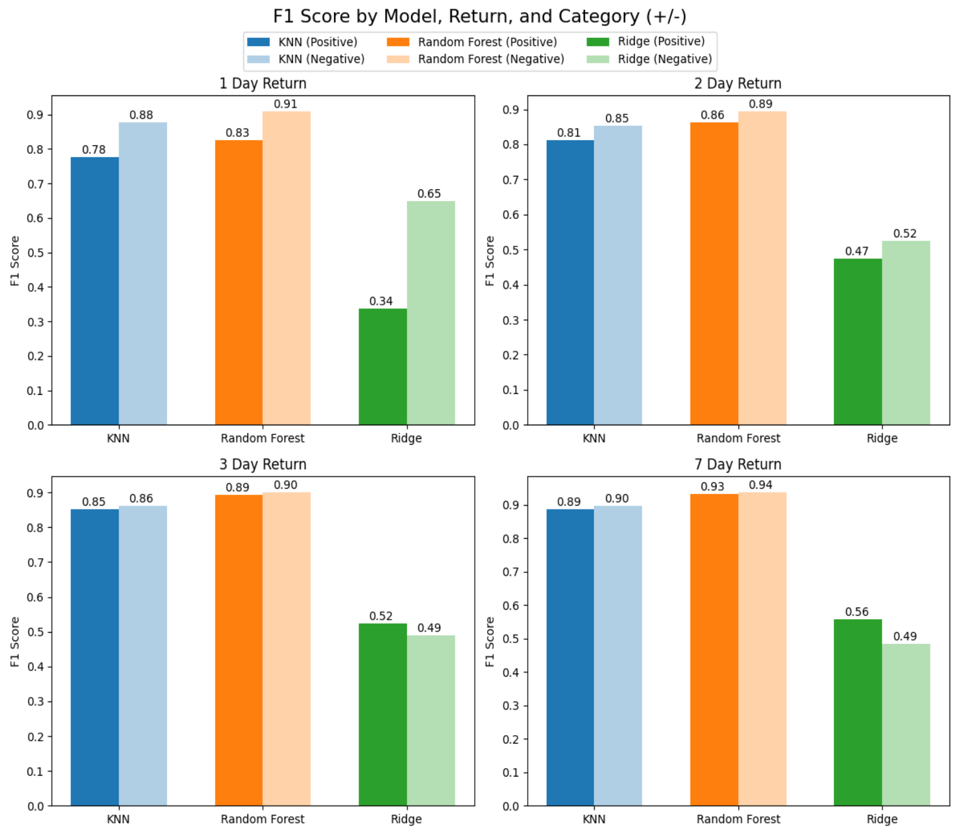


Figure 7. F1 score by model, return day, and category.

Table 4. The F1 scores by model, return day, and category.

	F1 Scores					
	KNN		Random Forest		Ridge	
	Positive	Negative	Positive	Negative	Positive	Negative
1 Day Return	0.7752	0.8760	0.8251	0.9091	0.3358	0.6486
2 Day Return	0.8128	0.8525	0.8619	0.8944	0.4742	0.5249
3 Day Return	0.8524	0.8619	0.8941	0.9010	0.5229	0.4904
7 Day Return	0.8873	0.8968	0.9318	0.9376	0.5577	0.4853

In addition, to evaluate how each model performed as a whole, we used a weighted averaged F1 score across both positive and negative categories. The weight of each category depends on its size. Figure 8 shows the averaged F1 scores for each model by return day. Random Forest results in the highest scores, with averages of 0.878, 0.880, 0.898, and 0.935 for 1, 2, 3, and 7 return days, respectively. Closely behind, KNN demonstrated F1 scores of 0.839, 0.835, 0.857, and 0.892 for 1, 2, 3, and 7 return days, respectively. Then, by far the lowest performing, is Ridge, with F1 scores of 0.535, 0.503, 0.506, and 0.520 for 1, 2, 3, and 7 return days, respectively.

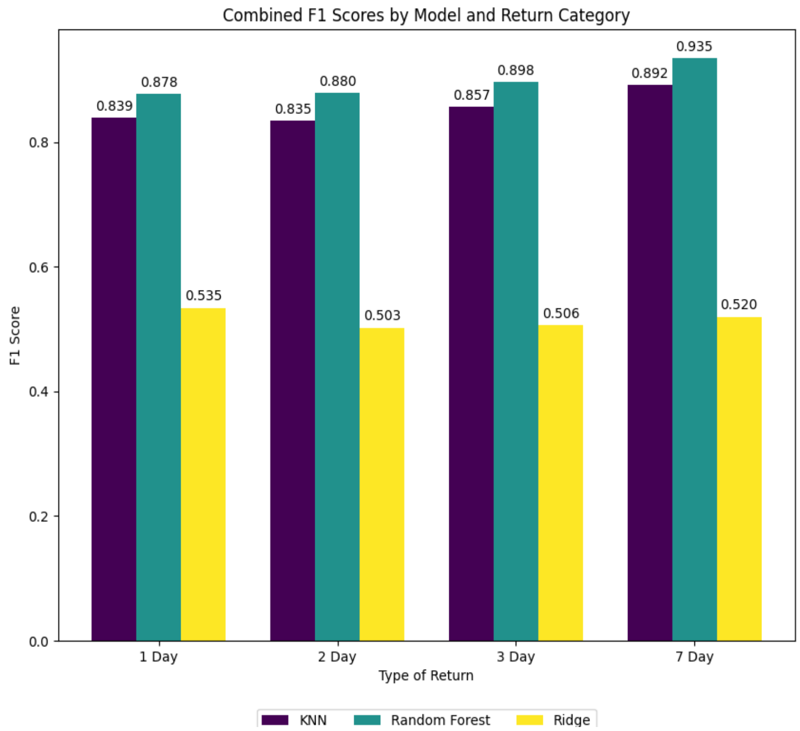


Figure 8. Combined F1 score by model, return date, and category.

Discussion

Overall Performance

Random Forest and KNN performed well in classification of stock movement using tweet messages. This result is somewhat unprecedented as KNN was not a commonly studied model for stock market prediction. F1 scores were greater than 0.8 for both models, across all types of returns, indicating a high level of accuracy and ability to predict the direction of the stock's movements.

Furthermore, we identified 2 trends regarding model performance across the time periods. First, the prediction for the negative category generally outperformed that of the positive category. Past studies have consistently demonstrated this trend in both KNN and Random Forest, our two best performing models, so our results reinforce previous findings that negative sentimental tweets have larger prediction value on the direction of stock returns than positive sentimental ones. A potential explanation for this references human psychology; people may be more influenced by the fear of huge losses following pessimistic news/tweets.

Secondly, the F1 score increased as the number of days increased. One may expect each tweet to have a decreasing impact as time goes on, but our results indicate the opposite. A possible explanation may be that over the observed time period of up to 7 days in this study, as more tweets of the same "content" or retweets are made, the original message's impact was amplified.

Besides F1 score, we also considered another factor for evaluation which assessed computation cost and runtime. Using a local Graphics Processing Unit (GPU), the runtime of Ridge was roughly 40x faster than that of Random Forest at only 1-2 seconds. On the other end, KNN had a runtime of close to an hour. When using Google's Tensor Processing Unit (TPU) v2, Ridge runs in less than a second, Random Forest runs in 5-6 seconds, and KNN takes approximately 2-3 minutes. While runtime is an important consideration, Ridge's quick runtime does not offset its poor performance. Thus, among these three models, Random Forest offers the best F1 score while still using a relatively short runtime.

Conclusion

This study shows that two machine learning models, Random Forest and KNN, can provide an accurate prediction about the direction of movement of a stock using tweet postings and volatility. An F1 score of over 0.8 indicates good model performance, and our models performed with an F1 score of over 0.9 in some conditions. Further research incorporating more data features and optimizing the machine learning methods can be explored to enhance and expand the application of machine learning in predicting stock market changes.

Future Work

Future studies should examine the long-term impact of tweet messages on the direction of stock return and whether the prediction performance of our three models extends to these longer time periods such as 14 or even 30 days. Analyses regarding the time point at which the performance of machine learning models will diminish and if the performance diminishes dramatically or gradually should be conducted. Since the results of our research highlighted only a gradual increase in tweet impact up to 7 days, an understanding of how the tweet's impact changes beyond this time period would be useful.

Additionally, more factors can be considered and explored to optimize the prediction performance. The number of followers of the tweeter's account may contribute to explaining the impact of certain tweets over others. For example, a tweet by a person with 1 million followers will likely have more influence than one from a user with 100 followers. Similarly, other data in the post can also be helpful in analyzing the impact. Data including the number of retweets, comments, and likes to capture the tweet's ability to reach a large audience and impact the market. Furthermore, due to computing limitations, the X data was limited to 1001 features, with 1000 from the text of a tweet, and 1 from the volatility over 10 days. Future models that allow for usage of more will offer better accuracy and predictions, albeit at the cost of runtime and resources used.

Finally, the results of this study can be utilized to design future studies. For example, we used stocks from some large sized companies, so future work can explore the generalizability of the findings of this study to smaller sized companies. In addition, more complex models can be applied to not only label tweets as binary category of positive or negative growth on the stock but also with multi-level complex sentiments (e.g., optimistic, cautious, etc.).

Acknowledgments

I would like to thank my mentor, Yunyue Zhang, for her support and guidance for the completion of this project.

References

[1] CHATGPT can forecast stock price movement with better accuracy than humans: Study. Business Today. (2023, April 27). https://www.businesstoday.in/technology/news/story/chatgpt-can-forecast-stock-price-movement-with-better-accuracy-than-humans-study-379071-2023-04-27

[2] Franchin, W. (2023, July 10). CHATGPT enters trading in the Financial Markets. Medium. https://medium.com/the-investors-handbook/revolutionizing-investment-strategies-chatgpts-ai-takes-on-the-financial-markets-53ef04a9b2c0

[3] Karolina Sowinska, & Pranava Madhyastha. (2021). A Tweet-based Dataset for Company-Level Stock Return Prediction [Data set]. Zenodo. https://doi.org/10.5281/zenodo.4662780

[4] Koehrsen, W. (2020, August 18). Random Forest Simple Explanation. Medium. https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d

[5] Javed Awan, M., Shafry Mohd Rahim, M., Nobanee, H., Munawar, A., Yasin, A., & Mohd Zain Azlanmz, A. (2021). Social Media and Stock Market Prediction: A big data approach. Computers, Materials & Continua, 67(2), 2569–2583. https://doi.org/10.32604/cmc.2021.014253

[6] Jcunningham. (2023, February 22). The Increasing Influence of Social Media on the Stock Market. Journal of High Technology Law. https://sites.suffolk.edu/jhtl/2023/02/22/the-increasing-influence-of-social-media-on-the-stock-market/

[7] Lopez-Lira, A., & Tang, Y. (2023). Can CHATGPT forecast stock price movements? return predictability and large language models. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4412788

[8] Mullainathan, S., & Thaler, R. (2000). Behavioral Economics. https://doi.org/10.3386/w7948

[9] Nuñez-Mora, J. A., & Mendoza-Urdiales, R. A. (2023). Social sentiment and impact in US equity market: An automated approach. Social Network Analysis and Mining, 13(1). https://doi.org/10.1007/s13278-023-01116-6

[10] Phillips, M., & Lorenz, T. (2021, January 27). “dumb money” is on gamestop, and it’s beating wall street at its own game. The New York Times. https://www.nytimes.com/2021/01/27/business/gamestop-wall-street-bets.html

[11] R, D. A. (2019, September 24). KNN visualization in just 13 lines of code. Medium. https://towardsdatascience.com/knn-visualization-in-just-13-lines-of-code-32820d72c6b6

[12] Rogoswami. (2023, April 4). Dogecoin jumps more than 30% after musk changes Twitter logo to image of Shiba Inu. CNBC. https://www.cnbc.com/2023/04/03/dogecoin-jumps-over-30percent-after-twitter-changes-logo-to-doges-symbol.html

[13] Sahayak, V., Shete, V., & Pathan, A. (2015). Sentiment Analysis on Twitter Data. International Journal of Innovative Research in Advanced Engineering, 2(1).

[14] Tellez, E. S., Miranda-Jiménez, S., Graff, M., Moctezuma, D., Suárez, R. R., & Siordia, O. S. (2017). A simple approach to multilingual polarity classification in Twitter. Pattern Recognition Letters, 94, 68–74. https://doi.org/10.1016/j.patrec.2017.05.024

[15] Twitter. (n.d.). The X rules: Safety, privacy, authenticity, and more. Twitter. https://help.twitter.com/en/rules-and-policies/x-rules

[16] What is ridge regression?. IBM. (2024, April 10). https://www.ibm.com/topics/ridge-regression

[17] What the finance industry tells us about the future of ai. Harvard Business Review. (2023, August 11). https://hbr.org/2023/08/what-the-finance-industry-tells-us-about-the-future-of-ai