Predicting Tesla Stock using Twitter Sentiment Analysis

Bing Gu David Perez Ethan Piper

1. Introduction

Our team, The Great Regression, decided to do our project on predicting Tesla stock prices. Elon Musk, the CEO of Tesla, is well known to be very active on Twitter. As a result, his various companies, including Tesla, are discussed very frequently on the platform. Therefore, we decided to try to use the tweets posted about Tesla and Elon Musk to predict stock prices.

One of the reasons we thought this would work is because there are well-known examples of Elon Musk's tweets seemingly having a large impact on the price of Tesla stock. On April 1st, 2018, Elon Musk tweeted saying that Tesla had gone "completely and totally bankrupt". The following day, Tesla stock fell 5% (McGregor, 2018). On August 7th, 2018, Elon tweeted another joke saying "Am considering taking Tesla private at $420. Funding Secured". On the same day, Tesla stock rose 10% (Ferris, 2020). After this specific example, even the SEC took notice (Choi, 2018). While extreme examples are few and far between, it still shows that the content on Twitter can potentially influence the stock market for Tesla.

Other machine learning research done on this topic comes to a similar conclusion. Dounis gathered the most popular tweets mentioning Elon Musk and Tesla over a 4 year period to try and predict the daily prices. After filtering out unnecessary parts of the tweet, they ran sentiment analysis on the tweets and used that metric to predict whether or not the stock went up or down over the course of a singular day. They were able to achieve 82% accuracy on the test set using Ensemble learning. In particular, he found that public perception of Elon Musk was especially predictive of Tesla's stock price (Dounis, 2020).

One study done by Briggs simply used tweets over a specific week and attempted to predict the actual value of the stock on a more per-tweet basis. Their results were questionable at predicting the actual value of the stock, but they showed that there is a general correlation between Twitter sentiment and the trends in Tesla stock (Briggs, 2020).

*Equal contribution . Correspondence to: Jonathan Sprinkle <jonathan.sprinkle@vanderbilt.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

For this project, we decided to follow the model of gathering tweets over a long period of time, running sentiment analysis on the tweets, and then trying to predict whether the stock price would go up or down. Because we believe that tweets have more of an immediate impact, we used tweets for a particular day to predict the price of stock on that day. Therefore, our project tried to predict more current stock prices instead of further in the future.

Our project is also different from other projects because we took into account the stock's relation to the S&P 500 index and the beta value in order to make the prediction. The beta value of a stock is a measure of the volatility of the stock in relation to the stock market as a whole. The beta value of Tesla is recently roughly 2 but has changed over Tesla's life span. When combining this beta value with data on the S&P 500 index, we can determine which of the days Tesla is doing better than the index vs. when it is not. While Tesla stock is technically considered within the S&P 500, it accounts for only 2% of the index and would be very difficult to remove from the data. Therefore we don't think removing it would cause a noticeable difference. This change will hopefully remove the overall trends of the market from our final dataset (Kenton, 2021).

In addition to including a Beta value, we also included data showing the metrics of the specific Twitter user that posted each tweet, such as follower count and favorites count. No other research we could find included this data.

2. Methods

2.1. Dataset

As stated in the previous section, we gathered our data ourselves using a Twitter scraper called snscrape. In order to get the most accurate information, we scraped the top tweets from each day back to 2013. We believe that the top tweets are what Twitter would be actively showing to the most users and also have the most interactions, and therefore are the tweets that would most influence the market. Replies were also excluded from this data. We stopped gathering the data on April 15th, 2022, which was roughly a day before the day we gathered it. This allowed us to make sure that the interaction data on all the tweets had already plateaued. We gathered 4 categories of tweets: tweets about Tesla, tweets posted by Tesla, tweets about Elon Musk, and tweets posted by Elon Musk. Finally, we used YFinance to acquire daily stock data on Tesla and Zacks to acquire the beta for Tesla stock each month.

2.2. Pre-processing

In the pre-processing step, we first did sentiment analysis on every single tweet. This gave us a value between .5 and 1 and another metric of whether it was positive or negative. We transformed this score into two different metrics: signed sentiment and continuous sentiment. Signed Sentiment simply takes the value we received and simply multiplies it by -1 if it is negative. Continuous sentiment takes the value and transforms it into a range from -1 to 1.

With these values, we combined each day into one sample. For each day, we created a few more features that weight the average daily sentiment by some of the other tweet metrics, such as like count or user followers count. We did this both for the signed sentiment and the continuous sentiment. This resulted in 37 different features, such as "average continuous sentiment weighted by like count" and "average signed sentiment weighted by user friends count". This allows us to get an overall sentiment value for a whole day but allows tweets with more likes to be counted higher than tweets with less. We also kept in the original basic features such as total like count and total retweet count.

For each day, we created 4 different labels or target values for each day. These 4 values are percentage increase, adjusted percentage increase, label raw increase, and label adjusted increase. To calculate these features, the first step was to calculate the percent change of the closing and opening value of the stock. The target values that are "adjusted" take into account the beta and S&P 500 index. The formula below shows how the adjusted percentage increase was calculated.

```
Adjusted percentage increase =
stock percentage increase -
index (S&P 500) percentage increase *
stock beta
```

Label adjusted increase is just a value that is 1 if the adjusted percentage increase is positive and 0 if it is negative. The same transformation is applied to the raw percentage change to create the label raw increase feature. However, neither raw nor adjusted seemed to consistently perform better in tests. In addition, our dataset labels show that Tesla stock went up 50.7% of the time, meaning there will be hardly any bias in our models only predicting up or down.

Because anecdotal evidence tells us that the stock price reacts to how it performed on previous days, we include the above four statistics of the previous two days as features for each day.

2.3. Feature Selection

In order to run feature selection, we used sklearn's SelectKBest algorithm using the f_classif metric. This is a metric that relates the overall variance of a specific feature relative to the variance of the dataset. The other algorithm we tried for SelectKBest was f_regression, which calculates the correlation between each feature and the target values. While f_regression may seem more useful, we found better initial results in training when using the features extracted using f_classif.

Using this method, the main features that were extracted were the percentage increase on the previous day and the average sentiment weighted by followers count and favorites count which both reflect the popularity of the user tweeting. However, this only performed better when running on neural networks as explained below.

2.4. Approaches and Experiments

Even though we started with 4 datasets, we found in our initial trials that the datasets containing tweets about Tesla and about Elon Musk performed better than the other 2 datasets, so we decided to focus on those two only.

The different machine learning approaches we used were linear regression, logistic regression, SVMs, and Neural Networks. We used keras to implement the neural network and sklearn to implement all other approaches. For all of these examples, we ran k-fold cross-validation and determined the average score over all tests. Specifically for the neural networks, we used keras's tuner along with k-fold cross-validation to select the best hyperparameters. After determining the best hyperparameters, the resulting model was then tested again on a second held-out set for validation.

Neural networks require a lot of data to train. While we know our dataset is not very large, our labels only have 2 possible values, 0 or 1. To make the dataset smaller, we used SelectKBest using the f_classif algorithm. While we still do not have sufficient data to use neural networks, we still received some of our best results using neural networks.

3. Results

For all linear regression, logistic regression, and SVMs, we gathered the average scores after running the algorithm 100 times using different training and testing sets. For simple linear regression, we got an R2 value of around -0.01. This is a very bad fit since the R2 value should approach 1 as the fit gets better. This shows that the linear regression model is not very good at predicting how much Tesla stock will rise or fall based on tweet sentiments.

Linear Regression	Accuracy with all features (R^2 measure)	Accuracy with feature selection (top 20 features)
All tweets with Tesla	-0.01909	-0.01191
Tweets including Elon/Musk	-0.030896	-0.01651

For logistic regression, the accuracy for both of the datasets were just barely over 50%, with or without feature selection. The only noticeable difference is that in Tweets including 'Elon or Musk', the logistic regression without feature selection was over 52%.

Logistic Regression	Accuracy with all features	Accuracy with feature selection (top 20 features)
All tweets with Tesla	0.506	0.508
Tweets including Elon/Musk	0.522	0.508

The support vector machine model on average did a bit better than the logistic regression model. There were more times where the accuracy on average was above 51%. For the most part, the linear kernel was just slightly better than the radial basis kernel SVM. In the same way, we can see the same accuracy trend favoring the model with all features compared to the model with feature selection.

SVMs Linear Kernel	Accuracy with all features	Accuracy with feature selection
All tweets with Tesla	0.510	0.506
Tweets including Elon/Musk	0.524	0.510

SVMs RBF Kernel	Accuracy with all features	Accuracy with feature selection (top 20 features)
All tweets with Tesla	0.507	0.523
Tweets including Elon/Musk	0.511	0.501

To make sure our model is not purely just predicting a single option (whether rise or fall for stock), we did a confusion matrix on one instance of our model. The confusion matrix for linear SVM showed that the model is not purely predicting one choice and getting 50% correct. However, the model is wrong quite frequently on the testing set. The model predicted a lot of false positives, which is also known as type 1 errors. This is most likely because our training set had many more positive samples than negative ones which resulted in the more positive predictions despite the testing set being necessarily more negative due to the inverse relationship of the training and testing set.

```
[[ 62 105]
 [ 48 101]]
```

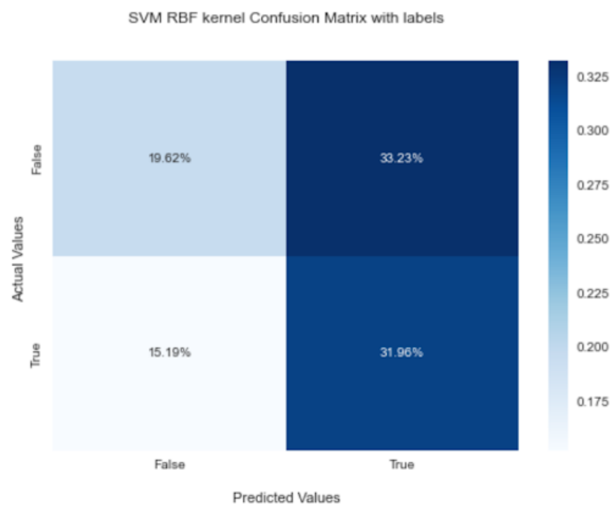

Figure 1. Confusion Matrix for an SVMs with the RBF Kernel

The result of a simple, fully connected neural network model using ReLu for the hidden layers and sigmoid for the output was very similar to the other models, which is just barely above 50% accuracy. There is also no significant change in accuracy between the model with simple feature selection and the model without.

Neural Network	Accuracy with all features	Accuracy with feature selection
All tweets with Tesla	0.510	0.505
Tweets including Elon/Musk	0.507	0.511

For the more sophisticated neural network model with hyperparameter selection, the results with that feature selection did significantly better than regular neural networks. This model was able to reach over 53% accuracy. However, as stated above, there is not enough data in our dataset to truly train a neural network well.

Neural Network with Hyper-parameter Tuning	Accuracy with all features	Accuracy with feature selection
All tweets with Tesla	0.525	0.509
Tweets including Elon/Musk	0.506	0.532

Since our accuracies were so close to 0.50, we sought to prove that our most successful scores were not just randomly high. We calculated the mean and standard deviation of 100 runs of Logistic Regression. We used this to reject the null hypothesis that the true score is 0.51. Additionally, because we have learned that accuracy can be misleading in certain situations, we calculate the F1 Macro, since the test set might not have an equal proportion of positive and negative samples. As you can see each p-value is in fact less than 0.01, suggesting that we can reject the null hypothesis that the true score is 0.51 which leads us to accept that our numbers are not just numerical flukes.

Statistics of Scores	Accuracy	F1 Macro
Mean	0.518	0.516
Std. Dev	0.0231	0.0239
p-value	0.000119	0.00200

Finally, we ran a test using our logistic regression model to determine if we would actually make money off of our algorithm instead of just holding Tesla stock. On days that our algorithm predicted we should buy stock, we bought it at the start of the day and sold it at the end of the day. Otherwise, we simply did not buy the stock. Over a random selection of days, our model made an average of 21% more money than an investor who held Tesla stock on each of those days. However, this is not a comparison to how well a professional trader might be able to make money. We suspect that a professional trader would make significantly more than our model.

4. Discussion & Conclusion

In all 4 machine learning models we selected, the results were all very similar. In the case of linear regression, we realized that the simple linear model is just really not a good model to use for the goals of this project. On the two datasets where we applied the other models to, we got an accuracy in the range of 50-54%. This, just like the linear model, did not do a very good job of predicting the percent stock price change from the beginning of the day to the end. A 50% is what you'd get with purely random predictions. We were only able to predict on average slightly better than that with all of our models. With these results, there is a good chance that Twitter does not have any noticeable impact on stock prices after all. As stated in the introduction, Elon Musk clearly has some influence over the Tesla stock. However, Twitter's opinion on Elon Musk and Tesla do not seem to influence the stock market as much as one would originally think. Perhaps the stock market is a much more complex system and a sentiment analysis on one social media platform cannot fully capture that complexity to make predictions on the percentage change of a stock at the end of the day. Even so, our algorithm was able to perform better than simply holding the stock over a certain duration. This shows that even a small advantage over a coinflip could potentially be very useful in the world of stocks.

Despite this conclusion, some of the previous studies did find more favorable results. One of the previous research we looked at was able to get a max accuracy of 77% on SVM with a radial kernel. However, they only looked at tweets from October to the end of 2019, which is only a 3 month period. Our project, however, gathered tweets since 2013 for our dataset. This may explain why they were able to get a higher accuracy compared to our own since their dataset and timeframe are much more controlled.

To prove that the shorter time period is impactful on the results, we split our dataset into quarters (denoted by quarter-end) and ran 100 Logistic Regressions on each quarter, and graphed the average accuracy and F1 Macro score below. The scores vary widely with some quarters having as good results as 0.73 and others having as poor scores as 0.31. The quarter mentioned in the above research, 2019 Q4 was in the time period that we achieved an accuracy of 0.57 which was significantly higher than the roughly 0.52 we achieved over the entire time period (Edman, 2020).

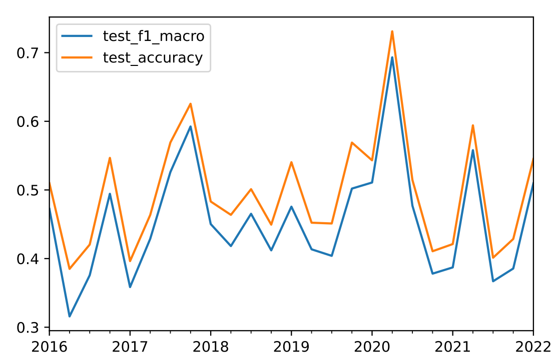

Figure 2. Average scores for the dataset split into quarters

Other research also used other categories of data such as news headlines in addition to Twitter to predict stock prices. This might have given them an advantage over our findings since more people will probably make stock decisions based on bigger and more important news headlines compared to random comments on twitter. Thus, we believe their predictions would be a more well-rounded approach, using more comprehensive data from other platforms in addition to Twitter to predict the complexity of stock prices.

One of the more challenging aspects of our project was gathering our dataset. This took a significant amount of time to learn to scrape tweets and a significant amount of computing time to do the sentiment analysis. After that, we still had to do pre-processing and also gather the stock data for the target values. Another very challenging part of our project was knowing if our numbers were actually significant or just by chance because they were so low.

Unexpectedly, we found that following the algorithm in buying and selling Tesla stock is actually better in the long run than just holding the stock for a long period of time. It was also generally unexpected that the sentiment of tweets and the trends of the Tesla stock were so uncorrelated. Based on previous research, we would have assumed that there would be a significant correlation.

If we were to continue to research this subject, we would gather more data so that we could actually prove that Twitter and Tesla do not correlate. We would do this by reaching out to other researchers and asking if we could borrow their datasets to see if we reach the same conclusions. In addition, we would do more analysis to see if all our numbers are statistically significant enough to say that Twitter does or does not influence the stock price.

5. Acknowledgements

Our team generally worked together for the project, but we did separate some specific tasks. Ethan worked on collecting Twitter data, running the sentiment analysis, and implementing hyper parameter selection with the neural networks. David worked on collecting stock data, the data pre-processing, cross-validation, and the stock simulation. Bing worked on data visualizations, linear and logistic regression, SVMs, and basic neural networks.

6. Code Overview

get_price_data.ipynb
Gets the TSLA stock price and S&P 500 stock price from yahoo finance and gets TSLA's beta value from Zacks. Saves this data to csv files.

gather_tweets_snscrape.py
Gather all the tweets starting from 2013 to 2022 related to Tesla and Elon Musk and saves them to files based on the day they were gathered.

sentiment_on_jsonl.py
Reads each tweet file and cleans up the text in order to perform sentiment analysis. Loads tweet into csv file.

data_agregation.ipynb
Combines tweets and stock data into daily sample values and saves this data to csv files. Also does model training and analysis.

feature_selection_and_NN.ipynb
Our implementation of feature selection on the various datasets and the neural networks with hyperparameter selection.

bing_project_code.ipynb
Using the dataset created to graph data visualizations and do linear regression, logistic regression, SVM, Simple Neural Network, and confusion matrices.

References

Briggs, J. Sentiment analysis for stock price prediction in python, 2020. URL https://tinyurl.com/briggsDataScience.

Choi, J. L. Sec complaint: Elon musk, 2018. URL https://www.sec.gov/litigation/complaints/2018/comp-pr2018-219.pdf.

Dounis, F. Elon's golden gift: Predicting the stock price of tesla with twitter and machine learning, 2020. URL https://tinyurl.com/dounisDataScience.

Edman, G. Predicting tesla stock return using twitter data, 2020. URL http://lup.lub.lu.se/student-papers/record/9025587.

Ferris, R. Tesla shares surge 10 percent after elon musk shocks market with tweet about going private, 2020. URL https://tinyurl.com/ferrisDataScience.

Kenton, W. Beta, 2021. URL https://www.investopedia.com/terms/b/beta.asp.

McGregor, J. Elon musk's april fools' tweets were 'not a joking matter,' experts say, 2018. URL https://tinyurl.com/mcgregorDataScience.