Tesla Stock forecasting feature analysis

Austin Li (jl3273), Lang Lei (ll674), Shichen Qi (sq89)

1. Introduction

In the modern quantitative research industry, innovation and the use of previously neglected data sources are what push the industry forward. Apart from traditional methods of stock price forecasting, which heavily associate with time series analysis, we’d like to explore ways of effectively performing such analysis and predictions without regarding the time series properties(moving average etc.) In this project, we investigate the influence of external factors as additional features to traditional forecasting models, comparing the performance of any combination of them as well. For instance, we will operate the sentiment analysis of Elon Musk’s Twitter to see if it fluctuates Tesla’s closing price.. State-of-the-art forecasting techniques such as recurrent networks and deep learning are referenced and modified in this project so that it is technically relevant in 2021.

2. Dataset

2.1 Dataset Description

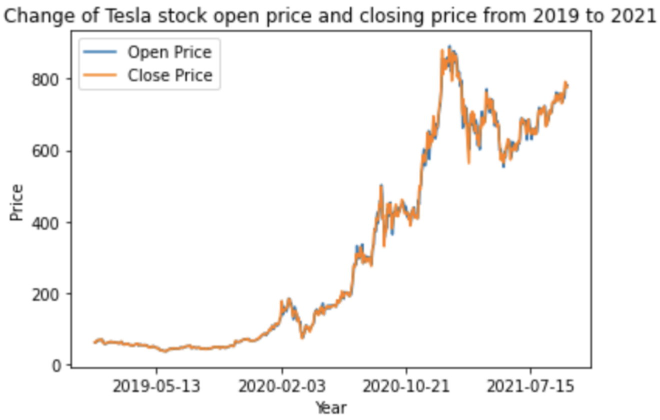
In order to analyze the relationship between Elon Musk’s Twitter content and Tesla stocking price, we subtracted Elon Musk’s Twitter data from the Twitter API, as well as stock pricing data of Tesla and its main competitors (Volkswagen, General Motors and Ford) from Yahoo Finance.

Competitors Stock Price dataset:

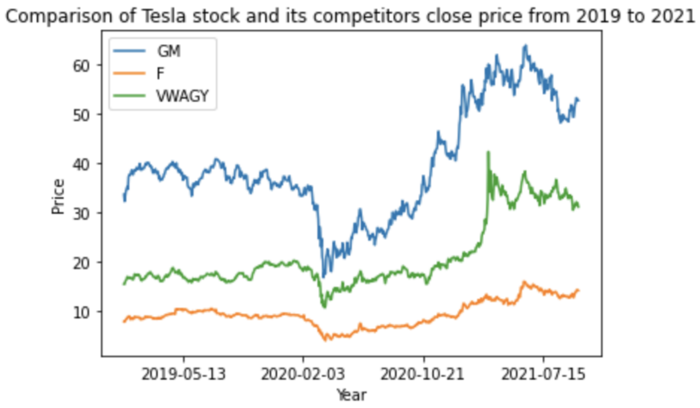
In this dataset, we collect three of Tesla's major competitors (Volkswagen, General Motors and Ford) open and close stock prices from Jan 1, 2019 to Sep 31, 2021.

Tesla Stock Price dataset:

In this dataset, we collect Tesla’s open and close stock prices from Jan 1, 2019 to Sep 31, 2021 shown below.



To get a better understanding of the relationship between each brand, we combined two datasets and utilized two line graphs to show the change of stock price from 2019 to 2021 shown below.



It is noticeable that the trend of General Motors looks more similar to the trend of Tesla, implying that principal component analysis may be conducted in the future topic of interest. Additionally, we add stock price difference columns for each brand to get more features to our model. One of the benefits by doing so is that we can easily tell the relationship of the stock price for the same brand.

Twitter dataset:

We extracted the content of Musk's Twitter from Sep 31, 2020 to Sep 31, 2021 using Twitter API.

2.2 Feature Engineering

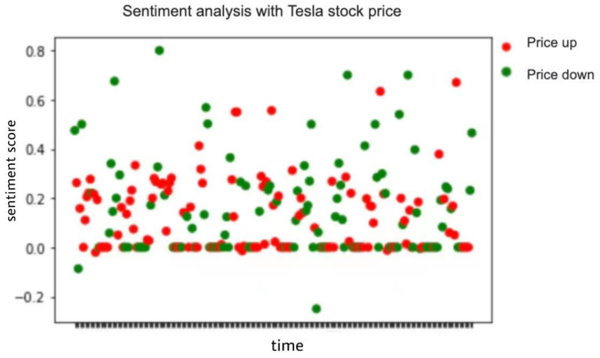
Twitter dataset:

1) Cleaning:

The main datasets that we extracted from the internet are Twitter. For the Twitter dataset, we used the regular expression to delete the user name (for instance, "@xxx"), the image url and the reference url, and then collected all the content in the same format. There would be multiple tweets per day, but we only need one measurement connected to each day. Thus, we decided to aggregate the tweets of the same day, and then conduct the sentimental analysis.

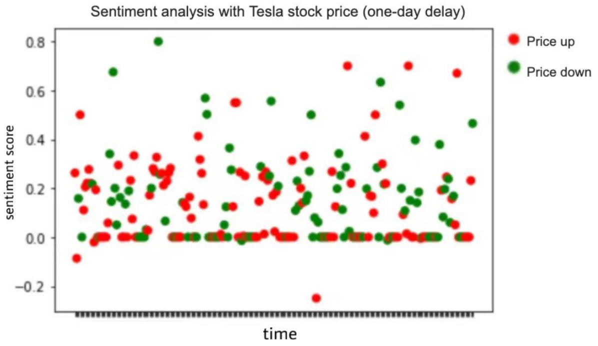
2) Sentiment Analysis:

We imported Textblob to implement the sentiment analysis and plot with the information of Tesla stock going up or down.



It is shown from the graph above that if Elon Musk's tweets have positive attitude or tone, the scatter points are above zero while if they are negative, scatter points are below zero. Points lying on the horizontal line of zero represent the neutral tone. For the stock price, green points mean price going up while red points represent price going down.

We also performed the sentiment analysis of Musk’s tweets and Tesla stock price with one day delay.

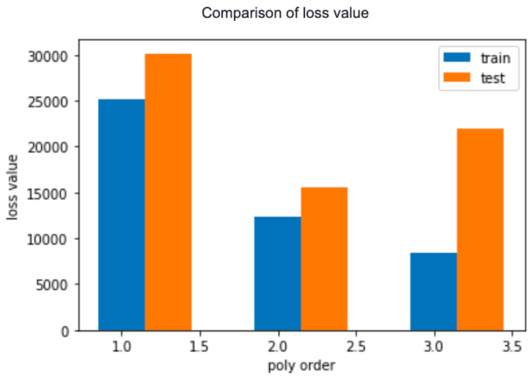


The graph shown above illustrates that if Elon Musk’s tweets one day before have some influence on the second-day stock price. It is noticeable that both graphs show some trend between tones and stock price, which is reasonable to conduct random forest in our model.

3. Model

3.1 Avoid overfitting

In order to find the best degree in different polynomial transformations, we compared the different performances according to the Mean Square Error (MSE). We used PolynomialFeatures from sklearn to construct the transformations with different exponents. Since the function generates all polynomial and interaction features including all polynomial combinations of the features with degrees less than or equal to the given degree, the number of features increases exponentially which means it is easy to cause overfitting. Therefore, we tried 1, 2, 3 as the given degree and compared the loss values. The data were divided into training and testing sets and calculated the MSE separately.

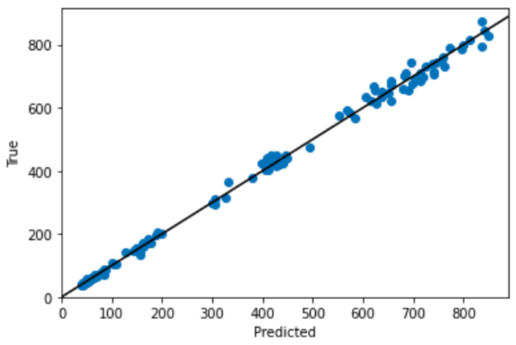


From the plot, when the degree increases from 1 to 3, the MSE of training sets decreases, while the MSEs of testing sets decrease and then increase, which means the model starts to be overfitting when the degree is larger than 2.

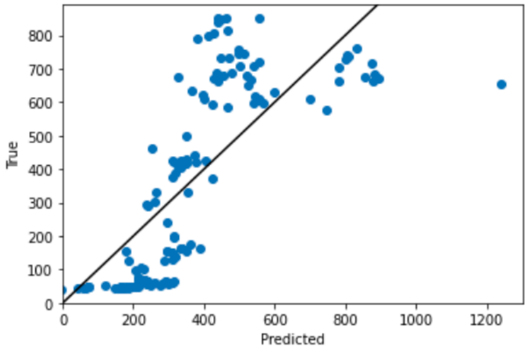
3.2 Model Performance Analysis

3.2.1 logistic regression

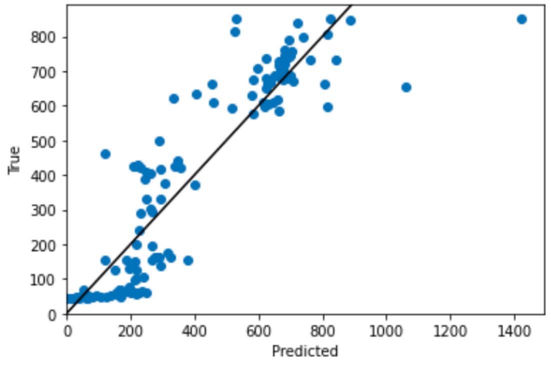
In the first part, we used the all the history stock prices, including both Tesla and other stocks, to predict the stock price of Tesla. From the plot below, we can see that the prediction and true data are almost identical and fit the linear regression model.


Logistic regression on all stock prices (Tesla and other)

In the second part, we excluded Tesla from the data set and only used other stocks to predict the stock price of Tesla. According to the plot below, much more outliers are apart from the linear regression line. This is a reasonable result since our prediction only depends on the market performance without Tesla itself, which might cause larger error from the actual price.

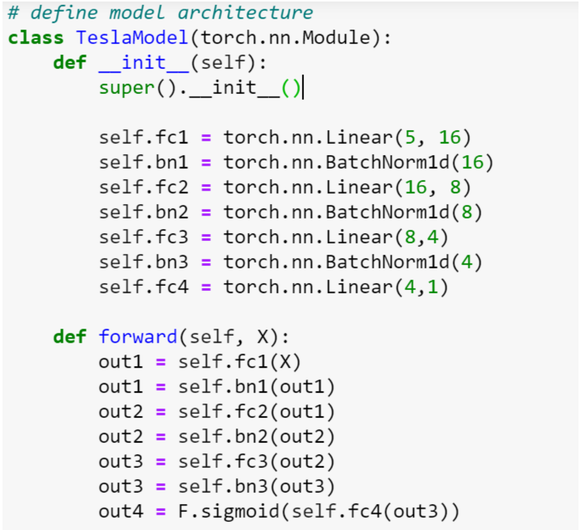

Logistic regression on all other stock prices (exclude Tesla)

Lastly, we still used the historical stock prices excluding Tesla as our data set. This time we tested the performance of the polynomial transformation of degree 2, since in the previous Avoiding overfitting part we found 2 is the best degree in the model. Specifically, less outliers are presented in the plot, which means the polynomial transformation model fits the data set better than the linear regression.


Polynomial transformation (degree 2) on all other stock prices (exclude Tesla)

3.3.2 Neural Network

After we did the feature engineering for the twitter dataset in the data processing part, we found out that there is a correlation between price and tweets tone. In the last models, we predicted the stock price with a polynomial transformation model and decreased the number of outliers to the actual price. Therefore, we decided to find the deep underlying relationships between the price trend and other factors. We used stock prices of the three competitors (GM, F, VWAGY), the sentiment scores and the Tesla's trend (True or False) the day before as predictors to predict the price trend the next day.



```
# define model architecture
class TeslaModel(torch.nn.Module):
    def __init__(self):
        super().__init__()

        self.fc1 = torch.nn.Linear(5, 16)
        self.bn1 = torch.nn.BatchNorm1d(16)
        self.fc2 = torch.nn.Linear(16, 8)
        self.bn2 = torch.nn.BatchNorm1d(8)
        self.fc3 = torch.nn.Linear(8,4)
        self.bn3 = torch.nn.BatchNorm1d(4)
        self.fc4 = torch.nn.Linear(4,1)

    def forward(self, X):
        out1 = self.fc1(X)
        out1 = self.bn1(out1)
        out2 = self.fc2(out1)
        out2 = self.bn2(out2)
        out3 = self.fc3(out2)
        out3 = self.bn3(out3)
        out4 = F.sigmoid(self.fc4(out3))
```

The code chunk shown above expresses the definition of the architecture of our neural network as well as its forward function. In the training step, we used 100 epochs and a constant learning rate of 1*10^-4 without learning rate decay. The loss values were calculated with a binary cross entropy loss function because our output is binary.

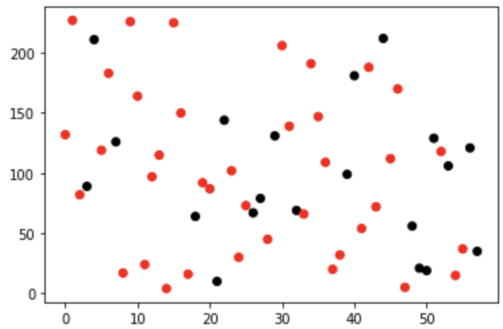
This accuracy, though below 0.5, has been the best we could obtain from modifying the parameters.

However, we aimed to get a better prediction of our analysis. Thus, we decided to get rid of several inputs and change to a new model to see if it can improve the performance.

3.3.3 Random Forest

In the previous feature engineering part, we conducted sentiment analysis of Elon Musk’s tweets with the performance of Tesla stock price on the same day. The visualization suggests that there is a correlation between two variables.

So for the third model, we are interested in using a random forest model to separate the tweets that potentially have positive influence on the market and otherwise. Compared with the traditional time series model, the feature of this random forest model is to see the connection between price and content of Musk’s tweets. We can only tell if the price is going up or down based on Elon Musk tweets. For this model, we both calculated the accuracy of the same-day input and the one-day delay input response. They are around **0.655** and **0.431** separately, which means for the same-day model, about 65.6% of the predictions are correct while for the one-day delay model only 43.1 percent of price trend predictions are correct.



The graph presented above shows the accuracy for the random forest model of the same-day input. Red scatter points represent the right prediction while black points show the wrong prediction.

4. Conclusion

It is obvious that the history stock price data set containing Tesla predicts the stock price best. If we exclude Tesla, the polynomial transaction model with degree of 2 improves the prediction compared to the linear logistic model. Then we calculated the accuracy of predicting price trends with different models. In the next model, we used 3 competitors' price, sentiment scores and Tesla price to predict the price trend, but the accuracy is lower than 0.5 which is meaningless. So we decided to reduce inputs and the random forest model proves to be effective. The accuracy increases from 0.431 to 0.655. Though our training set is small and it produces higher variance in the prediction, the model still shows a correlation between the price and Twitter content.

4.1 Weapon of Math Destruction

In the logistic regression model, our output is the predicted stock price based on different stock combinations. Thus, the outputs are floats that are easy to measure.
In the neural network model, we calculated if the Tesla price will go up or down and classified it as True and False. These binary outputs are also measurable. The output of the random forest is the same as the previous one.In all, all our response variables can be measured quantitatively.

Our models use the stock prices and Twitter from Musk to predict Tesla’s stock price and if the price will go up or not. All the resources and references are open to the public which means people are free to use them to build their own models and predict the prices. Thus, our model will not harm anyone.

Finally, since we only used history stock prices and Twitter content to predict the price, it will not create a feedback loop since we do not use predicted values as our features.

In conclusion, our project might not produce a Weapon of Math Destruction.

4.2 Fairness

Our team do not think fairness is very important to our models. We used all the stock prices and relevant Twitter information as our data set, so there is no discrimination and bias in the data set. Also since we try to predict Tesla's stock price, a small error to actual price is acceptable since a company will never invest in a single stock and a portfolio can also decrease the risk. Finally, predicting stock price will not affect legal status. Thus, fairness is not an important factor in the models.

5. Limitation and future improvements

5.1 Limitation

The largest limitation is lack of data. The first limitation is due to API. We are only allowed to extract one year of Twitter data and this insufficient data will increase variance and cause bias. Also we can only get the content of tweets and there is no number of likes and repos, which might also influence the prediction. Moreover, we planned to add the information of successful launch in SpaceX as our feature, but it was hard to get from the website. Finally, we only got open and close prices from the stock information, but there's insufficient information about the stock, such as volume, market capital, high and low prices.

5.2 Improvement

We can also use positive and negative news on Tesla correlated company to make important database and stock performance to better predict the stock price.

6. Appendix

1. SG:pub.10.1007/978-1-4614-9372-3 - springer nature scigraph. (n.d.). Retrieved December 5, 2021, from https://scigraph.springernature.com/pub.10.1007/978-1-4614-9372-3.

2. *Sentiment analysis of Twitter data - ACL member portal*. (n.d.). Retrieved December 5, 2021, from https://aclanthology.org/W11-0705.pdf.

3. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2018, June 5). *Scikit-Learn: Machine learning in Python*. arXiv.org. Retrieved December 5, 2021, from https://arxiv.org/abs/1201.0490.

4. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2018, June 5). *Scikit-Learn: Machine learning in Python*. arXiv.org. Retrieved December 5, 2021, from https://arxiv.org/abs/1201.0490.