On October 27, 2022, Elon Musk, the famed owner of Tesla and SpaceX, acquired twitter for a total price of $44 billion (Korn, 2022). Arguably, most, if not all, social media, in some cases, is dipped in the paint of chaos. Twitter is no exception and is often the center of many opinions on social events. Long before the idea of acquiring Twitter became public knowledge, Elon was a known social media troll. One has to look no further than 2021 when he was quoted as saying, "I keep forgetting you're still alive" to Senator Bernie Sanders after he suggested the wealthy pay their "fair share" (Maruf, 2021). Since his acquisition, the tables have been turned with Elon in sight. With this somewhat significant change in social perception, the wall of tweets appears more volatile than ever.

I am interested to see if I can use a model trained on different tweet data to predict the sentiment of Elon's tweets. Then look to see if any social metrics correlate with the sentiment of his tweets. A model like such is essential as it can be a solution or prodromal to a solution that helps others understand the context. Given the natural variability in the human brain, there are plenty of disorders making speech and text language challenging to interpret. Looking at someone responding or interacting with such volatility will prove a good challenge.

To get to the point of predicting Elon's data. I first had to train my model on another set of tweets. The "sentiment140 dataset" contains data for 1.6 million individual tweets (Sentiment140 Dataset with 1.6 Million Tweets, n.d.). The entire set contains six different variables; the target variable is defined as sentiment and is coded in 4 (positive) and 0 (negative). As well as ID, flag, user, date, and the tweets themselves. I only kept the sentiment, tweets, and date variables for the actual data analysis. I recoded sentiment for readability to binary 1 and 0, 1 defined as positive sentiment.

I split the date into the day, month, hour, and minutes. Month only comprised of three different months: June, May, and April, which I dummy-coded. For the day of the week, the date of the day, and the hour I ran a harmonic analysis on those same numerics. I did investigate using the year, and found that it only possessed one unique value in both datasets. To get the tweets legible to the computer, I utilized the model “BERT base uncased.” This is a masked language model trained in English that I downloaded from Hugging Face. I then encoded both my first data set and Elon’s tweets data set.

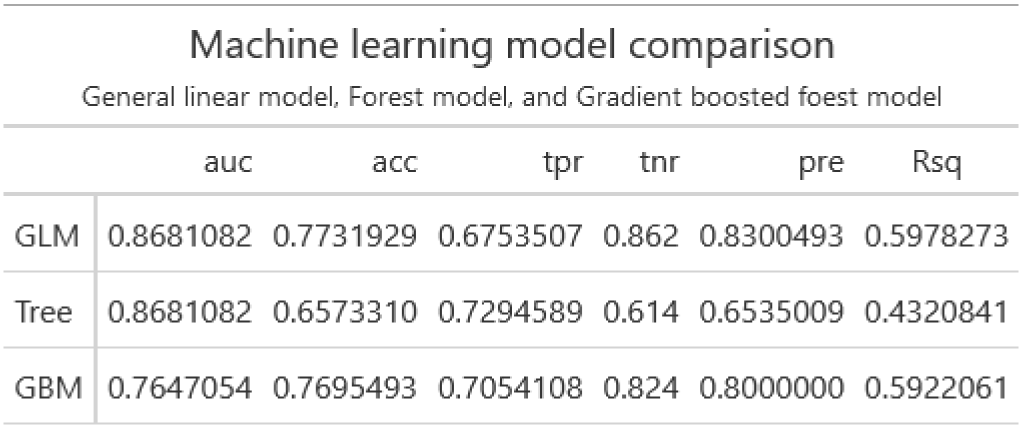
The Elon tweet dataset (Elon Musk’s Tweets Dataset 2022, n.d.) comprises 2668 tweets he posted during 2022. Besides the tweets, it also contains metrics such as the date, likes, and retweets. Like the first dataset, I broke apart the date for this data into its constituents.

For this task, I utilized three different models a simple general linear model (GLM), a decision tree model (TREE), and lastly, a gradient-boosted tree model (GBM). I chose a non-regularized GLM as the anchor point for this project as it is the most straightforward concept to understand. It is a linear model that fits data via a penalized maximum likelihood and is tuned via the lambda parameter. A decision tree model runs all dataset possibilities until it gets the best fit. It uses all possible points from all variables to find splits best capable of reducing the error. The tuning parameters for the decision tree; are minsplit, which is the minimum observations to perform a split; complexity parameter (cp) is this models version of lambda (the penalty term), maxDepth tells the tree how far down to go before stopping, and minBucket is saying to model that it needs one observation for a node to be calculated. The last model, the GBM, is similar to the decision tree because it utilizes nodes and trees.

Nevertheless, the difference is that it works sequentially to correct the errors of the prior tree, and it makes simple decisions and reiterates on them. It had four hyperparameters:

I. N.trees is the number of trees to fit.
II. Shrinkage is the learning rate (how fast we want to arrive at the outcome).
III. Interaction is the maximum depth of each iteration
IV. n.minobsinnode is the minimum number of observations per iteration.

I plan to evaluate model performance using accuracy, true positive rate, true negative rate, and precision.



Machine learning model comparison

General linear model, Forest model, and Gradient boosted foest model

	auc	acc	tpr	tnr	pre	Rsq
GLM	0.8681082	0.7731929	0.6753507	0.862	0.8300493	0.5978273
Tree	0.8681082	0.6573310	0.7294589	0.614	0.6535009	0.4320841
GBM	0.7647054	0.7695493	0.7054108	0.824	0.8000000	0.5922061

Figure 1.

Based on the metrics provided via the model predictions, I am most convinced by the overall performance of the GBM. It covers less area under the curve with similar accuracy to the GLM, which had the highest. It is more sensitive to positive hits than the GLM and less likely to produce a type 1 error. While not accounting for the most precision, it is still within .3 of the GLM and has very similar precision. I am more confident in this model than the other two because it effectively does the same job as the GLM.

Accounting for a smaller area suggests better performance. Because it was a classification problem, I did have to choose a cut-off point. I chose .6 as my cut-off because I did not want to go any lower to hit “at-chance” but felt that going higher may decrease the true positive rate.

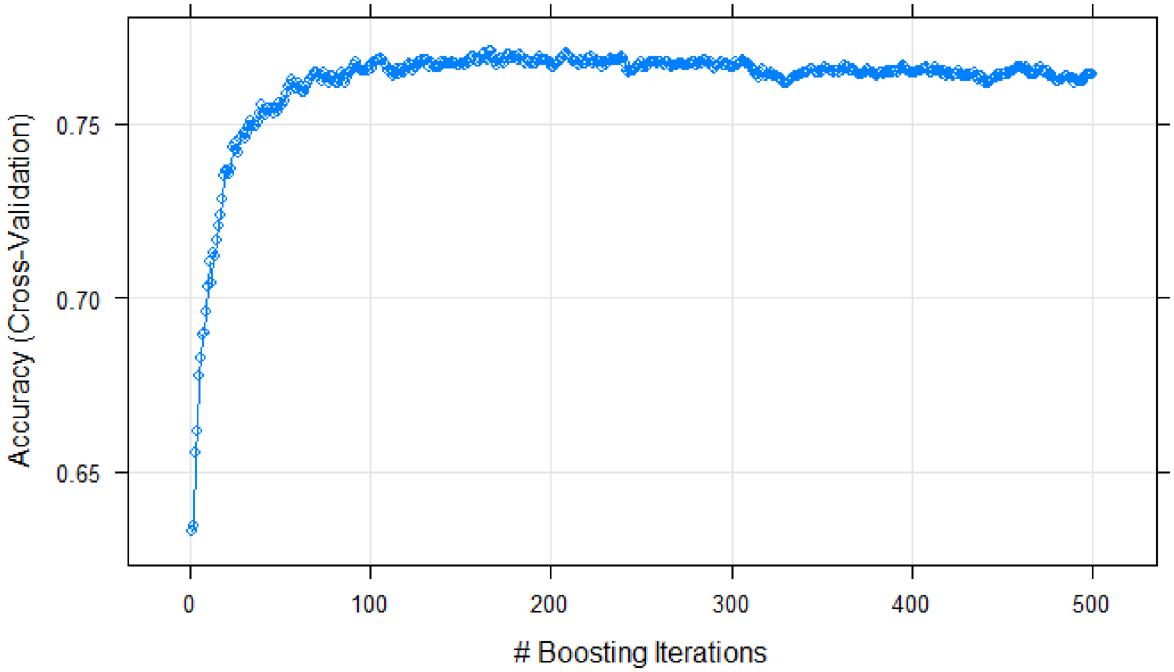


Figure 2.

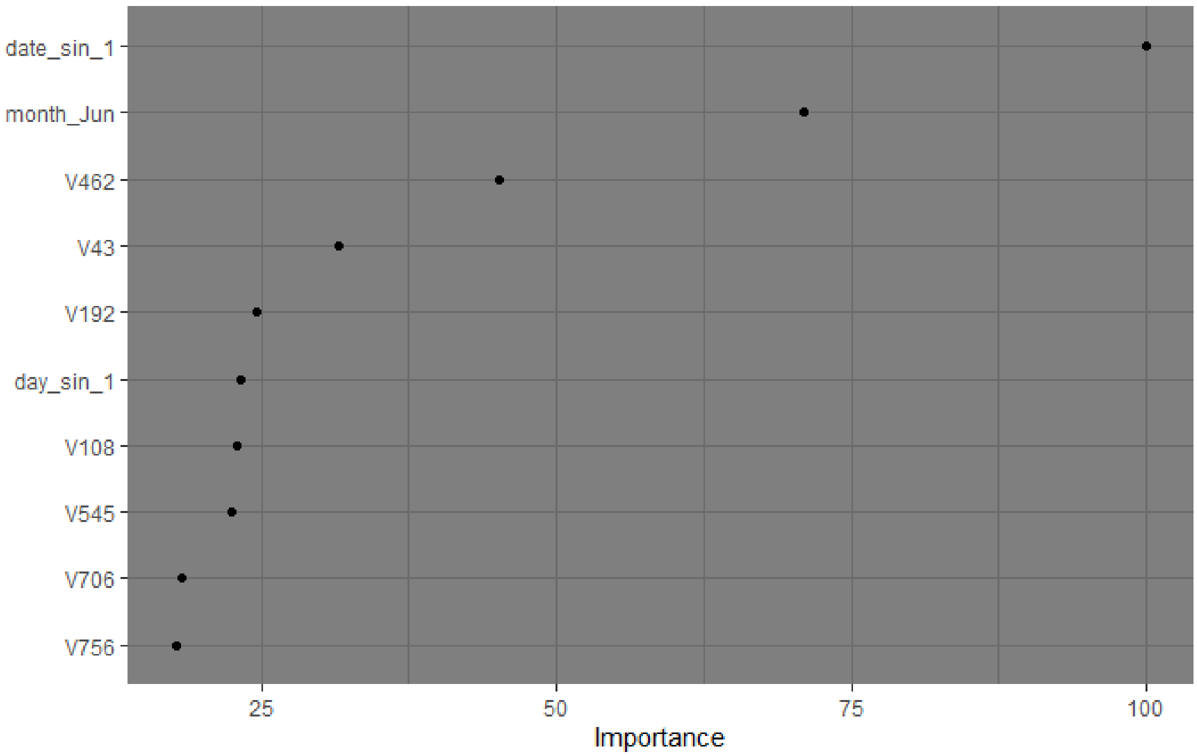


Figure 3.

Based on my conclusion, my model did a decent job predicting not only the tweet test set but also Elon's tweets. Working with a 77% accuracy is certainly nothing to scoff at regarding predictions. When looking at the predictors of importance, the day of the month had the most importance. I am honestly not sure what to make of this. It is surprising. It is hard to rationalize what might be going on there. The month of June is not surprising to me at all. If you think about it in terms of when summer is, that possibly makes sense for the variance seen. Mainly due to the idea that April and May are considered pre-summer months. In Figure 2. I found the variability as the model went on into higher iterations to be quite interesting. I wonder if the predictions themselves would be adding to the noise in the later updated nodes.

As mentioned, the gradient-boosted and general linear models were close, but the GLM accounted for far more of the curve than the GBM. I did expect this to occur, primarily based on the simplicity of a linear model. The GBM performed very well in coding his tweets, just based on my check through the data. I want to mention the social metrics I spoke of earlier in the paper as an outcome of interest. In running a quick linear model to assess the data, neither likes nor retweets demonstrated a significant difference regarding sentiment.

References

Elon Musk's Tweets Dataset 2022. (n.d.). Retrieved December 8, 2022, from https://www.kaggle.com/datasets/marta99/elon-musks-tweets-dataset-2022

Korn, J. (2022, May 17). *Elon Musk's bumpy road to owning Twitter: A timeline | CNN Business*. CNN. https://www.cnn.com/2022/05/17/tech/twitter-elon-musk-timeline/index.html

Maruf, R. (2021, November 14). *"I keep forgetting you're still alive:" Elon Musk trolls Bernie Sanders on Twitter | CNN Business*. CNN. https://www.cnn.com/2021/11/14/business/elon-musk-bernie-sanders-tweet/index.html

Sentiment140 dataset with 1.6 million tweets. (n.d.). Retrieved December 8, 2022, from https://www.kaggle.com/datasets/kazanova/sentiment140