Relationship of factors that determine tweet virality

Shawn Zhu

12/12/2023

1. Abstract

How can Artificial Intelligence predict the popularity of posts on Twitter (X)? Twitter's development in mainstream media has been profound, and as the platform increases more overtime, so does the need for predicting virality of tweets on the platform for opinionation. This paper addresses a growing dilemma: As Twitter becomes ever more influential in the modern landscape, predicting what will go viral can help in the discovery and pushing of new ideas and mitigation of risks associated with online networking. We provide an extensive analysis over a dataset consisting of user data and tweet data to compute a virality rating. Using a learning model, we managed to achieve a maximum of 67.32% accuracy on predicting virality based on all user data based on a 1-5 scale ranking.

2. Introduction

Our research aims to find the answer to the question of whether AI can predict the popularity of Twitter posts. The popularity of Twitter makes it a place to post thoughts and ideas and to have these ideas noticed by influential people around the world alongside the millions of other users. Pushing new ideas and preventing mass panic is relevant in the modern world with events such as gamestop stock and political tweets. To help answer this question, we will work with a supervised learning model of classification in order to classify tweets into different categories of nonviral (1) to viral (5). We work with data converted into numerical data for easier manipulation and organization of different features and had them labeled in a organized.

3. Background

Another article labeled Analyzing and Predicting Tweets highlights similar goals and methodology. They cite that predicting tweets is necessary for opinion formation and the need to understand the factors that make information spread quickly on sites such as Twitter. They also looked at Sentiment Analysis, the detection of emotion through tweets, such as classifying if a tweet had negative or positive tones, which can also have an impact, since normally tweets classified as more negative gain traction easily. Their methodology included a basic linear model trained on features and then given a virality score.

4. Dataset

Our dataset consisted of a collection of both vectorized media, textual media, user descriptions, and images. We have a total of 29625 tweets in our dataset. We consisted of using data from 52 different users with 14 features, with 11 numerical features and 3 categorical features. Our training data has 29625 samples to work with. We preprocessed our data to make our learning model as efficient as possible. We started by modifying the training data by dealing with missing data by figuring out the missing columns of both user and tweet data. We do this to prevent null values from skewing any results and replacing them with an integer of “0”. We also do one-hot encoding for our categorical features (such as topic ids) on every section, meaning we one-hot encode tweets and users. We also do cyclical encoding to ensure temporal features such as time stay formatted correctly. Then we preprocess the test data by doing similar things to the train data by filling null values with integer “0” and creating a final dataframe to have. We then preprocess the test data by filling null values with integer “0” again. We use more cyclical encoding for managing features that involve times and dates, and finally, we gather the image data of tweets and fit the data, doing the same for the user descriptions and the user profile images. We also merge all tables of tweets, users, and training tweets to make a dataframe.

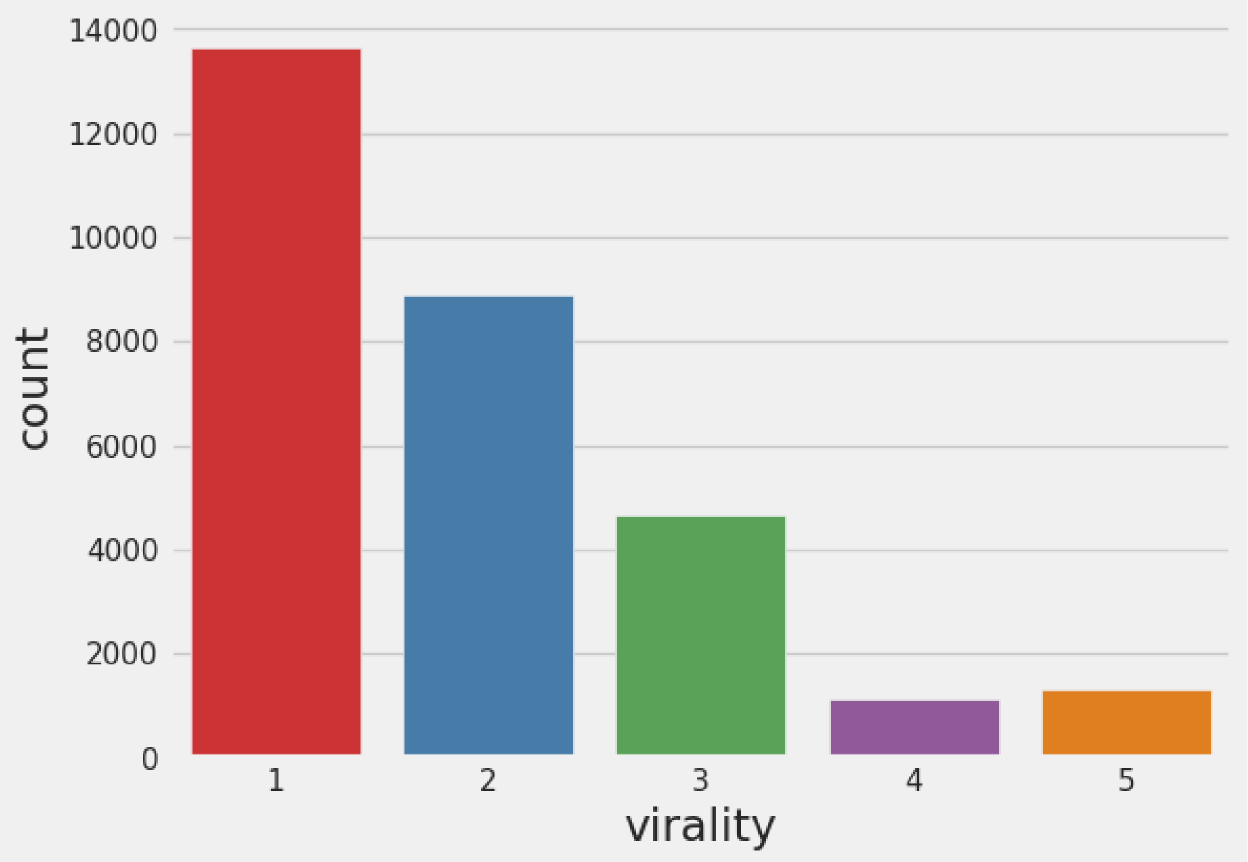

Figure 1: Spread of virality levels over all tweets.

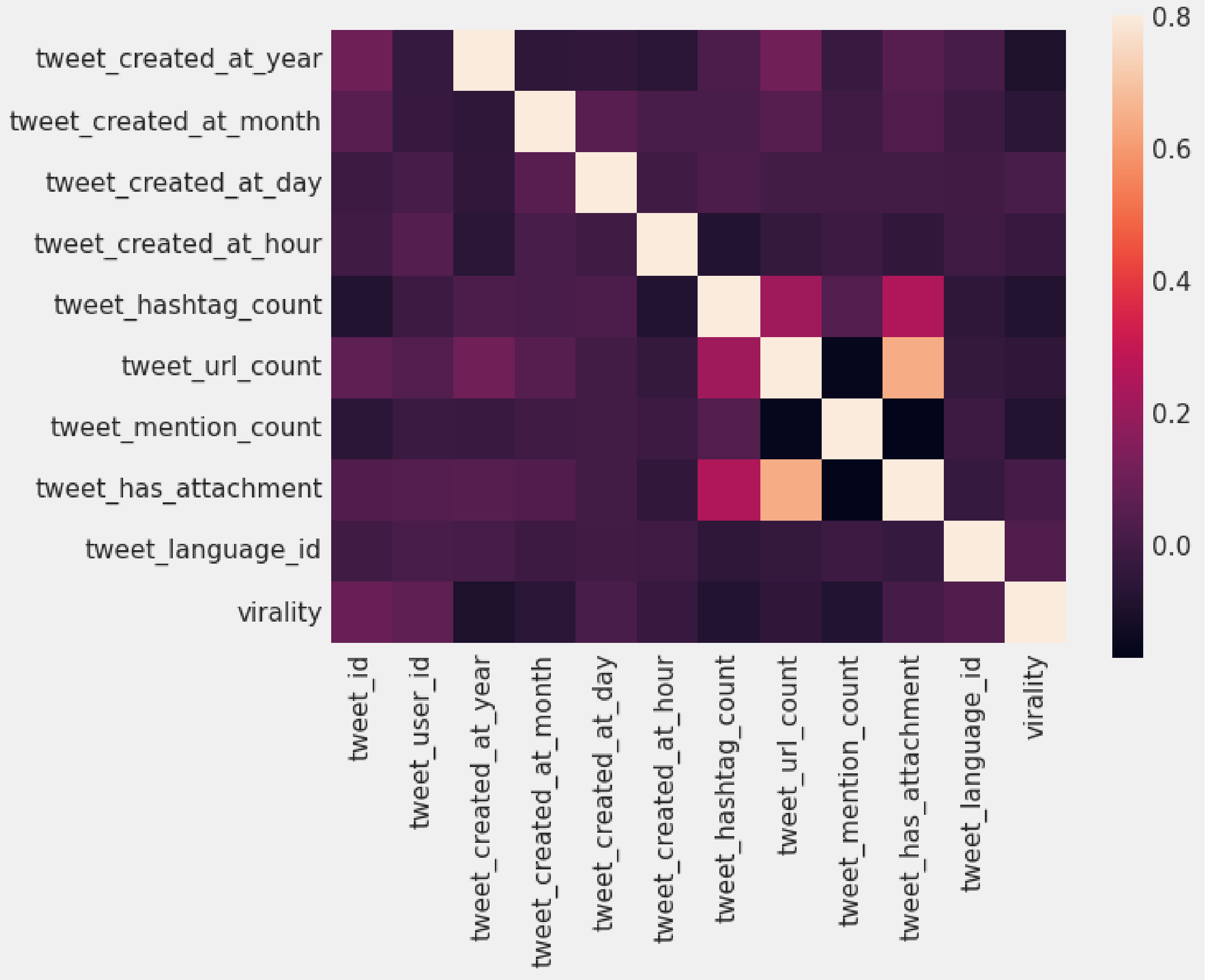

Figure 2: Correlation heatmap of different features and their correlation with virality.

5. Methodology/Models

We chose a classification model because these models' end output is going to categorize an output. This is because we deal with having to categorize a tweet as viral or not, and are seeking these to be classified as such. We used classification models such as RandomForestClassifier as well as a Lightgbm model.

RandomForestClassifier
The RandomForestClassifier is an estimator that fits a number of different decision tree classifiers. Each "tree" is a random subset of data, and then afterwards takes all trees and averages the results from them. We chose a RandomForestClassifier as it has the power of predictive accuracy due to it's random "tree" groupings alongside the controlling of overfitting.

LightBGM Model
We also used a LightBGM model that also used a tree based learning algorithm similar to RandomForest with predictive tree groupings and controlling overfitting. Its ability to handle large-scale data is appealing and the optimizations in speed, alongside the accuracy it provides when working with categorical features, a main component for our models since we categorize a tweet as viral or not for our output.

Neural Network (NN)
We also built a neural network that takes the input like any other classification model, and sets it through multiple hidden node pathways to eventually output an answer at the end. Neural networks are a part of deep learning algorithms and have it work by having multiple node layers that weigh in on the eventual output answer.

Test Train Split
We first did a test train and split to get our training and test datasets, and then fitted our model to our X and y training datasets. Afterwards we have our model predict (based on our training data) the virality of completely unknown tweets and to see the accuracy score it gives. Accuracy score is the only metric we evaluate due to only needing to know of true positives and nothing on false positives or false negatives. Other metrics are useful though in determining overall accuracy as a whole including false positives, false negatives, etc.

6. Results and Discussion

Our models showed promising accuracy scores. The model with the best outcome was the LightBGM model with 67.32% accuracy.
The next highest being the RandomForestRegressor with an accuracy score of 66.51%.
Finally the neural network at 53.99% accuracy.

All of these models took long amounts of time to train on datasets due to our datasets' large sample size of tweets we use of 29625 tweets, alongside our media that may be attached.

The LightBGM model's fit time was 6 minutes 2 seconds.
The second longest was the Neural network model fitting, with a time of 2 minutes 57 seconds,
And the quickest one was the RandomForestRegressor model fit time was 1 minute.

The results show signs of improvement in the model, with lower accuracies than a model would normally want. More hyperparameter tuning alongside more efficient methods of preprocessing data can help influence better results in the future. Potentially different model selection can play a role as well, or even refining which features to use and which to leave out of the calculation.

It is important to note that other metrics (such as recall, f1 score, and precision) were not as important as clearly defining whether the virality rating was correct or not, but does have an impact in evaluating what part of the model it categorized incorrectly.

An example of which is presented in the model that performed the best (LightBGM).
Accuracy: 67.32%
F1_score: 65.43%
Recall score: 67.26%
Precision score: 64.98%

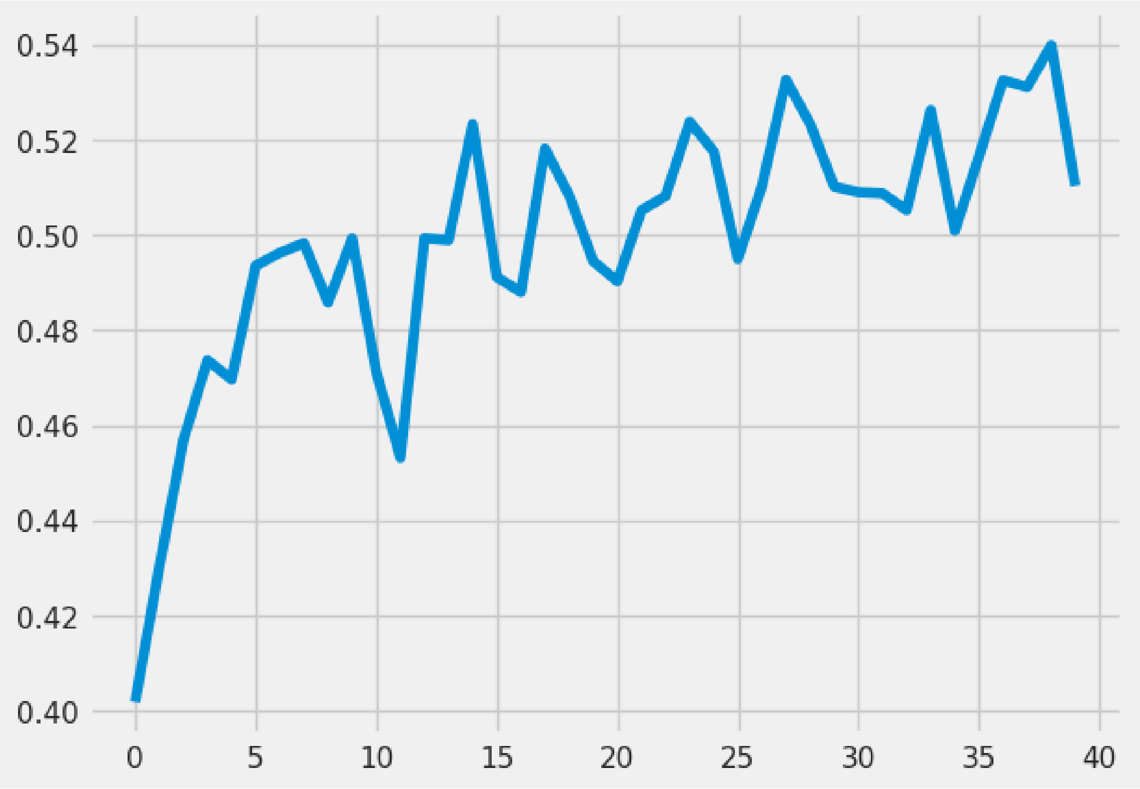

Figure 3: Neural network accuracy over epochs in neural network

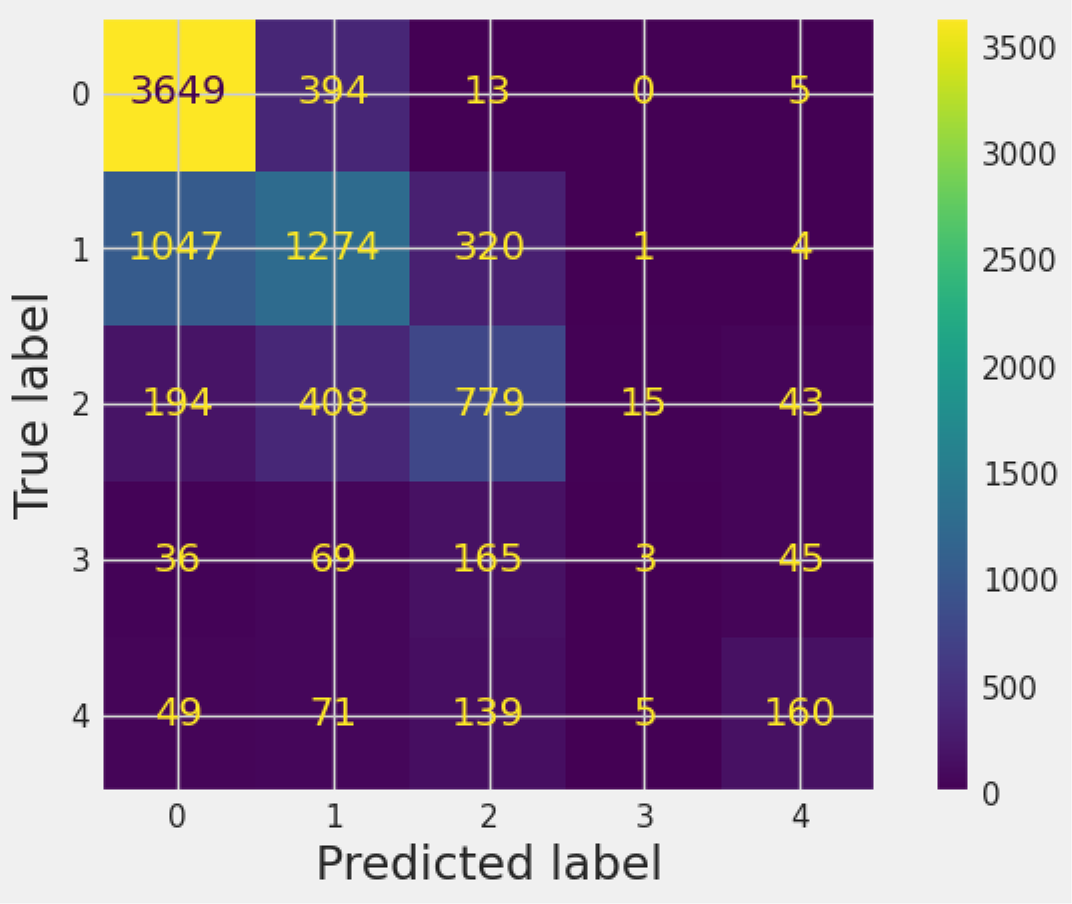

Figure 4: Confusion matrix of RandomForestRegressor on virality levels

7. Conclusion

The overall idea of being able to predict whether tweets go viral or not is possible. With an accuracy on our LightBGM model at 67%, it is possible to take the model to the next potential level. Even if all of the models' performance were poor to subpar, it highlights the potential of collecting the benefits of predicting tweet virality, predicting the next potential big thing to rave about in the modern age. The next steps in predicting virality include a multitude of things, from increasing our accuracy of models, other media apps, to potentially including Natural Language Processing (NLP) into consideration when figuring out textual patterns. A convolutional neural network (CNN) involving image deciphering could be helpful in the creation of a better model to deal with media processing. Accuracy could be fixed with better hyperparameter tuning, model changes, or better preprocessing. Other social media programs have a similar structure to Twitter, in which people can post anything and have a major, if not a bigger impact than Twitter. Twitter's power comes from the ability to reach out to millions about practically anything, and knowing what engages the most people as possible is a benefit in being a step above the rest.

8. Acknowledgements

I would like to thank Shreyas Ramesh for guiding me through the coding process of dataset searching and model building, alongside the creation of this research paper, this research paper could not be completed without the mentorship and guidance provided.

9. References

Jenders, Maximilian & Kasneci, Gjergji & Naumann, Felix. (2013). Analyzing and predicting viral tweets. WWW 2013 Companion - Proceedings of the 22nd International Conference on World Wide Web. 657-664. 10.1145/2487788.2488017.