Final Report: Machine Learning for evaluating the popularity of COVID-19-Related Posts on Twitter Platform

Feiyu Guo fguo35@wisc.edu
Rowan Shi zshi83@wisc.edu
Steven Yang yang558@wisc.edu
Zening Duan zening.duan@wisc.edu

Abstract

The objective of this project is to predict the diffusion engagement, or called popularity, of COVID-19 conversation-related tweets in the early phase of pandemic. We collected 2,285,379 tweets and meta data and processed them for analysis. The data are divided into 70% for the training set and 30% for the test set. We found that the data set is highly imbalanced so we implemented the SMOTE (Synthetic Minority Over-sampling Technique) and compared performance with/without SMOTE. Algorithms of KNN, Random Forest, Gradient Boosting, and Cat Boost were trained and models' performance were tested. In particular, we used the grid search for Cat Boost to improve the model. As a result, we found that the Random Forest is the best option for this task among other models we tested. The implications of our work and future directions were discussed in the end.

1. Introduction

An estimated 81 percent of Americans have a social media account (Statista, 2021). Social media is now a critical part of the way people in most walks of life communicate and a key part of how people's knowledge, attitudes, and behaviors get established - from real-time information streaming to online collective actions and beyond. Since the U.S. presidential election in 2016, the mass mobilization power of social media platforms like Twitter has raised attention and concerns from social scientists and the governments, and the public. Understanding how information diffuses and how the online climate of public opinion evolves over time are thus increasingly important for short-term intervention and long-term policy decisions.

This project aims to train a set of machine learning models to predict the engagement of COVID-19-related tweets on Twitter platforms. The matrix of tweet engagement, including the count of retweets/reply/like/quote/burst, jointly describes a particular tweet's popularity (or visibility) after a specified diffusion time window. We plan to train these models using descriptive data features parsed from raw JSON datasets requested from Twitter Research Track product API. After training, the models will output a variable depicting the estimated engagement of tweets, given inputs of tweet features such as follower/followee numbers, account history, sentiment and moral foundations in tweet content, media usage, etc. To evaluate the modes, we will designate test datasets and determine the accuracy of model outputs compared to the natural level of tweet engagement.

Our project could be utilized to support public health organizations and practitioners to potentially enhance their effectiveness and accuracy of online intervention, public opinion understanding and monitoring, misinformation correction, and science education toward the public. The findings would also deepen our understanding of online information dynamics and the nature of mass communication in digital spaces.

2. Related Work

Modeling popularity of information of different kinds, for instance, video clips on both YouTube and blogging social networks (Vallet, Berkovsky, Ardon, Mahanti, & Kafaar, 2015) [7], short videos on Tik Tok (Ling, Blackburn, Cristofaro, & Stinghini, 2021) [3], and messages on microblogging platforms (Wang, Bansal, & Frahm, 2018) [9], contributes fundamental knowledge to our understanding of online collective behavior of users.

A rich body of literature has been investigating the virality of online posts, such as tweets (Wang, Bansal, & Frahm, 2018; Wu & Shen, 2015) [9] [10]. Such a relevant line of efforts is two-fold: on the one hand, researchers proposed and evaluated a set of factors that were believed to impact tweet popularity over time. For example, Wang and colleagues (2018) examined the distinct influences of tweet semantics, embedded images, and authors' social relationships on Twitter diffusion; Tan et al. (2014) [6] studied the effects of tweet wording in affecting tweet popularity under controlling other variables; Xiao et al. (2020) [11] uncovered the role of another critical factor, post-publication time, in tweet popularity. On the other hand, existing works highlighted the divided boundary of different types of tweets and provided separate evidence on the instant and long-term engagement of tweets. For instance, a Science paper com-

pared the diffusion of true and false news posted in ten years on the Twitter platform (Vosoughi, Roy, & Aral, 2018) [8]; Other works explored the spread of news content (Wu & Shen, 2015) [10] and images (Cappallo, Mensink, & Snoek, 2015) [1] on Twitter.

Although predicting popularity evolution has primarily been explored, a multi-dimensional investigation on tweet engagement in a high-profile public health crisis like the COVID-19 pandemic remains open. In this paper, we revisit the tweet popularity prediction problem by considering all data modalities at three levels: tweet level (e.g., emotions and moral foundations), account level (e.g., account history and popularity), media-level (e.g., whether a media was embedded), and the diffusion process of tweets. Readers can find more details in the following method section.

3. Proposed Method

In this project, our group aims to use different machine learning models to give predictions of the diffusion of the original tweets about covid. The original dataset we collected contains 2,285,379 tweets and due to the overwhelming number of the tweets, we decided to use 10% of the data to train our models. The diffusion of the tweet is calculated using the increasing number of retweets, likes, replies, and quotes of the tweet in a certain time frame. The independent variables in the dataset consist of information from four levels: Account level, Tweet level, Media level, and Geolocation level and there are 34 independent variables in total. Because only 1% of the tweets have the Geolocation data, we decided to exclude all Geolocation variables. Fortunately, there is no missing data in other independent variables and dependent variables so the imputing method is not needed. The details of the dataset will be discussed in later sections.

Since the considerable amount of tweets we are using in this dataset, we decided to use the hold out method to evaluate the models. We used the "train test split" method from the ski-learn package to create a train set and a test set. To change our dependent continuous variables into categorical variables, we set a threshold of upper 95th quantile to split data into two levels of diffusion: Low level and High Level. The four dependent variables: retweets, likes, replies and quotes all follow this threshold into categorical variables for the convenience of model training and accuracy calculation. The reason why we use accuracy as the standard instead of mean square errors is because the accuracy is easier to understand whether the model is well fitted or not.Setting the 95th quantile to split data is because more than 75% responses are 0 among all four dependent variables, especially in the response of retweet and quote that about 95% responses are 0. In order to have an effective split in response variables, we set the 95th quantile as a metric. After changing those four continuous variables into categorical variables, we created a new response variable called General-POP (General Popularity) as our true response variable for model fitting. The metric of the General-POP variable is that if one of the original response variables is classified as high level, then we classify the General-POP as high level. The 95th quantile splitting also brings the problem of the imbalance of the dataset, in which about 95% data are labeled as low level and only about 5% data are labeled as high level. In order to solve the problem of the imbalance, we implement the SMOTE (Synthetic Minority Oversampling Technique) to over sample the minority group, the high level [2]. The detail of the data prepossessing of features will be discussed later.

3.1. SMOTE

The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. For instance, if the amount of over-sampling needed is 200%, only two neighbors from the five nearest neighbors are chosen and one sample is generated in the direction of each. Synthetic samples are generated in the following way: Take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the minority class to become more general. The synthetic examples cause the classifier to create larger and less specific decision regions. More general regions are now learned for the minority class samples rather than those being subsumed by the majority class samples around them. The effect is that decision trees generalize better.

3.2. KNN

The first model we presented here is k Nearest Neighbors. The k Nearest Neighbors (KNN) is a classification algorithm that the processing of the training examples is postponed until making predictions. To make a prediction, kNN model finds the k nearest neighbors of a query point and computes the class label with majority vote on the k nearest points. There are many distance measures that can be considered when looking for nearest points. For our project, we will only consider the Euclidean distance.

3.3. Random Forests

The second model we presented here is Random Forest. Random forest is an ensemble method built on decision trees. In random forests, we fit decision trees on different

bootstrap samples. While each decision tree gets the whole set of features, only a subset of features will be selected randomly for each note. The "optimal" number of features at each node is considered to be log2 m + 1, where m=total number of features. Besides using random forests to fit a model predicting the diffusion of twitters, we will also run "feature importance" via random forests to see how important each feature is for the overall prediction.

3.4. Gradient Boosting

The third algorithm we used is gradient boosting. Gradient boosting is one of the many boosting methods. Gradient boosting is very similar to AdaBoost, where they both successively train weak learners to create a strong ensemble. In other words, it trains decision tree stumps based on errors of the previous decision tree stump. Gradient boosting is fitting decision trees in an iterative fashion using prediction errors. It does not use the prediction errors for assigning sample weights, like AdaBoos doest; they are used directly to form the target variable for fitting the next tree.

3.5. CatBoost

The last model we presented here is CatBoost. CatBoost is an algorithm based on gradient boosted decision trees. During training, a set of decision trees is built consecutively. Each successive tree is built with reduced loss compared to the previous trees. We can control the learning rate and max-depth of each tree. CatBoost takes very less prediction time compared to other boosting algorithms and it can handle categorical features very well. Based on the good performance of CatBoost from the previous assignment and the fact that we have one categorical feature "account verified", we decide to run grid search on it to see the best accuracy we can get.

In short, we will be using KNN classifier, Random Forest, Gradient Boosting, and CatBoost with grid search cross validation process during model training, both without and with SMOTE. We split the data into a training set and a test set, each taking 70% and 30% of the whole dataset, respectively. We will also evaluate the feature importance via Random Forests to see what features are important regarding the prediction of diffusion of twitters, the General-POP label.

4. Experiments

We tested the KNN and Random Forest baseline models to evaluate those model's abilities on making accurate predictions of a tweet's diffusion. During our efforts in reaching higher accuracy, we realized the severe imbalance in our dataset. To achieve better balance, we used SMOTE to oversample the "high" class and retested the KNN, Random Forest, gradiant boosting, and Cat Boost with grid search models. With other metrics and confusion matrices, which will be covered later, we compared the performance between different classifiers and evaluate the improvement before and after SMOTE operation.

We are also interested in which features are the most important and least important to the model when making a prediction. Therefore, we inspected the importance score and gained insight into the random tree model with our tweet diffusion sample dataset.

4.1. Dataset

We collected English tweets related to the COVID-19 pandemic posted by global users from 15 June 2020 to 12 July 2020 (4 weeks) and metadata of user-level attributes. Twitter's COVID-19 Firehose stream endpoint (`https://developer.twitter.com/en/docs/labs/covid19-stream/overview`) was used to fetch data. As claimed by Twitter, this endpoint is a real-time stream of 100% public tweets that deliver full conversation about COVID-19, rather than just a sample. We designed a Python script to parse raw data from JSON to tabular format (i.e., CSV). To make our data scope more focused, we applied 25 keywords (see Table 1 in Appendix) to further filter our datasets. These keywords were believed to capture essential issues that happened in the four-week period. This process resulted in 41,077,714 tweets in the entire four-week period and 2,285,379 unique original tweets in the first two weeks. We took a random 10% sample of the total 2,285,379 unique original tweets for this project. The unit of analysis is each unique original tweet in the sample dataset. To study the diffusion of tweets, we investigated all original tweets generated in the first two weeks in June (i.e., 00:00 June 15th to 23:59 June 28th, UTC). We let each original tweet diffuse for another two weeks and monitored changes in their accumulative social engagement metrics. It is plausible to think that most messages on Twitter fully diffuse within a maturation period of two weeks since being posted (Vargo, 2014), with that said, a 14-day tracking allows us to capture the overall picture of tweet diffusion (Meng et al., 2018).

4.2. Data Pre-processing

This study examines multi-layered attributes of tweets that may affect the diffusion of messages on Twitter at different levels. We considered four communicative actions as dependent variables (DVs): liking, sharing, quoting, and commenting. For independent variables (IVs), we emphasized multidimensional moral foundations and emotions embedded in text along with a broad spectrum of other factors ranging from account-level metadata, content-level features, and media-level variances. Details are clarified below and see Table 2 in Appendix for descriptive statistics of variables.

To measure the four communicative actions, we retrieved the total count of likes an original tweet receives and took it as the indicator of liking; similarly, the total count of retweets represents sharing, the total count of quotes as quoting, and the count of replies represents commenting. Since all four communicative actions are accumulative by nature, we took the last record captured in the two-week diffusion period as the final static status of an original tweet.

To uncover the explanatory factors behind the communicative action dynamics, we first measured morally relevant information of tweets. We adopted the extended Moral Foundations Dictionary (eMFD) to extract five moral foundations from tweet content (Hopp et al., 2020). Each word in the text is assigned a vector of five values representing the five moral foundations, which are: Authority, Care, Fairness, Loyalty, and Sanctity. Each of these values, ranging from 0 to 1, denotes the probability that a particular word was annotated with a particular moral foundation. For some high-ranked sample tweets under each moral foundation, see the Appendix.

Second, we applied NRCLex 3.0.0, a Python library developed based on the NRC Word-Emotion affect lexicon (Mohammad & Turney, 2013), to recognize emotions of a given Tweet. Argued by many to be the basic and prototypical emotions (Plutchik, 1980), the following eight were predicted: fear, anger, trust, surprise, sadness, disgust, joy, and anticipation. We obtained the frequencies of each emotion within tweets, ranging from 0 to 1.

Last, we measured other factors at different levels and controlled them in baseline models. They are five metadata at the account level which descript basic information of an account: count of followers, count of friends, count of status, account history (by day), and account verification status (when true, indicates a verified account), respectively; and three variables at the tweet level which describe content elements: count of hashtag,count of URL, and count of user mentioned; and four variables at the media level which present the media usage in a tweet: Tweet has photo,tweet has video,tweet has gif, andtweet has media(when true, indicates that the tweet has certain type of media embedded). For all the control variables, we took the value as the first record exists in our dataset. We removed 5 data points due to errors in their account history.

4.3. Software and Hardware

The models implemented in this project were performed in the jupyter notebook running Python 3.8. The primary python libraries we used include numpy, pandas, matplotlib for general plotting and scikit-learn and mlxtend for machine learning models.

Each group member uses his/her own laptop.

5. Results and Discussion

Before we perform any machine learning algorithms to train the model, we firstly notice that only around 8% of data in General-POP are labeled as high and about 92% of the data are labeled as High. This phenomenon determines that the accuracy of most algorithms would be greater than 92% because without using any algorithms, we could easily get 92% accuracy by predicting all tweets have a low level of diffusion.

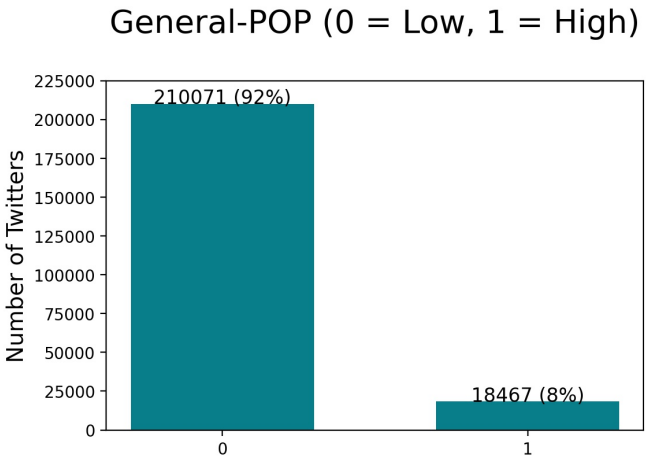

Figure 1: General-POP Distribution

5.1. KNN and Random Forest without SMOTE

We fitted a KNN model to the training set with k = 3. The model obtained an accuracy of 91.59%, which is even lower than the 92% and means that the KNN model with k = 3 poorly predicts the popularity of a tweet. We also fitted a Random forest model with default hyperparameter and n_estimators = 100. The model obtained the accuracy of 92.92%, which is only slightly higher than the 92%.

Looking at both accuracy of KNN and Random Forests generated, it is suggested that the accuracy can be a very bad metric that can be used to evaluate our models. The 92% accuracy cannot represent the true accuracy of our models. Even though 92% can be a high accuracy, our models are highly likely to perform poorly in reality. We need to use other evaluation metrics to further determine how our models actually perform. We then generated the confusion matrix and tried to evaluate the models based on performance metrics such as Precision, Recall, F1 Score, and MCC (Matthews Correlation Coefficient).

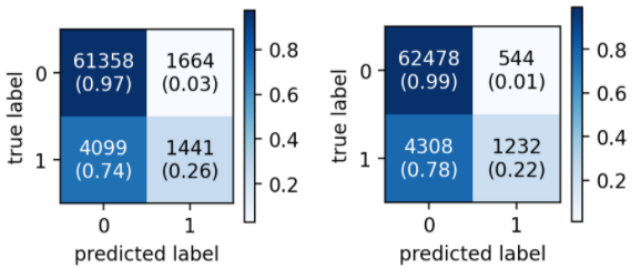

Figure 2: Confusion Matrix of KNN(Left) and Random Forest(Right) without SMOTE

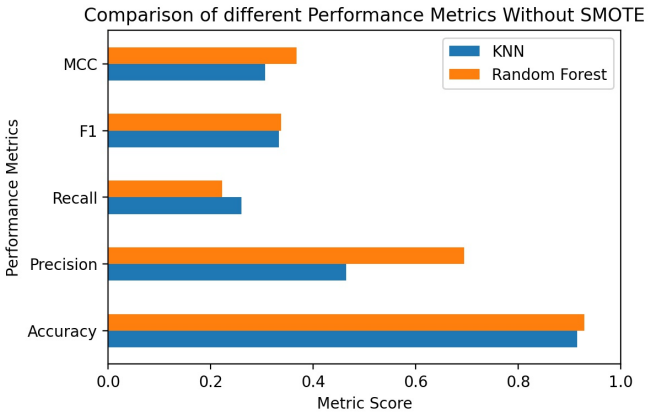

Figure 3: Comparison of Metrics of KNN and Random Forest without SMOTE

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. In other words, the question that this metric answers is of all Twitters that are predicted as popular, how many are actually popular? Higher precision relates to the lower false positive rate. From the figure, we can see that Random Forest is more precise than the KNN model. Thus, Random Forest does a better job at correctly predicting the true positive label.

Recall is the ratio of correctly predicted positive observations to the total actual positives. Hence, recall is actually just another term for True Positive Rate. It tells us how many of the actual positives we captured. This metric is the lowest among all metrics for both algorithms. This means that both KNN and Random Forest models are falsely predicting the popular Twitters as unpopular to a great extent. This is also due to the fact that our data is highly unbalanced where the majority of the label is "unpopular". From this metric, we can see that the 92% accuracy can be misleading and cannot be trusted.

F1 Score is the weighted average of Precision and Recall. This score takes both false positives and false negatives into account, which is a balance between Precision and Recall. F1 Score is more useful than accuracy when the data has an uneven distribution. Both algorithms have a F1 Score around 0.3. The score is relatively low so our models are not actually performing well.

MCC (Matthews correlation coefficient) is another metric that is considered especially helpful in unbalanced class settings. As discussed previously, neither F1, precision, or recall takes true negatives into account. MCC is a measure that takes all elements of a confusion matrix into account and it does suffer from class imbalance like accuracy. This measurement is ranged between -1 and 1, where -1 means total misclassification, 0 means random prediction, and 1 means perfect classification. Since in our predictions, the number of true negatives are relatively huge, this metric results in a higher score than Recall after taking this into account. However, an MCC of +0.3 is still relatively low and it indicates that our models are performing poorly.

5.2. KNN and Random Forest with SMOTE

After implementing SMOTE, we refitted the KNN and Random forest model. The KNN model with k = 3 obtains the accuracy of 91.63%. The KNN model still performs poorly in our dataset. It is partly because we did not do any hyperparameter tuning for the KNN model and partly because the KNN model probably is just not suitable for our dataset. Although it is even lower than before, the accuracy is much more reliable. By looking at the confusion matrix, the number of the true positives increases and the number of false negatives decreases.

Surprisingly, the random forest model with default setting and n_estimators = 100 has excellent accuracy of 99.99%. Through the confusion matrix, we could also find the problem of having low accuracy of predicting label 1 has been solved. More importantly, the number of false positives and false negatives is extremely low. This means that the Random Forest model performs really well and this accuracy is trustworthy.

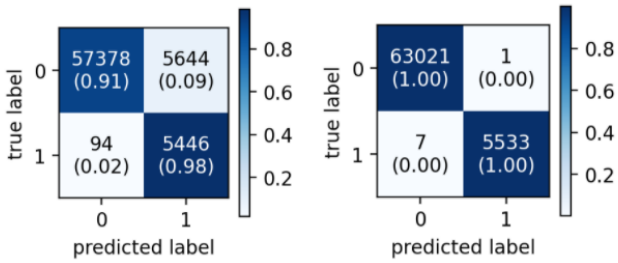

Figure 4: Confusion Matrix of KNN(Left) and Random Forest(Right) with SMOTE

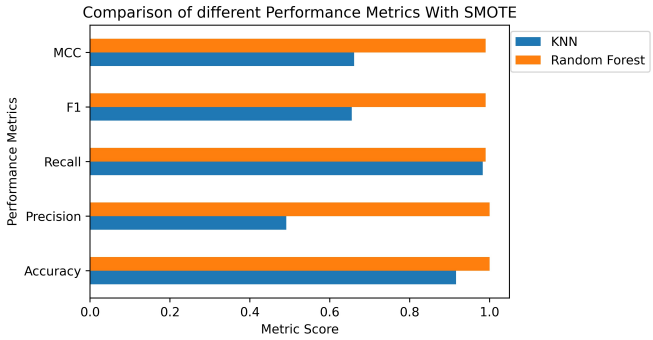

Figure 5: Comparison of Metrics of KNN and Random Forest with SMOTE

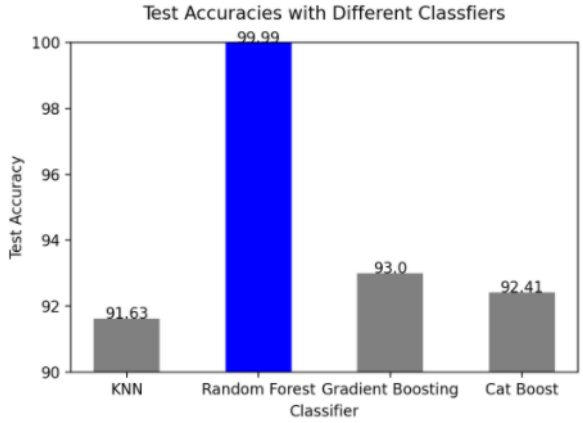

Figure 6: Final Model Accuracy Comparison

By looking at the Recall, F1, and MCC scores, we can see that they are greatly improved for both KNN and Random Forest after SMOTE. This also indicates that our models are classifying the labels more accurately after SMOTE. However, for the KNN model, the precision is not improved much (from 0.464 to 0.491). To have a more direct look, we can check the confusion matrix. As we can see, although the number of true positives increases, the number of false positives also increases to a great extent. In comparison, our Random Forests model has lower false positives while still having a higher number of true positives.

5.3. Gradient Boosting and CatBoost using SMOTE

After SMOTE, to observe the performance of other models and to prove that the excellent predictive ability of Random Forest is not universal. We also refitted the Gradient Boosting and Cat Boosting with Grid Search using SMOTE operation. The Gradient Boosting model was fitted with parameters (learning_rate=0.1, n_estimators=100, max_depth=8) and resulted in accuracy of 93.00%. Such an accuracy can be satisfying but it does perform no better than the Random Forest model. Next, we tried Cat Boosting with Grid Search, a hyperparameter tuning method, and set parameters as learning_rate': [0.01, 0.05], 'max_depth': [5, 8], 'n_estimators': [100, 200, 500]. The best accuracy score we achieved then was 92.41%, which is even worse. Through comparison, both Gradient Boosting and Cat Boosting after hyperparameter tuning are nowhere near as accurate as the previous random forest model.

5.4. Feature Importance with Random Forest

We believe the more accurate our model is, the more we can trust its interpretation of the feature importance. Since random forest achieved the highest accuracy among 4 models after SMOTE, we chose to gain insight from the General-POP importance score only from the random forest model.

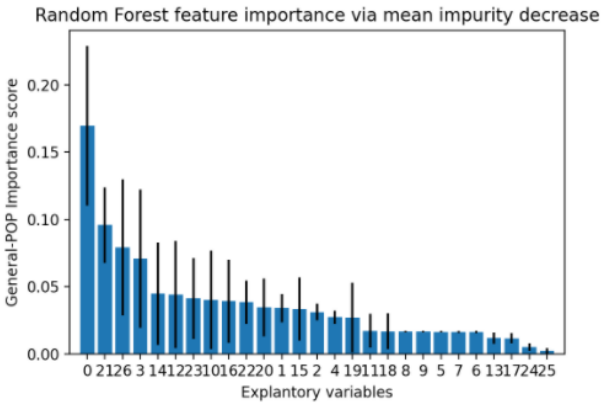

Figure 7: Random Forest Feature Importance Score

The Top five important features in random forest importance score are account_followers_count, url_count, seedmedia_hasmedia, account_verified, and positive. The most important feature is account_followers_count which represents the number of followers of whom post the tweet. It is reasonable to have this result because if one has millions of followers or more, his or her tweet generally would have relatively broad influence in the society compared to those who only have hundreds of followers or even less. The second most important feature is url_count, the number of urls in a tweet. The reason why the number of urls matter is probably because there are some tweets collecting

the news or information of the covid-19, which have far-ranging influence on the society. However, in the Pancer's research, he showed that if a tweet containing more url, user mentioned, and hashtag would decrease the probability to be widely spread because url, user mentioned, and hashtag would make the text inconvenient to read and understand [4]. More interestingly, according to Suh's paper, the number of urls and hashtags would increase the possibility of being retweeted, which is corresponding to our study [5]. Further study is needed to find the true relationship between the number of urls and diffusion of a tweet. The third most important feature, seedmedia_hasmedia might have the same reason as url_count that the presentation of information about covid-19 might receive strong attention from people. The fourth important feature is account_verified, representing whether the account has been verified by Twitter or not, and if the account has been verified, there is a high probability that the owner of the account is a public figure and generally they would have wide influence and considerable number of followers. The fifth most important feature, positive, represents the score of positiveness of a tweet ranging from 0 to 1. The closer to 1, the more positive a tweet is. The positive words are important for supporting people to get through the hard time in this pandemic and the tweet containing this positive sentiment is more likely to be seen by others.

6. Conclusions

Social media platforms like Twitter have become major channels for people to disseminate news, discuss politics, and engage in collective action. Every day, some information on Twitter goes viral while others stay relatively silent. Uncovering the mechanism of tweet popularity would help researchers and industry practitioners better understand how social media information naturally diffuses over time and the underlying principle to win the game of online attention attraction.

Align with previous analyses performed for forecasting tweet popularity based on various factors, we built and categorized factors at three levels: content level like tweet emotions, account level like account history, and media level like whether a tweet contains media component, we hope such a comprehensive investigation could contribute to a relevant line of scholarly efforts. We found that the top five significant features are: 1) Count of embedded URLs, 2) Whether a tweet contains media, 3) Account history, 4) Count of account followers, and 5) Positive emotion. Our fitted machine learning models have proved this finding.

Our report illustrates a machine learning approach of factor evaluation in predicting tweet popularity. Analyses in this report present model comparisons across four algorithms: KNN, Random Forest, Gradient Boosting, and Cat Boost, which were applied on the same COVID-19 dataset we collected and pre-processed. We adopted the SMOTE method to fix the unbalanced issue in our classification tasks. We evaluated model performance with clearly reporting accuracy, recall, and precision of models. Besides, we also plotted the Confusion Matrix to help us better evaluate and understand the fitted models.

Given the modeling results, we conclude that the Random Forest model performs the best in tweet popularity prediction compared to other models, with an overall accuracy of 99.99%, recall, a F1 score of 99.99% and a MCC score of 99.99%. This finding is consistent with previous study (Joseph, Sultan, Kar, & Ilavarasan, 2018) which found learning mechanisms like random forest are better than other models like decision tree.

Our findings provide novel insights into the highlighted features of COVID-19 tweets in contributing to the information engagement and tweet popularity in the early phase of the pandemic. We encourage future research to put forward more rigorous analyses. We hope our efforts could support scientists, governments, public health departments, and campaign practitioners to understand the spreading of online information better, build better social media tools, and help the public access high-quality content at a suitable time.

7. Acknowledgements

We would like to gratefully appreciate the guidance and teaching from Dr. Sebastian Raschka.

We thank the Computational Approaches and Message Effects research group for data sharing. Raw Datasets would not be used for other purposes beyond this in-class practice project unless with permissions.

8. Contributions

Zening Duan took samples from the original dataset from CAMER research group at Journalism School and cleaned up the data for further analysis. Feiyu Guo combined four dependent variables into one new response variable. Steven Yang came up with the idea of solving data imbalance with SMOTE. Feiyu and Steven trained the models, including KNN, Random Forest, Gradient Boosting and Cat Boosting with grid search, and put SMOTE into practice. Feiyu and Steven generated confusion matrices and other related graphs.

As for the report, Zening drafted Abstract, Introduction and Related Work. Feiyu and Steven Yang drafted the Proposed Methods. Rowan Shi and Feiyu wrote part of Experiments as well as the Results & Discussion section. Steven finished the rest of the Experiments and Results & Discussion section. The whole team worked on Conclusions and the overall structure of the report including paper style and reference.

9. Code and Appendix

The code notebooks that are implemented for this project and also the appendix document can be accessed at the following GitHub repository: `https://github.com/stevenYang914/Stat451_Twitter_Popularity`.

References

[1] S. Cappallo, T. Mensink, and C. G. Snoek. Latent factors of visual popularity prediction. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 195–202, 2015.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[3] C. Ling, J. Blackburn, E. De Cristofaro, and G. Stringhini. Slapping cats, bopping heads, and oreo shakes: Understanding indicators of virality in tiktok short videos. *arXiv preprint arXiv:2111.02452*, 2021.

[4] E. Pancer and M. Poole. The popularity and virality of political social media: hashtags, mentions, and links predict likes and retweets of 2016 u.s. presidential nominees' tweets. *Social Influence*, 11(4):259–270, 2016.

[5] B. Suh, L. Hong, P. Pirolli, and E. H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *2010 IEEE Second International Conference on Social Computing*, pages 177–184, 2010.

[6] C. Tan, L. Lee, and B. Pang. The effect of wording on message propagation: Topic-and author-controlled natural experiments on twitter. *arXiv preprint arXiv:1405.1438*, 2014.

[7] D. Vallet, S. Berkovsky, S. Ardon, A. Mahanti, and M. A. Kafaar. Characterizing and predicting viral-and-popular video content. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1591–1600, 2015.

[8] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[9] K. Wang, M. Bansal, and J.-M. Frahm. Retweet wars: Tweet popularity prediction via dynamic multimodal regression. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1842–1851, 2018.

[10] B. Wu and H. Shen. Analyzing and predicting news popularity on twitter. *International Journal of Information Management*, 35(6):702–711, 2015.

[11] C. Xiao, C. Liu, Y. Ma, Z. Li, and X. Luo. Time sensitivity-based popularity prediction for online promotion on twitter. *Information Sciences*, 525:82–92, 2020.