Classifying Conspiracy Posts during the 2020 Election

Cason Konzer
casonk@umich.edu

Ayuk Tambe
atambe@umich.edu

Jack Anderson
jackaa@umich.edu

Chris Kalo
kaloc@umich.edu

***Abstract*—The increase of users on the Internet has led to an increase of online communities. With this increase of online communities, we have an abundance of data that can be inspected for various reasons. The goal of this paper is to examine the performance of different Machine Learning models on classifying published posts. Reddit is a news aggregation, web content rating, and discussion website. We will determine if a post was published before or after the 2020 Presidential Election.**

I. Introduction

Reddit is a social news website and forum where content is socially curated and promoted by site members through voting. It is made up of millions of collective niche forums or groups called Subreddits. The 2020 presidential election led to an increase of activity on the internet. We will use Machine Learning models to classify published posts on subreddits. We will classify if each post was published before or after the 2020 presidential election, using the class labels "Before" and "After". We will start by preparing our data and then extracting key features, followed by the use our data on different data mining algorithms, concluded by an evaluation of each models performance.

II. Materials and Methods:

A. Data Preprocessing

The first step in our preprocessing of our data was attribute reduction. We got rid of several columns that contained arbitrary or repetitive information. For example, a subreddit column contained the same value for every entry, as all the data came from the same subreddit, r/conspiracy. In addition to this, we modified several attributes that had only two possible values (T/F, submission/comment, etc.) and changed them to binary values of 0 and 1. This made them much easier to use in any model that we desire. For the same reason, we broke down an attribute representing what kind of media was contained in the post (image, link, video, or self) into 4 separate binary columns that represent whether or not a post is of that certain type. These different types of media are seen in Figure 1 with the scores that correspond to them.

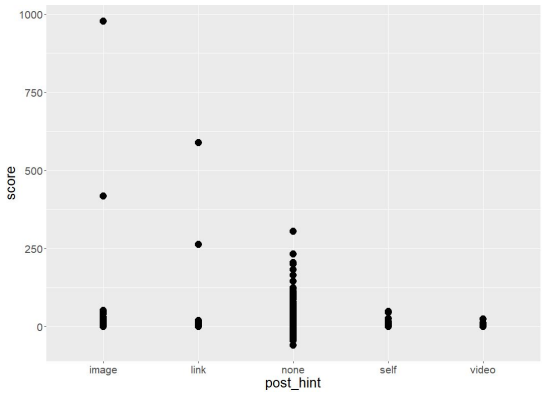


Figure 1

Furthermore, we added in columns that corresponded to the number of words in the title and body of the post respectively. This is a useful metric for many models and will help with classifying posts before or after the 2020 Presidential Election. Also added in were different centrality measures that represent how interconnected an author is in the conspiracy subreddit network. These included betweenness, closeness, and degree centrality. Similarly, we added in a pagerank which is based on the in-degree and out-degree of each author, which in this case is how many times an author is posted on or posts on other users respectively. All network features were implemented via the networkx python module [2]. We also made our classifier column based on when the different posts were posted, before or after the election. This allowed us to fully remove the data and time columns which makes sure the models are not dealing with information that could give them the classification directly.

The highest number of posts came in the days after the election was decided (11/07) as this is when the most amount of conspiracies were being generated after Biden was declared the winner of the election by most major media outlets [12]. Following this day the number of posts per day dropped off fairly consistently until the end of our data (11/15). The only major problem with the data is there is a significantly smaller amount of posts recorded on one of the days after the election (11/03) which is likely due to limited resources on the API we used. There were potentially other subreddits that had a higher priority over the r/conspiracy subreddit due to the time frame.

B. Method

As our problem is a classification problem, there are various well known models published of which we can leverage on the task. The nature of this problem is of the simplest type of classification, a binary classification, before or after the 2020 election. As the data itself is large, and includes a variety of aspects about the posts, it was unclear what model would be the best choice. For our method of approach we chose to thus select a variety of models, and then take those with the best performance. We implemented all models within scikit-learn's python module [6]. The models we chose to evaluate spanned a broad range of approaches and included the following: Naive Bayes, K-Nearest-Neighbors (KNN), Decision Tree, Random Forest, Ada Boost, Gradient Boost, Bagging, and Support Vector Machine (SVM).

For each model we utilized a consistent random state, and defaulted to a .75/.25 train/test split. We fed the same dataset across all models, for both training and testing, as to have directly comparable results. In the case of exceptional run time, solvers were placed under a hard iteration limit before convergence, of which only SVM utilized with 10,000 iterations as the cap.

C. Model Optimization

For each model we varied a traditional input and used the best parameterized model then as a benchmark for cross model comparison. The inputs optimized upon are as follows ...

Optimized Model Inputs

Model	Parameter Description
Naive Bayes	Laplace smoothing
Decision Tree	Maximum depth of tree
Random Forest	# of trees in the forest
Bagging	Number of base estimators
Ada Boost	Max # of estimators for boosting
Gradient Boost	# of boosting stages to perform
SVM	C regularization parameter
KNN	# of neighbors

Table 1

While each model has additional inputs, we chose to optimize these due to dependencies and run time availability. Other approaches were implemented without success, these included feature reduction and using an unsupervised algorithm to first cluster the posts then using the unique cluster id as an additional feature for the supervised model. For the decision tree, bagging, and gradient boost model we see a clear trend of increased performance by increasing the maximum depth, the size of the ensemble, and the number of boosting stages as show in figure 2, 3 and 4.

Unique to the decision tree approach, is an apparent turning point such that relative area under ROC curve falls off. Due to this phenomena we can see that maximization of AUC occurs before that of F1, and thus our optimized model falls at the intersection where max depth is 27, slightly below that of convergence. Both gradient boost and ada bosst approaches failed to reach their optimal values due to run time limitations. It is evident that the number of boosting stages for the gradient boost model was near convergent, but required an increase to the range of the parameter space. While not displayed in the figures, the ada boost model followed an identical trend. SVM models performed optimally with C=1, the lowest tested parameter. Due to unacceptable results with a linear kernel, SVM was additionally searched using a radial basis function kernel, yet optimal performance decreased. The Naive Bayes model showed no change with and without Laplace smoothing.

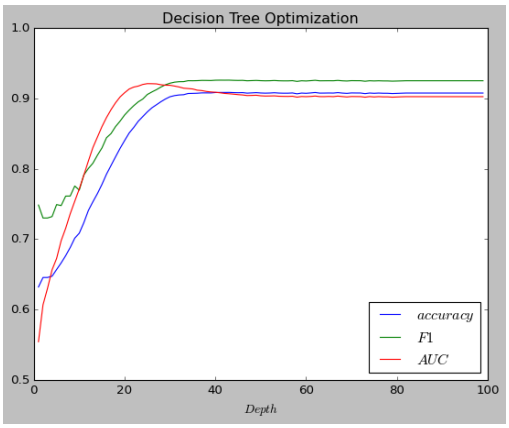


Figure 2

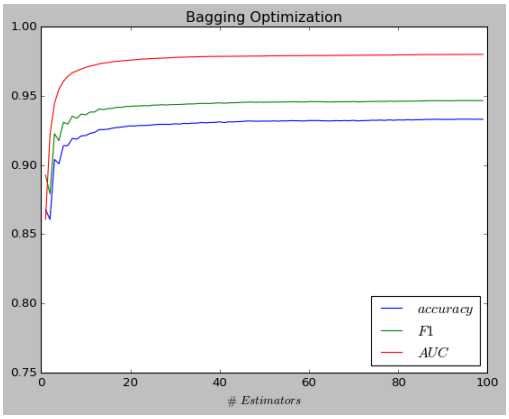


Figure 3

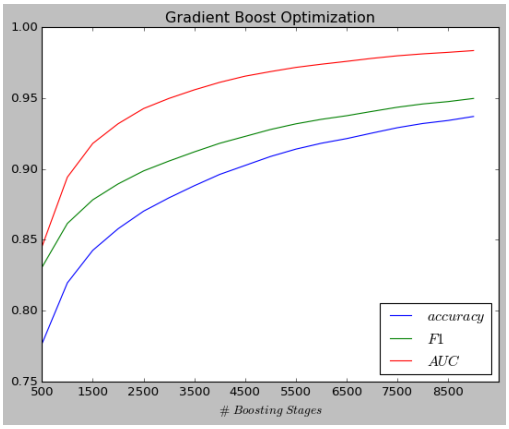


Figure 4

For feature reduction attempts first feature importance was ranked then the lowest ranking 14 features were dropped. The unsupervised learning model used to generate cluster ids was K-means cluster of which was searched on # of clusters. While these optimization techniques did not improve or maximum performance we did see an equalization between F1 and AUC scores such that while slightly lower than maximums, their trade-off was minimized.

D. Evaluation

In order to evaluate the classification of the data based on the models implemented, we must measure the degree to which the predicted classification matches the actual classification of each post. There are numerous different measures used to assess the efficacy of the classification models that were used on the data. These measures include accuracy, precision, recall, F1-score, and Area Under the Curve (AUC – based on ROC curve). We prioritized the metrics from the F1-score and Area Under the Curve as most important.

The goal of these metrics will reveal the rate of faults in the classification models implemented. In the *accuracy* score, we measure this by taking the number of correct classifications over the total cases (1). In the *precision* score (positive predictions), it is measured by taking the actual positives predicted as such, over the number of any case predicted as positive (2). In the *recall* metric (also called *sensitivity*, the calculation is based on the correctly predicted positives out of all actual positives (3).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

The main two metrics we will use to evaluate model performance are the F1-score and the AUC. The *F1-score* is calculated using the precision and recall (5). This represents the harmonic mean of these two metrics. The Area Under the Curve (AUC) is the value that represents the area under the Receiver Operating Characteristic (ROC) curve. The higher the values (ranging from 0 to 1 in decimal), the closer to perfect the ROC curve is calculated. The ROC curve measures the correlation between sensitivity (or recall) and specificity. Specificity is the ratio of correctly predicted negatives out of all actual negatives (4). Along with the graph of the ROC curve, we can look at the Precision-Recall Curve (PRC) graph, which is linked towards the F1-score. While the ROC curve should curve near the top-left corner of the graph, the PRC should curve near the top-right corner. The F1-score is a useful metric for understanding how well the positive class was classified, while the AUC is a more general score which incorporates the specificity (negative class) as well. Thus, the AUC could be seen as more of a useful metric in cases where the dataset is not heavily imbalanced towards one class. In our case, the dataset is not "heavily" imbalanced to one class (leans towards *after* around 2:1), so we will look at the AUC as likely the best metric, along with the F1-score as well.

$$Specificity = \frac{TN}{TN+FP} \quad (4)$$

$$F1-score = 2 \times \left(\frac{Prec. \times Rec.}{Prec. + Rec.}\right) \quad (5)$$

III. Results

After implementing the models on the dataset, we recorded the score metrics as stated in the prior section – *Evaluation*, in order to evaluate the performance of the classification models used. Higher scores would point to a more efficacious model. For instance with the F1-score, a higher score means the Positive class has been classified more effectively. With the AUC or ROC, more Area Under the Curve means higher True Positive Rate and lower False Positive Rate.

Metrics of the Classification Models

Model	Accuracy	F1-score	AUC
Naive-Bayes	0.644694	0.734468	0.659127
K-N Neighbors	0.667160	0.744938	0.692231
Decision Tree	0.890821	0.912262	0.920538
Random Forest	0.841079	0.875591	0.910908
ADA Boost	0.770116	0.824907	0.839598
Gradient Boost	0.936975	0.949672	0.983430
Bagging	0.932977	0.946488	0.979873
SVM	0.606678	0.749098	0.540591

Table 2

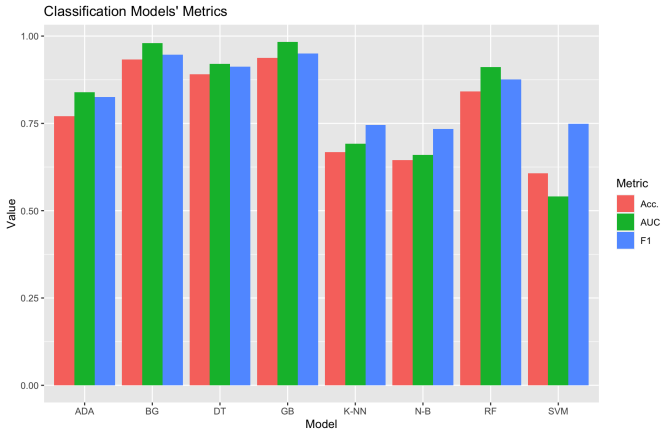
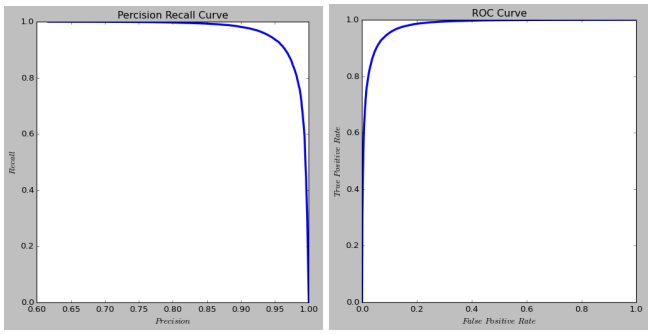


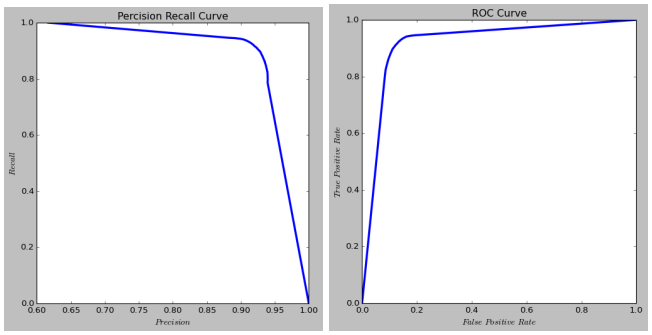
Figure 5

The graphs representing the PRC (Precision ∼ Recall) and ROC (True Positive Rate ∼ False Positive Rate; also called Sensitivity/Recall ∼ Specificity) curves were also plotted for the top 3 performing models – Gradient Boost, Bagging, and Decision Tree. The Bagging and Gradient Boost models that were implemented also use decision trees.

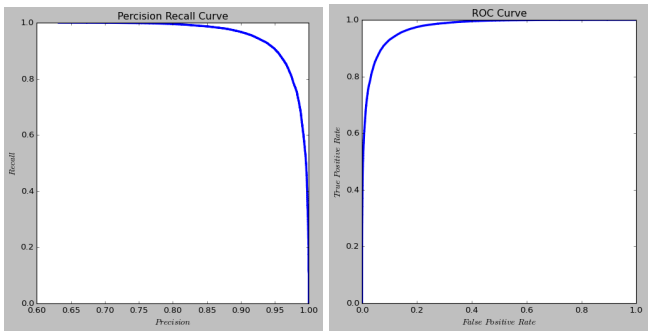
PRC & ROC Graphs



Bagging



Decision Tree



Gradient Boost

Figure 6

IV. Conclusions

After evaluation of the results, we can conclude that the top performers out of the models that were used on the dataset (in order) are – Gradient Boost, Bagging, Decision Tree, and Random Forest. These models yielded the highest AUC and F1-scores out of the models tested. All of their AUC scores were above 0.9, and considering the decimal range of the metric is 0 to 1, this would indicate that the True Positive Rate triumphed strongly over the False Positive Rate (since there was so much area under the ROC curve). Also, looking at the results of the top three models, we see that their PRC and ROC curves are quite close to perfect. When the AUC and F1-scores are high, such curves can be expected in these graphs. The common denominator in these four models is that they all were implemented over one or multiple Decision Trees. In our case of classifying whether the posts were made before or after the 2020 election, the features used had good information gain, so the Decision Trees worked out quite well.

V. Future Work

While we have demonstrated substantial predictive power for the given toy problem, a much more in depth analysis is possible. As a first step to a more robust generalization, the same models may be applied to a dataset spanning multiple subreddits. By considering those other than r/conspiracy we could show that the characteristics of the data learned by the models extend to the reddit platform as a whole. In a similar fashion, we may extend our time horizon to additional weeks, or months, before and after the election. If results were to hold, the influence of the single event as a pivot point in the platform's content would carry more weight. Final extensions to the work may include additional model evaluations and more extensive input parameter searches, especially for those models taking multiple unique inputs. We leave this work for future advocates and students of data mining.

References

[1] Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., Blackburn, J. (2020). The Pushshift Reddit Dataset. Proceedings of the International AAAI Conference on Web and Social Media, 14(1), 830-839.

[2] Hagberg, A., Swart, P., & S Chult, D. (2008). Exploring network structure, dynamics, and function using NetworkX.

[3] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95.

[4] Martin, Trevor. "Interactive Map of Reddit and Subreddit Similarity Calculator." Shorttails.io, Short Tails, 28 Nov. 2016, https://www.shorttails.io/interactive-map-of-reddit-and-subreddit-similarity-calculator/.

[5] McKinney, W., & others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).

[6] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

[7] Samory, M., Kesiz Abnousi, V., & Mitra, T. (2020). Characterizing the Social Media News Sphere through User Co-Sharing Practices. Proceedings of the International AAAI Conference on Web and Social Media, 14(1), 602-613.

[8] Samory, M., & Mitra, T. (2018). Conspiracies Online: User Discussions in a Conspiracy Community Following Dramatic Events. Proceedings of the International AAAI Conference on Web and Social Media, 12(1).

[9] Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.

[10] Novaković, J. D., Veljović, A., Ilić, S. S., Papić, Željko, Milica, T. (2017). Evaluation of Classification Models in Machine Learning. Theory and Applications of Mathematics Computer Science, 7(1), Pages: 39 -. Retrieved from https://uav.ro/applications/se/journal/index.php/TAMCS/article/view/158

[11] M, Hossin, and Sulaiman M.N. "A Review on Evaluation Metrics for Data Classification Evaluations." International Journal of Data Mining amp; Knowledge Management Process, vol. 5, no. 2, 2015, pp. 01–11., https://doi.org/10.5121/ijdkp.2015.5201.

[12] Detrow, S. D., Khalid, A. K. (2020, November 7). Biden Wins Presidency, According To AP, Edging Trump In Turbulent Race. NPR. Retrieved April 23, 2022, from https://www.npr.org/2020/11/07/928803493/biden-wins-presidency-according-to-ap-edging-trump-in-turbulent-race