Classifying Trolls on Twitter

Gene Tanaka
Department of Computer Science
Stanford University
gtanaka@stanford.edu

Tyler Consigny
Department of Symbolic Systems
Stanford University
tconsign@stanford.edu

1 Motivation

While the original purpose of social media was to connect the world, recent incidents have shown that if it is not regulated properly it can be used to divide it. Prominently, the Facebook-Cambridge Analytica scandal revealed how several political campaigns (including the 2016 US presidential election and Brexit) had been influenced by targeted social media posts and advertisements. Particularly, the influence came from politically charged, and often fake ("troll") posts. This is a pressing issue that has the potential to have serious repercussions to the integrity of democracy worldwide. This issue is especially critical to address now, with the 2020 presidential election on the horizon. Thus, we built a classifier to distinguish between real and troll tweets.

2 Process & Approach

Our tweet classifier went through several iterations. We began by using a binary classifier with logistic regression as a baseline model. We followed this baseline with a Naive Bayes model. While this iteration improved the performance, we advanced further by utilizing a stacked model that involved the basic models plus a neural network. The general framework is shown below (Figure A).

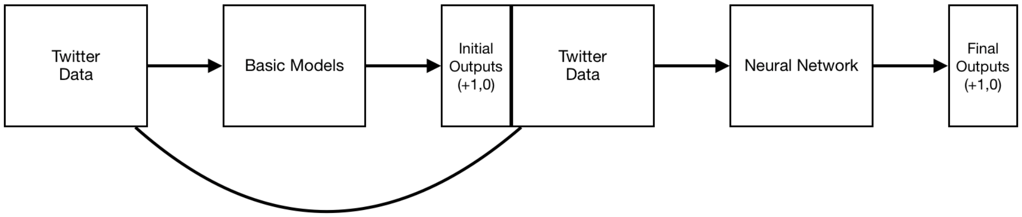


Figure A: Pipeline

The following paper will explain the development and results of this process. We will focus most heavily on the stacked neural network model.

3 Literature Review

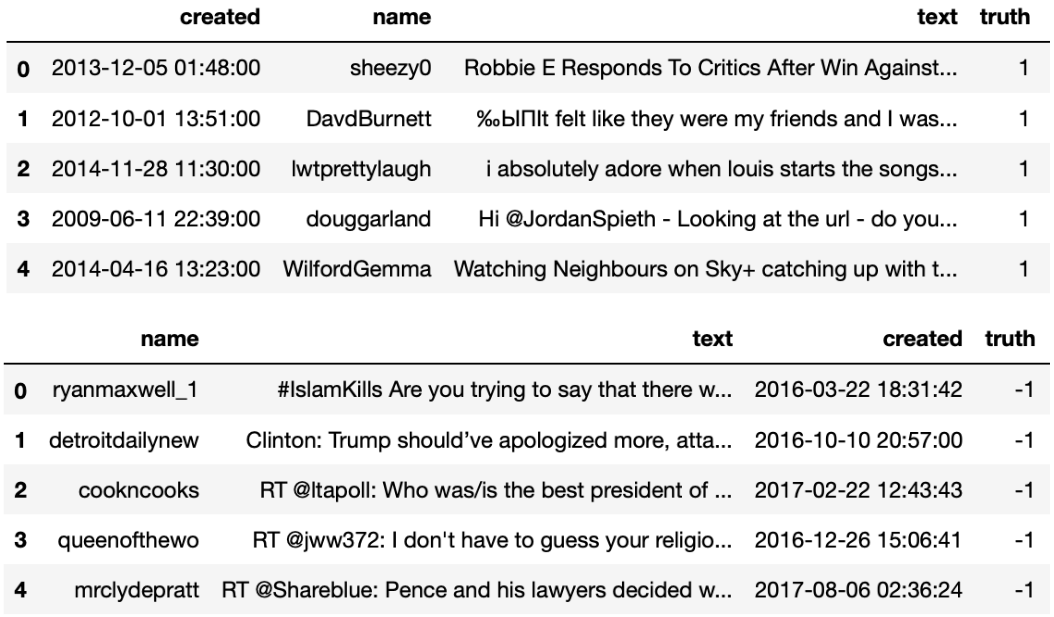
We found a couple papers that mirrored our goal of classifying troll tweets. One of those that was particularly interesting was *Still Out There: Modeling and Identifying Russian Troll Accounts on Twitter* by Jurgens et al.[5]. While they classified troll users and not troll tweets their methodology and thinking seemed to be quite similar. Using an adaptive boosted decision tree they obtained a precision of 94.2%. In their model they also utilized various creative features that greatly aided in their classification. For instance, they used features such as time since creation of account and ratio of followers to following as these seemed to correlate with troll users. They also included features such as which language was used and what hashtags were included in their tweets to further increase their precision. However, one of the most intriguing parts of the paper was the slightly more abstract analysis of the troll users profiles. The authors found a couple underlying similarities in the profiles of many troll accounts. For example, they found that there is often a disconnect between the user's twitter bios and the tweet content. They use the example of a troll having the name of a university in their bio but never tweeting about anything related to this university. Overall, this paper was very informative of some of interesting features that correlate troll users from a profile perspective and not just a tweet perspective.

4 Datasets

For our project, we utilized two pre-existing (unprocessed) datasets to train our model. The first dataset is titled *Russian Troll Tweets* and contains 200,000 examples of troll tweets that were released by NBC News after Twitter allegedly deleted them off of their platform [1]. This dataset contains the text of a tweet, the username of the account that tweeted it, and the time at which it was tweeted. The other dataset we obtained is titled *Twitter User Gender Classification* and contains 20,000 random tweets from real users. This dataset also provides features such as time of tweet and username that were used similarly to the first dataset. We had to combine two different datasets as there were no datasets that had both troll and real tweets. After searching through many datasets, these two seemed to have the least discernible differences, while also including tweets from around the same time period. Of these datasets, we randomly sampled 10,000 from the troll dataset and 10,000 from the normal dataset for our training set. (We initially used a larger training set, but in the end were limited by time and computing power) Excluding the examples chosen for our training set, we randomly chose another 1,000 troll tweets and 1,000 non-troll tweets for our test set.

Example of Real Tweet from the dataset:
("Should I bring my normie boyfriend to the rock gig tomorrow night to meet all my friends?",1)

Example of Troll Tweet from the dataset:
("Hillary Invented Birtherism: 11 Things the Media Won't Tell You",-1)



	created	name	text	truth
0	2013-12-05 01:48:00	sheezy0	Robbie E Responds To Critics After Win Against...	1
1	2012-10-01 13:51:00	DavdBurnett	‰ЫПIt felt like they were my friends and I was...	1
2	2014-11-28 11:30:00	lwtprettylaugh	i absolutely adore when louis starts the songs...	1
3	2009-06-11 22:39:00	douggarland	Hi @JordanSpieth - Looking at the url - do you...	1
4	2014-04-16 13:23:00	WilfordGemma	Watching Neighbours on Sky+ catching up with t...	1

	name	text	created	truth
0	ryanmaxwell_1	#IslamKills Are you trying to say that there w...	2016-03-22 18:31:42	-1
1	detroitdailynew	Clinton: Trump should've apologized more, atta...	2016-10-10 20:57:00	-1
2	cookncooks	RT @ltapoll: Who was/is the best president of ...	2017-02-22 12:43:43	-1
3	queenofthewo	RT @jww372: I don't have to guess your religio...	2016-12-26 15:06:41	-1
4	mrclydepratt	RT @Shareblue: Pence and his lawyers decided w...	2017-08-06 02:36:24	-1

Figure B: Original Data

We used Pandas to clean up the format and isolate certain features such as month, day and hour, which we thought would be beneficial in our model.

5 Baseline Model & Oracle

Our baseline predictor was a binary classifier using logistic regression. The features were simply all the words in the tweet separated by white space. Using this model we obtained a training accuracy of 94% and a test accuracy of 72%. While fairly accurate for a baseline, we were optimistic that we could greatly improve on this performance. Since there were no models online that were performing the same task on the same dataset, we decided the best option for our oracle was to attempt to classify the tweets ourselves. To do this, we both selected 50 tweets at random and predicted whether the tweet was a troll or legitimate. Averaging our scores, we came out with an accuracy of 65% and an f1 score of 66%. Obviously this is not an ideal oracle. We see that our baseline model actually performed better than our own predictions. With an accuracy of 65%, we see how difficult it is for humans to differentiate between real tweets and those that are trolls.

6 Simple Model: Naive Bayes

To improve upon our baseline model we decided to implement a Naive Bayes algorithm that utilizes a bigram feature to classify the tweets. We thought that Naive Bayes would better capture the patterns within the data as there were likely fairly nuanced similarities between troll tweets that were not captured by logistic regression. To create this model we first read in all the labeled tweets to gather bigrams and counts for both the 'troll' class and the 'real' class. These collected counts were used to establish the preliminary probabilities for each class and the bigrams were used to establish the posterior probabilities throughout the Naive Bayes algorithm.

After reading in the labeled training data, we were then ready to run the classifier on an unseen tweet. We first established the priors for both classes based on the number of training examples of each class. We then went through each of the bigrams in the unseen tweet to update the probabilities of each class based on the probabilities of each bigram appearing in the two classes. After going through all the bigrams in the tweet, we compared the two probabilities of each class and returned the more likely of the two.

Running our classifier on this Naive Bayes model using bigrams, we obtained a raw accuracy of 82% on the test set. However, we realized that raw percentage accuracy was not capturing the whole picture of our model. We also needed to take into account false positives and false negatives to get a truly well rounded picture of our model. Thus, we incorporated the use of a confusion matrix and an f1 score. Using this metric our Naive Bayes model achieved an f1 score of 74%.

7 Challenges

At this point, we had improved from our baseline but we knew we had to come at the problem from a new angle. We realized there was almost certainly similarities between the troll tweets. However, these seemed hard to pinpoint as the troll tweets were not always targeting the same groups (i.e. democrats or republicans). Thus, we couldn't just pinpoint certain keywords in the tweets. We had to get more inventive with our feature extractions and the type of model we were using. Taking all of this into account we settled on a final approach to our classifier. We decided to create a stacked model – a feed-forward neural network that utilizes more complex features along with the predictions outputted by our simple models.

8 Final Model: Stacked Model with Neural Network

For our final model, we utilized a stacked model that would allow us to improve upon the predictions we generated from our simple models. The pipeline for our final model is shown above (Figure A). The architecture of the neural network itself is below (Figure C). We had 2 hidden layers with 16 nodes and 8 nodes respectively. Additionally, we used a ReLU activation function with each of these to prevent the vanishing gradient problem. Our output utilized a sigmoid function in order to return a +1 or 0 for troll or non-troll.

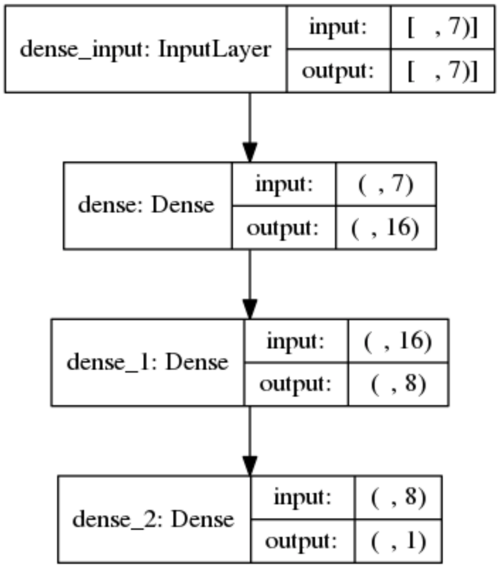


Figure C: Neural Network Architecture

As mentioned earlier, the most prominent features of our neural network are the predictions (+1,0) from our basic models: logistic regression models that trained on words, bi-grams, 3-grams, 4-grams, and a combination from the tweets, as well as a Naive-Bayes that trained on bi-grams. Our goal in using this stacked model was to smooth out the predictive errors from our basic models. Using this ensemble learning method, our neural network would find latent patterns in the strengths and weaknesses of each of our basic models.

We added 8 additional features to our neural network that we believed could improve to our predictions. The first 3 were month, day, and hour. We believed there must be some some time of the day when troll agencies were more likely actively pump out fake tweets. However, as shown by our heat map (Figure F), the hour did not have a high correlation with the rest of our features. Additionally, we thought there must be a month or season when trolls were more active (perhaps around elections, holidays, etc). However, this was also not the case, as shown in our heat map. Ultimately, we removed these features from our model.

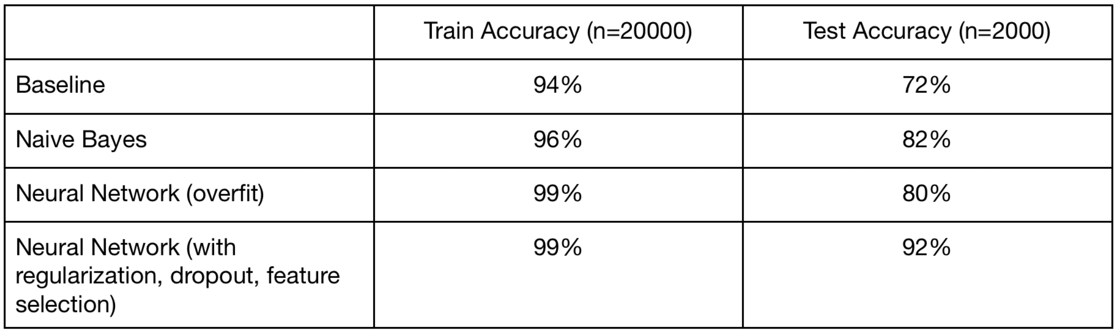
The other 5 features we added were the count of numerical digits in the username, as well as special (non-alphabetical) characters, capitalized letters, hashtags, and mentions in the text of the tweet. Although we initially used a raw count of each, our accuracy greatly improved when we normalized each to a fraction of the total number of characters in the respective tweet/username. Of these, the most powerful features were the numerical digits in the name and the number of non-alphabetical characters in the text. We were glad to see that our suspicions about usernames with many numerical digits was correct. We also would have been interested in utilizing the profile picture of a user as a feature. Unfortunately, we were unable to obtain this data.

Initially, we dealt with severe overfitting. We approached this issue from multiple angles. To start, we added L2 regularization to our loss function. This ensured that the norm of our weights would not become unreasonably large. Second, we utilized dropout, removing hidden nodes with a probability of 0.4. Third, we incorporated early stopping. Our loss and accuracy began converging, and as a result overfitting occurred after roughly 15 epochs (Figure D and E). Ultimately, we found that 10 epochs was optimal for our model. Lastly, we removed features that we deduced from our heatmap (Figure F) and pair-plots (Figure H) to not be very indicative of the true label. We ended up removing the date-related features as well as "capsInText," "hashTags," and "mentions."

Finally, we performed hyper-parameter tuning as part of our cross-validation. We first utilized a random search on optimizer and number of hidden layers, batch size, and dropout on a logarithmic scale. We followed up with a grid search on a linear scale. Unfortunately, our hyper-parameter tuning did not lead to noticeable improvement in our model.

Note: We also attempted to fit a random forest model with identical features, but it did not perform as well as the neural network. The decision tree from that model can be found at the end of this paper. (Figure I)

9 Results



	Train Accuracy (n=20000)	Test Accuracy (n=2000)
Baseline	94%	72%
Naive Bayes	96%	82%
Neural Network (overfit)	99%	80%
Neural Network (with regularization, dropout, feature selection)	99%	92%

Figure J: Highest % Accuracy Obtained for Each Model

10 Evaluation & Analysis

As previously mentioned, we decided to utilize a confusion matrix paired with an f1 score to better address the accuracy of our model. The final model we used ended up with achieve an f1 score of 92% [Figure G]. This was far better than the 82% achieved by our Naive Bayes model and the 66% achieved by our human prediction oracle. Looking deeper into the confusion matrix for the output of our final model, we noticed that it performed better on identifying real tweets ($\approx 95\%$) than on troll tweets ($\approx 89\%$). We believe this is due to the fact that the troll tweets were about a diverse array of topics, a good amount of which may have stood out as outliers. For example, some of the troll tweets in our dataset were about seemingly innocuous topics such as gaming. On the same note, we must keep in consideration that our train and test sets were not completely uniform due to the fact that the troll tweets and non-troll tweets had to come from two different sources. While we did our best to minimize implicit similarites between the respective types of tweets, we were unable to completely nullify this effect. It is possible that similarities within the troll dataset and non-troll dataset enhanced the performance of our model. Nonetheless, we feel that we did our best with the resources we had and learned a lot from this experience.

11 Social Importance

We will end this paper with a broader discussion on the social impact of this issue and how work along the same lines as our project fit into the picture.

After an analysis from Jurgens, et al.[5], they estimated that up to 4% of accounts that mention many top journalists on twitter are actually Russian troll accounts. And it is not only journalists who are interacting with these accounts. Twitter even released the fact Donald Trump's campaign advisor responded to one of these troll users on Twitter[1]. So, these troll accounts are still currently very active and influential on social media.

The question then becomes why companies such as twitter are not utilizing classifiers such as these to reduce the amount of impact that these troll accounts will have. The idea of free speech immediately jumps to mind as to why a company such as twitter would maybe be slow to adopt one of these classifiers to ban certain accounts. It seems rational that they would not want to bring about the backlash that wrongly banning certain political tweets would bring about.

However, that is why we believe work in the same vein as our project are socially important. It is only by analyzing these troll accounts that we will develop more accurate ways to identify the trolls and avoid the wrongful banning that likely scares Twitter. As these models become more precise, the collateral damage to real accounts will minimize. With the 2020 election coming in a few months, the timeline for this advancement is shrinking. Thus, we feel that identifying features correlated with troll accounts and developing models to flag these accounts is essential to foster the health of democratic debate on social media. As environments like Twitter have become one of the primary locations for political discussion, we feel there is a strong connection between the quality of conversation there and the political environment as a whole.

12 Figures

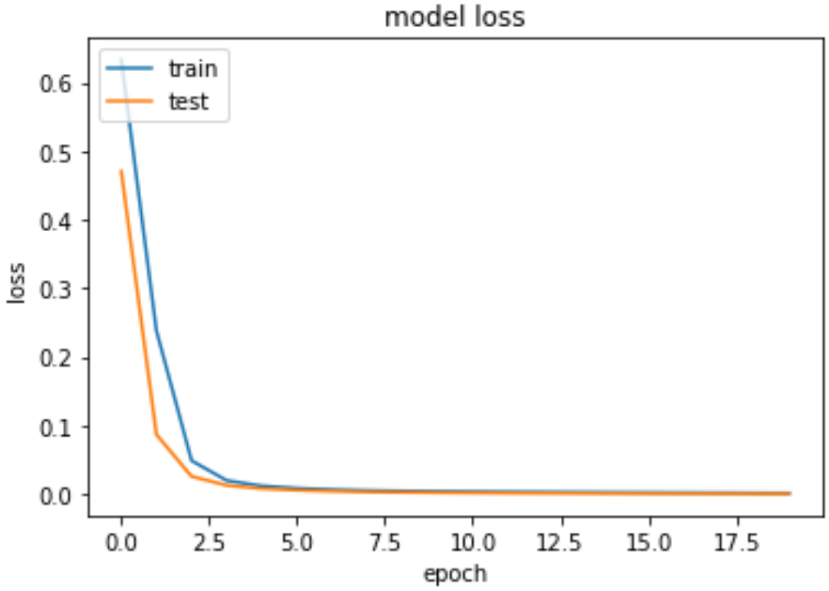


Figure D: Loss vs. Epoch

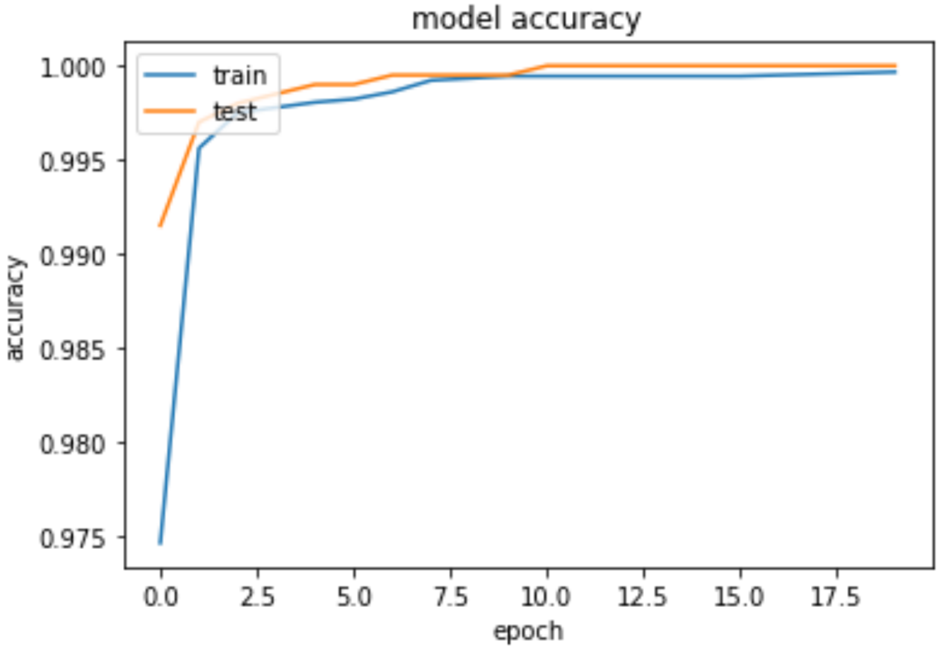


Figure E: Accuracy vs. Epoch

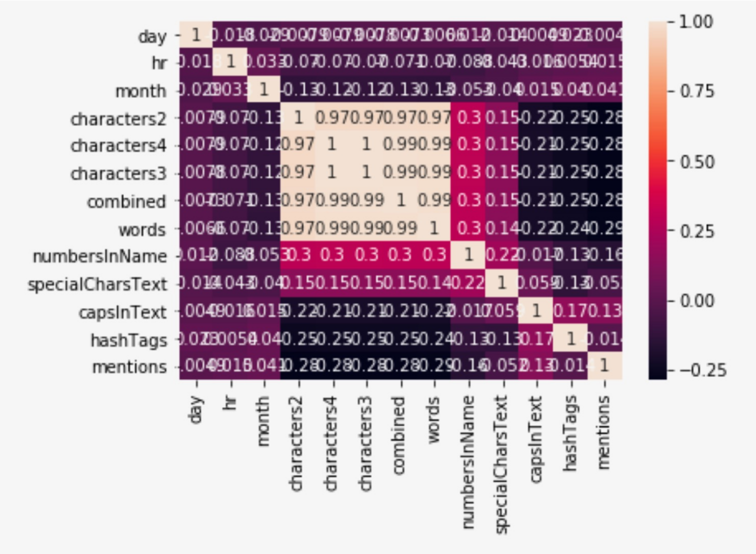
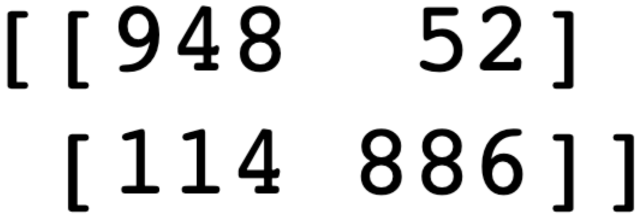


Figure F: Heatmap of Features



```
[[948   52]
 [114  886]]
```

Figure G: Confusion Matrix for Final Results

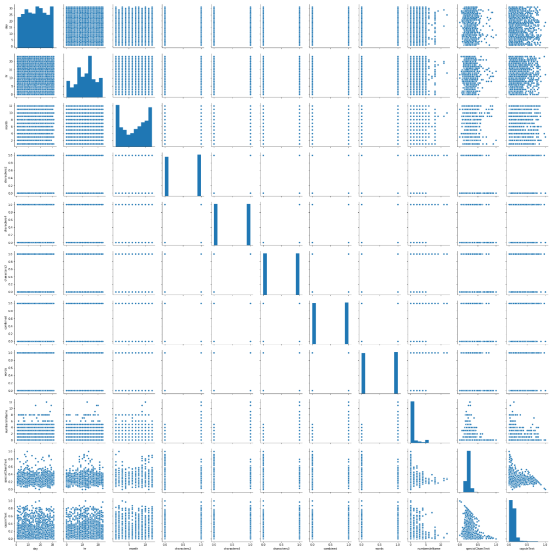


Figure H: Pair-plots of Features

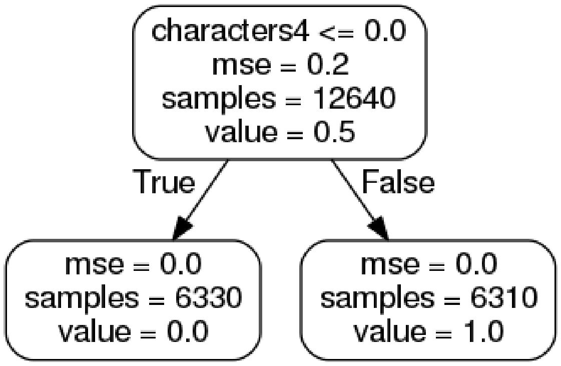


Figure I: Decision Tree for Random Forest Model

Linked Code/Datasets

https://github.com/gntnka/twitter-trolls

https://drive.google.com/file/d/1HiT8IKZySnKv-0PJBiUURep9MtprZX9H/view?usp=sharing

References

[1] Ollie. (2018, July 31). Why We're Sharing 3 Million Russian Troll Tweets. Retrieved from https://fivethirtyeight.com/features/why-were-sharing-3-million-russian-troll-tweets/.

[2] Vikas. (2018, February 15). Russian Troll Tweets. Retrieved from https://www.kaggle.com/vikasg/russian-troll-tweets.

[3] Kudugunta, S., & Ferrara, E. (2018). Deep neural networks for bot detection. Information Sciences, 467, 312-322. doi:10.1016/j.ins.2018.08.019

[4] Zheng, A. (2015). Evaluating Machine Learning Models. Retrieved from https: //www.oreilly.com/ideas/evaluating-machine-learning-models/page/5/ hyperparameter-tuning

[5] Jurgens, D. (2019, January 31). Still Out There: Modeling and Identifying Russian Troll Accounts on Twitter. Retreived from https://arxiv.org/pdf/1901.11162.pdf