Identifying tweet authorship

Avani Gowardhan

December 3, 2018

1 Introduction

This is the description of the final project for the course CS 4780. The project involved identifying whether tweets were authored by Donald Trump (referred to as Drumpf) as opposed to his staff (referred to as Minions). The data was provided for ~ 1000 tweets, with roughly equal numbers in both categories (referred to as D and M henceforth).

2 Methods and Results

We use the following as features: the time of the day, the length of the tweet (in words), the number of hashtags, links and twitter handles in each tweet, and the emotion (positivity, negativity and neutrality) of the tweet as derived using the Natural Language toolkit (NLTK) package's *SentimentAnalyser* [1], the number of times '@realDonaldTrump' is mentioned, and the number of times '#Trump2016' is mentioned (more likely to be mentioned by M). We also count the number of tweets in reply - D tends to start those with a "@person". We show the distribution of the most relevant features for D vs M in Fig. 1. It can be seen that D tends to have longer tweets, earlier in the day, has a lower positivity and higher negativity, mentions himself quite often - as seen in the wordcloud of twitter handles, and refers to his children much less frequently.

[2]

Using the above set of features, we use the Random Forest, AdaBoost, and GradientBoosting algorithms implemented in Scikit-Learn [2] to classify the tweets. We use the VotingClassifier to ensemble all algorithms with equal weights, using soft voting. For each of these algorithms and their ensemble, we used the function scikit-learn function cross_val_score with a cv = 5

Feature count	Occurrence - Minions	Occurrence - Drumpf
Number of links	418	55
Number of hashtags	449	83
Number of handles	123	594
@realDonaldTrump	1	119
#Trump2016	168	16
Ref. to children*	168	69
Number of allcaps words*	229	414

Table 1: Total number of each feature in all tweets by D and M - we only use the first 4 as features.

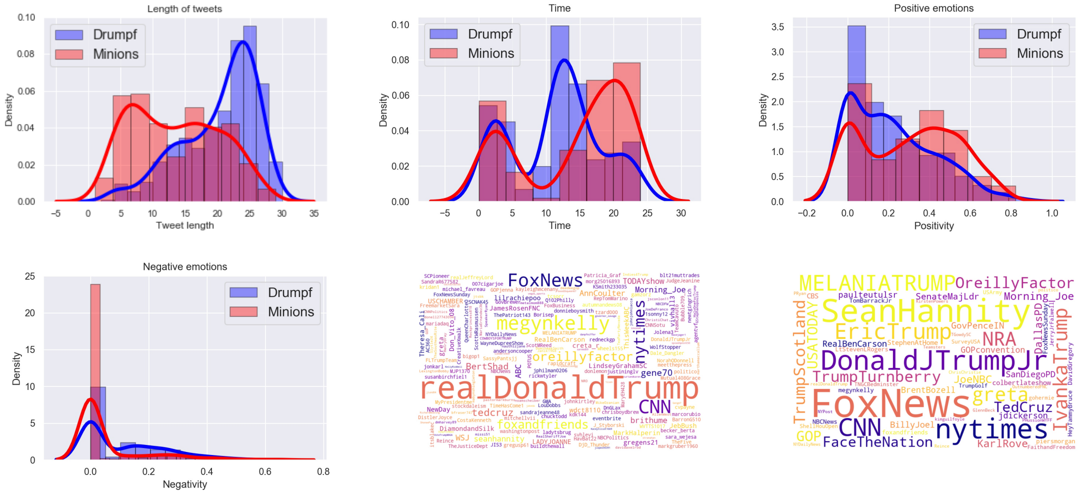

Figure 1: The histograms show the distribution of the time of the tweet, the length of the tweet, and the positivity and negativity index associated with the text of the tweet, for D and M. The wordclouds in the last panel show the frequency of different twitter handles as seen in tweets of D and M, demonstrating that D uses '@realDonaldTrump' more frequently than M, making this a usable feature.

Algorithm	Accuracy	top 3 features
Adaboost	0.88 ± 0.02	time of tweet, sentiments (positive/negative)
Random Forest	0.90 ± 0.01	number of links, time of tweet, length
Gradient Boost	0.88 ± 0.02	number of links, retweet, time of tweet
Ensemble	0.89 ± 0.02	-

Table 2: cross-validation accuracies for different models.

to get the accuracy of the classifiers, listed in Table 2. The different methods also applied very different importances to the different features, suggesting that the classifier would generalize better with ensemblimg, and making it likely that adding more classifiers would make it better yet. We optimized the number of estimators in each using a GridCV search for the best parameters, though it was very limited because of the large parameter space.

3 Kaggle submission

The predictions were created using the same data processing as the training data, before applying the ensemble classifier to the data. The submissions were made under the username: 'mona', and achieved a score of 0.85 on the public leaderboard. The code structure includes a jupyter notebook as well as a utils.py which includes the helper functions used for feature engineering.

References

[1] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.