Identification of Donald Trump's Tweets Using Machine Learning

Lina Ahmad, Amjed Al-Mousa
Computer Engineering Department
Princess Sumaya University for Technology
Amman, Jordan
a.almousa@psut.edu.jo

2021 18th International Multi-Conference on Systems, Signals &amp; Devices (SSD) | 978-1-6654-1493-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/SSD52085.2021.9429330

Abstract—The current president of the United States of America, Donald Trump, is well known for his active Twitter account where he shares his personal daily thoughts using the Twitter handle '@realDonaldTrump'. This paper presents the use of classical machine learning techniques in an attempt to analyze Trump's use of words. Afterward, multiple steps of data preprocessing are applied to build a dictionary of definitive words Trump continuously uses, subsequently feeding this dictionary to a number of machine learning models to correctly classify a tweet as written by Trump or not. The accurate result this project has produced proves that Donald Trump has developed a certain pattern of word usage in what he chooses to criticize or shed light upon.

Keywords—machine learning, text analysis, Twitter, Trump, random forest classifier, multinomial Naïve Bayes classifier, support vector classifier

I. Introduction

Donald Trump was known for his controversial tweets long before he became president. Since his first tweet in 2009, Donald Trump has been attracting attention with his political thoughts and on-going feuds with various news broadcasts, celebrities and the previous POTUS (President of The United States), Barack Obama. This project attempts to collect the entire dataset of Donald Trump's tweets from 2009 till 2019 along with the tweets of random users to understand the patterns and words used by Trump in his tweets and finally be able to tell whether a tweet is Trump's or not. Data will be cleaned using text analysis and data processing techniques to filter the tweets to be used in feature extraction, which will be fed to classical machine learning models used for text, based applications.

Section II of the paper presents a review of related work; section III introduces the details of the dataset along with the cleaning and preprocessing of data. The model training is presented in section IV and the results are in section V. Finally, future work and conclusion are outlined in VI and VII.

II. Related Work

Ever since Donald Trump was elected, many have taken notice of his Twitter account, and with the current technological impact of machine learning and AI, research has been conducted to analyze Trump's use of words and even predicting the times of the day he tends to tweet [1]. The project studied Trump's Twitter activity during his campaign and after he was elected as president. Although actual timing of his tweets was not as predictable as expected, a study of his activity during days of the week and hours of the day before and after his election was presented.

In [2] neural networks were used to distinguish tweets written by Trump himself and ones written by the White House staff, analysis of source tweets by device, iPhone, Android, or Web to distinguish the different words used, according to the article, the model was 100% confident that it is Trump himself tweeting. Another project was taken using sentiment analysis to predict Trump's trade moves in the market [3]. Each tweet identified as related to trade was used and assigned a sentiment score. The huge dataset collected and used by most projects relating to Trump's twitter account is in reference to the Trump Twitter Archive [4]. However, most research done has extracted and focused on data after 2016 when Trump was elected. Moreover, deep learning techniques were used and no projects of Donald Trump's text analysis using classical machine learning algorithms were found.

III. Data Preprocessing & Analysis

For this project batch learning will be used, performance will be measured in terms of the precision metric, as it is important that the accuracy of the model classifying tweets written by Trump to be truly his. The ROC curve will also be used in the diagnosis, as there are roughly equal numbers of instances for each class. Two sources were used to acquire the dataset of tweets by Trump [4], and tweets by normal random users [5].

A. Data Description

The Trump tweets initial dataset contained seven fields; 'Tweet ID', 'Text', 'Created At', 'Favorites', 'Retweets', 'Is Retweet', and 'Source'. Only the 'Text', 'Created At' and 'Is Retweet' fields were taken as the other fields are of no interest regarding that Trump's tweets would definitely get more retweets and favorites than normal users'.

978-0-7381-4392-7/21/$31.00 ©2021 IEEE

The Users' tweets dataset contained four fields; 'id', 'time', 'text', and 'gender'. The source itself has three datasets, which were combined into one. The 'time' and 'text' field were only used.

B. *Data Preparation*

All rows that had the 'is_retweet' field set as True in Trump's tweets dataset were dropped. After that, the whole column was dropped. The timestamp feature called 'created_at' that contained the time and date was split into two distinct attributes; 'time' and 'date'. A new feature named 'Is_Trump' was added to the dataset filled with 1s to indicate all instances tweeted by Trump. Trump tweets dataset contains 39488 instances.

The random users' data also had a timestamp named 'time', which was subsequently split into 'time' and 'date'. 'Is_Trump' feature was added and filled with 0s. User tweets dataset contains 55510 instances.

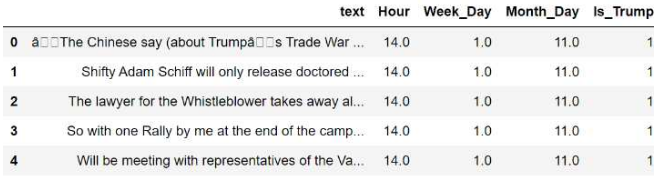
The 'time' feature in both datasets has been used to extract the 'Hour' feature then dropping the seconds and minutes. The 'Is_trump' feature will be used as the label for the model. The 'date' feature had to be converted to a unified format. Both datasets were uploaded and merged to form a new dataset with four features; 'text', 'date', 'Hour', 'Is_Trump'.

C. *Data Reformatting*

Using excel, the 'date' attribute was reformatted and used to filter out the day of the week and day of the month. 'Week_Day' represents the day of the week starting from Monday '1' till Sunday '7'. Day of the month is represented by 'Month_Day'.

Upon uploading the dataset into a Jupyter notebook, the dataset contains 94998 instances and 5 features. Table 1 shows a snapshot of the first five instances of the dataset.

Table 1: Top five rows of dataset



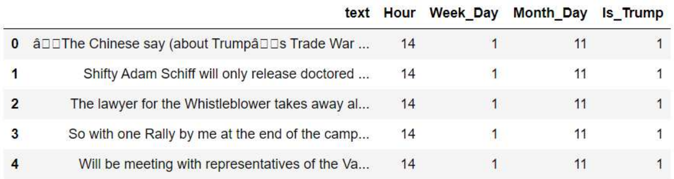
	text	Hour	Week_Day	Month_Day	Is_Trump
0	â□□The Chinese say (about Trumpâ□□s Trade War ...	14.0	1.0	11.0	1
1	Shifty Adam Schiff will only release doctored ...	14.0	1.0	11.0	1
2	The lawyer for the Whistleblower takes away al...	14.0	1.0	11.0	1
3	So with one Rally by me at the end of the camp...	14.0	1.0	11.0	1
4	Will be meeting with representatives of the Va...	14.0	1.0	11.0	1

D. *Data Investigation*

Upon data inspection, the 'Hour', 'Week_Day', and 'Month_Day' features contain null values. Moreover, the dataframe object identified the previously mentioned features as floating numbers. To solve the data type problem, all null values must be addressed first. Using the imputer function, they were filled by most frequent value.

Data type conversion was used to convert the three mentioned floating number fields to integers. Table 2 shows the first five instances after conversion.

Table 2: Top five rows after formatting



	text	Hour	Week_Day	Month_Day	Is_Trump
0	â□□The Chinese say (about Trumpâ□□s Trade War ...	14	1	11	1
1	Shifty Adam Schiff will only release doctored ...	14	1	11	1
2	The lawyer for the Whistleblower takes away al...	14	1	11	1
3	So with one Rally by me at the end of the camp...	14	1	11	1
4	Will be meeting with representatives of the Va...	14	1	11	1

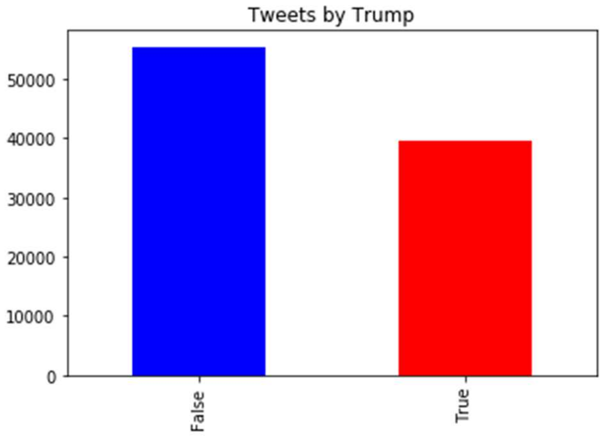


Figure 1: Value of Donald Trump's tweets against users'

The 'Is_Trump' feature was used to graphically represent the sum of instances that are by Trump (True) and those who are by random users (False).

Figure 1 resembles the number of tweets in the collected dataset that are by Donald Trump.

E. *Data Visualization*

In order to understand the data and its labels, the correlation of the numerical attributes against each other is of great importance. Table 3 shows the correlation matrix.

The 'Month_Day' attribute has resulted in a 54% negative correlation with 'Is_Trump', the output label. Going back to the dataset where user tweets were taken from, it can be seen that most tweets lie between the first and last days of every month. Using the describe function further confirms the bias, by inspecting the 50th percentile of the 'Month_Day' attribute, half the values are lower than 25. Thus, as to not produce any discriminatory models, this attribute will be dropped from training. The 'Hour' and 'Week_Day' attributes will also not be taken into consideration, as they do not provide attractive correlation with 'Is_Trump'. As a result, the aim of this project is to perform text analysis solely based on the tweets alone.

Table 3: Correlation matrix of numerical attributes

	Hour	Week_Day	Month_Day	Is_Trump
Hour	1.000	0.037	-0.135	0.016
Week_Day	0.037	1.000	-0.004	-0.154
Month_Day	-0.135	-0.004	1.000	-0.543
Is Trump	0.016	-0.154	-0.543	1.000

Before proceeding with data cleaning, visualizing the data and recognizing the words used in tweets has been taken. Figures 2 and 3 show the top 20 most frequently used words by Trump and Twitter users plotted using a bar chart figure.

As both figures show, not much information can be extracted from them, as they both present a high number of stop words such as (the, to, and). This indicates that text preprocessing must be taken before being fed into any machine-learning algorithm.

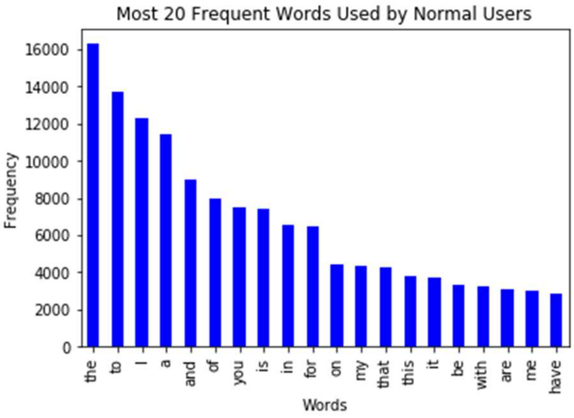


Figure 2: Most 20 frequent words by users

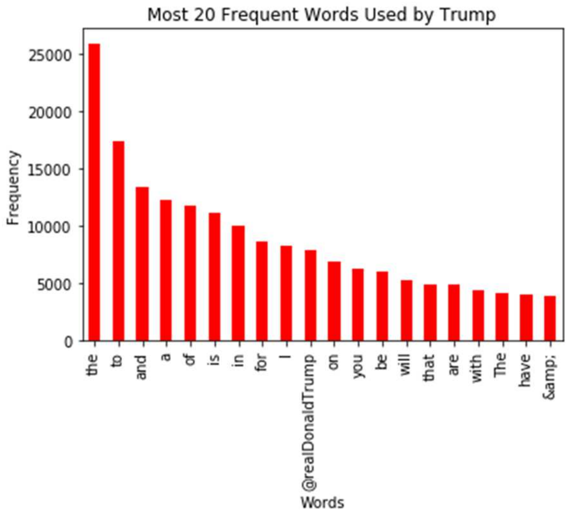


Figure 3: Most 20 frequent words by Trump

F. *Data Preprocessing*

The dataset has been split into training and test sets using cross validation. Text preprocessing occurs in a number of phases:

1. Make all letters lowercase
2. Remove non-alphanumeric characters
3. Remove stop words: such as ‘the’, ‘and’, etc.
4. Tokenization: splitting sentences to words

The flowchart in Figure 4 presents the framework of data preprocessing taken.

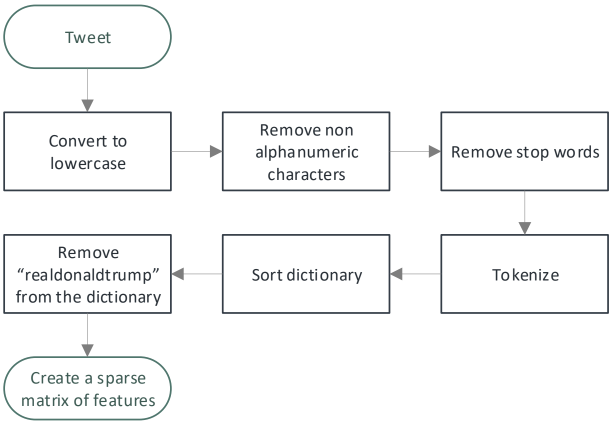

Tweet
Convert to lowercase
Remove non alphanumeric characters
Remove stop words
Tokenize
Sort dictionary
Remove “realdonaldtrump” from the dictionary
Create a sparse matrix of features

Figure 4: Text data preprocessing framework

As shown in Figure 4, each tweet will be fed into a preprocessing function that performs all previously mentioned phases of text preprocessing. This function will prepend each filtered tweet into a dictionary with key: word and value: count for recurrent words. The presented dictionary must be sorted in descending order according to value, so the most recurrent word would be the first in the dictionary. The function was used 3 times, as shown in Figures 5, 6, and 7 show the top 20 most used words by Trump, normal users, and then on the entire training set.

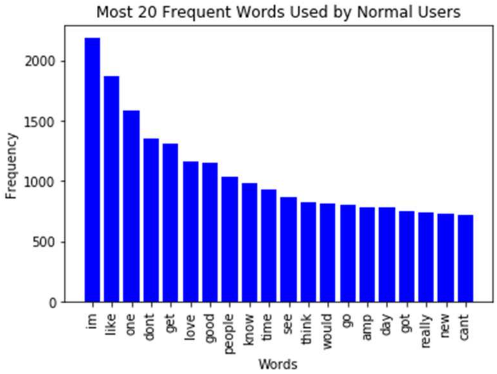


Figure 5: Top 20 words used by Normal Users

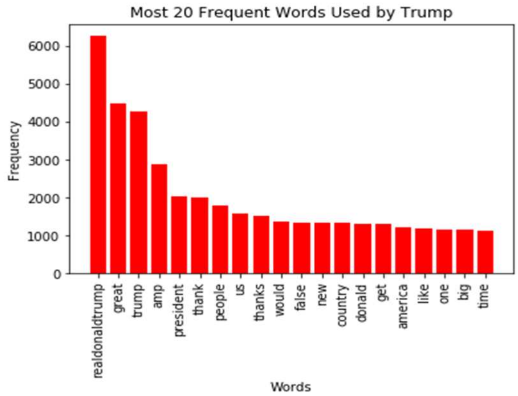


Figure 6: Top 20 words used by Donald Trump

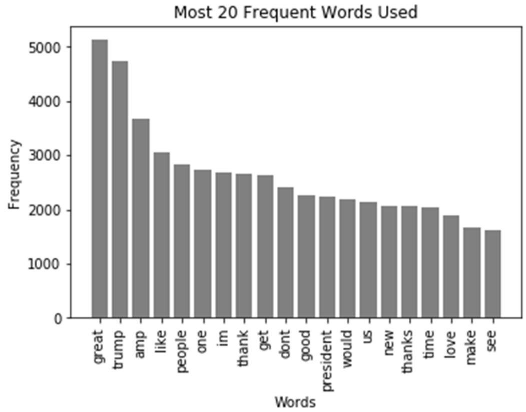


Figure 7: Top 20 most used words by Trump and users

From the previous three figures, interesting words used by Trump such as 'great', 'false' and others that show his political stance; in 'people', 'country', and 'america' as opposed to words used by normal users; 'love' and 'good'.

The word 'realdonaldtrump' in the dictionary created, which was run over the entire dataset, has been removed to create a dictionary of 63069 distinctive words.

The last step in the data preprocessing phase is to create the Bag of Words (BoW), which takes all unique words and adds them as new features to the dataset. It then calculates the sparse matrix that consists of each instance against each word in the dictionary, if a word in the tweet exits in the BoW, it takes value 1 for each matching word, and 0 otherwise. The Count Vectorizer was used to create the BoW.

IV. Model Training and Prediction

The research explores the use of light weight classical machine learning techniques for language recognition and for this reason, three different algorithms were chosen for evaluation: Random Forest, multinomial Naïve Bayes, and Support Vector Classifier. These models were trained on two dictionaries; the first was using the entire words extracted that resulted in adding 76967 features. The second dictionary was built using the 'max_features' attribute in the feature extraction function, a maximum of 30000 features was defined, this will create a dictionary based on the most common 30000 words in the dataset. Thus, the three models were run the second time on 30000 features. The first dictionary of 76967 features will only be used for comparison purposes of measuring performance and overall accuracy against the dictionary of 30000 features to see how well the models actually perform with a dictionary of only the most frequent words used instead of every word used throughout the dataset.

A. Random Forest Classifier

Random forest classifier is an ensemble algorithm that creates a set of decision trees; this project has used a number of 501 trees in the forest and criterion 'entropy', which splits based on information, gain. Table 4 shows the resulting confusion matrix after running the model on the test set.

Table 4: Random Forest Classifier confusion matrix

	Predicted as User	Predicted as Trump
Actual User	13426	485
Actual Trump	1131	8708

The Random Forest Classifier produced a precision of 94%, this can be evident from the confusion matrix as the model has only misclassified 485 Trump tweets as users. The Random Forest Classifier that was run using 76967 features resulted in a 97% test precision.

B. Multinomial Naïve Bayes Classifier

This classifier is a specialized version of Naïve Bayes that is designed more for text documents. Different models of Multinomial Naïve Bayes Classifiers were trained changing the regularization parameter 'alpha'. The 'alpha' hyperparameter controls how smooth the curve is in the classifier. The result was training 182 different Bayes models each with a different set of parameters. A selection of the best classifier was chosen upon the test precision. Table 5 shows the resulting confusion matrix.

Table 5: Multinomial Naïve Bayes Classifier confusion matrix

	Predicted as User	Predicted as Trump
Actual User	11737	2174
Actual Trump	348	9491

The best model chosen produced a total test precision score of 89%. Although the precision score seems low, the recall score was significantly higher with a 95%. Even though the model seemed to have classified almost all Trump tweets as Trump and only missing 348 tweets, it has misclassified 2174 user tweets as Trump, justifying the low precision. When run over the entire dictionary of words, precision score was 89%, the model does not perform better when increasing the features used (more words in the dictionary).

C. *Support Vector Classifier*

Support Vector Classifiers are great for categorizing data into classes. This algorithm was used to train different models changing the regularization parameter C; the C parameter controls the margin of the hyperplane that will be used by the model to classify instances. The best model has also been chosen based on the highest test precision. A total of 100000 support vector classifiers were trained. Table 6 shows the resulting confusion matrix.

Table 6: Support Vector Classifier confusion matrix

	Predicted as User	Predicted as Trump
Actual User	13577	334
Actual Trump	843	8996

The model produced a 96% precision score over the test data. With 334 user tweets misclassified as Trump and 843 Trump tweets misclassified as user tweets. The model trained on the entire dictionary returned a 97% precision score.

Table 7 shows the overall performance measurement percentage of the Random Forest Classifier and the best model trained with highest test precision score on the Multinomial Naïve Bayes Classifier and the Support Vector Classifier.

Table 7: Random Forest Classifier confusion matrix

Classifier	Dictionary Used in Training	Accuracy	Precision	Recall
Random Forest	Entire Dictionary	92	97	85
	Subset of Dictionary	93	94	88
Multinomial Naïve Bayes	Entire Dictionary	93	89	95
	Subset of Dictionary	93	89	95
Support Vector	Entire Dictionary	92	97	85
	Subset of Dictionary	94	96	88

As the Table shows, performance does not drastically change when selecting a smaller domain of words (30000 features instead of 76967). This appealing finding implies that with a smaller dataset (less memory space) all models can still produce high results. The Random Forest and Support Vector Classifiers were both very close in performance and had an overall increase in accuracy when trained on a smaller domain of words. However, the Multinomial Naïve Bayes Classifier's performance has not.

V. Results

Since the performance measurement to be taken in this project is the precision metric, as it is more vital that the model's ratio of correctly classified tweet as belonging to Trump would be high, the best model found was the SVC (Support Vector Classifier) with the highest precision test score. Not only has it performed better than the other two algorithms used but it has also showed great results even when given a dictionary of 30000 features.

Figures 8, 9, and 10 show the ROC (Receiver Operating Characteristic) curves of the three models with their respective AUC (Area Under the Curve). The ROC curve can effectively evaluate each model's performance of diagnostic tests. The closer the AUC is to one, the better the overall performance of the model against the test.

The resulting ROC curve of the SVC shows how close it is towards the top-left corner. Another way to compare the three models is using the AUC (Area Under the Curve) that is shown next to each ROC curve: 0.93 for Random Forest, 0.90 for Multinomial Naïve Bayes and 0.95 for SVC. These three values further prove the performance of the SVC in identifying Trump tweets.

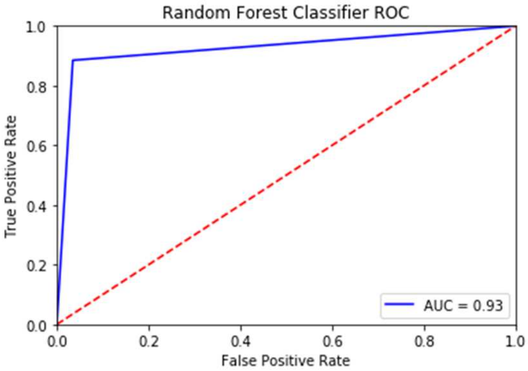


Figure 8: ROC curve of Random Forest Classifier

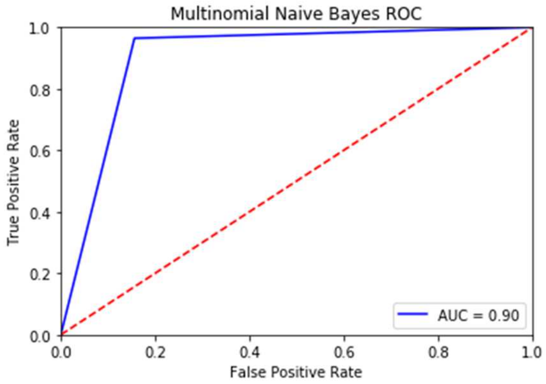


Figure 9 : ROC curve of Multinomial Naïve Bayes Classifier

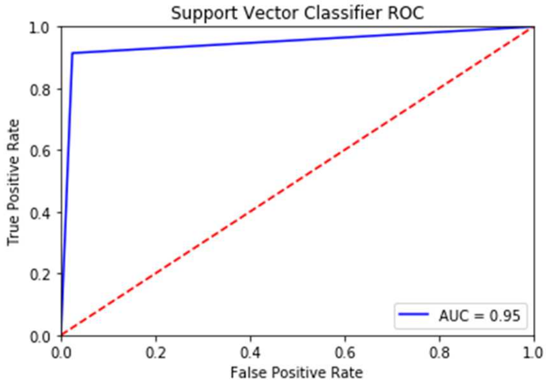


Figure 10: ROC curve of Support Vector Classifier

VI. Conclusion

In this study to identify Donald Trump's tweets, real data was captured from twitter and from Trump's account over the last 10 years then was used to construct a BoW. Even though no known existing work has used as a large data set this project has, as similar work done focus on his post-election days and make use of neural networks techniques to distinguish Trump's vocabulary. However, this project shows that even with the use of classical machine learning techniques over a large dataset, similar accurate results were obtained. Three different models were trained for text analysis purposes, the best model that resulted with an overall test accuracy of 96% was the Support Vector Classifier. This shows that Donald Trump seems to have a certain choice of words and vocabulary in what he decides to tweet about and they are not as unpredictable and random as others opt them to be. This enabled the classification of his tweets with high accuracy.

Going forward this work can be expanded to use deep neural networks to achieve similar task. While deep neural networks might have high computational cost, the target would be to achieve similar or better accuracy with pretrained deep neural networks.

Acknowledgments

The authors would like to thank PSUT for supporting the publication of this research, which is an extension of a machine learning class project.

References

[1] M. Karolian, *Trump's Tweets Are Unpredictable. But When He Sends Them Is a Little More Regular*, The Boston Globe, Jan. 19, 2017. Available: https://www.bostonglobe.com/metro/2017/01/19/trump-tweets-are-unpredictable-ever-but-when-they-happen-has-become-more-routine/IR8hT4V99c0MXenTdKpDUI/story.html

[2] J. Allen-Robertson, *Finding Trump with Neural Networks*, Medium, Jun. 15, 2018. Available: https://towardsdatascience.com/finding-trump-with-neural-networks-4419468e0624

[3] N. Cochrane, *Trump, Tweets, and Trade,* Medium, Sep. 6, 2019. Available: https://towardsdatascience.com/trump-tweets-and-trade-96ac157ef082

[4] B. Brown, *Trump Twitter Archive,* Available: http://www.trumptwitterarchive.com/

[5] A. Harless, *Tweet Files For Gender Guessing*, Kaggle, Available: https://www.kaggle.com/aharless/tweet-files-for-gender-guessing#twitgen_train_201906011956.csv

[6] Pablovargas, *Naive Bayes & Svm Spam Filtering*, Kaggle. Available: https://www.kaggle.com/pablovargas/naive-bayes-svm-spam-filtering