COMP 562 Final Project

Classifying News Articles as Fake or Real News by Article Title

by Jacob Smith, Noah Van Hook,
Gayathri Raghavendra, and Jacob Dallin

November 21st, 2020

1 Introduction

This calendar year has been one filled with unprecedented events and circumstances. The world has faced challenges this year that have never been faced before, and in a time of such significant change, people have looked everywhere and anywhere for answers. However, a new problem has arisen in the form of misinformation. With increasing access to technology and the rapid explosion of social media use, misinformation is easier to spread now than ever before. An analysis by Buzzfeed News found that fake news articles outperformed real news in the lead up to the 2016 election, meaning that more fake news was consumed, read, and shared than real news. With today's political climate, having access to factual, accurate news is absolutely crucial to remaining a well-informed citizen.

2 Our Objective

The purpose of this project is to build a robust model that can accurately predict whether a recent news article contains misinformation based solely upon that article's title. We decided to focus solely on an article's title as opposed to the body of prose because oftentimes the title of an article will be the deciding factor that hooks new readers. Our hope was that having an accurate model for predicting false news articles would actually highlight patterns in language that we hadn't noticed before. Then we could use these recognized patterns/behaviors in the future when deciding not only if we should read a news article, but if we should uphold its contents as being the truth.

3 Related Work

In beginning our work, we reviewed the work of Hadeer Ahmed, Issa Traore, and Sherif Saad who put out two separate papers on classifying fake and real news. In one, the method employed was text classification. They tokenized their dataset, then did feature extraction. They used those features for training their classifying, which would then label an input as false or truthful. They attempted a few different models, such as K-nearest neighbor, LR, and decision trees in addition to SVMs and stochastic gradient descent. In another paper, they created an n-gram model to train. An n-gram model utilized a sequence of n words as tokens as opposed to using individual words. This is a very popular technique in natural language processing. They actually discovered that they were achieving the highest accuracy on their models using a unigram, which is the same as tokenizing the data by individual words. We employed a unigram for our bag of words vectorization model.

4 The Plan

Our strategy to build this model is as follows. First, we gathered our data: we used datasets of "fake" and "true" news articles from Kaggle containing information for 17,903 and 20,826 unique news articles respectively. Each observation in our dataset contained the article's title, the body, and the date. Intuitively, all the articles in the "fake" dataset had been proven as misinformation, and the articles in "true" were deemed truthful.

Two methods we tried for representing the information in each headline were each headline as a bag of words, and analyzing character frequency. Given the observation that clickbait fake-news headlines typically overuse specific characters like exclamation points or capitalized letters, we expected that analysis based on characters as compared to words would provide more insightful results that could help users recognize false news in the future.

Additionally, we implemented a sentiment analysis model for our data. Our initial thought process was to see if there was any possible predictability between the positive, negative, neutral tone used in the headlines. We used the libraries VADER (Valence Aware Dictionary and sEntiment Reasoner) Sentiment and TextBlob to help aid in this analysis. We created a set of attributes for each article title: negative, neutral, positive, a composite sentiment score, and subjectivity. With these features and only two intended classification groups, Naive Bayes seemed like a natural choice. We also implemented the Support Vector Machine model for similar reasons, to identify any natural boundaries that might arise.

Similarly, when analyzing the titles based on specific characters, we decided to use a Naive Bayes model based on a Bernoulli distribution. We also continued our character-based analysis further and built a model using a support vector machine. These models both used a set of 119 features (characters) that ranged across all unique characters encountered in the dataset.

When analyzing the titles based upon the words contained within, we decided to use a 'Bag of Words' representation for the individual words that would make up our set of features. We continued to use the same model as before, a Bernoulli Naive Bayes distribution. However, the task of finding an effective set of feature values was much more difficult than before. The titles needed to be split into individual words, tokenized, cleaned up, and then re-evaluated. After this initial cleaning had been done, our feature count was still entirely too high. In order to shrink this list further, we filtered out "stopwords", removed all single-letter words, and only kept the 1,000 most common words, giving us 1000 features.

5 Results and Conclusions

Results Our final product was a model that can predict the legitimacy of a news article based on its title with 99% accuracy. The model we used to achieve this was the SVM Machine used with unique characters as features, not words.

Conclusions After building several models, we discovered that those which used unique alphanumeric characters as features were more accurate than identical models that used individual words as features. We believe this to be the result of extensive use of punctuation and capitalization in the titles of articles deemed as "fake news." One example of such a headline would be: "Racist Alabama Cops Brutalize Black Boy While He Is In Handcuffs (GRAPHIC IMAGES)". As previously mentioned, this headline includes use of capitalized words, a recurring pattern amongst "fake news" articles. But also from this example, one might hypothesize that the use of negative language may be a more accurate predictor.

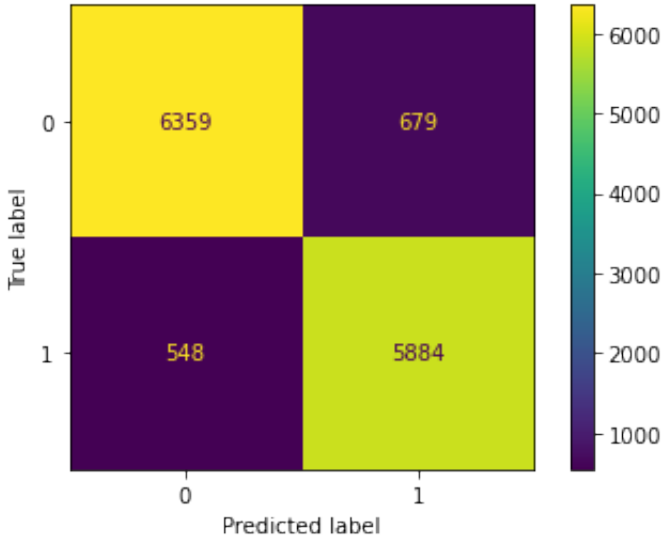

Figure 1: Confusion Matrix for Bag of Words using Naive Bayes

However, our models on sentiment analysis proved to be much less accurate than those that used characters for prediction. This may be because of the particular algorithm's weight on capitalization versus other factors of sentence structure. In addition, a quick analysis of the average polarity and subjectivity scores of article titles revealed that there were still quite a few real headlines registering with high polarity and subjectivity, when these metrics were hypothesized to be low. This might be so if a quote was used in the title or any indications of bias were detected, as real headlines are still not immune from usage of subjective language. A possible explanation for this phenomenon is that if a quote was used in the title or any indication of bias was detected a real headline, it would be flagged as fake even though it is still factually true.

The success of the model based on character frequency is, however, not necessarily a triumph of machine learning. It is very possible that the party responsible for the making of the dataset in question, when selecting articles that are fake vs articles that are real, were much more drawn to titles with heavy use of capital letters and punctuation. In this way, the model using characters would be doing the same thing as the individual looking for fake news articles, making it no surprise that the algorithm performed well. This model has a vulnerability against attackers, though: an analysis of headlines erroneously classified as fake news showed almost all of these headlines had some an element in all-caps, like "USAID", "NAFTA", and "NAACP", whereas headlines erroneously classified as true contained nothing in all-caps. If such a model were enacted as some sort of widespread safeguard against fake news, it would be easy for nefarious misinformation artists to sidestep the machine's gaze by using more lowercase letters, less punctuation, and more bombastic language to make up for the less emphatic lettering. With such results, it may be that our "fake news" model would be a better fit for rooting out clickbait than for identifying fake news.

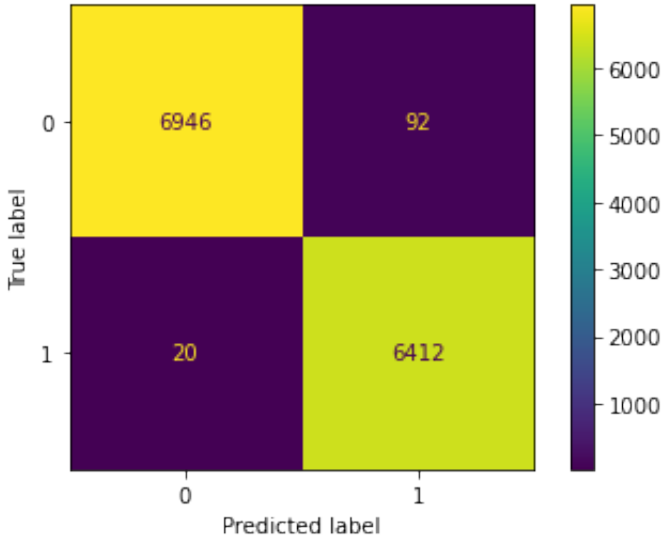

Figure 2: Confusion Matrix for SVM Character Analysis

6 References

Bisaillon, C. (2020, March). Fake and real news dataset, Version 1. Retrieved November 12, 2020 from https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset.

Ahmed H, Traore I, Saad S. "Detecting opinion spams and fake news using text classification", Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018.

Ahmed H, Traore I, Saad S. (2017) "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham (pp. 127-138).

Craig Silverman, Buzzfeed Founding Editor, Canada, "This Analysis Shows how Viral Fake Election News Stories Outperformed Real News on Facebook", Buzzfeed News, November 2016 from https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook.