Fake News Classification

Using Machine Learning to Detect Fake News

Abdul Basit
George Mason University
Fairfax, VA
abasit2@gmu.edu

Jacob Shade
George Mason University
Fairfax, VA
jshade3@gmu.edu

ABSTRACT

Ever since the 2016 United States presidential election, the spreading of misinformation online has become a hot topic in both informal discussion and academia. The authenticity of every news source has come into question, and manually authenticating them one by one is inefficient. We should consider classifying fake news using modern classification algorithms, allowing for quick determination of a piece of news's authenticity.

KEYWORDS

Machine Learning, Fake News, KNN, Decision Tree, Classification, Supervised

ACM Reference format:

Abdul Basit, Jacob Shade. 2020. Fake News Classification: Using Machine Learning to Detect Fake News. In *Proceedings of ACM. AMC, New York, NY, USA, 3 Pages.*

1 INTRODUCTION

The spreading of misinformation has unfortunately increased exponentially in the digital age. The advent of the internet has given individuals new routes of access to large heaps of knowledge. People are able to keep up to date with almost all local and global events using the devices in their pockets. These new avenues to information have also made misinformation on current events, or “fake news”, much easier to spread.

Since the 2016 United States presidential election, the spread of this misinformation has come into the spotlight. Whether motivated by maliciousness or ignorance, untrustworthy sources have spread this fake news at rates exceeding that of which verified information is distributed. Governments, respected news agencies, and citizens alike have considered ways to mitigate the spreading of fake news, with one common recommendation being to flag pieces containing false information.

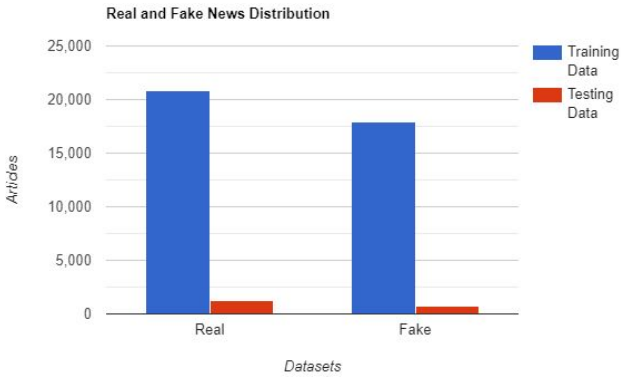
Manually flagging these news articles, websites, and more would be too time costly to do in any practical sense. As is the case with many repetitive tasks that would be too time costly for people, computers may be able to aid in the process.

Using machine learning to train a computer program to be able to classify any article’s text as either fake or real would speed up this process tremendously and was the motivation for our study. We will attempt to see if it is possible for classification algorithms to accomplish this task, and help in the war against fake news.

2 EXPERIMENT AND PROCEDURE

2.1 Data Sourcing

The training and testing datasets for this experiment came from two different Kaggle sources. The training dataset titled “Fake and real news dataset” consisted of 20826 true and 17903 fake news articles. The true news articles were found by scraping reuters.com articles. Fake news articles were collected from unreliable websites that were flagged by a fact-checking organization Politfact. This makes this dataset questionable because of the bias involved in the true category. It is not as reliable because the author assumed that all articles from one website were deemed true. Nonetheless, this was the biggest dataset available for training, so it should provide a good basis for training our models. Our testing dataset, “Source based Fake News Classification,” comes from data from websites that have been previously flagged containing fake news. This was



sourced from a variety of websites making this dataset more universally applicable to a real world scenario.

The training dataset was used to create a model for each of our different classification algorithms, while the test set was used to measure the accuracy of our model's predictions. We chose to use two different datasets sourced by different people for training and testing to get a better idea of how our models would perform in real world scenarios. This gives us a much better estimate than simply holding out a portion of a single dataset for testing as is done in cross-validation.

2.2 Classification Algorithms

Given that both our acquired datasets had labels associated with them already, supervised machine learning seemed favorable over unsupervised learning techniques. Unsupervised algorithms, such as k-means for clustering, would still have required some human interpretation of the results beyond simple measures of accuracy.

For classification algorithms, we chose to try k-nearest neighbors (KNN), decision trees, and random forests. These classifiers were chosen due to their differences in strengths and weaknesses, giving us a way to better understand which types of classifiers would be best for the problem at hand. These classifiers were also the ones we understood the best and were able to tune the most without making arbitrary decisions. KNN keeps track of the locations of each datapoint, news articles in our case, of our training set. When a new point requires classification, KNN assigns it to the majority class of the k nearest already classified points it keeps track of. This k value can be optimized using methods such as cross-validation. The decision tree classifier generates a tree which goes through a number of decisions to classify new data points based on its features. Most of the computation for this algorithm is done while building the tree and deciding what features, and what value of those features, to split on in what order. Random forests build multiple decision trees and merge them together to increase the stability of its predictions.

2.3 Text Processing and Document Representation

Natural language processing by computers requires the transformation of human readable speech into numerical values. This requires removing unnecessary information from the data and changing words into numbers.

We preprocessed both our datasets by removing numbers and punctuation from the text as they do not provide any useful information in terms of classification. We also removed stopwords, which are words that are very common in the english language and don't give us any idea of what type of text is being evaluated. Some removed stop words include: the, like, as, and is. We also chose to utilize lemmatization, which groups together words with similar meanings, such as change, changing, and changes, and replaces them with the base form of the word. This process differs from stemming, which simply chops off bits of the word to achieve a root that may not be a real word.

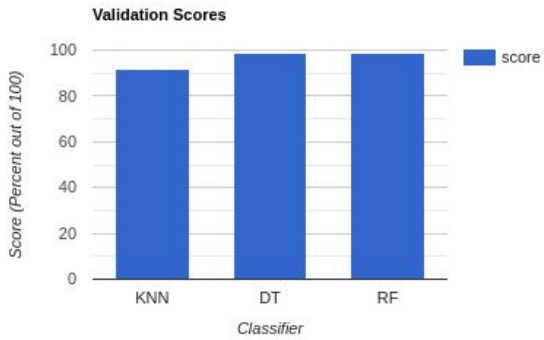
To represent our processed text in numerical form, we chose to utilize a TF-IDF representation. TF or term-frequency represents documents as vectors, where each vector's elements are the frequency of a term in the document. TF-IDF takes this representation and considers all documents inside the document set, giving less importance to words that appear frequently in many of the documents across the set. We chose this representation because we wanted to limit the impact that common words used in news articles would have on our classification.

3 RESULTS AND DISCUSSION

3.1 Training and Validation

In order to train our data, we used sklearn's libraries along with hyper tuning the parameters passed into each algorithm's function call. We found that in limiting the featureset, in both the KNN and tree algorithms, to a maximum of 5,000 for KNN and 10,000 features for the tree algorithms, our models performed the best in terms of validation score. This meant that for these models, 5,000 to 10,000 words could more effectively determine the classification of each news article.

The validation test results were fairly high for each algorithm. For KNN we were able to achieve a validation score of 92.3% with a k value of 4403. We used cross-validation to find the most optimal values for k. Both tree-based classifiers scored a 99% accuracy score on the validation set.

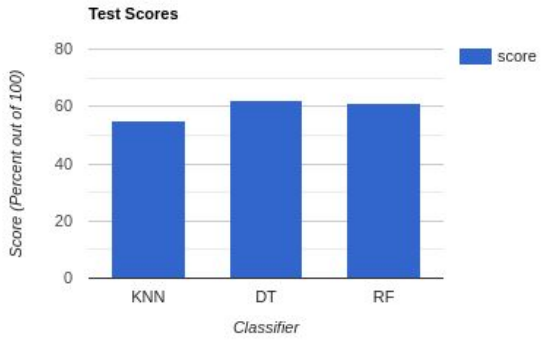


Although these scores look very promising, it should be noted that these scores are not test scores, but validation scores. These values were obtained by testing our model using data from the same set as our training data. To better gain an understanding of how our models perform on real data, we should observe the scores achieved by each on our test set.

3.2 Testing

As mentioned in section 2.1, our test set was sourced from another Kaggle dataset. We chose to do this to get a better understanding of how our models would perform if they were fed random news articles scraped from the web by any individual. The test set was provided by a different individual, and likely comes from a different distribution than our training set. Since the labels for this set were already provided, it made getting an idea of our models' performance on real data much easier. It should still be noted that although this test set is created by a different individual, there still may be a common bias shared between it and the training set, as both creators made them with a goal of classifying fake news in mind.

The test scores were significantly lower than our validation scores. KNN performed the worst, achieving an accuracy score of 55%. Our decision tree model scored 62%, while random forest received a max score of 61%. These scores suggest that of our chosen classifiers, tree-based algorithms performed the best with our dataset. Although these scores are better than random guessing, there is certainly room for improvement.



The large score difference between validation and testing suggest that our models were overfitted to our training data. This could have been mitigated by choosing a much larger training set. It should also be mentioned that the size disparity between the two sets may have also contributed to these lower scores. The test set was the main limiting factor in the number of features we were able to include in our final document vectors, as there were a lot less unique words in the test set than in our training set. We suggest to others attempting to create a model with the same goal to pick larger and more diverse training sets.

3.3 Ethics and Considerations

Whether or not fake news can be accurately detected by classifiers or not, the question still remains of whether it should be. We believe the answer to be yes, but with caution. The main ethical concern is one of bias. Since these models all require training data that is already labeled, the quality of that data and any bias in it will clearly be reflected in the models performance and classification of unlabeled data. Since humans are inherently biased, it proves to be very difficult to obtain completely unbiased training data, where all news pieces are fairly critiqued. For example, an article painting a particular politician or policy in a good light may be more likely to be fed to a model as fake news during its training. Given the highly polarized population, it may prove difficult to train models without this sort of bias. It may prove beneficial to have multiple individuals with varying political preferences and philosophies to provide training data.

Anyone interested in creating such a model should take extra care when selecting their training data. Training data should be unbiased as possible, large in size, and taken from many different sources. A metric for evaluating bias in the data, particularly subtle biases individuals recognize while reading or listening to news, should also be developed to evaluate said data.

4 CONCLUSIONS

Fake news has plagued our society in the modern age. The war on information has proven to be an important one, influencing elections, media, public perception, and more. Given the vast amount of news on the internet, it is impossible for individuals to go through all of it and determine whether each piece of news encountered is accurate or misinformation. Machine learning can greatly help in this pursuit by quickly classifying articles and other text based on the words present in them. Although the models we created only performed slightly better than random guessing, we believe with better and larger amounts of data, a model which could classify news with a high accuracy can be developed. We encourage others to work to develop such a model to aid in the fight over information. We also encourage the creation of the training data, keeping bias in mind.

A APPENDICES

A.1 Introduction

A.2 Experiment and Procedure

A.2.1 Data Sourcing

A.2.2 Classification Algorithms

A.2.3 Text Processing and Data Representation

A.3 Results and Discussion

A.3.1 Training and Validation

A.3.2 Testing

A.3.3 Ethics and Future Considerations

A.4 Conclusions

A.5 References

ACKNOWLEDGMENTS

This experiment in finding the legitimacy of news articles was only possible because of the authors of the datasets used. We thank Clément Bisaillon, Meg Risdal, and Ruchi Bhatia for their contribution by allowing their datasets to be publicly accessible.

REFERENCES

[1] Clément Bisaillon. 2020. Fake and real news dataset:Classifying the news.. https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

[2] Meg Risdal. 2016.Getting Real about Fake News:Text & metadata from fake & biased news sources around the web. https://www.kaggle.com/mrisdal/fake-news

[3] Ruchi Bhatia. 2020.Source based Fake News Classification:Classification of news by type and labels. https://www.kaggle.com/ruchi798/source-based-news-classification

Video Presentation Link: https://youtu.be/IQeZjNRObT4