Classifying News through Natural Language Processing

Diya Kaimal *

October 17, 2022

Abstract

Natural language processing works to help computers understand text as humans do. We use these techniques to identify an article as real or fake news through supervised learning and logistic regression.

1 Introduction

Machine Learning is the field of computer science that allows computers to learn from experience [JM15]. We consider several articles and their credibility. Our data contains roughly 40,000 articles ranging from March 2015 to February 2018 [ATS18]. Table 1 depicts fake news, giving the title, text, and date. A similar dataset depicts real news with the same information.

Title	Text	Date
Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn't wish all Americans ..	December 31, 2017
Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin ...	December 31, 2017
Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	December 30, 2017
Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	December 29, 2017

Table 1: Fake news in the data set

In this process, we will use the text of the article to determine whether it is real or fake, and so we drop the other data, removing titles and dates from our sets. Certain words will be common in real news and the same for fake news. Our model will help determine which word falls into which category.

In order to apply supervised learning techniques, we use information from many examples to predict [HTF09]. In this case, it is the credibility of other

*Advised by: Guillermo Goldsztein of the Georgia Institute of Technology

articles. The information given to the computer is referred to as the features, our text from the articles, and the prediction returned is the label, real or fake. We will divide our data in two sets: training and validation. The former will allow us to use most of the information to teach the model, and the latter will allow us to check our results, avoid overfitting, and choose the model with the best accuracy.

In section 2 of this article, we will go through preprocessing the text data, and converting it from words to numbers. Section 3 focuses on creating our features through feature engineering. In section 4, we will go over binary classification problems and logistic regression. Section 5 will go over our results and accuracy, and section 6 will end with a further analysis of the importance of this model.

2 Preprocessing the data

There is a great variety in the almost 40,000 articles being used; however, many contain unnecessary hyperlinks and other pieces of information that would hinder the efficiency of the process as the amount of data is large [FWS97]. An example of a sentence found in one article is:

> "The man tugging at his jacket to stop him is my favourite part. pic.twitter.com/M9824I4eCp"

This sentence contains both a link to twitter, and a picture link: neither are helpful to our goal, so we remove them. This is one step of our preprocessing process. Before the data can be used by the model, it must be transformed into numbers that will be understood. We will go through several preprocessing steps in order to only retain the text elements that will have the greatest effect on identifying the credibility of an article. This includes removing punctuation, most common words, and making everything lowercase. When the words are preprocessed and put into a list, the same quote looks like:

> "'man', 'tugging', 'jacket', 'stop', 'favourite', 'part'"

The next step would be stemming the data by removing suffixes and prefixes. When we begin to look at frequency in the next section, this will allow us to consider words like "tugging" and "tugged" as the same by referring to both as "tug". These techniques are performed to all articles in the data set.

3 Calculating frequencies

In order to create our two features, we consider the frequency of real or fake words. The real frequency signifies the number of times a word appears in a credible article, and the opposite for the fake frequency. To picture this, Table 2 shows two sentences that are not from our dataset but will serve as an example.

The real frequency of the word *she* in this example would be 2 because it appears twice in the real news sentence. However, it would have a fake frequency

Real	Fake
she had multiple jobs so she could make ends meet	she continued to meet him at future jobs

Table 2: Sample sentences

of 1 because it only appears in the fake sentence once. Following this pattern, we see that *meet* is in both sentences so it has a real and fake frequency of 1 in both categories. Table 3 below lists the frequencies of all words in the two sentences. In our dataset, each article has many sentences causing the frequencies to build up to a much larger scale than we see here.

Word	Real Frequency	Fake Frequency	Word	Real Frequency	Fake Frequency
she	2	1	ends	1	0
had	1	0	meet	1	1
multiple	1	0	continued	0	1
jobs	1	1	to	0	1
so	1	0	him	0	1
could	1	0	at	0	1
make	1	0	future	0	1

Table 3: Real and fake frequencies of given sentences

The next step is referred to as feature engineering, or creating new features that are not in the given set. We create two new features: real and fake. The real feature for each example, or article, is created by going through every word in the text and adding up the real frequencies as depicted in the previous example. In the sentence above, *"she continued to meet him at future jobs"*, we would add $2 + 0 + 0 + 0 + 1 + 0 + 0 + 0 = 3$ to get the real feature.

The fake feature is the same: adding up the fake frequencies in every word of every text. In the sentence above, *"she continued to meet him at future jobs"*, we would add $1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 8$ to get the fake news feature. These numbers become our final features and the label, real or fake, is substituted for 1 or 0 respectively. We can now proceed with the model.

4 The model and techniques

In datasets involving categories, there are two types of problems: multi classification and binary classification. Our labels can either be a 1 or 0, indicating that there are only two possible outcomes for this problem. This is referred to as a binary classification problem. In context with our situation, a label of 1 would represent a credible article. A label of 0 would represent a non-credible article. This processing of relabeling is one-hot encoding.

The notation for binary classification problems typically follows the following format: $\hat{y} = y(x_1, x_2)$. x_1 is the first feature, in our case, the real news feature. x_2 is the second feature, or our fake news feature. $\hat{y}$ refers to the label produced by the computer which will be a value between 0 and 1. If it is closer to 1, that indicates that the article is real news, and a value close to 0 would indicate an article with fake news.

Logistic regression can be used for binary classification problems. It utilizes the sigmoid function: $\sigma(x) = \frac{1}{1+e^{-x}}$ to transform from linear regression [BCD16]. The function can also be represented through the graph displayed below.

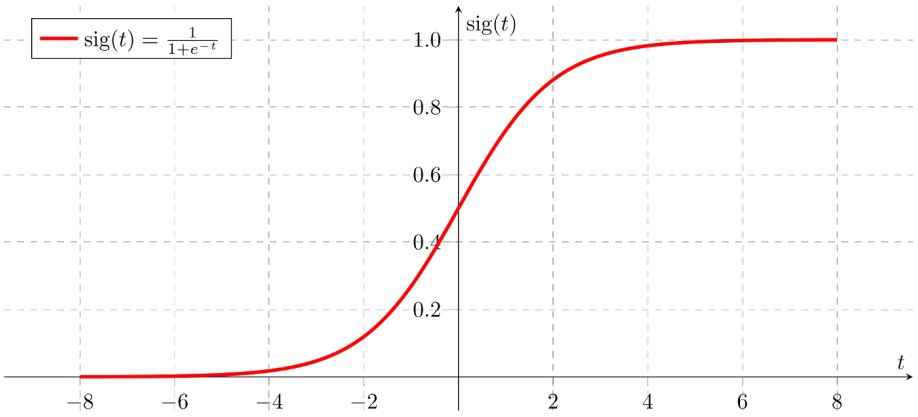

Figure 1: Graph of sigmoid function

The sigmoid function is continuous and increasing everywhere. Notice how $\sigma(0) = 0.5$ in both the expression and graph. A key component of this graph is the range which is (0,1). Our y value represents the label which as stated above, also ranges from 0 to 1. This correlation is what allows us to use the sigmoid function for binary classification problems. The full formula for logistic regression is as follows:

$$\hat{y} = \sigma(w_1x_1 + w_2x_2 + ... + w_kx_k + b)$$

where k represents the number of features per example. Our example has two features, so the formula used will be:

$$\hat{y} = \sigma(w_1x_1 + w_2x_2 + b)$$

w_1, w_2, w_k, and b are all examples of parameters. These values are chosen by the model to minimize the cross-entropy error on the training set. Mean binary cross-entropy error can be calculated as:

$$J = \frac{-1}{n}\Sigma_{i=1}^{n}(y_i)log(\hat{y}_i) + (1 - y_i)log(1 - \hat{y}_i)$$

y_i must either be 0 or 1 due to the properties of binary classification as mentioned above. This means that one of the two terms in the given formula will cancel out. Dealing with one term allows us to see that as $\hat{y}_i$ gets closer to y_i, the error goes down. For example, $-(1)log(.24)$ is much greater than $-(1)log(.64)$. Additionally, if $\hat{y}_i = y_i$, there is no error as $-(1)log(1) = 0$.

In setting up the rest of the model, we split the data into two sets: training and validation. The training set focuses on teaching the model what to do and what to look for. The validation set checks the model by computing the error on data the model has not seen yet. This allows us to stop any overfitting. In this project we used 75% of our data to train the model and the other 25% to test the model. A higher amount of training data gives the best amount of information to the software. Using common libraries found with Python such as Keras, we set up our model and test until we get the highest accuracy on both the training and validation set.

5 Results and accuracy

The data used provided almost 40,000 different articles and their credibility. We have created features for every article in the data set. Figure 2 puts these on a scatter plot to compare the divide. News marked credible are represented by an orange dot and news marked fake are represented by a blue dot. Notice that the feature values range up to 1×10^7 due to the sheer amount of words in each article which means high real and fake features for most articles.

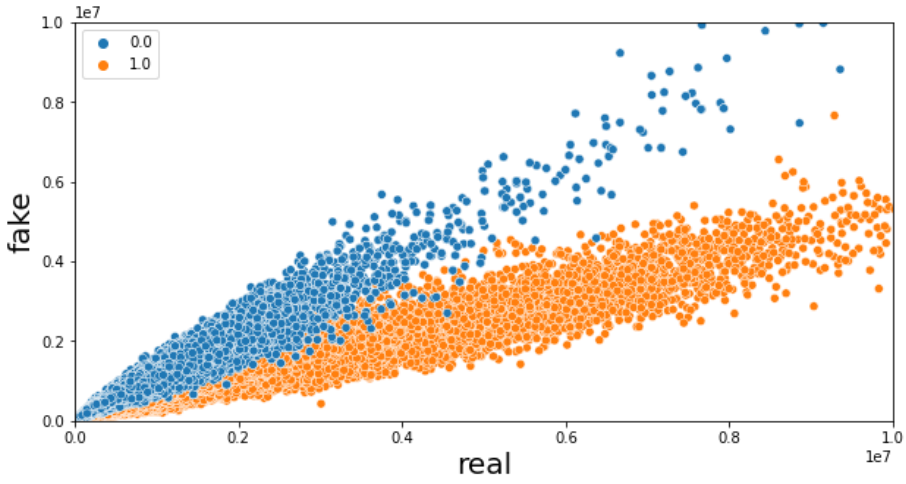

Figure 2: Scatter plot of all articles in the data set with their features

While the data closer to the origin is much more clumped together, there still does appear to be a clear divide. Using our model focusing on logistic regression, we achieve an accuracy of 96% on both the training and validation

set. Because our data sets are fairly balanced, this is a good estimation of our model's accuracy. The divide can be seen clearer using a decision boundary which will plot a straight line. This is shown in Figure 3. One side of the line will be the model's prediction of fake news and the other will be real news prediction.

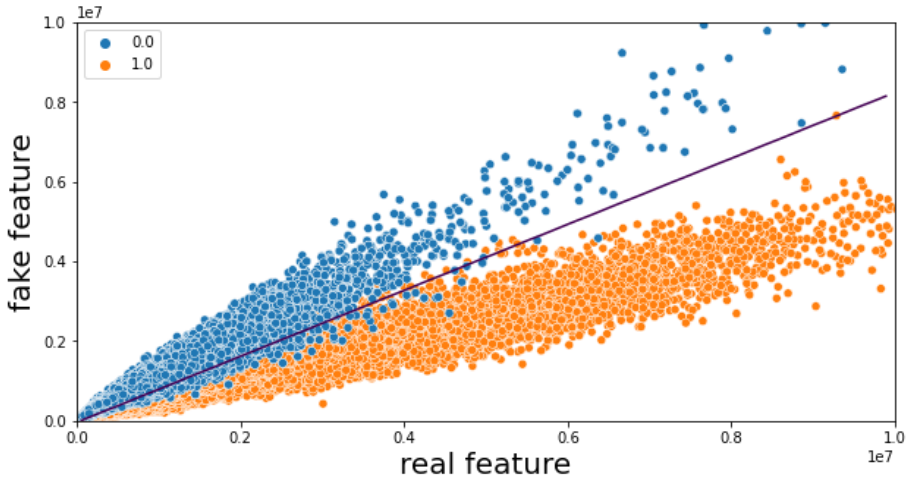

Figure 3: Scatterplot with decision boundary

There are a few clear errors as they can be found on the other side of the decision boundary; however, overall, the model is fairly accurate.

6 Conclusion

In this paper, we explained how natural language processing and feature engineering can be used in supervised learning techniques. We talked about the importance of logistic regression and binary classification. One of the challenges facing society today is the sheer amount of fake news going around. Our model was able to identify fake news with an accuracy of 96% on both the training and validation sets. With this model, we can work towards preventing the spread and maintaining the credibility of news.

References

[ATS18] H. Ahmed, I. Traore, and S. Saad. Detecting opinion spams and fake news using text classification. *Journal of Security and Privacy*, 1(1), 2018.

[BCD16] R. Buyya, R. N. Calheiros, and A. V. Dastjerdi. *Big data: principles and paradigms*. Elsevier/Morgan Kaufmann, 2016.

[FWS97] W. M. Famili, A.and Shen, R. Weber, and E. Simoudis. Data preprocessing and intelligent data analysis. *Intelligent data analysis*, 1(1), 1997.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman. *Overview of supervised learning*. Springer, 2009.

[JM15] M.I. Jordan and T.M. Mitchell. Machine learning: trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.