Fake News Detection



**Team 5**

Manish Ranjan

Rachana Sooraj

Raditya Fahritama

SWENG 545 - Data mining

Fall II, 2021

Introduction

The Internet is an incredible resource for news and information, but unfortunately not everything online is trustworthy. **Fake news** is any article or video containing untrue information disguised as a credible news source. While fake news is not unique to the Internet, it has recently become a big problem in today’s digital world.

Fake news typically comes from sites that specialize in bogus or sensationalized stories. It tends to use **provocative headlines**, like "Celebrity endorses not brushing teeth" or "Politician selling toxic waste on the black market". These headlines can seem suspicious or even unbelievable to the point of being silly, making it tempting to think of fake news as harmless.

In recent years, however, fake news has been responsible for a great deal of **misinformation** because more and more people have begun consuming and believing these articles without bothering to fact check or even read beyond the headlines. This acceptance of incorrect information has led to confusion, panic, and an inability to discuss the actual facts surrounding current events.

Problem Description

The problem starts when fake news often went viral and started to make readers uncomfortable with its content. Fake news generally is designed to shock and fool people. And people usually re-shared the news without checking whether the news is real or not. With that, we surely want to know which news is real and which news is fake. What are things that differentiate between fake and real news. And also, we want to know what classification method is the best for detecting fake news.

Dataset

We use data that we found from Kaggle for training our classifier. The data contains 5 columns and 20387 rows. The data consists of title of the news, author, the news itself, and label that explains whether that news is reliable or unreliable (0 for reliable, 1 for unreliable). With 10387 total reliable news and 10413 total of unreliable news. Total size for this train dataset is 99 MB.

Here are 20 words that have the highest importance based on 175 sample texts that is computed in the TF-IDF vector from the dataset:

Figure 1. Word Relevance Rank in the news text.

From this chart, we can see that the article of the dataset is mainly about political issues that happen in the United States.

The proportion of real and fake label is 50.1% to 49.9%. Almost 50 to 50. Meaning that the Distribution of reliable and unreliable news is equally distributed.

Chart, bar chart

Description automatically generatedChart, pie chart

Description automatically generated

Figure 2. Proportion of Real and Fake Figure 3. Total of Real and Fake

The dataset contains ‘author’ attribute. Containing who is responsible for creating the article. We calculate the article written by each author, and sort them in descending fashion. Here are the top 10 authors who created the most article from our dataset.

Chart, bar chart

Description automatically generated

Figure 4. Top 10 Author for the Articles

We also made charts of the frequent occurring unigram, bigram, and trigram. And by looking to the charts, we can see that the most occurring words generally are US’ political topics. Mainly discussing about presidential issues, election, national security, and war.

Chart

Description automatically generated

Figure 5. Frequent Unigram words of the Dataset

Chart, funnel chart

Description automatically generated

Figure 6. Frequent Bigram Words of the Dataset

Chart

Description automatically generated

Figure 7. Frequent Trigram words of the Dataset

Data Preprocessing

First, we look at our dataset and see if the data contains missing value. On our dataset, the null value for ‘id’ attribute is 0, 558 for ‘title’ attribute. 1957 for ‘author’ attribute, 39 for ‘text’ attribute, and 0 for label attribute. For dealing with missing value, we simply fill the row with empty string. We try to retain the index number of the original dataset. So that not even a single row is dropped from the dataset.

After that, we stem the title of the data. Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. We also do the process of removing stopwords. This process helps to remove unnecessary words that have no meaning to the computation (such as “the”, “a”, “an”, “in”). And we remove the numeric from the words and keeping them in lower case to prevent ambiguity.

**TF-IDF**

We apply vectorizer to the preprocessed titles. This transforms the title list to vector array. And applying TF-IDF vectorial representation to the computation. TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents.

This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.

TF-IDF for a word in a document is calculated by multiplying two different metrics:

-The term frequency of a word in a document. There are several ways of calculating this frequency, with the simplest being a raw count of instances a word appears in a document. Then, there are ways to adjust the frequency, by length of a document, or by the raw frequency of the most frequent word in a document.

-The inverse document frequency of the word across a set of documents. This means, how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. This metric can be calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.

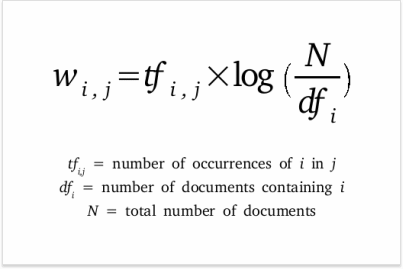


Figure 8. TF-IDF Formula

Then, we split the vectors into train and test vectors using train test split. The train vectors are used to train the machine to know what is considered fake news and what is considered real news based on the label. And after that, the test vectors are used for the machine to make predictions. The machine predicts each of the test vector then compare it to the label of it. The accuracy is ranked by how many the machine got right and got wrong.

Table

Description automatically generated

Figure 9. Example of TF-IDF Output

Methodology

We tried 4 different classifier methods that are mostly used in Natural Language Processing. And then chose which one is the best to be fitting in for this project, the algorithms are Multinomial Naïve Bayes. Passive Aggressive Classifier, Random Forest, Support Vector Machine, and Logistic Regression.

**Multinomial Naïve Bayes**

Multinomial Naïve Bayes is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output. Naive Bayes is a powerful algorithm that is used for text data analysis and with problems with multiple classes. Bayes theorem, formulated by Thomas Bayes, calculates the probability of an event occurring based on the prior knowledge of conditions related to an event.

**Passive Aggressive Classifier**

Passive-Aggressive algorithms are somewhat like a Perceptron model, in the sense that they do not require a learning rate. However, they do include a regularization parameter. Passive-Aggressive algorithms are called so because:

Passive: If the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model.

Aggressive: If the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.

It is one of the few ‘online-learning algorithms ‘. In online machine learning algorithms, the input data comes in sequential order and the machine learning model is updated step-by-step, as opposed to batch learning, where the entire training dataset is used at once.

**Random Forest**

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems.

**Logistic Regression**

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value).

**Support Vector Machine**

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be employed for both classification and regression purposes. SVMs are more commonly used in classification problems. SVMs are based on the idea of finding a hyperplane that best divides a dataset into two classes. Support vectors are the data points nearest to the hyperplane, the points of a data set that, if removed, would alter the position of the dividing hyperplane. Because of this, they can be considered the critical elements of a data set.

Results

After applying the classifiers to the vectors, we got the results of the accuracy score of the actual label and the prediction. We also create the confusion matrixes for all the classifier to visualize the true and predicted news count. Here are the results:

**Multinomial Naïve Bayes**

Multinomial Naïve bayes predict the train data at accuracy rate of 0.907(90.7%) and 0.864(86.4%) for the unlabeled test data.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 10. Accuracy Score of MultinomialNB

A screenshot of a computer

Description automatically generated with medium confidence

Figure 11. Classification Report of the MultinomialNB

The confusion matrix shows that the classifier got 2288 True fake, 3105 true real, 37 false fake, and 810 false real. The classifier got more error on the fake news much more than the real one.

Chart, treemap chart

Description automatically generated

Figure 12. Multinomial Naïve Bayes Confusion Matrix

**Passive Aggressive Classifier**

Passive Aggressive Classifier predict the train data at accuracy rate of 1(100%) and 0.96(96%) for the unlabeled test data.

Text

Description automatically generated

Figure 13. Accuracy Score of Passive Aggressive Classifier

Calendar

Description automatically generated

Figure 14. Classification Report of Passive Aggressive Classifier

The confusion Matrix for Passive Aggressive Classifier shows that the classifier got 2968 True Fake, 3023 True Real, 130 False Real, 119 False Fake. The error distribution of Real and Fake for this classifier is more equal.

Chart, treemap chart

Description automatically generated

Figure 15. Passive Aggressive Classifier Confusion Matrix

**Logistic Regression**

Logistic Regression predict the train data at accuracy rate of 0.974(97.4%) and 0.946(94.6%) for the unlabeled test data.

Text

Description automatically generated

Figure 16. Accuracy score of Logistic Regression

A screenshot of a computer

Description automatically generated with medium confidence

Figure 17. Classification Report of Logistic Regression

The confusion Matrix for Logistic Regression Shows that the classifier got 2969 true fake, 2931 True real, 129 False real, 211 False Fake. The error distribution is more on the real news with the difference of 82 than the fake news

Chart, treemap chart

Description automatically generated

Figure 18. Logistic Regression Confusion Matrix

**Random Forest**

Random Forest predict the train data at accuracy rate of 1(100%) and 0.927(92.7%) for the unlabeled test data.

Text

Description automatically generated

Figure 19. Accuracy Score of Random Forest

A screenshot of a computer

Description automatically generated with medium confidence

Figure 20. Classification Report of Random Forest

The confusion matrix for Random Forest shows that the classifier got 2796 True Fake, 2969 True Real, 302 False Real, 173 False Fake. The error distribution is more on the Fake news with difference of 129 than the real news.

Chart, treemap chart

Description automatically generated

Figure 21. Random Forest Confusion Matrix

**Support Vector Machine**

Support Vector Machine predict the train data at accuracy rate of 0.999(99.9%) and 0.956(95.6%) for the unlabeled test data.

Text

Description automatically generated

Figure 22. Accuracy Score of Support Vector Machine.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 23. Classification Report of Support Vector Machine

The confusion matrix for Support Vector Machine shows that the classifier got 2988 true fake, 2979 true real, 110 false real, 163 false fakes. The error distribution for this algorithm is approximately equal with more on the real news.

Chart, treemap chart

Description automatically generated

Figure 24. Support Vector Machine Confusion Matrix

**Comparison**

We created some comparison to see how the accuracy scores compare to each other. As it is shown in figure 21, both passive aggressive Classifier and Support Vector Machine got 96% of accuracy. But when we try to run support vector machine on our model, the algorithm took quite some time. To the point that it is too long than the passive aggressive classifier. The run time ranks of these 5 algorithms are:

1. Passive Aggressive Classifier
2. Multinomial Naïve bayes
3. Logistic Regression
4. Random Forest
5. Support Vector Machine

Passive aggressive classifier is much faster than support vector machine. So, for our model specifically, it is clear that passive aggressive classifier is better for predicting text because it is designed for huge amount of data. And the classifier itself is updated regularly, so it learns better about text validity than other classifiers.

Text

Description automatically generated

Figure 25. All result Comparison

**Wordcloud**

We created wordcloud to see if specific words occur more often on real and fake news. Here are the results:

Text

Description automatically generated

Figure 26. Fake News Wordcloud

From this Fake News Wordcloud, we could see that the words that appear often are generally strong words that refers to politics and war. We can see that political and war issues are topics that is faked the most. It is totally different from the real news wordcloud that as we can see. The occurrences of frequent words are more random and not referring to any specific topic.

Text

Description automatically generated

Figure 27. Real News Wordcloud

We created wordcloud based on the title and here are what we got:

A picture containing text, newspaper, screenshot

Description automatically generatedA picture containing text, newspaper

Description automatically generated

Figure 28. Fake title wordcloud Figure 29. Real title wordcloud

It seems that the words distribution from the real news title are more even and neutral than the fake news. Fake news tends to be focusing to political issue and addressing Political figure more often than reliable news. This is maybe because due to the fact that fake news purpose is to shock people and to obtain readers with clickbait. Fake news often creates sensational title.

Limitation

For our model, the highest accuracy we can get when predicting news is 96%. We still haven’t touched the 100% accuracy. Maybe in the future, the way to improve our model is that we will use different dataset for training the model to see if our accuracy score will increase. And also, our model is now only limited to labeled dataset. Because the dataset that we use to train the model is mainly about political issue, we think that our model is not good enough to predict other topics. Then, the way that we think we can do to improve our model is that we will do better data preprocessing steps to assure that our dataset is as clean as possible.

References

- GitHub. 2021. *Fake-News-Classsification/notebook97bb433428.ipynb at main · muskan9698/Fake-News-Classsification*. [online] Available at: <https://github.com/muskan9698/Fake-News-Classsification/blob/main/notebook97bb433428.ipynb> [Accessed 5 December 2021].

- GeeksforGeeks. 2021. *Python | Stemming words with NLTK - GeeksforGeeks*. [online] Available at: <https://www.geeksforgeeks.org/python-stemming-words-with-nltk/> [Accessed 5 December 2021].

- GCFGlobal.org. 2021. *Digital Media Literacy: What is Fake News?*. [online] Available at: <https://edu.gcfglobal.org/en/digital-media-literacy/what-is-fake-news/1/> [Accessed 5 December 2021].

**-** 2021. [online] Available at: <https://monkeylearn.com/blog/what-is-tf-idf/> [Accessed 5 December 2021]

-upGrad blog. 2021. *Multinomial Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2021 | upGrad blog*. [online] Available at: <https://www.upgrad.com/blog/multinomial-naive-bayes-explained/> [Accessed 5 December 2021].

-GeeksforGeeks. 2021. *Passive Aggressive Classifiers - GeeksforGeeks*. [online] Available at: <https://www.geeksforgeeks.org/passive-aggressive-classifiers/> [Accessed 5 December 2021].

- Built In. 2021. *A Complete Guide to the Random Forest Algorithm*. [online] Available at: <https://builtin.com/data-science/random-forest-algorithm> [Accessed 6 December 2021].

-En.wikipedia.org. 2021. *Logistic regression - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Logistic\_regression> [Accessed 6 December 2021].