**Identifying Fake and Real News**

Capstone project 2- Milestone Report-2

By: Muluemebet Ayalew

June, 2020

1. **Introduction**

In this computerized era spreading of information is very fast. If the information is not real, it could create a lot of damage. The damage could be for both individuals or groups on which the news was made or for the people who trust and follow the fake news. In the first case, it can create social, psychological, reputation and career damages on individuals. For example, misleading news about a political candidate during election season, can make the best candidate lose his/her vote. It could also make companies lose their customers and trustworthiness. On the other hand, people who follow and trust wrong news can be cheated to act wrongly. Thus, it is very important to identify fake news to protect ourselves from those damages.

The objective of this project is thus to identify fake and real news based on its content using machine learning techniques. The result can help news agencies to identify fake news, to falsify it, broadcast the truth and play a role for society. This saves the community from confusion, wrong decision, misinterpretation and wrong judgment coming as a result of fake news. From the news agency's perspective, unless they fight fake news, at some point they might lose their audience.

The data used for this project is available on Kaggle[[1]](#footnote-0) in two data frames; fake and real news. Each dataset contains the news title, text, subject and date at which the article was posted. The fake news dataset contains 23,481 articles(rows) and the real news contains 21,417 articles(rows).

The approach to perform the analysis and modeling was framed in the methodology section below.

1. **Methodology**

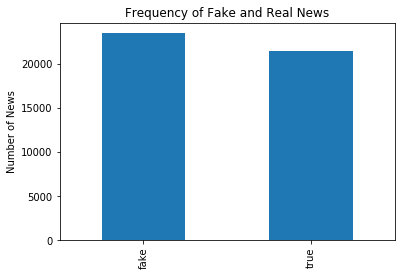
The data was explored to observe the difference between fake and real news. Before merging the two datasets, a new column “is\_fake” was added to each dataset to distinguish between fake and real news. Then data cleaning and feature engineering was done. Since the content of the news article was used to make predictions, Natural Language Processing (NLP) techniques and tools were used. Thus, the title and the text of the news were the main focus and were used as predictors. Data cleaning tasks such as removal of emoji, url links, html tags, punctuations, special characters, numbers and stopwords were performed. Stopwords are words commonly available in any text such as ‘the’, ‘of” and so on. Other data cleaning and preprocessing techniques such as tokenization, stemming/lemmatization, phrase modeling, word embedding, bags of words, term frequency inverse document frequency (tfidf) and word embedding were performed.

Once the data was prepared and features were selected, classification techniques such as Naive Bayes and logistic regression was used to fit the training data (70% of randomly selected data points from the merged data). And then the performance of the model was evaluated using test data (30% of the merged data).

Python libraries such as pandas, scikit-learn, nltk, gensim, wordcloud, numpy, matplotlib were used to explore, prepare, model and test the data. At the end of the project, the code, written report and a slide deck will be delivered.

1. **Data Exploration**

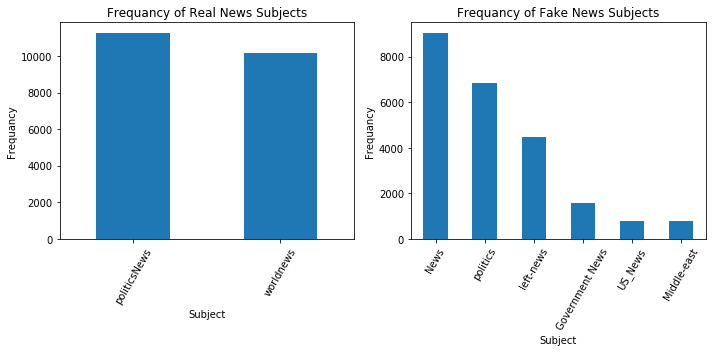
The data of both fake and real news were explored to observe the difference between the two datasets. As we can see in Figure 1, the two data sets are more or less balanced.



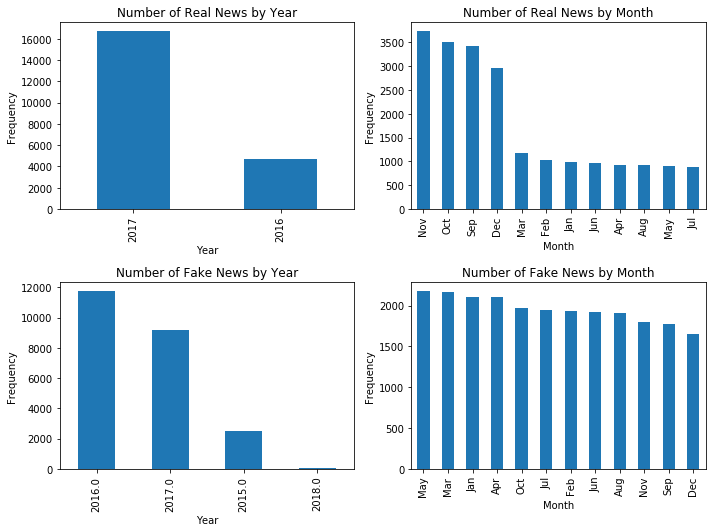
*Figure 1 Fake and real news fequency*

The subject of the news, the year and month on which most news were posted were observed and plotted (see Figure 2). Fake news has six types of subjects whereas true news has two subjects of politics news and world news.

The year span for the fake news was from 2015 to 2018 whereas the real data had news for 2016 and 2017. Larger numbers of real news were reported in September, October, November and December. The number of fake news articles were uniform by month (see Figure 3).



*Figure 2 News subject for fake and real news*



*Figure 3 Distribution news by year and month*

The frequency of emoji, url and other symbols in the title and text of the two news types were explored. Table 1 shows the number of news with emoji symbols, url links, special characters, html tags, empty text and digits on their title or text for both fake and real news. There were a larger number of url links (3356) in fake news text than the text of real news (41). Fake news text had also a larger number of tags, html, empty text and digits.

*Table 1 Number of news with specified pattern (symbols/tags, digit) or empty space for fake and real data*

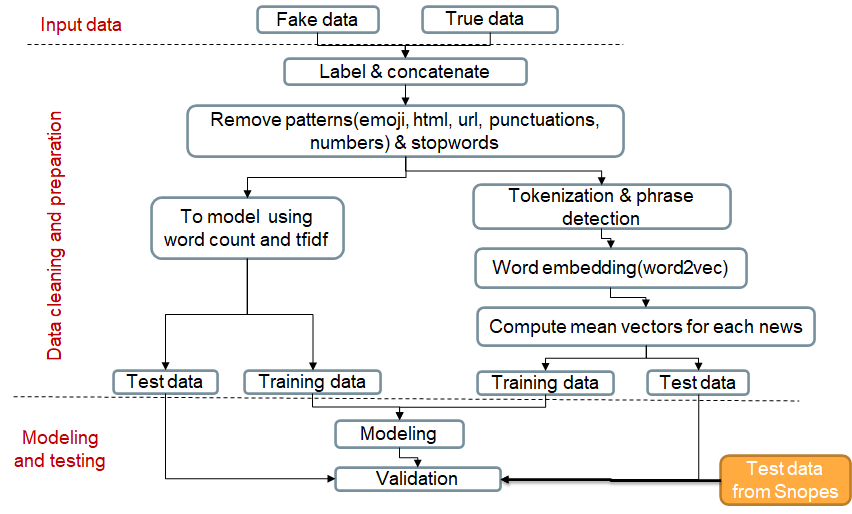
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pattern** | **Fake news** | | **True news** | |
| **Title** | **Text** | **Title** | **Text** |
| Emoji | 1 | 0 | 0 | 3(➡️, ☑️, 'ツ') |
| Url | 9 | 3356 | 0 | 41 |
| Tag((#,@,&) | 862 | 7682 | 25 | 1171 |
| Html | 0 | 79 | 0 | 8 |
| Empty (no text) | 0 | 626 | 0 | 1 |
| Digit | 3008 | 18945 | 1799 | 17309 |

1. **Data cleaning and data preparation**

After exploring the data and observing the characteristics of fake and real news, the two datasets were merged into one and cleaned. Emoji characters, url links and words containing numbers might not clearly reflect the content of the text and were removed. A function was defined to detect and remove the aforementioned patterns from the text. Special characters and stopwords were also removed. The text was also converted into lowercase in order to avoid repetition and to build case insensitive features. News without text (with empty space) was automatically ignored by the system. However, there is a possibility to replace it by a very unique word to represent it as empty and observe if it has impact on model prediction.

The data was prepared in two ways: the first was to use the cleaned data to make predictions using word count and tfidf , and the second was to use phrase detection and word embedding techniques to compute feature vectors and then making predictions based on those vectors. In the second approach, the cleaned data was tokenized and the common bigram phrases were detected using phrase detection techniques from the gensim module. Then, instead of using the pre-trained word embedding models which can be loaded using spacy or gensim modules, a custom word embedding model using word2vec was trained by the data at hand. Using this technique, each word can be represented as a set of numbers or vectors. We then computed the average of word vectors for each news article and this was used as input for classification. The overall data cleaning and preparation step is summarized in Figure 4.

Each of the prepared data was then randomly split into training (70%) and test data (30%) to train the model and then to test its generalizability for a new data set.



*Figure 4 Summary of data cleaning, preparation and modeling steps*

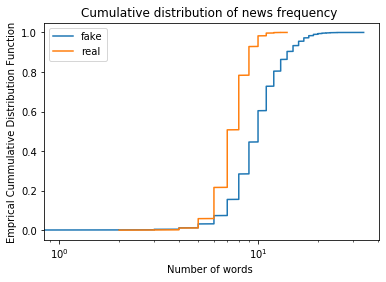
1. **Statistical Test: Is the length of the true news title shorter than fake news title?**

After exploring the length of both the title and the text of the news, we found that the title of fake news is longer than the title of real news even after cleaning the emoji, url, html tag, and digits (see Figure 5). The vertical axis represents the empirical cumulative distribution function and the horizontal axis represents the number of words appearing below a certain percent of the news in the dataset. The graph is interpreted as if for example 0.8 (80%) in the vertical axis means 80% of fake titles have less than 12 words (the corresponding horizontal coordinate) whereas 80% of true titles have less than 9 words. This raises a question about whether the length difference is statistically significant or not.

Thus, the hypothesis is defined as follow:

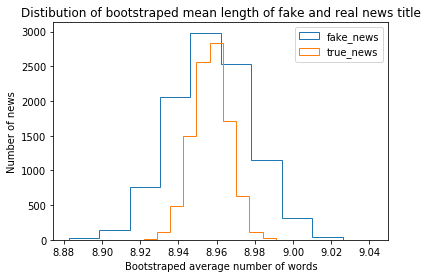
*The null hypothesis: The titles of fake and real news have the same word length.*

*Alternative hypothesis: Real news title has shorter word length.*

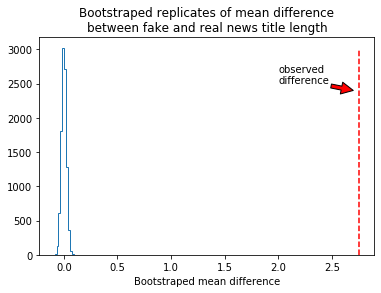
****

*Figure 5 Frequency of news below a certain number of words*

Once the hypothesis was defined, a bootstrapped hypothesis test was conducted with 5% rejection error (significance level). To perform this hypothesis test, the two datasets had to be shifted to have the same mean and then 10,000 bootstrap samples were drawn for both fake and real news separately. The 10,000 bootstrap replicates in this case the mean (see Figure 6) for each news type and the mean difference was computed(Figure 7). Then, the proportion that the bootstrapped mean difference is greater than the observed difference was computed. This value is called the p- value. The computed p-value was zero. Which means all of the bootstrapped mean differences were below the observed difference.

****

*Figure 6 Distribution of bootstrapped mean number of words*



*Figure 7 Distribution of bootstrapped mean difference*

***Conclusion from the statistical test :***

The p- value (0.0) is less than the pre specified significance level (0.05). Therefore, the mean length of fake news titles is statistically different from the mean length of real news titles. We can conclude that on average fake news titles are longer than real news titles.

**7)** **Modeling and Testing**

Different scenarios were made to make modeling. The first was to train the title and text of the news separately without cleaning and after cleaning using word count or tfidf. This returned a total of eight models using a single machine learning method. However, Multinomial Naive Bayes and Logistic regression methods were used to make modeling, resulted in a total of sixteen models with these scenarios. The second scenario was based on word vectors derived from word embedding technique. In this case two models using logistic regression were fitted using the title and text of the news separately. Naive Bayes was not used here because the vectors have negative values and it is not appropriate for those type of feature vectors. Overall a total of eighteen models were fitted and their accuracies for the test data set were computed (Table 2).

*Table 2 The accuracy of models fitted with different scenarios*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of Models** | | | **Machine learning method** | |
| **Data**  **preparation** | **Predictor** | **Vectorizer** | **Multinomial Naive Bayes** | **Logistic Regression** |
| Not cleaned | Title | Word count | 0.938 | 0.946 |
| Tfidf | 0.934 | 0.941 |
| Text | Word count | 0.950 | 0.996 |
| Tfidf | 0.937 | 0.984 |
| Cleaned | Title | Word count | 0.937 | 0.944 |
| Tfidf | 0.934 | 0.938 |
| Text | Word count | 0.956 | 0.997 |
| Tfidf | 0.947 | 0.985 |
| Word2Vec | Title | Mean vectors | - | 0.935 |
| Text | Mean vectors | - | 0.98 |

**Testing using external data**

In addition to the test data split before modeling, a new data from Snopes was extracted to validate the model. Snopes is a fact-checking website. It has news rated as false, true, mixed and so on. To test the model, the titles from the 2017 news archive were collected from Snopes[[2]](#footnote-1). This is because the data at hand was from 2015 to 2018. The current news might not be a good test as news in 2020 might be dominated by current issues such as covid-19. Thus, the older news was considered. Table 3 shows the title of the news extracted from Snopes archive their assigned id.

*Table 3 News titles extracted from Snopes and their assigned id*

|  |  |
| --- | --- |
| **News Id** | **Title of the news** |
| 1 | Is This James Earl Jones Dressed as Darth Vad |
| 2 | David Rockefeller's Sixth Heart Transplant Successful at Age 99 |
| 3 | Did Bloomington Police Di scover Over 200 Penises During Raid at a Mortician's Home? |
| 4 | Is the Trump Administration Price Gouging Puerto Rico Evacuees and Seizing Passports? |
| 5 | 2017 Tainted Halloween Candy Reports |
| 6 | Did President Trump Say Pedophiles Will Get the Death Penalty? |
| 7 | Michelle Obama Never Placed Her Hand Over Her Heart During the National Anthem? |
| 8 | Katy Perry Reveals Penchant for Cannibalism? |
| 9 | Is a Virginia Church Ripping Out an 'Offensive' George Washington Plaque? |
| 10 | Did Trump Retweet a Cartoon of a Train Hitting a CNN Reporter? |

As we can see in Table 2, models trained based on the title of the news had less accuracy for test data than models based on text of the news. The same phenomenon was observed for test data obtained from Snopes. For demonstration purpose only the accuracy of better performing six models and their prediction is summarized in Table 4. The first four models in the table were trained based on the text of the news either by using word count or tfidf. These models have better accuracy than the latter two. The last two models were trained by word vectors derived from word embedding models(word2vec). The misclassification labels predicted by the models are highlighted and crossed on the table. For consistency and modeling purposes, the ratings given by Snopes such as “mixed” and “mostly false” were considered fake news.

*Table 4 Prediction and accuracy of different models for news titles extracted from Snopes*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Model type** | | | | | |
|  |  | **Logistic Regression** | | **Multinomial Naïve Bayes** | | **Logistic Regression** | |
| **News**  **Id** | **Rate**  **by Snopes** | **Word count of text predictor** | **Tfidf of text predictor** | **Word count of text predictor** | **Tfidf of text**  **Predictor** | **Word2vec of title predictor** | **Word2vec of text**  **Predictor** |
| 1 | fake | fake | fake | fake | fake | fake | fake |
| 2 | fake | fake | fake | fake | fake | fake | TRUE |
| 3 | fake | fake | fake | fake | fake | TRUE | fake |
| 4 | fake | fake | fake | TRUE | TRUE | TRUE | TRUE |
| 5 | fake | fake | fake | fake | fake | fake | fake |
| 6 | mixed | fake | fake | fake | fake | fake | fake |
| 7 | fake | fake | fake | fake | fake | fake | fake |
| 8 | fake | fake | fake | fake | fake | fake | fake |
| 9 | mostly\_false | fake | fake | fake | fake | fake | fake |
| 10 | TRUE | fake | fake | fake | fake | fake | fake |
| **Accuracy** | | **0.9** | **0.9** | **0.8** | **0.8** | **0.7** | **0.7** |

**8)Conclusion**

The cleaned data performed better than the none cleaned data. Prediction based on the text of the article outperformed in terms of accuracy than based on just the title of the news. Generally, predictions based on word counts have slightly higher accuracy than that of tfidf. In all of the cases, the accuracies of logistic regression models were higher than that of multinomial naive bayes. Especially the difference is larger for models trained based on the text of the news articles. The two models based on feature vectors computed from word embedding had close but lower performance than the other models. Despite the limited number of the testing data extracted from Snopes, similar conclusions could be made.

Even though the overall performances of the models using test data were good, there might be rooms for improvement. I would recommend further experiment with different scenarios. For example, using pre-trained word embedding models trained by very large data set (which can be loaded using spacy or gensim libraries), or using a different word embedding techniques such as doc2vec, skip gram (other type of word2vec), Fast Text, or GloVe. Validation would also be more sensible if more test data can be extracted from Snopes or other news sources.

1. Bisaillon,C. “*Fake and Real Data Set Classifying the News”,* Kaggle, Accesses May 2, 2020 .<https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset> [↑](#footnote-ref-0)
2. <https://www.snopes.com/?s=2017+archive+news> [↑](#footnote-ref-1)