
# Fake News Detection

9/19/2020

—

Nusrat Islam
Springboard DS Career Track

# Introduction

*"Falsehood flies, and the Truth comes limping after it," Jonathan Swift once wrote.*

Although a hyperbole for several decades, it has now become a reality in this age of social media. We have seen on various occasions how fake news spreads like wildfire on social media and how it influences people on their decisions. From democracy to the global economy everything is affected largely by false information. In that context, it is really important to identify fake news as soon as it is available on social media to limit its influence on people.

Wikipedia defines fake news as a form of news consisting of deliberate **disinformation** or hoaxes spread via traditional **news media** (print and broadcast) or online **social media**."

The difference between simple hoaxes like "Moon landing was fake", etc. and fake news is the fact that it carefully mimics the "style" and "patterns" that real news usually follows. That's what makes it so hard to distinguish for the untrained human eye.

From a Natural Language Processing (NLP) perspective, this phenomenon offers an interesting and valuable opportunity to identify patterns that can be coded in a classifier.

In this project we will use NLTK ( Natural Language Toolkit) to predict true news from fake news.

# Data

The data is obtained from Kaggle Data of Fake and real news dataset. The dataset consists of two separate csv files from various news sources and are already labelled as fake and true. The dataset consists of 4 columns. The news are categorized into subjects and are ordered by date.Data source: [Fake news dataset]

A label column was added to identify the dataset and then was merged as one pandas file as shown below:

| | title | text | subject | date | label |
|---|---|---|---|---|---|
| 0 | As U.S. budget fight looms, Republicans flip t... | WASHINGTON (Reuters) - The head of a conservat... | politicsNews | December 31, 2017 | TRUE |
| 1 | U.S. military to accept transgender recruits o... | WASHINGTON (Reuters) - Transgender people will... | politicsNews | December 29, 2017 | TRUE |
| 2 | Senior U.S. Republican senator: 'Let Mr. Muell... | WASHINGTON (Reuters) - The special counsel inv... | politicsNews | December 31, 2017 | TRUE |
| 3 | FBI Russia probe helped by Australian diplomat... | WASHINGTON (Reuters) - Trump campaign adviser ... | politicsNews | December 30, 2017 | TRUE |
| 4 | Trump wants Postal Service to charge 'much mor... | SEATTLE/WASHINGTON (Reuters) - President Donal... | politicsNews | December 29, 2017 | TRUE |

The dataset is from 2016-2017 and focuses largely on US 2016 election.

## Tools Used

**Pandas:** Data loading, manipulation, wrangling and

**Scikit learn:** Libraries for text feature extraction, vectorizer, metrics, classifier models and cross validation

**NLP:** stop words removal, text processing, n-gram analysis

**Matplotlib and Seaborn:** Data visualization

## Data Wrangling

The data was read using pandas. The dataset consists of 2 csv files each containing news that are already labelled as true and fake. Each file consists of 4 columns. The news are categorized into subjects and are ordered by date. After reading the data, we created a column named label with two labels TRUE and FAKE. This will be our prediction variable later on.

We then used the describe method to get descriptive statistics about the dataset.

From the table below we can see that there are approximately 45000 news with 39000 being unique. There are 14 unique title and 11000 subjects.

We checked for missing values and there were none. In

| | title | text | subject | date | label |
|---|---|---|---|---|---|
| **count** | 44898 | 44898 | 44898 | 44898 | 44898 |
| **unique** | 38729 | 38646 | 8 | 2397 | 2 |
| **top** | Factbox: Trump fills top jobs for his administ... | | politicsNews | December 20, 2017 | FAKE |
| **freq** | 14 | 627 | 11272 | 182 | 23481 |

order for the NLP model to analyse text data we need to clean and prepare the data into a machine readable format. The first step in this data cleaning process is to remove tags and numbers as they do not convey any inherent information for the prediction. Also, contractions like ain't are difficult for models to read and so we write a function to convert any contractions into its expanded form.

We then scrape any additional white spaces and convert the text into lowercase. The next step is to apply lemmatization with stopwords removal. Lemmatization is the process of converting a word into its root form. For example having and have are essentially the same word except for the tense. With lemmatization having is converted to have. We also remove stopwords such as 'and' and 'the'. Stopwords removal depends on application of NLP. In our

case, we are analysing large articles which contain large numbers of stopwords. If these are not removed they will bias the model and make predictions useless. After that we remove any remaining punctuation. At this point we have a clean text data that is ready for analysis.

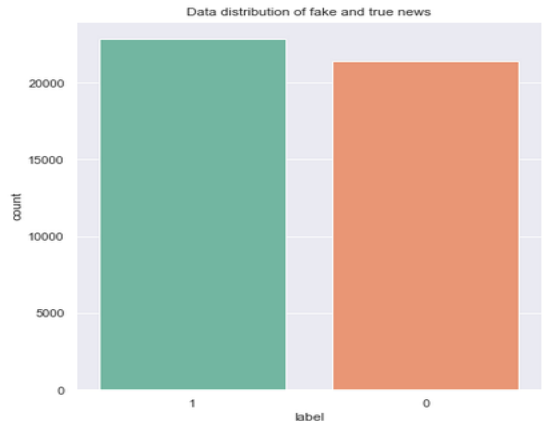
| | title | text | subject | date | label |
|---|---|---|---|---|---|
| 0 | as us budget fight loom republicans flip fisc... | washington reuters the head conservative re... | politicsnews | December 31, 2017 | TRUE |
| 1 | us military accept transgender recruit monday ... | washington reuters transgender people allow... | politicsnews | December 29, 2017 | TRUE |
| 2 | senior us republican senator let mr mueller job | washington reuters the special counsel inve... | politicsnews | December 31, 2017 | TRUE |
| 3 | fbi russia probe helped australian diplomat ti... | washington reuters trump campaign adviser g... | politicsnews | December 30, 2017 | TRUE |
| 4 | trump want postal service charge much amazon ... | seattlewashington reuters president donald ... | politicsnews | December 29, 2017 | TRUE |

We save the cleaned data into a csv file for exploratory data analysis.

## Exploratory Data Visualization

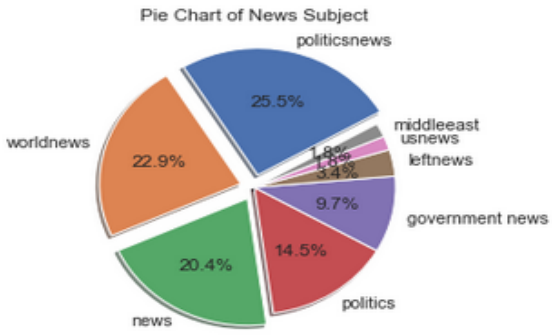
We read the cleaned data using pandas. We then map the label TRUE into 1 and FAKE into 0.

We then looked at the distribution of TRUE and FAKE labels in the dataset to verify if the dataset is balanced.




We have roughly an even balance between the two classes. This means we do not need to run any undersampling or oversampling.

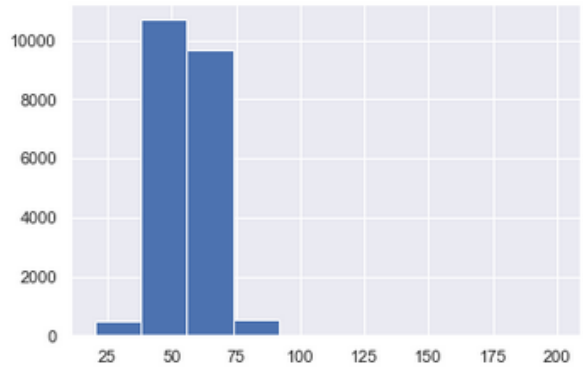
We then plotted a pie-chart to see the news sources. There were 4 predominant sources and the rest coming from 5 other sources.

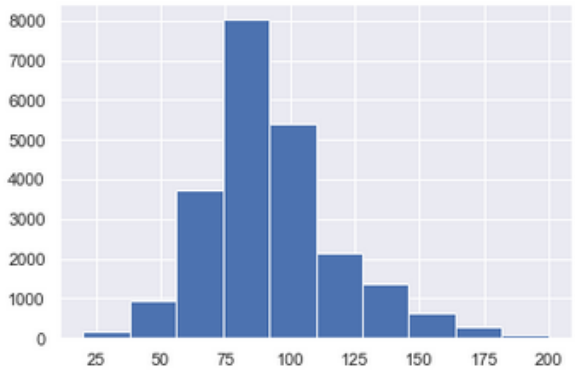



Next we explore the number of characters in each sentence and analyze if there is a

difference in the character count for Fake and True news.

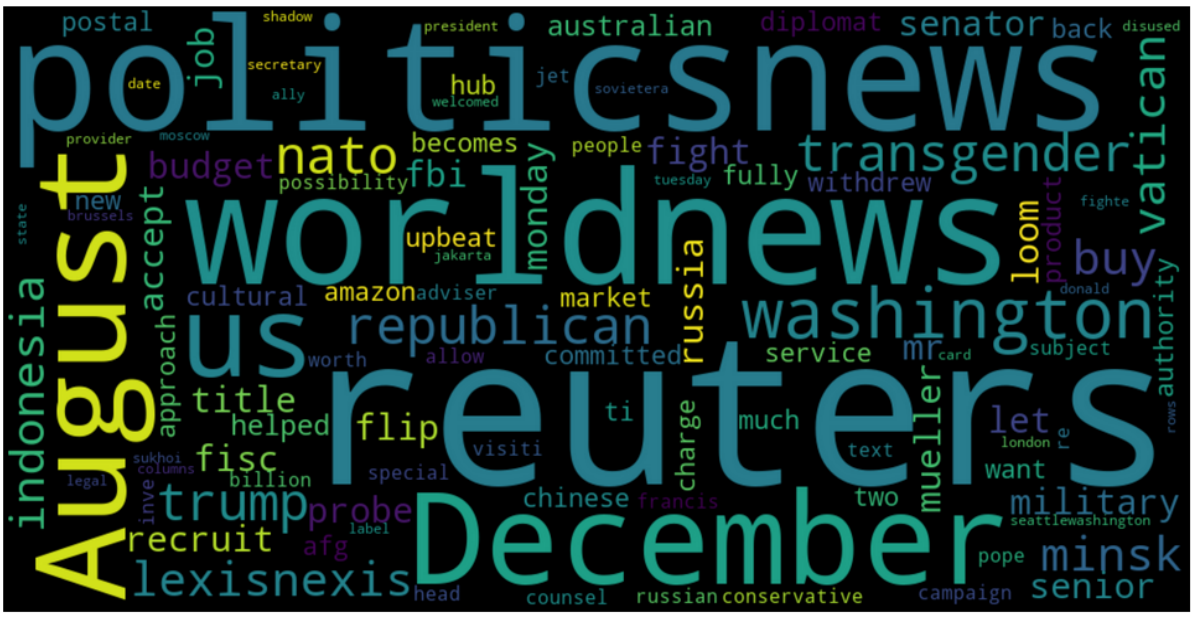
With the exception of the title both true and fake news have roughly the same number of characters. In the title we see that TRUE news (left) has a smaller range compared to Fake (right).



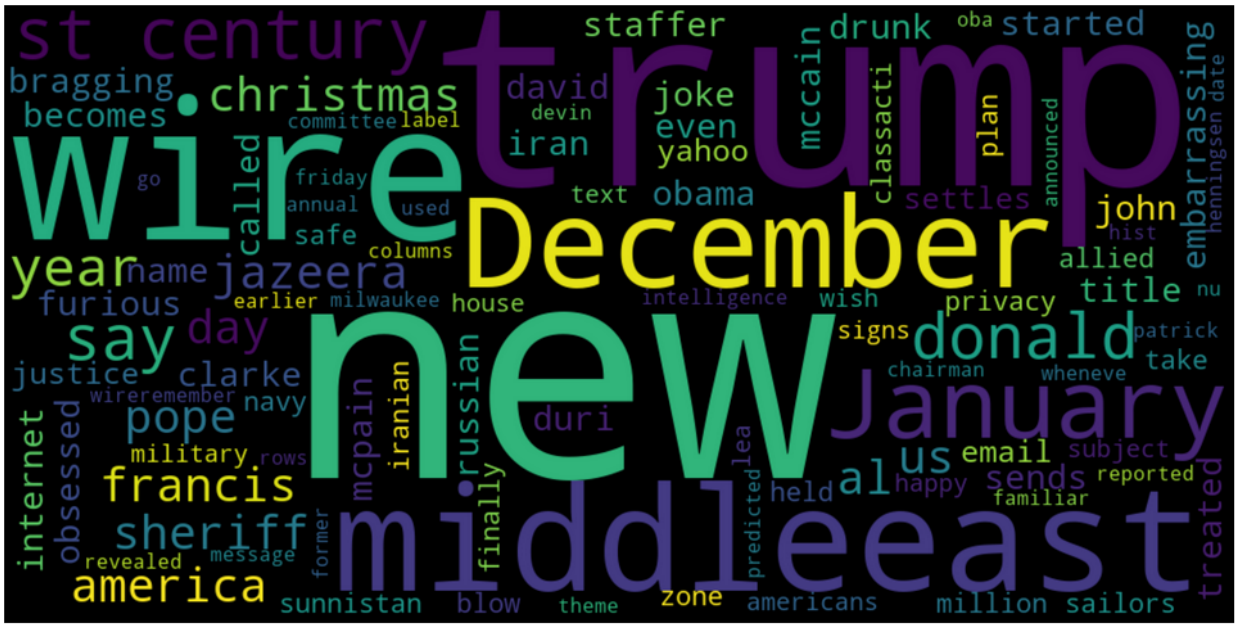




We then created wordcloud to see the words that were most common in each of the categories.

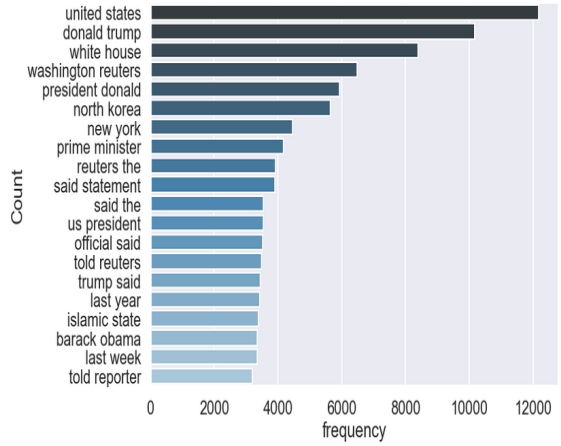
Reuters, worldnews, politics news,August, December, us and republican seemed to be the most prominent words in TRUE news.

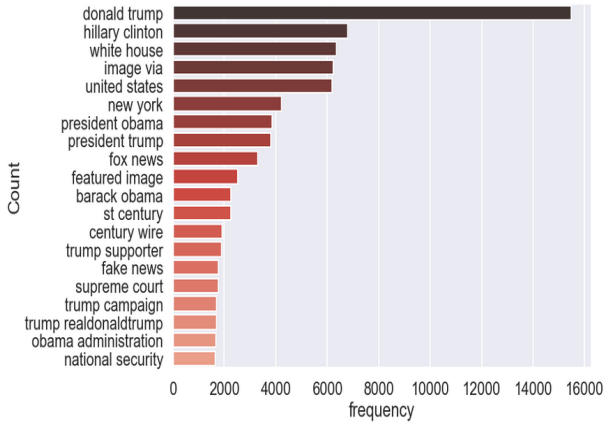



Looking at the wordcloud of fake news we find President Trump, wire, December, January, middle east being more prominent.




Text data are high dimensional and in order to visualize them we first need to convert them into vectors. We use countvectorizer from scikit learn to help convert text to vectors or n-grams and count the frequency for visualizations. We then compared the bigrams for both TRUE and FAKE labels to see the words that appear frequently in each classes.




Although president Trump was the top bi-gram in both classes, in the FAKE classes the next bi-gram was almost halved. In the TRUE news class other labels like white house and washington reuters were also frequent.

From the visualizations we can conclude that the news mostly contains political content and president Trump is most common across both labels.

## Feature Engineering

In order to perform feature extraction we used the scikit learn package TF-IDF transformer. **TF-IDF** stands for Term Frequency — Inverse Document Frequency and is a statistic that aims to better define how important a word is for a document, while also taking into account the relation to other documents from the same corpus.

This is performed by looking at how many times a word appears into a document while also paying attention to how many times the same word appears in other documents in the corpus. So then **TF-IDF is a score** which is applied to every word in every document in our dataset. And for every word, the TF-IDF value increases with every appearance of the word in a document, but is gradually decreased with every appearance in other documents. This score is then fed into the algorithm.

## Algorithms and ML model

Again we use scikit learn library to use a bunch of different algorithms to compare the accuracy and precision of the predictions. The models used in this learning are Multinomial Naive Bayes, Random Forest, Logistic Regression, Stochastic Gradient Descent.

From the 4 models it is observed that the exception of Naive Bayes, all of them have very few mis-classifications. The ROC_AUC score is 0.964 which is exceptionally well for text classification.

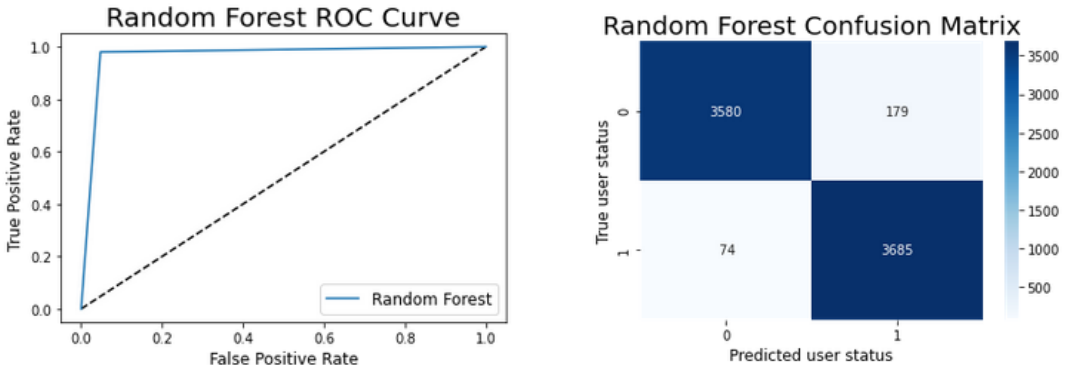



Fig: ROC-AUC curve and confusion matrix for Random Forest

- The top three models are Logistic Regression, Stochastic Gradient Descent and Random Forest

- Random Forest, SGD and Logistic Regression has comparable ROC-AUC score. Although Logistic Regression has the highest ROC-AUC score it trained only a max_df = 0.25. This means the modeled ignored terms that have a document frequency strictly lower than the given threshold. However, Random Forest is more scalable, and interpretable and also performs better with noisy data. Considering these, we chose Random Forest as the best model.

- Feature Importance of the model shows that 'president Donald Trump', Washington' 'President Obama' were given the highest importance. This is in alignment with the fact that this dataset is indeed a repersentation of the news during the 2016 US Presidential Election.

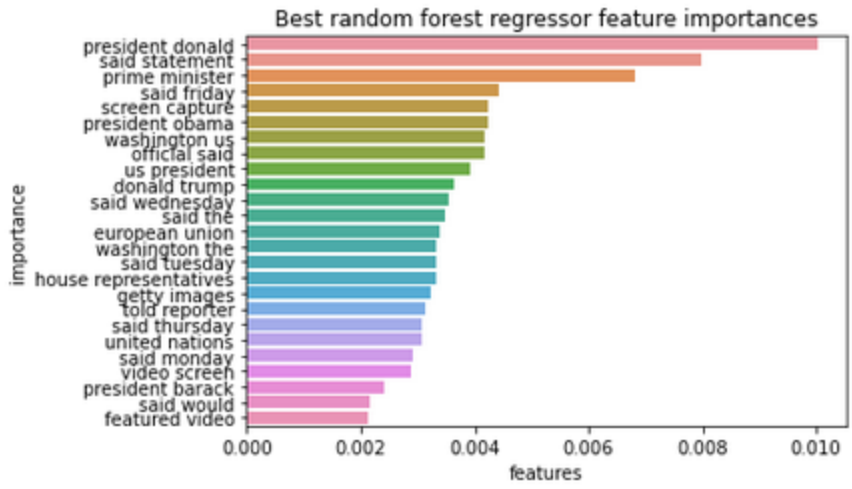


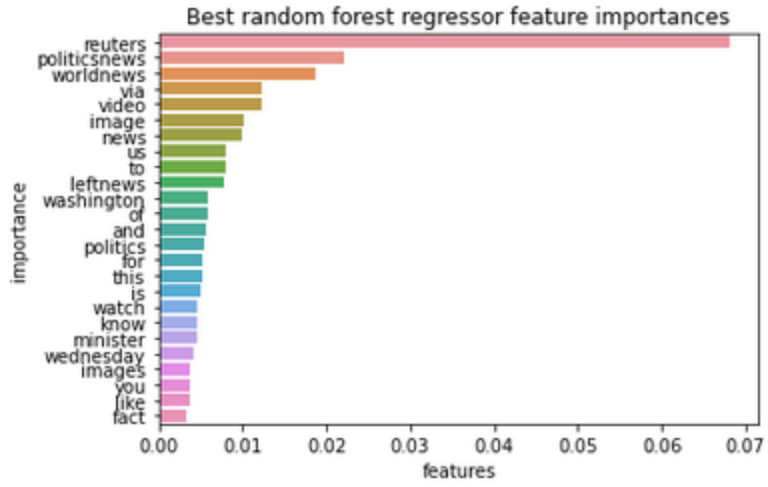

Fig: Feature importances of Random Forest model

Although the model performance is very good, looking at both the feature importance of the model the n-gram analysis from EDA it can be inferred that the data is biased towards US Presidential Election and any news outside this scope might be difficult for the model to predict. We need a bigger dataset covering a wide range of news for both TRUE and FAKE labels to make a more generalized model.

## Data leakage

After the exploratory data analysis, when we built the first model the accuracy was close to 100%. The ROC-AUC curve showed nearly perfect/theroretical pattern. This made us wonder if

the model is leaky. We analysed the feature importance from the model. The chart below shows the words that were given the most importance while making the predictions.




From the above chart, it is clear that the word 'reuter' is causing the model to be leaky. This word appears only 1% of the time in fake news and 65% of the time in true news. This might help create a pattern which might not be always available or 'true' in a real world setting. Removing this word will help get better generalization.

- Some of the headings of subject columns also appear in the top 25 feature importances. This is again misleading as both class of news does not come from the same subject. This can be another cause of leakage.
- Also, it will be better to apply stopwords removal one more time since we see terms like 'of' and 'and' being given top feature importances.
- We can also see that there are duplicate values in both text and title columns. If these classes are present in both test and train sets than that will also yield to the leakiness. We plan to remove the repeated text and title from the dataset.

## Conclusion

In this project we aimed to predict fake and true news using NLP and machine learning methods. We were successful in developing a Random Forest model that is capable of predicting TRUE and FAKE news with and ROC-AUC score of 0.964. Although the accuracy is high, the data is heavily biased towards the US 2016 Presidential Election and so will not be able to generalize well.