FAKE NEWS DETECTION

MSA 8200 PREDICTIVE ANALYTICS

Final Project Report

Supraja Ravipati
Shashank Goswami
Sanjay Jain
RaamKumar Thiyagarajan

Table of Contents

INTRODUCTION

About Fakebook

Fakebook is a fictitious web source created to share fake news among the clients. But recently some of the users have started sharing real news in place of fake news. It became important for the organization to identify the fake news Vs real news to get a real sense. Fakebook initially hired actual people to assess the nature of the news and identify the real news from the fakes. However due to the large number of users, it became inevitable to automate the process of classifying this news.

Please note that this website is purely fictional and created for research purposes and has nothing to do with any possible real life resemblance

Project Statement

With the advent of Digital world, there emerged a zillion of news distribution channels or mediums. Everyday millions of news articles are distributed across the internet. There are many surveys which echoes the fact that most percentage of the news being shared is falsified or fabricated that has nothing to do with the truth.

Hence, it became crucial for any organizations or general public to filter such news and identify the true news. With the scale of the data that needs to be analyzed it is almost impossible for organizations to rely on humans to do the task. So, with our project we are trying to explore

different text preprocessing techniques and machine learning models on unstructured data to classify the real vs fake news being shared on Fakebook.

Possible Questions to Answer

- ✓ How real Vs fake news is structured
- ✓ What are the new natural language processing techniques to be used?
- ✓ What are the most popular patterns in both the news

Text Classification

Text Classification is the process of assigning tags or categories to text according to its content. It is one of the fundamental tasks in Natural Language Processing (NLP) with many more applications. These techniques are specifically useful while dealing with unstructured data generated in the real world. Text can be extremely rich source of information, but extracting insights from unstructured data can be time consuming and costly. With the advance of digital world, there is unstructured data generating from various resources across the world. It is becoming crucial for businesses to understand this data to make more informed and accurate decisions. In our project, the focus is going to be classifying the text into two categories i.e. Real and Fake by taking advantage of the different NLP techniques.

Text Pre- Processing

Removal of empty rows

Removing the rows that have text column as null is a crucial step in preprocessing as this might break the model

Removal of Punctuations

Punctuations which are a part of the unstructured data and does not contribute to the model at all can act as noise to the model hence need to be removed.

Removal of Stopwords

Some vocabulary which is a part of natural language/unstructured data are considered as stowords. These words include as ‘the’,’in’,’at’. They do not help much in classifying the text, hence we remove them to reduce the data load/noise to the model.

Tokenization

Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. In the process of tokenization, the data is broken down into words. The tokens become the input for another process like parsing and text mining.

Word Stemming and lemmatization

Stemming is a process of *reducing* different forms of the word into its root form, which need not be the same root as its dictionary root. In this process, words that are derived from one another can be mapped to a central word or symbol, especially if they have the same core meaning. Lemmatization is the process of reducing a word to its dictionary root form. These two techniques help reducing in the variance in our vocabulary and normalizes the text.

N-grams

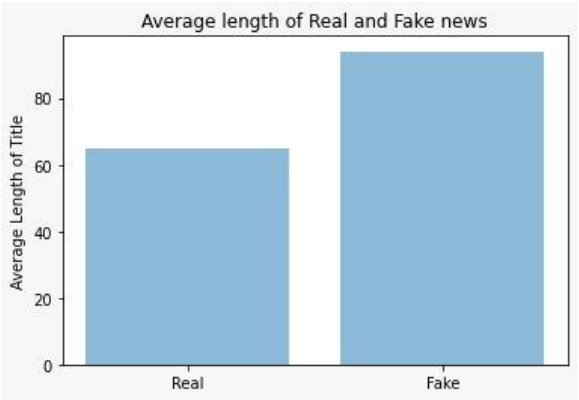
A combination of multiple words together is called N-Grams. N grams (N > 1) are generally more informative as compared to words and can be used as features for language modeling. N-grams can be easily accessed in NLTK function, which returns a tuple of n successive words

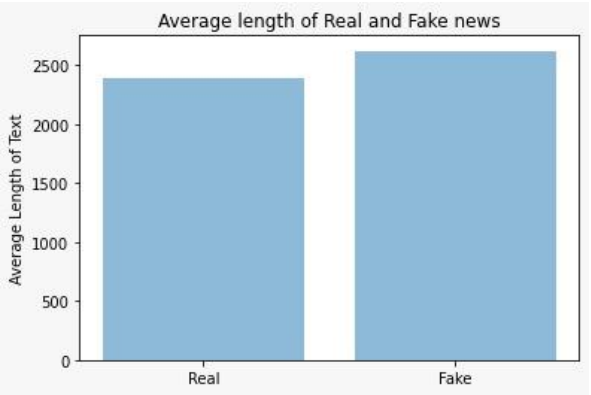
Note: We have used many of the packages available from NLTK library which can be seen in the code

Exploratory Data Analysis

To begin with, we have a historical dataset of news entries, where the fake and real nature of the news is specified. There are 38000 news entries in total and for each sample in the dataset the following features are provided:

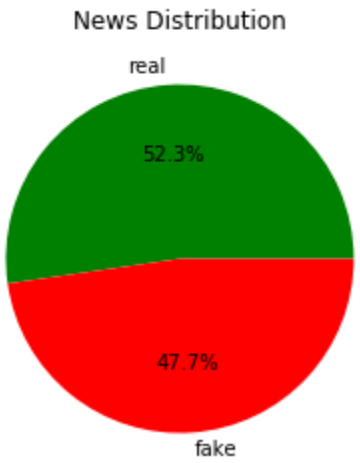
- Date: is the date of the news article was posted
- Title: is the title of the news article
- Text: is the content of the news, which could be several paragraphs
- Type: Represents the nature of the news, whether it is a fake or real story





	date	title	text	type
0	2016-03-11	Governor extends Flint water emergency as stat...	WASHINGTON (Reuters) - Michigan Governor Rick ...	real
1	2017-10-09	DEMOCRATS CONVENIENTLY FORGET 6,000 Prisoners ...	Democrats are calling for President Trump s sc...	fake
2	2018-02-01	Mexico recognizes Honduran president as winner...	MEXICO CITY (Reuters) - Mexico recognized Hond...	real
3	2016-11-20	BOOM! Wikileaks Shows Hillary Speech To Banker...	No wonder she didn t want anyone to see her sp...	fake
4	2017-07-06	Paul Ryan says confident tax reform will pass ...	WASHINGTON (Reuters) - Republicans will be abl...	real

```
Out[7]: Text(0.5, 1.0, 'News Distribution')
```



Our focus is going to be on 'Text' Column to dive into the news content and analyze the text in Real and Fake news and ultimately classifying the unseen test data.

Key Observations

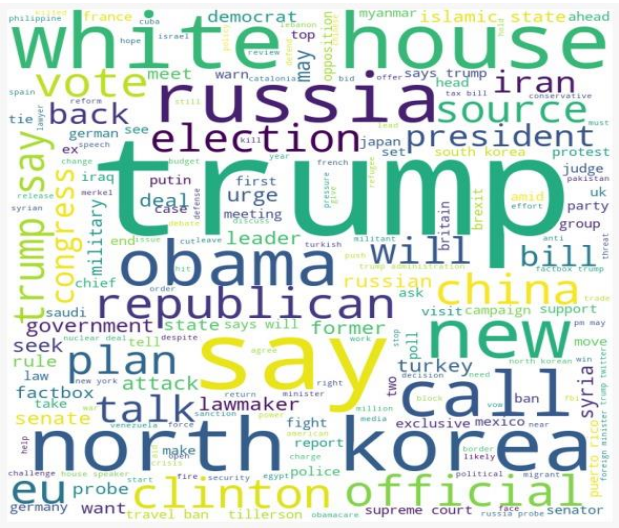
- This is a binary classification problem
- There are 534 missing rows in 'Text' column that need to be removed
- There is no imbalance in the response variables i.e. the labels 'Fake' and 'Real' are almost of the same size
- The keywords in both the labels
- The empty rows in text column is still classified as 'Real' indicating that there is information in title alone

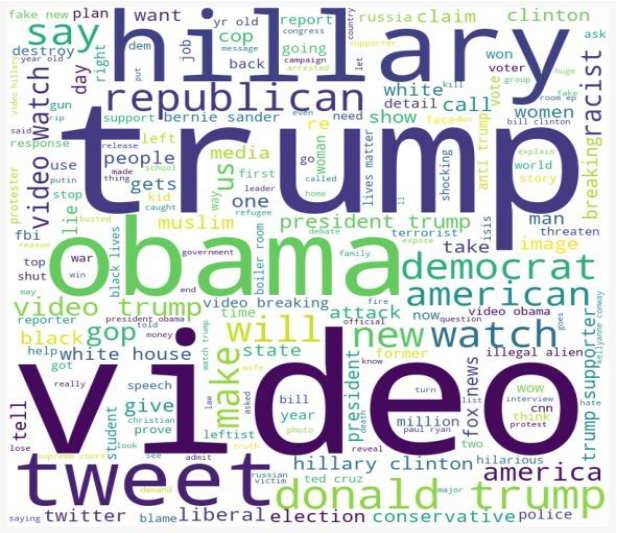
Word Cloud Data Visualization

A **Word cloud**, also known as a Tag cloud, is a visual representation of text data, typically used to depict keyword metadata (tags) on websites or to visualize free form text. Word clouds are a

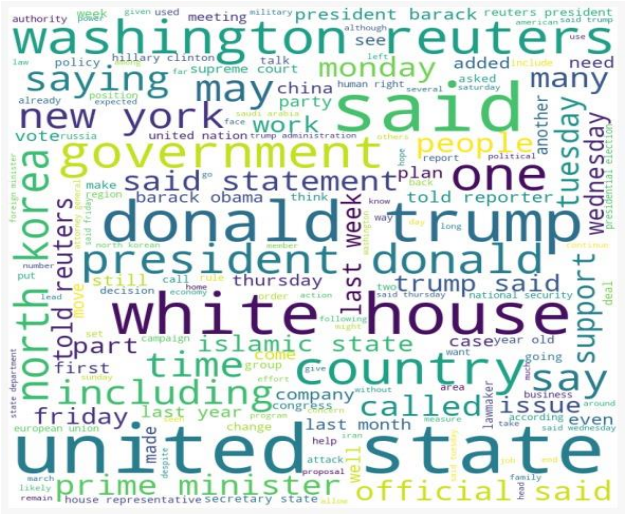
popular type of infographic with the help of which we can show the relative frequency of words in our data.

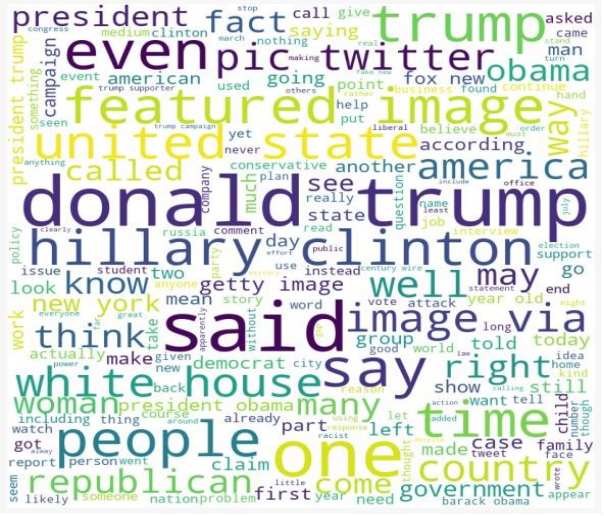
The following section comprises of the word cloud representation of Real Vs Fake news. This gives us a broader understanding of the keywords contained in both the types.





REAL TITLE **Vs** **FAKE TITLE**





REAL NEWS **Vs** **FAKE NEWS**

Natural Language Processing Techniques Used

Python's nltk library is one of the most famous and widely used natural language processing libraries. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more

Task	Package Used
Stopwords removal	stopwords
Tokenizing	word_tokenize
Stemming	Snowball Stemmer
Lemmatizing	Wordnet Lemmatizer

Feature Extraction

Feature Extraction is the final step before feeding the data into the model. We need to convert the preprocessed to text into vectors to run the model. While there are many techniques available to do this job, we ended up choosing the two most popular methods

1) ***Count Vectorizer:*** The Count Vectorizer provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also encode new documents using that vocabulary. It creates a huge and sparse vectors of all the unique words in our word corpus.
2) ***Tf-Idf Vectorizer:*** Scikit-learn's "Term Frequency – Inverse Document Freqiency" tries to highlight the words that are frequent in one document but not across the documents. It considers the rarity of the words thus overcoming the one disadvantage Bag of words and Count vectorizer carry.

Models and Validation

Explored various text classification models in this project starting from very basic Logistic Regression to advanced techniques like Support Vector Classifier.

Models Chosen

- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Classifier
- XG Boost Classifier

Validation Metric

- 10-fold cross validation is chosen to validate our models to reduce the vast variations in test errors that occur when we use validation set approach

Model Results

- Results When Count Vectorizer is used

Model	Accuracy (%)	Precision (%)	Recall (%)
Logistic Regression	98.62	100	100
Decision Tree Classifier	99.67	100	100
Random Forest	99.67	97	97
Support Vector Classifier	99.36	99	99
XG Boost Classifier	99.63	100	100

- Trained all the models with text column as input and Tf-Idf Vectorization instead of count vectorization

Model	Accuracy (%)	Precision (%)	Recall (%)
Logistic Regression	98.63	99	99
Decision Tree Classifier	99.60	100	100
Random Forest	99.12	99	99
Support Vector Classifier	99.37	99	99
XG Boost Classifier	99.78	100	100

- Text and title combined, to extract the information about news that has empty content but has some information in the title

Model	Accuracy (%)	Precision (%)	Recall (%)
Logistic Regression	98.73	99	99
Decision Tree Classifier	99.63	100	100
Random Forest	99.24	99	99
Support Vector Classifier	99.37	99	99
XG Boost Classifier	99.79	100	100

Conclusion

Due to the nature of the semi cleaned data we got, all the models ended up giving good results. In the real time scenario where the data is highly unstructured, it is advisable to use advanced feature extraction techniques such as Tf-Idf vectorizer over traditional Bag of words or Count vectorizer approaches. While choosing the models it is highly recommended to use

computationally effective techniques such as XG Boost, Support Vector classifiers to get good results.

Challenges Faced

- Exploring through various text analytics methods before choosing each one
- Computational Difficulties due to the size of the dataset
- Increased computational difficulties due to the challenges in validating the model (Cross validation)