

Fake News Detection

Identifying fake news has become an important issue. Increasing usage of social media has led to an increase in the number of people who can be influenced, thus the spread of fake news can potentially impact important events. Fake news has become a major societal issue and a technical challenge for social media companies to identify and has led many to extreme measures, such as WhatsApp deleting two million of its users every month to prevent the spread of fake news. The current problem of fake news is rooted in the historical problem of disinformation, which is false information intentionally, and usually clandestinely, disseminated to manipulate public opinion or obfuscate the truth. My work addresses the problem of identifying fake news by detecting and analyzing fake news features, identifying the textual and sociocultural characteristics of fake news features.

The problem is real and hard to solve because the bots are getting better and are tricking us. Is not simple to detect when the information is true or not all the time, so we need better systems that help us understand the patterns of fake news to improve our social media, communication and to prevent confusion in the world.



MAIN GOAL

- The aim of this project is to walk through the process of creating a machine learning model using python and NLP in order to successfully detect fake news.
- This project aims to apply different algorithms & techniques on Fake news data set and compare the results.
- Measures used to compare those are Precision and Recall.
- To get high accuracy to determine a news is fake or true.

- **The Data**

- The data comes from Kaggle, you can download it here:

- <https://www.kaggle.com/clmentbisailon/fake-and-real-news-dataset>

- There are two files, one for real news and one for fake news (both in English) with a total of 23481 “fake” tweets and 21417 “real” articles.



- In the class distribution we want to know:
- How many news are fake and how many are true?
- We have 23481 fake news and 21417 true news.
- We have a balanced mix of true and fake articles.

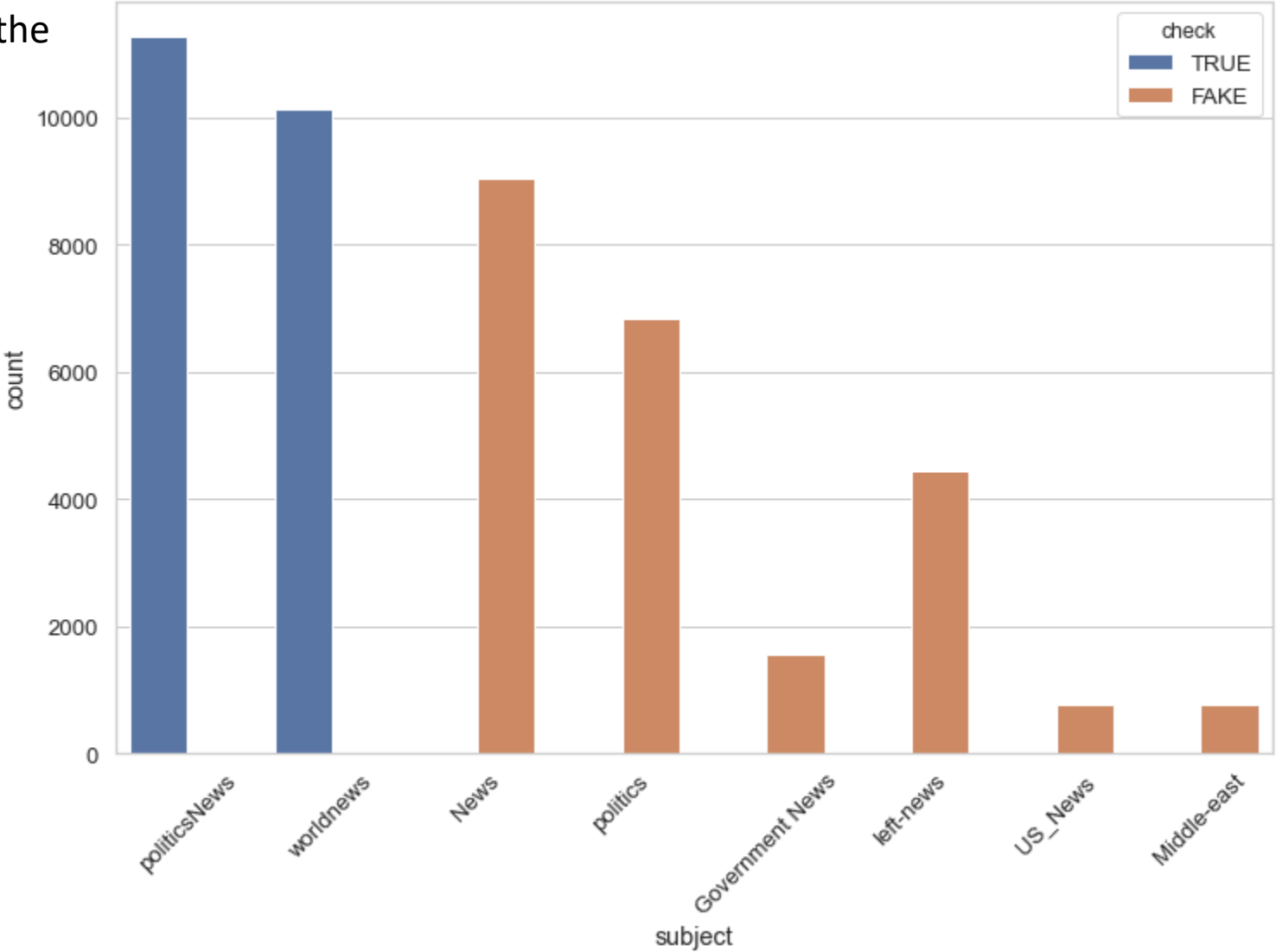
Methodology

- The approach proposed for this project is:
- Data Preprocessing
- Generating News Feature Vector
- Classification



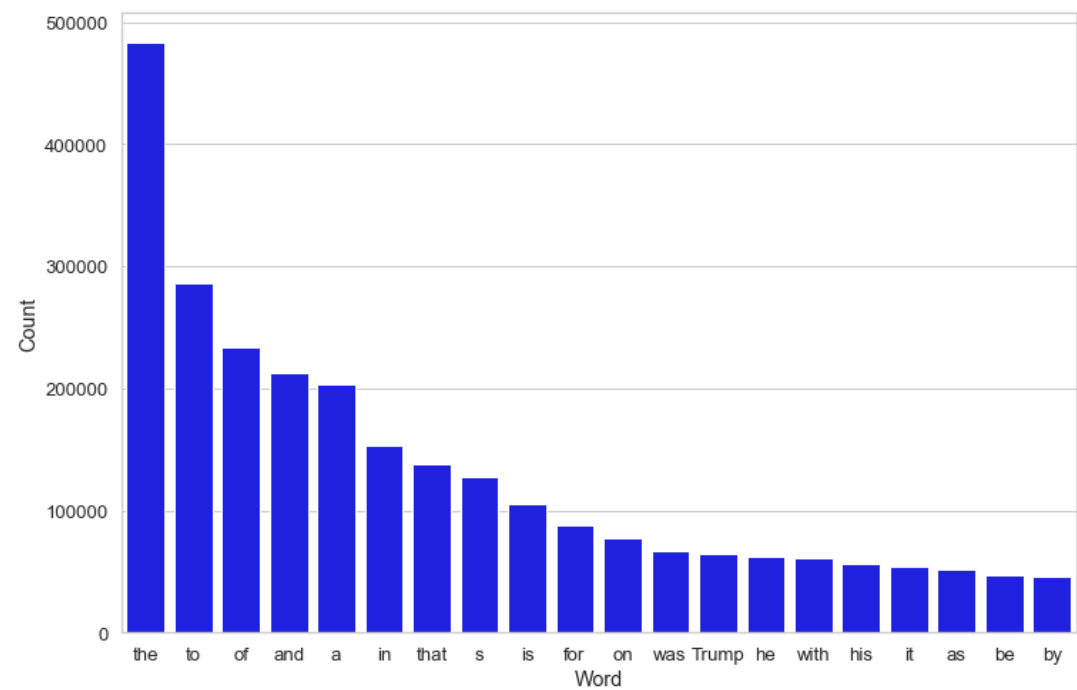
- We have politics News, world news as True and the rest of the news which include News, politics, Government News, left-NewNews,US_News, Middle-east are Fake.

subject	
Government News	1570
Middle-east	778
News	9050
US_News	783
left-news	4459
politics	6841
politicsNews	11272
worldnews	10145
Name: text, dtype: int64	

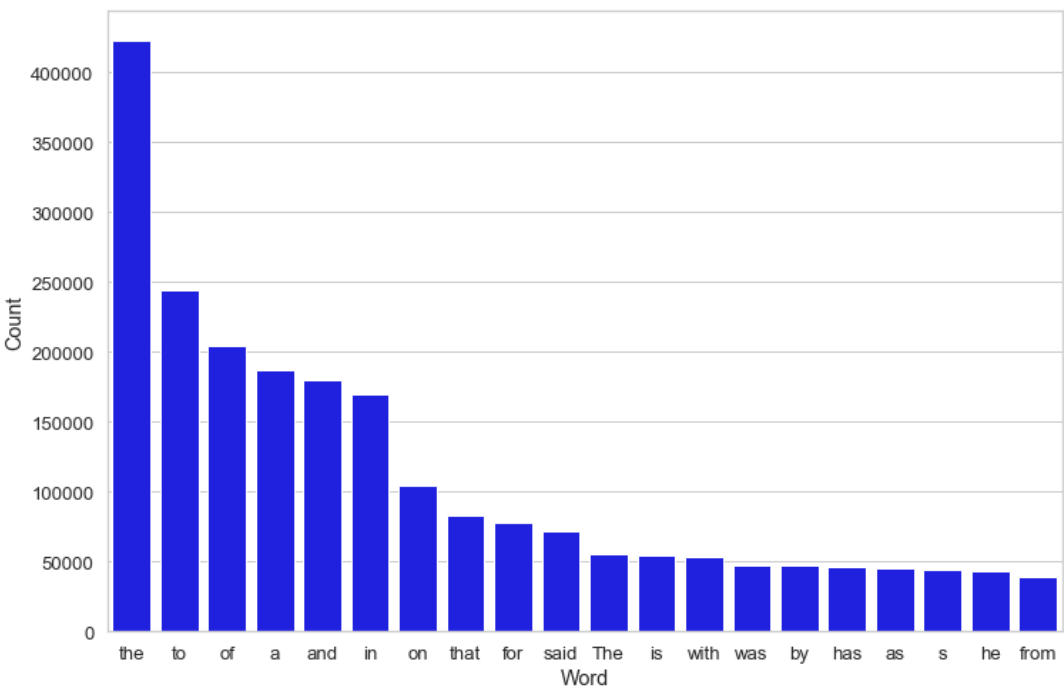


We can see most frequent words in real and fake news in plot:

Fake news:

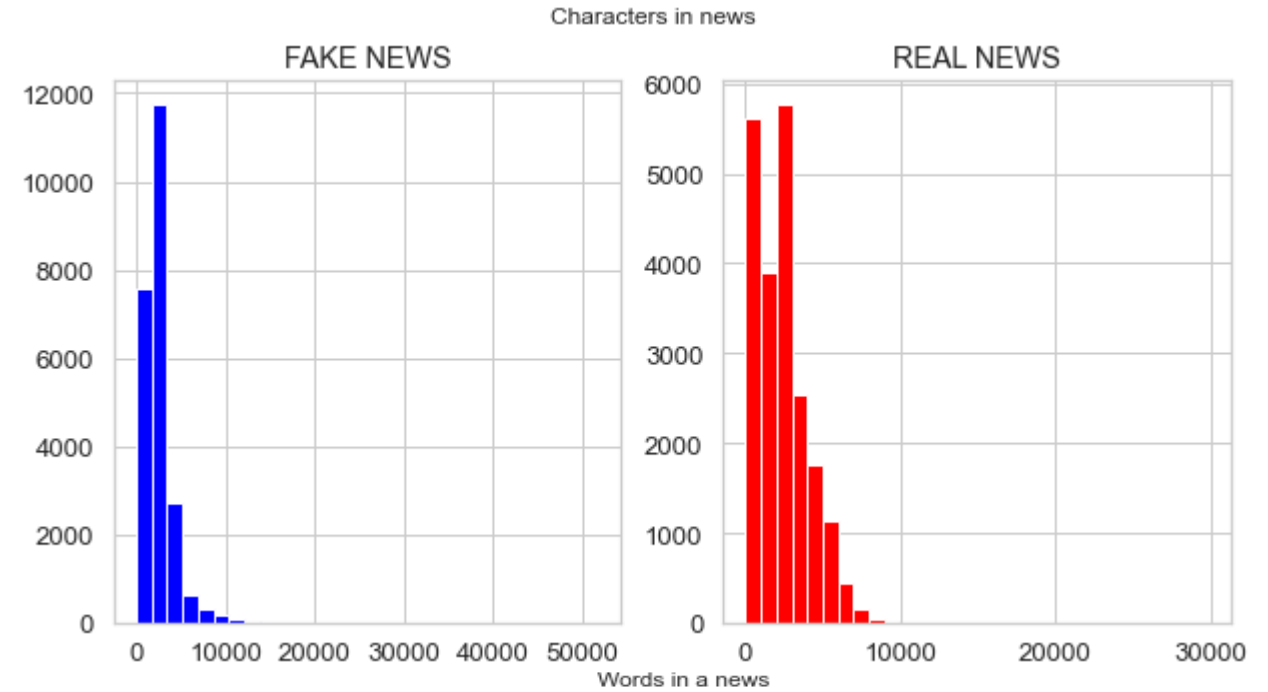


True news:



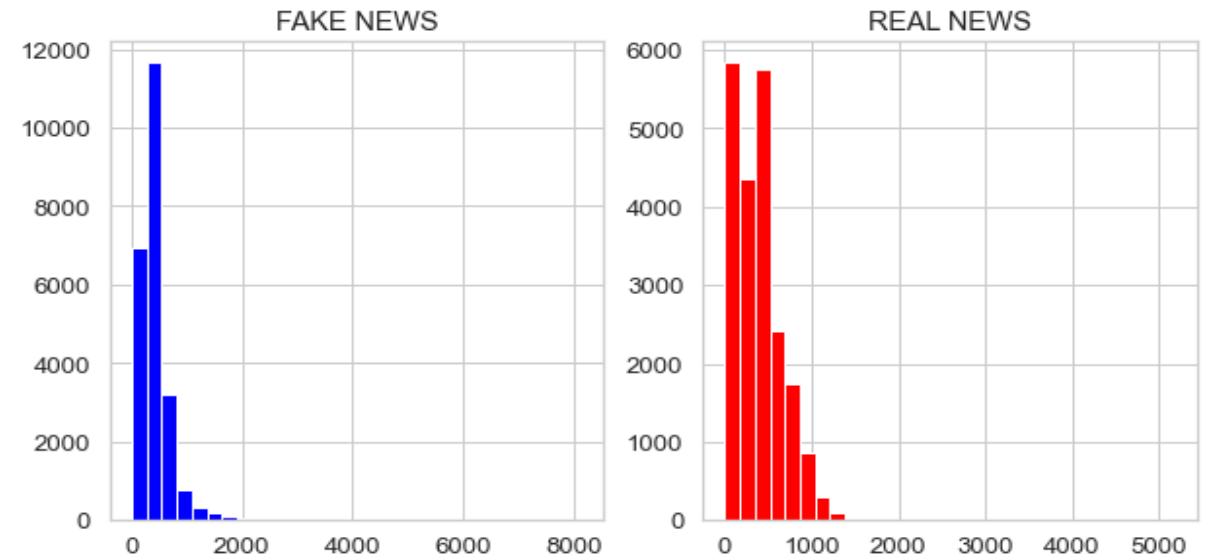
Number of characters in news

As we can see from analysis number of characters in Fake news is more than of Real news, because fake news generally use more characters to grab the attention.



Number of words in news

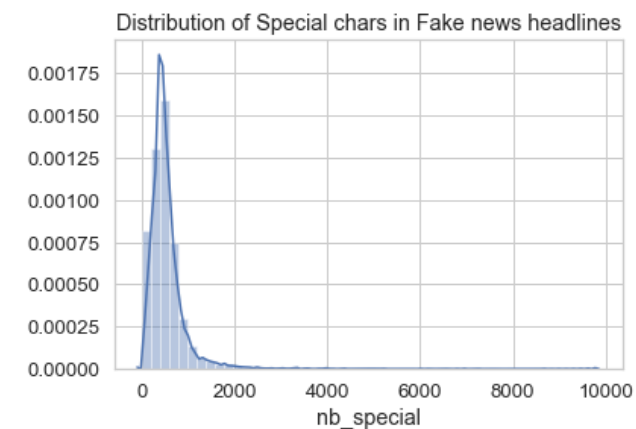
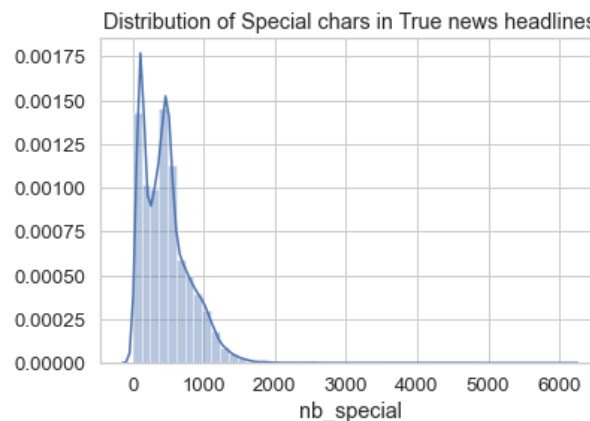
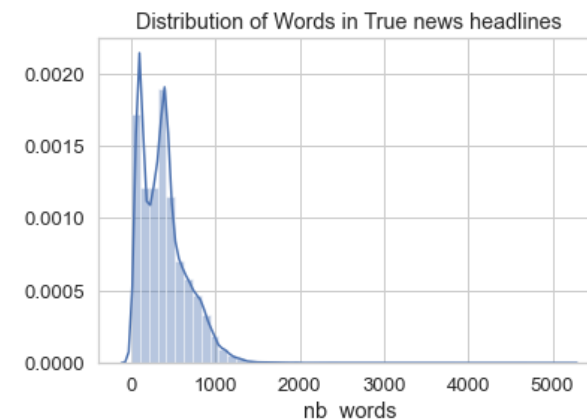
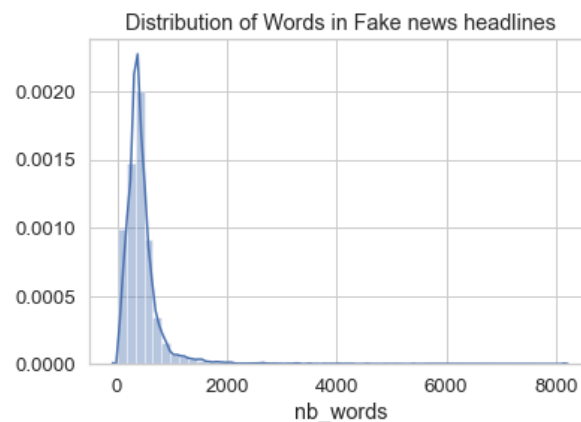
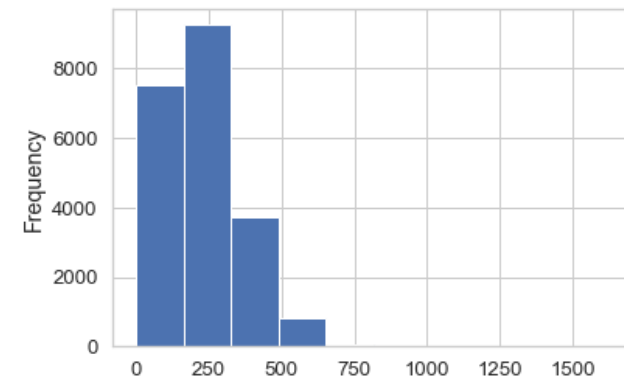
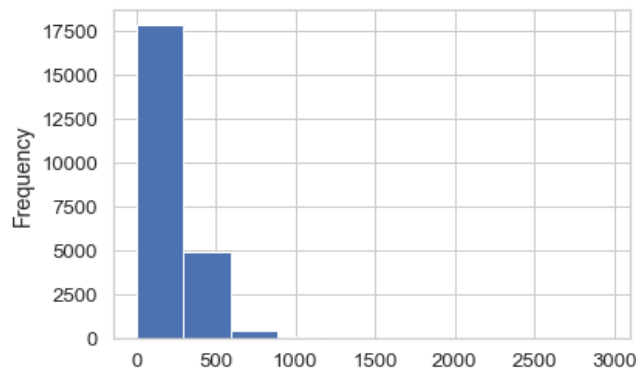
As we can see average number of words in real headlines is relatively less in comparison to fake news.



As we can see in these plots:

1. There are some more special Character in fake news than real News less use of special characters.

2. There are some more words In a fake news(right-skewed) than Real news, because fake news use superfluous language with more Words to grab the attention.



Detecting Fake News With Natural Language Processing (NLP):

As human being, when we read a sentence or A paragraph, we can interpret the words with The whole documents and understand the context.

It is possible to teach to a computer how to read
And understand the difference between real news
And the fake news using Natural Language
Processing
(NLP).



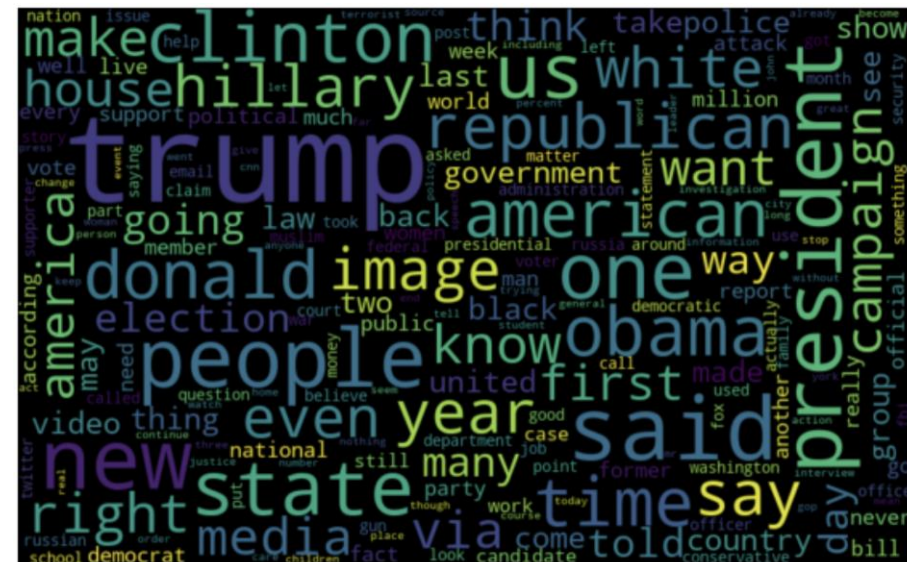
EDA

- The purpose of EDA is to enhance our understanding of trends in the dataset without involving complicated machine learning models.
- The last stage of my exploratory data analysis of the text is Word cloud analysis. Word cloud is a great way to represent text data. The size and color of each word that appears in the Word cloud indicate its frequency or importance.
- In the results, we can see how often some words used in the news text in fake or real news.

Word Cloud for fake news:



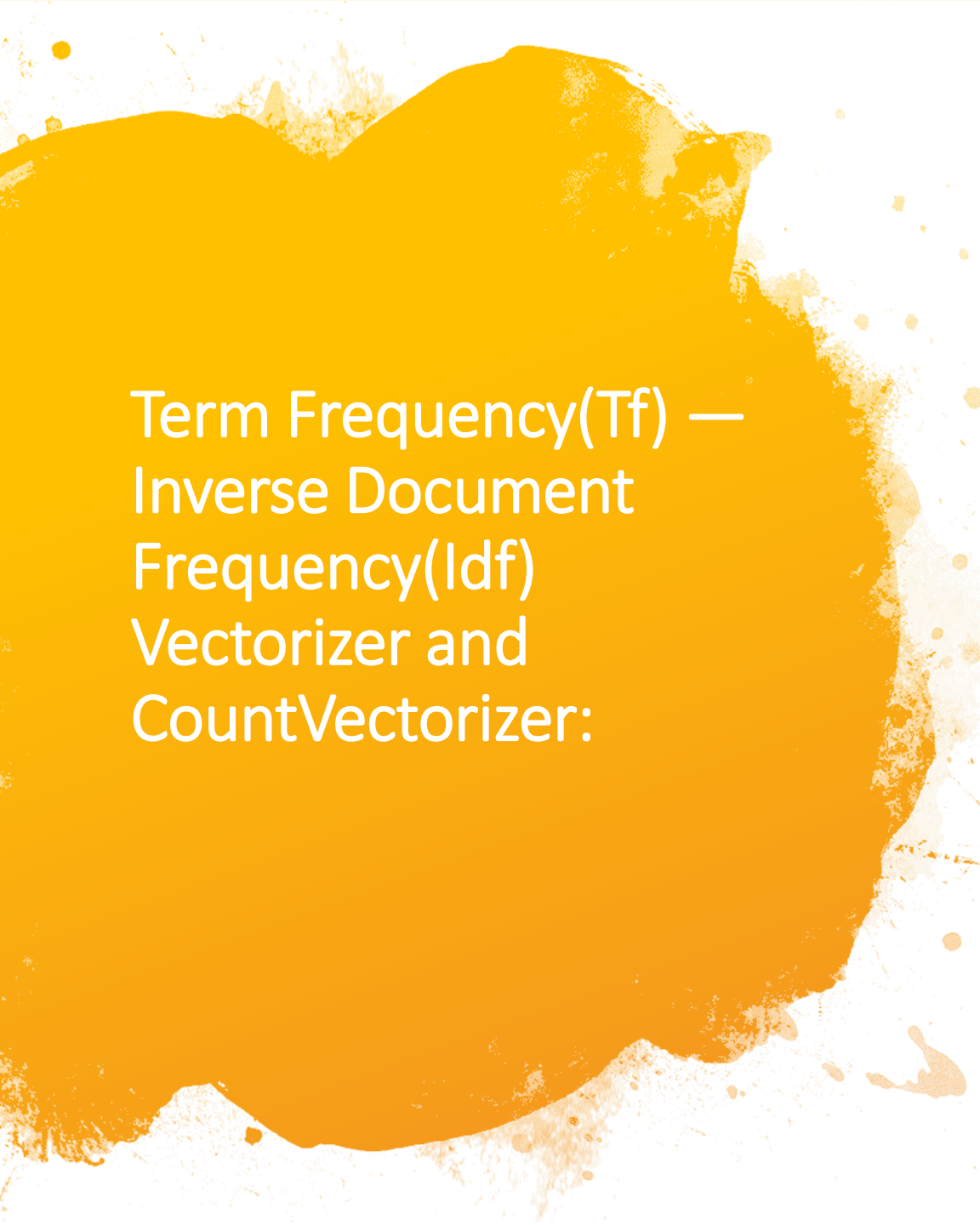
Word Cloud for real news:



A large, abstract orange watercolor splash shape on the left side of the slide, with various shades of orange and yellow, and some darker spots and splatters extending towards the right.

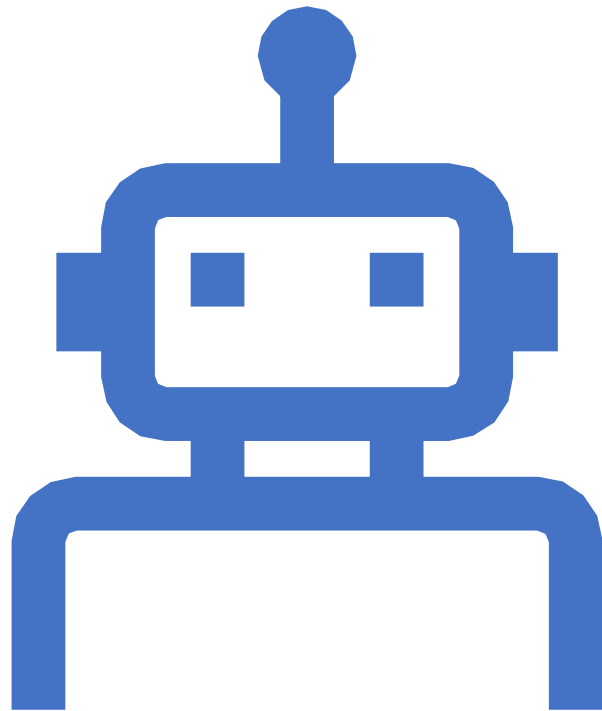
Preprocessing:

- In text preprocessing we have to convert all the text to lower case.
- Remove all the punctuations, stop words and digits.
- We tokenize the sentences using the keras tokenizer.
- After tokenizing we convert the sentences into sequence.
- The padd the sequence



Term Frequency(Tf) — Inverse Document Frequency(Idf) Vectorizer and CountVectorizer:

- Tf-Idf Vectorizer is a common algorithm to transform text into meaningful representation of numbers. It is used to extract features from text strings based on occurrence.
- In Preprocessing step we should convert the text to numbers which machine learning
- Using simple model like bag-of-words to deal with text data. To get a good idea if the words and tokens in the articles had a significant impact on whether the news was fake or true, I can use CountVectorizer and TfidfVectorizer.

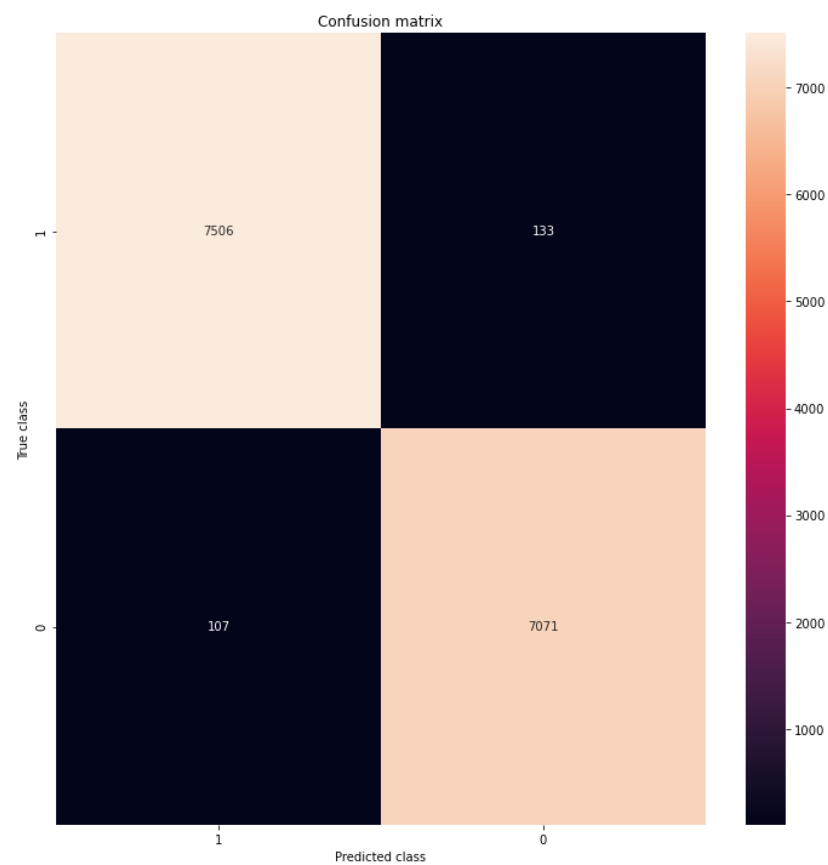


Classification Models

The model used were Logistic Regression(LG), PassiveAgressive Clasiifier, Naïve-Bayes(NB), Support Vector Machine(SVM), Random Forest(RF).

I will compare features and classifiers by their accuracy, Precision, Recall.

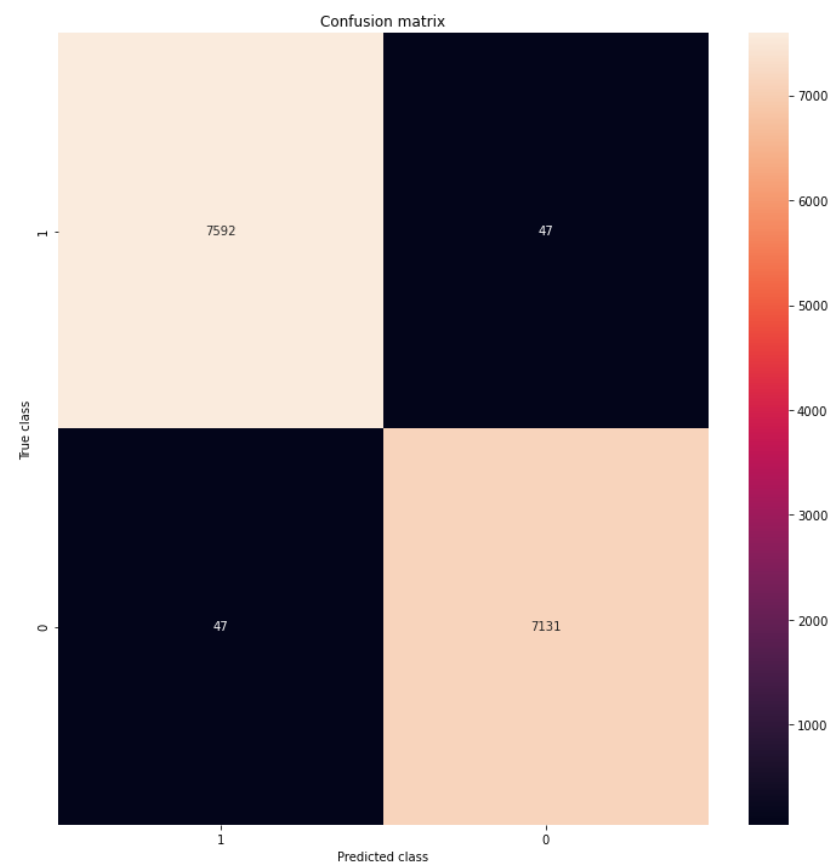
Logistic Regression



```
[[7506 133]
 [ 107 7071]]
```

	precision	recall	f1-score	support
FAKE	0.99	0.98	0.98	7639
TRUE	0.98	0.99	0.98	7178
accuracy			0.98	14817
macro avg	0.98	0.98	0.98	14817
weighted avg	0.98	0.98	0.98	14817

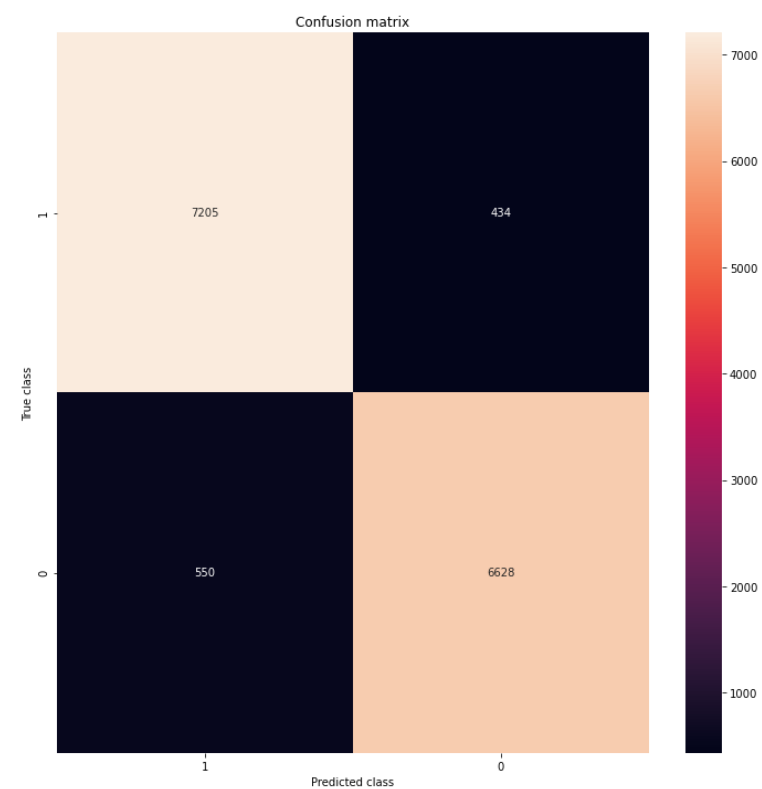
PassiveAggressiveClassifier



```
[[7592 47]
 [ 47 7131]]
```

	precision	recall	f1-score	support
FAKE	0.99	0.99	0.99	7639
TRUE	0.99	0.99	0.99	7178
accuracy			0.99	14817
macro avg	0.99	0.99	0.99	14817
weighted avg	0.99	0.99	0.99	14817

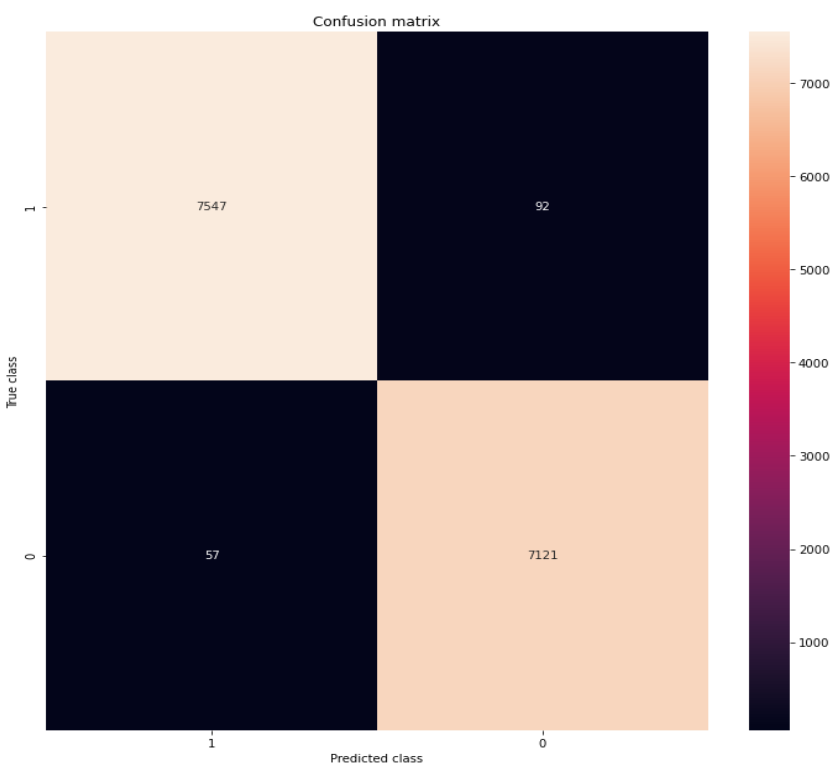
Naïve-Bayes



```
[[7205  434]
 [ 550 6628]]
```

	precision	recall	f1-score	support
FAKE	0.93	0.94	0.94	7639
TRUE	0.94	0.92	0.93	7178
accuracy			0.93	14817
macro avg	0.93	0.93	0.93	14817
weighted avg	0.93	0.93	0.93	14817

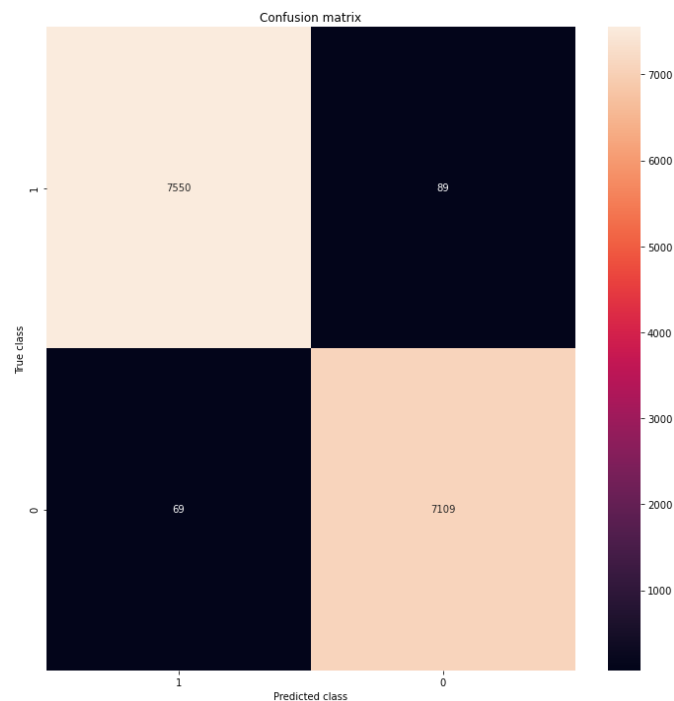
Support Vector Machine



```
[[7547  92]
 [  57 7121]]
```

	precision	recall	f1-score	support
FAKE	0.99	0.99	0.99	7639
TRUE	0.99	0.99	0.99	7178
accuracy			0.99	14817
macro avg	0.99	0.99	0.99	14817
weighted avg	0.99	0.99	0.99	14817

Random Forest



```
[[7550  89]  
 [  69 7109]]
```

	precision	recall	f1-score	support
FAKE	0.99	0.99	0.99	7639
TRUE	0.99	0.99	0.99	7178
accuracy			0.99	14817
macro avg	0.99	0.99	0.99	14817
weighted avg	0.99	0.99	0.99	14817

Model Performance

Model	Precision	Recall	Accuracy
Logistic Regression	0.99	0.98	0.98
PassiveAgressive	0.99	0.99	0.99
Naïve-Bayes	0.93	0.94	0.93
Support Vector Machine	0.99	0.99	0.99
Random Forest	0.99	0.99	0.99

Conclusion

- We successfully implemented a machine learning and
- Natural Language Processing model to detect whether
- An article was fake or real.
- We got 7592 articles correctly identified as Fake and 7131
- Correctly identified as real. When doing such a classification,
- it is important to check that we limit the number of false positives
- as they can cause real to be marked as fake.
- I would like to choose PassiveAggressiveClassifier method, because this method has less false positive and false negative.
- So overall PassiveAggressiveClassifier Method performed much better in determining in fake news cases which is around 99%