CM3070 FINAL PROJECT

Name - Shivam Maheshwari

Student Number - 190349118

FAKE NEWS DETECTION TEXT CLASSIFIER

Github - https://github.com/chunkydonut21/fake-news-classifier

Table of Content

I. Introduction

1. Project template

My final project is based on the CM3060 Natural Language Processing module. The project that I will be working on is Fake news Detection.

2. Project concept

What is Fake news?

Fake news is false or misleading information that is presented as a news. Fake news are stories that are fabricated, misleading with no verifiable facts. Fake news can often lead to damaging the reputation of a person or entity and brainwashing the society by misleading facts.

Source - https://en.wikipedia.org/wiki/Fake_news

Why Fake news is a problem?

In today's world, the rapid adoption of social media networks has increased the rate at which information spreads across the globe. The authenticity of information has become a major issue that affects businesses and societies, for both printed and digital media. The popularity of social media platforms has caused fake news to spread much faster than ever. Anyone can create or share information that may be misleading with no relevance to reality. Even an expert in a particular domain has to explore multiple aspects before writing an article and posting it across social media. Fake news can also reduce the impact of real news by competing with it.

Several factors have been implicated in the spread of fake news such as motivated reasoning, confirmation bias, political polarisation etc.

Source - https://www.sciencedirect.com/science/article/pii/S1877050917323086

3. Project Motivation

Fake news can reduce the impact of real news by competing with it. The "Fake" term in fake news can include various terms such as - misleading & manipulated content, false & fabricated content, scientific denialism. There have been many instances where fake news has caused major issues in creating the instability in the civilized society.

Let's take some examples of recent fake news during the ongoing COVID-19 pandemic:-

- Misinformation on fake Covid19 cures such as gargling with lemon water and injecting yourself with bleach.
- Virus being bioengineered in a lab in Wuhan or 5g cellular network is causing COVID-19 symptoms.
- The "Anti-Vaxx movement" related to Covid vaccines inefficiency and spread of mis-information on social media.

Source- https://www.frontiersin.org/articles/10.3389/fpsyg.2020.566790/full

What is the solution?

The solution would be to catch these fake news from spreading on the digital media as social network and the internet are one of the fastest source of conveying information in recent times. The solution that I can think of is to create a tax classifier based on NLP that distinguish between real and fake news. Then we can find a way to integrate this text classifier in search engine or social media applications to detect whether the information that people post are real or not. The dataset used in the classifier should also free from political beliefs.

What is Text classifier?

Text classification is the process of categorizing text into organized groups. It can automatically analyse text and then group then based on their categories or tags.

II. Literature Review

According to "**Fake News Detection Using Machine Learning Ensemble Methods**" while building a fake news detection classifier, there have been multiple instances where both supervised and unsupervised learning algorithms are used to classify text. However, most of the literature focuses on specific datasets or domains, and most of them belong to the politics domain.

Therefore, the algorithm trained works best on a particular domain but does not achieve optimal results when other domains are used with the same algorithm. Since articles from different domains have unique textual structures, it is difficult to train a generic algorithm that works best on all particular news domains.[1]

So to counter this shortcoming, they proposed an interesting solution i.e. to use a machine learning ensemble approach that was not thoroughly explored before.

The ensemble techniques used are bagging, boosting, and voting classifiers are explored to evaluate the performance over the multiple datasets. There were two different classifiers used that comprises of three learning models:

- ensemble of logistic regression, random forest, and KNN,
- voting classifier, consists of linear SVM, logistic regression, classification and regression trees.

The dataset they used contains both true and fake articles extracted from the World Wide Web. The true articles are extracted from reuters.com, while the fake articles were extracted from multiple sources, mostly websites which are flagged by politifact.com.

For voting classifier, they have trained the individual model and then tested the model on the basis of major votes by all three models. For bagging ensemble, the training dataset consisting of 100 decision trees is used, two boosting ensemble algorithms are used, XGBoost and AdaBoost. A k-fold (k = 10) cross-validation model is used for all ensemble learners.

The evaluation is done on 4 standard performance metrices - accuracy, precision, recall, and F1 score.

According to "**Rapid detection of fake news based on machine learning methods**", in order to attract the attention of recipients, a common technique used to spread fake news is to include catchy headlines, the so-called clickbait. They plan to develop a new model for the quick discovery of fake news based on the news title without the need to analyze the whole content of the article.

They used Ensemble methods by repeatedly running the base algorithm and formulating a vote based on the resulting hypotheses. Popular representatives of aggregation methods for groups of classifiers are bagging, boosting, SVM and random forests algorithms.

The methods and techniques involve data preprocessing which is performed at the start before classification. The data is retrieved and cleaned, including making all characters the same case and removing digits and non-alphabetic characters. Then tokenize text is performed to remove stop words. The second step is to use the dataset to learn algorithms. [2]

For this purpose, the mixed data of real and fake news is divided into a training set and a test set. The selected algorithm is trained on training data, and the resulting classifier is used to predict new data (a test set is used to determine its quality of classification).

All operations were running on Google Colab, which may be used for free. Resources were shared in the Colab tool but initially during execution, an environment containing 12 GB of RAM and an Intel Xeon processor within two cores clocked at 2.30GHz, were assigned.

While experimenting with different algorithms such as Random forest, SVM and ensemble approaches, they found out that SVM outperforms in every category whether it is to classify fake news on the basis of title or news content but with a drawback. The drawback is the running time of the algorithm which makes it substantially slower than the rest of them. They found out that the random forest algorithm, on the other hand, is a good balance between time and quality of classification with a slight trade-off of quality of classification.

According to "**Detecting Fake News in Social Media Networks**", The solution would be to design a tool with the aim of detecting and eliminating web pages that contain misinformation to mislead readers. They think that the user will have to download that tool and install it on a personal computer or make it compatible with the browser data commonly use all across the world. The working example of the tool is if the user tries to search a group of search terms on the web to fill run its operation and search for those dumb through the database and filter out misleading websites from the users. [3]

The methodology used was to find credible, clickbait databases, and then crawl the web to collect URLs. They mainly focus on social media websites such as Facebook, Reddit etc to find fake news and clickbait articles.

They use various algorithms such as a random tree, logistic regression, Naive Bayes, etc. to solve the classification problem.

They concluded that fake news and clickbait articles interfere with the ability of users to search for information on the internet when they are required to make critical decisions. They say that allowing users to install a simple tool in their personal browser and use it to detect and filter out potential clickbait and fake news work efficiently. The tools they created have shown outstanding performance in identifying the possible source of fake news. They’re planning to expand the database for fake news along with their approach to move it to make it more effective against new data sets.

According to "**Fake News Detection on Social Media: A Data Mining Perspective**", They think that fake news is intentionally written to mislead readers and alter their decision-making skills. They discussed solving the problem first by characterisation and then detection.

In characterisation, they discuss the basic social and psychological theories related to fake news, fabrications, hoaxes and satire.

After that, they used detection techniques using a knowledge-based, stance-based, style based etc. they discussed the existing fake news detection approaches from a data mining perspective, feature extraction and model construction.

From the above four works of literature, what I found is there are multiple ways to create a text classifier. But the accuracy will depend on the algorithm used along with the dataset that algorithm is trained on. We can't expect a classifier that performs really well with a dataset with only political news will also perform well on the datasets with other domains due to unique textural structures.

The ensemble method seems to work well with greater accuracy. This involves using two or more algorithms to arrive at a predictive model. SVM tends to outperform both Random forest and ensemble learning but it is quite slow with maybe quite a drawback for most people.

Citations

[1] Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf, and Muhammad Ovais Ahmad (2020), "Fake News Detection Using Machine Learning Ensemble Methods", https://www.hindawi.com/journals/complexity/2020/8885861/

[2] Barbara Probierza, Piotr Stefa Nskia, Jan Kozak (January 2021) "Rapid detection of fake news based on machine learning methods", https://www.researchgate.net/publication/355029017_Rapid_detection_of_fake_news_based_on_machine_learning_methods

[3] Monther Aldwairi, Ali Alwahedi (2018), "Detecting Fake News in Social Media Networks", https://www.sciencedirect.com/science/article/pii/S1877050918318210

[4] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu, "Fake News Detection on Social Media: A Data Mining Perspective", https://arxiv.org/pdf/1708.01967.pdf

Motivation from the literature

I initially thought of including couple of algorithms like Naive Bayes to compare the efficiency and performance of the algorithms but after going through the above literature, I now will also include several algorithm and ensemble learning to see and analyse the results with diverse dataset containing data from multiple domains. This will help me to find the best algorithm for my classifier. I also got to know a

diverse dataset is very essential. The dataset should comprise multiple domains with fake and real news to have equal distribution to have better accuracy and avoid overfitting.

III. Design Document

Objective

The objective of this project is to build a Text classifier to detect fake news from a given range of dataset. Later, we will do the sentiment analysis of the news dataset to know how the society perceive the information that they see on the internet. We will use various Natural Language Processing (NLP) techniques to solve this problem.

The project is not limited to identifying fake or real news but this text classification can also be used in other projects such as Spam filtering, sentiment analysis, topic labelling etc.

I have identified that there have been many instances where fake news spread at a lot faster rate than real news due to the rise of social media platforms and the internet in general. so it becomes essential for us to catch the fake news as early as possible. this text classifier will help to identify whether the news is fake or real.

Some examples that are already discussed in the introduction section are - the news regarding viruses being bioengineered in Wuhan labs along with the miss information on fake cures from covid etc.

After developing this text classifier, if successful we will be able to make a significant contribution to solving this problem. Some of these are as follows-

1. Fake news can be detected early on when it is published and can be corrected
2. The fake news can be taken down if many people or aware of the fake news on social media platforms
3. The users will be able to distinguish between fake and real news by using this text Classifier.
4. The panic situation created by this fake news can be avoided.
5. The situation regarding fake news can cause affect people's health. By combating fake news it can beneficial for society.

Project Features list and techniques

1. Pre-processing the data set which includes importing the data set yes checking if the dataset is balanced or not. cleaning the dataset which involves removing stop words, punctuations text normalisation such as Lemmatization, stemming, Analysing the clean dataset using lexical diversity, frequency distribution, world cloud, collocations such as Bigrams, trigrams, Visualising the dataset using pandas, matplotlib and seaborn.

2. Building a text classifier by using various algorithms such as Naive Bayes, random forest, logistic regression, ensemble learning by combining multiple algorithms, and LSTM To analyse the accuracy of various algorithms.

3. Evaluating the algorithms using various techniques such as accuracy, precision, recall, F1 score, and confusion matrix. The evaluation techniques will be used for all the algorithms. This will helps us to analyse which algorithms work best for text classification by keeping the balance between speed and Quality of classification.

4. Sentiment analysis will also be done on the datasets to analyse the user sentiments on both fake and real news. This will help us to find how users perceive fake news and whether they're sensitive when listening to fake news.

5. Finally, a conclusion will be made with the results that I have found over my journey of building the fake news text classifier and I will also be reporting which algorithms work the best or maybe a combination of algorithms using an ensemble approach may be best suited for this type of classifier. The conclusion will be made based on the combination of both speed and quality of classification as both parameters are equally important for it to be useful.

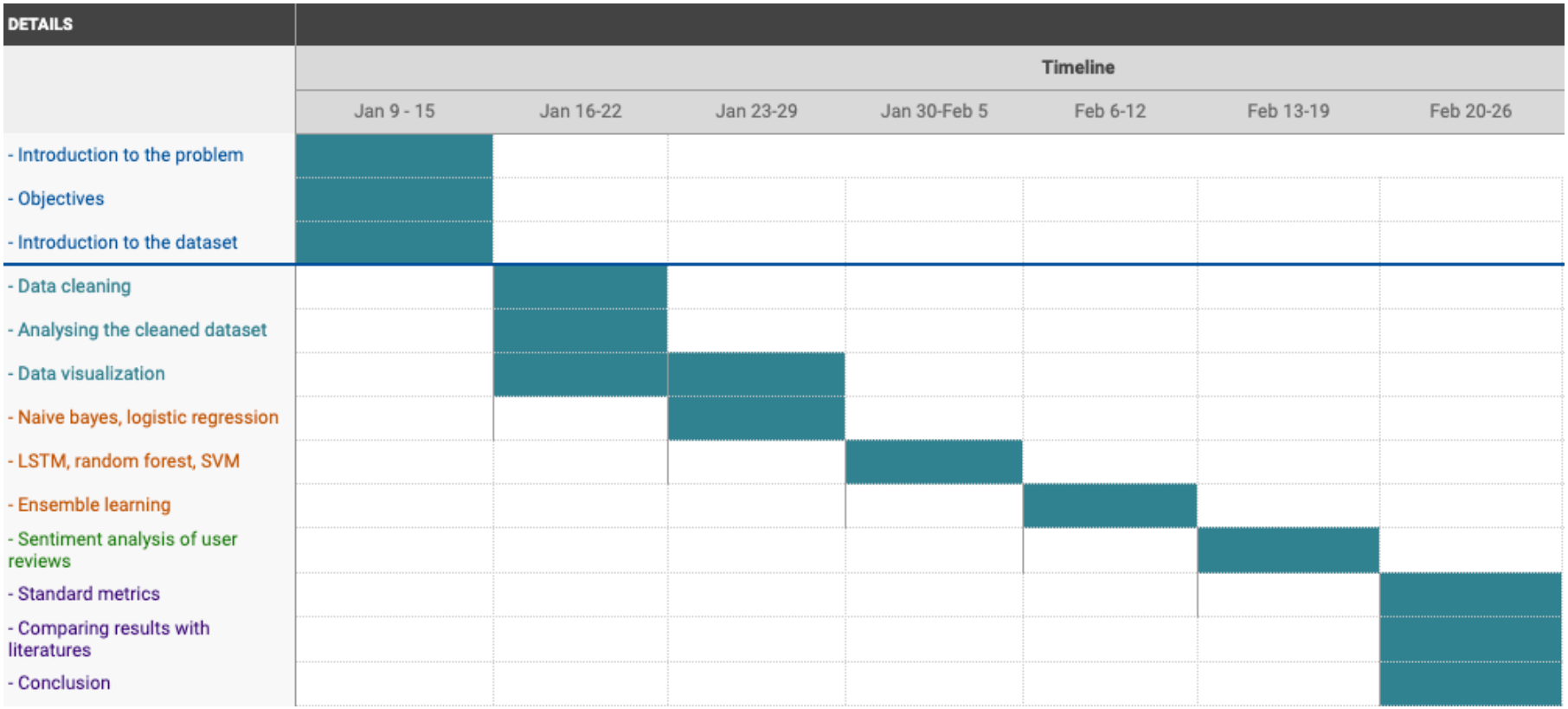
Structure of the project

As have already discussed in detail the features that I will be having in my Project. I will be showcasing the project structure which will tell the step-by-step journey of my project.

1. Preprocessing
    - a. Importing data sets and packages
    - b. visualising the content of the dataset
    - c. Visualising the distribution of the dataset
    - d. Cleaning the dataset (removing punctuation, stop words, duplicates, lemmatisation, stemming etc.
    - e. Analysing the clean data set using film frequency distribution, world cloud, lexical diversity, and collocations.
    - f. Visual analysis of the dataset using Numpy, pandas, seaborn and matplotlib.

2. Building a text classifier using the algorithms such as Naive Bayes, Logistic Regression, KNN, Random Forest, SVC (Sigmoid, Linear, RBF) and Long short-term memory (LSTM).
3. Evaluation using techniques discussed above such as accuracy, precision, recall, F1 score, and confusion matrix.
4. Sentiment analysis on the dataset.
5. Evaluation
6. Conclusions

Work Plan (Gantt chart)



Key technologies

The technologies that I will be using in this project or as follows:-

1. Python as a programming language
2. Jupiter notebook as an environment to develop this project
3. Python libraries and packages such as Numpy, pandas, Seaborn, sklearn, nltk, matplotlib etc.
4. Various algorithms such as Naive Bayes, random forest, LSTM etc.

Evaluation plan

We will be using standard evaluation metrices such as Accuracy, Precision, ecall and F1 score. We will also use Confusion Matrices to visualise the performance of our classification models.

1. **Accuracy** - means how often classifier makes the correct prediction. Accuracy is calculated by taking the ratio of total number of correct predictions to the total number of predictions. Accuracy = TP+TN/TP+FP+FN+TN

2. **Precision** - is the ratio of true positives predicitions to the total number of positives predicitons. It tells us what proportion of news we classifed as fake are actually fake news. Precision = TP/TP+FP

3. **Recall** - is the ratio of correctly positive prediction to the total the total number of predictions. Recall = TP/TP+FN

4. **F1 Score** - is the weighted average of Precision and Recall. F1 score = 2*(Recall* Precision) / (Recall + Precision)

5. **Confusion Matrics** - used to visualise performance of the classification model.

3. Introduction to Dataset

The dataset for the project is "Fake and real news dataset" which can be found on Kaggle.

The dataset is available at the following link - https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset.

The dataset contains 2 csv files - "Fake.csv" (containing fake news) and "True.csv" (containing real news). The size of the files are 62.79 MB and 53.58 MB respectively. Both the files have similar file format and structure each containing 4 columns - title, text, subject and date. The total number of news in "Fake.csv" and "True.csv" are 23,481 and 21,417 respectively.

The description of the each column in the datset

1. title - contains the headline of the news.

2. text - contains the description and reasoning / explanation of the headline.
3. subject - contains the category to which the news belongs to.
4. date - contains the date at which the news is published.

IV. Implemementation

A. Preprocessing

Preprocessing is a set of techniques applied to the raw data before it is used for data analysis. It involves techniques like Data cleaning, data transformation, integrating data from multiple sources etc.

Firstly we are going to import all the necessary libraries and dataset. After importing the dataset, we will check for the number of records of both true and fake news to check if the dataset is balanced before merging them together. We will add a status field in the dataset which tells whether the data in the dataset belongs to the Real or false news before merging.

1. Importing packages & dataset

In [166…

```
import pandas as pd
import re
import string
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# ML packages
import sklearn
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfTransformer
```

In [167…
```
# Loading the csv file with real news
df_true = pd.read_csv("True.csv")
df_true.head()
```

Out[167]:

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

In [168…
```
# checking the total number of records present in True.csv file
len(df_true)
```

Out[168]:
```
21417
```

In [169…
```
# check the shape of the dataset
df_true.shape
```

Out[169]:
```
(21417, 4)
```

In [170…
```
# Loading the csv file with fake news
df_false = pd.read_csv("Fake.csv")
df_false.head()
```

Out[170]:

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017

In [171…
```
# checking the total number of records present in Fake.csv
len(df_false)
```

Out[171]:
```
23481
```

In [172…
```
# check the shape of the dataset
df_false.shape
```

Out[172]:
```
(23481, 4)
```

In [173…
```
# adding a status column to both of the news type. "0" represents fake news and "1" represents true news.

df_true['status'] = 1
df_false['status'] = 0
```

In [174…
```
# Merging the two datasets and shuffling both the datasets randomly
df = pd.concat([df_true, df_false], sort=False).sample(frac=1)
```

In [175…
```
df.head()
```

Out[175]:

	title	text	subject	date	status
19216	BOOM! Companies That Openly Criticized Trump F...	Among the high profile companies opposing Trum...	left-news	Jan 31, 2017	0
4532	More Democratic senators oppose Trump's U.S. S...	WASHINGTON (Reuters) - Senate Democrats on Fri...	politicsNews	March 31, 2017	1
7511	Hate speech seeps into U.S. mainstream amid bi...	KOKOMO, Indiana (Reuters) - The lettering is c...	politicsNews	November 7, 2016	1
7826	Canadian court rules Trump can face claims in ...	TORONTO (Reuters) - U.S. Republican presidenti...	politicsNews	October 13, 2016	1
6478	WATCH: SNL's Church Lady Returns To Hilarious...	Fresh off of being compared to Satan by John B...	News	May 8, 2016	0

In [176…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 44898 entries, 19216 to 4775
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    44898 non-null  object
 1   text     44898 non-null  object
 2   subject  44898 non-null  object
 3   date     44898 non-null  object
 4   status   44898 non-null  int64 
dtypes: int64(1), object(4)
memory usage: 2.1+ MB
```

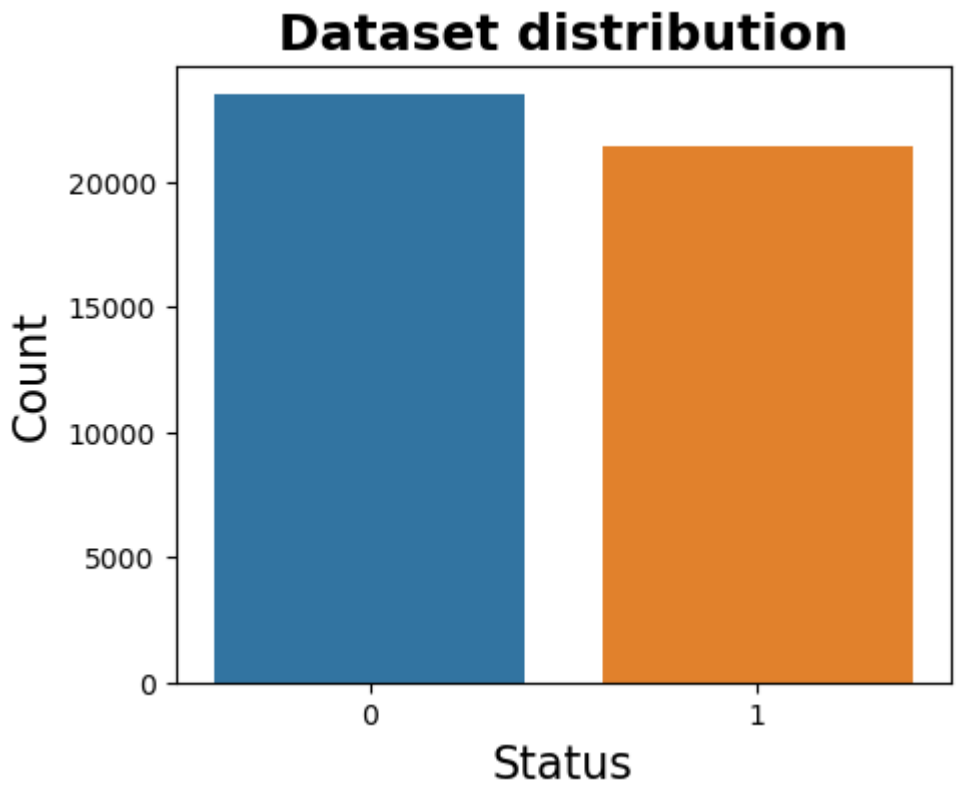
2. Visualising the dataset distribution

Data visualization means to represent data in a visual form, such as charts, graphs etc. The goal of data visualization is to make complex data easier to understand and share with others who may not even have much knowledge orr expertise to analyse data in the tabular form.

What we are going to do here is to represent the data using its status field to check the count of both fake and real news. Then we will check the total count of news in various categories and plot the chart.

In [177…

```
# Plotting the merged dataset and displaying the bar graph based on their status
# "0" represents fake news & "1" represents real news.

plt.figure(figsize= (5,4))
sns.countplot(x = 'status', data=df, fill=True)
plt.title('Dataset distribution', fontdict={'fontweight': 'bold', 'fontsize': 18})
plt.xlabel('Status', fontdict={'fontsize': 16})
plt.ylabel('Count', fontdict={'fontsize': 16})
plt.show()
```



The dataset is almost balanced. Both fake & real news dataset are in equal distribution to further analyze the dataset.

In [178…
```
# looking for the counts on the basis of news subject
df.subject.value_counts()
```

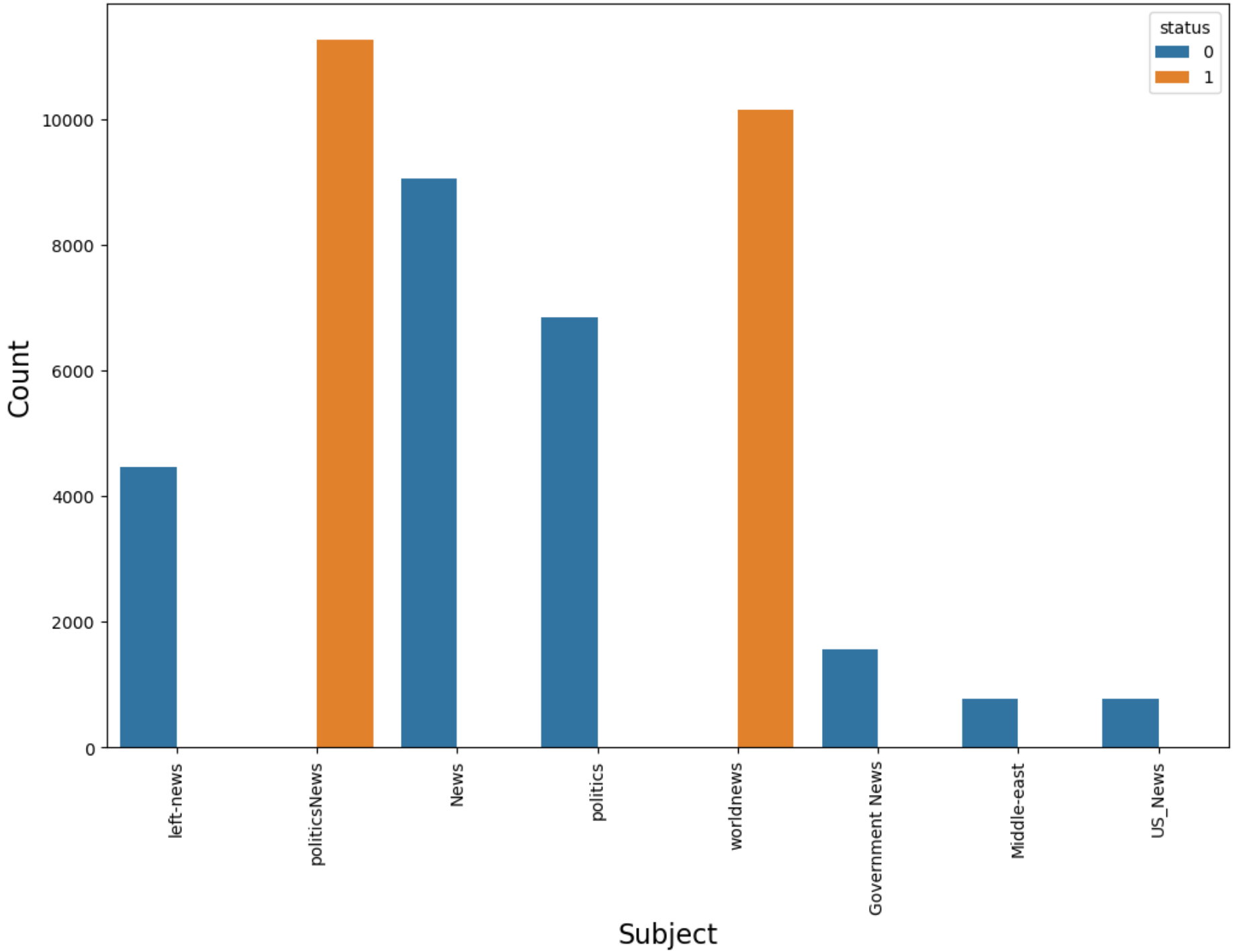
Out[178]:
```
politicsNews       11272
worldnews          10145
News                9050
politics            6841
left-news           4459
Government News     1570
US_News              783
Middle-east          778
Name: subject, dtype: int64
```

In [179…
```
plt.figure(figsize = (12,8))
bar_chart = sns.countplot(x = "subject", hue = "status" , data = df)
```

```
plt.xlabel('Subject', fontdict={'fontsize': 16})
plt.ylabel('Count', fontdict={'fontsize': 16})
bar_chart.set_xticklabels(bar_chart.get_xticklabels(),rotation=90)
```

Out[179]:

```
[Text(0, 0, 'left-news'),
 Text(1, 0, 'politicsNews'),
 Text(2, 0, 'News'),
 Text(3, 0, 'politics'),
 Text(4, 0, 'worldnews'),
 Text(5, 0, 'Government News'),
 Text(6, 0, 'Middle-east'),
 Text(7, 0, 'US_News')]
```


status
0
1
Count
10000
8000
6000
4000
2000
0
left-news
politicsNews
News
politics
worldnews
Government News
Middle-east
US_News
Subject

3. Cleaning the dataset

Cleaning the dataset refers to the process of identifying and removing inconsistencies, special characters, html entities, duplicates and stopwords. We do this to ensure that the dataset is accurate and consistent before it is ready to use for analysis.

The first task in did a cleaning that we are going to do is to combine title, text, subject fields in a dataset to have one field that contains all the information regarding the news.

Then we check and remove the duplicates present in the dataset. The next task would be to remove the punctuation marks, special characters and stop words as they don't add any value to a sentence and can safely be ignored. Finally, we will to text normalization on every news in our dataset to reduce reducing a word to its base form.

In [180…
```
# Merging the text data into one column for classification purposes
df['text'] = df['title'] + " " + df['text'] + df['subject']

# Removing the data that we won't use for the analysis
df.drop(columns=['title', 'subject'],inplace=True)
```

In [181…
```
# shape of the dataset
df.shape
```

Out[181]:
```
(44898, 3)
```

In [182…
```
# checking for duplicates in the dataset
df.duplicated().sum()
```

Out[182]:
```
209
```

In [183…
```
# remove duplicates
df = df.drop_duplicates(keep='first')
```

In [184…
```
df.duplicated().sum()
```

Out[184]:
```
0
```

In [185…
```
# converting string to lowercase.

df = df.astype(str).apply(lambda x: x.str.lower())
```

In [186…
```
df.head()
```

Out[186]:

	text	date	status
19216	boom! companies that openly criticized trump f...	jan 31, 2017	0
4532	more democratic senators oppose trump's u.s. s...	march 31, 2017	1
7511	hate speech seeps into u.s. mainstream amid bi...	november 7, 2016	1
7826	canadian court rules trump can face claims in ...	october 13, 2016	1
6478	watch: snl’s church lady returns to hilarious...	may 8, 2016	0

In [187…
```
# Lets look at the sample text
df['text'].iloc[0]
```

Out[187]: 'boom! companies that openly criticized trump for “making america safe again” take stock market hit among the high pr
ofile companies opposing trump s seven muslim country refugee and immigration ban: amazon.com (nasdaq: amzn) was dow
n 0.83 percent. amazon has big operations in the phoenix area. amazon ceo jeff bezos bezos sent a
memo to all of his employees stating, this executive order is one we do not support, and the memo listed several ac
tions the company was taking to opposed the order. we re a nation of immigrants whose diverse backgrounds, ideas, an
d points of view have helped us build and invent as a nation for over 240 years . it s a distinctive competitive adva
ntage for our country one we should not weaken. apple (nasdaq: aapl) was down 0.23 percent, apple reported strong ea
rnings tuesday afternoon. its shares were up in after-hours trading. apple has a big data center in mesa. starbucks
(nasdaq: sbux) was down 1.22 percent closing at $55.22 per share, according to google finance. starbucks has an educa
tion partnership with arizona state university.starbucks ceo howard schultz we are living in an unprecedented time,
schultz said in a memo to starbucks (sbux) employees. he pledged to hire 10,000 refugees over five years in the 75 co
untries where starbucks does business to reinforce our belief in our partners around the world. yesterday a huge #bo
ycottstarbucks campaign was started on twitter and facebook. starbucks took a big hit in the stock market today:here
s what we don t need advice from america s wealthiest welfare recipient, tesla s elon musk:the blanket entry ban on c
itizens from certain primarily muslim countries is not the best way to address the country s challenges elon musk (@
elonmusk) january 29, 2017tesla s elon musk criticized trump s temporary travel ban. musk has built a multibillion-do
llar fortune running companies that make electric cars, sell solar panels and launch rockets into space.and he s buil
t those companies with the help of billions in government subsidies. tesla motors inc., solarcity corp. and space exp
loration technologies corp., known as spacex, together have benefited from an estimated $4.9 billion in government su
pport, according to data compiled by the la times. microsoft (nasdaq: msft) was down 0.74 percent.microsoft said it
s providing legal advice and assistance to its employees affected by the executive order. we share the concerns about
the impact of the executive order on our employees from the listed countries, all of whom have been in the united sta
tes lawfully, the tech giant said in a statement.according to microsoft s general counsel brad smith, 76 microsoft e
mployees are citizens with a u.s. visa from the affected countries.mastercard inc. ma, +0.02% ceo ajay banga, who was
born in india, sent an email to company employees expressing his deep concern over the fracturing society. according
to the wall street journal, banga said mastercard has been in close contact with employees who ve been affected by th
e ban and is working to help them and their families. what affects one of us, affects all of us, he wrote. goldman
sachs (nyse: gs) one of the few wall street firms to speak out against trump s orders was down 1.96 percent closi
ng at $229.32 per share. goldman was one of the key drivers for the post-election rally and the dow hitting 20,000 la
st week. early twitter inc. twtr, -0.17% and uber investor chris sacca said he would match donations to the aclu up
to $75,000.twitter ceo jack dorsey had this to say about trump s temporary travel ban:the executive order\'s humanita
rian and economic impact is real and upsetting. we benefit from what refugees and immigrants bring to the u.s. http
s://t.co/hdwvgziect jack (@jack) january 28, 2017uber investor chris sacca offered to help the aclu with any legal c
harges incurred in their fight against making america more secure:i\'m inspired by all who are barely scraping by yet
still giving monthly to the @aclu. show me your receipts and i\'ll match \'em to $75k. https://t.co/dej1dxag3a chri
s sacca (@sacca) january 28, 2017 facebook (nasdaq: fb) was 0.5 percent. yeah thanks for your input mark. we re pret
ty sure everyone is aware by now of your leftist political leanings. why didn t you make the same proclamation when o
bama banned iraqi s from coming to the us in 2011? no need to answer, it s a rhetorical question netflix inc. (nasda
q: nflx) was down 0.36 percent. netflix inc. nflx, -0.08% chief executive reed hastings, in reaction, said on faceboo
k that it had been a very sad week. google parent alphabet (nasdaq; googl) was down 0.44 percent.in a staff memo, g
oogle ceo sundar pichai said the move affects at least 187 of the internet giant s staff. we re concerned about the i
mpact of this order and any proposals that could impose restrictions on googlers and their families, or that could cr
eate barriers to bringing great talent to the u.s., google said in a statement. we ll continue to make our views on
these issues known to leaders in washington and elsewhere. companies and executives not known for speaking out on pol

```
itical matters, such as nike inc. nke, -0.38% ceo mark parker, condemned the ban. in an internal letter parker mentio
ned nike athlete sir mo farah, a somali-born olympic gold medalist now living in oregon. what mo will always have  w
hat the entire nike family can always count on  is the support of this company. we will do everything in our power t
o ensure the safety of every member of our family: our colleagues, our athletes and their loved ones,  parker s email
read.ford, nyse was down .08% ford executive chairman bill ford and ceo mark fields in a memo to employees, they said
they do not support the ban.  respect for all people is a core value of ford motor company, and we are proud of the r
ich diversity of our company here at home and around the world,  they wrote.coca-cola chairman and ceo muhtar kent ke
nt said in a statement that  the coca cola (cchgy) company is resolute in its commitment to diversity, fairness and i
nclusion, and we do not support this travel ban or any policy that is contrary to our core values and beliefs. choban
i is not traded on the stock market, but is one of the biggest advocates for importing employers to the us from musli
m majority nations to twin falls, id                chobani ceo hamdi ulukayachobani is owned by turkish muslim immig
rant hamdi ulukaya. chobani has filled 30 percent of its 600 positions at the world s largest yogurt plant in twin fa
lls, idaho, with refugees resettled in america through a u.s. state department program carried out in cooperation wit
h the united nations.ann corcoran, author of the refugee resettlement watch blog, said the potential conflict of inte
rest is disturbing and should be questioned by twin falls residents. twin falls is really a microcosm of what we find
going on in so many of the refugee communities across the u.s., where you have people moving in and out of government
and the chamber of commerce with a vested interest in making sure a meatpacking plant or some other industry has cont
inuous access to refugee labor,  said corcoran.  only in this case we have a blatant example of conflicts of interest
by an elected official who is also the head of the chamber enticing companies to come in and make use of the steady i
nflux of cheap, overseas labor. these are jobs that americans would be happy to fill but they are forced to compete n
ow with someone from sudan or iraq who is used to working for a dollar a week. the local muslim community in twin fal
ls grew out of its mosque and built a new, much larger house of worship last year.here s chobani ceo s response to tr
ump s moratorium on refugees from 7 countries:  this is very personal for me,  ulukaya wrote in internal memo to his
staff that was obtained by cnn.  as an immigrant who came to this country looking for opportunity, it s very difficul
t to think about and imagine what millions of people around the world must be feeling right now. general electric ge
(nyse) stock was down .87%          ceo jeff immelt said, in a memo on the g (ge)employee blog, that he shares the
concern  felt by his employees and said the company has many employees from the countries named in the ban.  these em
ployees and customers are critical to our success and they are our friends and partners,  he wrote, adding that ge wo
uld  stand with them  and try to find a balance between security and  movement of law abiding people.  immelt was one
of 28 business leaders named to a council to advise trump on manufacturing growth.trip advisor ceo stephen kaufer tri
p advisor s (trip) ceo wrote in a linkedin post that trump s immigration ban is  not only heartless and discriminator
y, but also against the principles that make our country great.  kaufer also said in a tweet that  we need to do mor
e, not less, to help refugees,  and said the action was  wrong on humanitarian grounds, legal grounds, and won t make
us  safer.\'  in a separate tweet, he called out republican lawmakers:  you can t sit this one out. we need to do mor
e, not less, to help refugees. trumps action was wrong on humanitarian grounds, legal grounds, and won\'t make us "sa
fer."  stephen kaufer (@kaufer) january 29, 2017trump hasn t backed off his order and could change up visa programs u
sed by high-tech companies. biz journals, market watchleft-news'
```

Removal of Punctuation Marks, Special Characters and Stopwords

Stopwords, Punctuation marks and Special characters does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence.

In [188…

```
# removing stopwords and punctuations
from nltk.corpus import stopwords

stopwords = stopwords.words('english')


def remove_stopwords_and_punctuation(sent):
    lst = []
    for word in sent.split():
        if word not in stopwords and word not in string.punctuation and re.search('[a-zA-Z]', word):
            lst.append(word)

    return " ".join(lst)


def removing_unwanted_characters(sent):

    # remove unwanted characters
    sent = re.sub(r'[^\w\s]', ' ', sent)
    sent = re.sub(r'\[[^]]*\]', '', sent)
    sent = re.sub(r'http\S+', '', sent)

    return sent
```

In [189…

```
df['text'] = df['text'].apply(remove_stopwords_and_punctuation)
```

In [190…

```
df['text'] = df['text'].apply(removing_unwanted_characters)
```

In [191…

```
df.head()
```

Out[191]:

	text	date	status
19216	boom companies openly criticized trump makin...	jan 31, 2017	0
4532	democratic senators oppose trump s u s suprem...	march 31, 2017	1
7511	hate speech seeps u s mainstream amid bitter ...	november 7, 2016	1
7826	canadian court rules trump face claims toronto...	october 13, 2016	1
6478	watch snl s church lady returns hilariously m...	may 8, 2016	0

In [192…

```
# Lets look at the sample text now after removing stopwords, punctuations and unwanted characters
df['text'].iloc[0]
```

Out[192]: 'boom companies openly criticized trump making america safe again take stock market hit among high profile companies opposing trump seven muslim country refugee immigration ban amazon com nasdaq amzn percent amazon big operations phoenix area amazon ceo jeff bezos bezos sent memo employees stating executive order one support memo listed several actions company taking opposed order nation immigrants whose diverse backgrounds ideas points view helped us build invent nation years distinctive competitive advantage country one weaken apple nasdaq aapl percent apple reported strong earnings tuesday afternoon shares after hours trading apple big data center mesa starbucks nasdaq sbux percent closing per share according google finance starbucks education partnership arizona state university starbucks ceo howard schultz living unprecedented time schultz said memo starbucks sbux employees pledged hire refugees five years countries starbucks business reinforce belief partners around world yesterday huge boycottstarbucks campaign started twitter facebook starbucks took big hit stock market today here need advice america wealthiest welfare recipient tesla elon musk the blanket entry ban citizens certain primarily muslim countries best way address country challenges elon musk elonmusk january 2017tesla elon musk criticized trump temporary travel ban musk built multibillion dollar fortune running companies make electric cars sell solar panels launch rockets space and built companies help billions government subsidies tesla motors inc solarcity corp space exploration technologies corp known spacex together benefited estimated billion government support according data compiled la times microsoft nasdaq msft percent microsoft said providing legal advice assistance employees affected executive order share concerns impact executive order employees listed countries united states lawfully tech giant said statement according microsoft general counsel brad smith microsoft employees citizens u s visa affected countries mastercard inc ma ceo ajay banga born india sent email company employees expressing deep concern fracturing society according wall street journal banga said mastercard close contact employees affected ban working help families affects one us affects us wrote goldman sachs nyse gs one wall street firms speak trump orders percent closing per share goldman one key drivers post election rally dow hitting last week early twitter inc twtr uber investor chris sacca said would match donations aclu 75 000 twitter ceo jack dorsey say trump temporary travel ban the executive order s humanitarian economic impact real upsetting benefit refugees immigrants bring u s t co hdwvgziect jack jack january 2017uber investor chris sacca offered help aclu legal charges incurred fight making america secure i m inspired barely scraping yet still giving monthly aclu show receipts i ll match em 75k t co dej1dxag3a chris sacca sacca january facebook nasdaq fb percent yeah thanks input mark pretty sure everyone aware leftist political leanings make proclamation obama banned iraqi coming us need answer rhetorical question netflix inc nasdaq nflx percent netflix inc nflx chief executive reed hastings reaction said facebook sad week google parent alphabet nasdaq googl percent in staff memo google ceo sundar pichai said move affects least internet giant staff concerned impact order proposals could impose restrictions googlers families could create barriers bringing great talent u s google said statement continue make views issues known leaders washington elsewhere companies executives known speaking political matters nike inc nke ceo mark parker condemned ban internal letter parker mentioned nike athlete sir mo farah somali born olympic gold medalist living oregon mo always entire nike family always count support company everything power ensure safety every member family colleagues athletes loved ones parker email read ford nyse ford executive chairman bill ford ceo mark fields memo employees said support ban respect people core value ford motor company proud rich diversity company home around world wrote coca cola chairman ceo muhtar kent kent said statement coca cola cchgy company resolute commitment diversity fairness inclusion support travel ban policy contrary core values beliefs chobani traded stock market one biggest advocates importing employers us muslim majority nations twin falls id chobani ceo hamdi ulukayachobani owned turkish muslim immigrant hamdi ulukaya chobani filled percent positions world largest yogurt plant twin falls idaho refugees resettled america u s state department program carried cooperation united nations ann corcoran author refugee resettlement watch blog said potential conflict interest disturbing questioned twin falls residents twin falls really microcosm find going many refugee communities across u s people moving government chamber commerce vested interest making sure meatpacking plant industry continuous access refugee l

```
abor  said corcoran  case blatant example conflicts interest elected official also head chamber enticing companies co
me make use steady influx cheap  overseas labor  jobs americans would happy fill forced compete someone sudan iraq us
ed working dollar week  local muslim community twin falls grew mosque built new  much larger house worship last year
here chobani ceo response trump moratorium refugees countries  personal me  ulukaya wrote internal memo staff obtaine
d cnn  immigrant came country looking opportunity  difficult think imagine millions people around world must feeling
right now  general electric ge  nyse  stock ceo jeff immelt said  memo g  ge employee blog  shares concern felt emplo
yees said company many employees countries named ban  employees customers critical success friends partners  wrote  a
dding ge would stand try find balance security movement law abiding people  immelt one business leaders named council
advise trump manufacturing growth trip advisor ceo stephen kaufer trip advisor  trip  ceo wrote linkedin post trump i
mmigration ban heartless discriminatory  also principles make country great  kaufer also said tweet need more  less
help refugees  said action wrong humanitarian grounds  legal grounds  make us safer  separate tweet  called republic
an lawmakers  sit one out  need more  less  help refugees  trumps action wrong humanitarian grounds  legal grounds  m
ake us  safer  stephen kaufer  kaufer  january 2017trump backed order could change visa programs used high tech com
panies  biz journals  market watchleft news'
```

Lemmetization

Lemmatization aims to remove inflectional endings and return the base or dictionary form of a word, which is known as the lemma .

In [193…
```python
# Text normalization
import nltk
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

def text_normalization(sent):
    words = nltk.word_tokenize(sent)
    return " ".join([lemmatizer.lemmatize(item) for item in words])
```

In [194…
```python
df['text'] = df['text'].apply(text_normalization)
```

In [195…
```python
df.head()
```

Out[195]:

	text	date	status
19216	boom company openly criticized trump making am...	jan 31, 2017	0
4532	democratic senator oppose trump s u s supreme ...	march 31, 2017	1
7511	hate speech seeps u s mainstream amid bitter c...	november 7, 2016	1
7826	canadian court rule trump face claim toronto t...	october 13, 2016	1
6478	watch snl s church lady return hilariously moc...	may 8, 2016	0

In [196…

```
# Lets look at the sample text now after Lemmetization
df['text'].iloc[0]
```

Out[196]: 'boom company openly criticized trump making america safe again take stock market hit among high profile company oppo
sing trump seven muslim country refugee immigration ban amazon com nasdaq amzn percent amazon big operation phoenix a
rea amazon ceo jeff bezos bezos sent memo employee stating executive order one support memo listed several action com
pany taking opposed order nation immigrant whose diverse background idea point view helped u build invent nation year
distinctive competitive advantage country one weaken apple nasdaq aapl percent apple reported strong earnings tuesday
afternoon share after hour trading apple big data center mesa starbucks nasdaq sbux percent closing per share accordi
ng google finance starbucks education partnership arizona state university starbucks ceo howard schultz living unprec
edented time schultz said memo starbucks sbux employee pledged hire refugee five year country starbucks business rein
force belief partner around world yesterday huge boycottstarbucks campaign started twitter facebook starbucks took bi
g hit stock market today here need advice america wealthiest welfare recipient tesla elon musk the blanket entry ban
citizen certain primarily muslim country best way address country challenge elon musk elonmusk january 2017tesla elon
musk criticized trump temporary travel ban musk built multibillion dollar fortune running company make electric car s
ell solar panel launch rocket space and built company help billion government subsidy tesla motor inc solarcity corp
space exploration technology corp known spacex together benefited estimated billion government support according data
compiled la time microsoft nasdaq msft percent microsoft said providing legal advice assistance employee affected exe
cutive order share concern impact executive order employee listed country united state lawfully tech giant said state
ment according microsoft general counsel brad smith microsoft employee citizen u s visa affected country mastercard i
nc ma ceo ajay banga born india sent email company employee expressing deep concern fracturing society according wall
street journal banga said mastercard close contact employee affected ban working help family affect one u affect u wr
ote goldman sachs nyse g one wall street firm speak trump order percent closing per share goldman one key driver post
election rally dow hitting last week early twitter inc twtr uber investor chris sacca said would match donation aclu
75 000 twitter ceo jack dorsey say trump temporary travel ban the executive order s humanitarian economic impact real
upsetting benefit refugee immigrant bring u s t co hdwvgziect jack jack january 2017uber investor chris sacca offered
help aclu legal charge incurred fight making america secure i m inspired barely scraping yet still giving monthly acl
u show receipt i ll match em 75k t co dej1dxag3a chris sacca sacca january facebook nasdaq fb percent yeah thanks inp
ut mark pretty sure everyone aware leftist political leaning make proclamation obama banned iraqi coming u need answe
r rhetorical question netflix inc nasdaq nflx percent netflix inc nflx chief executive reed hastings reaction said fa
cebook sad week google parent alphabet nasdaq googl percent in staff memo google ceo sundar pichai said move affect l
east internet giant staff concerned impact order proposal could impose restriction googlers family could create barri
er bringing great talent u s google said statement continue make view issue known leader washington elsewhere company
executive known speaking political matter nike inc nke ceo mark parker condemned ban internal letter parker mentioned
nike athlete sir mo farah somali born olympic gold medalist living oregon mo always entire nike family always count s
upport company everything power ensure safety every member family colleague athlete loved one parker email read ford
nyse ford executive chairman bill ford ceo mark field memo employee said support ban respect people core value ford m
otor company proud rich diversity company home around world wrote coca cola chairman ceo muhtar kent kent said statem
ent coca cola cchgy company resolute commitment diversity fairness inclusion support travel ban policy contrary core
value belief chobani traded stock market one biggest advocate importing employer u muslim majority nation twin fall i
d chobani ceo hamdi ulukayachobani owned turkish muslim immigrant hamdi ulukaya chobani filled percent position world
largest yogurt plant twin fall idaho refugee resettled america u s state department program carried cooperation unite
d nation ann corcoran author refugee resettlement watch blog said potential conflict interest disturbing questioned t
win fall resident twin fall really microcosm find going many refugee community across u s people moving government ch
amber commerce vested interest making sure meatpacking plant industry continuous access refugee labor said corcoran c
ase blatant example conflict interest elected official also head chamber enticing company come make use steady influx
cheap overseas labor job american would happy fill forced compete someone sudan iraq used working dollar week local m

```
uslim community twin fall grew mosque built new much larger house worship last year here chobani ceo response trump m
oratorium refugee country personal me ulukaya wrote internal memo staff obtained cnn immigrant came country looking o
pportunity difficult think imagine million people around world must feeling right now general electric ge nyse stock
ceo jeff immelt said memo g ge employee blog share concern felt employee said company many employee country named ban
employee customer critical success friend partner wrote adding ge would stand try find balance security movement law
abiding people immelt one business leader named council advise trump manufacturing growth trip advisor ceo stephen ka
ufer trip advisor trip ceo wrote linkedin post trump immigration ban heartless discriminatory also principle make cou
ntry great kaufer also said tweet need more le help refugee said action wrong humanitarian ground legal ground make u
safer separate tweet called republican lawmaker sit one out need more le help refugee trump action wrong humanitarian
ground legal ground make u safer stephen kaufer kaufer january 2017trump backed order could change visa program used
high tech company biz journal market watchleft news'
```

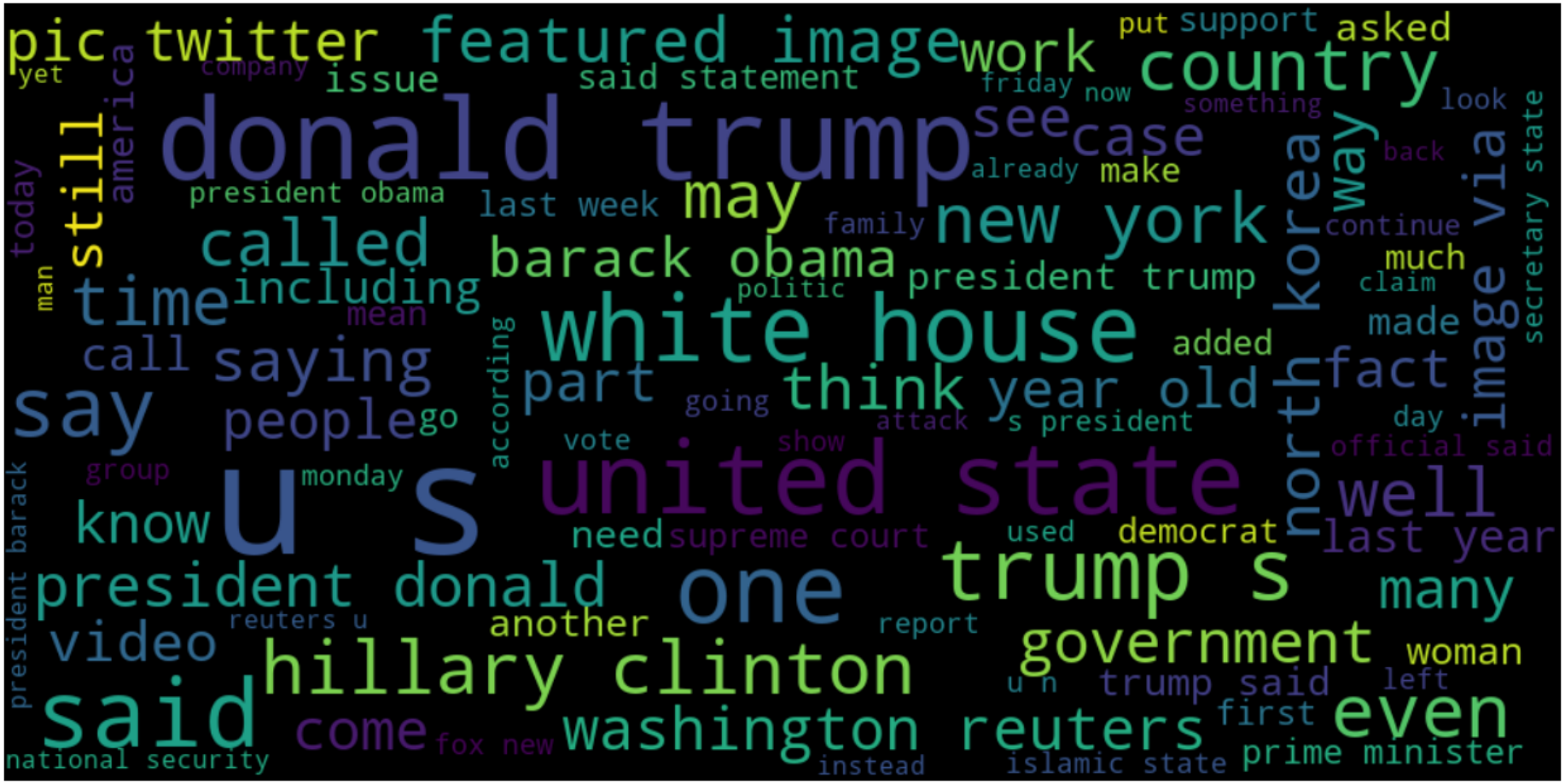
4. Analysing the cleaned dataset

Now, we will start analysing the cleaned dataset. We will first generate Word cloud for both real and fake news. The visualisation may help us to find the words that are frequently used in the dataset. Then we will compare the word count between real and fake news to check if there is a difference in number of words used in both fake and real news. Then, we aim to calculate Lexical Diversity which is a measure of how many different words are used in the text. Frequency distribution is also an important parameter to analyse clean data. We will use frequency distribution to find the frequwency of each word used in the dataset and to list out most common words. Finally, we will do an n-gram analysis to find the words that stick together (collocations) - unigrams, bigrams and trigrams and plot the chart for it.

Word Cloud lists the words used in the combination of both the datasets in a visual form. (Includes both Real & Fake News)

In [197…

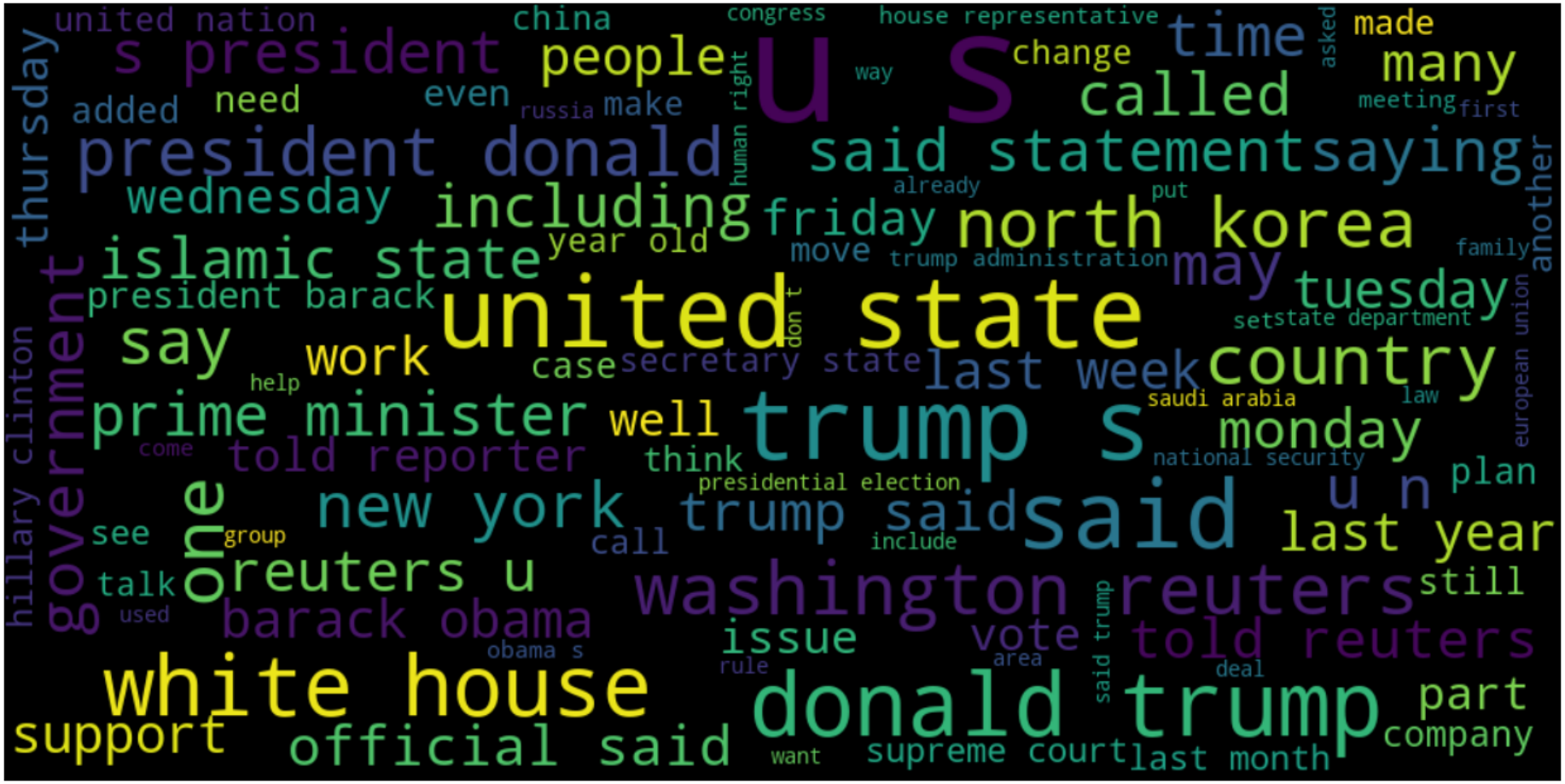
```
# word cloud

from wordcloud import WordCloud, STOPWORDS

wordcloud = WordCloud(max_font_size = 100, max_words = 100 , width = 1000 , height = 500 , stopwords = STOPWORDS).gene

plt.figure(figsize=(16,8))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```



Word Cloud (Only Real News)

In [198…

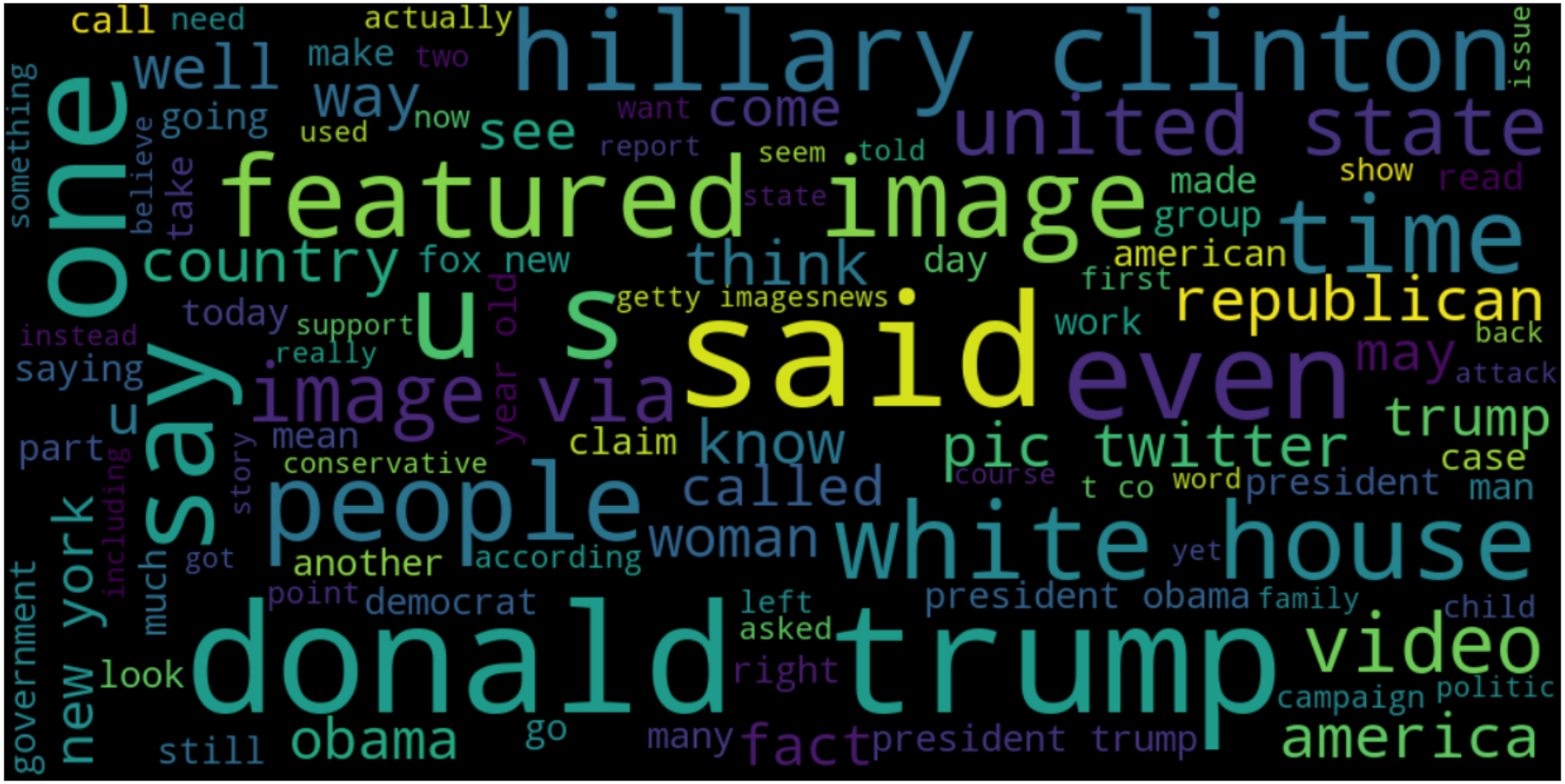
```
# word cloud

from wordcloud import WordCloud, STOPWORDS

wc = WordCloud(max_font_size = 100, max_words = 100 , width = 1000 , height = 500, stopwords = STOPWORDS).generate(" "

plt.figure(figsize=(16,8))
plt.imshow(wc, interpolation="bilinear")
plt.axis("off")
plt.show()
```



Word Cloud (Only Fake News)

In [199…

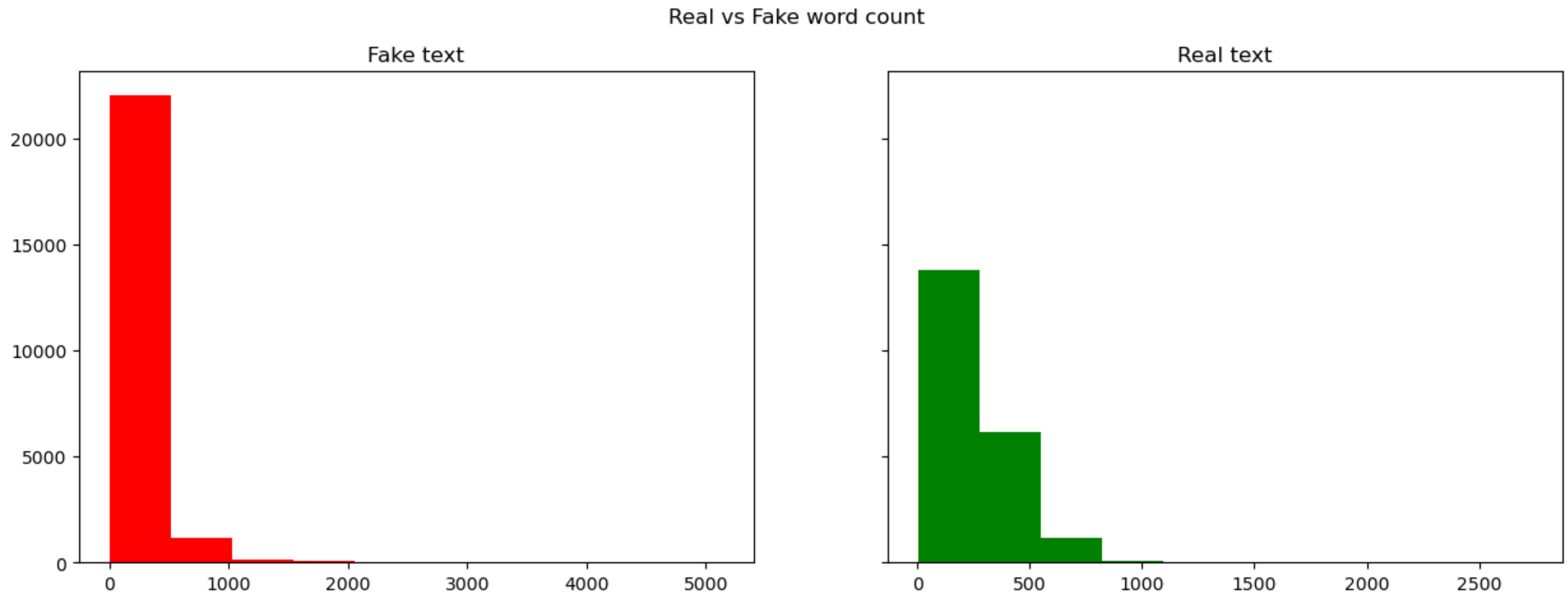
```
# word cloud

from wordcloud import WordCloud, STOPWORDS

wc = WordCloud(max_font_size = 100, max_words = 100 , width = 1000 , height = 500, stopwords = STOPWORDS).generate(" "

plt.figure(figsize=(16,8))
plt.imshow(wc, interpolation="bilinear")
plt.axis("off")
plt.show()
```



Number of words in each text

In [200…

```
fig, axes = plt.subplots(1, 2, figsize=(15, 5), sharey=True)
fig.suptitle('Real vs Fake word count')

text_len=df[df['status']=="0"]['text'].str.split().map(lambda x: len(x))
axes[0].hist(text_len,color='red')
axes[0].set_title('Fake text')

text_len=df[df['status']=="1"]['text'].str.split().map(lambda x: len(x))
axes[1].hist(text_len,color='green')
axes[1].set_title('Real text')
plt.show()
```



Lexical Diversity - is a measure of how many different words are used in the text

In [201…
```
def lexical_diversity(text):
    return len(set(text)) / len(text)
```

In [202…
```
df_str = " ".join(df['text'].tolist())
words_tok = nltk.word_tokenize(df_str)
```

In [203…
```
lexical_diversity(words_tok)
```

Out[203]:
```
0.010013479295034891
```

Frequency Distribution - finds the frequency of words used in the dataset to find the most common words.

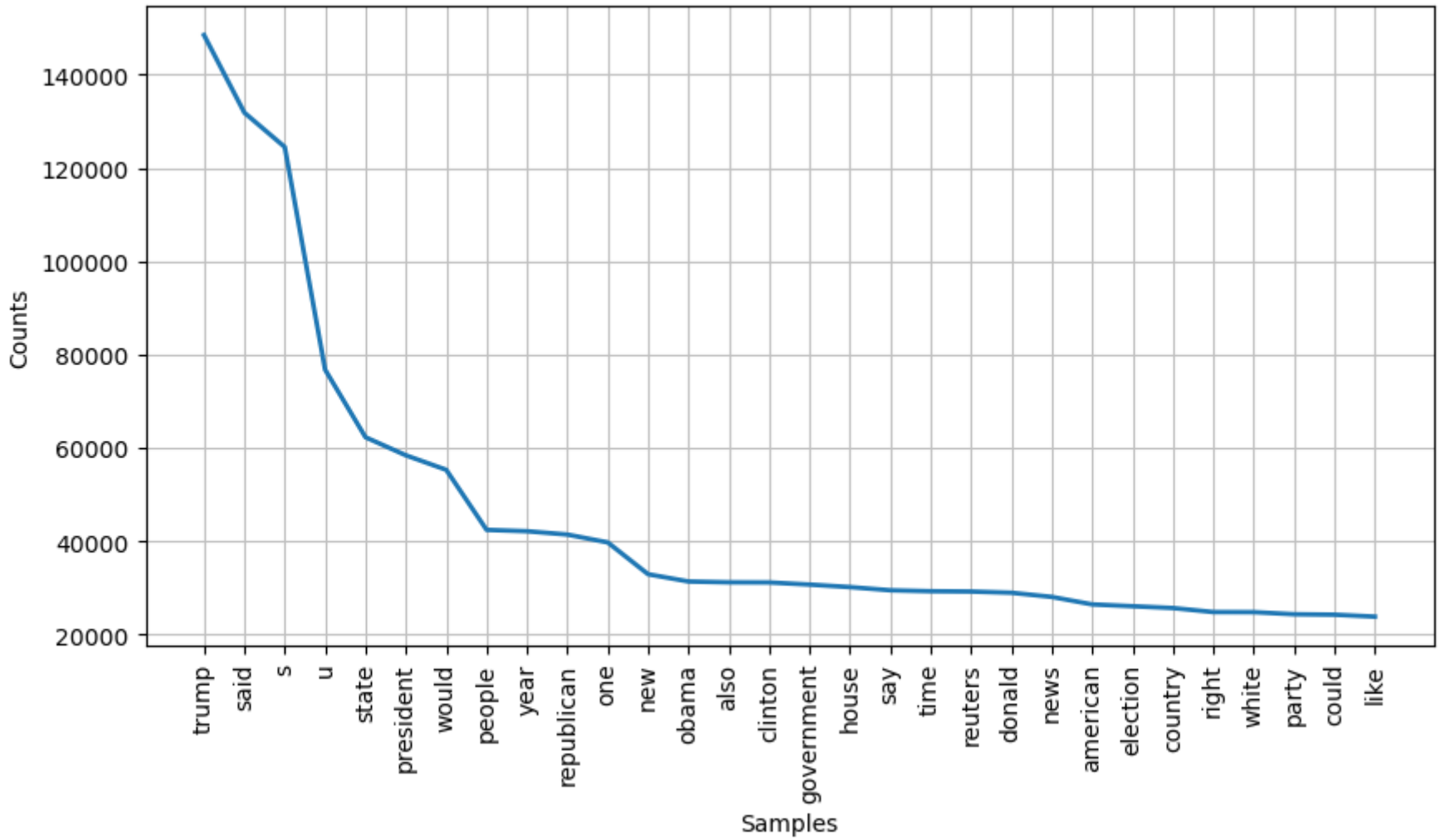
In [204…
```
frequency = nltk.FreqDist(words_tok)
frequency
```

Out[204]:
```
FreqDist({'trump': 148467, 'said': 131837, 's': 124479, 'u': 76765, 'state': 62286, 'president': 58414, 'would': 5530
2, 'people': 42432, 'year': 42145, 'republican': 41431, ...})
```

In [205…
```
# most common words
frequency.most_common(10)
```

Out[205]:
```
[('trump', 148467),
 ('said', 131837),
 ('s', 124479),
 ('u', 76765),
 ('state', 62286),
 ('president', 58414),
 ('would', 55302),
 ('people', 42432),
 ('year', 42145),
 ('republican', 41431)]
```

In [206…
```
# plotting frequency distribution

plt.figure(figsize= (10,5))
frequency.plot(30, cumulative=False)
plt.show()
```



N-grams Analysis

In [207…
```
unigrams = nltk.ngrams(nltk.word_tokenize("".join(df['text'])), 1)
text_unigrams = pd.DataFrame((pd.Series(unigrams).value_counts()), columns = ["unigrams"])
```

In [208…
```
text_unigrams.head(10)
```

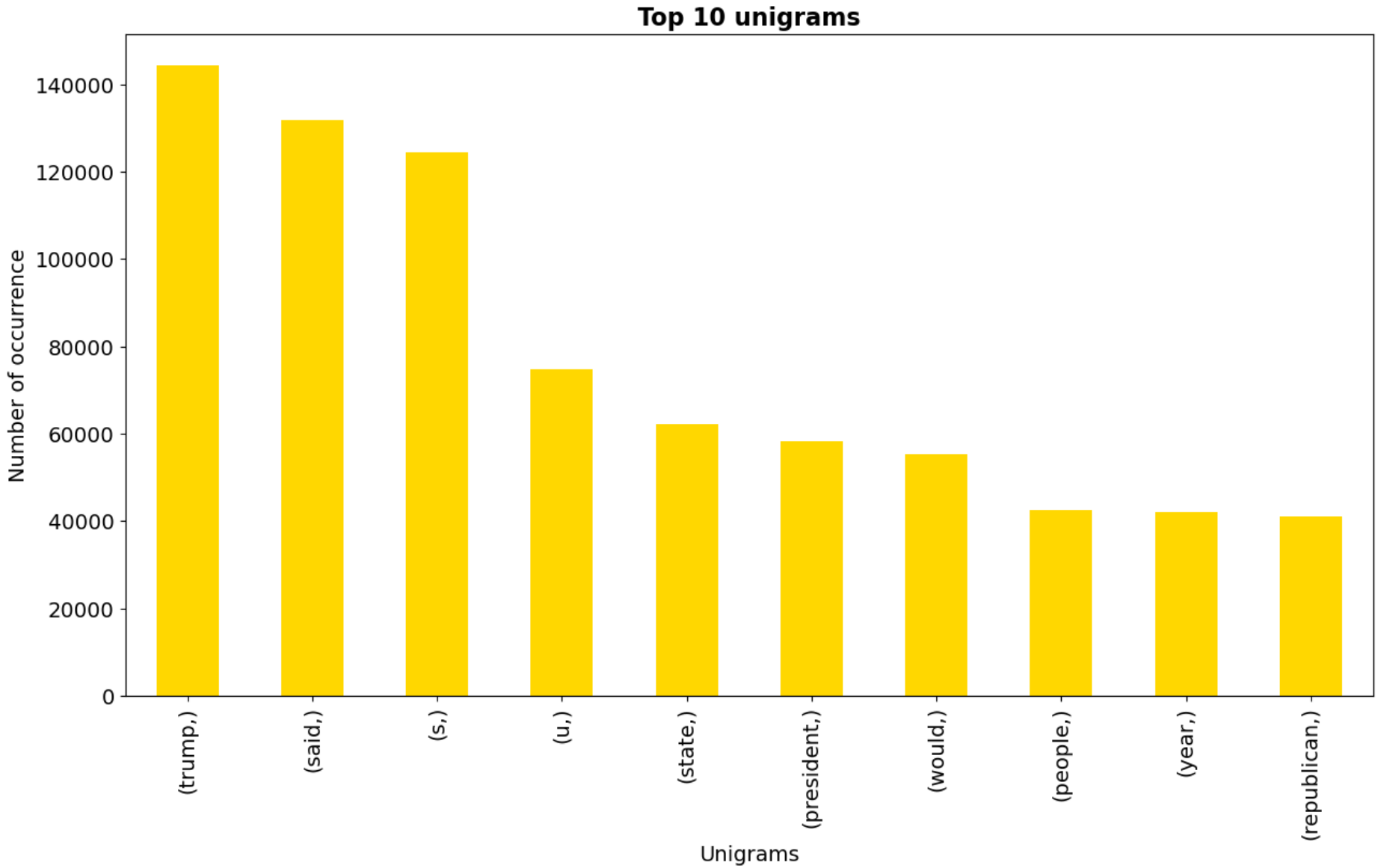
Out[208]:

	unigrams
(trump,)	144292
(said,)	131837
(s,)	124473
(u,)	74785
(state,)	62212
(president,)	58237
(would,)	55289
(people,)	42407
(year,)	42120
(republican,)	40925

In [209…

```
# plotting unigrams

plt.figure(figsize= (15,10))
text_unigrams.head(10).plot(kind = 'bar', figsize  = (15, 8), fontsize = 14, legend = False, color='gold')
plt.title('Top 10 unigrams', fontdict={'fontweight': 'bold', 'fontsize': 16})
plt.xlabel('Unigrams', fontdict={'fontsize': 14})
plt.ylabel('Number of occurrence', fontdict={'fontsize': 14})
plt.show()
```

```
<Figure size 1500x1000 with 0 Axes>
```



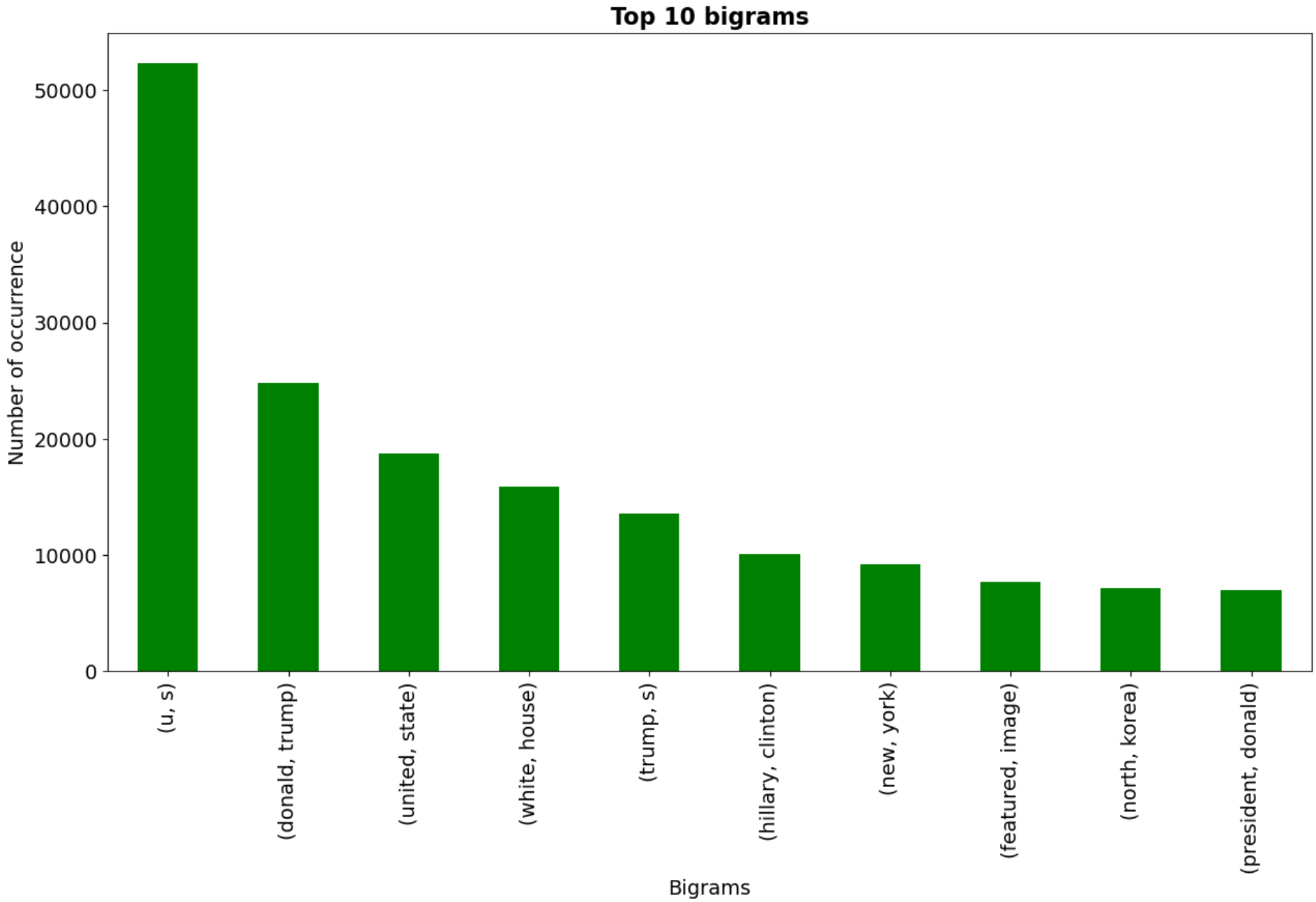
In [210…

```
# bigrams

bigrams = nltk.bigrams(nltk.word_tokenize("".join(df['text'])))
text_bigrams = pd.DataFrame((pd.Series(bigrams).value_counts()), columns = ["bigrams"])
```

In [211…

```
# plotting bigrams

plt.figure(figsize= (15,10))
text_bigrams.head(10).plot(kind = 'bar', figsize  = (15, 8), fontsize = 14, legend = False, color='green')
plt.title('Top 10 bigrams',  fontdict={'fontweight': 'bold', 'fontsize': 16})
plt.xlabel('Bigrams', fontdict={'fontsize': 14})
plt.ylabel('Number of occurrence', fontdict={'fontsize': 14})
plt.show()
```

```
<Figure size 1500x1000 with 0 Axes>
```



In [212…

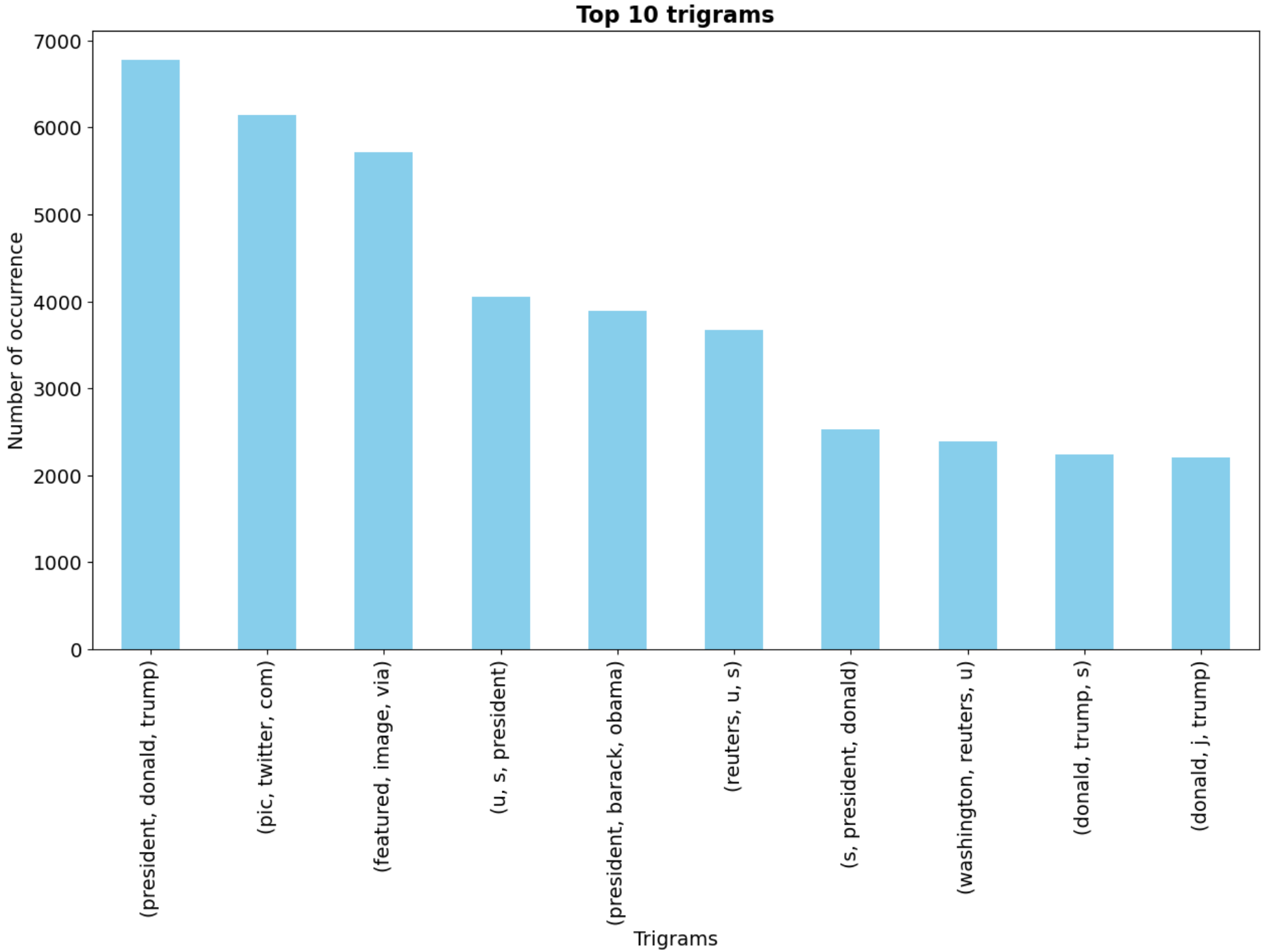
```
# trigrams
```

```
trigrams = nltk.ngrams(nltk.word_tokenize("".join(df['text'])), 3)
text_ngrams = pd.DataFrame((pd.Series(trigrams).value_counts()), columns = ["trigrams"])
```

In [213…

```
# plotting trigrams

plt.figure(figsize= (15,10))
text_ngrams.head(10).plot(kind = 'bar', figsize  = (15, 8), fontsize = 14, legend = False, color='skyblue')
plt.title('Top 10 trigrams', fontdict={'fontweight': 'bold', 'fontsize': 16})
plt.xlabel('Trigrams', fontdict={'fontsize': 14})
plt.ylabel('Number of occurrence', fontdict={'fontsize': 14})
plt.show()
```

```
<Figure size 1500x1000 with 0 Axes>
```


Top 10 trigrams
Number of occurrence
7000
6000
5000
4000
3000
2000
1000
0
(president, donald, trump)
(pic, twitter, com)
(featured, image, via)
(u, s, president)
(president, barack, obama)
(reuters, u, s)
(s, president, donald)
(washington, reuters, u)
(donald, trump, s)
(donald, j, trump)
Trigrams

B. Classification

Now it's time to build a text classifier, we will go through all the algorithms that we've stated during the introduction section. We will initialise features and labels, then do a feature extraction, train the dataset, test the dataset, model the dataset using a given algorithm and finally we will make a prediction. We will also calculate evaluation metrics like accuracy, precision, recall and F1 score but the analysis and evaluation along with confusion matrices will be discussed in the evaluation section.

1. Naive Bayes

Naive Bayes is a supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

Source - https://scikit-learn.org/stable/modules/naive_bayes.html

The steps that we need to follow while building the classifier are as follows:-

1. Initializing X (features) and Y (labels)
2. Feature extraction
3. Dividing data into sets of training and testing.
4. Modeling the dataset using Multinomial Naive implementation.
5. Make Predictions

In [228…
```
# initializing features (X) and labels (Y)
X = df['text']
Y = df['status'].apply(lambda x: int(x)).values
```

In [229…
```
# feature extraction using TfidfVectorizer
tfidf = TfidfVectorizer().fit(X)
X_vect = tfidf.transform(X)
```

In [230…
```
# splitting the dataset between test and train dataset (test size = 20%)
X_train,X_test,Y_train,Y_test = train_test_split(X_vect,Y,test_size=0.2,random_state=2)
```

In [231…
```
# Modeling the dataset using Multinomial Naive Bayes algorithm

classifier_nb = MultinomialNB()
classifier_nb.fit(X_train,Y_train)
```

Out[231]:
```
MultinomialNB()
```

In [232…
```
# making prediction
Y_pred_nb = classifier_nb.predict(X_test)
```

In [233…
```
# standard evaluation

def standard_evaluation(Y_test, Y_pred):
    accuracy = metrics.accuracy_score(Y_test,Y_pred) * 100
    precision = metrics.precision_score(Y_test, Y_pred, pos_label=1) * 100
    recall = metrics.recall_score(Y_test, Y_pred, pos_label=1) * 100
    f1_score = metrics.f1_score(Y_test, Y_pred, pos_label=1) * 100

    return accuracy, precision, recall, f1_score
```

In [234…
```
accuracy_nb, precision_nb, recall_nb, f1_score_nb = standard_evaluation(Y_test, Y_pred_nb)
accuracy_nb, precision_nb, recall_nb, f1_score_nb
```

Out[234]:
```
(94.80868203177445, 95.50890895777398, 93.32220367278798, 94.40289505428228)
```

2. Logistic Regression

Logistic regression is a data analysis technique that finds the relationships between two data factors. It then uses this relationship to predict the value of one of those outcomes. The prediction usually has a finite number of outcomes, like yes or no.

Source - https://aws.amazon.com/what-is/logistic-regression/

In [235…
```
from sklearn.linear_model import LogisticRegression
classifier_lr = LogisticRegression()
classifier_lr.fit(X_train,Y_train)
Y_pred_lr = classifier_lr.predict(X_test)

accuracy_lr, precision_lr, recall_lr, f1_score_lr = standard_evaluation(Y_test, Y_pred_lr)
```

```
accuracy_lr, precision_lr, recall_lr, f1_score_lr
```

Out[235]:
```
(99.0937569926158, 99.11610129001434, 98.95063200572383, 99.03329752953813)
```

3. KNN

The k-nearest neighbors algorithm, also known as KNN is a supervised learning classifier, which uses proximity to make predictions about the grouping of an individual data point. It is a non-parametric algorithm, meaning that it makes no assumptions about the underlying distribution of the data.

Source - https://www.ibm.com/in-en/topics/knn

In [236…
```
from sklearn.neighbors import KNeighborsClassifier
# check for 5 nearest neighbor and decide using minkowski metric and eucledian distance
classifier_knn = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2)
classifier_knn.fit(X_train,Y_train)
Y_pred_knn = classifier_knn.predict(X_test)

accuracy_knn, precision_knn, recall_knn, f1_score_knn = standard_evaluation(Y_test, Y_pred_knn)

accuracy_knn, precision_knn, recall_knn, f1_score_knn
```

```
/Users/shivammaheshwari/opt/anaconda3/lib/python3.9/site-packages/sklearn/neighbors/_classification.py:228: FutureWarn
ing: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves th
e axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdims` will become False, t
he `axis` over which the statistic is taken will be eliminated, and the value None will no longer be accepted. Set `ke
epdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

Out[236]:
```
(87.18952785858134, 81.97650020981956, 93.17910803720486, 87.21955575399039)
```

4. Random Forest Classifier

Random Forest is based on the concept of ensemble learning. Random Forest Classifier is a type of machine learning algorithm that is used for classification tasks. The algorithm is to build multiple decision trees, where each tree is trained on a random subset of the training data and the features.

In [237…
```
from sklearn.ensemble import RandomForestClassifier

# run decision tree in random forest 20 times
classifier_rf = RandomForestClassifier(n_estimators=20, random_state=2, criterion="entropy")

classifier_rf.fit(X_train,Y_train)
Y_pred_rf = classifier_rf.predict(X_test)
accuracy_rf, precision_rf, recall_rf, f1_score_rf = standard_evaluation(Y_test, Y_pred_rf)

accuracy_rf, precision_rf, recall_rf, f1_score_rf
```

Out[237]:
```
(99.23920340120831, 99.49604031677465, 98.87908418793226, 99.1866028708134)
```

5. SVC

SVC stands for **Support Vector Classifier**, a machine learning algorithm used for classification tasks. It uses support vector machines (SVM) to separate data into separate classes. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that it can be easily put into a new data point in the correct category in the future.

Source - https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

5.1 SVC (Sigmoid)

In [238…
```
from sklearn.svm import SVC

classifier_sig = SVC(kernel='sigmoid', gamma=1.0)

classifier_sig.fit(X_train,Y_train)
Y_pred_svc = classifier_sig.predict(X_test)
accuracy_svc, precision_svc, recall_svc, f1_score_svc = standard_evaluation(Y_test, Y_pred_svc)

accuracy_svc, precision_svc, recall_svc, f1_score_svc
```

Out[238]:
```
(99.74267173864399, 99.71387696709584, 99.73765800143096, 99.72576606653153)
```

5.2 SVC (Linear)

In [239…
```
from sklearn.svm import SVC

classifier_lin = SVC(kernel='linear', gamma=1.0)

classifier_lin.fit(X_train,Y_train)
Y_pred_lin = classifier_lin.predict(X_test)
accuracy_lin, precision_lin, recall_lin, f1_score_lin = standard_evaluation(Y_test, Y_pred_lin)

accuracy_lin, precision_lin, recall_lin, f1_score_lin
```

Out[239]: (99.78742447974939, 99.78530534351145, 99.76150727402813, 99.77340488968396)

5.3 SVC (RBF)

In [240…
```
from sklearn.svm import SVC

classifier_rbf = SVC(kernel='rbf', gamma=1.0)

classifier_rbf.fit(X_train,Y_train)
Y_pred_rbf = classifier_rbf.predict(X_test)
accuracy_rbf, precision_rbf, recall_rbf, f1_score_rbf = standard_evaluation(Y_test, Y_pred_rbf)

accuracy_rbf, precision_rbf, recall_rbf, f1_score_rbf
```

Out[240]: (99.73148355336764, 99.73753280839895, 99.6899594562366, 99.71374045801527)

6. LSTM

A long short-term memory is a type of recurrent neural network. LSTMs are used to learn, process, and classify sequential data. The common uses involves sentiment analysis, language modeling, speech recognition etc.

In [241…
```
from tensorflow.keras.layers import Embedding
from tensorflow.keras.preprocessing.text import one_hot

from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dropout
```

In [242…]
```
# convert text in a one hot representation with fixed vocabulary size
vocab_size=10000
onehot_repr=[one_hot(words,vocab_size)for words in df['text']]
```

We need to make sure that all inputs in our model are of same shape and size. Some sentence may be shorter, other may be longer. We use padding to make sure that sentence are of same size.

In [243…]
```
# limit sentence length
sent_length=5000

# padding the sentences
embedded_docs=pad_sequences(onehot_repr,padding='pre',maxlen=sent_length)
embedded_docs
```

Out[243]:
```
array([[   0,    0,    0, ..., 2766, 6712, 3489],
       [   0,    0,    0, ..., 5867, 9778, 1628],
       [   0,    0,    0, ..., 5828, 4022, 1628],
       ...,
       [   0,    0,    0, ...,  210, 7359, 1188],
       [   0,    0,    0, ..., 7189, 3388, 1466],
       [   0,    0,    0, ..., 3150, 6563, 1628]], dtype=int32)
```

We can see all the sentences are of equal length with the length of 5000.

In [244…]
```
embedded_docs[0]
```

Out[244]:
```
array([   0,    0,    0, ..., 2766, 6712, 3489], dtype=int32)
```

In [245…]
```
# creating the lstm model

embedding_vector_features=40

model=Sequential()
model.add(Embedding(voc_size,embedding_vector_features,input_length=sent_length))
model.add(Dropout(0.3))
model.add(LSTM(100))
model.add(Dropout(0.3))
model.add(Dense(1,activation='sigmoid'))

model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 embedding_2 (Embedding)      (None, 5000, 40)          400000

 dropout_4 (Dropout)          (None, 5000, 40)          0

 lstm_2 (LSTM)                (None, 100)               56400

 dropout_5 (Dropout)          (None, 100)               0

 dense_2 (Dense)              (None, 1)                 101

=================================================================
Total params: 456,501
Trainable params: 456,501
Non-trainable params: 0
_________________________________________________________________
```

In [246…
```
X_final=np.array(embedded_docs)
Y_final=np.array(Y)
```

In [265…
```
X_train_lstm, X_test_lstm, y_train_lstm, y_test_lstm = train_test_split(X_final, Y_final, test_size=0.2,
                                                                        random_state=2)

model.fit(X_train_lstm,y_train_lstm,validation_data=(X_test_lstm,y_test_lstm),epochs=1,batch_size=64)
```

```
559/559 [==============================] - 4561s 8s/step - loss: 0.0067 - accuracy: 0.9987 - val_loss: 0.0015 - val_ac
curacy: 0.9999
```

Out[265]:
```
<keras.callbacks.History at 0x7fb44d2828e0>
```

In [266…
```
Y_pred_lstm = model.predict(X_test_lstm)
Y_pred_lstm = np.round(Y_pred_lstm).astype(int)
```

```
280/280 [==============================] - 369s 1s/step
```

In [267…
```
accuracy_lstm, precision_lstm, recall_lstm, f1_score_lstm = standard_evaluation(y_test_lstm, Y_pred_lstm)
accuracy_lstm, precision_lstm, recall_lstm, f1_score_lstm
```

Out[267]: (99.98881181472366, 99.97615641392466, 100.0, 99.98807678550136)

C. Sentiment Analysis

Sentiment Analysis is the process of analysing the content of the dataset to determine whether it is positive, negative or neutral.

In [268…
```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.sentiment.vader import SentimentIntensityAnalyzer as sid
```

In [269…
```
sent_analysis = sid()

df['scores'] = df['text'].apply(lambda text: sent_analysis.polarity_scores(text))
```

In [270…
```
df.head()
```

Out[270]:

	text	date	status	scores
19216	boom company openly criticized trump making am...	jan 31, 2017	0	{'neg': 0.088, 'neu': 0.741, 'pos': 0.171, 'co...
4532	democratic senator oppose trump s u s supreme ...	march 31, 2017	1	{'neg': 0.08, 'neu': 0.757, 'pos': 0.163, 'com...
7511	hate speech seeps u s mainstream amid bitter c...	november 7, 2016	1	{'neg': 0.193, 'neu': 0.726, 'pos': 0.081, 'co...
7826	canadian court rule trump face claim toronto t...	october 13, 2016	1	{'neg': 0.102, 'neu': 0.801, 'pos': 0.097, 'co...
6478	watch snl s church lady return hilariously moc...	may 8, 2016	0	{'neg': 0.125, 'neu': 0.635, 'pos': 0.24, 'com...

In [271…
```
df['compound'] = df['scores'].apply(lambda scores_dict: scores_dict['compound'])

df.head()
```

Out[271]:

	text	date	status	scores	compound
19216	boom company openly criticized trump making am...	jan 31, 2017	0	{'neg': 0.088, 'neu': 0.741, 'pos': 0.171, 'co...	0.9971
4532	democratic senator oppose trump s u s supreme ...	march 31, 2017	1	{'neg': 0.08, 'neu': 0.757, 'pos': 0.163, 'com...	0.9870
7511	hate speech seeps u s mainstream amid bitter c...	november 7, 2016	1	{'neg': 0.193, 'neu': 0.726, 'pos': 0.081, 'co...	-0.9972
7826	canadian court rule trump face claim toronto t...	october 13, 2016	1	{'neg': 0.102, 'neu': 0.801, 'pos': 0.097, 'co...	-0.3612
6478	watch snl s church lady return hilariously moc...	may 8, 2016	0	{'neg': 0.125, 'neu': 0.635, 'pos': 0.24, 'com...	0.9888

In [272…

```
df['sentiment'] = df['compound'].apply(lambda c: 'positive' if c>0.25 else ('negative' if c<-0.25 else 'neutral'))
df.head()
```

Out[272]:

	text	date	status	scores	compound	sentiment
19216	boom company openly criticized trump making am...	jan 31, 2017	0	{'neg': 0.088, 'neu': 0.741, 'pos': 0.171, 'co...	0.9971	positive
4532	democratic senator oppose trump s u s supreme ...	march 31, 2017	1	{'neg': 0.08, 'neu': 0.757, 'pos': 0.163, 'com...	0.9870	positive
7511	hate speech seeps u s mainstream amid bitter c...	november 7, 2016	1	{'neg': 0.193, 'neu': 0.726, 'pos': 0.081, 'co...	-0.9972	negative
7826	canadian court rule trump face claim toronto t...	october 13, 2016	1	{'neg': 0.102, 'neu': 0.801, 'pos': 0.097, 'co...	-0.3612	negative
6478	watch snl s church lady return hilariously moc...	may 8, 2016	0	{'neg': 0.125, 'neu': 0.635, 'pos': 0.24, 'com...	0.9888	positive

In [273…

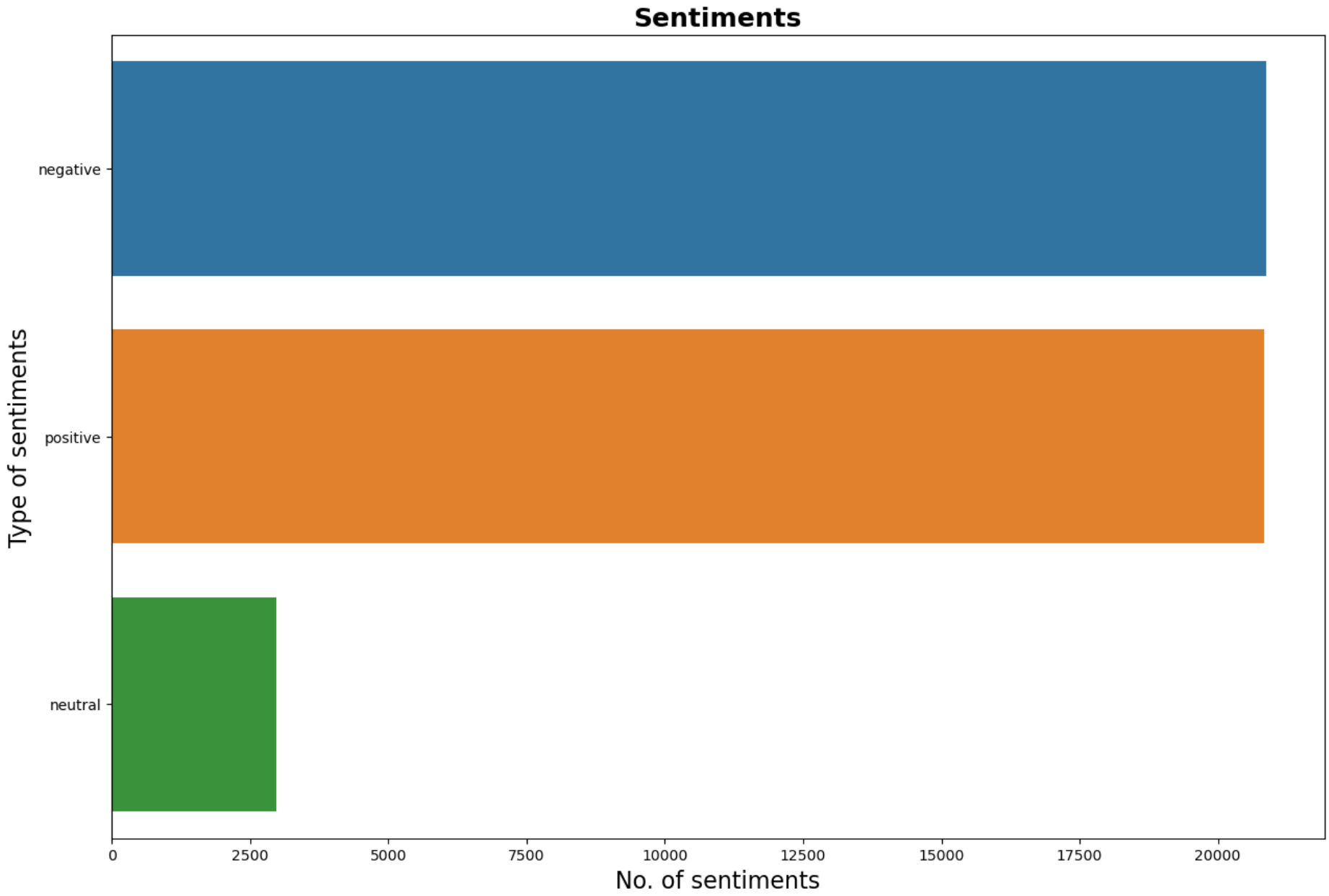
```
df['sentiment'].value_counts()
```

Out[273]:

```
negative    20880
positive    20831
neutral      2978
Name: sentiment, dtype: int64
```

In [274…

```
plt.figure(figsize= (15,10))
sns.countplot(data = df, y = 'sentiment', order=df['sentiment'].value_counts().index)
plt.title('Sentiments', fontdict={'fontweight': 'bold', 'fontsize': 18})
plt.xlabel('No. of sentiments', fontdict={'fontsize': 16})
plt.ylabel('Type of sentiments', fontdict={'fontsize': 16})
plt.show()
```



V. Evaluation

So now as we have build the models, now it's time to do an evaluation. We have already calculated the standard metrices like Accuracy, Precision, Recall, F1 score in the implementation section, here we will analyse & compare them with different algorithms and then build confusion metrics to find out which algorithm works best for our use case.

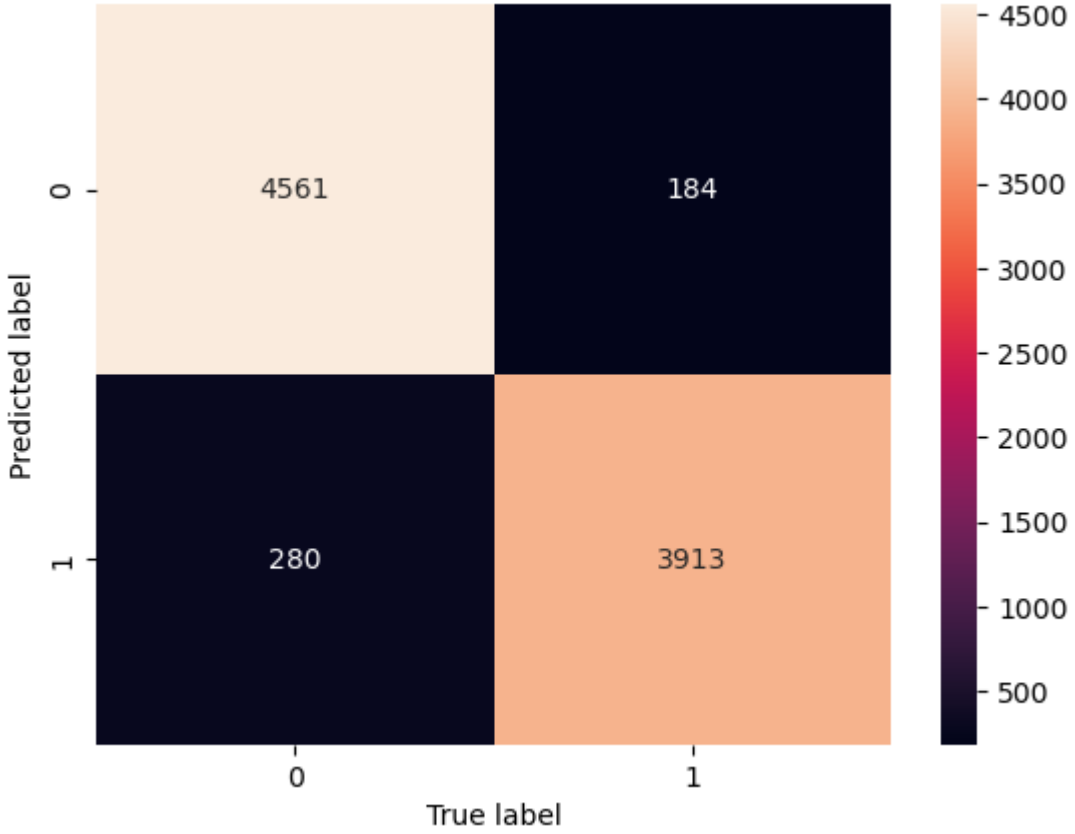
Naive Bayes Implmentation Evaluation

In [248…
```
print("Accuracy: ", accuracy_nb)
print("Precision: ", precision_nb)
print("Recall:", recall_nb)
print("F1 Score:", f1_score_nb)
```

```
Accuracy:  94.80868203177445
Precision:  95.50890895777398
Recall: 93.32220367278798
F1 Score: 94.40289505428228
```

In [249…
```
conf_matrix = metrics.confusion_matrix(Y_test, Y_pred_nb)
sns.heatmap(conf_matrix, annot = conf_matrix, fmt="d")
plt.xlabel("True label")
plt.ylabel("Predicted label")
```

Out[249]:
```
Text(50.722222222222214, 0.5, 'Predicted label')
```



The scores are above 90% which implies the model is predicting correctly. The algorithm is also quite fast in executing and predicting.

Confusion Matrix shows that 280 fake news were incorrectly predicted as Real news whereas 184 real news were incorrectly predicted as fake news.

Logistic Regression Implementation Evaluation

In [250…

```
print("Accuracy: ", accuracy_lr)
print("Precision: ", precision_lr)
print("Recall:", recall_lr)
print("F1 Score:", f1_score_lr)
```

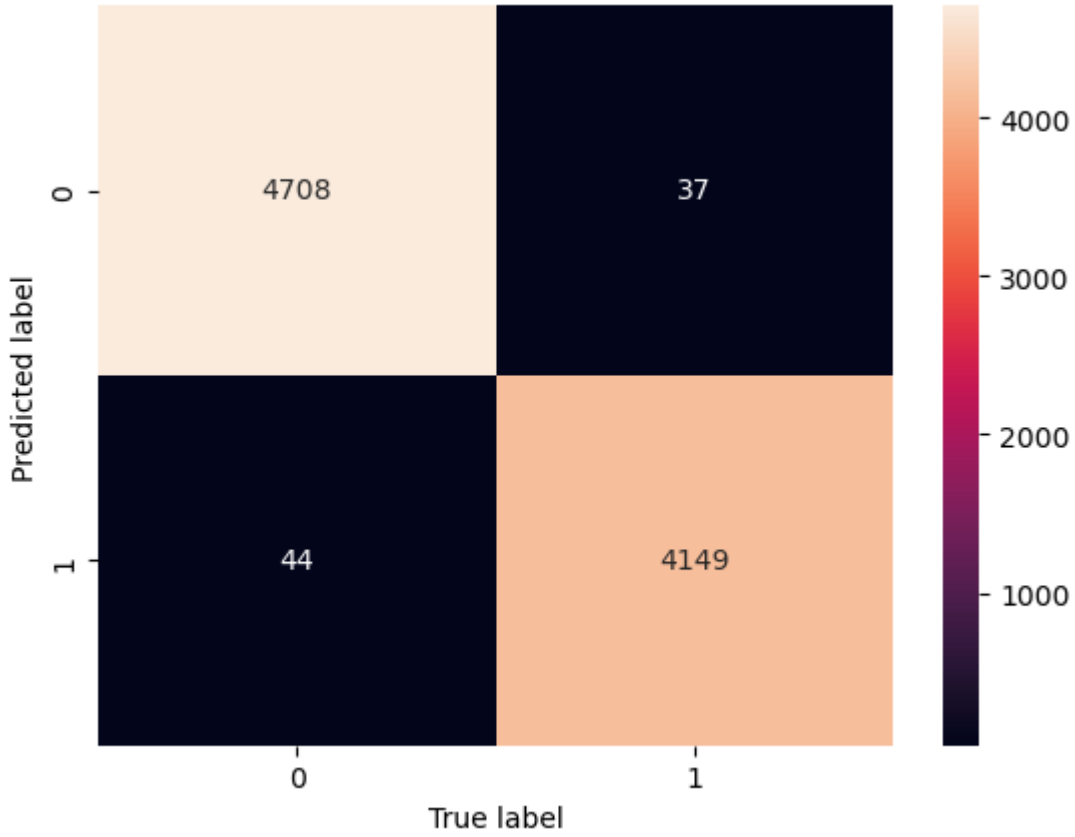
```
Accuracy:  99.0937569926158
Precision:  99.11610129001434
Recall: 98.95063200572383
F1 Score: 99.03329752953813
```

The scores generated using **Logistic Regression** is also above 90%. It seems **Logistic Regression** performs considerably better in terms of every metrices 99% vs 93-45% compared to **Naive Bayes**. Recall and F1 score are similar for both the algorithms. The computation speed is slight slower than Naive bayes.

Lets check and analyze the incorrect predictions made by our classifier.

In [251…
```
conf_matrix = metrics.confusion_matrix(Y_test, Y_pred_lr)
sns.heatmap(conf_matrix, annot = conf_matrix, fmt="d")
plt.xlabel("True label")
plt.ylabel("Predicted label")
```

Out[251]:
```
Text(50.722222222222214, 0.5, 'Predicted label')
```



Confusion Matrix shows that 44 fake news were incorrectly predicted as Real news whereas 37 correct news were incorrectly predicted as fake news.

KNN Implementation Evaluation

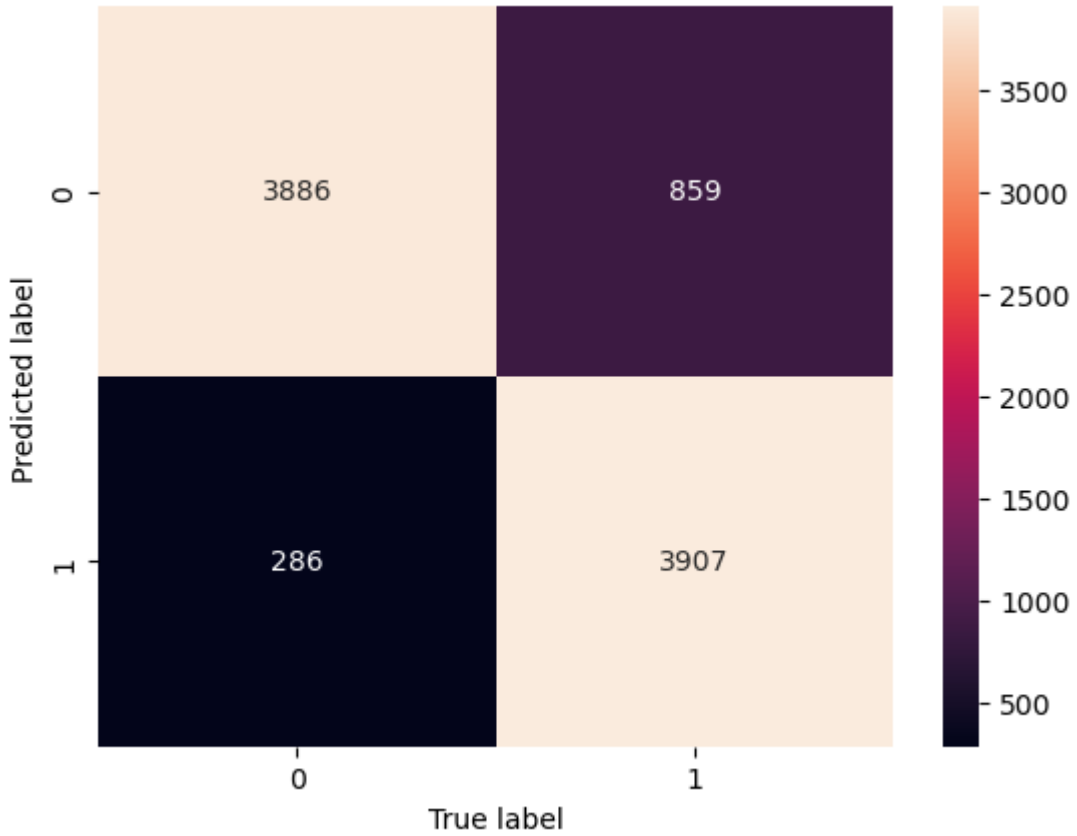
In [252…

```
print("Accuracy: ", accuracy_knn)
print("Precision: ", precision_knn)
print("Recall:", recall_knn)
print("F1 Score:", f1_score_knn)
```

```
Accuracy:  87.18952785858134
Precision:  81.97650020981956
Recall: 93.17910803720486
F1 Score: 87.21955575399039
```

In [253…
```
conf_matrix = metrics.confusion_matrix(Y_test, Y_pred_knn)
sns.heatmap(conf_matrix, annot = conf_matrix, fmt="d")
plt.xlabel("True label")
plt.ylabel("Predicted label")
```

Out[253]:
```
Text(50.722222222222214, 0.5, 'Predicted label')
```



Confusion Matrix shows that 286 fake news were incorrectly predicted as Real news whereas 859 correct news were incorrectly predicted as fake news.

I found that KNN performs substantially worse than both Naive Bayes and logistic regression and the speed is also slightly slower than logistic regression. So it does not make sense to use KNN for distinguishing between fake and real news.

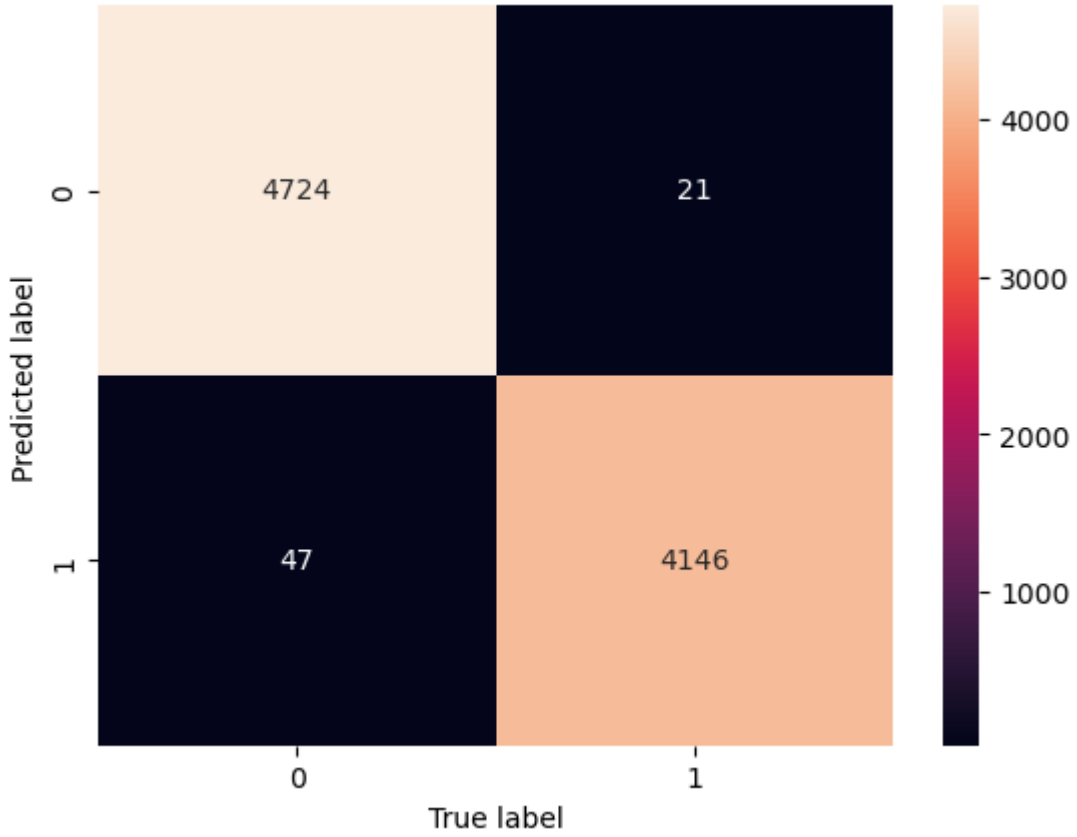
Random Forest Implementation Evaluation

In [254…
```
print("Accuracy: ", accuracy_rf)
print("Precision: ", precision_rf)
print("Recall:", recall_rf)
print("F1 Score:", f1_score_rf)
```

```
Accuracy:  99.23920340120831
Precision:  99.49604031677465
Recall: 98.87908418793226
F1 Score: 99.1866028708134
```

In [255…
```
conf_matrix = metrics.confusion_matrix(Y_test, Y_pred_rf)
sns.heatmap(conf_matrix, annot = conf_matrix, fmt="d")
plt.xlabel("True label")
plt.ylabel("Predicted label")
```

Out[255]:
```
Text(50.722222222222214, 0.5, 'Predicted label')
```



Confusion Matrix shows that 47 fake news were incorrectly predicted as Real news whereas 21 correct news were incorrectly predicted as fake news.

I found that Random Forest performance substantially better than both Naive Bayes & KNN and slightly better than logistic regression and the speed is also slightly faster than logistic regression. This makes it better then all algorithms tested. Random forest has a balance between performance and quality of classification.

SVC Implementation Evaluation

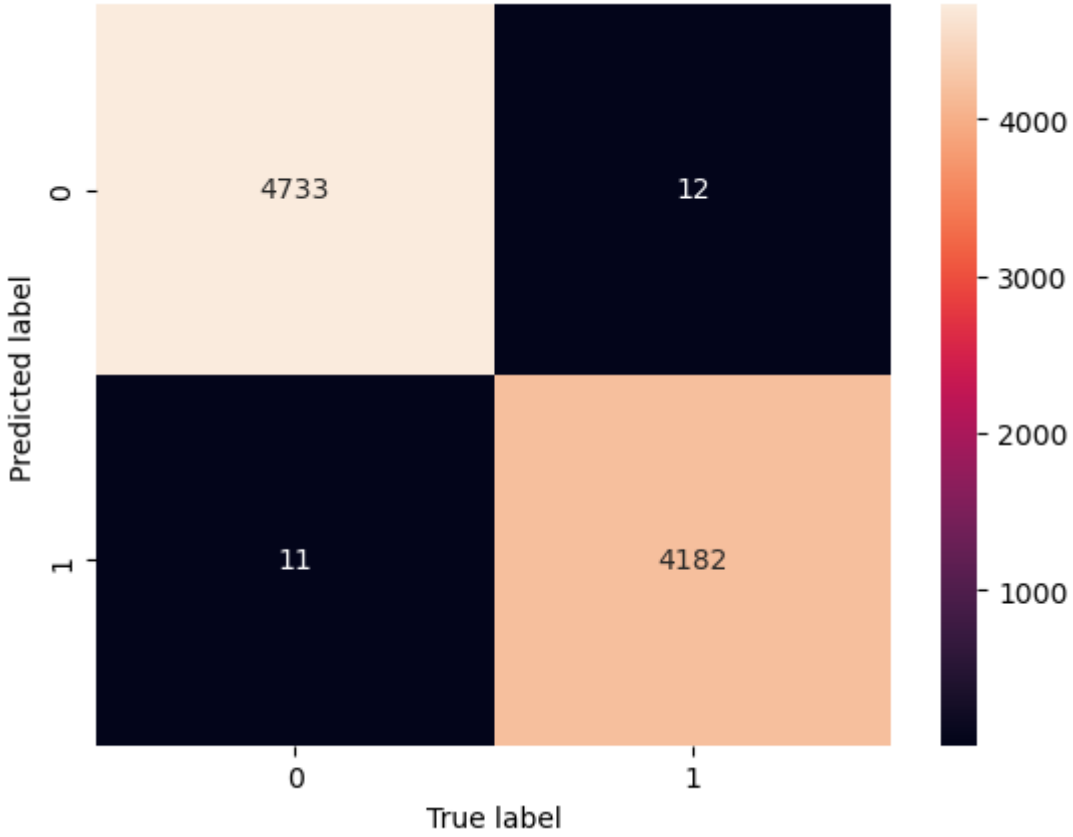
SVC (sigmoid)

In [256…
```
print("Accuracy: ", accuracy_svc)
print("Precision: ", precision_svc)
print("Recall:", recall_svc)
print("F1 Score:", f1_score_svc)
```

```
Accuracy:  99.74267173864399
Precision:  99.71387696709584
Recall: 99.73765800143096
F1 Score: 99.72576606653153
```

In [257…
```
conf_matrix = metrics.confusion_matrix(Y_test, Y_pred_svc)
sns.heatmap(conf_matrix, annot = conf_matrix, fmt="d")
plt.xlabel("True label")
plt.ylabel("Predicted label")
```

Out[257]: Text(50.722222222222214, 0.5, 'Predicted label')



Confusion Matrix shows that 11 fake news were incorrectly predicted as Real news whereas 12 correct news were incorrectly predicted as fake news.

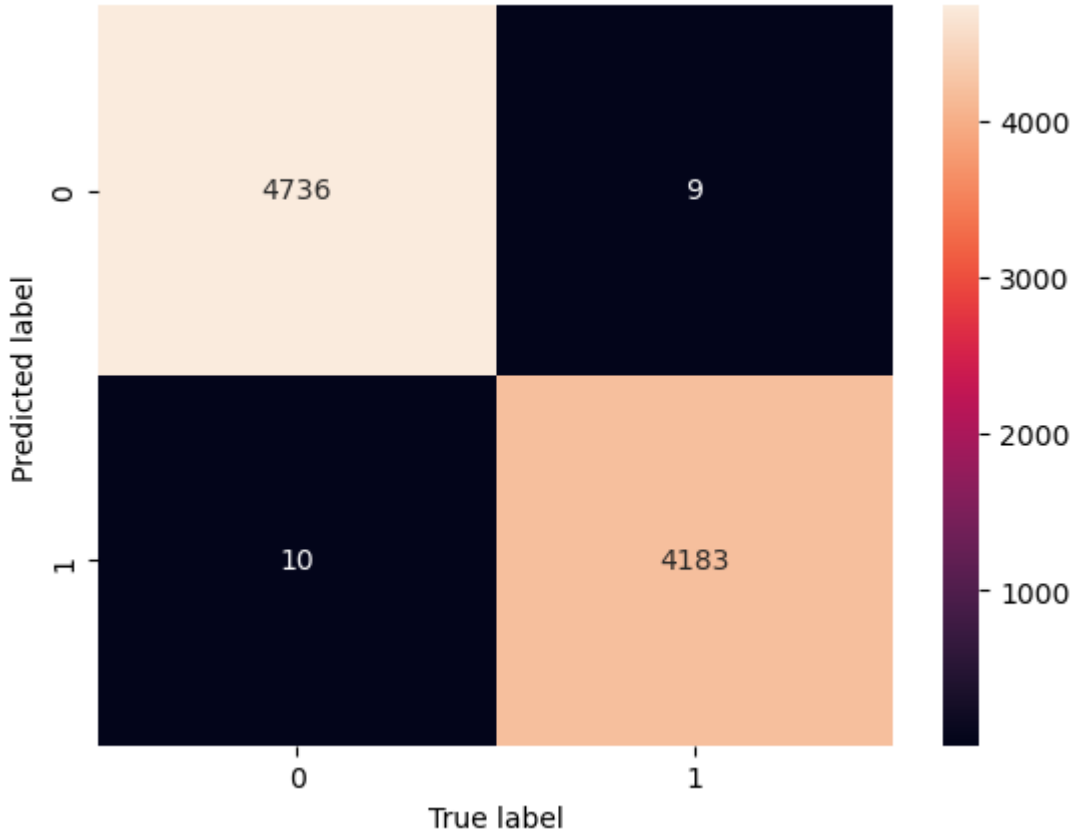
SVC (linear)

In [258…
```
print("Accuracy: ", accuracy_lin)
print("Precision: ", precision_lin)
print("Recall:", recall_lin)
print("F1 Score:", f1_score_lin)
```

```
Accuracy:  99.78742447974939
Precision:  99.78530534351145
Recall: 99.76150727402813
F1 Score: 99.77340488968396
```

In [259…
```
conf_matrix = metrics.confusion_matrix(Y_test, Y_pred_lin)
sns.heatmap(conf_matrix, annot = conf_matrix, fmt="d")
plt.xlabel("True label")
plt.ylabel("Predicted label")
```

Out[259]:
```
Text(50.722222222222214, 0.5, 'Predicted label')
```



Confusion Matrix shows that 10 fake news were incorrectly predicted as Real news whereas 9 correct news were incorrectly predicted as fake news.

SVC (RBF)

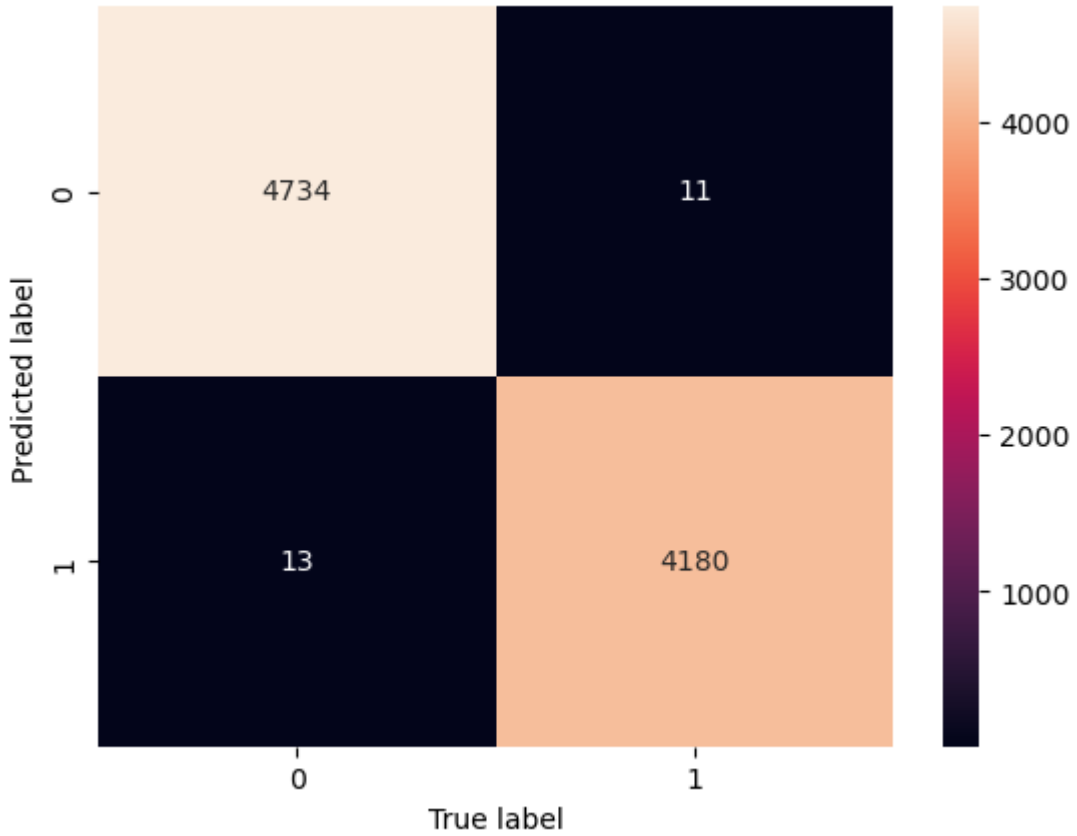
In [260…

```
print("Accuracy: ", accuracy_rbf)
print("Precision: ", precision_rbf)
print("Recall:", recall_rbf)
print("F1 Score:", f1_score_rbf)
```

```
Accuracy:  99.73148355336764
Precision:  99.73753280839895
Recall: 99.6899594562366
F1 Score: 99.71374045801527
```

In [261…
```
conf_matrix = metrics.confusion_matrix(Y_test, Y_pred_rbf)
sns.heatmap(conf_matrix, annot = conf_matrix, fmt="d")
plt.xlabel("True label")
plt.ylabel("Predicted label")
```

Out[261]:
```
Text(50.722222222222214, 0.5, 'Predicted label')
```



Confusion Matrix shows that 13 fake news were incorrectly predicted as Real news whereas 11 correct news were incorrectly predicted as fake news.

The scores generated by SVC algorithm are much better than the algorithms that we have tested out before. Although there is an issue i.e. the algorithms are considerably slow than knn, logistic regression, Naive Bayes, Random forest algorithms etc. In All 3 SVC models that I have tested, I found SVC (rbf) to be the slowest, SVC (sigmoid) a bit faster and SVC (linear) to be the fastest out of three. Even the fastest SVC (linear) is considerably slower than rest of the algorithms.

If we compare SVC (linear) to Random forest which has outperformed all the algorithms that we have tested so far, the SVC (linear) has slightly better results compare to Random forest but at the expense of performance, the speed of SVC algorithm is very slow and it doesn't make sense to use it over the Random forest algorithm.

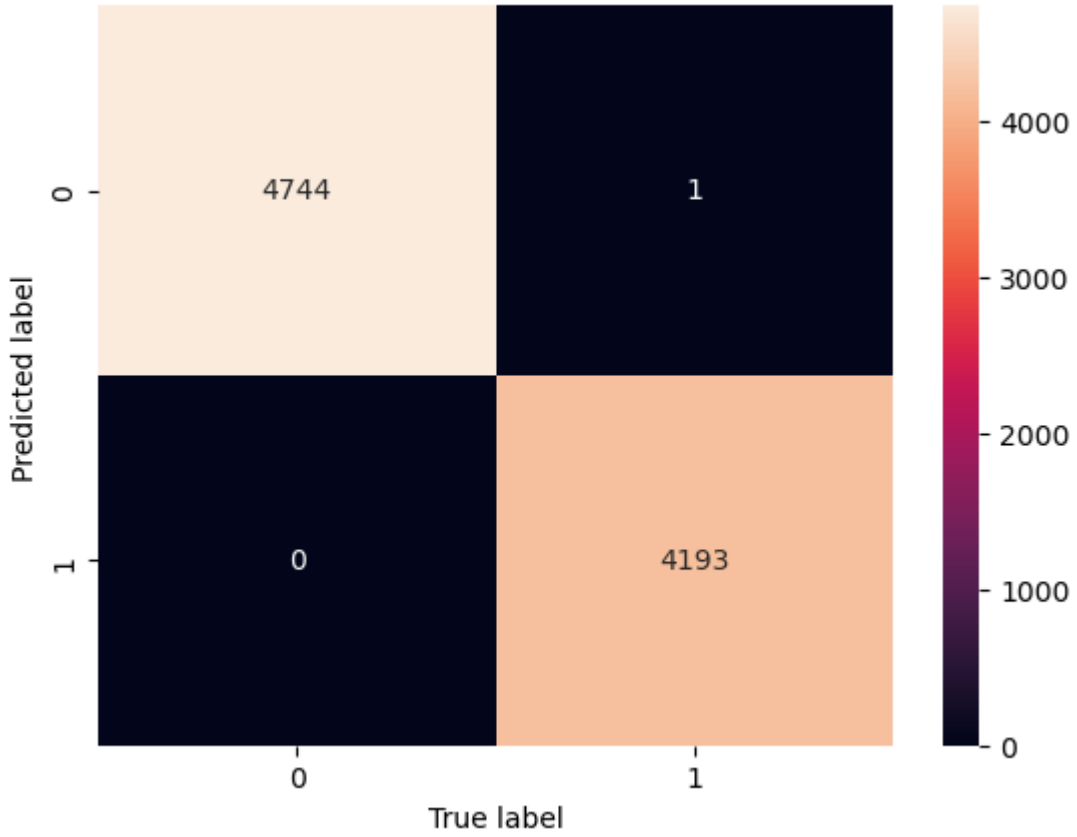
LSTM Implementation Evaluation

In [275…
```
print("Accuracy: ", accuracy_lstm)
print("Precision: ", precision_lstm)
print("Recall:", recall_lstm)
print("F1 Score:", f1_score_lstm)
```

```
Accuracy:  99.98881181472366
Precision:  99.97615641392466
Recall: 100.0
F1 Score: 99.98807678550136
```

In [277…
```
conf_matrix = metrics.confusion_matrix(Y_test, Y_pred_lstm)
sns.heatmap(conf_matrix, annot = conf_matrix, fmt="d")
plt.xlabel("True label")
plt.ylabel("Predicted label")
```

Out[277]:
```
Text(50.722222222222214, 0.5, 'Predicted label')
```



LSTM produces the best results out of all the algorithms with metrics close to 100. Although the LSTM algorithm is also very effective in detecting fake and real news but the problem here is the same that we face with SVC algorithm i.e. the algorithm is very slow compare to all the other algorithms. So, it does not make sense to us to compromise on performance.

VI. Conclusions

The project have been very useful in understanding the various functionalities and algorithm in solving Text classification problems. The techniques used in this project is not limited to just "Identifying Fake or Real news", it can be also applied to other text classfication projects like Spam filtering, sentiment analysis, topic labelling etc.

In this project we begin from importing and cleaning the datasets, analysing the word frequencies in the datset, calculating lexical diversity - to measure how many different words are used in the dataset, generating word cloud to find most used words, finding unigrams, bigrams and trigrams. Then we build our prediction model using various different algorithms such as Naive Bayes, Random forest, LSTM etc. to analyse how well they perform on a given dataset. We used metrics like Accuracy, Precision, Recall and F1 score to evaulate our findings along with the Confusion matrix to analysis how well we did and where we need to improve on.

We also found that random forest classifier produces the best results by giving the balance between the performance and quality of the prediction for our dataset.

Although we found SVC and LSTM perform slightly better in terms of quality of classification but the performance of both of these algorithm were significantly worse than random forest classifier. We also analysed the sentiment of fake and real news in the dataset to analyse people's sentiments on the information that is provided to them. The other things that I learned while doing this project is that we need to have balanced dataset it that is free from any bias to build a better classification model. We can also extend this classifier to make it more robust by training it on a large data sets.