V.R.S College of Engineering and Technology

Arasur , Villupuram , Tamil Nadu 607 107

Presented by :
Team head :
Madhumitha A
Team Members:
Kanchana S
Kayalvizhi G
Ramana N
Saranya S

FAKE NEWS DETECTION USING NLP

Phase-5

Introduction:

- ✓ Fake news detection is the process of identifying and verifying the accuracy of news or information that is intentionally false, misleading, or fabricated. It has become a critical concern in today's digital age, where misinformation can spread rapidly through various media channels. Here's an introduction to the topic:

- ✓ **Definition of Fake News:** Fake news encompasses various types of misinformation, including fabricated stories, manipulated images or videos, and misleading headlines. It

can be spread through websites, social media, or traditional media outlets.

- ✓ **Motivations for Fake News:** Fake news can be created for various reasons, such as political manipulation, financial gain, or simply for entertainment. It often seeks to exploit emotions, biases, or controversy to gain attention and traction.

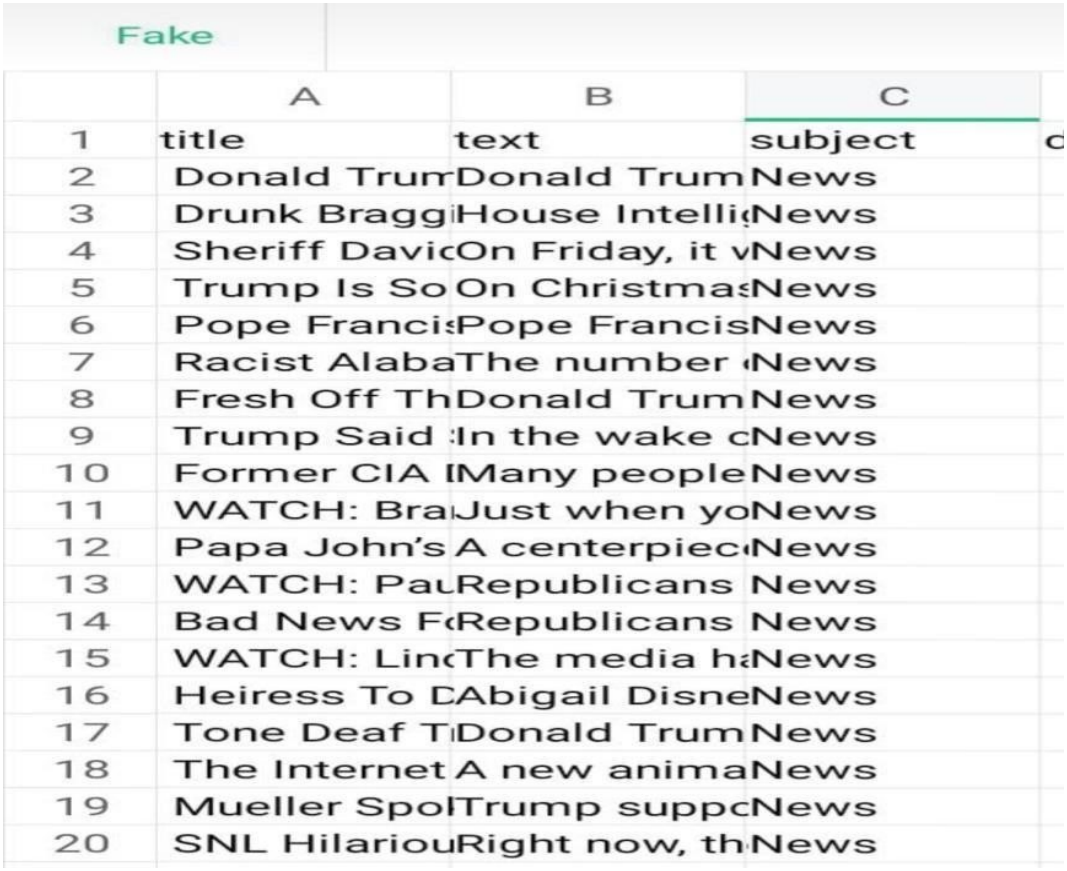
- ✓ **Impact of Fake News:** Fake news can have serious consequences, including influencing public opinion, swaying elections, causing panic, or harming individuals' reputations. It can erode trust in journalism and democratic processes.

- ✓ **Challenges in Fake News Detection:** Detecting fake news is a complex task due to its constantly evolving nature. Some challenges include the speed at which fake news spreads, the use of sophisticated techniques to make it appear legitimate, and the fine line between satire and actual misinformation.

Data Source:

A good data source for Fake news detection using NLP should be Accurate, Complete, Covering the geographic area of interest, Accessible.

Dataset Link:(https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-newsdataset)



Fake

	A	B	C
1	title	text	subject
2	Donald Trum	Donald Trum	News
3	Drunk Bragg	House Intelli	News
4	Sheriff Davi	On Friday, it v	News
5	Trump Is So	On Christmas	News
6	Pope Francis	Pope Francis	News
7	Racist Alaba	The number	News
8	Fresh Off Th	Donald Trum	News
9	Trump Said	In the wake c	News
10	Former CIA	Many people	News
11	WATCH: Bra	Just when yo	News
12	Papa John's	A centerpiec	News
13	WATCH: Pau	Republicans	News
14	Bad News F	Republicans	News
15	WATCH: Lin	The media h	News
16	Heiress To D	Abigail Disne	News
17	Tone Deaf T	Donald Trum	News
18	The Internet	A new anima	News
19	Mueller Spo	Trump supp	News
20	SNL Hilariou	Right now, th	News

DATA COLLECTION AND PREPARATION :

- ✓ Gather a diverse dataset of news articles or social media posts, including both real and fake examples. These articles should cover a wide range of topics and sources.

- ✓ Prepare the text data by removing stop words, punctuation, and converting text to lowercase. Tokenization and stemming or lemmatization may also be applied to standardize the text.

FEATURES EXTRACTION AND LABELLING:

- ✓ Convert the textual content into numerical features that machine learning algorithms can understand. Common techniques

include TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings like Word2Vec or GloVe.

- ✓ Annotate your dataset to indicate which articles are real and which are fake. This labeled data will be used for training and testing your model.

MODEL SELECTION AND TRAINING:

- ✓ Choose an appropriate NLP model or algorithm. Common choices include logistic regression, random forests, or more advanced methods like recurrent neural networks (RNNs) or transformer-based models like BERT.

- ✓ Use the labelled dataset to train your NLP model. The model learns to recognize patterns and features that distinguish real news from fake news.

TRAINING AND EVALUATION:

- ✓ Use the labelled dataset to train your NLP model. The model learns to recognize patterns and features that distinguish real news from fake news.

- ✓ Assess the performance of your model using metrics such as accuracy, precision, recall, and F1-score on a separate validation or test dataset. Fine-tune your model to improve its performance.

FEATURE ENGINEERING AND BIAS DETECTION:

- ✓ Experiment with different features or techniques, such as n-grams, to enhance your model's ability to detect fake news.

- ✓ Be aware of potential biases in your dataset and model. Ensure that your model doesn't unfairly label certain sources or topics as fake news.

DEPLOYMENT :

- ✓ Once satisfied with the model's performance, deploy it to analyze and classify news articles or social media content in real-time.

- ✓ Continuously monitor your model's performance and update it as needed to adapt to evolving fake news tactics.

ETHICAL CONSIDERATION:

- ✓ Be mindful of ethical considerations, such as privacy and freedom of speech, when developing and deploying fake news detection systems.

- ✓ Remember that fake detection is a challenging task, and achieving high accuracy can be difficult due to the evolving nature of fake news. It often requires ongoing research and adaptation to stay effective in identifying misinformation and disinformation online.

- ✓ It's important to balance the detection of fake news with respect for free speech and privacy. Striking this balance can be challenging and requires careful consideration.

CONTINUOUS EVOLUTION:

- ✓ Fake news detection methods must continually adapt to new tactics used by purveyors of misinformation. Ongoing research and collaboration are crucial in this ever-changing landscape.

TEXT ANALYSIS:

- ✓ NLP techniques are used to analyze the content of news articles or social media posts. This includes sentiment analysis, identifying unusual language patterns, and examining the tone of the text.
- ✓ Creating meaningful features from the text data is crucial. Features might include word frequency, readability scores, or linguistic features that can help distinguish fake from real news.

SUPERVISED LEARNING:

- ✓ Most fake news detection models are trained using supervised learning. They learn from labeled datasets that contain examples of both fake and real news to make predictions on new, unlabeled data.

ENSEMBLE METHODS:

- ✓ Combining the predictions of multiple machine learning models can enhance accuracy. Techniques like Random Forests or Gradient Boosting are commonly used.

PROGRAM:

FAKE NEWS DETECTION

IMPORT LIBRARIES:

In[1]:

```
Import numpy as np
```

```
Import pandas as pd
Import matplotlib.pyplot as plt
Import seaborn as sns

Import nltk
Import re
Import string

From sklearn.model_selection import train_test_split
From sklearn.metrics import classification_report

Import keras
From keras.preprocessing import text,sequence
From keras.models import Sequential
From keras.layers import Dense,Embedding,LSTM,Dropout

Import warnings
Warnings.filterwarnings('ignore')

Import os
For dirname, _, filenames in os.walk('/kaggle/input'):
For filename in filenames:
Print(os.path.join(dirname, filename))
```

LOAD AND CHECK DATA:

In[2]:

Real_data = pd.read_csv(‘/kaggle/input/fake-and-real-news-dataset/True.csv’)

Fake_data = pd.read_csv(‘/kaggle/input/fake-and-real-news-dataset/Fake.csv’)

	title	text	s
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	p
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	p
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	p
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	p
	Trump		

In[3]:

real_data.hea

d

	title	text	subject
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that	News

In[5]:

real_data[‘target

’] = 1

fake_data[‘targe

t’] = 0

In[6]:

real_data.tail()

Out[6]:

	title	text	subject
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews

In[7]:

Data = pd.concat([real_data, fake_data], ignore_index=True, sort=False) Data.head()

Out[7]:

	title	text	s
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	p
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	p
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	p
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	p
	Trump		

In[8]:

data.isnull().sum

()

Out[8]:

Title 0

Text 0

Subject 0

Date 0

Target 0

Dtype: int64

VISUALIZATION

Count of Fake and Real Data

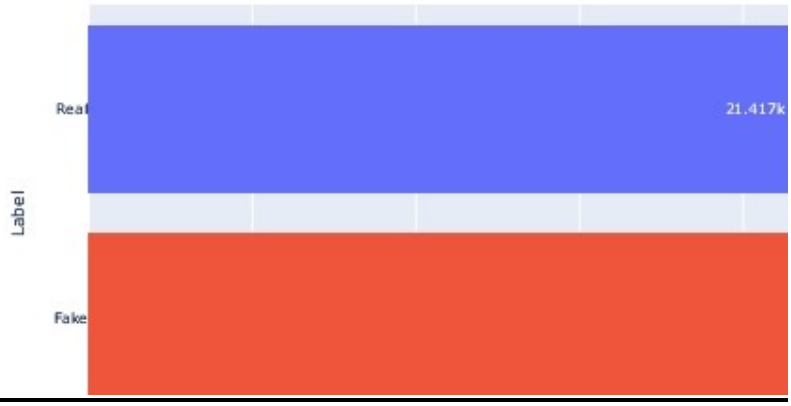
In[9]:

```
print(data[“target”].value_c
ounts()) fig, ax =
plt.subplots(1,2,
figsize=(19, 5))
G1 = sns.countplot(data.target,ax=ax[0],palette=”pastel”);
G1.set_title(“Count of real and fake data”)
G1.set_ylabel(“Count”)
G1.set_xlabel(“Target”)
G2 =
```

plt.pie(data[“target”].value_counts().values,explode=[0,0],labels=data .target.value_counts().inde x, autopct=’%1.1f%%’,colors=[‘SkyBlue’,’PeachPuff’]) fig.show()

0 1 21417

Name: target, dtype: int64



Distribution of The Subject According to Real and Fake Data

In[9]:

print(data.subject.value_counts())

plt.figure(figsize=(10, 5))

ax = sns.countplot(x=”subject”, hue=’target’, data=data, palette=”pastel”) plt.title(“Distribution of The Subject According to Real and Fake Data”)

politicsNews

11272 worldnews

10145 News

9050

Politics 6841

Left-news

4459

Government News 1570

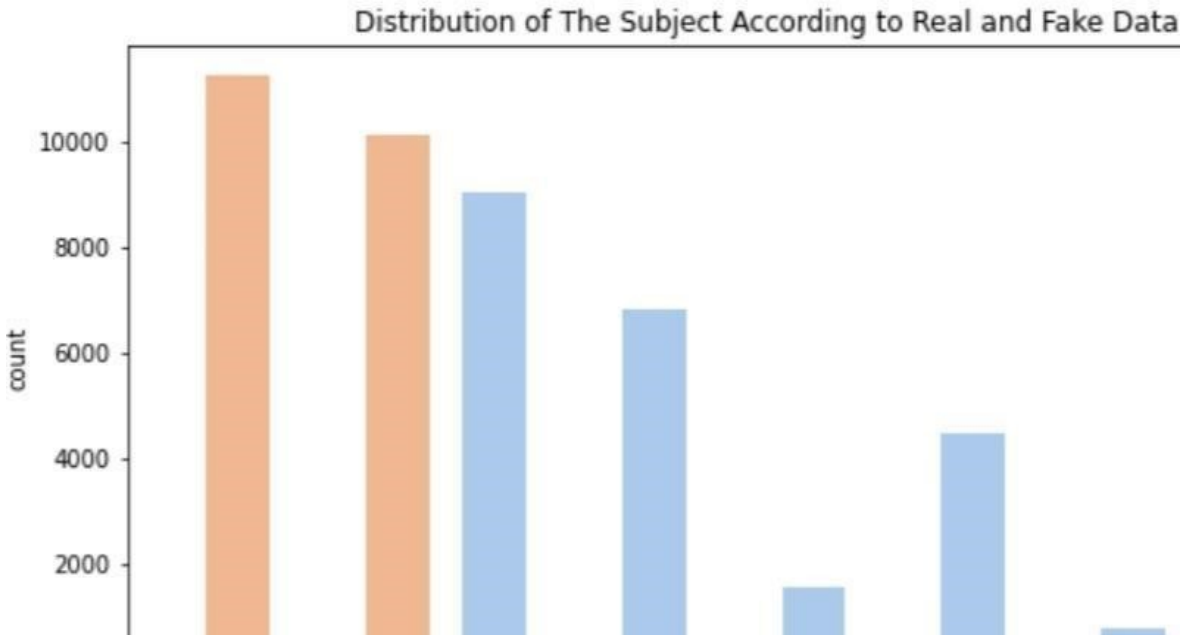
US_News 783

Middle-east 778

Name: subject, dtype: int64

Out[10]:

Text(0.5, 1.0, 'Distribution of The Subject According to Real and Fake Data')



DATA CLEANING

In[11]:

data[‘text’]= data[‘subject’] + “ “ + data[‘title’]

+ “ “ + data[‘text’] del data[‘title’] del

data[‘subject’] del data[‘date’] data.head()

Out[11]:

	text	Label
14423	RAMALLAH/WASHINGTON (Reuters) - Palestinian of...	NaN
18027	ZURICH (Reuters) - Switzerland on Sunday calle...	NaN
39100	Is there a more corrupt and power hungry group...	NaN
22351	Leaked text messages between the daughters of ...	NaN
11652	BEIJING/TAIPEI (Reuters) - A Beijing court on ...	NaN
29114	When Donald Trump isn t bragging about the siz...	NaN
14935	BERLIN (Reuters) - Environmental policy domina...	NaN
15391	HONG KONG (Reuters) - Some activists in Hong K...	NaN

Int[12]: from wordcloud import

WordCloud,STOPWORDS

plt.figure(figsize = (15,15))

Wc = WordCloud(max_words = 500 , width = 1000 , height = 500 , stopwords =
STOPWORDS).generate(“ “.join(data[data.target == 1].text))

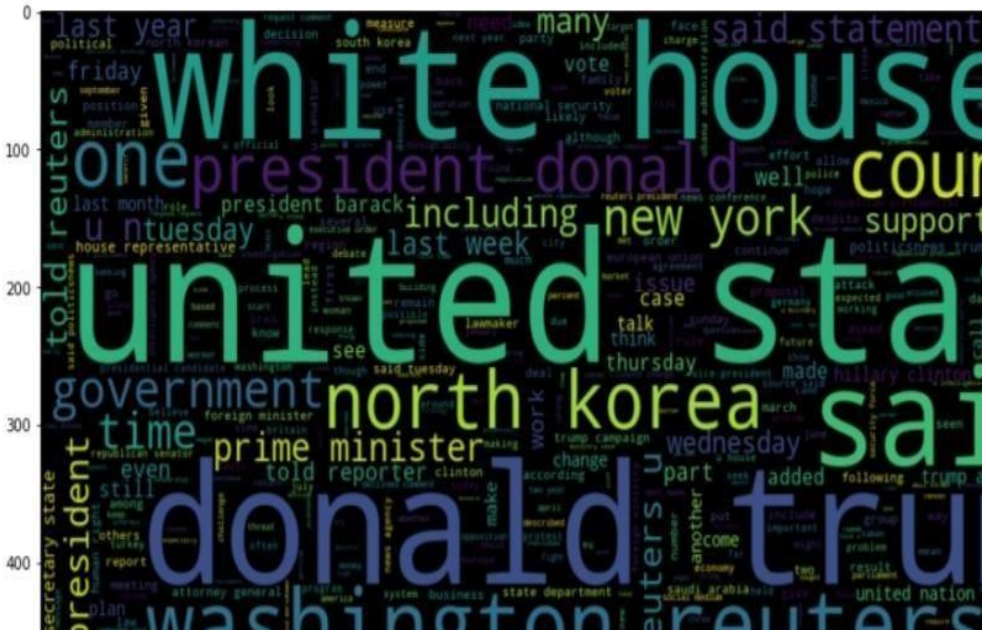
Plt.imshow(wc , interpolation = ‘bilinear’)

Out[12]:

<matplotlib.image.AxesImage at 0x7f6934fd2750>

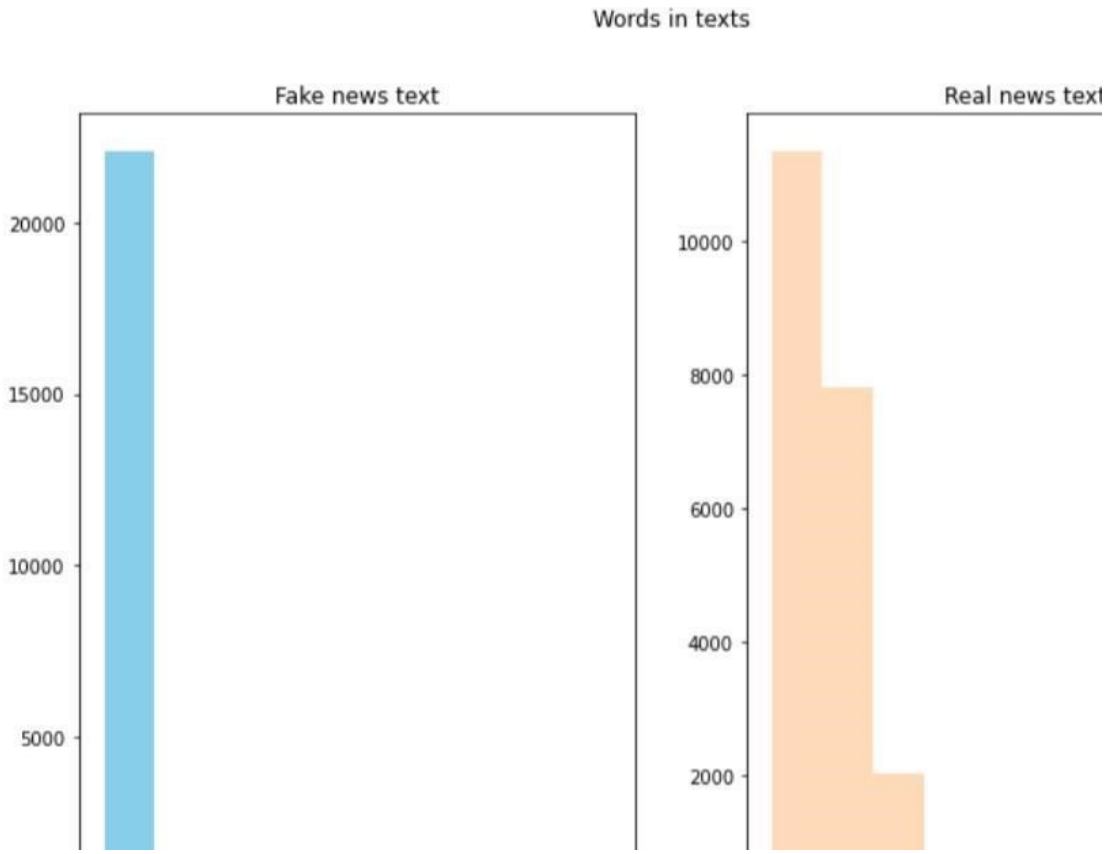
Int[13]:

Number of words in each text



fig,(ax1,ax2)=plt.subplots(1,2,figsize=(12,8))

text_len=data[data[‘target’]==0][‘text’].str.split().map(lambda x: len(x)) ax1.hist(text_len,color=’SkyBlue’) ax1.set_title(‘Fake news text’) text_len=data[data[‘target’]==1][‘text’].str.split().map(lambda x: len(x)) ax2.hist(text_len,color=’PeachPuff’) ax2.set_title(‘Real news text’) fig.suptitle(‘Words in texts’) plt.show()



The number of words seems to be a bit different. 500 words are most common in real news category while around 250 words are most common in fake news category.

N-Gram Analysis

Int[14]:

Texts = ‘ ‘.join(data[‘text’]

Int[15]:

String = texts.split(“ “)

Int[16]:

```
def draw_n_gram(string,i):
N_gram = (pd.Series(nltk.ngrams(string, i)).value_counts())[:15]
N_gram_df=pd.DataFrame(n_gram)
N_gram_df = n_gram_df.reset_index()
N_gram_df = n_gram_df.rename(columns={“index”: “word”, 0: “count”})
Print(n_gram_df.head())
Plt.figure(figsize = (16,9))
Return sns.barplot(x=’count’,y=’word’, data=n_gram_df)
```

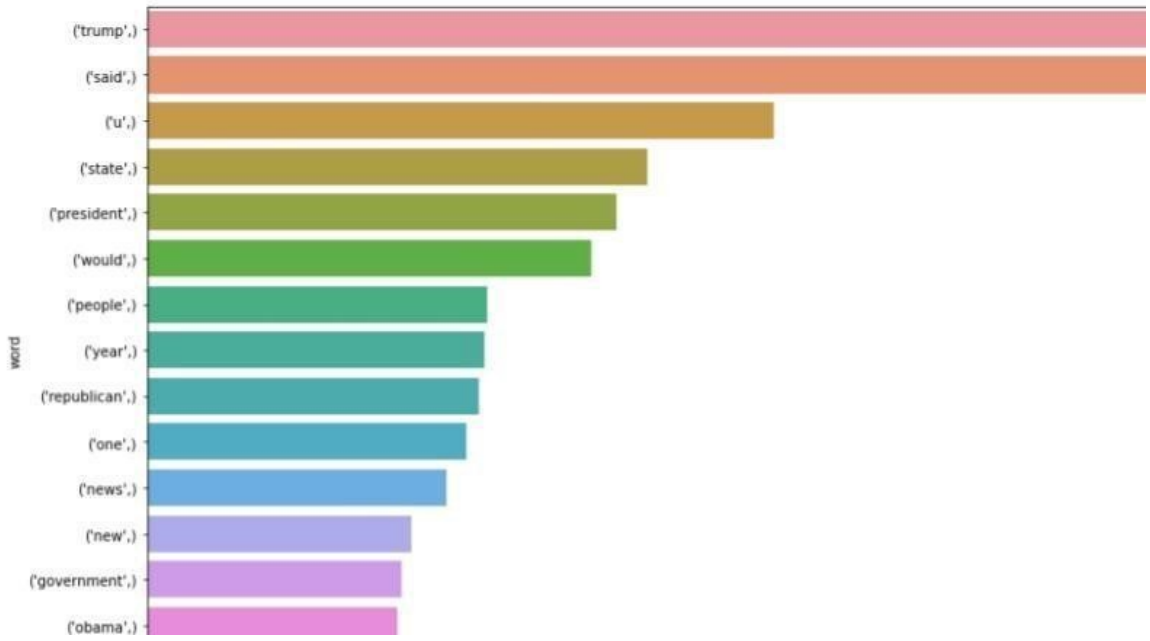
Unigram Analysis

Int[17]:

```
Draw_n_gram(string )
word   count
0      (trump,)  149603
1      (said,)  133030
2      (u,)   78516
3      (state,)   62726 4  (president,)   58790
```

Out[17]:

```
<AxesSubplot:xlabel=’count’, ylabel=’word ’>
```



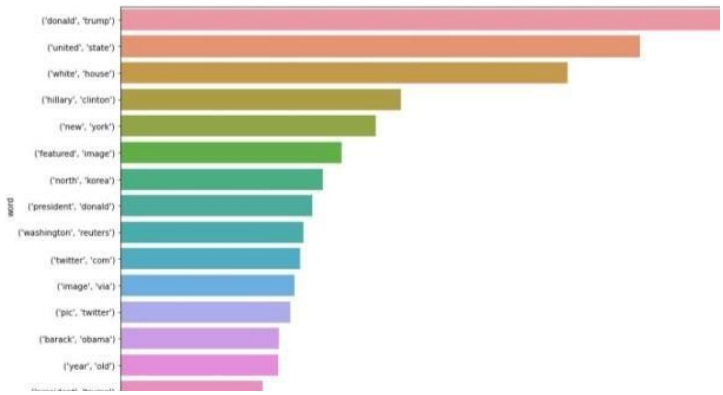
Bigram Analysis

Int[18]:

Draw_n_gram(string,2)

0	(donald, trump) 25203
1	(united, state) 18943
2	(white, house) 16296
3	(hillary, clinton) 10217
4	(new, york) 9305

Out[18]:



<AxesSubplot:xlabel=’count’, ylabel=’word’>

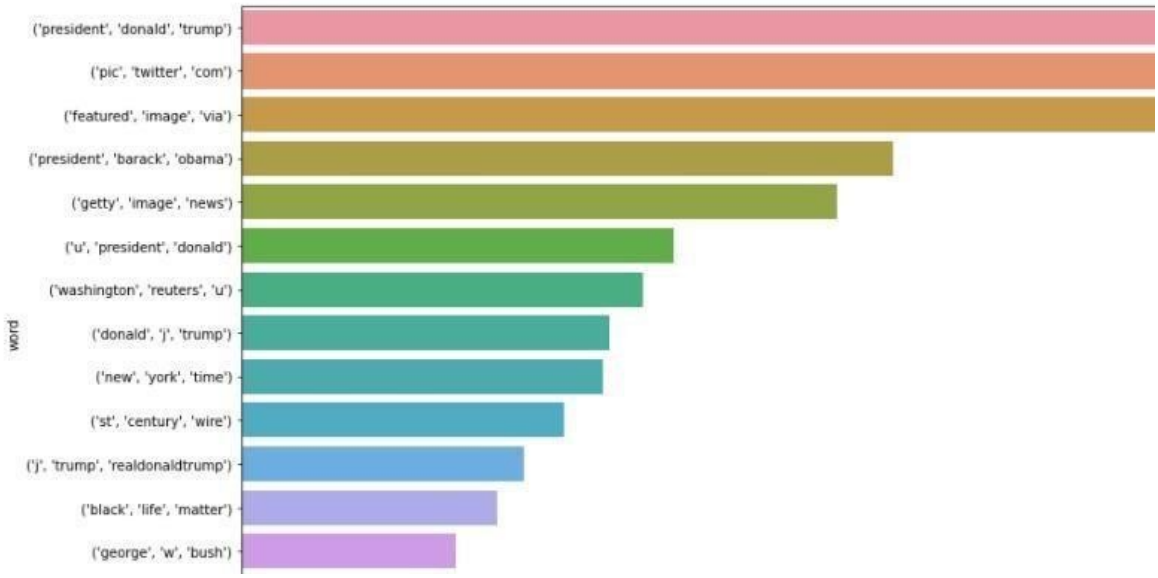
Trigram Analysis

Int[19]:

Draw_n_gram(string,3)

Out[19]

<AxesSubplot:xlabel=’count’, ylabel=’word’>



Train Test Split

Int[20]:

X_train, X_test, y_train, y_test = train_test_split(data[‘text’], data[‘target’], random_state=0)

Tokenizing

Tokenizing Text -> Repsesenting each word by a number

Mapping of orginal word to number is preserved in word_index property of tokenizer

CONCLUSION AND FUTURE WORK(Phase4):

Project Conclusion:

Fake news detection

Introduction:

- ✓ Fake news detection using Natural Language Processing (NLP) is a critical application in the field of data science and information security. NLP techniques can be employed to automatically identify and classify fake or misleading information in textual content. Here's a brief introduction to the process

- ✓ **Data collection:** The first step is to gather a diverse dataset of news articles, encompassing both genuine and fake news, preferably labeled or annotated.

- ✓ **Text processing:** cleanand preprocess the text data by removing stopwords, punctuation, and other irrelevant elements. Tokenization and stemming/lemmatization can also be applied.

- ✓ **Future Extraction:** Convert the textual data into numerical features that machine learning algorithms can work with. Common techniques include TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings like Word2Vec or GloVe..

- ✓ **Training:** Train the selected model on the labeled dataset. This involves feeding the model both the text data and their corresponding labels (fake or genuine).

- ✓ **Validation and Testing:** Assess the model's performance using validation data to fine-tune hyperparameters. Then, evaluate the model's accuracy, precision, recall, and F1-score on a test dataset.

- ✓ **Ensemble Methods:** Combining multiple models or using ensemble techniques can often improve detection accuracy.

- ✓ **Post-processing:** Apply post-processing techniques to refine the model's output, such as setting a confidence threshold for classifying news as fake.

- ✓ **Continuous Learning:** Fake news evolves, so the model should be updated regularly to adapt to new forms of disinformation.

	A	B	C	
1	title	text	subject	c
2	Donald Trum	Donald Trum	News	
3	Drunk Bragg	House Intelli	News	
4	Sheriff Davic	On Friday, it v	News	
5	Trump Is So	On Christmas	News	
6	Pope Franci	Pope Francis	News	
7	Racist Alaba	The number	News	
8	Fresh Off Th	Donald Trum	News	
9	Trump Said	In the wake c	News	
10	Former CIA I	Many people	News	
11	WATCH: Bra	Just when yo	News	
12	Papa John's	A centerpiec	News	
13	WATCH: Pau	Republicans	News	
14	Bad News F	Republicans	News	
15	WATCH: Lin	The media ha	News	
16	Heiress To D	Abigail Disne	News	
17	Tone Deaf T	Donald Trum	News	
18	The Internet	A new anima	News	
19	Mueller Spo	Trump suppc	News	
20	SNL Hilariou	Right now, th	News	
21	Republican	Senate Majo	News	

Given data set:

Input:1

```
fake = pd.read_csv('../input/fake-and-real-news-dataset/Fake.csv')
fake['flag'] = 0
fake
```

Output:1

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

Input:2

```
true = pd.read_csv('../input/fake-and-real-news-dataset/True.csv')
true['flag'] = 1
true
```

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

Output:2

Input:3

```
df = pd.DataFrame()
df = true.append(fake)
```

Input:4

```
df.info()
```

Input:5

```
df = df.drop_duplicates()
df = df.reset_index(drop=True)
```

Input:6

```
df['date'] = df['date'].replace(['19-Feb-18'],'February 19, 2018')
df['date'] = df['date'].replace(['18-Feb-18'],'February 18, 2018')
df['date'] = df['date'].replace(['17-Feb-18'],'February 17, 2018')
df['date'] = df['date'].replace(['16-Feb-18'],'February 16, 2018')
df['date'] = df['date'].replace(['15-Feb-18'],'February 15, 2018')
df['date'] = df['date'].replace(['14-Feb-18'],'February 14, 2018')
df['date'] = df['date'].replace(['13-Feb-18'],'February 13, 2018')


df['date'] = df['date'].str.replace('Dec ', 'December ')
df['date'] = df['date'].str.replace('Nov ', 'November ')
df['date'] = df['date'].str.replace('Oct ', 'October ')
df['date'] = df['date'].str.replace('Sep ', 'September ')
df['date'] = df['date'].str.replace('Aug ', 'August ')
df['date'] = df['date'].str.replace('Jul ', 'July ')
df['date'] = df['date'].str.replace('Jun ', 'June ')
df['date'] = df['date'].str.replace('Apr ', 'April ')
df['date'] = df['date'].str.replace('Mar ', 'March ')
df['date'] = df['date'].str.replace('Feb ', 'February ')
df['date'] = df['date'].str.replace('Jan ', 'January ')
```

Input:7

```
df['date'] = df['date'].str.replace(' ', '')
```

Input:8

```
for i, val in enumerate(df['date']):
df['date'].iloc[i] = pd.to_datetime(df['date'].iloc[i],
format='%B%d,%Y', errors='coerce')
```

Input:9

```
df['date'] = df['date'].astype('datetime64[ns]')
```

Input:10

```
df.info()
```

Input:11

```
import datetime as dt
df['year'] = pd.to_datetime(df['date']).dt.to_period('Y')
df['month'] = pd.to_datetime(df['date']).dt.to_period('M')

df['month'] = df['month'].astype(str)
```

Input:12

```
sub = df[['month', 'flag']]
sub = sub.dropna()
sub = sub.groupby(['month'])['flag'].sum()
```

Input:13

```
sub = sub.drop('NaT')
```

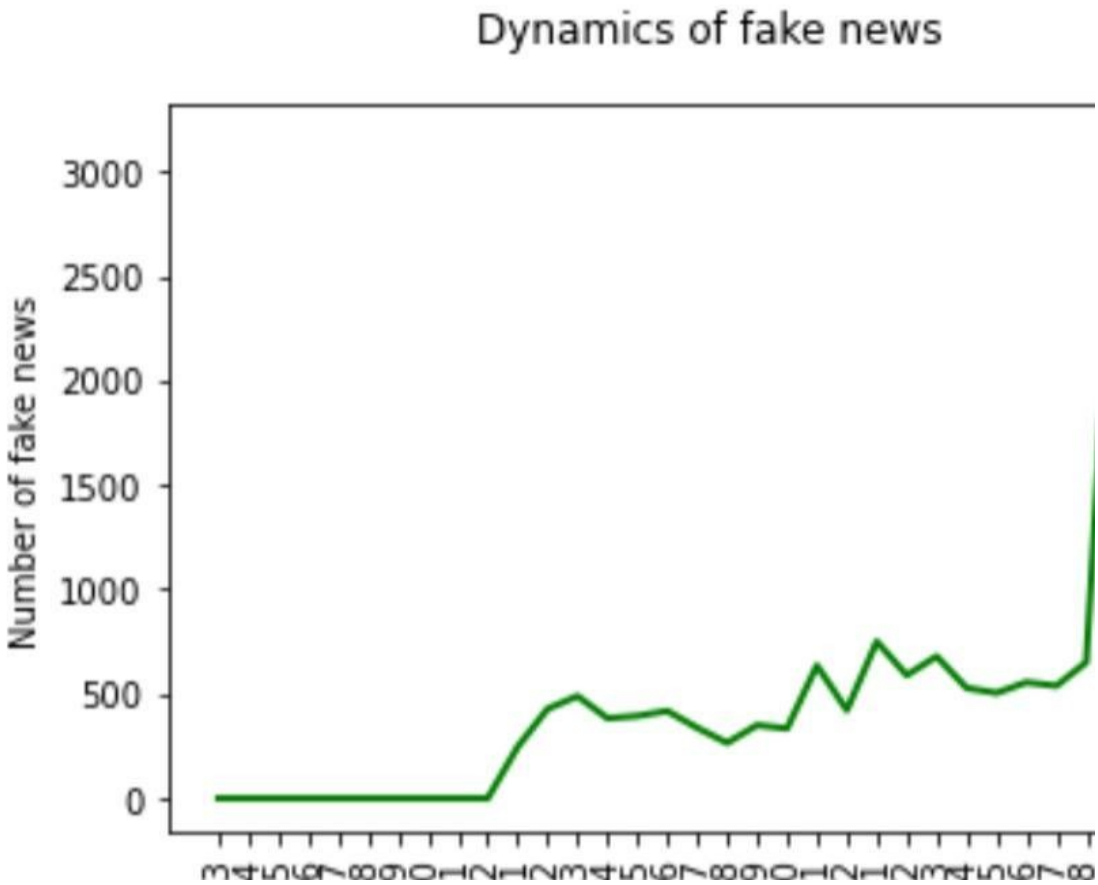
Input:14

```
import matplotlib.pyplot as plt
```

```
plt.suptitle('Dynamics of fake news')
plt.xticks(rotation=90)
plt.ylabel('Number of fake news')
plt.xlabel('Month-Year')
plt.plot(sub.index, sub.values, linewidth=2, color='green')
```

Output:14

```
[<matplotlib.lines.Line2D at 0x7f
24f10>]
```



Input:15

```
sub2 = df[['subject', 'flag']]
sub2 = sub2.dropna()
sub2 = sub2.groupby(['subject'])['flag'].sum()
```

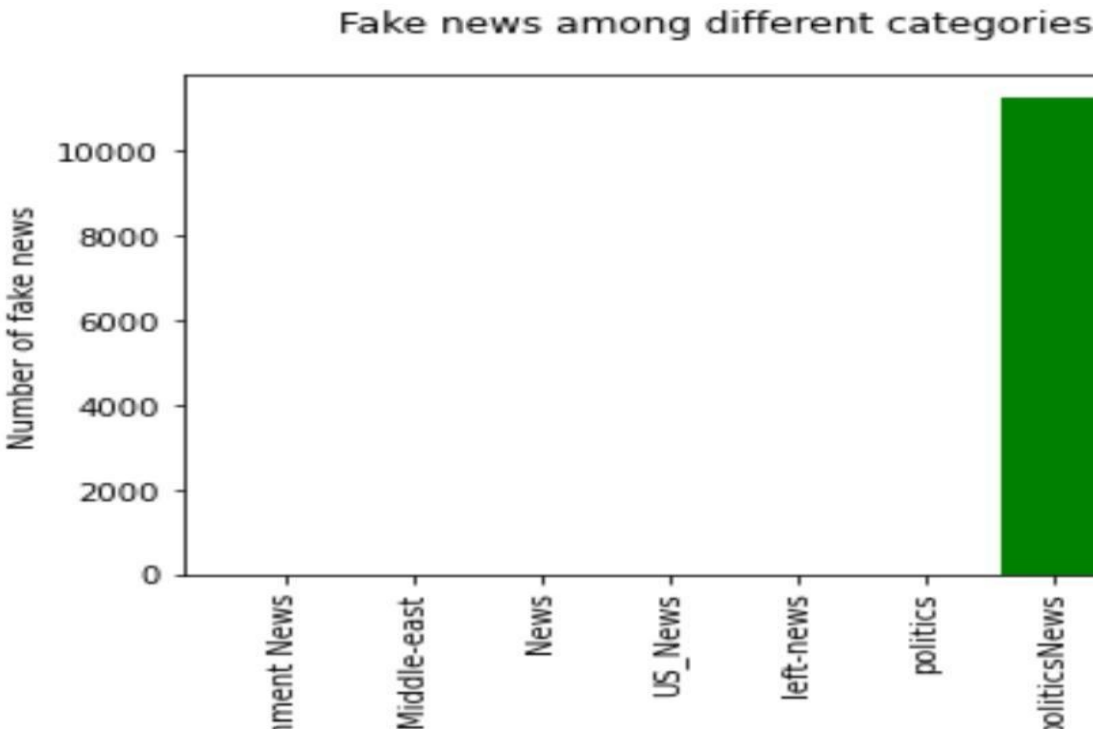
Input:16

```
plt.suptitle('Fake news among different categories')
plt.xticks(rotation=90)
plt.ylabel('Number of fake news')
plt.xlabel('Category')

plt.bar(sub2.index, height=sub2.values, color='green')
```

Output:16

```
<BarContainer object of 8 artists>
```



Input:17

```
nlp = df
```

Input:18

```
from sklearn.feature_extraction.text import TfidfVectorizer

corpus = nlp[nlp['flag'] == 1]['title'].iloc[0:500]
tfidf1 = TfidfVectorizer()
vecs = tfidf1.fit_transform(corpus)

feature_names = tfidf1.get_feature_names()
dense = vecs.todense()
list_words = dense.tolist()
df_words = pd.DataFrame(list_words, columns=feature_names)
```

Input:19

```
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator
df_words.T.sum(axis=1)
Cloud = WordCloud(background_color="white",
max_words=100).generate_from_frequencies(df_words.T.sum(axis=1
))
```

Input:20

```
import matplotlib.pyplot as plt
plt.figure(figsize=(12,5))
plt.imshow(Cloud, interpolation='bilinear')
```

```
<matplotlib.image.AxesImage at 0x1
19d6490>
```

Output:20

Input:21

```
import nltk
nltk.download('punkt')
from nltk import word_tokenize

nlp['title'] = nlp['title'].apply(lambda x: word_tokenize(str(x)))
```

Input:22

```
from nltk.stem import SnowballStemmer

snowball = SnowballStemmer(language='english')
nlp['title'] = nlp['title'].apply(lambda x: [snowball.stem(y) for y in x])
```

Input:23

```
nlp['title'] = nlp['title'].apply(lambda x: ' '.join(x))
```

Input:24

```
from nltk.corpus import stopwords
```

```
nltk.download('words')
nltk.download('stopwords')
stopwords = stopwords.words('english')
```

Input:25

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer()
X_text = tfidf.fit_transform(nlp['title'])
```

Input:26

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_text, nlp['flag'],
test_size=0.33, random_state=1)
```

Input:27

```
scores = {}
```

Input:28

```
from sklearn.svm import LinearSVC
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score

clf = LinearSVC(max_iter=100, C=1.0)
clf.fit(X_train, y_train)

y_pred_SVM = clf.predict(X_test)
print(cross_val_score(clf, X_text, nlp['flag'], cv=3))
print(accuracy_score(y_pred_SVM, y_test))

scores['LinearSVC'] = accuracy_score(y_pred_SVM, y_test)
```

Output:28

```
[0.91105592 0.93031686 0.92696026]
0.958706265256306
```

Input:29

```
from sklearn.naive_bayes import MultinomialNB

clf2 = MultinomialNB()
clf2.fit(X_train, y_train)

y_pred_MNB = clf2.predict(X_test)
print(cross_val_score(clf2, X_text, nlp['flag'], cv=3))
print(accuracy_score(y_pred_MNB, y_test))

scores['MultinomialNB'] = accuracy_score(y_pred_MNB, y_test)
```

Output:29

```
[0.88957508 0.89406552 0.92883996]
0.939924057499322
```

Input:30

```
from xgboost import XGBClassifier

clf3 = XGBClassifier(eval_metric='rmse', use_label_encoder=False)
clf3.fit(X_train, y_train)

y_pred_XGB = clf3.predict(X_test)
print(cross_val_score(clf3, X_text, nlp['flag'], cv=3))
print(accuracy_score(y_pred_XGB, y_test))
```

```
scores['XGB'] = accuracy_score(y_pred_XGB, y_test)
```

Output:30

```
[0.88615157 0.92353652 0.90695489]
0.9374830485489558
```

Input:31

```
pip install pycaret
```

Input:32

```
from pycaret.nlp import *

caret_nlp = setup(data=nlp, target='title', session_id=1)
```

Description	Val
session_id	1
Documents	44
Vocab Size	75

Output:32

Input:33

```
lda = create_model('lda')
```

Input:34

```
lda_data = assign_model(lda)
```

Input:35

```
lda_data
```

	title	text
0	budget fight loom flip fiscal script	WASHINGTON (Reuters) - The head of a conservat...
1	transgend	WASHINGTON (Reuters) - Transgender people will...
2	senior let job	WASHINGTON (Reuters) - The special counsel inv...
3	diplomat	WASHINGTON (Reuters) - Trump campaign adviser ...
4	trump want postal charg much shipment	SEATTLE/WASHINGTON (Reuters) - President Donal...
...	...	...
	furious	21st Century Wire says

Output:35

Input:36

```
from catboost import CatBoostClassifier
```

Input:37

```
input_cat =
lda_data.drop(['text','date','Perc_Dominant_Topic','flag','year'],
axis=1)
input_cat['month'] = input_cat['month'].astype(str)
target_cat = lda_data['flag']
```

Input:38

```
from sklearn.model_selection import train_test_split
X_train_cat, X_test_cat, y_train_cat, y_test_cat =
train_test_split(input_cat, target_cat, test_size=0.33, random_state=1)
```

Input:39

```
clf4 = CatBoostClassifier(iterations=1000,
cat_features=['title','subject','Dominant_Topic','month']
)
```

Input:40

```
clf4.fit(X_train_cat, y_train_cat, early_stopping_rounds=10)
```

Output:40

```
<catboost.core.CatBoostClassifier at 0x7f167ddb8a50>
```

Input:41

```
scores['CatBoost'] = clf4.score(X_test_cat, y_test_cat)
```

Input:42

```
scores['CatBoost'] = clf4.score(X_test_cat, y_test_cat)
```

Output:42

{'LinearSVC': 0.958706265256306,
'MultinomialNB': 0.939924057499322,
'XGB': 0.9374830485489558,
'CatBoost': 1.0}

Input:43

```
plt.bar(scores.keys(), scores.values())
```

```
<BarContainer object of 4 artists>
```

Output:43

FAKE NEWS

- ✓ A sort of sensationalist reporting, counterfeit news embodies bits of information that might be lies and is, for the most part, spread through web-based media and other online media.

- ✓ This is regularly done to further or force certain kinds of thoughts or for false promotion of products and is frequently accomplished with political plans.

- ✓ Such news things may contain bogus and additionally misrepresented cases and may wind up being virtualized by calculations, and clients may wind up in a channel bubble.

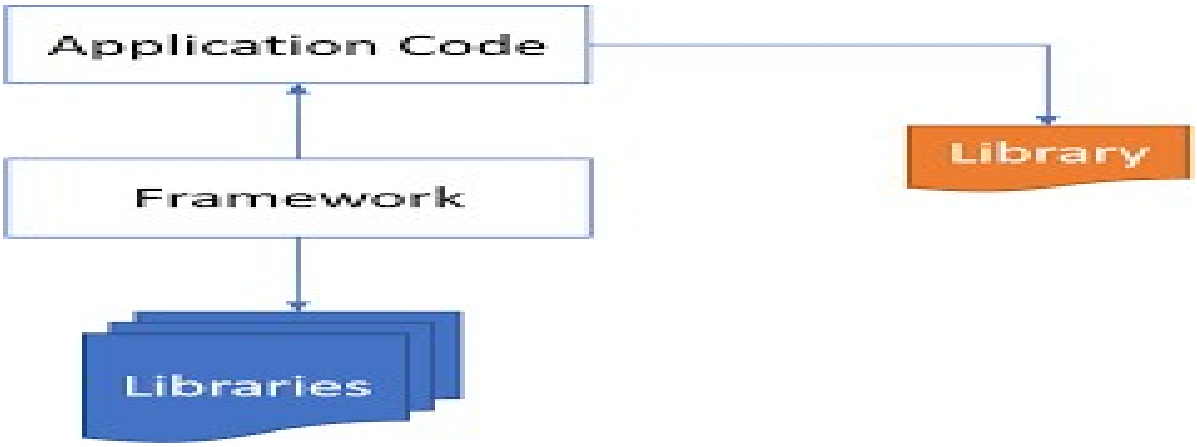
DATA ANALYSIS

Here I will explain the dataset.

In this python project, we have used the CSV dataset. The dataset contains 7796 rows and 4 columns.
This dataset has four columns,

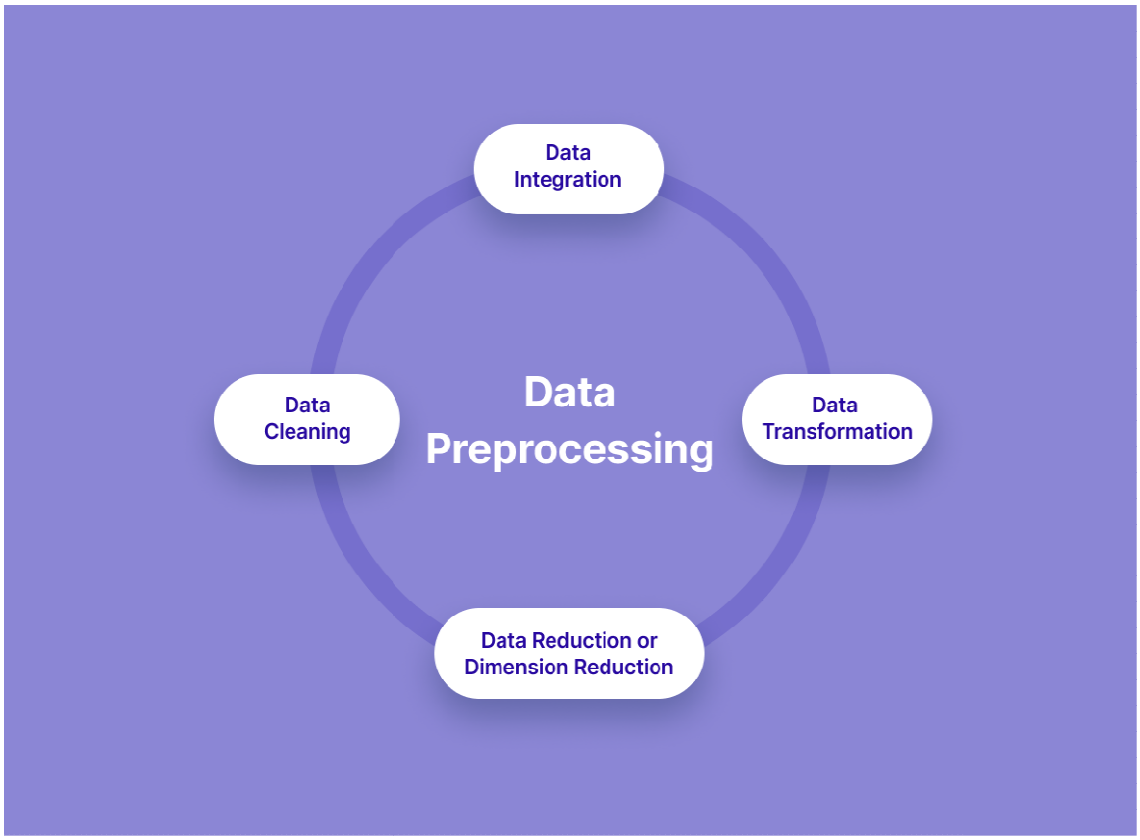
1. **title**: this represents the title of the news.
2. **author**: this represents the name of the author who has written the news.
3. **text**: this column has the news itself.
4. **label**: this is a binary column representing if the news is fake (1) or real (0).

LIBRARIES



The very basic data science libraries are sklearn, pandas, NumPy e.t.c and some specific libraries such
Is transformers.

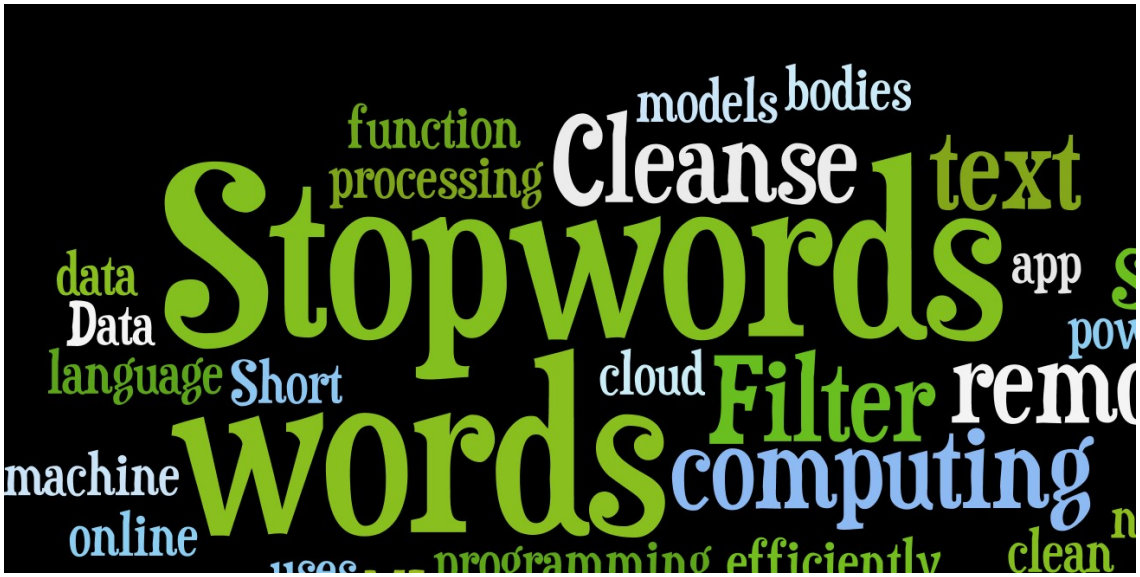
DATA PREPROCESSING



- ✓ In data processing, we will focus on the text column on this data which actually contains the news part.
- ✓ We will modify this text column to extract more information to make the model more predictable.
- ✓ To extract information from the text column, we will use a library, which we know by the name of '**nltk'**.
- ✓ Here we will use functionalities of the **'nltk'** library named Removing Stopwords, Tokenization, and Lemmatization.

  So we will see these functionalities one by one with these three examples. Hope you will have a better understanding of extracting information from the text column after this.

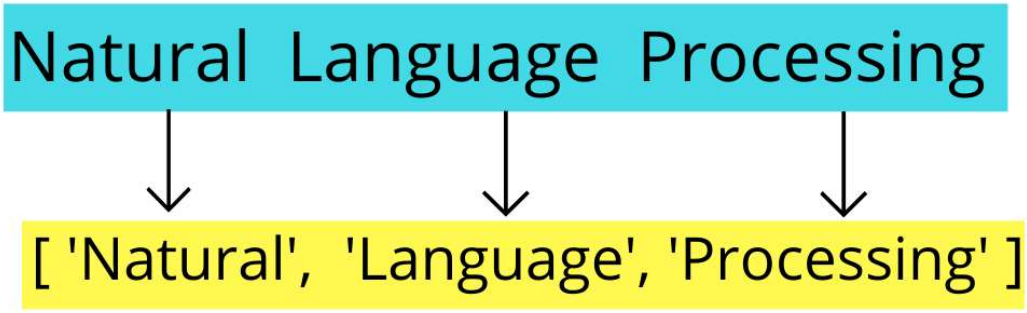
REMOVING STOPWORDS



- ✓ These are the words that are used in any language used to connect words or used to declare the tense of sentences.
- ✓ This means that if we use these words in any they do not add much meaning to the context of the sentence so even after removing the stopwords we can understand the context.

TOKENIZATION

Tokenization

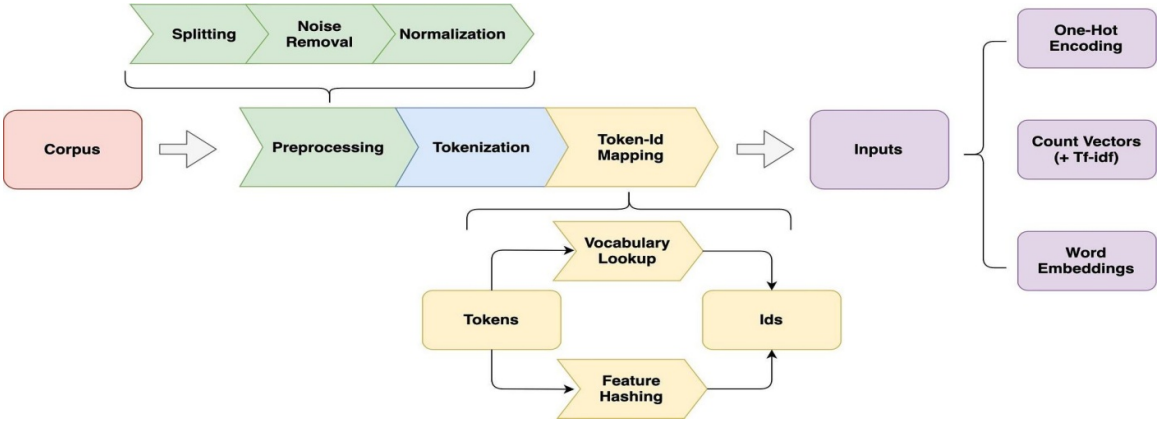


- ✓ Tokenization is the process of breaking text into smaller pieces which we know as tokens.Each word, special character, or number in a sentence can be depicted as a token in NLP.
- ✓ Tokenization is the process of breaking down a piece of code into smaller units called tokens.

CONVERTING LABELS

Tokenization is the process of breaking text into smaller pieces which we know as tokens.Each word, special character, or number in a sentence can be depicted as a token in NLP.Tokenization is the process of breaking down a piece of code into smaller units called tokens.

```
df.label = df.label.astype(str)
df.label = df.label.str.strip()
dict = { 'REAL' : '1' , 'FAKE' : '0'}
df['label'] = df['label'].map(dict)df.head()
```



VECTORIZATION

- ✓ Vectorization is a methodology in NLP to map words or phrases from vocabulary to a corresponding vector of real numbers which is used to find word predictions, word similarities/semantics.
- ✓ To make documents' corpora more relatable for computers, they must first be converted into some numerical structure. There are few techniques that are used to achieve this such as 'Bag of Words'.
- ✓ Here, we are using vectorizer objects provided by Scikit-Learn which are quite reliable right out of the box.

THE MOST USED VECTORIZERS

Count Vectorizer: The most straightforward one, it counts the number of times a token shows up in the document and uses this value as its weight.

Hash Vectorizer: This one is designed to be as memory efficient as possible. Instead of storing the tokens as strings, the vectorizer applies the hashing trick to encode them as numerical indexes. The downside of this method is that once vectorized, the features' names can no longer be retrieved.

TF-IDF Vectorizer: TF-IDF stands for "term frequency-inverse document frequency", meaning the weight assigned to each token not only depends on its frequency in a document but also how recurrent that term is in the entire corpora. More on that here.

```
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
count_vectorizer = CountVectorizer()
count_vectorizer.fit_transform(x_df)
freq_term_matrix = count_vectorizer.transform(x_df)
tfidf = TfidfTransformer(norm = "l2")
tfidf.fit(freq_term_matrix)
tf_idf_matrix = tfidf.fit_transform(freq_term_matrix)
print(tf_idf_matrix)
```

MODELLING

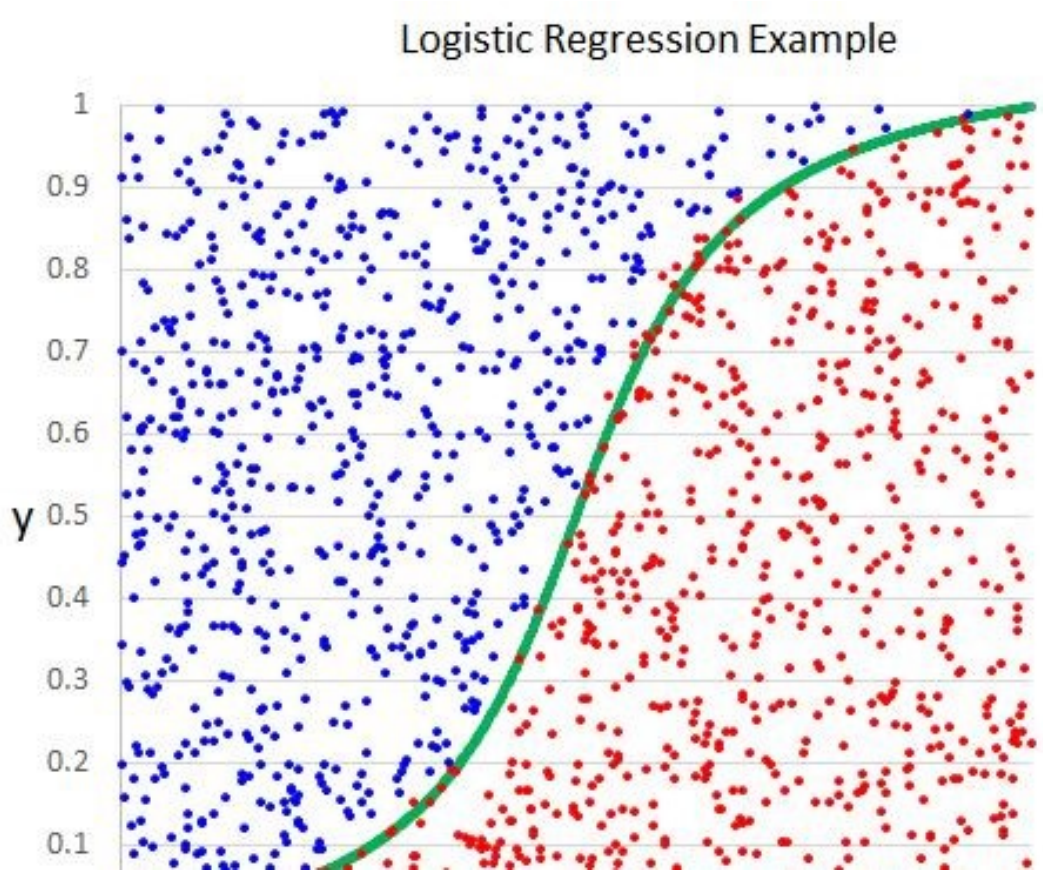
After Vectorization, we split the data into test and train data.
Splitting the data into test data and train data

x_train, x_test, y_train, y_test = train_test_split(tf_idf_matrix,y_df, random_state=0)

I fit four ML models to the data,
Logistic Regression, Naive-Bayes, Decision Tree, and Passive-Aggressive Classifier.
After that, predicted on the test set from the TfidfVectorizer and calculated the accuracy with accuracy_score() from sklearn. metrics.

LOGISTIC REGRESSION

In natural language processing, logistic regression is the base- line supervised machine learning algorithm for classification, and also has a very close relationship with neural networks.

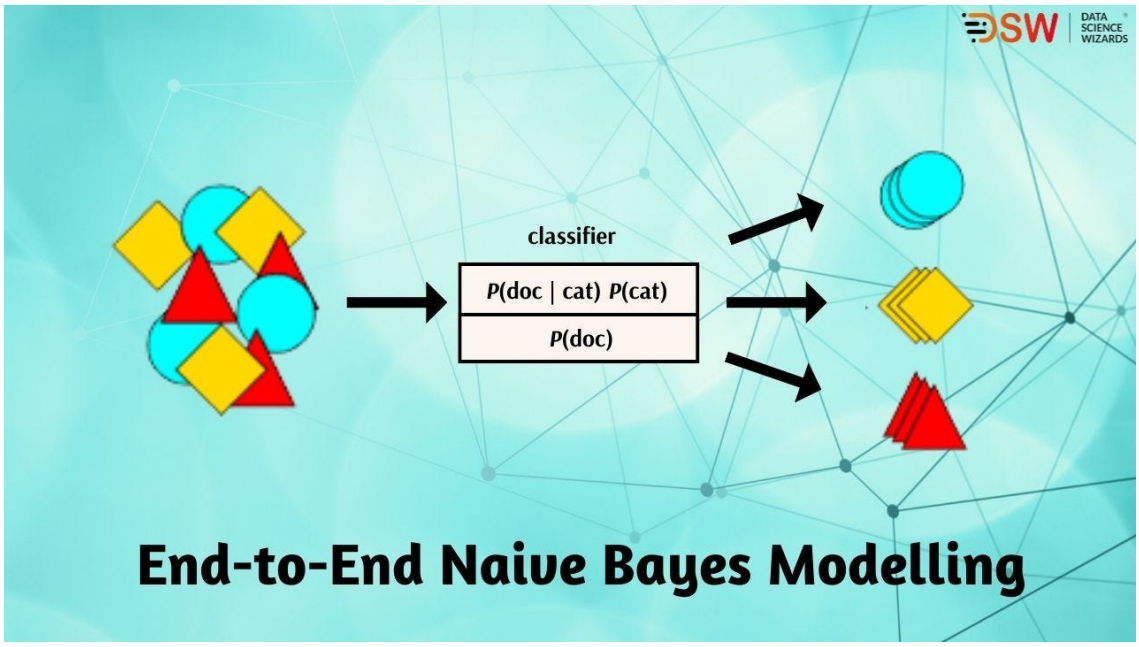



```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(x_train, y_train)
Accuracy = logreg.score(x_test, y_test)
print(Accuracy*100)
```

Accuracy: 91.73%

NAVIE BAYES

- ❖ The Naive Bayes algorithm is a supervised machine learning algorithm based on the Bayes' theorem.
- ❖ It is a probabilistic classifier that is often used in NLP tasks like sentiment analysis (identifying a text corpus' emotional or sentimental tone or opinion).

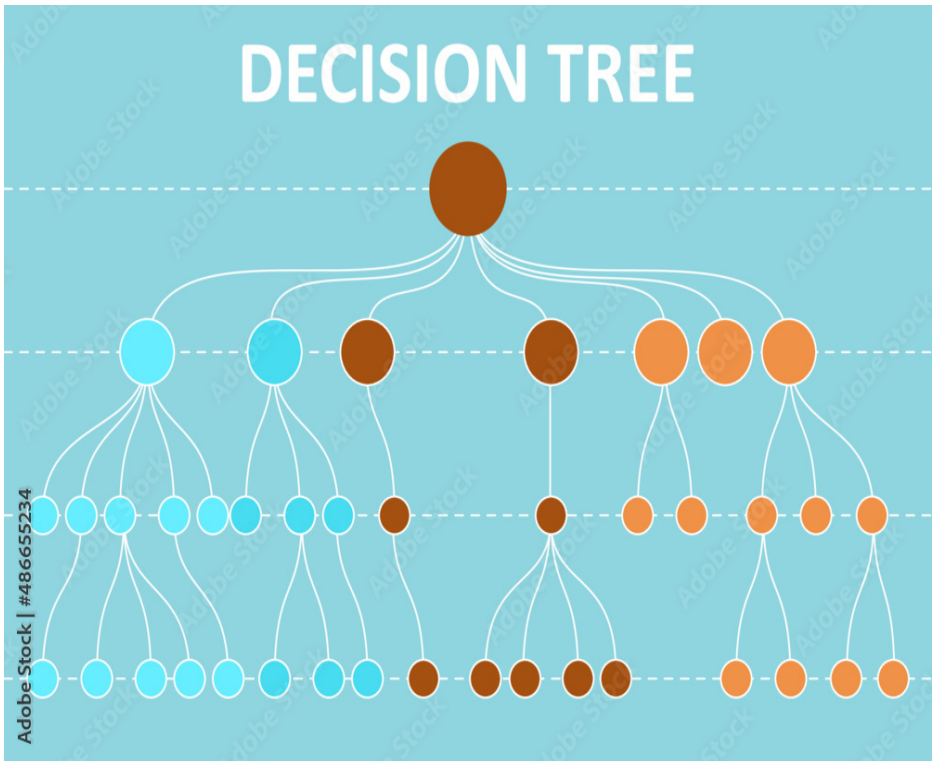



```
from sklearn.naive_bayes import MultinomialNB
NB = MultinomialNB()
NB.fit (x_train, y_train)
Accuracy = NB.score(x_test, y_test)
Print (Accuracy*100)
```

Accuracy: 82.32 %

DECISION TREE

- ✓ Decision trees are induced with three algorithms; the first two produce generalized trees, while the third produces binary trees.
- ✓ To meet the requirements of the linguistic datasets, all three algorithms are able to handle set-valued attributes.



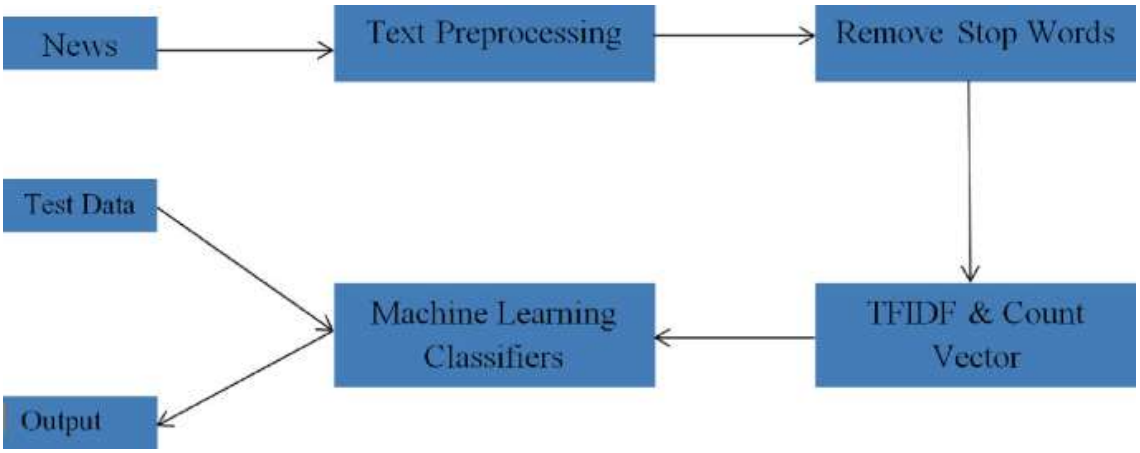

```
from sklearn.tree import DecisionTreeClassifierclf
= DecisionTreeClassifier()

clf.fit (x_train, y_train)
Accuracy = clf.score(x_test, y_test)
Print (Accuracy*100)
```

Accuracy: 80.49%

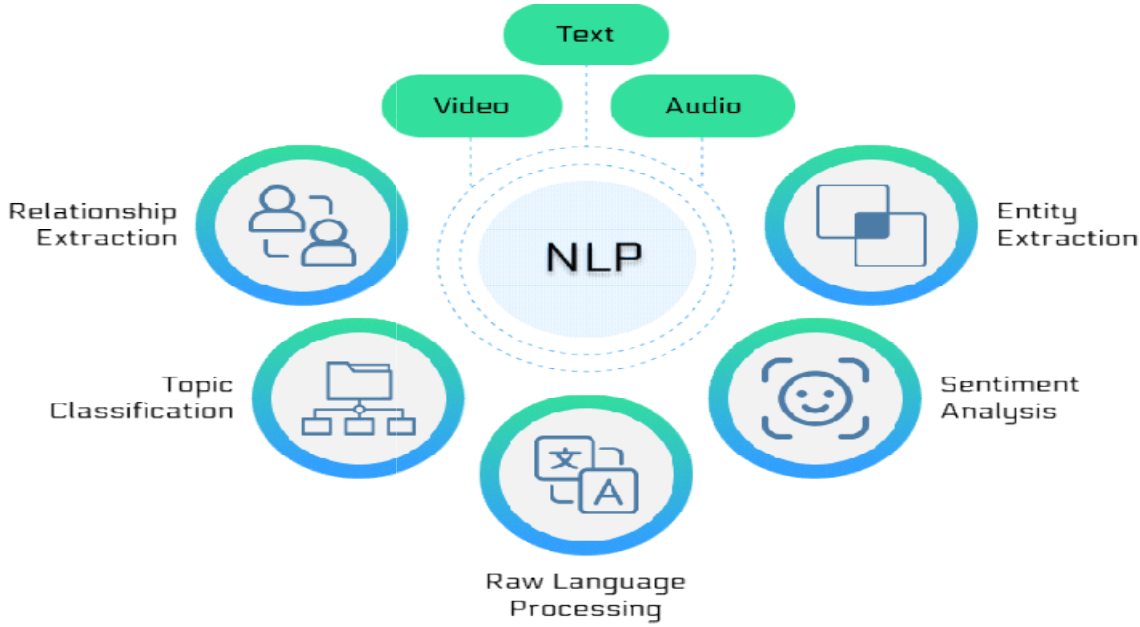
PASSIVE AGGRESSIVE CLASSIFER

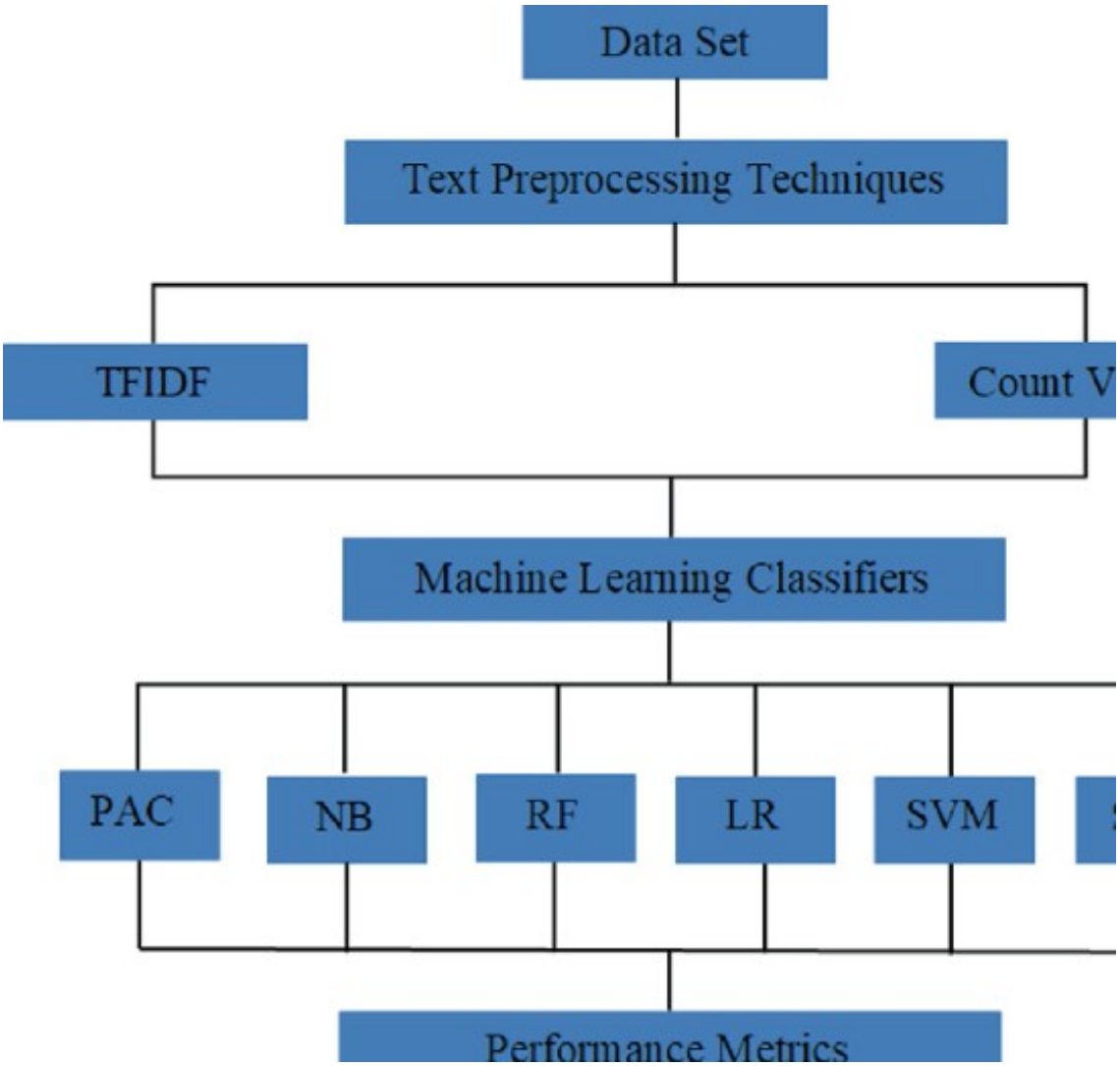
- ✓ Passive Aggressive is considered algorithms that perform online learning (with for example Twitter data).
- ✓ Their characteristic is that they remain passive when dealing with an outcome that has been correctly classified, and become aggressive when a miscalculation takes place, thus constantly self-updating and adjusting.




```
from sklearn.metrics import accuracy_score
from sklearn.linear_model import
PassiveAggressiveClassifier
pac=PassiveAggressiveClassifier(max_iter=50)
pac.fit(x_train,y_train)
#Predict on the test set and calculate accuracy
y_pred=pac.predict(x_test)
score=accuracy_score(y_test,y_pred)
print (f'Accuracy: {round(score*100,2)} %'))
```

Output: Accuracy: 93.12%





CONCLUSION

- ✓ The passive-aggressive classifier performed the best here and gave an accuracy of 93.12%.
- ✓ We can print a confusion matrix to gain insight into the number of false and true negatives and positives
- ✓ Fake news detection techniques can be divided into those based on style and those based on content, or fact-checking. Too often it is assumed that bad style (bad spelling, bad punctuation, limited vocabulary, using terms of abuse, ungrammaticality, etc.) is a safe indicator of fake news.

- ✓ More than ever, this is a case where the machine's opinion must be backed up by clear and fully verifiable indications for the basis of its decision, in terms of the facts checked and the authority by which the truth of each fact was determined.
- ✓ Collecting the data once isn't going to cut it given how quickly information spreads in today's connected world and the number of articles being churned out.
- ✓ I hope you might find this helpful. You can comment down in the comment sections for any queries.