Cover Page

Bahrain Polytechnic بوليتكنك البحرين

MSc in AI - Assessment Cover Sheet

Complete and attach this cover sheet to your assessment before submitting

Course Code and Title:	IT9002– Natural Language Processing
Assessment Title:	Fake News Classification

Learning Outcomes:

LO1 – Understanding the critical knowledge of fundamental concepts, algorithms, and models in Natural Language processing (NLP) for performing various linguistic NLP tasks.

LO2 - Demonstrate professional levels of insight, interpretation by utilizing the various NLP packages like Natural Language Tool Kit (NLTK) to apply, solve, implement, evaluate, and improve the real time significant applications of NLP

Student IDs:	202306803
Student Names:	Maram Aljasim
Tutor:	Sini Raj Pulari
Individual Project Submission Date	**Wednesday, 13th December 2023 by 11:55 p.m.**
Late Rule:	**The maximum grade granted for late submission is 60 % for up to 3 calendar days. A grade of 0 will be allocated for submission after 3 days**

By submitting this assessment for marking, either electronically or as hard copy, I confirm the following:

- *This assignment is **our own work***
- *Any information used has been properly referenced.*
- *I understand that a copy of my work may be used for moderation.*
- *I have kept a copy of this assignment*

Do not write below this line. For Polytechnic use only.

Assessor:	**Date of Marking:**
Grade/Mark:	

Table of Contents

Introduction

The modern digital era's information flood has caused a fast and vast spreading of news, accompanied by the proliferation of fake news, a disruptive force undermining trust in credible sources and distorting public discourse. Addressing this challenge necessitates innovative approaches leveraging machine learning and Natural Language Processing (NLP) techniques. This project aims to develop a robust machine learning model harnessing NLP methodologies to discern genuine news from false information. It encompasses data preprocessing, converting textual data into machine-understandable formats, employing various NLP techniques in model training, and careful performance evaluation.

Task 1 – Problem Statement Formulation and definition

Task 1.1 - Motivation

In the context of cybersecurity, the rapid increase of fake news has become a significant concern. Misinformation and disinformation campaigns can lead to severe consequences, including social unrest, financial loss, and compromised security. The motivation here is to leverage Natural Language Processing (NLP) techniques to develop a system capable of identifying and classifying fake news, thereby enhancing cyber defense strategies.

Task 1.2 - Problem Statement / Project Definition

The primary objective of this project is to develop a robust machine learning model using NLP techniques to classify news articles or textual content as either genuine or fake. This involves preprocessing and cleaning the dataset, representing the textual information in a format suitable for machine learning algorithms, training the model using various NLP techniques, and evaluating its performance.

Task 1.3 - Expected Result

- **Accurate Classification**: Develop a model that accurately classifies news articles as real or fake, thereby aiding in the identification of potentially harmful or misleading content.
- **Comparison and Analysis**: Evaluate and compare the performance of multiple machine learning algorithms to determine the most effective approach for fake news classification.

Task 2 – Selection of an Appropriate Data Set (Data Collection)

In [1]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, ca
ll drive.mount("/content/drive", force_remount=True).
```

Task 2.1 - Source of Dataset

- **Dataset Name**: Fake News Classification on WELFake Dataset
- **Dataset Link**: https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification
- **Owner of Dataset**: Saurabh Shahane
- **Update Frequency**: Updated 2 months ago
- **Motivation Behind Choosing the Dataset**: The dataset, WELFake, was selected for its substantial volume of news articles (72,134 entries), comprising both real (35,028 or 48.56%) and fake (37,106 or 51.44%) news. This dataset combines information from multiple reputable sources like Kaggle, McIntire, Reuters, and BuzzFeed Political. The reason for choosing this dataset is to avoid classifier overfitting while providing a diverse range of textual data for robust machine learning training.
- **Dataset Type**: CSV file
- **Size of the Dataset**: 245.09 MB
- **Shape of the Dataset**:
  - Data Frame consists of 72,134 entries
    - Real news (35,028 entries or 48.56%)
    - Fake news (37,106 enteries or 51.44%)
  - Each entry has four columns:
    - *Serial number*

    - *Title*
    - *Text*
    - *Label*
- **'Label' column** denotes whether the news is fake (0) or real (1), so it is numerical in nature.

In [2]:
```
import pandas as pd
import numpy as np
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/NLP_Project/WELFake
display(df.shape)
```

```
(72134, 4)
```

Task 2.2 - Visualize the Dataset

In [3]:
```
df.head(10)
```

Out[3]:

	Unnamed: 0	title	text	label
0	0	LAW ENFORCEMENT ON HIGH ALERT Following Threat...	No comment is expected from Barack Obama Membe...	1
1	1	NaN	Did they post their votes for Hillary already?	1
2	2	UNBELIEVABLE! OBAMA'S ATTORNEY GENERAL SAYS MO...	Now, most of the demonstrators gathered last ...	1
3	3	Bobby Jindal, raised Hindu, uses story of Chri...	A dozen politically active pastors came here f...	0
4	4	SATAN 2: Russia unvelis an image of its terrif...	The RS-28 Sarmat missile, dubbed Satan 2, will...	1
5	5	About Time! Christian Group Sues Amazon and SP...	All we can say on this one is it s about time ...	1
6	6	DR BEN CARSON TARGETED BY THE IRS: "I never ha...	DR. BEN CARSON TELLS THE STORY OF WHAT HAPPENE...	1
7	7	HOUSE INTEL CHAIR On Trump-Russia Fake Story: ...		1
8	8	Sports Bar Owner Bans NFL Games...Will Show Only...	The owner of the Ringling Bar, located south o...	1
9	9	Latest Pipeline Leak Underscores Dangers Of Da...	FILE – In this Sept. 15, 2005 file photo, the ...	1

In [4]:
```
df["label"].value_counts()
```

Out[4]:
```
1    37106
0    35028
Name: label, dtype: int64
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 72134 entries, 0 to 72133
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  72134 non-null  int64 
 1   title       71576 non-null  object
 2   text        72095 non-null  object
 3   label       72134 non-null  int64 
dtypes: int64(2), object(2)
memory usage: 2.2+ MB
```

Task 2.3 - Clean the Dataset

In [6]:
```
df.drop(columns='Unnamed: 0', inplace=True)
df.head(5)
```

Out[6]:

	title	text	label
0	LAW ENFORCEMENT ON HIGH ALERT Following Threat...	No comment is expected from Barack Obama Membe...	1
1	NaN	Did they post their votes for Hillary already?	1
2	UNBELIEVABLE! OBAMA'S ATTORNEY GENERAL SAYS MO...	Now, most of the demonstrators gathered last ...	1
3	Bobby Jindal, raised Hindu, uses story of Chri...	A dozen politically active pastors came here f...	0
4	SATAN 2: Russia unvelis an image of its terrif...	The RS-28 Sarmat missile, dubbed Satan 2, will...	1

In [7]:
```
df.dropna(inplace=True)
df = df[:10000]
display(df.shape)
```

```
(10000, 3)
```

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10000 entries, 0 to 10072
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   title   10000 non-null  object
 1   text    10000 non-null  object
 2   label   10000 non-null  int64 
dtypes: int64(1), object(2)
memory usage: 312.5+ KB
```

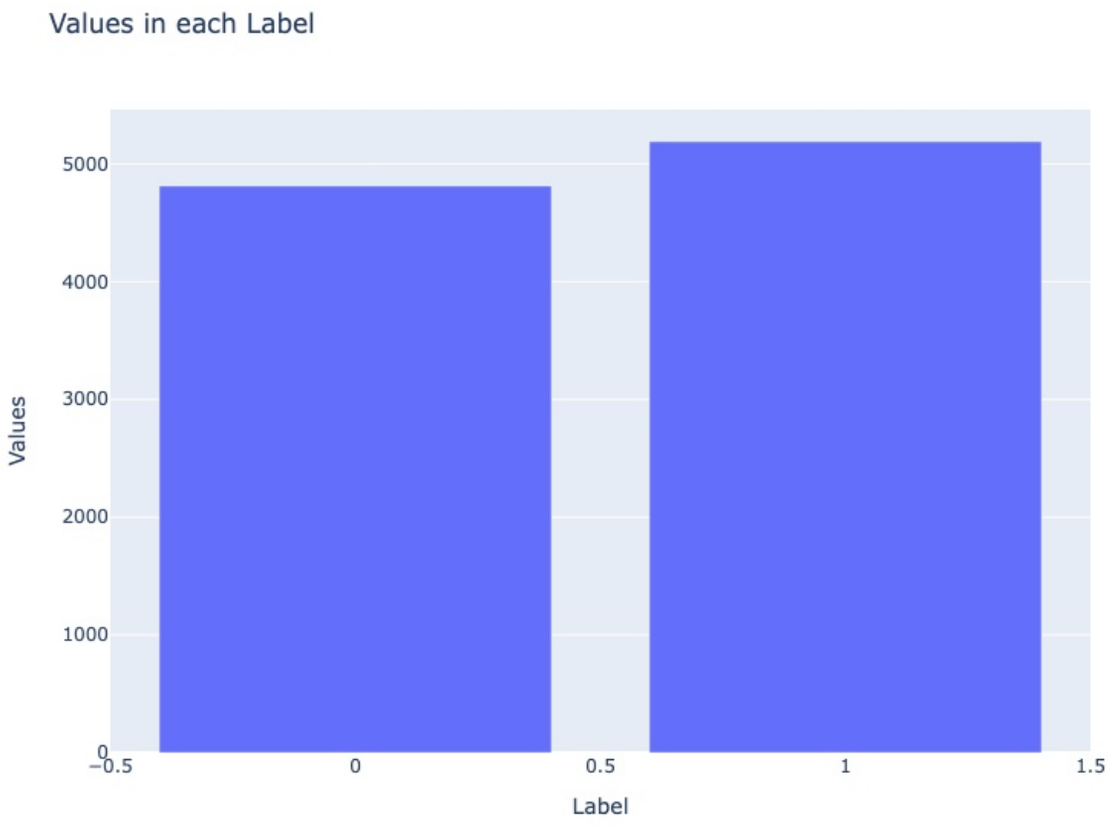
In [9]:
```
df["label"].value_counts()/df.shape[0]
```

Out[9]:
```
1    0.5188
0    0.4812
Name: label, dtype: float64
```

In [10]:
```
import plotly.graph_objects as go
```

In [11]:
```
fig = go.Figure([go.Bar(x=df['label'].value_counts().index, y=df['label'].va
fig.update_layout(
```

```
    title="Values in each Label",
    xaxis_title="Label",
    yaxis_title="Values",
    height=600, width=800
    )
fig.show()
```



Task 3 – Text Preprocessing

The text preprocessing techniques that have been implemented are:

- Lower Casing
- Punctuation Removal
- Stopword Removal
- Lemmatization

These text preprocessing techniques collectively assist in:

- **Noise Reduction**: Removing irrelevant information like punctuation and stopwords.
- **Normalization**: Standardizing text, reducing the vocabulary size, and ensuring consistency in representation.

- **Improving Model Performance**: By providing clean and more focused data to machine learning models, which in turn helps in better understanding and predicting patterns within text data.

By employing these techniques, the text data becomes more refined, structured, and conducive for effective analysis, leading to improved performance in NLP models.

Task 3.1 - Lower Casing

Lowercasing is a fundamental step that converts all text to lowercase. It helps standardize the text and avoids treating the same word differently due to variations in capitalization.

The output can be shown in the new columns created below "title_lower" and "text_lower". As well as a sample row printed to show the difference between the original and the lowercased sentence.

In [12]:
```
#Lower Casing

def text_lowercase(text):
  return text.lower()

df["title_lower"] = df["title"].apply(text_lowercase)
df["text_lower"] = df["text"].apply(text_lowercase)
df.head()
```

Out[12]:

	title	text	label	title_lower	text_lower
0	LAW ENFORCEMENT ON HIGH ALERT Following Threat...	No comment is expected from Barack Obama Membe...	1	law enforcement on high alert following threat...	no comment is expected from barack obama membe...
2	UNBELIEVABLE! OBAMA'S ATTORNEY GENERAL SAYS MO...	Now, most of the demonstrators gathered last ...	1	unbelievable! obama's attorney general says mo...	now, most of the demonstrators gathered last ...
3	Bobby Jindal, raised Hindu, uses story of Chri...	A dozen politically active pastors came here f...	0	bobby jindal, raised hindu, uses story of chri...	a dozen politically active pastors came here f...
4	SATAN 2: Russia unvelis an image of its terrif...	The RS-28 Sarmat missile, dubbed Satan 2, will...	1	satan 2: russia unvelis an image of its terrif...	the rs-28 sarmat missile, dubbed satan 2, will...
5	About Time! Christian Group Sues Amazon and SP...	All we can say on this one is it s about time ...	1	about time! christian group sues amazon and sp...	all we can say on this one is it s about time ...

In [13]:
```
print("First row of 'title':")
print(df.loc[0, 'title'])

print("\nFirst row of Lower Cased title:")
print(df.loc[0, 'title_lower'])
```

```
First row of 'title':
LAW ENFORCEMENT ON HIGH ALERT Following Threats Against Cops And Whites On
9-11By #BlackLivesMatter And #FYF911 Terrorists [VIDEO]

First row of Lower Cased title:
law enforcement on high alert following threats against cops and whites on
9-11by #blacklivesmatter and #fyf911 terrorists [video]
```

Task 3.2 - Punctuation Removal

Removing punctuation involves eliminating any punctuations (like '.', ',', '?') from the text. It's beneficial as punctuation often doesn't add significant value to NLP tasks.

The output can be shown in the new columns created below "title_wo_punct" and "text_wo_punct". As well as a sample row printed to show the difference between the previous step and the sentence without the punctuation.

In [14]:
```
#Removal of Punctuations
import string

PUNCT_TO_REMOVE = string.punctuation
def remove_punctuation(text):
    return text.translate(str.maketrans('', '', PUNCT_TO_REMOVE))

df["title_wo_punct"] = df["title_lower"].apply(lambda text: remove_punctuati
df["text_wo_punct"] = df["text_lower"].apply(lambda text: remove_punctuatio
df.head()
```

Out[14]:

	title	text	label	title_lower	text_lower	title_wo_punct	text_wo
0	LAW ENFORCEMENT ON HIGH ALERT Following Threat...	No comment is expected from Barack Obama Membe...	1	law enforcement on high alert following threat...	no comment is expected from barack obama membe...	law enforcement on high alert following threat...	no com expecte barack m
2	UNBELIEVABLE! OBAMA'S ATTORNEY GENERAL SAYS MO...	Now, most of the demonstrators gathered last ...	1	unbelievable! obama's attorney general says mo...	now, most of the demonstrators gathered last ...	unbelievable obama's attorney general says mos...	now demons gathe
3	Bobby Jindal, raised Hindu, uses story of Chri...	A dozen politically active pastors came here f...	0	bobby jindal, raised hindu, uses story of chri...	a dozen politically active pastors came here f...	bobby jindal raised hindu uses story of christ...	a pc active came
4	SATAN 2: Russia unvelis an image of its terrif...	The RS-28 Sarmat missile, dubbed Satan 2, will...	1	satan 2: russia unvelis an image of its terrif...	the rs-28 sarmat missile, dubbed satan 2, will...	satan 2 russia unvelis an image of its terrify...	the rs28 missile satan 2
5	About Time! Christian Group Sues Amazon and SP...	All we can say on this one is it s about time ...	1	about time! christian group sues amazon and sp...	all we can say on this one is it s about time ...	about time christian group sues amazon and spl...	all we on this s about

In [15]:
```
print("First row of Lower Cased title:")
print(df.loc[0, 'title_lower'])
```

```
print("\nFirst row of title without Punctuation:")
print(df.loc[0, 'title_wo_punct'])
```

```
First row of Lower Cased title:
law enforcement on high alert following threats against cops and whites on
9-11by #blacklivesmatter and #fyf911 terrorists [video]

First row of title without Punctuation:
law enforcement on high alert following threats against cops and whites on
911by blacklivesmatter and fyf911 terrorists video
```

Task 3.3 - Stopword Removal

Stopwords are common words (e.g., 'the', 'is', 'and') that are often removed from text data as they occur frequently and don't add much meaning to the context.

The output can be shown in the new columns created below "title_wo_stop" and "text_wo_stop". As well as a sample row printed to show the difference between the previous step and the sentence without the stopwords.

In [16]:
```
import nltk
from nltk.corpus import stopwords

nltk.download('stopwords')

STOPWORDS = set(stopwords.words('english'))
def remove_stopwords(text):
    return " ".join([word for word in str(text).split() if word not in STOPW

df["title_wo_stop"] = df["title_wo_punct"].apply(lambda text: remove_stopwo
df["text_wo_stop"] = df["text_wo_punct"].apply(lambda text: remove_stopwords
df.head()
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[16]:

	title	text	label	title_lower	text_lower	title_wo_punct	text_wo
0	LAW ENFORCEMENT ON HIGH ALERT Following Threat...	No comment is expected from Barack Obama Membe...	1	law enforcement on high alert following threat...	no comment is expected from barack obama membe...	law enforcement on high alert following threat...	no com expecte barack m
2	UNBELIEVABLE! OBAMA'S ATTORNEY GENERAL SAYS MO...	Now, most of the demonstrators gathered last ...	1	unbelievable! obama's attorney general says mo...	now, most of the demonstrators gathered last ...	unbelievable obama's attorney general says mos...	now demons gathe
3	Bobby Jindal, raised Hindu, uses story of Chri...	A dozen politically active pastors came here f...	0	bobby jindal, raised hindu, uses story of chri...	a dozen politically active pastors came here f...	bobby jindal raised hindu uses story of christ...	pc active came
4	SATAN 2: Russia unvelis an image of its terrif...	The RS-28 Sarmat missile, dubbed Satan 2, will...	1	satan 2: russia unvelis an image of its terrif...	the rs-28 sarmat missile, dubbed satan 2, will...	satan 2 russia unvelis an image of its terrify...	the rs28 missile satan 2
5	About Time! Christian Group Sues Amazon and SP...	All we can say on this one is it s about time ...	1	about time! christian group sues amazon and sp...	all we can say on this one is it s about time ...	about time christian group sues amazon and spl...	all we on this s about

In [17]:
```
print("First row of title without Punctuation:")
print(df.loc[0, 'title_wo_punct'])

print("\nFirst row of title without Stop Words:")
print(df.loc[0, 'title_wo_stop'])
```

```
First row of title without Punctuation:
law enforcement on high alert following threats against cops and whites on
911by blacklivesmatter and fyf911 terrorists video

First row of title without Stop Words:
law enforcement high alert following threats cops whites 911by blacklivesma
tter fyf911 terrorists video
```

Task 3.4 - Lemmatization

Lemmatization reduces words to their base or root form, considering the context of the word. It helps in normalizing the text and reducing inflectional forms.

The output can be shown in the new columns created below "title_lemmatized" and "text_lemmatized". As well as a sample row printed to show the difference between the previous step and the sentence that is lemmatized.

In [18]:
```
#Lemmatization
import nltk
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()
```

```
def lemmatize_words(text):
    return " ".join([lemmatizer.lemmatize(word) for word in text.split()])

df["title_lemmatized"] = df["title_wo_stop"].apply(lambda text: lemmatize_wo
df["text_lemmatized"] = df["text_wo_stop"].apply(lambda text: lemmatize_word
df.head()
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[18]:

	title	text	label	title_lower	text_lower	title_wo_punct	text_wo
0	LAW ENFORCEMENT ON HIGH ALERT Following Threat...	No comment is expected from Barack Obama Membe...	1	law enforcement on high alert following threat...	no comment is expected from barack obama membe...	law enforcement on high alert following threat...	no com expecte barack m
2	UNBELIEVABLE! OBAMA'S ATTORNEY GENERAL SAYS MO...	Now, most of the demonstrators gathered last ...	1	unbelievable! obama's attorney general says mo...	now, most of the demonstrators gathered last ...	unbelievable obama's attorney general says mos...	now demons gathe
3	Bobby Jindal, raised Hindu, uses story of Chri...	A dozen politically active pastors came here f...	0	bobby jindal, raised hindu, uses story of chri...	a dozen politically active pastors came here f...	bobby jindal raised hindu uses story of christ...	pc active came
4	SATAN 2: Russia unvelis an image of its terrif...	The RS-28 Sarmat missile, dubbed Satan 2, will...	1	satan 2: russia unvelis an image of its terrif...	the rs-28 sarmat missile, dubbed satan 2, will...	satan 2 russia unvelis an image of its terrify...	the rs28 missile satan 2
5	About Time! Christian Group Sues Amazon and SP...	All we can say on this one is it s about time ...	1	about time! christian group sues amazon and sp...	all we can say on this one is it s about time ...	about time christian group sues amazon and spl...	all we on this s about

In [19]:

```
print("First row of title without Stop Words:")
print(df.loc[0, 'title_wo_stop'])

print("\nFirst row of title Lemmatized:")
print(df.loc[0, 'title_lemmatized'])
```

```
First row of title without Stop Words:
law enforcement high alert following threats cops whites 911by blacklivesma
tter fyf911 terrorists video

First row of title Lemmatized:
law enforcement high alert following threat cop white 911by blacklivesmatte
r fyf911 terrorist video
```

In [20]:

```
# Dropping unnecessary columns that will not be used
df.drop(columns=['title', 'text', 'title_lower', 'text_lower', 'title_wo_pur
df.head()
```

Out[20]:

	label	title_lemmatized	text_lemmatized
0	1	law enforcement high alert following threat co...	comment expected barack obama member fyf911 fu...
2	1	unbelievable obama's attorney general say char...	demonstrator gathered last night exercising co...
3	0	bobby jindal raised hindu us story christian c...	dozen politically active pastor came private d...
4	1	satan 2 russia unvelis image terrifying new 's...	rs28 sarmat missile dubbed satan 2 replace ss1...
5	1	time christian group sue amazon splc designati...	say one time someone sued southern poverty law...

Task 4 – Text Representation

The text representation techniques that have been implemented are:

- TF-IDF
- POS Tagging
- NER Tagging
- Bag of Words

However, TF-IDF was chosen for its ability to emphasize significant terms crucial for distinguishing fake from real news, aligning well with classification tasks. While POS tagging, NER tagging, and Bag of Words offer valuable insights, they are computationally intensive, especially with larger datasets. Hence, they were demonstrated on a smaller subset (500 rows), showcasing its scalability and effectiveness for fake news classification.

Task 4.1 - TF-IDF

TF-IDF measures the importance of a term in a document relative to a collection of documents. It is calculated by considering the frequency of a term in a document, balanced by how rare the term is across all documents.

Importance: TF-IDF is effective in highlighting terms that are frequent in a document but relatively rare in the entire corpus. This helps in identifying important and distinctive terms.

In [21]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
docs = list(df["text_lemmatized"])
tfidf_vectorizer = TfidfVectorizer(use_idf=True, max_features = 20000)
tfidf_vectorizer_vectors = tfidf_vectorizer.fit_transform(docs)
docs = tfidf_vectorizer_vectors.toarray()
print(docs.shape)
```

```
(10000, 20000)
```

In [22]:
```
print(docs)
```

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

In [23]:
```
print("Displaying first 5 rows and middle columns (13:15):")
print(docs[:5, 13:15])
```

```
Displaying first 5 rows and middle columns (13:15):
[[0.03975548 0.        ]
 [0.         0.        ]
 [0.         0.        ]
 [0.         0.0578454 ]
 [0.         0.        ]]
```

Task 4.2 - POS Tagging

POS tagging assigns a grammatical category (such as noun, verb, adjective) to each word in a sentence. This helps in understanding the syntactic structure of a sentence.

Importance: It assists in understanding the role and relationships between words in a sentence, aiding in tasks like parsing, sentiment analysis, and information extraction.

In [24]:
```
# Decreasing the size of the dataframe to 500 instead of 10000
# to be able to run the POS code smoothly and in a shorter amount of time (1

df_short = df[:500]
```

In [25]:
```
import nltk
from nltk.tokenize import word_tokenize
from nltk import pos_tag
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

# convert text into word_tokens with their tags
def pos_tagging(text):
        word_tokens = word_tokenize(text)
        return pos_tag(word_tokens)

df_short["title_POS"] = df_short["title_lemmatized"].apply(pos_tagging)
df_short["text_POS"] = df_short["text_lemmatized"].apply(pos_tagging)
df_short.head()
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
<ipython-input-25-8825e208baa5>:12: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

<ipython-input-25-8825e208baa5>:13: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[25]:

	label	title_lemmatized	text_lemmatized	title_POS	text_POS
0	1	law enforcement high alert following threat co...	comment expected barack obama member fyf911 fu...	[(law, NN), (enforcement, NN), (high, JJ), (al...	[(comment, NN), (expected, VBN), (barack, JJ),...
2	1	unbelievable obama’s attorney general say char...	demonstrator gathered last night exercising co...	[(unbelievable, JJ), (obama, NN), (’, NN), (s,...	[(demonstrator, NN), (gathered, VBD), (last, J...
3	0	bobby jindal raised hindu us story christian c...	dozen politically active pastor came private d...	[(bobby, NN), (jindal, NN), (raised, VBD), (hi...	[(dozen, NN), (politically, RB), (active, JJ),...
4	1	satan 2 russia unvelis image terrifying new ‘s...	rs28 sarmat missile dubbed satan 2 replace ss1...	[(satan, JJ), (2, CD), (russia, NN), (unvelis,...	[(rs28, NN), (sarmat, NN), (missile, NN), (dub...
5	1	time christian group sue amazon splc designati...	say one time someone sued southern poverty law...	[(time, NN), (christian, JJ), (group, NN), (su...	[(say, VB), (one, CD), (time, NN), (someone, N...

In [26]:

```
print("First row of title Lemmatized:")
print(df_short.loc[0, 'title_lemmatized'])

print("\nFirst row of title with POS Tagging:")
print(df_short.loc[0, 'title_POS'])
```

```
First row of title Lemmatized:
law enforcement high alert following threat cop white 911by blacklivesmatte
r fyf911 terrorist video

First row of title with POS Tagging:
[('law', 'NN'), ('enforcement', 'NN'), ('high', 'JJ'), ('alert', 'NN'), ('f
ollowing', 'VBG'), ('threat', 'NN'), ('cop', 'JJ'), ('white', 'JJ'), ('911b
y', 'CD'), ('blacklivesmatter', 'NN'), ('fyf911', 'NN'), ('terrorist', 'N
N'), ('video', 'NN')]
```

Task 4.3 - NER

NER tagging identifies and classifies named entities (such as names of persons, organizations, locations) within a text.

Importance: It's crucial for extracting specific information and identifying entities, which can be valuable for tasks like information retrieval, question answering, and content analysis.

In [27]:
```
# Decreasing the size of the dataframe to 500 instead of 10000
# to be able to run the NER code smoothly and in a shorter amount of time (1

df_short = df[:500]
```

In [28]:
```
import spacy
from spacy import displacy

spacy.cli.download("en_core_web_sm")
#Command line interface download

NER = spacy.load("en_core_web_sm")

def spacy_large_ner(document):
  return {(ent.text.strip(), ent.label_) for ent in NER(document).ents}

#spaCy has the property ents on Doc objects. We will use it to extract named
df_short["title_NER"] = df_short["title_lemmatized"].apply(spacy_large_ner)
df_short["text_NER"] = df_short["text_lemmatized"].apply(spacy_large_ner)
df_short.head()
```

```
✔ Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
<ipython-input-28-dfedeb2d39f1>:13: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

<ipython-input-28-dfedeb2d39f1>:14: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[28]:

	label	title_lemmatized	text_lemmatized	title_NER	text_NER
0	1	law enforcement high alert following threat co...	comment expected barack obama member fyf911 fu...	{}	{(one two, CARDINAL), (239 minute, TIME), (ton...
2	1	unbelievable obama’s attorney general say char...	demonstrator gathered last night exercising co...	{(obama, GPE), (charlotte rioter, PERSON), (no...	{(loretta lynch, PERSON), (eric holder, PERSON...
3	0	bobby jindal raised hindu us story christian c...	dozen politically active pastor came private d...	{(christian, NORP), (bobby, PERSON), (2016, DA...	{(missouri, GPE), (42, CARDINAL), (white house...
4	1	satan 2 russia unvelis image terrifying new ‘s...	rs28 sarmat missile dubbed satan 2 replace ss1...	{(russia, GPE), (satan 2, DATE)}	{(7km, QUANTITY), (satan 2, DATE), (1945, CARD...
5	1	time christian group sue amazon splc designati...	say one time someone sued southern poverty law...	{(christian, NORP), (amazon splc designation h...	{(christian, NORP), (today, DATE), (james kenn...

In [29]:
```
print("Second row of title Lemmatized:")
print(df_short.loc[2, 'title_lemmatized'])

print("\nSecond row of title with NER Tagging:")
print(df_short.loc[2, 'title_NER'])
```

```
Second row of title Lemmatized:
unbelievable obama’s attorney general say charlotte rioter “peaceful” prote
sters…in home state north carolina video

Second row of title with NER Tagging:
{('obama', 'GPE'), ('charlotte rioter', 'PERSON'), ('north carolina', 'GP
E')}
```

Task 4.4 - Bag of Words

Bag of Words represents text data by counting the frequency of words in a document without considering the order of words.

Importance: It's a simple yet effective technique to convert text into numerical features, forming the basis for many text classification algorithms.

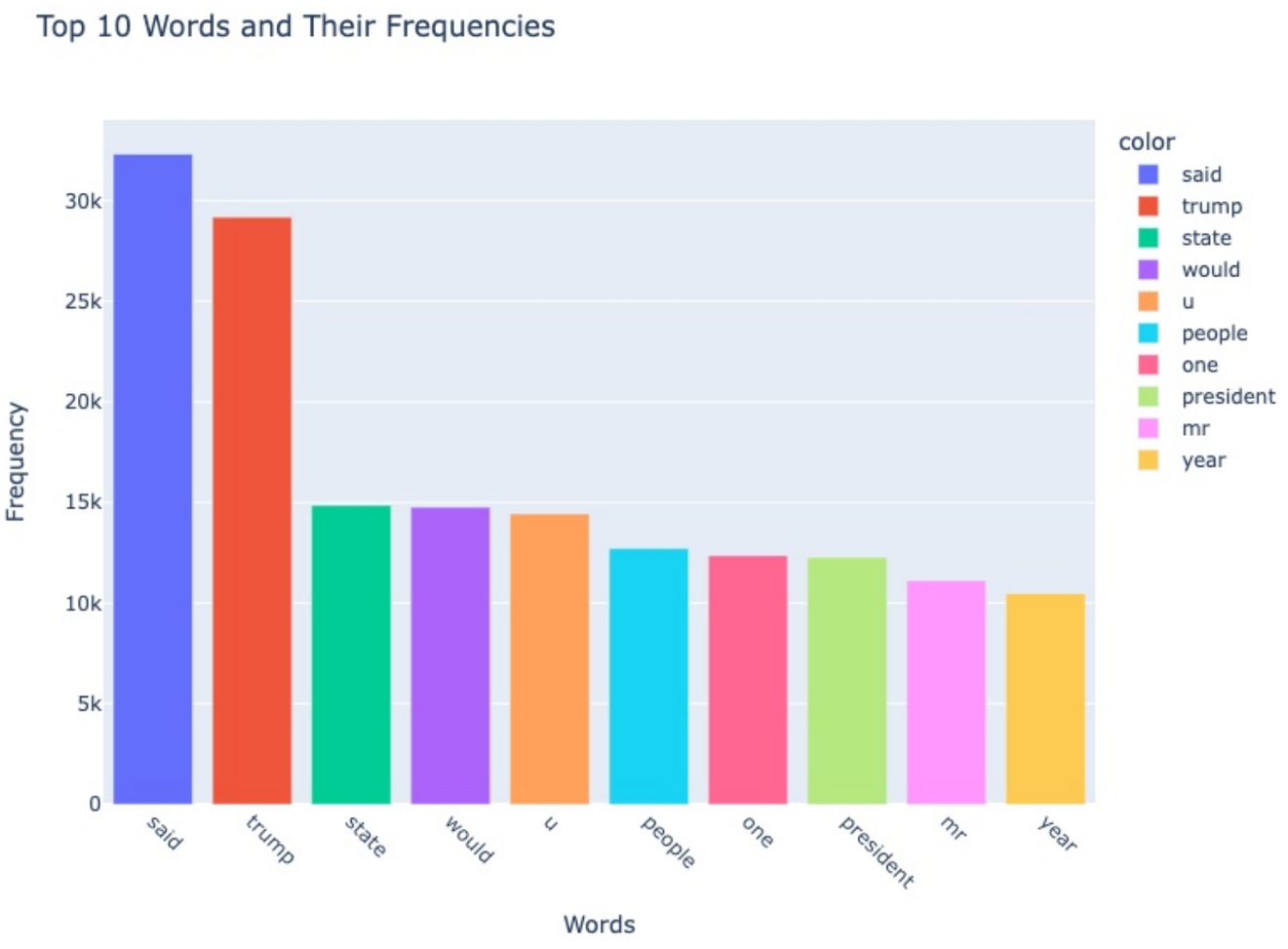
In [30]:
```
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from collections import Counter

# Tokenization and lowercasing
tokens = [word_tokenize(doc.lower()) for doc in df["text_lemmatized"]]

# Remove stopwords and punctuation
stop_words = set(stopwords.words('english'))
filtered_tokens = [[word for word in token if word.isalnum() and word not in

# Create a vocabulary
```

```
vocabulary = set(word for token in filtered_tokens for word in token)

# Initialize a BoW dictionary with word counts
bow = {word: 0 for word in vocabulary}

# Count word occurrences
for token in filtered_tokens:
    for word in token:
        bow[word] += 1

sorted_bow = dict(sorted(bow.items(), key=lambda item: item[1], reverse=True
print(dict(list(sorted_bow.items())[:30]))
```

```
{'said': 32309, 'trump': 29188, 'state': 14856, 'would': 14761, 'u': 14429,
'people': 12713, 'one': 12358, 'president': 12275, 'mr': 11120, 'year': 104
80, 'new': 9809, 'clinton': 9725, 'time': 9179, 'also': 9151, 'republican':
8178, 'like': 8001, 'american': 7367, 'government': 7293, 'could': 6932, 's
ay': 6844, 'obama': 6831, 'country': 6713, 'house': 6253, 'even': 6153, 'do
nald': 6071, 'election': 6049, 'campaign': 6046, 'two': 5979, 'many': 5959,
'united': 5905}
```

In [31]:

```
import plotly.express as px

# Take the top 10 words and their frequencies
top_words = list(sorted_bow.keys())[:10]
word_freq = list(sorted_bow.values())[:10]

# Create a Plotly Express bar chart
fig = px.bar(x=top_words, y=word_freq, labels={'x': 'Words', 'y': 'Frequency
             title='Top 10 Words and Their Frequencies', color=top_words,
             height=600, width=800)

# Rotate x-axis labels for better readability
fig.update_layout(xaxis={'tickangle': 45})

# Show the plot
fig.show()
```



Task 5 – Text Classification / Prediction

The text classifiers that have been implemented are:

- Guassian Naive Bayes
- Multinomial Naive Bayes
- Logistic Regression
- Support Vector Machines

Using these models allows to leverage ML methods (Gaussian NB, Multinomial NB, Logistic Regression, SVM) and explore their performance on the fake news classification.

Gaussian NB and Multinomial NB are particularly suited for text data, while Logistic Regression and SVM offer different strategies for classification.

This approach allows us to compare the effectiveness and efficiency of both traditional and more complex models in handling the fake news classification problem.

The selection enables a diverse assessment of various classification techniques, providing insights into which models perform best for your specific text-based classification task.

However, the dataset has to be divided and prepared to train the classifer.

Task 5.1 - Dividing the Dataset and Preparing Training the Classifier

In [32]:
```
X = docs
y = df['label']
print(X.shape, y.shape)
```

```
(10000, 20000) (10000,)
```

In [33]:
```
from sklearn.model_selection import train_test_split
#Train-Test Split
SEED=123
X_train,X_test,y_train,y_test=train_test_split(X, y, test_size=0.2, random_s
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

```
(8000, 20000) (8000,)
(2000, 20000) (2000,)
```

Task 5.2 - Gaussian Naive Bayes

- **Applicability**: Gaussian Naive Bayes is suitable for classification tasks with continuous features (like TF-IDF) assuming a Gaussian distribution.
- **Efficiency**: It's computationally efficient and works well with smaller datasets.

In [34]:
```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

gnb = GaussianNB()
%time gnb.fit(X_train, y_train)

y_pred_train_gnb = gnb.predict(X_train)
y_pred_test_gnb = gnb.predict(X_test)
print("\nTraining Accuracy score:",accuracy_score(y_train, y_pred_train_gnb)
print("Testing Accuracy score:",accuracy_score(y_test, y_pred_test_gnb))
```

```
CPU times: user 1.56 s, sys: 1.8 s, total: 3.37 s
Wall time: 3.41 s

Training Accuracy score: 0.958125
Testing Accuracy score: 0.7895
```

In [35]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred_test_gnb, target_names=['fake',
```

```
              precision    recall  f1-score   support

        fake       0.76      0.82      0.79       962
        real       0.82      0.76      0.79      1038

    accuracy                           0.79      2000
   macro avg       0.79      0.79      0.79      2000
weighted avg       0.79      0.79      0.79      2000
```

Task 5.3 - Multinomial Naive Bayes

- **Suitability for Text Data**: Ideal for classification tasks involving text data represented using TF-IDF or Bag of Words.
- **Frequentist Approach**: It assumes a multinomial distribution of features and works well with discrete counts.

In [36]:
```
from sklearn.naive_bayes import MultinomialNB

mnb = MultinomialNB()
%time mnb.fit(X_train, y_train)

y_pred_train_mnb = mnb.predict(X_train)
y_pred_test_mnb = mnb.predict(X_test)
print("\nTraining Accuracy score:",accuracy_score(y_train, y_pred_train_mnb)
print("Testing Accuracy score:",accuracy_score(y_test, y_pred_test_mnb))
```

```
CPU times: user 776 ms, sys: 115 ms, total: 891 ms
Wall time: 961 ms

Training Accuracy score: 0.8945
Testing Accuracy score: 0.8655
```

In [37]:
```
print(classification_report(y_test, y_pred_test_mnb, target_names=['fake',
```

```
              precision    recall  f1-score   support

        fake       0.87      0.84      0.86       962
        real       0.86      0.89      0.87      1038

    accuracy                           0.87      2000
   macro avg       0.87      0.86      0.87      2000
weighted avg       0.87      0.87      0.87      2000
```

Task 5.4 - Logistic Regression Classifier

- **Interpretability and Performance**: Logistic Regression is known for its interpretability and can handle linearly separable data well. It serves as a good baseline model for binary classification tasks.
- **Scalability**: It's relatively efficient with larger datasets and can handle large feature spaces.

In [38]:
```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(random_state=SEED)
%time lr.fit(X_train, y_train)
```

```
y_pred_train_lr = lr.predict(X_train)
y_pred_test_lr = lr.predict(X_test)
print("\nTraining Accuracy score:",accuracy_score(y_train, y_pred_train_lr))
print("Testing Accuracy score:",accuracy_score(y_test, y_pred_test_lr))
```

```
CPU times: user 8.22 s, sys: 1.55 s, total: 9.77 s
Wall time: 11.3 s

Training Accuracy score: 0.95025
Testing Accuracy score: 0.9195
```

In [39]:
```
print(classification_report(y_test, y_pred_test_lr, target_names=['fake', '
```

```
              precision    recall  f1-score   support

        fake       0.93      0.90      0.91       962
        real       0.91      0.94      0.92      1038

    accuracy                           0.92      2000
   macro avg       0.92      0.92      0.92      2000
weighted avg       0.92      0.92      0.92      2000
```

Task 5.5 - Support Vector Machines

- **Robustness and Flexibility**: SVM is effective in high-dimensional spaces and works well with both linearly and non-linearly separable data.
- **Kernel Trick**: SVM can use different kernel functions to transform data into higher dimensions, potentially capturing complex relationships.

In [40]:
```
from sklearn.svm import LinearSVC

svc =  LinearSVC(class_weight='balanced')
%time svc.fit(X_train, y_train)

y_pred_train_svc = svc.predict(X_train)
y_pred_test_svc = svc.predict(X_test)
print("\nTraining Accuracy score:",accuracy_score(y_train, y_pred_train_svc))
print("Testing Accuracy score:",accuracy_score(y_test, y_pred_test_svc))
```

```
CPU times: user 1.33 s, sys: 36.3 ms, total: 1.37 s
Wall time: 1.55 s

Training Accuracy score: 0.999
Testing Accuracy score: 0.943
```

In [41]:
```
print(classification_report(y_test, y_pred_test_svc, target_names=['fake',
```

```
              precision    recall  f1-score   support

        fake       0.94      0.94      0.94       962
        real       0.94      0.95      0.95      1038

    accuracy                           0.94      2000
   macro avg       0.94      0.94      0.94      2000
weighted avg       0.94      0.94      0.94      2000
```

Task 6 – Evaluation, Inferences, Recommendation and Reflection

The Classifiers were evaluated using the following methods:

- By testing the classifiers manually (using a sample dataset)
- Confusion Matrix
- ROC Curves
- Accuracy
- Precision
- Recall
- F1 Score

Task 6.1 - Test the Classifiers

In [42]:
```
def preprocessing_steps(text):
  text = text_lowercase(text)
  text = remove_punctuation(text)
  text = remove_stopwords(text)
  text = lemmatize_words(text)
  return text
```

In [43]:
```
import pandas as pd
import numpy as np
df_test = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/NLP_Project/WE

df_test = df_test.tail(10)
df_test.dropna(inplace=True)
df_test.drop(columns=['Unnamed: 0', 'title'], inplace=True)
df_test.head(10)
```

Out[43]:

	text	label
72124	PARIS — In the age of Donald J. Trump, “Bre...	0
72125		1
72126	The flag at Desert Hot Springs' Condor Gun Sho...	0
72127	An email released by WikiLeaks on Sunday appea...	1
72128	Judge Jeanine lets it rip! She s concerned wit...	1
72129	WASHINGTON (Reuters) - Hackers believed to be ...	0
72130	You know, because in fantasyland Republicans n...	1
72131	Migrants Refuse To Leave Train At Refugee Camp...	0
72132	MEXICO CITY (Reuters) - Donald Trump’s combati...	0
72133	Goldman Sachs Endorses Hillary Clinton For Pre...	1

In [44]:
```
df_test['text'] = df_test['text'].apply(preprocessing_steps)

tfidf_vectors = tfidf_vectorizer.transform(df_test['text'])

tfidf_vectors_dense = tfidf_vectors.toarray()

df_test['gnb_predictions'] = gnb.predict(tfidf_vectors_dense)
df_test['mnb_predictions'] = mnb.predict(tfidf_vectors_dense)
df_test['lr_predictions'] = lr.predict(tfidf_vectors_dense)
df_test['svc_predictions'] = svc.predict(tfidf_vectors_dense)
```

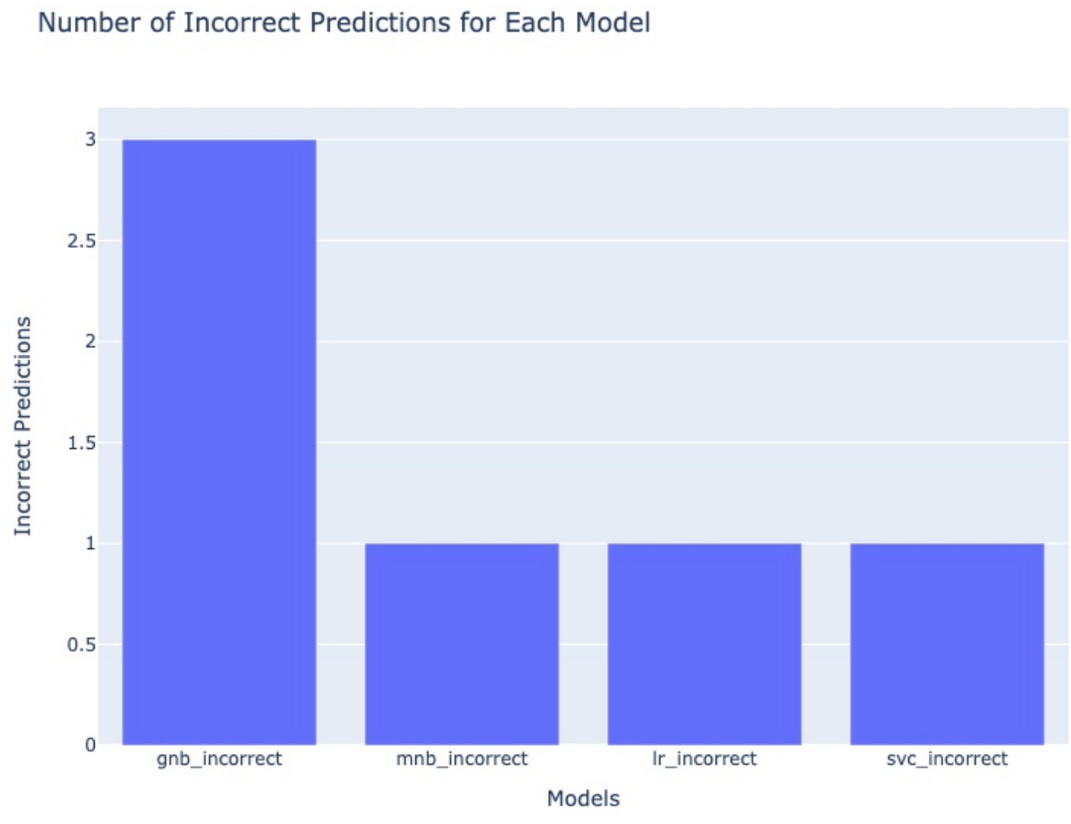
```
df_test.head(10)
```

Out[44]:

	text	label	gnb_predictions	mnb_predictions	lr_predictions	svc_predictions
72124	paris — age donald j trump “brexit” resurgent ...	0	0	0	0	0
72125		1	0	1	1	1
72126	flag desert hot spring condor gun shop flew ha...	0	1	1	1	1
72127	email released wikileaks sunday appears show f...	1	1	1	1	1
72128	judge jeanine let rip concerned silencing cons...	1	1	1	1	1
72129	washington reuters hacker believed working rus...	0	1	0	0	0
72130	know fantasyland republican never questioned c...	1	1	1	1	1
72131	migrant refuse leave train refugee camp hungar...	0	0	0	0	0
72132	mexico city reuters donald trump’s combative s...	0	0	0	0	0
72133	goldman sachs endorses hillary clinton preside...	1	1	1	1	1

In [45]:

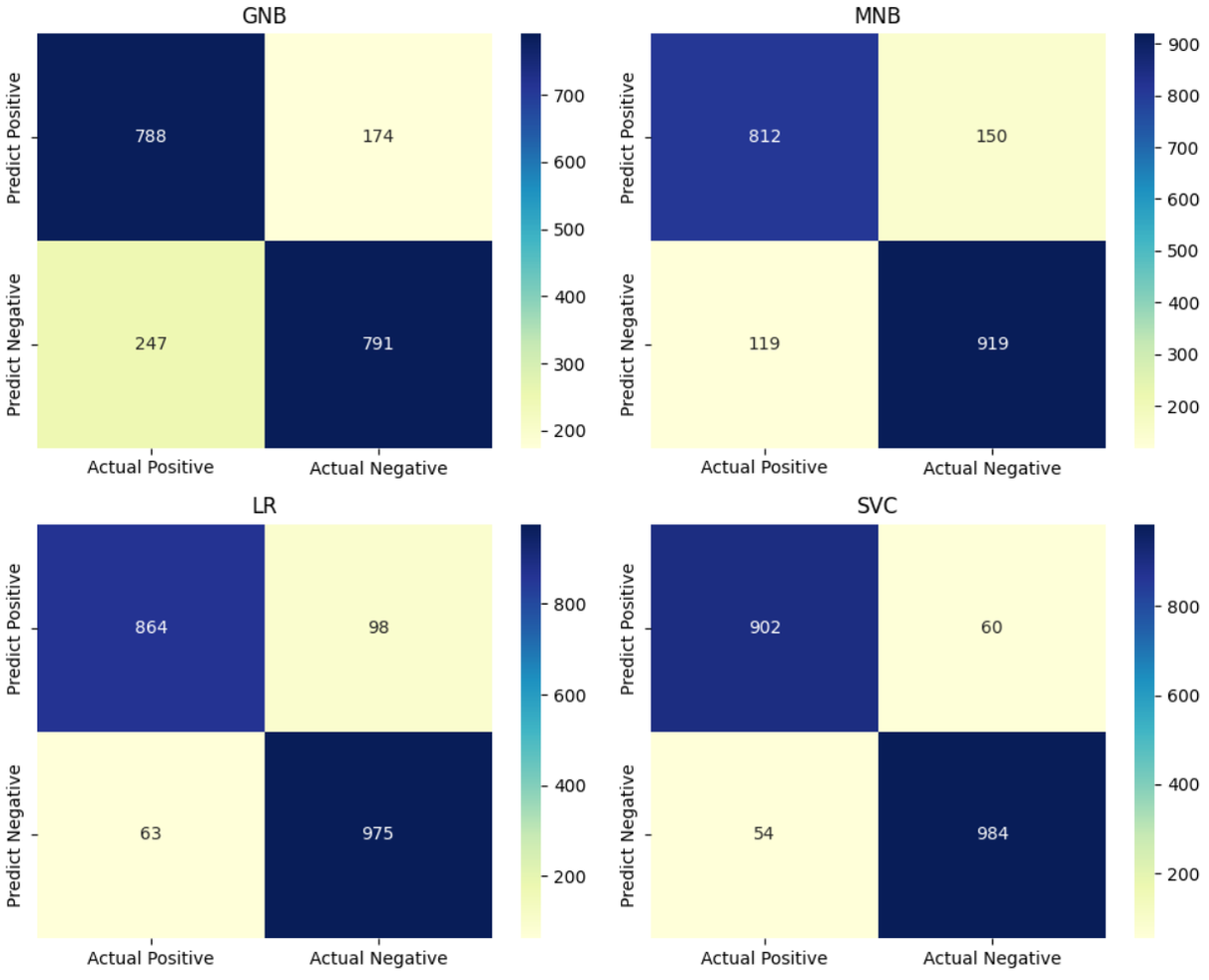
```
import plotly.express as px

# Calculate the number of incorrect predictions for each model
df_test['gnb_incorrect'] = df_test['gnb_predictions'] != df_test['label']
df_test['mnb_incorrect'] = df_test['mnb_predictions'] != df_test['label']
df_test['lr_incorrect'] = df_test['lr_predictions'] != df_test['label']
df_test['svc_incorrect'] = df_test['svc_predictions'] != df_test['label']

# Count the number of incorrect predictions for each model
incorrect_counts = df_test[['gnb_incorrect', 'mnb_incorrect', 'lr_incorrect

# Create a DataFrame for visualization
accuracy_df = incorrect_counts.reset_index()
accuracy_df.columns = ['Model', 'Incorrect Predictions']

# Create a count plot using Plotly Express
fig = px.bar(accuracy_df, x='Model', y='Incorrect Predictions',
             title='Number of Incorrect Predictions for Each Model',
             width=800, height=600)

fig.update_layout(xaxis_title='Models', yaxis_title='Incorrect Predictions')
fig.show()
```



Task 6.2 - Evaluation - Confusion Matrix

In [46]:

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Create a figure and subplots using Matplotlib
# GNB
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 8))

cm_gnb = confusion_matrix(y_test, y_pred_test_gnb)

cm_matrix_gnb = pd.DataFrame(data=cm_gnb, columns=['Actual Positive', 'Actua
                             index=['Predict Positive', 'Predict Negative'])
sns.heatmap(cm_matrix_gnb, ax=axes[0, 0], annot=True, fmt='d', cmap='YlGnBu
axes[0, 0].set_title('GNB')

# MNB
cm_mnb = confusion_matrix(y_test, y_pred_test_mnb)

cm_matrix_mnb = pd.DataFrame(data=cm_mnb, columns=['Actual Positive', 'Actua
                             index=['Predict Positive', 'Predict Negative'])
sns.heatmap(cm_matrix_mnb, ax=axes[0, 1], annot=True, fmt='d', cmap='YlGnBu
axes[0, 1].set_title('MNB')

# LR
cm_lr = confusion_matrix(y_test, y_pred_test_lr)

cm_matrix_lr = pd.DataFrame(data=cm_lr, columns=['Actual Positive', 'Actual
                             index=['Predict Positive', 'Predict Negative'])
sns.heatmap(cm_matrix_lr, ax=axes[1, 0], annot=True, fmt='d', cmap='YlGnBu'
axes[1, 0].set_title('LR')

# SVC
cm_svc = confusion_matrix(y_test, y_pred_test_svc)

cm_matrix_svc = pd.DataFrame(data=cm_svc, columns=['Actual Positive', 'Actua
                             index=['Predict Positive', 'Predict Negative'])
sns.heatmap(cm_matrix_svc, ax=axes[1, 1], annot=True, fmt='d', cmap='YlGnBu
axes[1, 1].set_title('SVC')

plt.tight_layout()
plt.show()
```

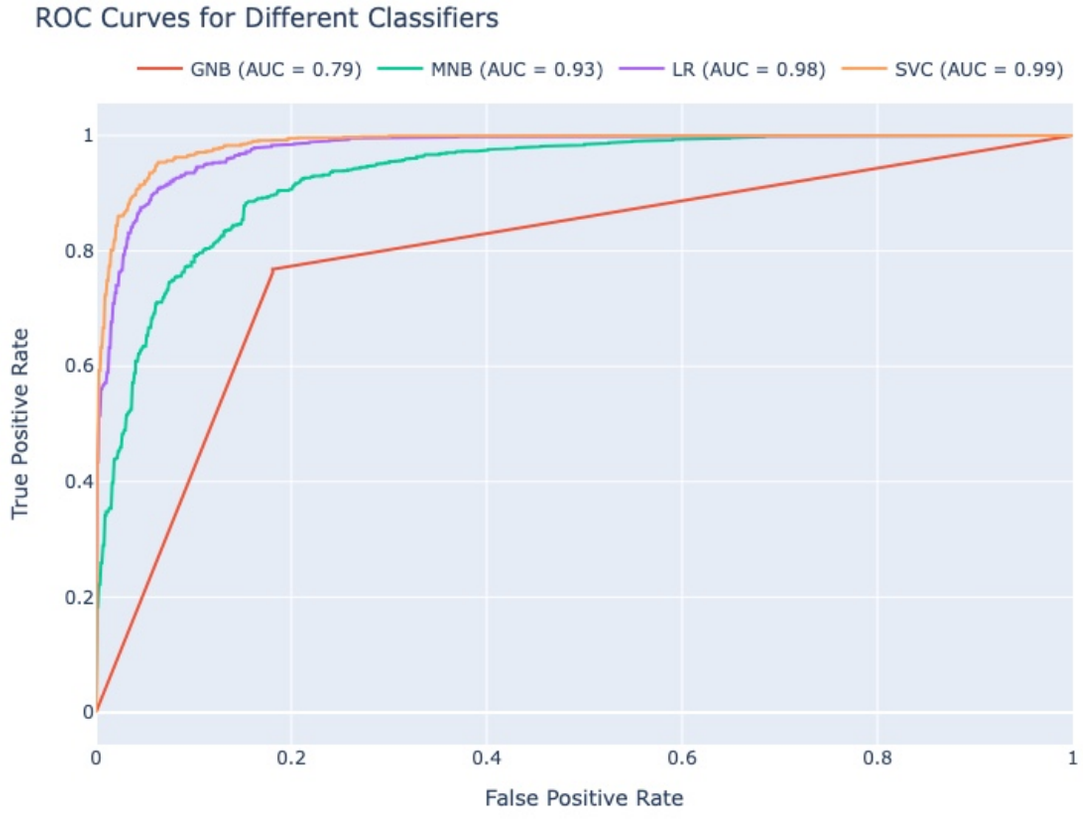


Task 6.3 - Evaluation - ROC Curves

In [47]:

```
import plotly.express as px
import sklearn.metrics as metrics

# GNB
probs_gnb = gnb.predict_proba(X_test)
preds_gnb = probs_gnb[:,1]
fpr_gnb, tpr_gnb, threshold_gnb = metrics.roc_curve(y_test, preds_gnb)
roc_auc_gnb = metrics.auc(fpr_gnb, tpr_gnb)

# MNB
probs_mnb = mnb.predict_proba(X_test)
preds_mnb = probs_mnb[:,1]
fpr_mnb, tpr_mnb, threshold_mnb = metrics.roc_curve(y_test, preds_mnb)
roc_auc_mnb = metrics.auc(fpr_mnb, tpr_mnb)

# LR
probs_lr = lr.predict_proba(X_test)
preds_lr = probs_lr[:,1]
fpr_lr, tpr_lr, threshold_lr = metrics.roc_curve(y_test, preds_lr)
roc_auc_lr = metrics.auc(fpr_lr, tpr_lr)

# SVC
probs_svc = svc._predict_proba_lr(X_test)
preds_svc = probs_svc[:,1]
fpr_svc, tpr_svc, threshold_svc = metrics.roc_curve(y_test, preds_svc)
roc_auc_svc = metrics.auc(fpr_svc, tpr_svc)

fig = px.line(title='ROC Curves for Different Classifiers', width=800, heigh
```

```
fig.add_scatter(x=fpr_gnb, y=tpr_gnb, mode='lines', name=f'GNB (AUC = {roc_a
fig.add_scatter(x=fpr_mnb, y=tpr_mnb, mode='lines', name=f'MNB (AUC = {roc_a
fig.add_scatter(x=fpr_lr, y=tpr_lr, mode='lines', name=f'LR (AUC = {roc_auc_
fig.add_scatter(x=fpr_svc, y=tpr_svc, mode='lines', name=f'SVC (AUC = {roc_a

fig.update_layout(
    xaxis_title='False Positive Rate',
    yaxis_title='True Positive Rate',
    legend=dict(orientation='h', yanchor='bottom', y=1.02, xanchor='right',
)

fig.show()
```



Task 6.4 - Evaluation - Accuracy, Precision, Recall, F1 Score

In [48]:
```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f

# GNB
accuracy_gnb = accuracy_score(y_test, y_pred_test_gnb)
precision_gnb = precision_score(y_test, y_pred_test_gnb)
recall_gnb = recall_score(y_test, y_pred_test_gnb)
f1_gnb = f1_score(y_test, y_pred_test_gnb)
```

In [49]:
```
# MNB
accuracy_mnb = accuracy_score(y_test, y_pred_test_mnb)
precision_mnb = precision_score(y_test, y_pred_test_mnb)
recall_mnb = recall_score(y_test, y_pred_test_mnb)
f1_mnb = f1_score(y_test, y_pred_test_mnb)
```

In [50]:
```
# LR
accuracy_lr = accuracy_score(y_test, y_pred_test_lr)
precision_lr = precision_score(y_test, y_pred_test_lr)
recall_lr = recall_score(y_test, y_pred_test_lr)
f1_lr = f1_score(y_test, y_pred_test_lr)
```

In [51]:
```
# SVC
accuracy_svc = accuracy_score(y_test, y_pred_test_svc)
precision_svc = precision_score(y_test, y_pred_test_svc)
recall_svc = recall_score(y_test, y_pred_test_svc)
f1_svc = f1_score(y_test, y_pred_test_svc)
```

In [52]:
```
import plotly.graph_objects as go

# Define data for the table
classifiers = ['GNB', 'MNB', 'LR', 'SVC']
accuracy = [accuracy_gnb, accuracy_mnb, accuracy_lr, accuracy_svc]
precision = [precision_gnb, precision_mnb, precision_lr, precision_svc]
recall = [recall_gnb, recall_mnb, recall_lr, recall_svc]
f1 = [f1_gnb, f1_mnb, f1_lr, f1_svc]

# Round the metrics to three decimal places
accuracy = [round(val, 3) for val in accuracy]
precision = [round(val, 3) for val in precision]
recall = [round(val, 3) for val in recall]
f1 = [round(val, 3) for val in f1]

# Create a table using plotly.graph_objects
fig = go.Figure(data=[go.Table(
    header=dict(values=['Classifier', 'Accuracy', 'Precision', 'Recall', 'F1
    cells=dict(values=[classifiers, accuracy, precision, recall, f1]))
])

# Update layout
fig.update_layout(title='Classifier Comparison Metrics', height=400, width=8

# Show the table
fig.show()
```

Classifier Comparison Metrics

Classifier	Accuracy	Precision	Recall	F1 Score
GNB	0.79	0.82	0.762	0.79
MNB	0.866	0.86	0.885	0.872
LR	0.92	0.909	0.939	0.924
SVC	0.943	0.943	0.948	0.945

In [53]:

```
import plotly.graph_objects as go
from plotly.subplots import make_subplots

# Assuming you have calculated metrics for GNB, MNB, LR, and SVC
# Replace the values with your actual calculated metrics

# Define data for the bar chart
classifiers = ['GNB', 'MNB', 'LR', 'SVC']
accuracy = [accuracy_gnb, accuracy_mnb, accuracy_lr, accuracy_svc]
precision = [precision_gnb, precision_mnb, precision_lr, precision_svc]
recall = [recall_gnb, recall_mnb, recall_lr, recall_svc]
f1 = [f1_gnb, f1_mnb, f1_lr, f1_svc]

# Create subplots using make_subplots
fig = make_subplots(rows=2, cols=2, subplot_titles=('Accuracy', 'Precision',

# Add traces for each metric in the corresponding subplot
fig.add_trace(go.Bar(x=classifiers, y=accuracy, name='Accuracy'), row=1, co
fig.add_trace(go.Bar(x=classifiers, y=precision, name='Precision'), row=1, c
fig.add_trace(go.Bar(x=classifiers, y=recall, name='Recall'), row=2, col=1)
fig.add_trace(go.Bar(x=classifiers, y=f1, name='F1 Score'), row=2, col=2)

# Update layout
fig.update_layout(title='Classifier Comparison Metrics', height=600, width=8

# Show the interactive plot
fig.show()
```

Task 6.5 - Inferences

- **Impact of False Predictions on Fake News Classifier**:
  - *False Positives Impact*:
    - A false positive prediction (classifying genuine news as fake) might discredit credible sources, impacting their reputation and credibility.
    - It could also contribute to skepticism among readers, leading to a loss of trust in authentic news outlets.
  - *False Negatives Impact*:
    - A false negative prediction (misclassifying fake news as genuine) risks the spread of misinformation.
    - Such instances might perpetuate misleading narratives, potentially influencing public opinion or causing harm if the information is harmful or inaccurate.
- **SVC and LR Outperformance**:
  - Support Vector Classifier (SVC) and Logistic Regression (LR) emerged as top performers, exhibiting remarkable accuracy, precision, recall, and F1 scores.

    - SVC notably excelled with the highest accuracy (94.3%) and closely followed LR with a commendable accuracy of (92%).
- **SVC and LR and False Predictions**:
    - *SVC and Minimizing False Positives*:
        - SVC exhibited a lower number of false positives compared to LR and other models.
        - The lower false positive rate indicates SVC's robustness in discerning genuine news, reducing the risk of discrediting reliable sources.
    - *LR and Lower False Negatives*:
        - LR demonstrated a relatively lower false negative rate, indicating its capacity to identify a higher proportion of fake news accurately.

Task 6.6 - Reflection

- The choice of TF-IDF representation proved effective for most classifiers, with SVC showcasing superior performance due to its capacity to handle high-dimensional spaces efficiently.
- The project underscores the critical balance between minimizing false positives to protect credibility and reducing false negatives to prevent the propagation of misinformation.
- It highlights the importance of continual refinement in machine learning models to address the complexity of fake news classification.

Task 6.7 - Recommendations

- **Addressing False Positives**:
    - Strategies to mitigate false positives should be explored, possibly through feature engineering or model optimization, to enhance the reliability of the classifier in discerning genuine news.
- **False Negatives Mitigation**:
    - Efforts to reduce false negatives, especially by fine-tuning model parameters or leveraging more advanced NLP techniques, could minimize the spread of misinformation.
- **Ensemble Approach**:
    - Considering an ensemble of models or hybrid techniques might offer a balanced trade-off between minimizing false positives and false negatives, enhancing the overall efficacy of the fake news classifier.

Conclusion

In conclusion, the pursuit and evaluation of machine learning models for fake news classification mark a significant step toward countering misinformation. Through percise NLP techniques and diverse classification methods, this project shows different models' effectiveness in distinguishing genuine news from false information. The comprehensive evaluation highlights the delicate balance between minimizing false positives and false

negatives—crucial facets in preserving trustworthiness and curtailing misinformation's propagation. As the digital landscape continues evolving, this project underscores the iterative nature of enhancing classifiers, paving the way for more robust and reliable models in combating fake news.

Log File

- Mentor: Sini Raj Pulari
- Project Name: Fake News Classifier
- Student Name: Maram Aljasim
- Student ID: 202306803
- NLP (IT9002) Project Log File

Date	Week	Task Name	Work done during the week
11/5/2023	1	Task 1: Problem Statement Formulation and Definition	- Researched and identified a suitable business problem.
			- Studied existing works and scope of the project topic.
			- Formulated a problem statement, motivation, and objectives.
			- Defined the expected result.
11/12/2023	2	Task 2: Selection of an Appropriate Data Set (Data Collection)	- Selected a suitable dataset.
			- Justified the choice of dataset.
			- Visualized the initial data to understand its distribution and summary.
			- Visualized based on labels.
			- Cleaned the dataset.
11/19/2023	3	Task 3: Text Preprocessing	- Applied four text preprocessing techniques.
			- Shown the appropriate output for each step.
11/26/2023	4	Task 4: Text Representation	- Decided on the type of text representation techniques to use for the business problem.
			- Applied four text representation techniques.
12/3/2023	5	Task 5: Text Classification / Prediction	- Selected the most appropriate Text Classification / Prediction methods based on the business problem selected.
			- Applied four models on the cleaned and readied data.
12/10/2023	6	Task 6: Evaluation, Inferences, Recommendation and Reflection	- Evaluated and compared how well the four models have performed.
			- Used six metrics to evaluate the results.

Date	Week	Task Name	Work done during the week
			- Made inferences and reflections on learning and improvements that could have been done on the project.

GitHub

https://github.com/MaramAljasim/NLP_FakeNewsClassifier

References

- Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with python. O'reilly.
- Raj, S. (2023). IT9002 - Natural Language Processing. https://moodle.polytechnic.bh/moodle/course/view.php?id=1452
- Shahane, S. (2023, October 8). Fake news classification. Kaggle. https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification/data
- Thanaki, J. (2017). Python Natural Language Processing: Explore NLP with machine learning and Deep Learning Techniques. Packt Publishing.

Appendix

Saving the ipynb file in different formats

Method 1: !jupyter nbconvert --to html "mount/MyDrive/Colab Notebooks/NLP_Project/FakeNews_03.ipynb"

Method 2: ! sudo apt-get update ! sudo apt-get install texlive-xetex texlive-fonts-recommended texlive-plain-generic

from google.colab import drive drive.mount('./mount')

!jupyter nbconvert --to pdf "mount/MyDrive/Colab Notebooks/NLP_Project/FakeNews_03.ipynb"

In [54]:
```
!jupyter nbconvert --to html "/content/drive/MyDrive/Colab Notebooks/NLP_Pr
```

```
[NbConvertApp] Converting notebook /content/drive/MyDrive/Colab Notebooks/N
LP_Project/IT9002_202306803.ipynb to html
[NbConvertApp] Writing 1518495 bytes to /content/drive/MyDrive/Colab Notebo
oks/NLP_Project/IT9002_202306803.html
```