**FAKE NEWS DETECTION USING NLP**

**PHASE 3: DEVELOPMENT PART1**

**Batch members:**

Arthi.C(2021504505)

Madhumithaa.V(2021504021)

Sudheksha.R(2021504047)

**OBJECTIVE:**

Start building the Fake News Detection Model by loading and pre-processing the dataset**.**

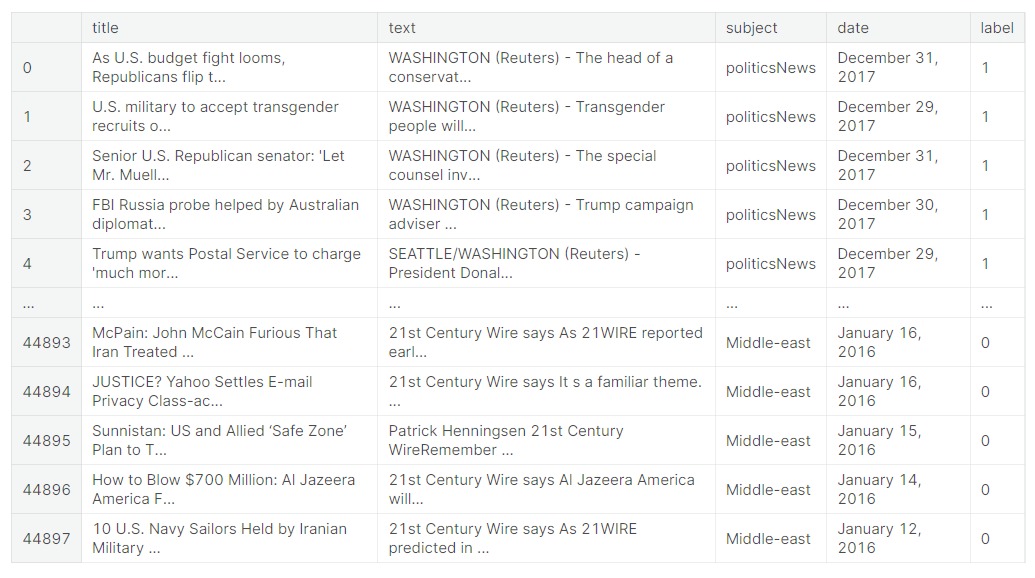
**INTRODUCTION:**

In an age defined by a deluge of information, the battle against misinformation has never been more critical. Introducing our groundbreaking Fake News Detector, fortified by the prowess of Natural Language Processing (NLP). This cutting-edge tool represents a paradigm shift in discerning veracity from falsehood within news narratives.

Employing sophisticated linguistic analysis, our system meticulously dissects articles, scrutinizing linguistic patterns, semantic subtleties, and contextual cues. Through this comprehensive evaluation, it renders a precise verdict on the authenticity of the content. The application of NLP allows for an unparalleled depth of understanding, enabling our detector to discern even the most artfully cloaked misinformation.

With a commitment to upholding the integrity of information, our Fake News Detector stands as a beacon of truth in a landscape fraught with deception. Join us in the fight against disinformation and the safeguarding of informed discourse. Experience the future of information integrity with our state-of-the-art NLP-driven Fake News Detector. Together, let us unmask the truth.

**DATASET**

****

**PROCEDURE:**

**1.Import Libraries:**

Start by importing the necessary libraries. The very basic data science libraries are sklearn, pandas, NumPy ,nltk e.t.c and some specific libraries such as transformers.

**2.Load the Dataset:**

Load your dataset into a Pandas DataFrame. You can typically find

Fake news datasets in CSV format, but you can adapt this code to other

formats as needed. The dataset contains 25117 rows and 5 columns.

This dataset has 5 columns,

* Id:represent the unique id of each news.
* title: this represents the title of the news.
* author: this represents the name of the author who has written the news.
* text: this column has the news itself.
* label: this is a binary column representing if the news is fake (1) or real (0).

**3. Exploratory Data Analysis (EDA):**

Perform EDA to understand the data better. This includes

checking for missing values, exploring the data's statistics, and

visualizing it to identify patterns. This includes

* Check for missing values
* Explore statistics
* Visualize the data (e.g., histograms, scatter plots, etc.)

**4.Filtering the data for modeling**

Removing the stop words and segments text data and image data separately. Stop words 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 sentence they do not add much meaning to the context of the sentence so even after removing the stopwords we can understand the context.

**5. Split the Data:**

Split the dataset into training and testing sets. This helps you evaluate

model's performance later.

Importance of Data Splitting:Data splitting is essential to assess how well your model generalizes to unseen data. It helps prevent overfitting (when a model learns the training data too well but performs poorly on new data) and ensures a reliable evaluation.

Types of Data Splits:

a. Training Set: This is the largest portion of the dataset. It's used to train the model to learn patterns, relationships, and features in the data.

b. Test Set: This is kept completely separate and untouched during training and hyperparameter tuning. It's used only after the model is fully trained to evaluate its performance.

**PROGRAM:**

**1.Import libraries**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

pd.set\_option("display.max\_columns",50)

pd.set\_option("display.max\_rows",50)

import nltk

from nltk.stem.porter import PorterStemmer

from wordcloud import WordCloud

from wordcloud import STOPWORDS

from PIL import Image

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import accuracy\_score

import warnings

warnings.filterwarnings('ignore')

from nltk.corpus import stopwords

stopwords=stopwords.words('english') + ['of','the','to','be','that','with','on','for','by','at','one']

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

for filename in filenames:

print(os.path.join(dirname, filename))

**2.Load dataset**

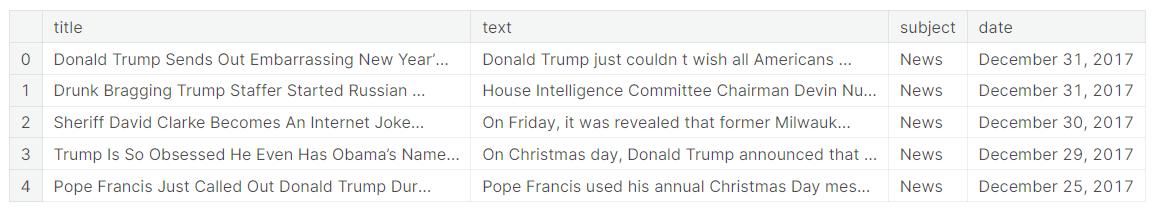
df\_fake=pd.read\_csv("/kaggle/input/fake-and-real-news-dataset/Fake.csv")

df\_true=pd.read\_csv("/kaggle/input/fake-and-real-news-dataset/True.csv")

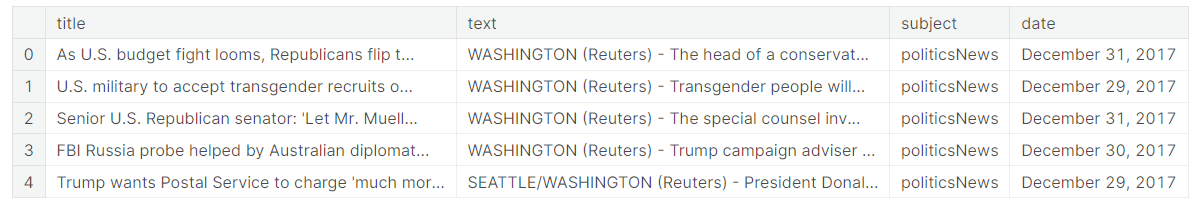
print("Fake verimiz: ",df\_fake.shape)

print("True verimiz: ",df\_true.shape)

df\_fake.head()



df\_true.head()



**3. Exploratory data analysis:**

**(I)Checking for missing values:**

df=df.fillna('')

df.isnull().sum()

title 0

text 0

subject 0

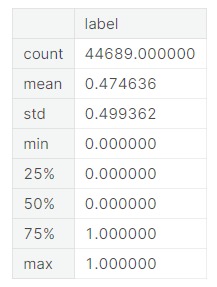
date 0

label 0

dtype: int64

**(II)Explore statistics:**

df.describe()



df=df\_true.append(df\_fake)

df.info()

<class 'pandas.core.frame.DataFrame'>

Int64Index: 44898 entries, 0 to 23480

Data columns (total 5 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 title 44898 non-null object

1 Fake/True 44898 non-null object

2 text 44898 non-null object

3 subject 44898 non-null object

4 date 44898 non-null object

dtypes: object(5)

memory usage: 2.1+ MB

df['Fake/True'].value\_counts()

Fake 23481

True 21417

Name: Fake/True, dtype: int64

df['subject'].value\_counts()

politicsNews 11272

worldnews 10145

News 9050

politics 6841

left-news 4459

Government News 1570

US\_News 783

Middle-east 778

Name: subject, dtype: int64

df.columns

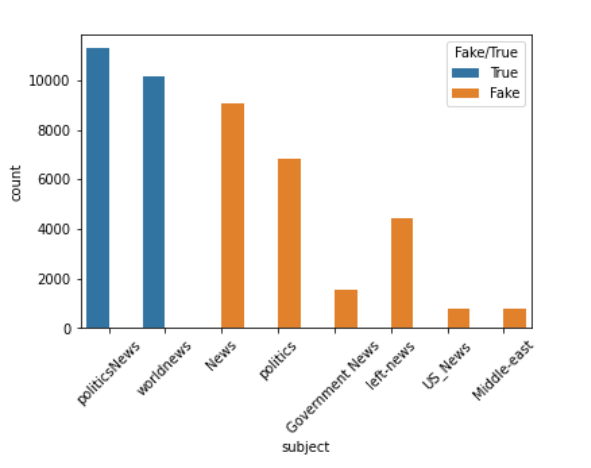
columns: Index(['title', 'text', 'subject', 'date'], dtype='object')

df=df.drop(columns=['title','date'],axis=1)

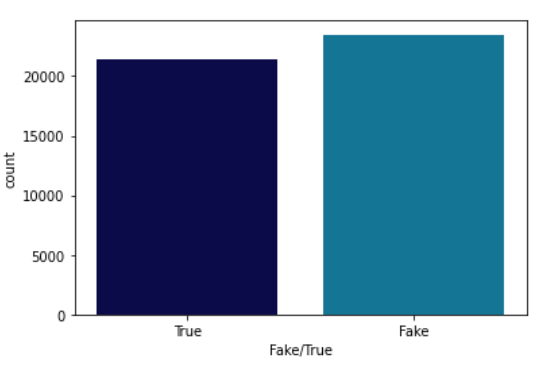
**(III)Data Visualization:**

sns.countplot(df['subject'],hue='Fake/True',data=df)

plt.xticks(rotation=45);



sns.countplot(df['Fake/True'],data=df,palette='ocean');



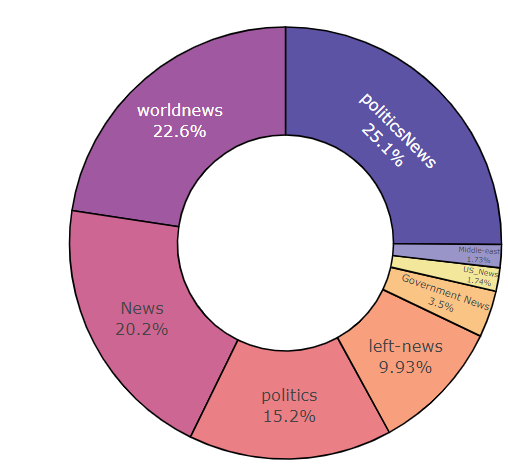
import plotly.express as px

fig = px.pie(df, names = "subject", title = "News Subject", hole = 0.5,

width = 1000, height = 500, color\_discrete\_sequence = px.colors.sequential.Sunset\_r)

fig.update\_traces(textposition = "inside", textinfo = "percent+label",

marker = dict(line = dict(width = 1.2, color = "#000000")))

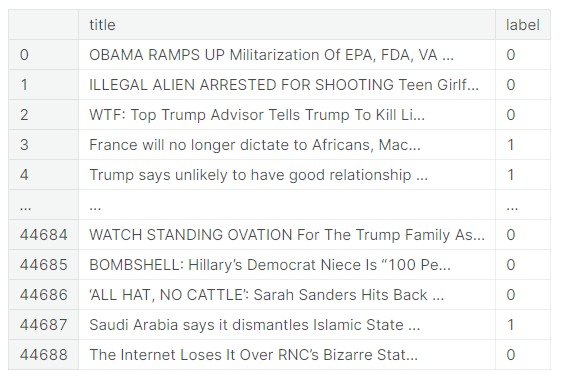




**4. Filtering the data for modeling**

mod\_df = fin\_df[['title','label']]

mod\_df



#separating title and label

x= mod\_df['title']

y = mod\_df['label']

**5. Splitting the data**

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=101)

**CONCLUSION**

In conclusion, the integration of Natural Language Processing (NLP) techniques in fake news detection represents a significant stride towards combating misinformation in the digital age. NLP models leverage linguistic features, syntactic structures, and semantic analysis to discern subtle cues indicative of deceptive content. This approach has shown promising results, achieving commendable accuracy rates in distinguishing fake from authentic news sources. However, it's important to note that the battle against fake news is a dynamic and ever-evolving one. Staying at the forefront of this challenge requires continuous research, dataset refinement, and the incorporation of multi-modal information sources. Moreover, a holistic strategy that combines NLP with other advanced technologies like image analysis and social network analysis will fortify the detection process. Collaborative efforts between academia, industry, and policymakers are essential in refining and deploying these systems effectively. As we move forward, the collective commitment to truth and accuracy will play a pivotal role in mitigating the impact of fake news on our information ecosystem.