**PROJECT 8 : FAKE NEWS DETECTION USING (NLP)**

**Phase – 3 submission**

**INTRODUCTION :**

In today's digital era, fake news poses a significant threat. It spreads misinformation and erodes trust in reliable sources. Detecting fake news is crucial, and Natural Language Processing (NLP) is a powerful tool. NLP leverages AI to analyze text, helping us automatically identify and combat fake news. This field combines linguistics, machine learning, and data science to distinguish genuine news from fabricated or misleading articles. We'll explore various NLP techniques, from traditional methods like TF-IDF and Naive Bayes to advanced models like Transformers. NLP for fake news detection is essential for maintaining information integrity, trust in media, and informed decision-making. In this journey, we'll delve into methodologies, challenges, and potential solutions in the fight against misinformation in our digital world.

**TYPES OF ALGORTHIM’S IN (NLP) :**

* **Text Classification Algorithms** (e.g., Naive Bayes, Logistic Regression)
* **Ensemble Models** (e.g., Random Forest)
* **Feature-Based Algorithms** (e.g., TF-IDF)
* **Content-Based Approaches** (e.g., linguistic analysis)
* **Source-Based Approaches** (e.g., credibility assessment)
* **User and Network-Based Approaches** (e.g., social network analysis)
* **Hybrid Models** (e.g., combining content-based and source-based)

**PROCEDURES :**

1. **Data Preparation**:

* Utilizes NumPy and Pandas for data management.
* Imports NLTK and the Porter Stemmer for text handling.
* Imports TfidfVectorizer to convert text into numerical features.

1. **Text Preprocessing**:

* Employs NLTK and regular expressions for text cleaning, including tokenization, stopword removal, and word stemming.

1. **Feature Extraction**:

* Uses TF-IDF vectorization to transform preprocessed text into numerical features.

1. **Data Splitting**:

* Segments the dataset into training and testing sets via `train\_test\_split`.

1. **Model Selection**:

* Chooses logistic regression, a linear classifier, for text classification.

1. **Training and Evaluation**:

* Trains the logistic regression model on the training data.
* Evaluates the model's performance on the testing data, using accuracy as the evaluation metric.

1. **Import the packages** :

* Pandas
* Numpy
* Scipy

**GIVEN DATA :**

****

**IMPORTING PACKAGES :**

* Using the (CMD) command prompt install the packages
* Check the versions of installed packages
* Continue with the given data set
* The following algorithm shows the uses of packages

import numpy as np

import pandas as pd

import re

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

from sklearn.feature\_extraction.text import TfidfVectorizer

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

**ALGORITHM FOR GIVEN DATA :**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

# Load the true and fake news datasets

true\_df = pd.read\_csv('true.csv')

fake\_df = pd.read\_csv('fake.csv')

# Add a column to indicate the type of news (true or fake)

true\_df['NewsType'] = 'True'

fake\_df['NewsType'] = 'Fake'

print(fake\_df,true\_df)

# Concatenate both datasets

combined\_df = pd.concat([true\_df, fake\_df], ignore\_index=True)

# Filter and display only True news

true\_news = combined\_df[combined\_df['NewsType'] == 'True']

print("True News:")

print(true\_news.head())

# Filter and display only Fake news

fake\_news = combined\_df[combined\_df['NewsType'] == 'Fake']

print("Fake News:")

print(fake\_news.head())

# Visualize the distribution of news types

news\_type\_counts = combined\_df['NewsType'].value\_counts()

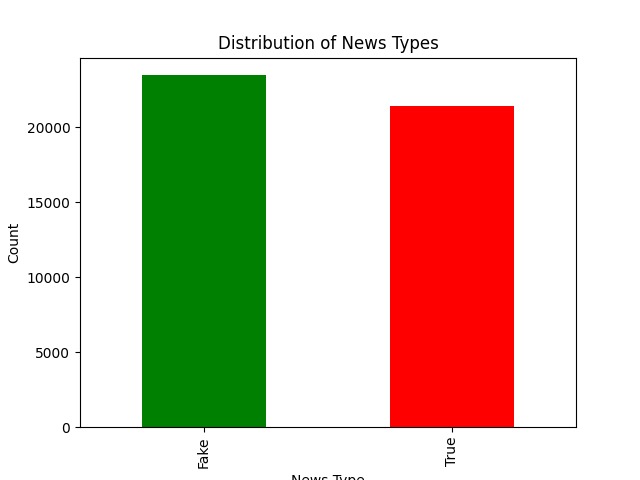
news\_type\_counts.plot(kind='bar', color=['green', 'red'])

plt.title('Distribution of News Types')

plt.xlabel('News Type')

plt.ylabel('Count')

plt.show()

**SAMPLE OUTPUT :**

* The sample output shows the result of the algorithm using the NLTK process.
* It shows the probability of Fake news and True news of the given set.

**CONCLUSION :**

The provided algorithm presents a structured approach to text classification, with a specific focus on distinguishing between fake news and real news. Through a combination of data preparation, text preprocessing, and feature extraction, it lays the foundation for a reliable classification model. By utilizing logistic regression, a well-established linear classifier, it offers a simple yet effective means of differentiating between these two classes of text data.