IBM NAAN MUDHALVAN

ARTIFICIAL INTELLIGENCE-GROUP 3

PROJECT: TEAM MEMBER:

FAKE NEWS DETECTON NAME:SRIRAJ.R

USING NLP REG NO:721921243111

PHASE 4: DEVELOPMENT PART-II



**INTRODUCTION:**

This project aims to develop a fake news detection system using natural language processing techniques to help users differentiate between real and fake news articles.This process involves data collection,text preprocessing and the application of deep learning models for classification.

**ALGORITHM:**

STEP 1: Import the library packages and modules.

STEP 2: Load the Datasets.

STEP 3: Exploring the Datasets.

STEP 4: Split the dataset into training and testing sets.

STEP 5: Preprocess the text data .

STEP 6: Select the algorithm to be used[Here,Logistic regression].

STEP 7:Evaluate the Model.

STEP 8:Print the results.

**PROGRAM:**

In the Development phase part-2 , I’ve completed my project such as txt preprocessimng ,Feature xtraction ,Model training and Model evaluation.

import pandas as pd

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

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix, precision\_score, recall\_score

fake\_data = pd.read\_csv('Fake.csv')

fake\_data['label'] = 'FAKE'

true\_data = pd.read\_csv('True.csv')

true\_data['label'] = 'REAL'

data = pd.concat([fake\_data, true\_data], ignore\_index=True)

# Split the dataset into training and testing sets

X = data['title'] + ' ' + data['text'] + ' ' + data['subject'] + ' ' + data['date']

y = data['label']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Preprocess the text data using TF-IDF vectorization

tfidf\_vectorizer = TfidfVectorizer(stop\_words='english')

X\_train\_tfidf = tfidf\_vectorizer.fit\_transform(X\_train)

X\_test\_tfidf = tfidf\_vectorizer.transform(X\_test)

#Using Logistic regression

epochs = 10

model = LogisticRegression(max\_iter=epochs)

#Validation Phase

model.fit(X\_train\_tfidf, y\_train)

y\_pred = model.predict(X\_test\_tfidf)

accuracy = accuracy\_score(y\_test, y\_pred) #Accuracy

print("Accuracy:", accuracy)

cm = confusion\_matrix(y\_test, y\_pred) #Confusion matrix

print("Confusion Matrix:")

print(cm)

precision = precision\_score(y\_test, y\_pred, pos\_label='FAKE') #Precision

print("Precision:", precision)

recall = recall\_score(y\_test, y\_pred, pos\_label='FAKE') #Recall

print("Recall:", recall)

new\_text = input("Enter a news heading:") #Getting input

new\_text\_tfidf = tfidf\_vectorizer.transform([new\_text])

prediction = model.predict(new\_text\_tfidf)

print("Predicted Label:", prediction) #Print the result

**Feature extraction:**

1.TF-IDF VECTORIZATION :

Data preprocessing is done by using TF-IDF vectorization.Term frequency measures the frequency of word in a document.Inverse Document Frequency measures how important a term is across a collection of documents.

2. Word Embeddings:

Techniques like Word2Vec, GloVe, or FastText can convert words into dense vector representations. These word embeddings can capture semantic relationships between words and are useful in understanding the context of words in news articles.

3.Topic Modeling:

Methods like Latent Dirichlet Allocation (LDA) or Non-Negative Matrix Factorization (NMF) can be used to identify topics within a collection of news articles. The topics can then serve as features for fake news detection

4. Named Entity Recognition (NER):

Identifying named entities like people, organizations, and locations can provide additional context for fake news detection.

**MODEL EVALUATION:**

1.Confusion Matrix:

The confusion matrix provides a summary of model predictions. It includes four values: true positives (correctly identified fake news), true negatives (correctly identified real news), false positives (real news misclassified as fake), and false negatives (fake news misclassified as real).

2.Accuracy:

Accuracy measures the overall correctness of the model's predictions and is calculated as (TP + TN) / (TP + TN + FP + FN).

3.Precision:

Precision assesses the proportion of positive predictions (fake news) that were correct and is calculated as TP / (TP + FP).

4.Recall (Sensitivity):

Recall measures the proportion of actual positive cases (fake news) that the model correctly identified and is calculated as TP / (TP + FN)

**CONCLUSION:**

In conclusion, I have build my fake news detection model using natural language processing and evaluate the performance of the model using accuracy ,confusion matrix,precision and recall.