IBM NAAN MUDHALVAN

# ARTIFICIAL INTELLIGENCE-GROUP 3

PROJECT: TEAM MEMBER:

FAKE NEWS DETECTON NAME:SRIRAJ.R

USING NLP REG NO:721921243111

PHASE 5:

PROJECT DOCUMENTATION AND SUBMISSION



**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.

**Problem Definition:**

The problem is to develop a fake news detection model using a Kaggle dataset. The goal is to distinguish between true and fake news articles based on their titles and text. This project involves using natural language processing (NLP) techniques to preprocess the text data, building a model for classification, and evaluating the model's performance.

**DIGITAL THINKING:**

* **Data Source:**

Choose the fake news dataset available on Kaggle, containing articles titles and text, along with their labels (genuine or fake).

* **Data Preprocessing:**

Clean and preprocess the textual data to prepare it for analysis.

* **Feature Extraction:**

Utilize techniques like TF-IDF (Term Frequency- Inverse Document Frequency) or word embeddings to convert text into numerical features.

* **Model Selection:**

Select a suitable classification algorithm (e.g.,

Logistic Regression, Random Forest, or Neural Networks) for the fake news detection task.

* **Model Training:**

Train the selected model using the preprocessed data.

* **Evaluation:**

Evaluate the model's performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.

**PHASES OF DEVELOPMENT[ALGORITHM]**

1: Import the library packages and modules.

2: Load the Datasets.

3: Exploring the Datasets.

4: Split the dataset into training and testing sets.

5: Preprocess the text data .

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

7:Evaluate the Model.

8:Print the results.

**DATASET:**

* **Fake News Dataset:** This dataset contains fake news articles and their

sources. It includes various types of fake news, such as satire, hoaxes, and propaganda. The dataset also provides information about the sources of these articles.

* **True News Dataset:** The True news dataset contains statements labeled as true. Each statement is accompanied by additional metadata, making it suitable for research on fake news detection
* **Dataset Link:**  https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-

news- dataset

**CLASSIFICATION ALGORITHM:**

Logistic regression is a common machine learning algorithm used in fake news detection with NLP (Natural Language Processing). In brief, here's how it works:

* Data Preparation: Collect a dataset of text articles, where each article is labeled as either real or fake news. Preprocess the text data by cleaning, tokenizing, and converting it into numerical features.
* Feature Extraction: Use techniques like TF-IDF (Term FrequencyInverse Document Frequency) or word embeddings (e.g.,

Word2Vec or GloVe) to represent the text as feature vectors.

* Model Training: Train a logistic regression classifier on the labeled data. The logistic regression model learns to assign probabilities to each article being real or fake based on the extracted features.
* Prediction: Given a new article, the trained model can predict the probability of it being fake news. If the probability exceeds a certain threshold, it is classified as fake; otherwise, it's considered real.
* Evaluation: Assess the model's performance using metrics like accuracy, precision, recall, and F1-score. Fine-tune the model and hyperparameters as needed

**DATA PREPROCESSSING:**

Data preprocessing involves cleaning and transforming raw data into a format that can be used for training machine learning models.

1.TEXT CLEANING:

It involves removing html tags,special characters and punctuation marks, converting text to lowercase to ensure uniformity, handling

contractions(e.g:can’t to cannot)and eliminating extra white spaces.

2.TOKENIZATION:

Tokenization is used to splitting the text into individual words or tokens.This step is essential for further analysis.

3.STOP WORD REMOVAL:

This Method removes the commom step words like “the” , “and”,”in” as they often do not provide meaningful information for fake news detection.

4.HANDLING NUMERICAL DATA :

This step dealing with the numbers such as replacing them with

the placeholders(e.g.”123” with “num”).

5.HANDLING MISSING DATA:

This step is the process of addressing missing or null values in the dataset through imputation or removal.

6.REMOVING IRRELEVANT INFORMATION:

Removing non-textual content,such as URLs,usernames and hastags,which are common in social media and data

7.TEXT NORMALIZATION:

Standardizing spellings,abbreviations and accronyms to ensure consistency in the data.

**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)

**PROGRAM:**

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

**CONCLUSION:**

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