TEAM MEMBERS:

1.C.K. AKSHAYA PRATHIKSHA

2.SO.SONIA

3.A.DIVYA

4.KUPPAIAH PAVITHRA

# FAKE NEWS DETECTION USING NPL -ARTIFICIAL INTELLEGENCE

PYTHON CODE:

python

import pandas as pd

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

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import accuracy\_score, classification\_report

import nltk

from nltk.corpus import stopwords

# Download the NLTK stopwords dataset

nltk.download('stopwords')

# Load the dataset (you need to have a labeled dataset with 'text' and 'label' columns)

# For simplicity, let's assume you have a CSV file with 'text' and 'label' columns.

# Replace 'your\_dataset.csv' with your actual dataset file.

df = pd.read\_csv('your\_dataset.csv')

# Data preprocessing

df.dropna(inplace=True) # Drop any rows with missing values

df.reset\_index(drop=True, inplace=True)

# Text preprocessing: removing stopwords and converting to lowercase

stop\_words = set(stopwords.words('english'))

df['text'] = df['text'].apply(lambda x: ' '.join([word.lower() for word in x.split() if word.lower() not in stop\_words]))

# Split the dataset into training and testing sets

X = df['text']

y = df['label'] # Assuming 0 for fake news and 1 for real news

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

# Feature extraction using CountVectorizer

vectorizer = CountVectorizer()

X\_train\_vectorized = vectorizer.fit\_transform(X\_train)

X\_test\_vectorized = vectorizer.transform(X\_test)

# Train a classifier (Multinomial Naive Bayes in this example)

clf = MultinomialNB()

clf.fit(X\_train\_vectorized, y\_train)

# Predictions

y\_pred = clf.predict(X\_test\_vectorized)

# Evaluate the model

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

print("Accuracy:", accuracy)

# Print classification report

print("Classification Report:")

print(classification\_report(y\_test, y\_pred))