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, classification\_report

import re

import nltk

from nltk.corpus import stopwords

# Download stopwords if you haven't already

nltk.download('stopwords', quiet=True)

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

# Load your social media dataset

try:

df = pd.read\_csv('social\_media\_fake\_news.csv')

# Assuming your text column is 'tweet\_text' and label is 'is\_fake'

X = df['tweet\_text'].astype(str) # Ensure it's string type

y = df['is\_fake']

except FileNotFoundError:

print("Error: social\_media\_fake\_news.csv not found. Please provide the correct path.")

exit()

# Preprocessing function

def preprocess\_text(text):

text = re.sub(r'@\w+', '', text) # Remove user mentions

text = re.sub(r'#\w+', '', text) # Remove hashtags

text = re.sub(r'http\S+|www\S+|https\S+', '', text) # Remove URLs

text = re.sub(r'[^a-zA-Z\s]', '', text, re.IGNORECASE) # Remove special chars and punctuation

text = text.lower()

tokens = text.split()

tokens = [w for w in tokens if not w in stop\_words]

return " ".join(tokens)

X\_processed = X.apply(preprocess\_text)

# Split data

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

# Feature extraction (TF-IDF)

tfidf\_vectorizer = TfidfVectorizer(max\_df=0.7)

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

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

# Train Logistic Regression model

model = LogisticRegression(random\_state=42)

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

# Make predictions

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

# Evaluate

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

print(f"Accuracy: {accuracy:.2f}")

print("\nClassification Report:")

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

# Example of predicting on new social media text

new\_social\_texts = [

"@elonmusk Just saw a doge fly to the moon! #dogecoin #tothemoon",

"Breaking: Scientists discover a new species of bioluminescent fish in the deep sea.",

"You won't believe what this celebrity said! Click here to find out (link).",

"The government just announced a major investment in renewable energy infrastructure."

]

new\_social\_texts\_processed = [preprocess\_text(text) for text in new\_social\_texts]

new\_social\_texts\_tfidf = tfidf\_vectorizer.transform(new\_social\_texts\_processed)

predictions = model.predict(new\_social\_texts\_tfidf)

print("\nPredictions for new social media texts:")

for text, prediction in zip(new\_social\_texts, predictions):

if prediction == 0:

print(f"'{text}' - Predicted: Real")

else:

print(f"'{text}' - Predicted: Fake")