

# fake\_news\_detection.py

import pandas as pd   
import numpy as np   
import string   
import nltk   
from sklearn.model\_selection import train\_test\_split   
from sklearn.feature\_extraction.text import TfidfVectorizer from sklearn.linear\_model import PassiveAggressiveClassifier from sklearn.metrics import accuracy\_score, confusion\_matrix

nltk.download('stopwords')   
from nltk.corpus import stopwords

# Load dataset   
df = pd.read\_csv("fake\_news.csv") # Update path if needed df = df[['text', 'label']] # Ensure only relevant columns

# Preprocessing function   
def clean\_text(text):   
 text = text.lower()   
 text = ''.join([ch for ch in text if ch not in string.punctuation]) words = text.split()   
 stop\_words = stopwords.words('english')   
 filtered\_words = [word for word in words if word not in stop\_words] return ' '.join(filtered\_words)

# Apply cleaning   
df['text'] = df['text'].astype(str).apply(clean\_text)

# Split data   
X\_train, X\_test, y\_train, y\_test = train\_test\_split(df['text'], df['label'], test\_size=0.2, random\_state=7)

# Vectorize text   
vectorizer = TfidfVectorizer(max\_df=0.7)   
tfidf\_train =   
vectorizer.fit\_transform(X\_train)   
tfidf\_test = vectorizer.transform(X\_test)

# Train model   
model = PassiveAggressiveClassifier(max\_iter=50) model.fit(tfidf\_train, y\_train)

# Predict and evaluate   
y\_pred = model.predict(tfidf\_test)   
score = accuracy\_score(y\_test, y\_pred)   
conf\_matrix = confusion\_matrix(y\_test, y\_pred)



print(f"Accuracy: {score \* 100:.2f}%")   
print("Confusion Matrix:")   
print(conf\_matrix)