# fake\_news\_detection.py

import pandas as pd

import numpy as np

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 nltk

import string

from nltk.corpus import stopwords

import re

nltk.download('stopwords')

# Load dataset

def load\_data(filepath):

return pd.read\_csv(filepath)

# Preprocess text

def preprocess(text):

text = text.lower()

text = re.sub(r"http\S+", "", text) # Remove URLs

text = re.sub(r"[^\w\s]", "", text) # Remove punctuation

text = re.sub(r"\d+", "", text) # Remove numbers

tokens = text.split()

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

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

return " ".join(tokens)

# Prepare dataset

def prepare\_dataset(df):

df = df[['text', 'label']]

df['text'] = df['text'].apply(preprocess)

return df

# Train model

def train\_model(X\_train, y\_train):

tfidf = TfidfVectorizer(max\_features=5000)

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

model = LogisticRegression()

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

return model, tfidf

# Evaluate model

def evaluate(model, tfidf, X\_test, y\_test):

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

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

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

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

# Predict single news

def predict\_news(model, tfidf, text):

cleaned = preprocess(text)

vect = tfidf.transform([cleaned])

prediction = model.predict(vect)

return prediction[0]

# Run full pipeline

def main():

df = load\_data("fake\_news\_dataset.csv") # dataset must have columns: text, label (1=fake, 0=real)

df = prepare\_dataset(df)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(df['text'], df['label'], test\_size=0.2, random\_state=42)

model, tfidf = train\_model(X\_train, y\_train)

evaluate(model, tfidf, X\_test, y\_test)

# Try with your own example

example\_news = "The government has announced a secret vaccine for immortality."

prediction = predict\_news(model, tfidf, example\_news)

print(f"Prediction: {'Fake' if prediction == 1 else 'Real'}")

if \_\_name\_\_ == "\_\_main\_\_":

main()