# ✅ Install necessary libraries

!pip install -q nltk gradio

# ✅ Import libraries

import zipfile, io, re, string

import pandas as pd

import numpy as np

import nltk

import gradio as gr

from google.colab import files

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

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.linear\_model import PassiveAggressiveClassifier

from sklearn.utils import resample

# ✅ Download stopwords

nltk.download('stopwords')

from nltk.corpus import stopwords

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

# ✅ File upload

uploaded = files.upload() # Upload the zip file containing Fake.csv and True.csv

# ✅ Extract uploaded zip

zip\_file = list(uploaded.keys())[0]

with zipfile.ZipFile(io.BytesIO(uploaded[zip\_file]), 'r') as zip\_ref:

zip\_ref.extractall()

# ✅ Load datasets

fake = pd.read\_csv("Fake.csv")

real = pd.read\_csv("True.csv")

fake['label'] = 0

real['label'] = 1

# ✅ Combine and clean

df = pd.concat([fake, real]).reset\_index(drop=True)

df['combined'] = df['title'] + " " + df['text']

def clean\_text(text):

text = text.lower()

text = re.sub(r"http\S+|www.\S+", "", text)

text = re.sub(r"<.\*?>", "", text)

text = re.sub(r"[%s]" % re.escape(string.punctuation), "", text)

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

text = re.sub(r"\s+", " ", text)

return " ".join([word for word in text.split() if word not in stop\_words])

df['clean\_text'] = df['combined'].apply(clean\_text)

# ✅ Balance dataset

df\_fake = df[df['label'] == 0]

df\_real = df[df['label'] == 1]

df\_real\_downsampled = resample(df\_real, replace=True, n\_samples=len(df\_fake), random\_state=42)

df\_balanced = pd.concat([df\_fake, df\_real\_downsampled]).sample(frac=1, random\_state=42)

# ✅ Train/Test split

X = df\_balanced['clean\_text']

y = df\_balanced['label']

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

# ✅ TF-IDF Vectorization

vectorizer = TfidfVectorizer(max\_df=0.7, min\_df=5, max\_features=5000)

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

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

# ✅ Model training

model = PassiveAggressiveClassifier(max\_iter=1000)

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

# ✅ Gradio prediction function

def predict\_news(text):

cleaned = clean\_text(text)

vec = vectorizer.transform([cleaned])

pred = model.predict(vec)[0]

return "🟢 REAL News" if pred == 1 else "🔴 FAKE News"

# ✅ Launch Gradio interface

demo = gr.Interface(

fn=predict\_news,

inputs=gr.Textbox(lines=10, placeholder="Paste news content here..."),

outputs=gr.Label(label="Prediction"),

title="📰 Fake News Detector",

description="Paste a news article to check if it's REAL or FAKE using NLP + ML.",

)