import pickle

import re

import nltk

from fastapi import FastAPI

from pydantic import BaseModel

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

from nltk.stem import PorterStemmer

from sklearn.feature\_extraction.text import TfidfVectorizer

# Download NLTK dependencies

nltk.download('stopwords')

nltk.download('punkt')

# Load the trained model and vectorizer

with open("model.pkl", "rb") as model\_file:

model = pickle.load(model\_file)

with open("vectorizer.pkl", "rb") as vectorizer\_file:

vectorizer = pickle.load(vectorizer\_file)

# Initialize FastAPI

app = FastAPI()

class ReviewRequest(BaseModel):

review: str

# Preprocessing function

def preprocess\_text(text):

text = text.lower()

text = re.sub(r'[^a-zA-Z\s]', '', text)

tokens = word\_tokenize(text)

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

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

stemmer = PorterStemmer()

tokens = [stemmer.stem(word) for word in tokens]

return ' '.join(tokens)

@app.post("/predict/")

def predict\_review(data: ReviewRequest):

processed\_review = preprocess\_text(data.review)

transformed\_review = vectorizer.transform([processed\_review])

prediction = model.predict(transformed\_review)

return {

"review": data.review,

"fake\_review": bool(prediction[0]),

"message": "Fake review detected!" if prediction[0] else "Review is genuine."

}

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

import uvicorn

uvicorn.run(app, host="0.0.0.0", port=8000)