import os

#libraries

import numpy as np

import tensorflow as tf

from tensorflow.keras.models import Model

from tensorflow.keras.layers import Input, Dense, Embedding, Layer, LayerNormalization, Dropout

from tensorflow.keras.preprocessing.text import Tokenizer

from tensorflow.keras.preprocessing.sequence import pad\_sequences

from tensorflow.keras.utils import to\_categorical

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

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

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import accuracy\_score, classification\_report

import joblib

#used for data cleaning

import re

import nltk

from nltk.corpus import stopwords

from sklearn.feature\_extraction.text import TfidfVectorizer

#Loading the fake news dataset

import kagglehub

path = kagglehub.dataset\_download("emineyetm/fake-news-detection-datasets")

print("Path to dataset files:", path)

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename))

#Creating the dataframes for true and fake

dataframeT = pd.read\_csv('/kaggle/input/fake-news-detection-datasets/News \_dataset/True.csv')

dataframeF = pd.read\_csv('/kaggle/input/fake-news-detection-datasets/News \_dataset/Fake.csv')

dataframeT

dataframeF

#Combining the datasets into one dataframe

dataframeT['label'] = 1

dataframeF['label'] = 0

dataframe = pd.concat([dataframeT, dataframeF], ignore\_index=True)

nltk.download('stopwords')

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

def clean\_text(text):

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

    text = re.sub(r'\d+', '', text)  #removes numbers

    text = text.lower()  #turns everything to lowercase

    text = ' '.join([word for word in text.split() if word not in stop\_words]) #gets rid of stop words

    return text

dataframe['text'] = dataframe['text'].apply(clean\_text) #applies changes

#some train test splitting, 20%

X = dataframe['text']

y = dataframe['label']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=142857)

#turns everything into vectors

vectorizer = TfidfVectorizer(max\_features=5000)

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

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

# Create the directory if it doesn't exist

os.makedirs('/kaggle/working/', exist\_ok=True)

# Now you can save the vectorizer

joblib.dump(vectorizer, ("/kaggle/working/version4\_vectorizer.pkl"))

#This project uses the naive bayes model

model = MultinomialNB()

model.fit(X\_train\_vectorized, y\_train) #doe the actual data fitting

joblib.dump(model, ("/kaggle/working/version4\_NB.pkl"))

predictions = model.predict(X\_test\_vectorized) #stores alll of the predictions into a variable

print(f"Accuracy: {accuracy\_score(y\_test, predictions):.2f}") #compares with actual results

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