DETECTING FAKE NEWS IN SOCIALMEDIA WITH PYTHON AND MACHINE LEARNING

// MODEL TRAINING

PLATFORM USED : Google Collab

1. !pip install pandas numpy scikit-learn nltk

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. from google.colab import drive

# Mount Google Drive

drive.mount('/content/drive')

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. import pandas as pd

# Load dataset from 'fr' folder

fake\_df = pd.read\_csv('/content/drive/MyDrive/fake/fr/Fake.csv')

real\_df = pd.read\_csv('/content/drive/MyDrive/fake/fr/True.csv')

# Display the first few rows of both datasets

print("Fake Dataset:")

print(fake\_df.head())

print("\nReal Dataset:")

print(real\_df.head())

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. import pandas as pd

from sklearn.feature\_extraction.text import TfidfVectorizer

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

from sklearn.linear\_model import PassiveAggressiveClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix

import joblib

# Load Datasets from 'fr' folder

fake\_df = pd.read\_csv('/content/drive/MyDrive/fake/fr/Fake.csv')

real\_df = pd.read\_csv('/content/drive/MyDrive/fake/fr/True.csv')

# Combine datasets into one (if needed, adjust based on your requirements)

fake\_df['label'] = 0 # Assign 0 for fake news

real\_df['label'] = 1 # Assign 1 for real news

# Combine both datasets into one

df = pd.concat([fake\_df, real\_df], ignore\_index=True)

df.dropna(inplace=True)

# Preprocessing

X = df['text']

y = df['label']

# Vectorization

vectorizer = TfidfVectorizer(stop\_words='english', max\_df=0.7)

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

# Split Data

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

# Model Training

model = PassiveAggressiveClassifier(max\_iter=50)

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

# Evaluate Model

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

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Accuracy: {round(accuracy\*100)}%")

print("Confusion Matrix:")

print(confusion\_matrix(y\_test, y\_pred))

# Save Model and Vectorizer to Google Drive

joblib.dump(model, '/content/drive/MyDrive/fake\_news\_model.pkl') # Corrected path

joblib.dump(vectorizer, '/content/drive/MyDrive/vectorizer.pkl') # Corrected path

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.import joblib

import pandas as pd

# Load the saved model and vectorizer

model = joblib.load('/content/drive/MyDrive/fake\_news\_model.pkl')

vectorizer = joblib.load('/content/drive/MyDrive/vectorizer.pkl')

# Function to classify a new news text

def classify\_news(news\_text):

# Vectorize the new news text

news\_vectorized = vectorizer.transform([news\_text])

# Predict the class using the trained model

prediction = model.predict(news\_vectorized)

# Interpret the result (0 for Fake, 1 for Real)

if prediction == 0:

return "Fake News"

else:

return "Real News"

# Example: Classify a new news text

new\_news = "Aam Aadmi Party chief Arvind Kejriwal on Monday (December 30, 2024) announced that if the party wins the Delhi Assembly election due in February, they will give ₹18,000 per month to priests in temples and granthis of gurudwaras under the ‘Pujari Granthi Samman Yojana";

result = classify\_news(new\_news)

# Display the result

print(f"The news is: {result}")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6import joblib

# Saved the trained model to Google Drive

joblib.dump(model, '/content/drive/MyDrive/fake\_news\_model.pkl')

# Saved the trained vectorizer to Google Drive

joblib.dump(vectorizer, '/content/drive/MyDrive/vectorizer.pkl')

# Confirming message that both the model and vectorizer have been saved

print("Model and Vectorizer saved successfully!")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CONNECTING BACKEND USING STREAMLIT

PLATFORM: VS CODE

STEP1: In Vs code Open your Project folder (make sure that trainedmode.pkl,vectorizer.pkl should be i n folder)

/project-folder

│

├── fake\_news\_model.h5

├── vectorizer.pkl

STEP2: Create a new file in your Project Folder (requirements.txt)

//write these in that .txt file

streamlit

pandas

numpy

scikit-learn

tensorflow

joblib

Open New Terminal and Run this.txt File

COMMAND: pip install -r requirements.txt

After Succesful Running, In the terminal we observe that steamlit-env is created in folder

STEP 3: Create an Another File in your Project Folder(app.py)

//CODE

import streamlit as st

import joblib

# 🎯 Title and Description

st.title('📰 Fake News Detection App')

st.write('Enter news text to predict whether it is Fake or Real.')

# 🚀 Loading the Trained Model and Vectorizer

@st.cache\_resource # Cache the model and vectorizer to avoid reloading every interaction

def load\_model\_and\_vectorizer():

try:

# Loading your model and vectorizer from files

model = joblib.load('fake\_news\_model.pkl') # Ensure this is the correct path

vectorizer = joblib.load('vectorizer.pkl')

return model, vectorizer

except Exception as e:

st.error(f"❌ Error loading model or vectorizer: {e}")

return None, None

model, vectorizer = load\_model\_and\_vectorizer()

# 📥 User Input Section

st.header('📝 Enter News Text')

# User input for text

news\_text = st.text\_area("Paste the news article here:")

# 🧠 Make Predictions

if st.button('🔍 Predict'):

if model and vectorizer:

try:

# Vectorize the user input text

news\_vectorized = vectorizer.transform([news\_text])

# Make prediction

prediction = model.predict(news\_vectorized) # Ensure model has predict method

result = '🛑 Fake News' if prediction == 0 else '✅ Real News'

st.success(f'Prediction: {result}')

except Exception as e:

st.error(f"❌ Error during prediction: {e}")

else:

st.warning('⚠️ Model or vectorizer is not loaded. Please check your setup.')

# 📊 Additional Information

st.sidebar.header('ℹ️ About')

st.sidebar.info(

'This app uses a trained machine learning model to classify news articles as Fake or Real. '

'Simply paste the news text and click "Predict" to get the result.'

)

# Footer

st.markdown('---')

st.caption('Developed with ❤️ by Jakku Kumarswami')

RUN in terminal using command : streamlit run app.py

It will redirect to edge or browser to display the output