**#Upload the Dataset**

from google.colab import files

uploaded = files.upload()

**# Load the Dataset**

import pandas as pd

df = pd.read\_csv('evaluation.csv', delimiter=';')

df.head()

**# Data Exploration**

df.info()

df.describe()

df['label'].value\_counts()

**# Check for Missing Values and Duplicates**

print("Missing Values:\n", df.isnull().sum())

print("\nDuplicate Rows:", df.duplicated().sum())

df = df.drop\_duplicates()

**# Visualize a Few Features**

import matplotlib.pyplot as plt

import seaborn as sns

sns.countplot(x='label', data=df)

plt.title("Real (1) vs Fake (0) News")

plt.show()

**#Identify Target and Features**

X = df['text']

y = df['label']

**# Feature Scaling — NLP: TF-IDF**

from sklearn.feature\_extraction.text import TfidfVectorizer

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

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

**#Train-Test Split**

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

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

**#Model Building**

from sklearn.linear\_model import LogisticRegression model = LogisticRegression() model.fit(X\_train, y\_train)

**#Evaluation**

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

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

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

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

**# Make Predictions from New Input**

def predict\_news(news\_text):

vector = tfidf.transform([news\_text])

prediction = model.predict(vector)[0]

return 'Real News' if prediction == 1 else 'Fake News'

predict\_news("Government announces new economic reforms.")

**# Predict the Final Grade**

from sklearn.metrics import accuracy\_score

final\_grade = accuracy\_score(y\_test, y\_pred) \* 100

print(f"Final model grade: {final\_grade:.2f}% accuracy")

**# Deployment — Building an Interactive App**

!pip install gradio

import gradio as gr

**# Create a Prediction Function**

def fake\_news\_predictor(text):

vector = tfidf.transform([text])

prediction = model.predict(vector)[0]

return "Real News ✅ " if prediction == 1 else "Fake News ❌ "

**# Create the Gradio Interface**

import gradio as gr

interface = gr.Interface(fn=fake\_news\_predictor, inputs="text", outputs="text", title="Fake News Detector")

interface.launch()