**1. Upload the Dataset**

from google.colab import files

uploaded = files.upload()

**2. Load the Dataset**

import pandas as pd

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

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

**3. Data Exploration**

print("Fake News Sample:")

print(fake.head())

print("\nReal News Sample:")

print(real.head())

**4. Check for Missing Values and Duplicates**

print("Missing in Fake:", fake.isnull().sum())

print("Duplicates in Fake:", fake.duplicated().sum())

print("Missing in Real:", real.isnull().sum())

print("Duplicates in Real:", real.duplicated().sum())

**5. Visualize a Few Features**

import matplotlib.pyplot as plt

fake['text\_length'] = fake['text'].apply(len)

real['text\_length'] = real['text'].apply(len)

plt.hist(fake['text\_length'], bins=50, alpha=0.7, label='Fake')

plt.hist(real['text\_length'], bins=50, alpha=0.7, label='Real')

plt.legend()

plt.title('Article Length Distribution')

plt.show()

**6. Identify Target and Features**

fake['label'] = 0

real['label'] = 1

data = pd.concat([fake[['text', 'label']], real[['text', 'label']]], ignore\_index=True)

**7. Train-Test Split**

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

from sklearn.feature\_extraction.text import TfidfVectorizer

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

X = vectorizer.fit\_transform(data['text'])

y = data['label']

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

**8. Model Building**

from sklearn.linear\_model import LogisticRegression

model = LogisticRegression()

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

**9. Evaluation**

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

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

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

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))

**10. Make Predictions from New Input**

def predict\_news(text):

vector = vectorizer.transform([text])

pred = model.predict(vector)

return "Real" if pred[0] == 1 else "Fake"

print(predict\_news("Government unveils new economic plan to boost job growth."))

**11. Convert to DataFrame and Encode**

sample\_df = pd.DataFrame({

"text": ["Aliens landed on the White House lawn.", "Vaccines reduce disease spread."],

})

sample\_df['text'] = sample\_df['text'].apply(lambda x: vectorizer.transform([x]))

**12. Deployment - Building an Interactive App**

!pip install gradio

**13. Create a Prediction Function**

def fake\_news\_detector(text):

vector = vectorizer.transform([text])

prediction = model.predict(vector)

return "Real News" if prediction[0] == 1 else "Fake News"

**14. Create the Gradio Interface**

import gradio as gr

iface = gr.Interface(fn=fake\_news\_detector, inputs="text", outputs="text", title="📰 Fake News Detector")

iface.launch()