**1. Data Collection**

python

CopyEdit

import pandas as pd

true\_df = pd.read\_csv('true.csv')

fake\_df = pd.read\_csv('fake.csv')

true\_df['NewsType'] = 'True'

fake\_df['NewsType'] = 'Fake'

combined\_df = pd.concat([true\_df, fake\_df], ignore\_index=True)

**2. Data Preprocessing**

python

CopyEdit

import re

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

from nltk.stem import PorterStemmer

nltk.download('punkt')

nltk.download('stopwords')

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

stemmer = PorterStemmer()

def preprocess\_text(text):

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

tokens = word\_tokenize(text.lower())

tokens = [stemmer.stem(word) for word in tokens if word not in stop\_words]

return ' '.join(tokens)

combined\_df['processed\_text'] = combined\_df['text'].apply(preprocess\_text)

**3. Feature Extraction**

python

CopyEdit

from sklearn.feature\_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max\_features=5000)

X = vectorizer.fit\_transform(combined\_df['processed\_text'])

**4. Model Training**

python

CopyEdit

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

from sklearn.linear\_model import LogisticRegression

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

y = combined\_df['NewsType'].apply(lambda x: 1 if x == 'Fake' else 0)

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

model = LogisticRegression()

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

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

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

print(f"Classification Report:\n{classification\_report(y\_test, y\_pred)}")

**5. Model Evaluation**

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay

import matplotlib.pyplot as plt

cm = confusion\_matrix(y\_test, y\_pred)

disp = ConfusionMatrixDisplay(confusion\_matrix=cm, display\_labels=['Real', 'Fake'])

disp.plot(cmap=plt.cm.Blues)

plt.show()