8. Naïve Bayes Classifier: Sentiment analysis on a Twitter dataset

# Install required packages if not already installed:

# pip install nltk scikit-learn pandas

import pandas as pd

import nltk

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

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.naive\_bayes import MultinomialNB

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

# Download stopwords if not already present

nltk.download('stopwords')

from nltk.corpus import stopwords

import string

# Sample Twitter dataset (replace with your own dataset for real application)

data = {

'tweet': [

"I love this phone!",

"This movie is terrible...",

"Had an awesome day today :)",

"I hate waiting in traffic",

"Such a boring game.",

"Best concert ever!",

"I'm so sad right now.",

"What a great experience!",

"Worst customer service.",

"Feeling happy and blessed!"

],

'sentiment': [

'positive', 'negative', 'positive', 'negative', 'negative',

'positive', 'negative', 'positive', 'negative', 'positive'

]

}

# Load the dataset

df = pd.DataFrame(data)

# Preprocessing function

def preprocess\_text(text):

text = text.lower()

text = ''.join([char for char in text if char not in string.punctuation])

tokens = text.split()

tokens = [word for word in tokens if word not in stopwords.words('english')]

return ' '.join(tokens)

# Apply preprocessing

df['clean\_tweet'] = df['tweet'].apply(preprocess\_text)

# Convert text to feature vectors

vectorizer = CountVectorizer()

X = vectorizer.fit\_transform(df['clean\_tweet'])

y = df['sentiment']

# Split dataset

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

# Train Naïve Bayes classifier

nb\_classifier = MultinomialNB()

nb\_classifier.fit(X\_train, y\_train)

# Make predictions

y\_pred = nb\_classifier.predict(X\_test)

# Evaluate the model

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

print("\nClassification Report:")

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