**Sentiment Analysis Project on Twiiter Comments**

***CODE:***

import numpy as np

import pandas as pd

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

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.preprocessing import LabelEncoder

from sklearn.pipeline import Pipeline

from sklearn.naive\_bayes import MultinomialNB

from sklearn.ensemble import RandomForestClassifier

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

import spacy

# Read the dataset with name "Emotion\_classify\_Data.csv" and store it in a variable df

columns = ['id','country','Label','Text']

df = pd.read\_csv(r"C:\Manit\AIML\archive\twitter\_training.csv", names=columns)

# Print the shape of dataframe

print(df.shape)

# Print top 5 rows

print(df.head(5))

df.info()

# Check the distribution of Emotion

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

# Show sample

for i in range(5):

    print(f"{i+1}: {df['Text'][i]} -> {df['Label'][i]}")

df.dropna(inplace=True)

# load english language model and create nlp object from it

nlp = spacy.load("en\_core\_web\_sm")

# use this utility function to get the preprocessed text data

def preprocess(text):

    # remove stop words and lemmatize the text

    doc = nlp(text)

    filtered\_tokens = []

    for token in doc:

        if token.is\_stop or token.is\_punct:

            continue

        filtered\_tokens.append(token.lemma\_)

    return " ".join(filtered\_tokens)

df['Preprocessed Text'] = df['Text'].apply(preprocess)

df

le\_model = LabelEncoder()

df['Label'] = le\_model.fit\_transform(df['Label'])

df.head(5)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(df['Preprocessed Text'], df['Label'],

                                                    test\_size=0.2, random\_state=42, stratify=df['Label'])

print("Shape of X\_train: ", X\_train.shape)

print("Shape of X\_test: ", X\_test.shape)

# Create classifier

clf = Pipeline([

    ('vectorizer\_tri\_grams', TfidfVectorizer()),

    ('naive\_bayes', (MultinomialNB()))

])

# Model training

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

# Get prediction

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

# Print score

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

# Print classification report

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

clf = Pipeline([

    ('vectorizer\_tri\_grams', TfidfVectorizer()),

    ('naive\_bayes', (RandomForestClassifier()))

])

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

# Get the predictions for X\_test and store it in y\_pred

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

# Print Accuracy

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

# Print the classfication report

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

test\_df = pd.read\_csv(r"C:\Manit\AIML\archive\twitter\_validation.csv", names=columns)

test\_df.head()

test\_text = test\_df['Text'][10]

print(f"{test\_text} ===> {test\_df['Label'][10]}")

test\_text\_processed = [preprocess(test\_text)]

test\_text\_processed

test\_text = clf.predict(test\_text\_processed)

classes = ['Irrelevant', 'Natural', 'Negative', 'Positive']

print(f"True Label: {test\_df['Label'][10]}")

print(f'Predict Label: {classes[test\_text[0]]}')

***OUTPUT:***

*(74682, 4)*

*id ... Text*

*0 2401 ... im getting on borderlands and i will murder yo...*

*1 2401 ... I am coming to the borders and I will kill you...*

*2 2401 ... im getting on borderlands and i will kill you ...*

*3 2401 ... im coming on borderlands and i will murder you...*

*4 2401 ... im getting on borderlands 2 and i will murder ...*

*[5 rows x 4 columns]*

*<class 'pandas.core.frame.DataFrame'>*

*RangeIndex: 74682 entries, 0 to 74681*

*Data columns (total 4 columns):*

*# Column Non-Null Count Dtype*

*--- ------ -------------- -----*

*0 id 74682 non-null int64*

*1 country 74682 non-null object*

*2 Label 74682 non-null object*

*3 Text 73996 non-null object*

*dtypes: int64(1), object(3)*

*memory usage: 2.3+ MB*

*1: im getting on borderlands and i will murder you all , -> Positive*

*2: I am coming to the borders and I will kill you all, -> Positive*

*3: im getting on borderlands and i will kill you all, -> Positive*

*4: im coming on borderlands and i will murder you all, -> Positive*

*5: im getting on borderlands 2 and i will murder you me all, -> Positive*

*Shape of X\_train: (59196,)*

*Shape of X\_test: (14800,)*

*0.7312837837837838*

*precision recall f1-score support*

*0 0.95 0.46 0.62 2575*

*1 0.65 0.90 0.76 4472*

*2 0.84 0.63 0.72 3622*

*3 0.71 0.81 0.76 4131*

*accuracy 0.73 14800*

*macro avg 0.79 0.70 0.71 14800*

*weighted avg 0.77 0.73 0.72 14800*

*0.9114864864864864*

*precision recall f1-score support*

*0 0.97 0.86 0.91 2575*

*1 0.93 0.93 0.93 4472*

*2 0.94 0.89 0.91 3622*

*3 0.85 0.94 0.90 4131*

*accuracy 0.91 14800*

*macro avg 0.92 0.91 0.91 14800*

*weighted avg 0.91 0.91 0.91 14800*

*The professional dota 2 scene is fucking exploding and I completely welcome it.*

*Get the garbage out. ===> Positive*

*True Label: Positive*

*Predict Label: Positive*