



```
In [20]: # Step 1: Import Libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re

# Download NLTK data
nltk.download('stopwords')
nltk.download('wordnet')

# Step 2: Load Dataset
df = pd.read_csv(r"C:\TE sem 7\machine learning\lab\tweet_emotions.csv") # reading dataset

# Step 3: Text Preprocessing
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

def preprocess_text(text):
    text = str(text).lower() # lowercase
    text = re.sub(r'[^\w\s]', ' ', text) # remove punctuation/numbers
    tokens = text.split()
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_words]
    return ' '.join(tokens)

df['clean_content'] = df['content'].apply(preprocess_text)

# Step 4: Feature Extraction using TF-IDF
vectorizer = TfidfVectorizer(max_features=3000)
X = vectorizer.fit_transform(df['clean_content'])
y = df['sentiment']

# Step 5: Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 6: Model Training
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)

# Step 7: Model Evaluation
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))

# Step 8: Predict Emotion for New Text
def predict_emotion(text):
    text = preprocess_text(text)
    vector = vectorizer.transform([text])
```

```
    return model.predict(vector)[0]

# Example
sample_text = "Feeling so happy and excited today!"
print("Predicted Emotion:", predict_emotion(sample_text))
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]      C:\Users\vidhi\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]      C:\Users\vidhi\AppData\Roaming\nltk_data...
[nltk_data] Package wordnet is already up-to-date!
```

```
Accuracy: 0.34625
```

```
C:\Users\vidhi\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1
531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in l
abels with no predicted samples. Use `zero_division` parameter to control this
behavior.
    _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\vidhi\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1
531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in l
abels with no predicted samples. Use `zero_division` parameter to control this
behavior.
    _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\vidhi\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1
531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in l
abels with no predicted samples. Use `zero_division` parameter to control this
behavior.
    _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

Classification Report:

	precision	recall	f1-score	support
anger	0.00	0.00	0.00	19
boredom	0.00	0.00	0.00	31
empty	0.33	0.01	0.01	162
enthusiasm	0.00	0.00	0.00	163
fun	0.08	0.01	0.02	338
happiness	0.33	0.35	0.34	1028
hate	0.48	0.16	0.24	268
love	0.49	0.38	0.43	762
neutral	0.34	0.56	0.42	1740
relief	0.32	0.02	0.04	352
sadness	0.33	0.25	0.29	1046
surprise	0.32	0.05	0.09	425
worry	0.33	0.48	0.39	1666
accuracy			0.35	8000
macro avg	0.26	0.17	0.17	8000
weighted avg	0.33	0.35	0.31	8000

Confusion Matrix:

```
[[ 0  0  0  0  0  1  0  0  7  0  1  0  0 10]
 [ 0  0  0  0  0  0  0  1 10  0  5  0 15]
 [ 0  0  1  0  0  8  1  5 92  0 10  0 45]
 [ 0  0  0  0  1 26  2  5 82  0  9  2 36]
 [ 0  0  0  0  4 79  3 24 122 1 22  6 77]
 [ 0  0  0  0 17 358  0 102 352 3 36  7 153]
 [ 0  0  0  0  2  8  42  2 70  0 48  2 94]
 [ 0  0  0  0  5 177  1 289 163 2 27  3 95]
 [ 0  0  2  0 11 151  7 54 979 6 93 14 423]
 [ 0  0  0  0  0 70  1 21 134 7 28  1 90]
 [ 0  0  0  0  3 50  12 26 260 0 264  5 426]
 [ 0  0  0  0  0 61  2 23 147 2 28  23 139]
 [ 0  1  0  0  5 105 16 41 466 1 218 10 803]]
```

Predicted Emotion: happiness

In []: