In [1]:
```
import pandas as pd
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 accuracy_score, classification_report, confusion_matrix
```

In [2]:
```
data = pd.read_csv("C:/Users/SAI/OneDrive/Desktop/Data Science/my experiments/emotion/t
```

In [3]:
```
print(data.head())
```

```
                                                text  label
0      i just feel really helpless and heavy hearted      4
1  ive enjoyed being able to slouch about relax a...      0
2  i gave up my internship with the dmrg and am f...      4
3                         i dont know i feel so lost      0
4  i am a kindergarten teacher and i am thoroughl...      4
```

In [4]:
```
X = data['text']
y = data['label']
```

In [5]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4
```

In [6]:
```
vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)
```

In [7]:
```
naive_bayes_model = MultinomialNB()
naive_bayes_model.fit(X_train_vectorized, y_train)
```

Out[7]:
```
MultinomialNB()
```

In [8]:
```
y_pred = naive_bayes_model.predict(X_test_vectorized)
```

In [9]:
```
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
```

In [10]:

```
print(f"Accuracy: {accuracy:.2f}")
print("\nConfusion Matrix:")
print(conf_matrix)
print("\nClassification Report:")
print(class_report)
```

```
Accuracy: 0.86

Confusion Matrix:
[[22780   649   100   358   289    15]
 [  920 26160   749   150   163    36]
 [  357  2179  4106    67    36     3]
 [ 1004   492    53  9647   271     7]
 [  866   475    33   388  7810   108]
 [  381   871    29    47   690  1079]]

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.94      0.90     24191
           1       0.85      0.93      0.89     28178
           2       0.81      0.61      0.69      6748
           3       0.91      0.84      0.87     11474
           4       0.84      0.81      0.82      9680
           5       0.86      0.35      0.50      3097

    accuracy                           0.86     83368
   macro avg       0.86      0.75      0.78     83368
weighted avg       0.86      0.86      0.85     83368
```

In [19]:

```
user_input = input("Enter a text: ")
user_input_vectorized = vectorizer.transform([user_input])
predicted_label = naive_bayes_model.predict(user_input_vectorized)
print(f"The predicted emotion label for the input is: {predicted_label[0]}")
```

```
Enter a text: Love All Serve All
The predicted emotion label for the input is: 1
```

In [14]:

```
#sadness (0), joy (1), love (2), anger (3), fear (4), and surprise (5)
```

In [ ]:

In [ ]: