In [1]:

```
!pip install scikit-learn pandas
```

```
Requirement already satisfied: scikit-learn in c:\users\dnhac\anaconda3\lib\site-packages (0.24.2)
Requirement already satisfied: pandas in c:\users\dnhac\anaconda3\lib\site-packages (1.3.4)
Requirement already satisfied: scipy>=0.19.1 in c:\users\dnhac\anaconda3\lib\site-packages (from scikit-learn) (1.10.1)
Requirement already satisfied: joblib>=0.11 in c:\users\dnhac\anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dnhac\anaconda3\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: numpy>=1.13.3 in c:\users\dnhac\anaconda3\lib\site-packages (from scikit-learn) (1.20.3)
Requirement already satisfied: pytz>=2017.3 in c:\users\dnhac\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\dnhac\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\dnhac\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas) (1.16.0)
```

In [5]:

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

# Load the dataset
df = pd.read_csv('training.1600000.processed.noemoticon.csv', encoding='latin-1', header=None)

# Rename the columns
df.columns = ['target', 'id', 'date', 'flag', 'user', 'text']

# Preprocess the dataset (optional)
# ...

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['text'], df['target'], test_size=0.2, random_state=42)

# Create a TF-IDF vectorizer
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)

# Transform the training set
tfidf_train = tfidf_vectorizer.fit_transform(X_train)

# Transform the testing set
tfidf_test = tfidf_vectorizer.transform(X_test)

# Create a Logistic Regression classifier
lr_classifier = LogisticRegression()

# Train the classifier
lr_classifier.fit(tfidf_train, y_train)

# Make predictions on the testing set
y_pred = lr_classifier.predict(tfidf_test)

# Evaluate the classifier
print(classification_report(y_test, y_pred))
```

```
C:\Users\dnhac\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
              precision    recall  f1-score   support

           0       0.79      0.76      0.78    159494
           4       0.77      0.80      0.79    160506

    accuracy                           0.78    320000
   macro avg       0.78      0.78      0.78    320000
weighted avg       0.78      0.78      0.78    320000
```

In [ ]:

In [ ]: