5. TWEETS CLASSIFICATION

EX.N0 : 5	**DEVELOP A CLASSIFIER SYSTEM FOR TWEETS CLASSIFICATION**
DATE : /0 /2025	

AIM:

To write a program to develop a classifier system for tweets classification.

ALGORITHM:

Step 1: Start

Step 2: Import necessary libraries (pandas, sklearn, nltk)

Step 3: Load and preprocess tweet dataset (text + label)

Step 4: Clean tweets (remove hashtags, mentions, punctuation, stopwords)

Step 5: Split the dataset into train/test sets

Step 6: Train a classifier and Evaluate using accuracy and confusion matrix

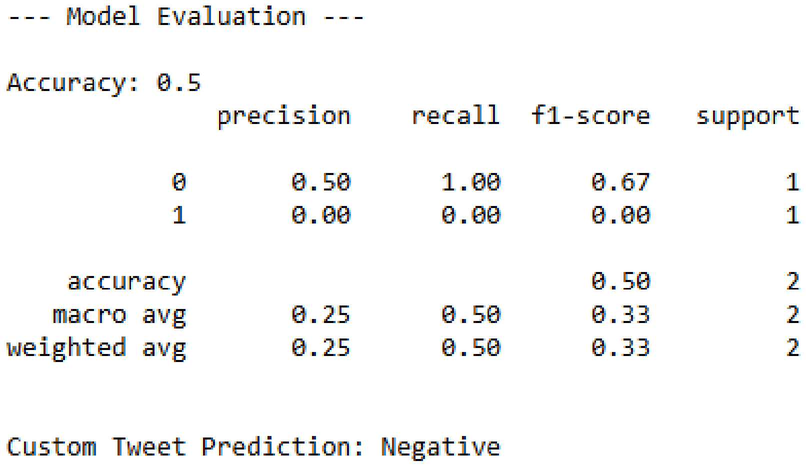
Step 7: Test on custom tweet

PROGRAM:

```
import pandas as pd
import re
import nltk
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
nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
```

```
def clean_tweet(tweet):
tweet = re.sub(r"http\S+|@\S+|#\S+|[^A-Za-z\s]", "", tweet)  # remove links, mentions, hashtags, special chars
tweet = tweet.lower().split()
tweet = [word for word in tweet if word not in stop_words]
return " ".join(tweet)
data = {
'tweet': [
"I love the new features of the iPhone!",
"This is the worst product I've ever used.",
"Absolutely fantastic performance!",
"I hate this phone, it's so slow.",
"Great service by the company.",
"Terrible experience, will not recommend." ],
'label': ['positive', 'negative', 'positive', 'negative', 'positive', 'negative'] }
df = pd.DataFrame(data)
df['cleaned'] = df['tweet'].apply(clean_tweet)
df['label'] = df['label'].map({'positive': 1, 'negative': 0})
X_train, X_test, y_train, y_test = train_test_split(df['cleaned'], df['label'], test_size=0.2, random_state=42)
vectorizer = TfidfVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
clf = LogisticRegression()
clf.fit(X_train_vec, y_train)
y_pred = clf.predict(X_test_vec)
print("\n--- Model Evaluation ---\n")
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
test_tweet = "I am really enjoying this amazing product!"
cleaned_test = clean_tweet(test_tweet)
test_vec = vectorizer.transform([cleaned_test])
result = clf.predict(test_vec)
print("\nCustom Tweet Prediction:", "Positive" if result[0] == 1 else "Negative")
```

OUTPUT:



```
--- Model Evaluation ---

Accuracy: 0.5
              precision    recall  f1-score   support

           0       0.50      1.00      0.67         1
           1       0.00      0.00      0.00         1

    accuracy                           0.50         2
   macro avg       0.25      0.50      0.33         2
weighted avg       0.25      0.50      0.33         2


Custom Tweet Prediction: Negative
```

RESULT:

Thus a program to develop a classifier system for tweets classification has been executed successfully.