SATHYABAMA INSTITUTE OF SCIENCE & TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCSA 2604 NATURAL LANGUAGE PROCESSING LAB

LAB 3: TEXT CLASSIFICATION

AIM: To perform Text classification using python and scikit-learn

PROCEDURE:

This algorithm outlines the steps involved in the text classification task using LinearSVC on the 20 Newsgroups dataset. It provides a structured approach to implementing the program and understanding the workflow.

ALGORITHM:

Algorithm: Text Classification using LinearSVC

1. Load the 20 Newsgroups dataset with specified categories.
   - Import the necessary libraries: fetch_20newsgroups from sklearn.datasets.
   - Specify the categories of interest for classification.
   - Use fetch_20newsgroups to load the dataset for both training and testing sets.

2. Split the dataset into training and testing sets.
   - Import train_test_split from sklearn.model_selection.
   - Split the dataset into X_train, X_test, y_train, and y_test.

3. Create a pipeline for text classification.
   - Import make_pipeline from sklearn.pipeline.
   - Create a pipeline with TF-IDF Vectorizer and LinearSVC classifier.

4. Train the model on the training data.

- Call the fit method on the pipeline with X_train and y_train as input.

5. Predict labels for the testing data.
   - Use the trained model to predict labels for X_test.

6. Evaluate the model's performance.
   - Calculate accuracy_score to measure the accuracy of the model.
   - Print classification_report to see precision, recall, and F1-score for each class.

End Algorithm

PROGRAM:

```
# Install scikit-learn if not already installed
!pip install scikit-learn

# Import necessary libraries
import pandas as pd

from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import make_pipeline
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report

# Load the 20 Newsgroups dataset
categories = ['sci.med', 'sci.space', 'comp.graphics', 'talk.politics.mideast']
newsgroups_train = fetch_20newsgroups(subset='train', categories=categories)
newsgroups_test = fetch_20newsgroups(subset='test', categories=categories)

# Split the data into training and testing sets
```

```
X_train = newsgroups_train.data
X_test = newsgroups_test.data
y_train = newsgroups_train.target
y_test = newsgroups_test.target

# Create a pipeline with TF-IDF vectorizer and LinearSVC classifier
model = make_pipeline(
    TfidfVectorizer(),
    LinearSVC()
)

# Train the model
model.fit(X_train, y_train)

# Predict labels for the test set
predictions = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, predictions)
print("Accuracy:", accuracy)
print("\nClassification Report:")
print(classification_report(y_test, predictions))
```

OUTPUT:

```
Requirement already satisfied: scikit-learn in
/usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.23.5)
Requirement already satisfied: scipy>=1.3.2 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.2.0)

Accuracy: 0.9504823151125402
```

Classification Report:

	precision	recall	f1-score	support
0	0.89	0.97	0.93	389
1	0.96	0.91	0.94	396
2	0.98	0.94	0.96	394
3	0.98	0.98	0.98	376
accuracy			0.95	1555
macro avg	0.95	0.95	0.95	1555
weighted avg	0.95	0.95	0.95	1555