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:

Here we use scikit-learn's Support Vector Machine (SVM) classifier with TF-IDF vectorization for text classification. It demonstrates a simple classification task using a small sample dataset, where text snippets are labeled as positive or negative.

ALGORITHM:

Install scikit-learn: Installs scikit-learn library if not already installed.

Import Libraries: Imports required libraries including pandas for data handling, scikit-learn for machine learning functionalities.

Dataset Creation: Creates a small sample dataset (you should replace this with your dataset).

Data Preprocessing: Converts data into a pandas DataFrame and splits it into train and test sets.

TF-IDF Vectorization: Uses TF-IDF vectorization to convert text data into numerical form.

Classifier Training: Initializes and trains a Support Vector Machine (SVM) classifier.

Prediction and Evaluation: Transforms test data, predicts labels, and calculates accuracy and classification report.

PROGRAM:

```
# Install scikit-learn if not already installed

!pip install scikit-learn


# Import necessary libraries

import pandas as pd
```

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

# Sample dataset (you can replace this with your dataset)
data = {
    'text': [
        'This is a positive sentence',
        'I am happy today',
        'Negative review, very bad service',
        'I do not like this product'
    ],
    'label': ['positive', 'positive', 'negative', 'negative']
}

# Convert data to DataFrame
df = pd.DataFrame(data)

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(df['text'], df['label'], test_size=0.2, random_state=42)

# Initialize TF-IDF vectorizer
tfidf_vectorizer = TfidfVectorizer()

# Fit and transform the training data
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)

# Initialize SVM classifier
svm_classifier = SVC(kernel='linear')
```

```
# Train the classifier
svm_classifier.fit(X_train_tfidf, y_train)

# Transform the test data
X_test_tfidf = tfidf_vectorizer.transform(X_test)

# Predict on the test data
y_pred = svm_classifier.predict(X_test_tfidf)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

# Display classification report
print(classification_report(y_test, y_pred))
```

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) (1.11.3)
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: 0.00
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00       0.0
    positive       0.00      0.00      0.00       1.0

    accuracy                           0.00       1.0
   macro avg       0.00      0.00      0.00       1.0
weighted avg       0.00      0.00      0.00       1.0
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```
