Vidyavardhini's College of Engineering and Technology

AY: 2025-26

Department of Artificial Intelligence & Data Science

Class:	**BE-AI&DS**	**Semester:**	**VII**
Course Code:	**CSDOL7011**	**Course Name:**	**NLP Lab**

Name of Student:	BARI ANKIT VINOD
Roll No. :	61
Experiment No.:	**9**
Title of the Experiment:	**Training and Evaluating a Text Classification Model Using Proper Experimental Methodology**
Date of Performance:	
Date of Submission:	

CSDOL7011: Natural Language Processing Lab

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Performance	4-5	2-3	1
Understanding	4-5	2-3	1
Journal work and timely submission	8-10	5-8	1-4

Checked by

Name of Faculty :

Signature :

Date :

Aim: To implement a text classification model and evaluate its performance using standard experimental procedures including data splitting, cross-validation, and evaluation metrics.

Objective: To build and evaluate a text classification model using standard machine learning methodology and evaluation metrics.

Tools Required:

1. Python (Jupyter Notebook or Google Colab)
2. scikit-learn
3. pandas, matplotlib
4. Dataset: SMS Spam Collection Dataset or any labeled text classification dataset

Procedure:

1. Import required libraries:
   a. import pandas as pd
   b. from sklearn.feature_extraction.text import TfidfVectorizer
   c. from sklearn.model_selection import train_test_split, cross_val_score
   d. from sklearn.naive_bayes import MultinomialNB
   e. from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
   f. import matplotlib.pyplot as plt
   g. import seaborn as sns

2. Load the dataset:
   a. For SMS Spam Dataset: Download from UCI ML Repository

3. Preprocess the text:

Lowercase conversion, stopword removal (optional), and TF-IDF feature extraction

4. Split the dataset:

Use train_test_split() to divide into training and testing sets (e.g., 80%-20%)

5. Train the model:

Use MultinomialNB() or LogisticRegression() classifier

6. Evaluate the model:
    a. Predict on test data
    b. Use the following evaluation metrics:
        i. Accuracy
        ii. Precision
        iii. Recall

iv. F1-Score
v. Confusion Matrix

7. (Optional): Perform 5-fold cross-validation and compare with hold-out evaluation.
8. Visualize results:

Plot confusion matrix using seaborn.heatmap()

Description of the Experiment:

This experiment teaches students how to build and evaluate a complete text classification system using real-world data. It emphasizes experimental methodology, including the importance of data splits, model evaluation, and fair performance comparison.

Detailed Description of the NLP Technique:

1. Text Classification:

The task of assigning a category or label to a given text (e.g., spam vs. ham). It's widely used in:

a. Spam filtering
b. Sentiment analysis
c. Topic categorization

2. Pipeline Stages:
   a. Text Preprocessing & Vectorization: Convert raw text into numeric features using TF-IDF.
   b. Model Training: Use supervised machine learning algorithms like Naive Bayes or SVM.
   c. Model Evaluation: Use appropriate metrics to evaluate the model's generalization on unseen data.

3. Evaluation Metrics:
    a. Accuracy: Ratio of correctly predicted instances
    b. Precision: True Positives / (True Positives + False Positives) Recall: True Positives / (True Positives + False Negatives) F1-Score: Harmonic mean of precision and recall
    c. Confusion Matrix: Shows TP, TN, FP, FN counts

4. Best Practices in Experimental Methodology:
    a. Train-test split ensures model evaluation on unseen data.
    b. Cross-validation helps in robust performance estimation.
    c. Random seed control improves reproducibility.

Conclusion:

The results show that the text classification model effectively learned to categorize text into predefined classes. Evaluation metrics such as Accuracy, Precision, Recall, and F1-score indicate that the model performs well on both training and test data. The experiment demonstrates that standard machine learning techniques, combined with proper feature extraction, can achieve reliable performance in text classification tasks.