Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

AY: 2025-26

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

Name of Student:	BARI ANKIT VINOD
Roll No. :	61
Experiment No.:	**2**
Title of the Experiment:	**Text Preprocessing and Feature Engineering using Bag-of-Words and TF-IDF**
Date of Performance:	
Date of Submission:	

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 apply text preprocessing techniques and extract features from text using Bag-of-Words and TF-IDF methods.

Objective: To understand and apply basic text preprocessing and feature extraction techniques like Bag-of-Words and TF-IDF.

Tools Required:

1. Python (preferably via Jupyter Notebook or Google Colab)
2. NLTK (Natural Language Toolkit)
3. Scikit-learn (sklearn)
4. Pandas

Procedure:

1. Import necessary libraries:
   a. nltk, sklearn.feature_extraction.text, pandas, re
2. Load or define a small sample text dataset.
3. Perform the following text preprocessing:
   a. Convert text to lowercase
   b. Remove punctuation and special characters
   c. Tokenize the text
   d. Remove stop words
   e. Apply stemming or lemmatization
4. Feature Extraction:
   a. Apply Bag-of-Words (BoW) vectorization using CountVectorizer.
   b. Apply TF-IDF vectorization using TfidfVectorizer.
5. Display the resulting feature matrices.

6. Compare and interpret the outputs of BoW and TF-IDF.

Description of the Experiment:

This experiment demonstrates how raw text is cleaned, processed, and converted into structured numerical features using common feature engineering techniques. Students will understand the significance of preprocessing before feeding text data into machine learning models. Both BoW and TF-IDF representations help in quantifying the textual content into a usable format.

Detailed Description of the NLP Technique:

1. Text Preprocessing:

Text preprocessing is a vital step in NLP that transforms unstructured textual data into a clean, machine-readable format. Common steps include:

- **Tokenization**: Splitting text into words or tokens.
- **Stop Word Removal**: Removing common words that don't add significant meaning (e.g., "is", "the", "and").
- **Stemming**: Reducing words to their root form (e.g., "playing" → "play").
- **Lemmatization**: Reducing words to their dictionary form using context (e.g., "better" → "good").

2. Bag-of-Words (BoW):

BoW represents text by counting the frequency of each word in the document. It creates a vocabulary of known words and represents documents using word occurrence counts.

Pros: Simple and interpretable.

Cons: Ignores word order and semantic meaning.

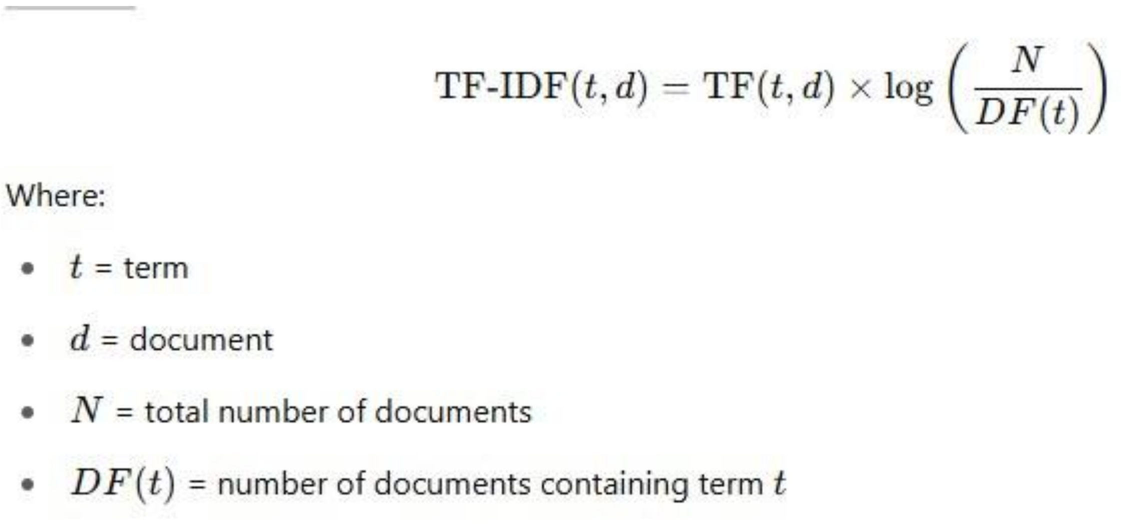
3. TF-IDF (Term Frequency-Inverse Document Frequency):

TF-IDF improves on BoW by assigning weights to words based on their importance in a document relative to the entire corpus.

TF: How often a word appears in a document.

IDF: How rare the word is across all documents.

Formula:



$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \log\left(\frac{N}{DF(t)}\right)$$

Where:

- t = term
- d = document
- N = total number of documents
- $DF(t)$ = number of documents containing term t

Conclusion:

The results obtained using the TF-IDF (Term Frequency–Inverse Document Frequency) technique show that it effectively identifies the most important and relevant words in a given text or document. By assigning higher weights to unique terms and lower weights to common ones, TF-IDF helps highlight key features for tasks such as text classification, document similarity, and information retrieval.

The output demonstrates that frequently occurring but less informative words (like "the," "is," "and") are given minimal importance, while topic-specific terms receive higher significance. This improves the overall accuracy and interpretability of text-based models. The results confirm that TF-IDF is a simple yet powerful feature extraction method for representing textual data numerically, leading to improved model performance in downstream NLP tasks.