Knowledge is Power

Dr. D. Y. PATIL
EDUCATIONAL FEDERATION
Varale Campus

ACCREDITED WITH GRADE
A
NAAC

Academic Year: 2025-26

LABORATORY MANUAL

Name of the Student:		
Class: BE	**Division:**	**Roll No.:**
Subject: Computer Laboratory II **(2019 Course) [417526]**		**Exam Seat No.:**
Department of Artificial Intelligence and Data Science		

Program Outcomes (PO's):

POs are statements that describe what students are expected to know and be able to do upon graduating from the program. These relate to the skills, knowledge, analytical ability attitude and behavior that students acquire through the program.

- **PO1: Engineering Knowledge:**
  Graduates will be able to apply the Knowledge of the mathematics, science and engineering fundamentals for the solution of engineering problems related to IT.

- **PO2: Problem Analysis:**
  Graduates will be able to carry out identification and formulation of the problem statement by requirement engineering and literature survey.

- **PO3: Design/Development of Solutions:**
  Graduates will be able to design a system, its components and/or processes to meet the required needs with consideration for public safety and social considerations.

- **PO4: Conduct Investigations of Complex Problems:**
  Graduates will be able to investigate the problems, categorize the problem according to their complexity using modern computational concepts and tools.

- **PO5: Modern Tool Usage:**
  Graduates will be able to use the techniques, skills, modern IT engineering tools necessary for engineering practice.

- **PO6: The Engineer and Society:**
  Graduates will be able to apply reasoning and knowledge to assess global and societal issues

- **PO7: Environment and Sustainability:**
  Graduates will be able to recognize the implications of engineering IT solution with respect to society and environment.

- **PO8: Ethics:**

  Graduates will be able to understand the professional and ethical responsibility.

- **PO9: Individual and Team Work:**

  Graduates will be able to function effectively as an individual member, team member or leader in multi -disciplinary teams.

- **PO10: Communication:**

  Graduates will be able to communicate effectively and make effective documentations and presentations.

- **PO11: Project Management and Finance:**

  Graduates will be able to apply and demonstrate engineering and management principles in project management as a member or leader.

- **PO12: Life-long Learning:**

  Graduates will be able to recognize the need for continuous learning and to engage in life- long learning.

Course Objectives and Course Outcomes (COs)

Course Objectives:

- Apply regression, classification and clustering algorithms for creation of ML models
- Introduce and integrate models in the form of advanced ensembles.
- Conceptualized representation of Data objects.
- Create associations between different data objects, and the rules.
- Organized data description, data semantics, and consistency constraints of data

Course Outcomes:

On completion of the course, students will be able to–

CO1: Implement regression, classification and clustering models

CO2: Integrate multiple machine learning algorithms in the form of ensemble learning.

CO3: Apply reinforcement learning and its algorithms for real world applications.

CO4: Analyze the characteristics, requirements of data and select an appropriate data model.

CO5: Apply data analysis and visualization techniques in the field of exploratory data science

CO6: Evaluate time series data.

Dr. D. Y. Patil Educational Federation's

Dr. D. Y. PATIL COLLEGE OF ENGINEERING &INNOVATION

Department of Artificial Intelligence and Data Science

Academic Year 2025-26

CERTIFICATE

This is to certify that Mr. /Ms. ____.

of Class BE - AI-DS, Roll No. Examination Seat No. __________________________.

has completed all the practical work in the Computer Laboratory - II [417526]

Satisfactorily, as prescribed by Savitribai Phule Pune University, Pune in the academic year 2025-26 (Term-I).

Place:

Date:

Course In-charge
Department of
AI-DS

HOD
Department of
AI-DS

Principal
DYPCOEI,
Varale

Index

Department of Artificial Intelligence and data science **Class:** BE

Experiment No: 1

Title: Write a program for pre-processing of a text document such as stop word removal, stemming.

Name of the Student: ____________________________________

Class: BE

Batch:

Date:

Mark: /10

Signature of the Course In-charge: ____________________

Signature of the HOD: _____________________________

Experiment No: 1

Aim: Write a program for pre-processing of a text document such as stop word removal, stemming.

Outcome:

1. Understand and implement basic text pre-processing techniques.
2. Remove stop words from a document.
3. Apply stemming to words using NLP libraries.

Hardware Requirement:

1. Processor: Intel i3/i5 or above
2. RAM: Minimum 4GB
3. Storage: 2 GB of free space
4. OS: Windows / Linux / macOS

Software Requirement:

1. Python 3.x
2. Jupyter Notebook / VS Code / PyCharm
3. NLTK (Natural Language Toolkit) library
4. Internet connection (for downloading NLTK corpora)

Theory:

Text Preprocessing in NLP – Detailed Description

Text pre-processing is the **first and most crucial step** in Natural Language Processing (NLP) and text mining. Raw text is often noisy and unstructured. To extract meaningful insights or build NLP models, the text must be **cleaned, normalized, and structured**.

1. Tokenization

Definition:
Tokenization is the process of breaking down a large chunk of text into **smaller units**, called **tokens**. These tokens can be words, characters, or sub words.

Types of Tokenization:

- **Word Tokenization**: Splits text into words.
  Example: “NLP is fun!” → ['NLP', 'is', 'fun', '!']
- **Sentence Tokenization**: Splits text into sentences.
  Example: “NLP is fun. It’s useful.” → ['NLP is fun.', 'It’s useful.']

Importance:

- Tokenization is the base step for other NLP operations like parsing, POS tagging, and named entity recognition.
- Helps in feature extraction for ML models.

2. Stop Word Removal

Definition:
Stop words are **commonly used words** in a language (like “the”, “is”, “in”, “and”, etc.) that are **filtered out** before processing because they usually don’t carry significant meaning.

Why Remove Stop Words?

- They appear frequently and add noise to the data.
- Reducing them decreases dimensionality and improves model efficiency.
- They rarely help in distinguishing documents for tasks like classification or clustering.

Caution:

- Not all NLP tasks benefit from removing stop words. For example, **sentiment analysis** might require words like “not”, “very”, etc., which are sometimes considered stop words.

3. Stemming

Definition:
Stemming is the process of reducing a word to its **base/root form** (also known as the "stem") by removing prefixes or suffixes.

Example:

- "running", "runs", "ran" → "run"
- "studies", "studied" → "studi" (not always a real word)

Popular Stemming Algorithms:

- **Porter Stemmer**: Commonly used, rule-based stemmer.
- **Snowball Stemmer**: An improvement over the Porter stemmer.
- **Lancaster Stemmer**: More aggressive than Porter.

Usefulness:

- Helps in grouping similar words together, which improves accuracy in tasks like document classification and clustering.

Limitation:

- Stemming may not produce real words, and may affect readability and interpretation.
- It's **language-dependent** and may not work equally well across different languages.

Why Text Preprocessing Matters:

- Reduces **noise** in data.
- Converts raw text into a form suitable for NLP models.
- Helps in **normalizing** textual input.
- Improves **accuracy and speed** of NLP algorithms.
- Essential for applications like chat bots, search engines, recommendation systems, and sentiment analysis.

Advantages:

- Reduces the size of the data for processing.
- Helps focus on meaningful words.
- Improves model performance by reducing noise.
- Reduces dimensionality of features.

Disadvantages:

- Stemming may result in non-dictionary words (e.g., "studying" → "studi").
- Stop word lists may vary and could remove meaningful words in some contexts.
- Language-dependent — each language needs its own pre-processing.

Conclusion: Hence we conclude, Text pre-processing is a critical step in NLP. Techniques like stop word removal and stemming help in cleaning and standardizing the data, making it ready for machine learning models or further linguistic analysis.

Questions:

Q1. What is the purpose of stop word removal?

Q2. Define stemming and give an example?

Q3. What is the importance of tokenization in NLP?

Q4. What are the limitations of stemming?

Coding Efficiency	**Viva**	**Timely Completion**	**Total**	**Dated Sign of Course In-charge**
5	**3**	**2**	**10**	

Experiment No: 2

Title: Implement a program for retrieval of documents using inverted files.

Name of the Student: ______________________________________

Class: **BE**

Batch:

Date:

Mark: /10

Signature of the Course In-charge: ____________________

Signature of the HOD: ______________________________

Experiment No: 2

Aim: To implement a program that retrieves documents based on keyword queries using the **inverted file indexing technique**.

Outcome:

1. Understand and implement **inverted indexing**.
2. Store and retrieve documents efficiently based on keywords.
3. Apply the concept in **search engines** and **Information Retrieval Systems**.

Hardware Requirement:

1. Processor: Intel i3/i5/i7 or equivalent
2. RAM: Minimum 4 GB
3. Disk Space: Minimum 1 GB free
4. Keyboard & Monitor

Software Requirement:

1. Operating System: Windows/Linux/MacOS
2. Programming Language: Python 3.6 or above
3. IDE: VS Code / PyCharm / Jupyter Notebook
4. Libraries: No external library required (basic Python I/O and dictionary structures)

Theory:

Inverted file indexing is a method of storing a mapping from content words (terms) to their locations in a set of documents. It is the backbone of **modern search engines**.

Steps involved:

1. **Tokenization** – Split documents into individual words.
2. **Stop Word Removal** – Remove common words (optional).
3. **Inverted Index Construction** – Map each word to the document IDs in which it occurs.
4. **Query Processing** – Accept a query and return documents containing the query term(s).

Example:
Documents:

- D1: "data science is powerful"
- D2: "machine learning is part of data science" Inverted Index:
- "data": [D1, D2]
- "science": [D1, D2]
- "machine": [D2]
- "learning": [D2]
- "powerful": [D1]

Conclusion: Hence we conclude, The inverted file indexing method was successfully implemented. It demonstrated how: Words can be efficiently mapped to documents, Queries can be quickly resolved to return relevant documents, The basic mechanism behind search engines works using **inverted indexes**. This approach can be extended to include **stop word removal**, **stemming**, **ranking**, and **Boolean queries** for more advanced Information Retrieval tasks.

Questions:

Q1. What is an inverted index?

Q2. What are the steps in building an inverted index?

Q3. Why do we use inverted indexing?

Q4. What data structure is used to implement an inverted index in Python?

Coding Efficiency	**Viva**	**Timely Completion**	**Total**	**Dated Sign of Course In-charge**
5	**3**	**2**	**10**	

Experiment No: 3

Title: Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using the standard Heart Disease Data Set (You can use Java/Python ML library classes/API**.**

Class: BE **Batch:**

Date:

Mark: /10

Signature of the Course In-charge: ____________________

Signature of the HOD: ______________________________

Experiment No: 3

Aim: To construct a **Bayesian Network** using medical data and use it to diagnose heart disease using the **UCI Heart Disease Dataset**.

Outcome:

1. Understand and implement a **Bayesian Network** using Python.
2. Learn how to **model uncertain knowledge** using probabilistic graphical models.
3. Use the Bayesian model for **predictive diagnosis** of heart patients

Hardware Requirement:

1. CPU: Intel i3/i5/i7 or AMD Ryzen
2. RAM: Minimum 4 GB
3. Storage: Minimum 1 GB free disk space
4. Input/Output Devices: Keyboard, Mouse, Monitor

Software Requirement:

1. Operating System: Windows/Linux/macOS
2. Python Version: 3.6 or above
3. Libraries:

    - pandas
    - pgmpy
    - numpy
    - scikit-learn

4. IDE: VS Code, Jupyter Notebook, or PyCharm

Theory:

A **Bayesian Network** is a **probabilistic graphical model** that represents a set of variables and their conditional dependencies via a **directed acyclic graph (DAG)**.

- **Nodes** represent variables (features like age, cholesterol, etc.).

- **Edges** represent conditional dependencies.
- **CPT (Conditional Probability Table)** is used for probability computation.
- Suitable for domains where **uncertainty** is involved (e.g., medical diagnosis).

Heart Disease Dataset (UCI):
Common features:

- Age
- Gender
- Chest Pain Type
- Blood Pressure
- Cholesterol
- Fasting Blood Sugar
- RestECG
- Thalassemia
- Heart Disease (target variable)

Conclusion: Hence we conclude, we successfully constructed a **Bayesian Network** using pgmpy. We learned to model **conditional dependencies** between symptoms and heart disease. Using **inference**, we predicted the **probability of a patient having heart disease** given evidence. Such systems can be used in **decision support** for doctors in hospitals.

Questions:

Q1. What is a Bayesian Network?

Q2. What is a Conditional Probability Table (CPT)?

Q3. Why do we use Bayesian Networks in medicine?

Q4. What is the purpose of the pgmpy library?

Coding Efficiency	**Viva**	**Timely Completion**	**Total**	**Dated Sign of Course In-charge**

5	3	2	10	

Experiment No: 4

Title: Implement e-mail spam filtering using text classification algorithm with appropriate dataset.

Name of the Student: ________________________________

Class: BE **Batch:**

Date:

Mark: /10

Signature of the Course In-charge: ____________________

Signature of the HOD: ______________________________

Experiment No: 4

Aim: To implement an **email spam filter** using a **text classification algorithm** (Naive Bayes) with a suitable dataset.

Outcome:

1. Understand how to process text data for classification.
2. Apply machine learning algorithms for **spam detection**.
3. Train and evaluate a model to **classify emails as spam or ham (not spam)**

Hardware Requirement:

- Processor: Intel i3 or above / AMD Ryzen
- RAM: 4 GB minimum
- Disk: 1 GB of free storage
- I/O Devices: Keyboard, Monitor

Software Requirement:

1. Operating System: Windows / Linux / macOS
2. Python 3.6 or above
3. Libraries:
    - `pandas`
    - `scikit-learn`
    - `nltk`
4. IDE: Jupyter Notebook / PyCharm / VS Code

Theory:

Spam filtering is the process of identifying and classifying unsolicited and unwanted email messages (spam).
Text classification algorithms, like **Naive Bayes**, are used to automatically classify emails based on content.

Common Steps:

1. **Preprocessing**: Clean email text (remove symbols, lowercasing, remove stop words).
2. **Vectorization**: Convert text to numerical data using methods like **TF-IDF** or **Bag-of-Words**.
3. **Model Training**: Use a classifier (e.g., Naive Bayes) to learn from labeled data.
4. **Prediction**: Classify new/unseen emails as **spam** or **ham**.

Conclusion: hence we conclusion, The Naive Bayes model was able to classify spam messages with **high accuracy**. Pre-processing, TF-IDF, and splitting data correctly were key steps. This type of spam filter is used in **Gmail, Outlook**, and other email systems.

Questions:

Q1. What is spam filtering?

Q2. Which algorithm did you use and why?

Q3. What is TF-IDF?

Q4. What are the labels used in the dataset?

Coding Efficiency	**Viva**	**Timely Completion**	**Total**	**Dated Sign of Course In-charge**
5	**3**	**2**	**10**	

Experiment No: 5

Title: Implement Page Rank Algorithm. (Use python or beautiful soup for implementation).

Name of the Student: ________________________________

Class: **BE**

Batch:

Date:

Mark: /10

Signature of the Course In-charge: ____________________

Signature of the HOD: ______________________________

Experiment No: 5

Aim: To implement the **PageRank algorithm** using Python (and optionally Beautiful Soup for link extraction from webpages).

Outcome:

1. Understand the concept of **PageRank** and how it evaluates the importance of web pages.
2. Implement PageRank using **iterative algorithm**.
3. Optionally, use **Beautiful Soup** to extract links from a web page (mini web crawler).
4. Simulate search engine ranking mechanism.

Hardware Requirement:

1. Processor: Intel i3/i5/i7 or AMD Ryzen
2. RAM: Minimum 4 GB
3. Storage: Minimum 1 GB
4. Keyboard, Monitor

Software Requirement:

1. Operating System: Windows / Linux / macOS
2. Python 3.x
3. Libraries:
    - numpy
    - requests (optional)
    - beautifulsoup4 (optional, for HTML parsing)
4. IDE: Jupyter Notebook / VS Code / PyCharm

Theory:

PageRank is an algorithm developed by **Larry Page and Sergey Brin** (founders of Google) to rank websites in their search engine results.

- A page is important if **many other important pages link to it**.
- PageRank works like a **voting system**.
- Pages distribute their rank value to the pages they link to.

PageRank Formula:
PR(A)=1−dN+d(∑i∈In(A)PR(i)L(i))PR(A) = \frac{1 - d}{N} + d \left(\sum_{i \in In(A)} \frac{PR(i)}{L(i)} \right)PR(A)=N1−d+di∈In(A)∑L(i)PR(i)

Where:

- ddd = damping factor (usually 0.85)
- NNN = total number of pages
- In(A)In(A)In(A) = pages linking to A
- L(i)L(i)L(i) = number of outgoing links from page i

Conclusion: Hence, we conclude, The PageRank algorithm was successfully implemented. The rank of each page was calculated based on incoming links. This algorithm simulates how **search engines rank web pages**. Optional use of Beautiful Soup allows extraction of real webpage links for graph building.

Questions:

Q1. What is PageRank?

Q2. Who developed the PageRank algorithm?

Q3. What is the damping factor in PageRank?

Q4. What is the use of Beautiful Soup in this practical?

Coding Efficiency	**Viva**	**Timely Completion**	**Total**	**Dated Sign of Course In-charge**
5	**3**	**2**	**10**	