(DEEMED TO BE UNIVERSITY)

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

23AD2001O

STUDENT ID:

STUDENT NAME:

ACADEMIC YEAR: 2024-25

ODD SEMESTER

K L E F

KONERU LAKSHMAIAH EDUCATION FOUNDATION

(Deemed to be university estd, u/s, 3 of the UGC Act, 1956)

(NAAC Accredited "A" Grade University)

Vision

To be a globally renowned university.

Mission:

To impart quality higher education and to undertake research and extension with emphasis on application and innovation that cater to the emerging societal needs through all-round development of the students of all sections enabling them to be globally competitive and socially responsible citizens with intrinsic values.

Table of Contents

A.Y. 2024-25 LAB CONTINUOUS EVALUATION

S.No.	Date	Experiment Name	Pre-Lab (10M)	In-Lab (25M)			Post-Lab (10M)	Viva Voce (5M)	Total (50M)	Faculty Signature
				Program/ Procedure (5M)	Data and Results (10M)	Analysis & Inference (10M)				
1.		Implementation of Basic Python Programs								
2.		Implement a Random Movement Reflex Agent								
3.		Find the path to reach the target from a source node given a Graph								
4.		Solve 15 Puzzle problems using a Priority Queue								
5.		Implement stochastic Hill climbing Algorithm for Robot path finding								
6.		Implement Mini-Max algorithm to solve given graph								
7.		Implement a Cryptographic Arithmetic Problem								
8.		Implement various Data preprocessing techniques on a given data set								
9.		Predict the price of a House based on a set of characteristics using linear regression								
10.		Implement the Naïve Bayes classifier using the play tennis dataset to predict the label for unknown data								

S.No.	Date	Experiment Name	Pre-Lab (10M)	In-Lab (25M)			Post-Lab (10M)	Viva Voce (5M)	Total (50M)	Faculty Signature
				Program/ Procedure (5M)	Data and Results (10M)	Analysis & Inference (10M)				
11.		Perform clustering in terms of purchasing behaviour, demographics, or other relevant attributes using K-Means clustering algorithm								
12.		Implement 3 input AND &OR Boolean functions using the McCulloch-Pitts Model using Python								
13.		Implement an ANN to predict the Customer churn								

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment #1: Implementation of Basic Python Programs

Aim/Objective:

Implementation of Basic Python Programs.

Description:

Students will learn and understand the basic concepts in Python programming language.

Pre-Requisites:

- Basic Computer Skills
- Basic Mathematics
- Logical Thinking
- Text Editor or Integrated Development Environment (IDE)

Pre-Lab:

1. What is the purpose of implementing basic python programs in this lab?

2. What is python, and how do you comment in Python?

3. What are variables in Python? How do you print output in Python?

4. How do you take user input in Python? What are data types in Python?

5. How do you create a function in Python? How do you define a conditional statement in Python?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

In-Lab:

Implementation of Basic Python Programs.

1. Write a python Program to Find Largest of 3 Numbers using nested if-else
2. Write a python Program to Swap Two Variables using Third Variable
3. Write a python Program to find FACTORIAL of a given number.
4. Write a python Program to find the PRIME NUMBERS in the given range
5. Write a python Program to Print Fibonacci Series up to N Term
6. Create a list of integers and *append, insert*, and *remove* elements from the list. Access elements using *indexing* and *slicing*
7. Perform the list methods of *append(), extend(), insert(), remove(), pop(), clear(), index(), count(), sort(), reverse(), copy()* on a sample list and observe the changes.
8. Create a tuple of integers and Access elements using indexing and slicing.
9. Create a dictionary with tuples as keys and access and modify the dictionary using these keys.
10. Define a function that takes your name as input and returns a greeting message

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Sample VIVA-VOCE Questions (In-Lab):

1. What is the difference between a list and a tuple in Python?

2. How does Python handle memory management?

3. Explain the purpose of the self-parameter in Python classes.

4. Describe how you can handle exceptions in Python. Provide an example.

5. What are decorators in Python and how are they used? Provide an example.

Post-Lab:

Implementation of Advanced Python Programs.

1. Write a function that takes two numbers and returns their sum, difference, product, and quotient.
2. Implement a *recursive function* to calculate the factorial of a number.
3. Define a simple class **Person** with attributes *name* and *age*. Create **instances** of the Person class and print their attributes. Add a method *greet* to the Person class that prints a greeting message. Call the *greet* method on different instances and observe the output.
4. Create a subclass **Student** that inherits from the **Person** class. Add a new attribute *student_id* and a method *study* to the Student class. Create instances of **Student** and call methods from both **Person** and **Student.**

- **Procedure/Program:**

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date

Experiment #2: Implement a Random Movement Reflex Agent.

Aim/Objective:

Implement a simple reflex agent for a vacuum cleaner.

Description:

Students will create a simple reflex agent to clean a grid-based environment. The agent will perceive the status of the current cell and decide whether to clean, move, or do nothing.

Pre-Requisites:

The simplicity of the agent's decision-making process makes it an introductory exercise in artificial intelligence, allowing students to understand the concept of reflex agents and their application in autonomous systems.

Pre-Lab:

1. What is the purpose of implementing a reflex agent for a vacuum cleaner in this lab?

2. What is a reflex agent, and how does it differ from other types of intelligent agents?

3. Describe the grid-based environment in which the vacuum cleaner agent will operate. How is it structured?

4. What are the possible states that a cell in the grid-based environment can have? How are these states represented?

5. What are the available actions that the vacuum cleaner agent can take in response to the current cell's status?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

In-Lab:

The environment consists of a grid of squares(A,B), each of which can be either clean or dirty. The vacuum cleaner can perform the following actions:

- **Move Left:** Move one square to the left (if not at the leftmost edge).
- **Move Right:** Move one square to the right (if not at the rightmost edge).
- **Move Up:** Move one square up (if not at the top edge).
- **Move Down:** Move one square down (if not at the bottom edge).
- **Suck:** Clean the current square if it is dirty.

Write a program to create a vacuum cleaner agent that can clean all dirty squares in the grid efficiently. The agent should be able to sense its current position and the status (clean or dirty) of the square it occupies.

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Sample VIVA-VOCE Questions (In-Lab):

1. How would you define a reflex agent in the context of artificial intelligence?
2. Explain the logic behind implementing random movement for a reflex agent.
3. What are the possible actions that the agent can take in response to the current cell's status?
4. How do you ensure that the agent's movements are indeed random and not following a predictable pattern?
5. How does the agent decide whether to clean, move, or do nothing based on the current cell's status?
6. What are the potential challenges you might face when implementing a random movement reflex agent, and how would you address them?

Post-Lab:

create a vacuum cleaner agent with three grid of squares (A,B and C) environment.

a) Procedure/Program:

b) Data and Results:

c) Analysis and Inferences:

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment # 3: Find the path to reach the target from a source node in given Graph

Aim/Objective:

Implementing BFS for finding the path from a source to the goal node in the graph.

Description:

Students will create a simple graph and traverse the graph using Breadth-first search. BFS is a graph traversal algorithm that starts traversing the graph from the root node and explores all the neighbor nodes. Then, it selects the nearest node and explores all the unexplored nodes.

Pre-Requisites:

a) Basic understanding of search algorithms.
b) Familiarity with Python programming language.
c) Knowledge of Graph traversing.

Pre-Lab:

1. What is Breadth-First Search (BFS)? Explain the basic idea behind BFS and how it explores a graph or a maze.

2. What data structure(s) are commonly used in BFS? Describe their purpose and how they help in implementing the algorithm efficiently.

3. How can a grid-based maze be represented in a data structure? Discuss the options and explain which one you would choose for this implementation.

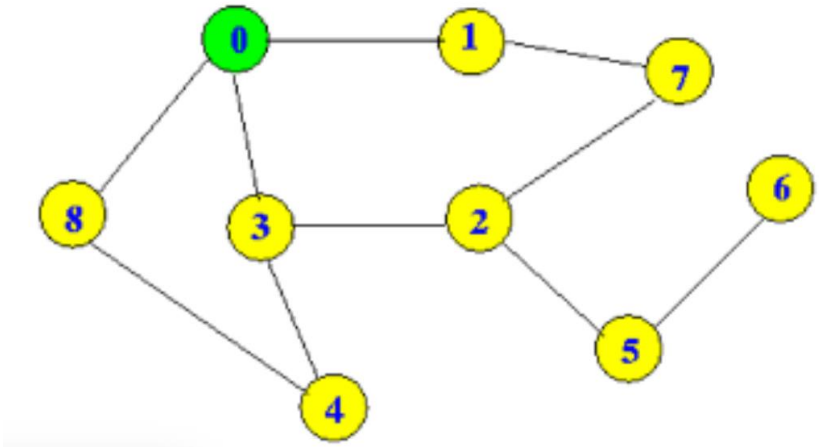
4. What are the steps involved in implementing BFS to solve a maze problem? Provide a high-level overview of the algorithm.

5. What is the purpose of marking visited cells in BFS? How can you keep track of visited cells efficiently during the traversal process?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

In-Lab:

Implement Breadth-First Search using Python and find a path from *node 0* to *node 6* for the graph below using BFS.



Procedure/Program:

- **Data and Results:**

- **Analysis and Inferences:**

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

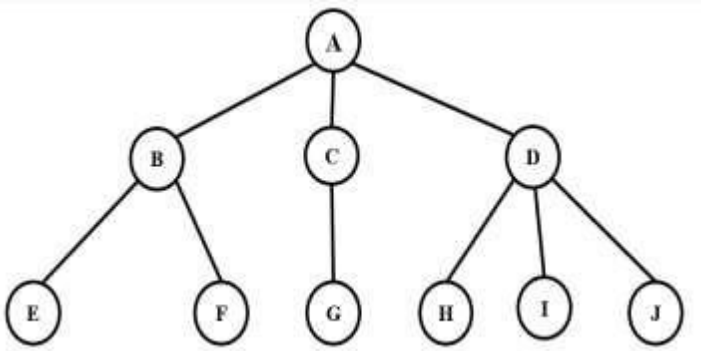
Sample VIVA-VOCE Questions (In-Lab):

1. What are the different algorithms available for finding the shortest path in a graph, and how do they differ?
2. Explain the working of Dijkstra's algorithm and discuss its time complexity.
3. How does the Breadth-First Search (BFS) algorithm work for finding the shortest path in an unweighted graph?
4. What are the advantages and limitations of using Depth-First Search (DFS) for pathfinding in a graph?
5. Describe how you would implement the A search algorithm for pathfinding in a graph, and explain the role of the heuristic function. *

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Post-Lab:

Implement DFS for the given tree.



- **Procedure/Program:**

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

- **Data and Results:**

- **Analysis and Inferences:**

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date

Experiment # 4: 15 Puzzle problem using a Priority Queue

Aim/Objective:

Implement the A* search algorithm to solve the 15-puzzle problem.

Description:

Students will implement the A* search algorithm using heuristics such as the *Manhattan distance* to estimate the cost of reaching the goal state from each possible move. By considering both the cost incurred so far and the estimated cost to reach the goal, the A* search algorithm intelligently explores state space to find an optimal solution to the 15-puzzle problem.

Pre-Requisites:

- Basic understanding of search algorithms.
- Familiarity with Python programming language.
- Knowledge of the 8-puzzle problem and its rules.

Pre-Lab:

1. How does the A* algorithm utilize a priority queue to solve the 15 Puzzle problem?

2. What is the role of the heuristic function in the A* algorithm for the 15 Puzzle problem, and how does it affect the priority queue?

3. What is the A* search algorithm, and how does it work?

4. What is a heuristic function, and why is it important in A* search?

5. What is the Manhattan distance heuristic, and how is it calculated in the context of the 8-puzzle problem?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

In-Lab:

Implement the A* search algorithm using a priority queue to solve a 15 puzzle problem.

Description: The 15-puzzle problem involves a 4x4 grid with 15 numbered tiles and one empty space. The goal is to rearrange the tiles from a given initial state to a desired goal state by sliding the tiles into the empty space. The A* search algorithm is used to find an optimal solution by considering both the cost incurred so far and the estimated cost to reach the goal state using a heuristic function.

Procedure/Program:

- **Data and Results:**

- **Analysis and Inferences:**

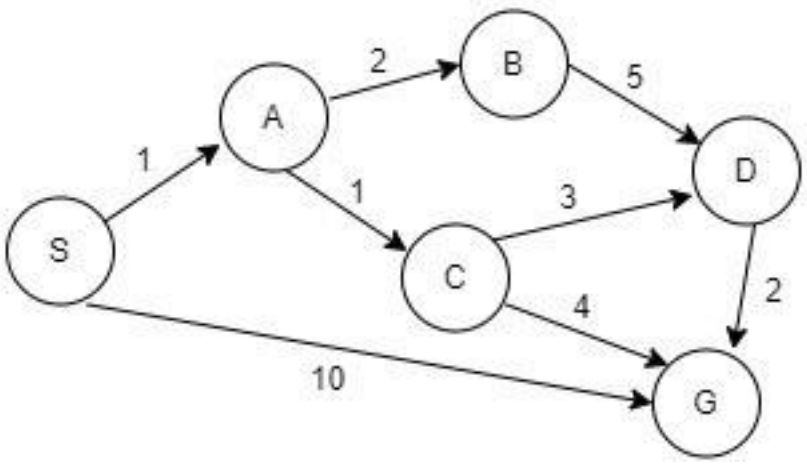
Sample VIVA-VOCE Questions (In-Lab):

1. What is the purpose of using the A* search algorithm in the 8-puzzle problem?
2. Explain the role of the heuristic function in the A* search algorithm.
3. How is the Manhattan distance calculated in the context of the 8-puzzle problem?
4. What are some advantages and limitations of the A* search algorithm?
5. Can the A* search algorithm guarantee finding an optimal solution in all cases?

Post-Lab:

Experiment: Implement the A* search algorithm for the following graph.

State	h(n)
S	5
A	3
B	4
C	2
D	6
G	0



Procedure/ Program:

- **Data and Results:**

- **Analysis and Inferences:**

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment # 5: Implement stochastic Hill climbing Algorithm for Robot path finding

Aim/Objective:

Implementing a stochastic Hill climbing Algorithm for Robot path finding

Description:
Students will learn the Stochastic Hill Climbing (SHC) which is a local search algorithm commonly used in optimization problems, including pathfinding for robots in dynamic environments.

Pre-Requisites:

- Students 'll be well-equipped to understand, implement, and customize the Stochastic Hill Climbing algorithm for robot pathfinding effectively.
- Familiarity with programming concepts and basic knowledge of Python programming language

Pre-Lab:

1. What is Stochastic Hill Climbing?

2. How does Stochastic Hill Climbing differ from regular Hill Climbing?

3. What are the advantages of using Stochastic Hill Climbing for robot pathfinding?

4. What are the key components required to implement Stochastic Hill Climbing for robot pathfinding?

In-Lab:

A robot is located in a grid of size nXm. The robot can move up, down, left, or right, but cannot move through obstacles. The goal is to find a path from the start position to the goal position using a stochastic hill climbing algorithm.

Stochastic Hill Climbing:

1. **Initial State**: Start at the initial position.
2. **Neighbor Selection**: Randomly choose one of the neighboring states.
3. **Evaluation**: Move to the chosen neighboring state if it improves the evaluation function.
4. **Termination**: Stop when the goal is reached or no better neighbors are found.

Evaluation Function

The evaluation function for the hill climbing algorithm can be the Manhattan distance to the goal: Manhattan Distance=$|x1-x2|+|y1-y2|$ where (x1,y1)(x2,y2) are the coordinates of the current position and are the coordinates of the goal.

Procedure/Program:

- **Data and Results:**

- **Analysis and Inferences:**

VIVA-VOCE Questions (In-Lab):

1. What is the stochastic hill climbing algorithm, and how does it differ from deterministic hill climbing?
2. How do you ensure that the robot finds an optimal path using the stochastic hill climbing algorithm?
3. What are the advantages and potential drawbacks of using stochastic hill climbing for pathfinding in a robotic environment?
4. How do you handle local maxima, plateaus, and ridges in the context of the hill climbing algorithm?
5. Can you explain a scenario where stochastic hill climbing would be more effective than other pathfinding algorithms?

Post-Lab:

Place N queens on an N×N chessboard so that no two queens threaten each other by starting with a random configuration and use stochastic hill climbing to move a queen to a new position if it reduces the number of conflicts.

Description: Stochastic hill climbing is a variant of the hill climbing algorithm used in optimization problems. It introduces randomness into the selection of the next move, which can help avoid local maxima. In this problem move a queen to a new position so that it reduces the number of conflicts.

- **Procedure/Program:**

- **Data and Results:**

- **Analysis and Inferences:**

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date

Experiment # 6: Implementing Min-Max algorithm to solve given graph

Aim/Objective:

Implement Min-Max algorithm to solve given graph.

Description:

The aim of this lab Experiment is to implement the Mini-Max algorithm for solving the Tic-Tac-Toe game, we'll create a program that allows the computer to play optimally against a human player. Students will learn how to play optimally using the Mini-Max algorithm, ensuring a challenging opponent for the human player in the game of Tic-Tac-Toe.

Pre-Requisites:

- Basic understanding of machine learning concepts, including neural networks and classification algorithms.
- Familiarity with programming concepts and basic knowledge of Python programming language

Pre-Lab:

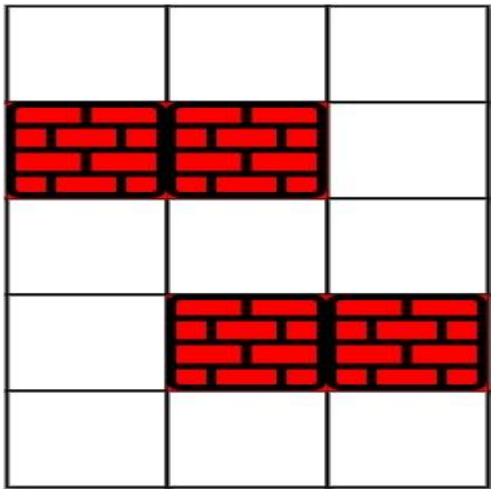
- Describe the rules of Tic-Tac-Toe. What are the win conditions and how is a draw determined?

- What is the role of recursion in the Mini-Max algorithm? Provide an example of how recursion is used in the algorithm.

- Define what utility function means in the context of game playing algorithms.

- Describe how you would represent the Tic-Tac-Toe board and manage game states within your implementation.

In-Lab:



Description: Students will implement Min-Max algorithm to solve a given problem.
You are given an m x n integer matrix grid where each cell is either 0 (empty) or 1 (obstacle). You can move up, down, left, or right from and to an empty cell in one step. Return *the minimum number of steps to walk from the upper left corner* (0, 0) *to the lower right corner* (m - 1, n - 1) *given that you can eliminate at most* k *obstacles*. If it is not possible to find such walk return -1.

Input: grid = [[0,0,0],[1,1,0],[0,0,0],[0,1,1],[0,0,0]], k = 1
Output: 6

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

VIVA-VOCE Questions (In-Lab):

1. How does the Mini-Max algorithm work in the context of a Tic-Tac-Toe game?
2. What is the purpose of the evaluation function in the Mini-Max algorithm?
3. How do you implement alpha-beta pruning to optimize the Mini-Max algorithm?
4. What are the main components of a cyber-security system for incident response?
5. How do you prioritize incidents and determine the appropriate response in a cyber-security system?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Post-Lab:

Implement a Cyber Security System for Optimizing defense strategies in network security games where an attacker and defender have conflicting objectives. Use the Minimax algorithm to model and solve the interaction between attacker and defender as a strategic game on a network graph

Description: The Min-Max algorithm is a recursive algorithm used in decision-making and game theory. It is designed for two-player zero-sum games, where one player maximizes their score (often referred to as the maximizing player, typically the AI), and the other player minimizes the score (the minimizing player, often the human opponent).

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date

Experiment # 7: Implement a Cryptographic Arithmetic Problem

Aim/Objective:

Implement a Cryptographic Arithmetic Problem

Description:

The aim of this lab experiment is to implement Cryptographic arithmetic problems typically involve operations on large integers where the details of the numbers are obscured or hidden through encryption techniques. These problems are often used in cryptography for secure communication, digital signatures, and other cryptographic protocols.

Pre-Requisites:

a. To effectively understand and work with cryptographic arithmetic, especially in the context of cryptographic protocols like RSA, several prerequisites are essential. These prerequisites span mathematical concepts, computer science fundamentals, and specific knowledge related to cryptography.
b. Familiarity with programming concepts and basic knowledge of Python programminglanguage

Pre-Lab:

- Define Constraint Satisfaction Problem (CSP)?

- Give an example of a cryptographic arithmetic problem that can be modeled as a CSP.

- What are the advantages of using CSPs in cryptography?

- How can CSPs contribute to the security and efficiency of cryptographic protocols?

In-Lab:

Implement a Cryptographic Arithmetic Problem to solve the following equation

SEND + MORE = MONEY

AIM: Students will be able to apply constraint satisfaction problem to solve Cryptographic Arithmetic Problem.

Description: Crypto-arithmetic problems, also known as verbal arithmetic or alphametic puzzles, are a type of constraint satisfaction problem (CSP) where digits are assigned to letters to satisfy a given arithmetic equation.

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

VIVA-VOCE Questions (In-Lab):

1. What are the basic principles of cryptographic arithmetic, and how do they ensure data security?
2. How do you implement basic cryptographic operations such as encryption and decryption using arithmetic techniques?
3. Explain the n-Queen problem and the role of forward-checking in solving it.
4. What are the advantages of using forward-checking in constraint satisfaction problems like the n-Queen problem?
5. How do you determine the efficiency of your cryptographic arithmetic implementation?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Post-Lab:

Implement an n-Queen problem using forward-checking

Description: Implementing the n-Queens problem using forward-checking involves using constraint satisfaction techniques to find all possible placements of n queens on n×n chessboard such that no two queens threaten each other. Forward-checking is a technique that reduces the domain of variables (in this case, potential queen positions) by considering the constraints imposed by already placed queens.

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment # 8: Implement various Data preprocessing techniques on a given data set

Aim/Objective:

This experiment aims to implement data pre-processing techniques to clean, transform, and prepare raw data for further analysis or machine learning tasks

Description:

In this experiment, students will learn the importance of data pre-processing in the data science workflow. They will understand the various steps involved in cleaning and transforming raw data to make it suitable for analysis or model building. Students will implement a data pre-processing pipeline using Python and relevant libraries, gaining hands-on experience in handling missing values, outliers, categorical variables, feature scaling, and more.

Pre-Requisites:

Basic understanding of data types, including numerical and categorical variables.

Familiarity with Python programming and data manipulation libraries such as pandas

Pre-Lab:

1. Why data are dirty?
2. What is data preprocessing? Why is it important in machine learning?
3. What are some common problems that occur during data processing? How can they be fixed?
4. How do you handle the missing data?
5. What is the difference between missing value treatment and outliers treatment?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

In-Lab:

Implement a Python program to find and impute the missing data in the following dataset.

Dataset Link:

https://www.kaggle.com/datasets/bharatnatrayn/movies-dataset-for-feature-extracion-prediction/data?select=movies.csv

Procedure/Program:

- **Data and Results:**

- **Analysis and Inferences:**

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

VIVA-VOCE Questions (In-Lab):

1. What is the difference between normalization and standardization?
2. What are the different encoding techniques for categorical data?
3. What are some common techniques for data reduction?
4. How do you preprocess time-series data?
5. What is data integration and what challenges are associated with it?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Post-Lab:

Implement a Python program to apply various data preprocessing techniques on the following dataset.
Dataset Link:

https://catalog.data.gov/dataset/electric-vehicle-population-data/resource/fa51be35-691f-45d2-9f3e-535877965e69

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date

Experiment # 9: Implement the Naïve Bayes classifier using the Iris dataset to predict the label for unknown data

Aim/Objective: Implementing the Naïve Bayes classifier using the Iris dataset to predict the label for unknown data

Description: Students will implement the Naïve Bayes classifier works on Bayes' theorem of probability to predict the class of unknown data sets. This approach is based on the assumption that the features of the input data are conditionally independent given the class, allowing the algorithm to make predictions quickly and accurately.

Pre-Requisites:

Basic programming knowledge, Understanding of Bayes Theorem.

Pre-Lab:

1. What is the Naive Bayes Classification Algorithm? What are the steps to implement a Naive Bayes classifier?

2. What are the Different Types of Naive Bayes Models?

3. What are the applications of the Naive Bayes Classification Algorithm?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

In-Lab:

Implement the Naïve bayes classifier to predict the label for unknown data using **Iris dataset**.

Description: The Naive Bayes classifier is implemented using the scikit-learn library by importing the standard ***iris*** dataset to predict the ***Species*** of flowers.

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Sample VIVA-VOCE Questions (In-Lab):

1. How does the Naïve Bayes classifier work, and what assumptions does it make?
2. What steps do you follow to implement the Naïve Bayes classifier using the Iris dataset?
3. Explain the K-Nearest Neighbors (KNN) algorithm and how it is used for breast cancer detection.
4. What are the key differences between the Naïve Bayes and KNN classifiers?
5. How do you evaluate the performance of your classifiers for predicting unknown data?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Post-Lab:

Implement Breast Cancer detection system using KNN

Description: Implementing a Breast Cancer detection system using the K-Nearest Neighbors (KNN) algorithm involves using a dataset of breast cancer features to classify whether a tumor is benign or malignant based on its characteristics.

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date

Experiment # 10: Predict the price of a House based on a set of characteristics using linear regression

Aim/Objective:

Program to Predict the price of a House based on a set of characteristics using linear regression
.

Description:

Description: Predicting the price of a house based on a set of characteristics using linear regression involves building a model that learns the relationship between the features (characteristics) of houses and their corresponding prices.

Pre-Requisites:

a. Basic understanding the prerequisites for implementing a house price prediction system using linear regression involves foundational knowledge in several key areas.
b. Familiarity with programming concepts and basic knowledge of Python programminglanguage would be beneficial.

Pre-Lab:

1. Describe the key features (independent variables) that you think might influence house prices. How would you categorize these features (numerical, categorical)?

2. Explain the terms "coefficients" and "intercept" in the context of linear regression. How are these values determined during model training?

3. Define and explain the purpose of commonly used metrics such as Mean Squared Error (MSE) and R-squared (coefficient of determination). How do these metrics help evaluate the accuracy of a linear regression model?

4. What is feature engineering, and why is it important in the context of machine learning models like linear regression?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

In-Lab:

Predict the price of a House based on a set of characteristics using ***linear regression.***
Data set: https://www.kaggle.com/datasets/yasserh/housing-prices-dataset

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

VIVA-VOCE Questions (In-Lab):

1. What is linear regression, and how is it used to predict house prices?
2. What are the key features you consider when predicting house prices using linear regression?
3. Explain the Support Vector Machine (SVM) algorithm and its application in document classification.
4. How do you preprocess data for training an SVM model for document classification?
5. What metrics do you use to evaluate the performance of your linear regression and SVM models?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Post-Lab:

Train an SVM classifier on the Iris dataset and evaluate its performance using metrics like accuracy, precision, recall, and F1 score.

Dataset: Iris dataset.

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date

Experiment # 11: Perform grouping of customers into distinct segments based on their similarities in terms of purchasing behavior, demographics, or other relevant attributes using K-Means clustering algorithm

Aim/Objective:

The aim of this experiment is to implement the Perform grouping of customers into distinct segments based on their similarities in terms of purchasing behavior, demographics, or other relevant attributes using K-Means clustering algorithm.

Description:

In this experiment students will work Performing grouping of customers into distinct segments based on their similarities using the K-Means clustering algorithm involves identifying clusters of customers who exhibit similar purchasing behavior, demographics, or other relevant attributes.

Pre-Requisites:

- Students should have a basic understanding of before delving into customer segmentation using the K-Means clustering algorithm, it's important to have a solid foundation in several key areas.
- Pre-Lab:

1. What is clustering in the context of machine learning? How does it differ from classification?

2. Describe the K-Means clustering algorithm. What are its main steps? How does it determine cluster centroids?

3. Why is feature scaling important for clustering algorithms like K-Means? Explain with an example.

4. How do you interpret the clustering results produced by K-Means? What information do cluster centroids provide about customer segments?

In-Lab: Perform grouping of customers into distinct segments based on their similarities in terms of purchasing behavior, demographics, or other relevant attributes using K-Means clustering algorithm.

Data Set: https://www.kaggle.com/code/heeraldedhia/kmeans-clustering-for-customer-data/input.

Description: Students will implement customer segmentation using K-Means clustering allows businesses to discover meaningful patterns in customer data and divide customers into distinct segments based on similarities in purchasing behavior, demographics, or other relevant attributes.

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Analysis and Inferences:

VIVA-VOCE Questions (In-Lab):

1. How does the K-Means clustering algorithm work for grouping customers based on similarities?
2. What are the key steps involved in implementing the K-Means clustering algorithm?
3. How do you determine the optimal number of clusters when using K-Means clustering for the COVID-19 dataset?
4. What metrics or methods can you use to evaluate the effectiveness of your clustering results?
5. How do you handle the issue of cluster initialization in K-Means clustering?

Post-Lab:

By considering the COVID-19 data set and performing k-means clustering for a range of k values (k=1 to 10) and finding the optimal number of clusters

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment # 12: Implement 3 input AND and OR Boolean functions using the McCulloch-Pitts Model using Python

Aim/Objective: To implement 3 input AND and OR Boolean functions using the McCulloch-Pitts Model using Python.

Description: The purpose of implementing 3-input AND and OR Boolean functions using the McCulloch-Pitts (M-P) model in Python is to demonstrate how simple computational units (artificial neurons) can perform logical operations based on input signals. The McCulloch-Pitts model was one of the earliest proposed neural network models and is foundational in understanding the principles of neural computation.

Pre-Requisites: Before delving into implementing Boolean functions using the McCulloch-Pitts model or any neural network model, it's essential to have a solid foundation in several key areas.

Pre-Lab:

1. What is a neural network? Explain the analogy between artificial neurons and biological neurons.

2. What is the McCulloch-Pitts model? How does it simulate the behavior of a neuron using binary inputs and a threshold function?

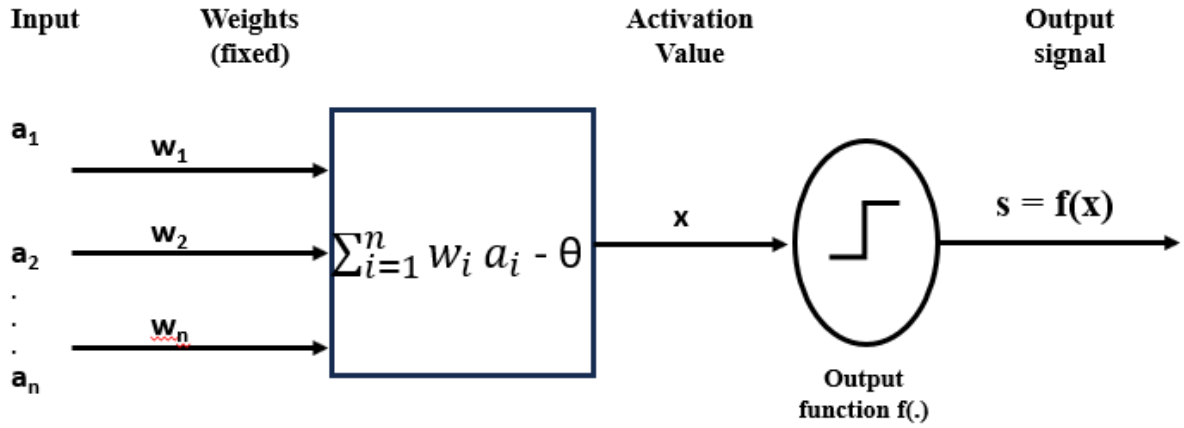
3. Provide examples of Boolean functions (e.g., AND, OR) and explain how they can be implemented using the McCulloch-Pitts model.

4. Discuss McCllouch-Pitts model and Perceptron model with suitable diagram?

5. Describe the steps involved in implementing a basic neural network model in Python using libraries like NumPy or TensorFlow.

In-Lab:

Implement 3 input AND, OR, NOR and NAND Boolean functions using McCulloch-Pitts Model using python programming language**.**



McCulloch-Pitts Model

Situation	a1	a2	a3	Ysum	Yout (AND)	Yout (OR)	Yout (Nand)	Yout (Nor)
1	0	0	0	0	0	0	1	1
2	0	0	1	1	0	1	1	0
3	0	1	0	1	0	1	1	0
4	0	1	1	2	0	1	1	0
5	1	0	0	1	0	1	1	0
6	1	0	1	2	0	1	1	0
7	1	1	0	2	0	1	1	0
8	1	1	1	3	1	1	0	0

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Sample VIVA-VOCE Questions (In-Lab):

1. What is the McCulloch-Pitts model, and how is it used to implement Boolean functions?
2. How do you implement the AND and OR Boolean functions using the McCulloch-Pitts model?
3. What changes are needed to extend the McCulloch-Pitts model to implement NOR and NAND Boolean functions?
4. How do you validate the correctness of your McCulloch-Pitts model implementation for Boolean functions?
5. What are the limitations of the McCulloch-Pitts model in representing complex Boolean functions?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Post-Lab:

Implement McCulloch-Pitts for solve AND and OR gates.

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment # 13: Implement an ANN to predict the Customer Churn

Aim/Objective: To implement an ANN to predict the **Customer Churn**

Description: The purpose of this lesson is to teach the students that how to use Python to build a simple neural network architecture.

Pre-Requisites: Basic ANN architecture, Knowledge of python programming language

Pre-Lab:

1. List out the differences between the Biological neural network (BNN) and Artificial Neural network (ANN)

2. Explain the Characteristic of artificial neural network (ANN) and its applications?

3. What is the Boston Housing dataset? Describe its features and target variable.

4. What are activation functions, and why are they used in neural networks? Discuss common activation functions used in ANN models.

5. What are hyperparameters in the context of neural networks? Give examples of hyperparameters that can be tuned to improve model performance.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

In-Lab: Implement an ANN to predict the Customer Churn:

Customer Churn

It is when an existing customer, user, subscriber, or any kind of return client stops doing business or ends the relationship with a company.

Types of Customer Churn –

- **Contractual Churn :** When a customer is under a contract for a service and decides to cancel the service e.g. Cable TV, SaaS.
- **Voluntary Churn :** When a user voluntarily cancels a service e.g. Cellular connection.
- **Non-Contractual Churn :** When a customer is not under a contract for a service and decides to cancel the service e.g. Consumer Loyalty in retail stores.
- **Involuntary Churn :** When a churn occurs without any request of the customer e.g. Credit card expiration.

Data set https://www.kaggle.com/datasets/hassanamin/customer-churn

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Sample VIVA-VOCE Questions (In-Lab):

1. What is an Artificial Neural Network (ANN), and how does it work for classification?
2. What are the key components and architecture of the ANN you implemented for predicting house costs?
3. How do you preprocess the data for training your ANN model?
4. What metrics do you use to evaluate the performance of your ANN model in Customer Churn?
5. What are some potential improvements you could make to enhance the accuracy of your ANN model?

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Post-Lab:

Implement Convolutional Neural Networks (CNNs) for image recognition tasks in computer vision.
Data Set: https://www.kaggle.com/datasets/nitinsss/fashion-dataset-with-over-15000-labelled-images

Procedure/Program:

Data and Results:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Analysis and Inferences:

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date