**Artificial and Computational Intelligence**

**Assignment 2**

**Part A : Gaming**

**Title: Implement a Human vs. AI Drop Token Game with Minimax**

You are required to develop a **Drop Token game on a 5×5 grid**, where a human player competes against an AI-controlled opponent using the **Minimax algorithm** with a **fixed depth of 3.**

**Requirements:**

* **Grid Representation:** Use a **5-row × 5-column** 2D array to represent the board.
* **Human Player:** The human selects a column to drop their token, which lands in the **lowest available row** in that column. **Ensure that the column is not already full.**
* **Computer Player:** The AI player uses the **Minimax algorithm** with a **fixed depth of 3** to decide the best move based on **winning, blocking, and strategic positioning**.
* **Game Logic:** The players take turns dropping tokens into the grid. After each move, the board updates and checks for a win (**three in a row** horizontally, vertically, or diagonally) or a **draw** (if all columns are full).
* **Output:** Display the board after each move and declare a **winner** or a **draw** when the game ends.

**Rules to Play Drop Token:**

1. **Objective**: The goal is to align three of your tokens in a row (horizontally, vertically, or diagonally) on a 5×5 grid.
2. **Turns**: The players take turns dropping a token into a column. The token falls to the lowest available row in that column.
3. **Winning Condition**: A player wins if they align three tokens in a row.
4. **Draw Condition**: The game ends in a draw if all columns are full and no player has won.

**Sample Moves:**

* **Move 1**: Human drops a token into column 2 → It lands in row 5, column 2.
* **Move 2**: AI drops a token into column 3 → It lands in row 5, column 3.
* **Move 3**: Human drops a token into column 2 → It lands in row 4, column 2.

**Tasks:**

a. You are free to choose your own static evaluation function. Justify your choice of static evaluation value design and explain with a sample game state. Do not use any machine learning model for the evaluation function.

b. Similar to the virtual lab example, one of the players must be a human ie., it must get dynamic inputs from us. The other player must be simulated using the program.

c. Implement Python code for the design under part a, using Minimax Algorithm.

**Part B: Logic**

**Title: Employee Attrition Prediction Using Decision Trees and Prolog**

1. Age (in years)
2. Department (HR, R&D, Sales)
3. DistanceFromHome (in km)
4. Education Level (1 to 5)
5. JobRole (Manager, Engineer, Sales Rep, etc.)
6. JobSatisfaction (1 to 5)
7. MaritalStatus (Single, Married, Divorced)
8. MonthlyIncome (in USD)
9. NumCompaniesWorked (total past jobs)
10. OverTime (Yes/No)
11. TotalWorkingYears (in years)
12. YearsAtCompany (in years)
13. Attrition (Target: Yes/No)

You are required to do the following:

1. Use a **decision tree algorithm** to model the classification of attrition.
2. **Extract rules** from the decision tree.
3. Implement those rules in a **Prolog knowledge base**.
4. Accept **employee details as user input** and **classify whether the employee is likely to leave the organization** or not.

Dataset: [IBM HR Analytics Employee Attrition & Performance Dataset](https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset)

**Submission requires below  documents.**

* 1. Word file containing the detailed explanation of  theory part of both Part A and B. A screenshot of the output should be appended in the document and the entire code execution too with clear output flow.
  2. The Python notebooks (.ipynb file) for the algorithms mentioned in the problem statement.
  3. Prolog file