Artificial and Computational Intelligence
Assignment 2

Problem statement 1 : Gaming

Simulate the working of TicTacToe game extended to three players with coins 'X', 'O' and 'T' with the below sample larger game board of dimension 6*6. The first player to match any four consecutive coins in the same row or same column or same diagonal wins. A sample state of the game board is given below for reference. Your program must start from empty configuration.

X				X	
	X	O		T	
		O	T	T	
		T	T	O	
		X			O

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 two players must be simulated using the program.

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

Problem statement 2: Logic

We have been gathering vital data from engineering students as part of a thorough investigation of the effect of 40-100 on people's success. We ask the following extremely objective questions in a totally voluntary survey that all students are obliged to complete:

- Do you attend parties? [party: yes/no]
- Are you smart? [smart: yes/no]
- Are you innovative? [innovative: yes/no]
- Will you be up-to-date to finish your assignments? [assignment: yes/no]
- Do you have a laptop? [laptop: yes/no]
- Does your project succeed between 40-100? [project: yes/no]
- Have you derived your project goal successfully? [goal: yes/no]
- Do you feel happy now? [happy: yes/no]

You can get the comma-separated survey results from following link.

https://drive.google.com/file/d/1qHqCFLfBt1EZ7BQO7h904WqHp3iCskkS/view?usp=sharing

Evaluations will be based on the following.

1. Use Min-Max algorithm and implement the game in PYTHON (35% marks)
2. Derive the rules from the given decision tree and code as Prolog rules. (35% marks)
3. Interactive implementation. Dynamic inputs-based run of the game with step wise board display and error free game ending. (15% marks)
4. Interactive implementation. Dynamic inputs-based run of the logic expert system with step wise options display and error free recommendation & ending. (15% marks)

Important Note:

- You are provided with the python notebook template which stipulates the structure of code and documentation. Use well intended python code.
- Use a separate MS word document for explaining the theory part. Do not include the theory part in the Python notebook except Python comments.
- The implementation code must be completely original and executable.
- Please keep your work (code, documentation) confidential. If your code is found to be plagiarized, you will be penalized severely. Parties involved in the copy will be considered equal partners and will be penalized severely.