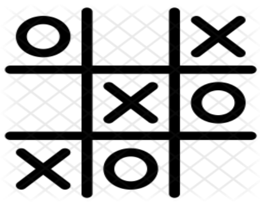
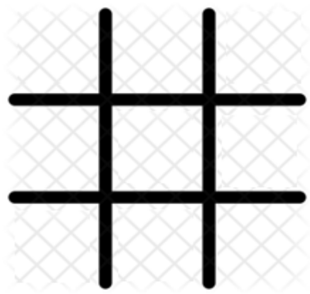
Linear Regression Application

Game playing

A Tic-Tac-Toe game is played between 2 individuals. It has perfect information and is deterministic game.

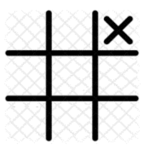
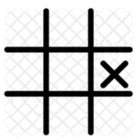
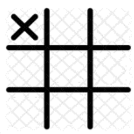


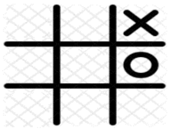
1. Give a formal description of this application in terms of Task, Experience, and Performance. (Make it a well posed problem)
2. Write a function Initialize() that initializes the board state for tic-tac-toe. Board can be represented as 3x3 matrix in python.

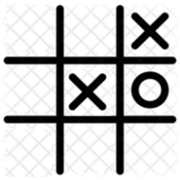
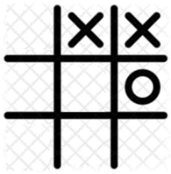


3. Write a function *PossibleMoves(b,p)* that shows all the possible moves for a player *'p'* given the current board state *'b'*.

For instance, possible moves for player X given the empty board state are 9

  ……………… 

Possible moves for player X given the board state  are 7.

 ……..  … and so on

4. Write a function *BestMove(b,p)* to find the best move for a player ‘p’ given a board state ‘b’

   Given a board state b, find all the *PossibleMoves(b,p)*. For each possible move, *Evaluate(b)* and check the highest score.

5. Write a function *Evaluate(b)* to find the score for board state b.

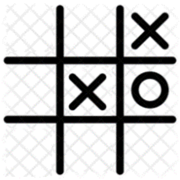
   The features to be considered are

- x1 = # of instances where there are 2 x’s in a ROW with an open subsequent square.
- x2 = # of instances where there are 2 o’s in a ROW with an open subsequent square.
- x3 = # of instances where there is an x in a completely open ROW
- x4 = # of instances where there is an o in a completely open ROW
- x5 = # of instances of 3 x’s in a ROW (value of 1 signifies end game)
- x6 = # of instances of 3 o’s in a ROW (value of 1 signifies end game)

ROW means 3 subsequent squares in diagnols or in rows or in columns

$$\hat{V}(b) = \theta^T \cdot X$$

where X has 6 features mentioned above. θ can be initialized to (0.5,0.5,.....0.5). Since 6 features are there, θ will have 7 values including bias.

For instance, the board state  is represented as

x1 = 1 (since there are 2 X's in subsequent squares of a right diagonal)
x2 = 0
x3 = 3 (only one X in the topmost row; only one X in the middle column; only one X in the left diagonal)
x4 = 0
x5 = 0
x6 = 0

The score for the board state is calculated as follows.

- V(b) = 100 if end of game and you won.
- V(b) = -100 if end of game and you lost.
- V(b) = 0 if end of game and a draw.
- V(b)= $\hat{V}$(b)

6. Extra credit : How will the V(b) change if following function is used

V(b) = $\hat{V}$*(BestMove(b,p))* in intermediate board states ?