REPORT

Introduction:

Tic Tac Toe is a basic game played by two people. To win, you need to get three of our traditional symbols (X or O) in a row, either vertically, horizontally, or diagonally. When humans play Tic Tac Toe against each other, it can get boring after a while. However, when humans play against a computer, it becomes more interesting because they want to see if they can beat the computer. To play Tic Tac Toe against a computer, two algorithms are used: MiniMax and Reinforcement learning.

Minimax:
It's a powerful tool that can help you make decisions in two-player games by considering all possible moves and outcomes.

Here's how it works:

1. Consider all possible moves you can make. This might seem like a lot, but the Minimax algorithm is very efficient and can quickly narrow down the options.

2. For each of your moves, consider all possible moves your opponent can make. This is where the recursion comes in. The Minimax algorithm calls itself again for each of your opponent's moves, and so on, until the game ends.

3. Assign a value to each possible outcome. This is where things get a little more complicated. The value of an outcome depends on who wins and loses. For example, if you win, the outcome is worth +1. If you lose, the outcome is worth -1. And if the game ends in a draw, the outcome is worth 0.

4. Choose the move that leads to the best possible outcome. This is the heart of the Minimax algorithm. It will always choose the move that maximises your chances of winning or minimises your chances of losing, assuming that your opponent will also play optimally.

The Minimax algorithm is a powerful tool that can help you make better decisions in two-player games.

Reinforcement Learning:
It's a powerful tool that can help AI agents learn how to play games by letting them learn from their mistakes and improve over time.

Here's how it works:

1. Set a goal for the AI agent. This could be anything from winning a game of Tic Tac Toe to solving a complex puzzle.

2. Let the AI agent interact with the environment. This means giving the agent the opportunity to make decisions and observe the results.

3. Provide feedback to the AI agent. This could be in the form of rewards for good decisions and penalties for bad decisions.

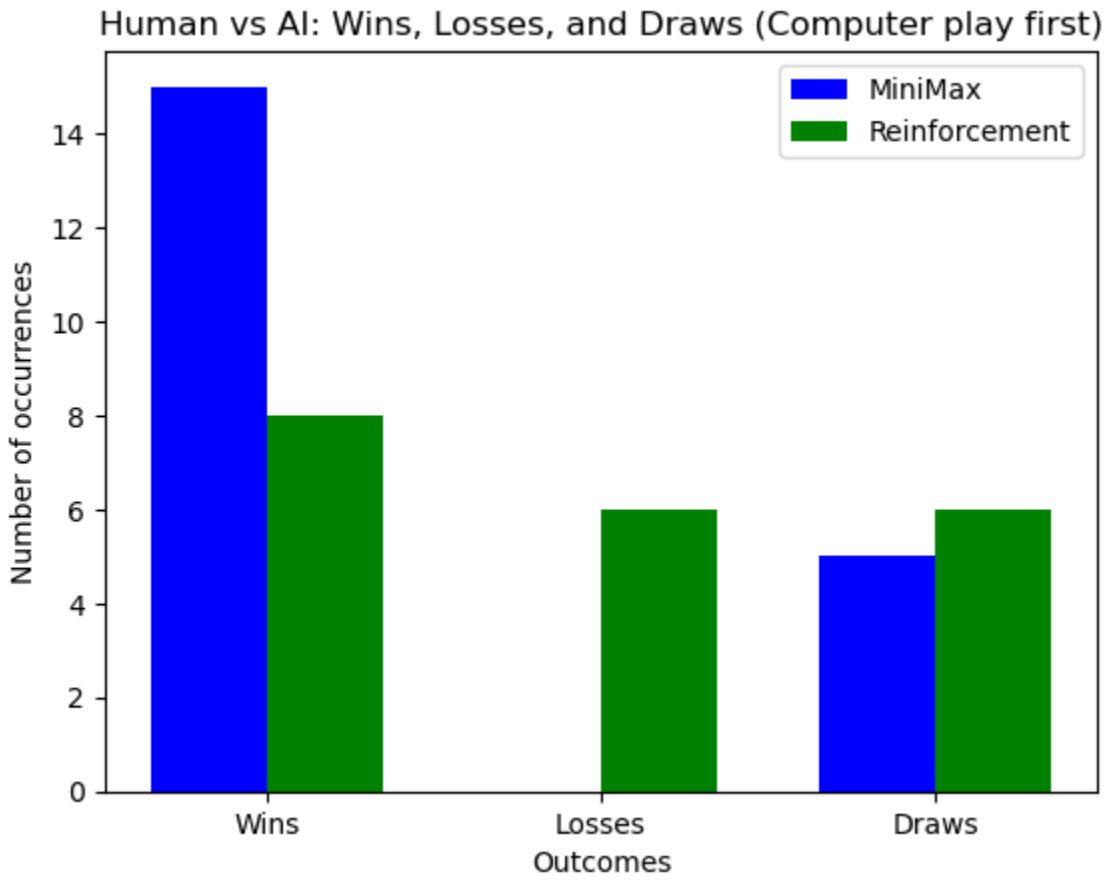
4. Over time, the AI agent will learn to make better decisions in order to get more rewards. This is because the agent will start to associate certain actions with positive outcomes and avoid actions that lead to negative outcomes.

Reinforcement Learning can be a slow process, and it often requires a lot of data to train an AI agent.

Analysis of algorithms:

The bar graph shows the number of wins, losses, and draws in games of MiniMax and Reinforcement Learning (RL) against human players.

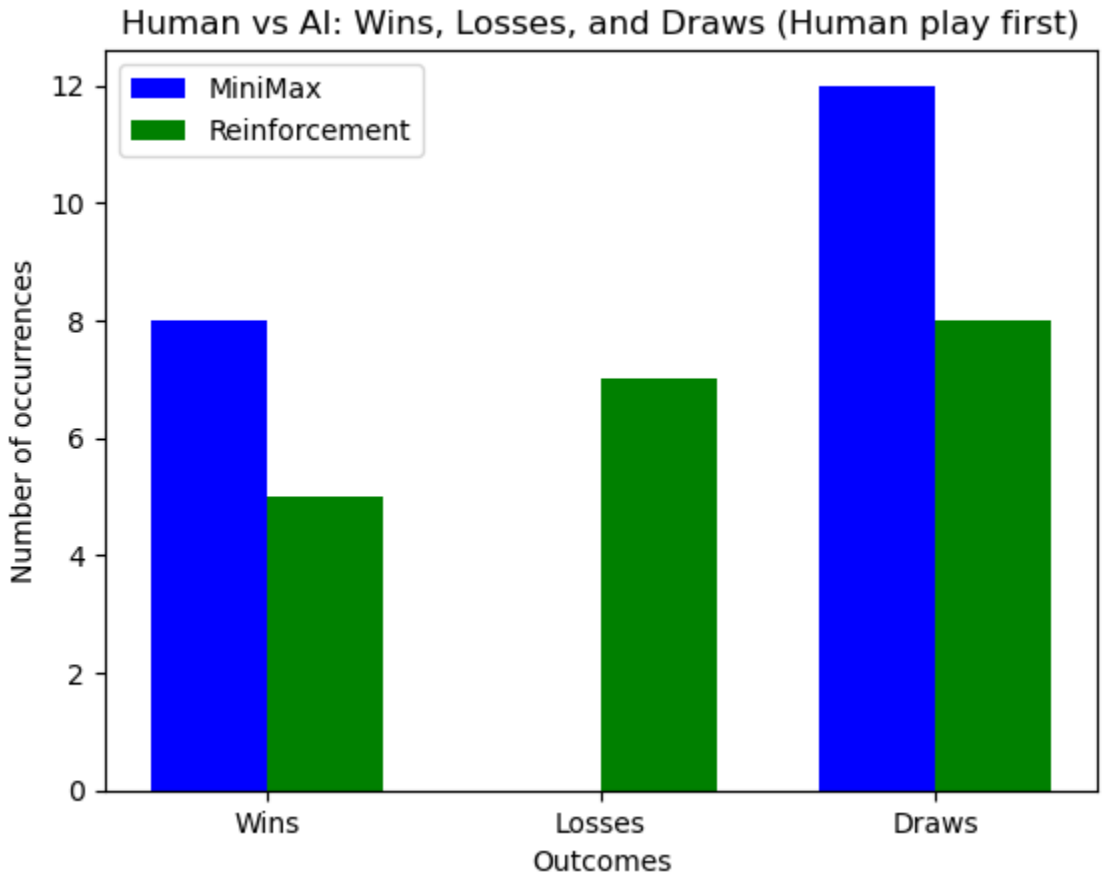
Case I: When the computer is playing first.



The bar-graph above shows the outcomes of a game when the computer makes the first move. In the case of the MiniMax algorithm, the computer always plays optimally, choosing the best possible moves. This results in the computer either winning the game or tying it, with no losses.On the other hand, when a human player makes the best move, the game either ends in a draw or a loss for the human. The graph shows that with the MiniMax algorithm, the game either ends in a win for the computer or a draw.

Whereas, when using Reinforcement Learning, the computer can win or draw the game, but it can also lose against human players. This is because the Reinforcement Learning strategy allows the computer to take risks.

Case II: When Human is playing first.



The bar-graph above shows the results of a game when a human player makes the first move. In this situation, if the computer uses the MiniMax strategy, it will always play the game in the best possible way. This means the computer will either win the game or it will end in a draw, with no losses for the computer.However, there's a twist. The probability of the game ending in a draw increases when the human player starts the game. If the human player makes the best move, the game will either end in a draw or a loss for the human player.The graph shows that with the MiniMax strategy, the game either ends in a win for the computer or a draw. But when the computer uses Reinforcement Learning, it can win or draw the game, but it can also lose against human players.

Analysis of Saturation point

Iteration per state	MiniMax_play_first	RL_play_first	Remark
500	MiniMax won	Minimax won	As we are increasing the number of iterations, we reach the draw state from which we can infer that there is a saturation point.
1000	MiniMax won	MiniMax won	
5000	Minimax won	Draw	

Analysis of Json Storage:

Storage type	Memory Space	Remark
1-D array with nested parent-child relationship	72000 KB	Initially I was using a 1-D array as a state then I used String as states with nested parent-child relation. But, in the end I used only unique states-value pairs in the dictionary. From this I saved a lot of space.
States as string with parent-child relationship	41000 KB	
Unique states-value pair in dictionary	54 KB	

Credits -: Kirtan (Helped in explanation of how to approach the code)