# **Tic Tac Toe - Creating Unbeatable AI by using the minimax algorithm.**

This time we are going to build an unbeatable AI agent that plays the classic **Tic Tac Toe** game. First we will learn the concept of the **Minimax** algorithm that is widely and successfully used across the fields like **Artificial Intelligence**, **Economics**, **Game Theory**, **Statistics** or even **Philosophy**.

# **Minimax algorithm**

Minimax is a recursive algorithm which is used to choose an optimal move for a player assuming that the opponent is also playing optimally. As its name suggests, its goal is to minimize the maximum loss. This algorithm sees a few steps ahead and puts itself in the shoes of its opponent. It keeps playing ahead until it reaches a terminal arrangement of the board (**terminal state**) resulting in a tie, a win, or a loss. Once in a terminal state, the AI will assign an arbitrary positive score (+10) for a win, a negative score (-10) for a loss, or a neutral score (0) for a tie.

The algorithm also evaluates the moves that lead to a terminal state based on the players’ turn. It will choose the move with maximum score when it is the AI’s turn and choose the move with the minimum score when it is the human player’s turn. Using this strategy, Minimax avoids losing to the human player.

A Minimax algorithm can be defined as a recursive function that does the following things:

1. return a value if a terminal state is found (+10, 0, -10)
2. go through available spots on the board
3. call the minimax function on each available spot (recursion)
4. evaluate returning values from function calls
5. and return the best value

