Ultimate Tic Tac Toe Strategy

Modeling: I represent the board as a list of strings, where each string represents a sub board of the larger board. Indices 0-9 match to the board in the following way:

0 1 2

3 4 5

6 7 8

This applies to both subboards and the list representing the larger board. In each string, “.” is used to represent an empty space, while “x” and “o” are the tokens.

Strategy: My method for scoring the board entails three methods. They are described below.

sub\_score(board, square): This method returns a score for the sub board in index square, which is represented by board[square]. First, if X or O has won this board, I return 10,000 or -10,000, respectively. Otherwise, I check if either X or O is close to winning, meaning two out of three winning squares are filled. For each group of squares, if this case happens, I add 1,000 or -1,000, depending on whether X or O is close to winning. Finally, based on the positions of X and O, I add a weighted sum using the weights [2, 1, 2, 1, 4, 1, 2, 1, 2], because center square > corner square > edge squares.

get\_weights(subboard): This method returns an list of weights for each square in the sub board. Using the same method as above, I weight squares that are key to winning more heavily. This method is a helper method for score().

score(board): If X or O has won, then I return plus or minus one billion, depending on who has won. Otherwise, I make a sub board based on how many sub boards X or O has won. For clarification, this sub board is a string of 9 characters, each one corresponding to one of the 9 sub boards in the actual board. I use the get\_weights() method to return the proper weighting for the score. Then, I add the weighted sum of the score of the sub boards using the sub\_score() method and return this value.