Heuristic Analysis

I experimented with three different heuristics and compared the difference of their performance.

These evaluation functions operate well within the allocated time for each move. It's important that an evaluation function executes quickly to maximize the depth of game tree searched.

Custom Score 1:

The evaluation score is based on number of moves available for the player minus the number of moves available to opponent.

```
my_moves - 4 * opponent_moves
```

I added a weight of 4 (this was found on trial and error basis) to opponent moves to penalize when there is less number of moves available to opponent and this has improved the win rate to 70% compared to 64.3% of AB_Improved. Increasing the number of opponent's moves by 4 times ensured the penalization of boards where opponent had more number of available moves.

Custom Score 2:

```
own_moves * own_moves - opp_moves * opp_moves
```

The base idea is same as 1 but the evaluation function is quadratic. And the performance improved slightly i.e. 71.4% as compared with 64.3% win rate of AB_Improved.

Custom Score 3:

```
blank_spaces / 2 - 2 * my_moves
```

This time I experimented with number of blank spaces left minus number of my moves. To penalize player, I added weight to current player's moves and halved the blank space available. But this was not best heuristic compared to

AB_Improved as the performance decreased to 61.4%. This was I believe because the number of blank space does not mean they are all legal moves for the current player.

I have included the results of my custom evaluation functions below:

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	10	0	9	1	7	3
2	MM_Open	6	4	6	4	8	2	6	4
3	MM_Center	8	2	9	1	10	0	10	0
4	MM_Improved	7	3	6	4	6	4	6	4
5	AB_Open	4	6	7	3	5	5	5	5
6	AB_Center	6	4	7	3	6	4	5	5
7	AB_Improved	5	5	4	6	6	4	4	6
	Win Rate:	64.3%		70.0%		71.4%		61.4%	

Conclusion:

I decided to use Custom score 1 for evaluation because of following reasons:

a. Custom score 1 has higher win rate.
b. The function is easy to understand and intuitive, enabling us to expand on the function in future.
c. This function does not require much computation power as there are not many complex computations involved. Thus allowing the game tree to be searched as far as possible without timeouts.

Further Improvement for Evaluation Functions:

Evaluation functions can also be derived using Genetic algorithm and Ensemble methods but this was out of scope for this project.