The following is the analysis and the results of the custom heuristics compared with the baseline performance of an agent called "AB_Improved" that uses iterative deepening and improved score heuristic defined in “sample_players.py”.

The improved score heuristic function outputs a score equal to the difference in the number of moves available to the two players. As our goal was to develop a heuristic such that custom heuristics (AB_Custom) outperforms the improved score heuristic (AB_Improved), I have come up with the following three types of custom heuristics which is similar to the improved score heuristic:

1. **AB_Custom:** Outputs a score equal to the difference in the number of moves available (square rooted) to the two players. By square rooting the number of moves available for both players, lower moves available are weighted more vs. higher moves available.
2. **AB_Custom2:** Outputs a score equal to the difference in the number of moves available (squared) to the two players. By squaring the number of moves available for both players, higher moves available are weighted more vs. lower moves available.
3. **AB_Custom3:** Outputs a score equal to the difference of the square of the distance from the center of the board to the position of each player

To lower the variance of the results, I have performed 100 matches x 10 times to assess the performance of each custom heuristic. The following table shows the results of the matches performed (1000 matches in total).

		AB_Improved		**AB_Custom**		**AB_Custom_2**		**AB_Custom_3**	
Match #	**Opponent**	**Won**	**Lost**	**Won**	**Lost**	**Won**	**Lost**	**Won**	**Lost**
1	Random	942	58	946	54	930	70	937	63
2	MM_Open	753	247	740	260	759	241	682	318
3	MM_Center	871	129	885	115	906	94	874	126
4	MM_Improved	712	288	730	270	709	291	621	379
5	AB_Open	515	485	551	449	534	466	436	564
6	AB_Center	600	400	600	400	629	371	497	503
7	AB_Improved	493	507	486	514	501	499	422	578
	Total:	**4886**	**2114**	**4938**	**2062**	**4968**	**2032**	**4469**	**2531**
	Win Rate:	**69.8%**		**70.5%**		**71.0%**		**63.8%**	

As to assess the performance of the results objectively, the following three assessments were used to compare the performance: 1) overall Win Rate, 2) Win Rate against each opponent, and 3) Win Rate against AB_Improved. The following table summarizes the results of the three assessments performed and the analysis for each custom heuristic.

	AB_Custom	**AB_Custom_2**	**AB_Custom_3**
1. Overall Win Rate	△	○	×
2. Win Rate against each opponent	○	○	△
3. Win Rate against AB_Improved	×	△	×

1. **AB_Custom:** The performance of the heuristic was good where the overall Win Rate was greater than AB_Improved. However, the Win Rate against AB_Improved was lower than 50%.
2. **AB_Custom2:** The performance of this heuristic was the best among the three heuristics. Overall Win Rate outperformed AB_Improved and all Win Rate against each opponent (including AB_Improved) was greater than 50%.
3. **AB_Custom3:** The performance of this heuristic was bad. Overall Win Rate was lower than AB_Improved, and all Win Rate for the opponent which used the AB tests were lower than 50%. This result exemplified that the more complex the heuristic is, the less iterative deepening can be performed which results in a lower Win Rate. A similar result was also observed where MM_Center and AB_Center had lower Win Rate compared to other heuristics in the sampled player.

Based on the results of the analysis performed above, I would recommend AB_Custom2 (outputs a score equal to the difference in the number of moves available (squared) to the two players) to use in my submitted agent for the following three reasons:

1) Overall Win Rate was the highest among the other heuristics

2) The code is simple to evaluate the board state

3) The heuristic traverses the game tree deeper compared with the other heuristics (especially AB_Custom3)