Heuristics Analysis

Scoring method 1: Improved Score
This scoring method was inspired by a Udacity lecture/quiz, where intuition suggested that the more open moves for active player, and the less open moves for the opponent, the better.
Scoring function = player open moves – opponent open moves

Scoring method 2: custom_score, Improved Score variation
This heuristic was inspired by the Improved Score, but focus on the active player open moves
Scoring function = player open moves x 2 – opponent open moves

Scoring method 3: custom_score_2, Improved Score variation
This heuristic was inspired by the Improved Score, but with emphasis on the opponent's open moves
Scoring function = player open moves – opponent open moves x 2

Scoring method 4: custom_score_3, distance between active player and opponent
This heuristic focused on the distance between player and opponent. The larger the distance between opponents the better, encouraged player to move away from the opponent
Scoring function = euclidean distance between player and opponent

Scoring method 5: custom_score_4, distance between active player and opponent
This heuristic focused on the distance between player and opponent. The smaller the distance between opponents the better, encouraged player to move closer to the opponent
Scoring function = -1 x euclidean distance between player and opponent

Tournament Results

Summary:

- Looking at the Random Algorithm, it still had 22% win rate, indicating room for improvement for implemented scoring functions/heuristics.
- AlphaBeta Search had higher win rate than Minimax Search. This was not surprising as AlphaBeta was more effective in going deeper into the Search.
- Of all the scoring functions tested, scoring method 2 performed the best overall, with 66.43% win rate.

Match #	Opponent	AB_Improved		AB_Custom (Improved, emphasis on Open)		AB_Custom_2 (Improved, emphasis on Opp Open)		AB_Custom_3 (The further away from opponent the better)		AB_Custom_4 (The closer to opponent the better)		Opponent Win Rate
Results (games)		Won	Lost	Won	Lost	Won	Lost	Won	Lost	Won	Lost	
1	Random	15	5	16	4	16	4	16	4	15	5	22.00%
2	MM_Open	13	7	12	8	13	7	16	4	13	7	33.00%
3	MM_Center	15	5	17	3	16	4	15	5	14	6	23.00%
4	MM_Improved	15	5	14	6	14	6	7	13	13	7	37.00%
5	AB_Open	11	9	9	11	9	11	12	8	10	10	49.00%
6	AB_Center	9	11	14	6	10	10	11	9	13	7	43.00%
7	AB_Improved	10	10	11	9	13	7	8	12	13	7	45.00%
My Win Rate:		62.86%		66.43%		65.00%		60.71%		65.00%		

- Even though scoring method 2 was the overall winner, scoring method performances varied widely against opponents using different search or scoring methods. For example:
  - Scoring method 4 performed the best against Open scoring method.
  - Scoring method 2 performed the best against Center scoring method
  - Scoring method 3 and 5 performed the best against Improved scoring method.

Match #	Opponent	AB_Improved	AB_Custom (Improved, emphasis on Open)	AB_Custom_2 (Improved, emphasis on Opp Open)	AB_Custom_3 (The further away from opponent the better)	AB_Custom_4 (The closer to opponent the better)
1	Random	75%	80%	80%	80%	75%
2	MM_Open	65%	60%	65%	80%	65%
3	MM_Center	75%	85%	80%	75%	70%
4	MM_Improved	75%	70%	70%	35%	65%
5	AB_Open	55%	45%	45%	60%	50%
6	AB_Center	45%	70%	50%	55%	65%
7	AB_Improved	50%	55%	65%	40%	65%

Other Thoughts

- Even though my heuristics weren't the best, it was possible to optimize the heuristics by plotting parameter against player performance.
- Supervised Machine learning could be used to shorten the depth of the search process. I could train the model to recognize the outcomes on a smaller grid/state or geographies of all available moves, and where the opponent was.