Heuristic Analysis

# Heuristic Descriptions

I will start by going through the three heuristics I created, and describing how they work from a high level:

* Custom\_Score is a copy cat strategy, trying as hard as it can to do what the other player did their previous move
* Custom\_Score\_2 can be described as an extended AB\_improved
* Custom\_Score\_3 can be described as a modified AB\_improved

## Custom\_Score Description

Custom\_Score is a copy cat strategy:

It is incredibly simple: it will try to place pieces as close to where the other player placed their piece last turn.

## Custom\_Score\_2 Description

Custom\_Score\_2 can be described as an extended AB\_improved:

* Custom\_Score\_2 calculates the difference between it’s moves, and its opponents moves the same way that AB\_Improved does,
  + If Custom\_Score\_2 is ahead of the opponent, it will try to place pieces as close to the bottom right corner as possible
  + Otherwise, if Custom\_Score\_2 is behind, it will play the same AB\_Improved
* The thinking behind Custom\_Score\_2’s strategy, is that if placing pieces in the bottom right corner is outperforming AB\_Improved, then it will continue to keep placing pieces in the bottom right corner, which should lead to it winning,
  + If placing pieces in the bottom right, isn’t working, then it will revert to the same strategy as AB\_Improved, which should give it a chance to win a game it is falling behind in

## Custom\_Score\_3 Description

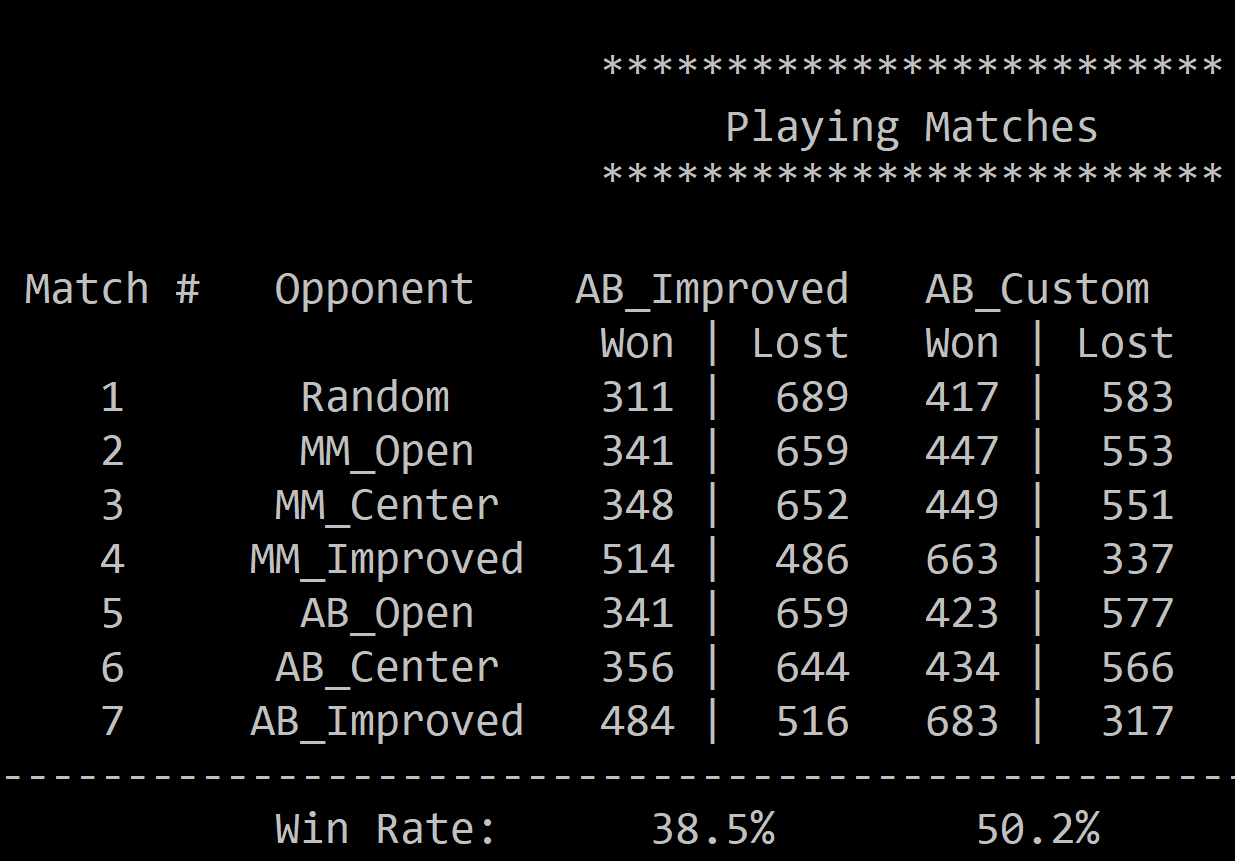
Custom\_Score\_3 can be described as a modified AB\_Improved:

* It uses two parts to calculate utility:
  + Part 1: the difference between players and the opponents moves
  + Part 2: the distance between the player and the opponent, the player is incentivized to try to get away from the other player
* These two parts are added together and returned to drive the agents behaviour

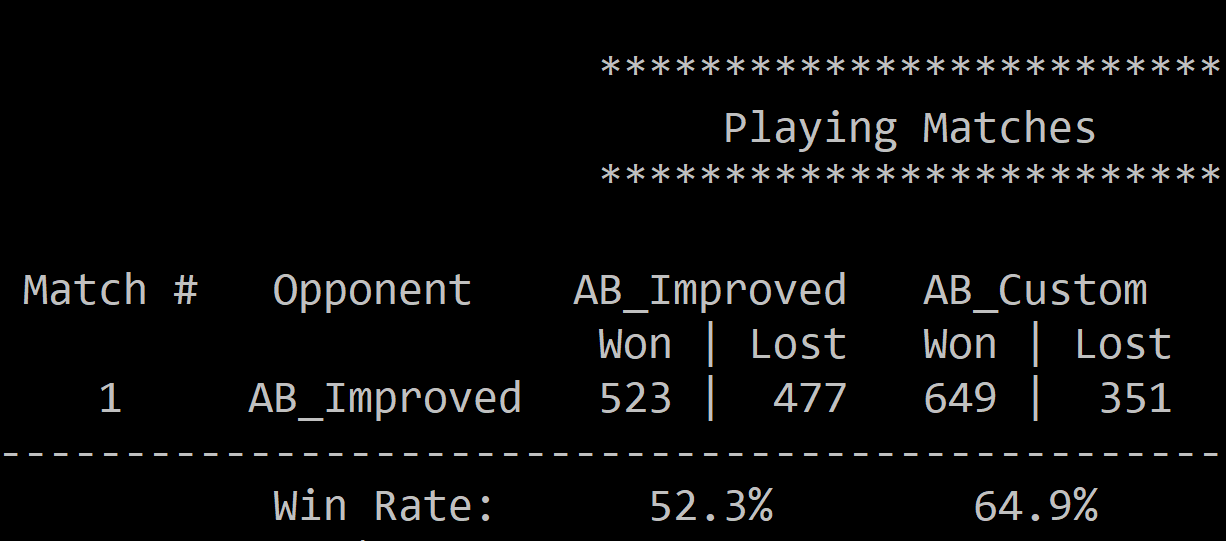
# Heuristic Performance

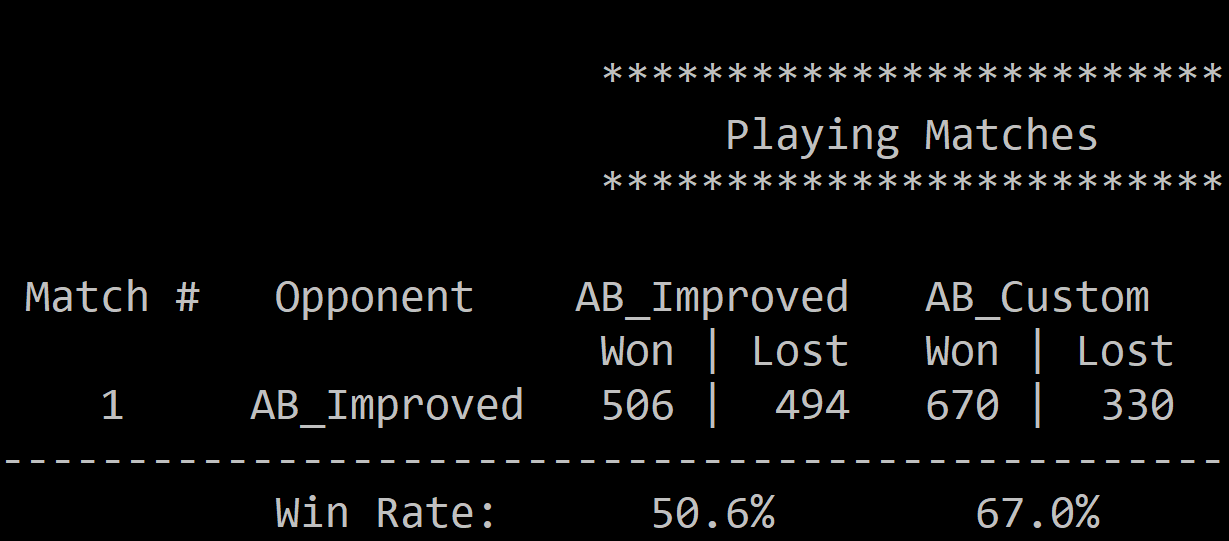
## Custom\_Score Performance

Each table was created running tournament.py with 500 matches (1000 games total). The Custom\_Score heuristic doesn’t tend to do very well against the majority of example opponents.



However, Custom\_Score does well against AB\_Improved, winning roughly 2/3’s of the games when it plays 500 matches against AB\_Improved. For this reason, **I choose Custom\_Score as the best heuristic of the three I created**.





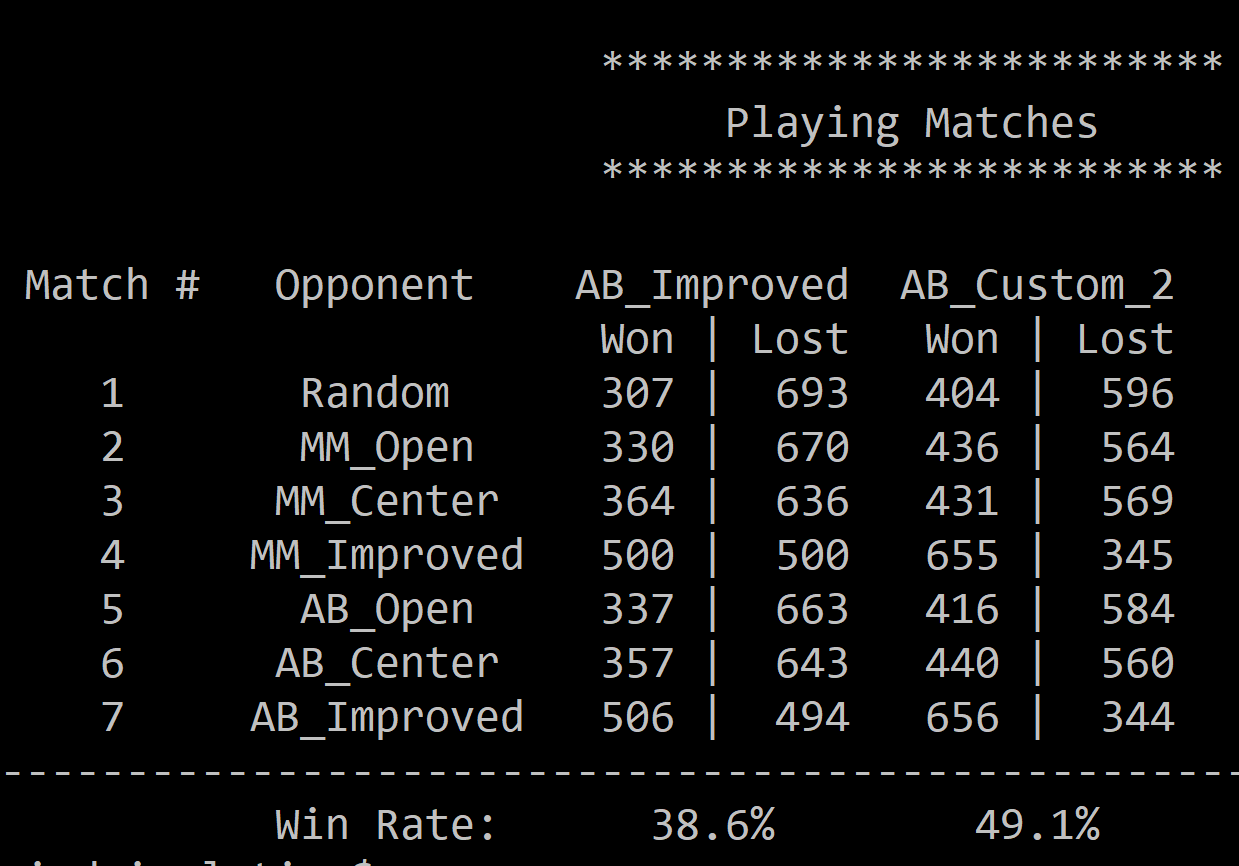
Why does Custom\_Score do so well against AB\_Improved:

**( I DON’T KNOW )**

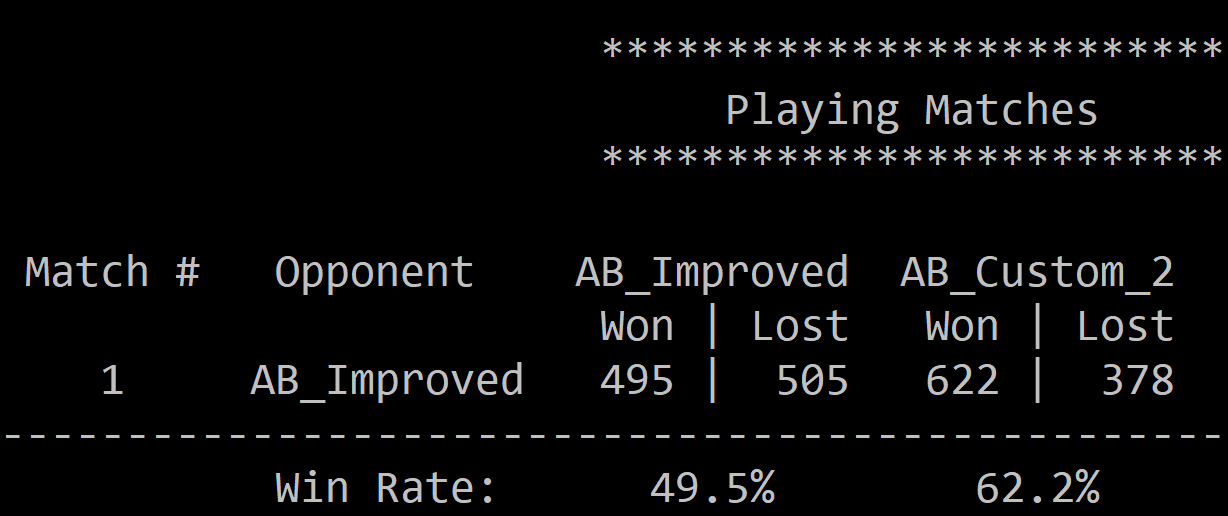
For these reason I think Custom\_Score is the heuristic that should be choosen.

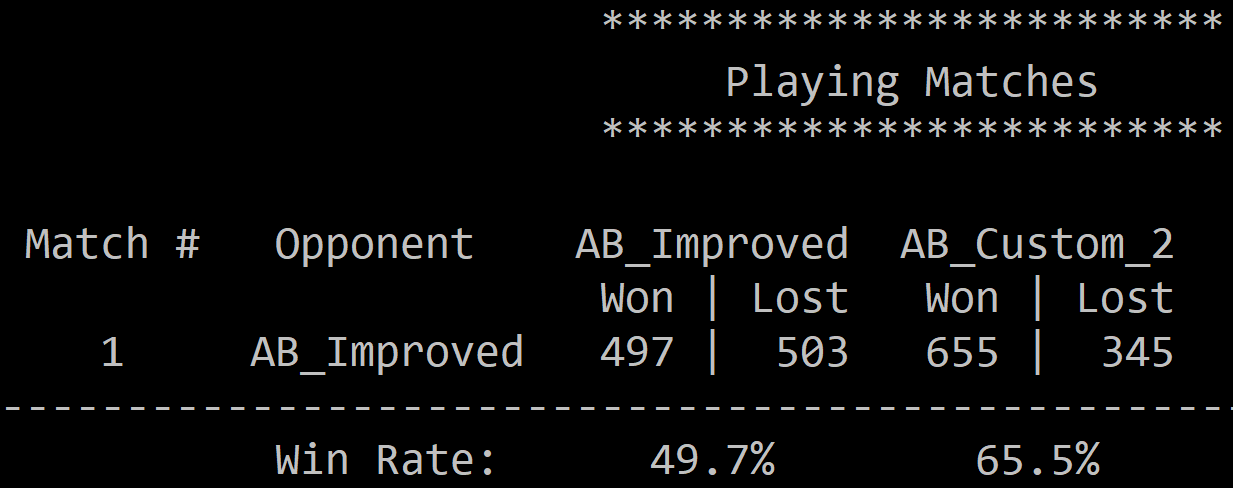
## Custom\_Score\_2 Performance

From all my tests, it looks as though Custom\_Score\_2 is always close to having a 50%-win rate against the 7 example agents, but never manages to get above 50%.



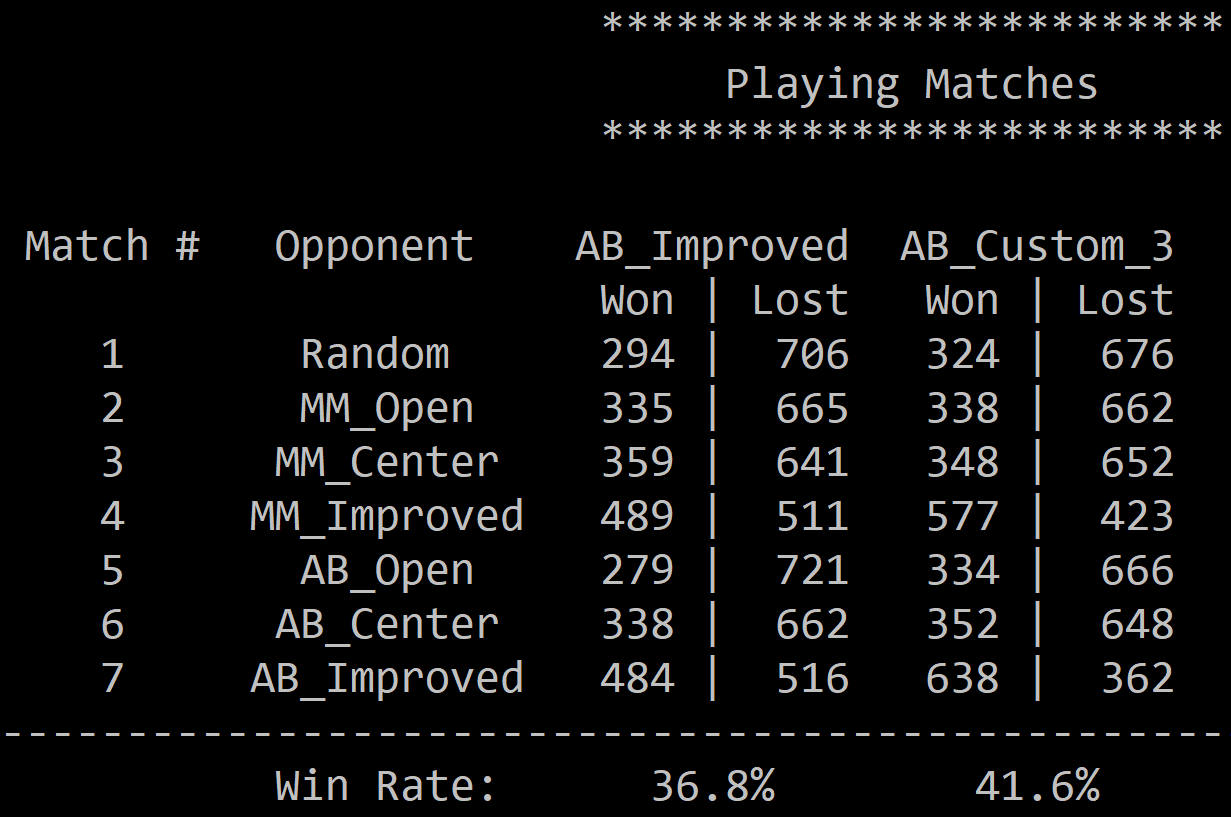
Custom\_Score\_2 performs similarly against AB\_Improved, winning roughly 2/3 of games. However it does perform slightly worse than Custom\_Score.





## Custom\_Score\_3 Performance

Custom\_Score\_3 does the worst against the other 7 test agents.



Custom\_Score\_3 wins roughly 60% of it’s games against AB\_Improved, which is a little worse than the roughly 66% that Custom\_Score and Custom\_Score\_2 win.

