Shallow Mind –
Chess AI

Will Herzog

How do you teach a computer to play chess?

How do you evaluate a game of chess?

How about a single move?

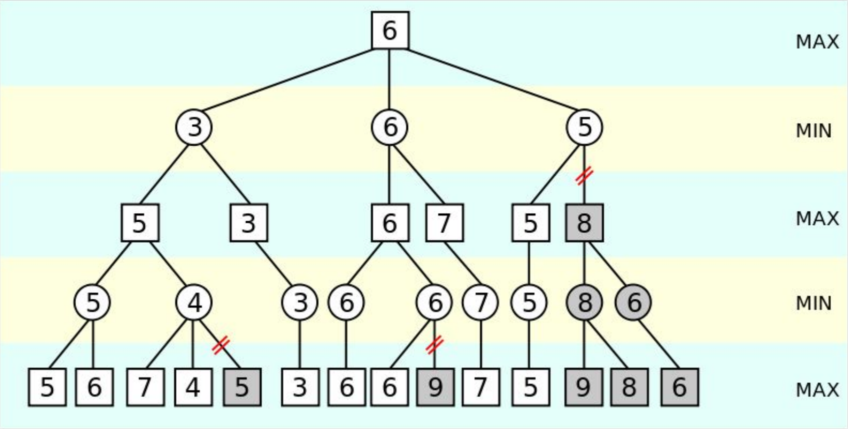
Methodology:

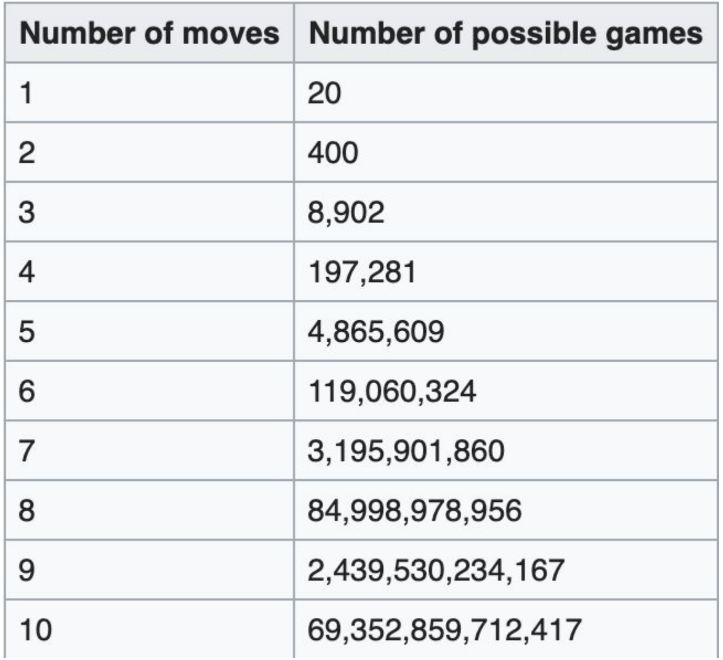
- Train the model to accurately approximate complex chess positions
- Evaluate possible future board states
- Pick the best corresponding move
- Win

https://stockfishchess.org/

StockFish

- One of the Strongest chess engines
- Alpha-Beta search with aggressive pruning
  - Enables greater search depth of moves





Number of moves	Number of possible games
1	20
2	400
3	8,902
4	197,281
5	4,865,609
6	119,060,324
7	3,195,901,860
8	84,998,978,956
9	2,439,530,234,167
10	69,352,859,712,417

How to Evaluate Chess Positions

Available Data:

- Over 250 Million game strings
  - ~ 3M GM selected

Embedded Data:

- Positional
  - White or black
  - 6 types of pieces
  - 8 x 8 playing area
- Turn in game
  - White or black

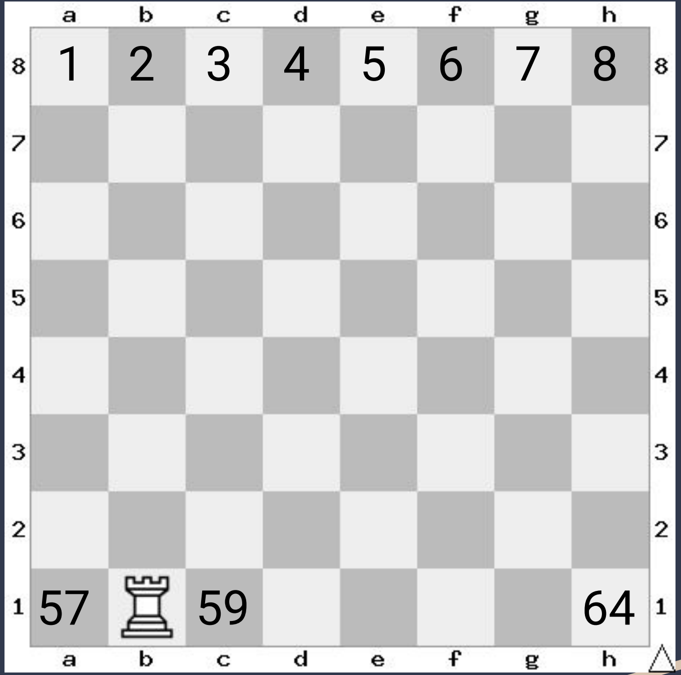
~ 2 Seconds per move

Positional value

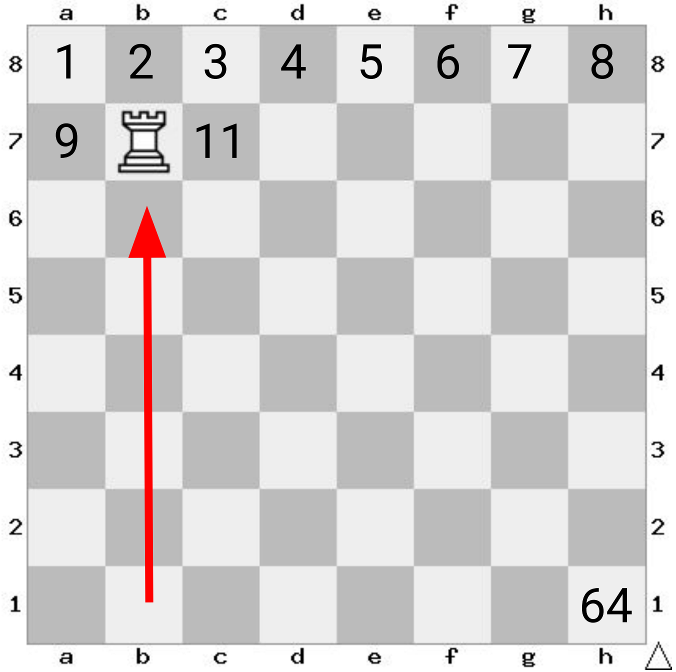
*Target Variable

Goal:

Train a model to detect these patterns and correctly approximate the positional value without the computational overhead of Stockfish



Rook57	Rook58	. . .	Rook64
0	1	0	0
0	0	0	0
. . .	. . .	. . .	. . .



	. . .	Rook09	Rook10	. . .
Move 1	0	0	0	0
Move 2	0	0	1	0
etc	. . .	. . .	. . .	. . .

Model Results

Primary metric:

Root Mean Squared Error (RSME)

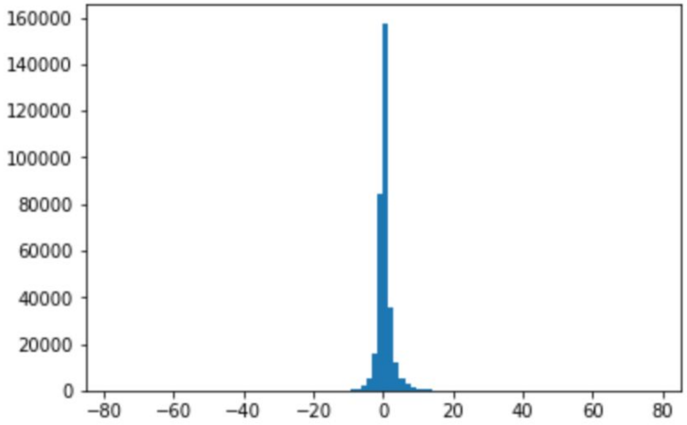
- How close are the predicted positional values are to the Stockfish evaluation

XGBoost: 1.78 ~ pawns

Artificial Neural Network: 1.48 ~ pawns

Future Work

- Exploration of KDTrees for clustering and approximating comparable board states
- Utilize undersampling to reduce kurtosis in cp score distribution
    - Will encourage predictions outside of the predominant range
- Further tuning of Neural Network hyperparameters
- Application of PySpark to improve data processing times



Histogram of Stockfish cp-score