# Table of contents

# Table of Figures

# Table of Tables

# Abstract

Here I will outline the actions performed in this thesis along with a brief description of the topic and results.

# Chapter 1: Introduction

This section will define the problem, background behind it, and the expected outcome as a result of the analytics performed

# Chapter 2: Research Design

A brief introduction into the planning behind the design and the impact of this design on the research.

## Primary Data

Primary data for this project will be established through an experimental process which will apply chess engines to chess mating puzzles with a fixed length. The engines will be tested for their depth efficiency at which they determine an endgame is a mating pattern, implying a higher possibility for depth of searching given a chess position.

## Problem Identification and Clarification

The problem set for this project is that of the incorporation of chess assessments into a chess opening position recommendation algorithms. Engines view problems as mathematical issues and do not assess with nuance like a human brain. Perhaps this straight to the point assessment can be deterministic in the development of a recommendation model?

## Research Objectives

### Chess Engines and the Horizon Effect

Chess engines incorporate different algorithms for the determination of the optimal line of play in a given position. Stockfish incorporates an Alpha-Beta Pruning search algorithm that uses human trained heuristics and weightings to determine the value of search paths and the cutoff points for the pruning of its search nodes. Leela Chess Zero uses an alternative search method known as the Monte-Carlo Tree search, which randomly iterates through more linear search nodes to come to a reach an optimal solution. Unlike Stockfish, Leela Chess Zero’s is trained through a neural network, which through a series of games played against itself, it developed it developed its own heuristics and weights to evaluate each search node.

The horizon effect is an obstacle for search algorithms where a solution to a problem lies beyond the depth restriction of a search algorithm, such as the depth limit option in Universal Chess Interface compatible engines like Stockfish and Leela Chess Zero.

The objective here is to evaluate and justify any measurable difference in the parameters required by engines with different problem solving approaches in finding the pre-determines solution based on end game puzzles.

## Chess Engine Evaluation and Real-World Outcomes

Chess Openings are defined by a classification based on codes from the Encyclopaedia of chess Openings [REFERENCE] (ECO), which can be further subdivided into sections: Opening, Line, and Variation. The Opening is initial moves played, the Line is the approach taken after these moves, and the Variation is a slight difference to change the overall tactical approach of the opening.

A famous example the “Queen’s Gambit” and its lines and variations are classified under the umbrella of code D06 and its variations under codes D07-D69. Each opening has a use case depending on the players desired approach to the game with respect to time control and even knowledge of the opponents repertoire, but this is more so the case in classical over the board games.

Each opening has various success rates with different players. This is due to the opponent being able to vary the game significantly with every possible move after an opening is played hugely increasing the number of permutations a board can take.

## Player Rating and Opening Move Distributions

## Recommendation Model

## Validity Type

## Ethical Considerations

# Chapter 3: Literature Review

# Chapter 4: Methodology

## 4.1 Data Collection and Pre-Processing

All data for this project was sourced from Lichess.org. Lichess provides a database of all games played on the website on a monthly basis and makes them available for use under the Creative Commons 0 License. Along with this they also publish all their puzzles along with games played.

The games are available as Portable Game Notations (PGN) files. These can be read as text documents and contain key information about games such as player names, their ratings, the result, time-control, and the moves played in the game.

PGN as a database format is standardised for Universal Chess Interface (UCI) software but does not allow for easy importation into analytics tools such as Excel or in this case Python, specifically Pandas. To account for this, a program with a specific function was developed to process a PGN format database of 200 plus Gigabytes of data into a usable CSV format.

The key problem to solve with this was handling the massive amounts of data due to be processed. To circumvent this issue, the text file was read in line by line where the indicator of information piece for each game would allow it to be assigned a position in a dictionary. This dictionary was then continually appended to itself for every new game and once enough games had been processed, was converted to a format that could be deposited in a Comma Separated Values (CSV) file.

Once a sufficient number of games had been reached, in this case [INSERT GAME AMOUNT HERE] the function was interrupted, and the data was available for use.

## 4.2 Puzzle Analysis

A database of puzzles was obtained from Lichess Puzzle Database (Lichess, 2023). This is an open database published by Lichess to provide its users with access to all the puzzles they use on their site. It also includes summary statistics for the puzzles.

The Comma Separated Values file provided by Lichess.org was imported into a Python environment using Pandas where it was filtered to extract all the puzzles fitting a ‘Mate in X’ description. This provided a variety of different puzzles on which an analysis could be performed to establish the difference between engine horizon problem solving capabilities.

To provide sufficient testing for the engines, a range of mate lengths were chosen from the longest available. The database consisted of a range of ‘Mate in X’ problems ranging from 1 move to 10 moves. Five was chosen as the lowest value to provide a depth where the engines would need to execute a minimal search and 8 was chosen as the upper limit to allow for enough comparison in distribution for the selected engine parameters. Both populations for ‘Mate in 9’ and ‘Mate in 10’ puzzles would not have provided a large enough sample to compare distributions, both having 9 and 2 puzzles respectively, whereas puzzles with 5, 6, 7, and 8 moves have 3058, 2320, 503, and 183 puzzles in their samples.

A Python program using Python-Chess library (Python-Chess, 2023) was developed to implement an experimental procedure to test both Stockfish and Leela Chess Zero, two of the top open-source engines. In this program a custom function was developed to utilise the python-chess Universal Chess Interface (UCI) analyse function in an open ended analysis of each puzzle. This analysis allowed for the engines to access the puzzle and perform their search until a mate with the expected X value was found. Once found the function would cease its execution and return the parameters the engine had used to find its solution to the ‘Mate in X’ puzzle.

This approach was found to more granularity for analytics as opposed to the Python-Chess engine limit protocol. By providing defined limits for depth of search and time searching, the engine would be tested multiple times to see at which combination of parameters it could determine a solution. The results of this process showed multiple combinations of depth and time that provided solutions for each problem but did not allow the engine to approach the problem in it’s base state. The change to an indefinite analysis pending a solution allowed each tested engine to approach the puzzle and return an exact combination without intervention by the tester.

The results from each search were returned by the engine as an information dictionary as dictated by each of their UCI protocols. Although these Python dictionaries had different formatting, they both contained the parameters of interest for this experiment being ‘time’, ‘seldepth’, and ‘nodes’. The ‘time’ is defined as the time until a solution was reached, ‘seldepth’ is defined as the depth at which the accepted solution was selected from, and ‘nodes’ are the number of branches searched by an engine in its approach to finding a solution.

Once all the above parameters were extracted from the engines during the analysis process, each distribution was compared to its corresponding parameters for each categorical ‘Mate in X’ group to provide a one to one experiment.

All statistical tests performed were test with an alpha = 0.05. This is an established value proposed by Fisher as a means to suggest a cutoff point where confidence could be attributed to a result (Fisher, 1954). This value is used conventionally in many types of research and is generally considered that 95% is an acceptable confidence level (Thiese & Ronna, 2016).

As distributions were being compared, the correct statistical tests had to be selected to ensure an accurate comparison of data was being performed. The choice of test was primarily driven by whether the data was likely to conform to a normal distribution or not, which required its own statistical test. The Shapiro-Wilk test was used to assess whether the Null Hypothesis that each of the distributions for the individual variables split by category were normally distributed. As both ‘seldepth’ and ‘nodes’ are discrete values, non-parametric tests would be the choice for them.

The Shapiro-Wilk-test tests for normality in distributions by comparing the assuming the distribution would adhere to a normal curve and generating a test statistic based on the deviation from normality (SHAPIRO & WILK, 1965). The hypothesis for the test in this case was for ‘time’ distributions being normal given a P-Value of 0.05. All results for this test showed that the distributions were not normally distributed meaning the comparative tests would need to be non-parametric.

The Mann-Whitney U test is a non-parametric comparative test that is considered as powerful as the parametric t-test (MacFarland & Yates, 2016) which is one of the most common tests used in research journals (Yim, et al., 2010). Due to its usefulness as a comparative test, it was implemented into this analysis to compare equivalent engine parameters between engines within their respective ‘Mate in X’ categories.

The results of the Mann-Whitney U test were checked to determine whether the distributions were similar, and boxplots were used to plot the display the parameter values for visual comparison of quartiles and median values. The values for ‘seldepth’ were plotted with a normal range y axis of values and the ‘time’ and ‘nodes’ parameters plotted using a Log range y axis to allow the data to be read with granularity with the skewed nature of the values.

## 4.3 Opening Analysis

## 4.4 Rating distribution investigation

Section 2 - Analysis of Openings

Excluded those that ECO code not joined on as Lichess ads some descriptions to non-descript openings

Chapter 5: Implementation

# Chapter 6: Results

## 6.1 Puzzle Analysis

Table 1 - Result of Engines Solving Mate in X Problems

|  |  |  |
| --- | --- | --- |
| **Mate Puzzle Solved?** | **Stockfish** | **Leela Chess Zero** |
| Yes | 3046 | 3023 |
| No | 1 | 24 |

Here Stockfish shows an indication that it has a better overall ability to overcome search horizons more effectively by solving all but one of the Mate in X puzzles provided to it. Leela Chess Zero also performed well given the task, solving all but 0.8% of the puzzles, with one of the unsolved being the same as the one Stockfish did not manage to solve.

Table 2 - Result of Shapiro-Wilk test for Stockfish and Leela Chess Zero Evaluation ‘time’

|  |  |  |
| --- | --- | --- |
| **Mate in X** | **Stockfish (Statistic, P-Value)** | **Leela Chess Zero**  **(Statistic, P-Value)** |
| 5 | (0.06541, <0.001) | (0.24242, <0.001) |
| 6 | (0.13197, <0.001) | (0.28906, <0.001) |
| 7 | (0.37938, <0.001) | (0.35686, <0.001) |
| 8 | (0.59305, <0.001) | (0.40843, <0.001) |

Table 3 - Result of Whitney-Mann U test for Parameter Comparison

|  |  |  |  |
| --- | --- | --- | --- |
| **Mate in X** | **Time**  **Stockfish < Leela Chess Zero (Statistic, P-Value)** | **SelDepth**  **Stockfish < Leela Chess Zero**  **(Statistic, P-Value)** | **Nodes**  **Stockfish > Leela Chess Zero**  **(Statistic, P-Value)** |
| 5 | (171289, <0.001) | (1714897, <0.001) | (4305251, <0.001) |
| 6 | (19794.5, <0.001) | (89463.5, <0.001) | (195024.5, <0.001) |
| 7 | (7321, <0.001) | (12230.5, <0.001) | (27207.5, <0.001) |
| 8 | (299, <0.001) | (490, 0.002) | (1135, <0.001) |

The Shapiro-Wilk test uses the Null Hypothesis that a distribution follows a normal distribution. The results of the Shapiro-Wilk test displayed in Table 1 for Stockfish parameter normality shows a consistent result of P-value < 0.05 wherein the Null Hypothesis can be rejected such that the data is considered non-normal. Non-normal distributions are seen in Table 2, where a P-value < 0.05 indicates the distribution of ‘time’ values obtained from the engine analysis by Leela Chess Zero can be compared with Stockfish ‘time’ values using a non-parametric test.

With the results seen in Table 2 indicating non-normal distribution of values, the Mann-Whitney U test was used. The results can be seen in Table 3 where a collection of test statistics and P-values are displayed for each engine parameter collected between the two chess engines for each of the categorical ‘Mate in X’ values. In these results it can be seen that Stockfish outperforms Leela Chess Zero in terms of a reduction lower time needed to solve, a lower overall depth of search required to come to a ‘Mate’ conclusions and a higher volume of nodes searched within the time of analysis. The ‘time’ and ‘seldepth’ are indicative of higher performance with respect to search horizon problems, whereas the ‘nodes’ difference is expected given the different search structure of Alpha-Beta Pruning algorithms versus Monte-Carlo Tree Searches.

The combination of a both higher solve rate and more efficient search parameters indicates that Stockfish has an overall better performance with respect to the Horizon Effect, making it the choice engine for this project in analysing openings and their effect on a chess game.

# Chapter 7: Discussion

# Chapter 8: Conclusion

# Appendix A: Workflow

# Appendix C: Data Permissions

# Reference List