# Chess Opening Analyzer

## PM4, Adrian Obermühlner, Freja Rasmussen

## Problem and Novelty

Chess is a very popular game, primarily played online nowadays. As a result, there is an increasing number of tools available to help players improve in specific aspects of the game. The endgame, which is the final part of a game with few pieces left on the board, was one of the first game segments to be "solved"1. Players can practice tactical aspects of the game, which involve a sequence of moves that provide a significant advantage, through Chess Puzzles2. While there are numerous courses and methods to explore existing chess openings3, no tool specifically analyzes and directly informs players about the flaws in their opening strategies.The lack of such a tool can be attributed to various factors, including the financial implications of analyzing numerous games and chess positions, which can be computationally expensive and slow.

Our goal is to create such a tool and optimize it in order for it to run smoothly and relatively fast.

Our opening analyzer aims to address two main challenges: first, to cluster openings, even with mistakes or deviations, into categories that are understandable to users, identify mistakes in the opening and display the results in an interactive and understandable way. Second, to find the most efficient method to compute many chess positions.

## Idea

The solution of the proposed problem will be an application which allows the user to upload their games through either “chess.com”’s4 or “lichess.org”’s5 APIs. Chess games are stored in the PGN, a standard plain text format that stores all the game moves as well as metadata of the game6. Each game will be shortened to the first 15 moves, the opening of a game, while keeping some of the metadata as well. The analysis is going to be twofold, one part is going to be classifying the opening according to the standard opening classification7, as well as tracking if and which variation of the opening it is. The second analysis will be a position validation to determine where mistakes happen using the chess engine Stockfish8. The results from the analyses can be used to determine common mistakes, strengths and weaknesses.

The data will subsequently be made available to the dashboard. It will display different plots and rankings containing strong and weak openings, for individual openings it will show at which move mistakes are commonly made and where deviations from book play happen.

## Methodology

Our initial approach will be to analyze all the positions from a given player's games and validate them using a linear method with Stockfish, which will be slow. To improve the speed of the analysis we will try two approaches.

The two proposed solutions are to parallel processing or itemization of prior knowledge. To parallel process the analysis with Stockfish we would have to use a Big Data framework, such as Spark9.

We will compare this method with another approach that saves every sequence of moves we have already seen in an indexed table in a database. If a sequence of moves has been played before, we will use the previously calculated value. If the position is new, we will use Stockfish to calculate the position. We will compare both solutions and implement the optimal solution in our final product.

## Plan

Our main goal is to have a working application which is able to import data from the chosen API’s (*Chess.com* and *Lichess.org*), analyze the opening sequence with focus on opening classification, deviation from book play and identification of mistakes, displaying the result in a user friendly dashboard. We aim for our solution to be fast and not have a long waiting time between upload and finished analysis. We plan to execute the project as listed in the table below.

| **Timeframe** | **What** | **Description** |
| --- | --- | --- |
| 28.02 - 13.03 | Setup of project | Initial setup, deciding on techstack, planning of project, first presentation, write proposal, testing out data analysis |
| 14.03 - 03.04 | First solution | Implement first solution idea, test how long application needs for analysis |
| 04.04 - 17.04 | Itemization | Implement a bigger database with all analyzed games, itemize these to improve on initial solution speed, give second presentation |
| 18.04 - 01.05 | Parallelization | Implement parallelization to gain speed in data analysis and import |
| 02.05 - 24.05 | Final product | Comparison of the three previous solutions, implement a final improved solution which contains the best part of each of the other ones, give presentation and write final report |

If we are faster than predicted, we would like to add a similar analysis, as there will be for one player, for groups of players with similar play level/ ELO rating10. We might be able to implement this while working on our “itemization” approach, as we would create this database at that point.

## Sources

1. Andrist, R. B. & Haworth, G. MC. Deeper model endgame analysis. *Theor. Comput. Sci.* **349**, 158–167 (2005).

2. Chess Puzzle on lichess.org. *lichess.org* https://lichess.org/training/f5lud.

3. De Marzo, G. & Servedio, V. D. P. Quantifying the complexity and similarity of chess openings using online chess community data. *Sci. Rep.* **13**, 5327 (2023).

4. Chess.com. *Chess.com* https://www.chess.com/ (2024).

5. Lichess. *lichess.org* https://lichess.org (2024).

6. Edwards, S. J. Standard: Portable Game Notation Specification and Implementation Guide. (1993).

7. Matanović, A., Rabar, B., Molerović, M. & Abramov, L. *Encyclopaedia of Chess Openings: V. B (2nd Edition)*. (Chess Informant, Beograd, 1984).

8. Romstad, T. *et al.* Stockfish - Open Source Chess Engine. (2011).

9. Zaharia, M. *et al.* Apache Spark: a unified engine for big data processing. *Commun. ACM* **59**, 56–65 (2016).

10. Elo, A. E. *The USCF Rating System: Its Development, Theory, and Applications*. (United States Chess Federation, 1966).