TILBURG UNIVERSITY

Predicting chess rating based on a single game

Tim Tijhuis

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES OF TILBURG UNIVERSITY

Word count: 8499

STUDENT NUMBER

2075049

COMMITTEE

dr. Paris Mavromoustakos Blom
dr. Fred Blain

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

December 2, 2022

Predicting chess rating based on a single game

Tim Tijhuis

Abstract

The goal of this thesis is to predict a players' chess rating based on the moves in a single game. In chess, both online as in person, players are matched with an opponent based on their rating. The rating of a player is represented by a number, which is derived from the results of all previously played matches within a pool of players. When a player joins a new group, many matches must be played to get a rating that is representative of their relative strength. This thesis aims to solve this problem by using machine learning to predict rating after a single game. The dataset consists of 457,464 games, which initially only consists of the name of the player, combined with their rating, the colour of the pieces, and the algebraic notation containing the moves played. From the algebraic notation a total of 30 features are extracted that can be an indication of a players' relative chess strength. The dataset was then ordered based on rating and split into 10 equally large bins. To find the best model for predicting chess rating, this thesis compares two machine learning models. A random forest classifier and SVM model were tested on a subset of the data. Both models made predictions with a similar accuracy, but the random forest classifier was chosen based on its faster computation speed. This model was run for several classification tasks, to analyse the effect of bin width and the rating range between the two bins. As expected, the accuracy went down when more bins were added and the rating range between the two classes got smaller. The model could predict with 79.29% accuracy which class a player belongs to, when predicting whether a game belonged to the weakest or strongest 10%. The accuracy went down to 64.26% when the predictions were based on the top and bottom half.

1. Data Source/Code/Ethics Statement

Data for this thesis comes from the public database of the online chess platform *Lichess*, which can be found here: https://database.lichess.org/. Lichess remains the owner of the data and the data does not contain any personal information.

2. Project Definition

Chess is a popular board game that is played all over the world and has many business models built around it. In example, online chess platforms, chess clubs and organizations, and the (inter)national chess tournaments involving prize money. In an interview Erik Allebest, the CEO of *chess.com* (the most popular online chess platform), states that the annual income of his chess platform is "… somewhere between 50 and a 100 million" (Allerbest, 2021). The prize money for the 2019 world chess championship was a total of $1,600,000 (Lewis, 2021), and according to the website of the Dutch chess federation, there are a total of 435 chess organizations in the Netherlands (Structuur, 2022). Each organization and online chess platform measures chess rating based on the players within their pool. The goal this thesis is to help chess organizations and players by predicting chess rating based on a single game. This chapter will clarify what chess rating is and what the benefit can be of being able to predict chess rating based on a single game, after which the societal and scientific relevance of the thesis will be explained. The theory behind the selection of moves that can be indicative of skill will be explained in the literature review.

2.1 Chess Rating

In chess, rating systems are used to create a ranking list of players in a pool and calculate their relative strength within that pool. Each player is assigned a rating, that is represented by a number which normally ranges from 0 to 3000 (Glickman & Jones, 1999). A rating above 3000 can be achieved, but rarely ever happens. The rating number is derived from the results of a player's previously played games. The rating of players can be an accurate indicator of their winning chances when matched against each other. The advantage of using a rating system is that the skill between two players in the same pool can be compared without them ever having played a game against each other.

The most famous chess rating system is called the Elo rating system, invented by Arpad Elo, and adapted by the World Chess Federation (FIDE) in 1970 (Elo,

1986) . Almost all chess organizations still use the Elo rating system or a variation of the Elo system (Glickman & Jones, 1999). In this thesis, we will attempt to predict the ratings of players of an online chess application. In particular, the online chess platform called *Lichess* (http://lichess.org). Lichess is a popular online chess platform with open access for players all over the world. The data used in this thesis are from chess games played on Lichess, which uses a variation of the Elo system called “Glicko-2”. In this proposal the word rating will refer to the Glicko-2 rating system, unless otherwise specified.

Players with a similar rating can be considered to be of the same relative strength. Therefore, the FIDE organizes chess tournaments based on ranges of Elo rating of about 200 (i.e., 1400-1600). In these tournaments it is not necessarily the case that the highest rated player will win the tournament. For this thesis, the players will be ordered based on their rating and divided into 10 equally large groups. The rating category will be used as the target variable. This means that the prediction can be based on classifying relative strength and not predicting the exact rating number.

As mentioned before, chess rating is based on relative strength in a pool of players. Therefore, the rating of a player is not transferable to a different pool of players. The problem with this system is that a player joining a new group will have to play many games to be assigned a rating representative of their strength (Zhang, et al., 2022). Playing against an opponent of disproportional strength is proven to have a negative effect on how much a player enjoys the game (Zhang, et al., 2022).

2.2 Motivation & Relevance

The societal relevance of this thesis is being able to predict a player’s chess rating relative to a pool of players by analysing their moves in a single game. This enables rapid and efficient estimation of an unknown player's strength within an unknown player pool, which can prevent playing matches against an opponent of disproportional strength. Because the data for this thesis consists only of games played on Lichess, the strength of a player will be predicted relative to players on Lichess. However, if the model would be trained on a different pool of players, then a single game of an unknown player can be used to classify the relative strength within that pool.

Even though computers playing games is a thoroughly studied subject in artificial intelligence (AI), there currently is very little research in predicting the game-related strength of a player. The scientific relevance of this thesis is to fill the literature gap by using machine learning to predict chess rating based on a single game. While the outcome will show what moves are most indicative of relative strength, it will also show to what extent chess strategy is applied in moves. This can be used for lower rated players to identify what moves higher rated players make and interpret these to become a better chess player.

3. Literature review

The literature review of this proposal is made up of two parts. The first part of the literature review will define the context of chess. Here an explanation will be given of the basics of chess, what phases a chess game consists of, and how chess notation works and how it is related to this research. Following the context of chess will be the related work. The related work consists of three parts. First the development of AI in chess will be explained and how the use of heuristics can be used in this thesis. Then literature of rating prediction in other games will be investigated, followed by chess theory and strategies that can be used to determine what moves can be indicative of a players' chess rating.

3.1 Context of chess

A chess board consists of 8 rows and columns, in which both sides (black and white) each have 16 pieces. The row and column names are viewed from the perspective of the player with the white pieces and each square is a combination of the column letter (from a to h) and row number (from 1 to 8). The names of the squares are important to understand chess notation, which describes all the moves made during a game.

The 16 pieces of each player consist of 6 different types. The different types of pieces are: 8 pawns, 2 rooks, 2 knight, 2 bishops, a king, and a queen. Each type of piece has their own rule to how they can be used and have their own value. The value of a piece is based on how useful they are and their ability to move on the chess board. Table 1 shows an overview of the pieces and their respective value. Notice that the king is not mentioned, because the king cannot be captured, and the game is over when the king can no longer make a legal move.

Piece	*Pawn*	*Rook*	*Knight*	*Bishop*	*Queen*
Value	1	5	3	3	9

Table 1 - Chess pieces and their value

Phases of a chess game

There are three phases of a chess game. The first is the opening, which is generally considered to be the first 10 moves. The goal of the opening is to develop pieces to a good position. While there aren't exact rules for what a good position is, there are some principles that chess players can follow. For example, controlling and defending the middle of the board with pawns. The middle game, the second phase, is where most pieces are exchanged, and it is played until only a few pieces are left. This is when the final phase, the end game, begins. Each phase has its own strategy, which will be used in this thesis to derive patterns and moves that can be indicative of a players' strength.

Chess notation

The moves of the games being studied for this research are registered in the algebraic notation. The algebraic notation is a method used to describe chess moves by combining the name of the piece that is being moved and the location where it is being moved to. Before the algebraic notation can be used to analyse a game, it is necessary to understand how it works.
Figure 1 is an image of a chess board with the algebraic notation of the first move for each player, namely "1. e4 e5" from a game on Lichess.

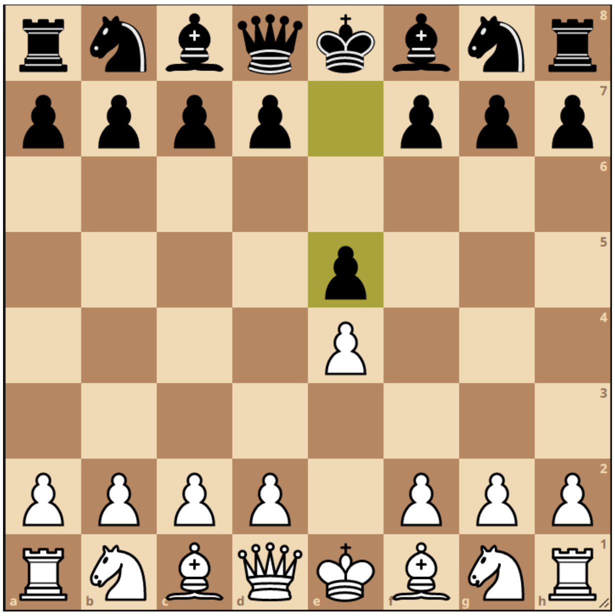


Figure 1 - Chess board with algebraic notation 1. e4 e5 (chess.com)

At any point during a game, the position of the board can be described using the FEN (Forsyth-Edwards Notation). FEN uses a single string to describe all the pieces and empty squares in a chess position. The FEN notation for the first move for each player, as show in figure 1, is “rnbqkbnr/pppp1ppp/8/4p3/4P3/8/PPPP1PPP/RNBQKBNR”. In FEN, the black pieces are described with lower-cased letters and capitalized letters are used for the white pieces. Empty squares are noted with a number, which is equal to the number of empty squares between two pieces.

This thesis will use both the algebraic notation and FEN to extract features. The advantage of the algebraic notation is that it can be used to analyse a sequence of moves and after how many moves a certain piece is used. For example, after how many moves is the king first moved. Using the algebraic notation, it is difficult to analyse the position of the board after a certain number of moves. This is where the FEN notation will be used. For example, to analyse the pawn positions of the board after the opening phase (first 10 moves).

3.2 Related Work

Development of AI in Chess

The term ‘Artificial Intelligence’ was first used in 1956 in the Dartmouth summer Research Project by John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude Shannon (Dick, 2019). McCarthy et al proposed their research based on the conjecture that “… every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it” (McCarthy, Minksy, Rochester, & Shannon, 1955). After the research project, the main approach of artificial intelligence was to identify the human decision process and try to mimic this in an automated way. In the past 30 years, game-playing agents have been created to defeat some of the best players in different categories of games. These categories include board games, card games, first person shooter games, and real time strategy games (Yin, et al., 2022). However, many of these algorithms are no longer based on mimicking human decision making, but instead learn the game by playing against themselves. This is often referred to as reinforcement learning (Yin, et al., 2022).

The same is true for chess-playing agents. Early chess-playing agents used a search engine and an evaluation function which was based on ideas from expert level chess players. Because chess is a turn-based game (game where players take turns), computers can use a search engine to compute all possible moves and the counter moves played by the opponent, if computationally allowed. In practice, however, chess has too many possible combinations, which makes it unfeasible for an algorithm to try every possible combination. The number of possible chess positions (without pawn promotion) is estimated to be around 10^{46} (Steinerberger, 2014) . Therefore, chess engines use an evaluation function to support the search engine and conclude what moves need no further investigation. In 1997, Deep Blue became the first chess algorithm to defeat the world champion, Garry Kasparov, in a six-game match (Campbell, Hoane Jr, & Hsu, 2022). Deep Blue used a search engine that could look at different moves in parallel, while using an evaluation function to determine when a search no longer needed to be investigated. To narrow down all the possible combination even more, Deep Blue used a database from Grandmaster chess players containing openings and end games in chess. While not all the details of the evaluation function are given, it is known that Deep Blue used about 8000 features to calculate the best move (Campbell, Hoane Jr, & Hsu, 2022). Some of these features are based on chess heuristics, such as placing the bishops on a strong diagonal or moving the rook to the 7th rank. With the increase of computational power, recent chess engines have been able to further improve their chess playing abilities. Stockfish, one of the best chess-playing agents, still uses similar search and evaluation methods as Deep Blue did (Maharaj, Polson, & Turk, 2021).

Related research shows that chess moves can be evaluated by using heuristics designed by expert level chess players. This thesis will use a similar type of heuristics based on chess theory to find moves and patterns that can be evaluated to predict a players' rating.

Rating prediction in games

Many online games use a matchmaking system to create teams of equal winning probabilities. The same is true for chess, which uses a rating system to find an opponent with a similar relative strength. Finding a fair match directly impacts the way a player enjoys a game, which is why Zhang et al propose the QuickSkill framework for skill estimation of new players in online games (Zhang, et al., 2022). The goal of QuickSkill is to improve a players' game experience by accurately predicting the matchmaking rate (MMR) of new players after only a few games. QuickSkill is a framework designed for multiplayer online battle arena (MOBA) games, such as Mobile Legends. In MOBA games, a players' skill is profiled by their statistics at the end of a game (Zhang, et al., 2022). The authors propose to take snapshots of the statistics every 3 minutes during a game to retrieve a more complete profile of a player.

For the game StarCraft II, Bialecki et al propose using data processing to find determinants of victory (Bialecki, Gajewski, Bialecki, & Phatak, 2022). They state that using descriptive data of the game can be used to predict the winner and to identify the main determinant of victory. This thesis will use a combination of the two methods. FEN notation will be used for the creation of snapshots at different points in a chess game, which can then be used to extract data based on moves and strategies that can be indicative of chess skill.

Chess strategy & Guidelines

As mentioned before, chess engines use heuristics to evaluate moves. While some of these heuristics will be used to predict the rating of a player, most will be taken from chess strategy books. An example of a heuristic from Deep Blue is having a rook on the seventh row (Campbell, Hoane Jr, & Hsu, 2022). Having control of this row with a rook is usually a strong position for the rook. Chess books describe moves, strategies and positions that are generally considered to be "good", such as controlling the centre and developing your pieces (Engqvist, 2016). Engqvist also mentions positions in chess that can either be a strength or a weakness, such as isolated pawns (a pawn without another pawn on either side). The features that will be extracted in this thesis are based on chess theory or are based on evaluations made by the chess engine Stockfish. Stockfish evaluates the position on the board after every move and calculates which player is winning based on centipawns. This means that if a player loses a pawn, it is down 100 points. Even if a player makes a bad move and doesn't lose a piece, the engine can calculate the decrease in centipawns of the position. The evaluation value from stockfish will be used to determine when a player makes a mistake or blunders.

This thesis aims to predict chess rating based on a single game by using a machine learning model. To do so it is important to use appropriate machine learning model. To find the best machine learning model, the first research question is:

1. What models most accurately classify chess rating?

The evaluation of the model will be based on prediction accuracy after extracting the features. A total of thirty features are extracted for this thesis. To find out which features are most important for the predictions, the second research question is:

2. Which patterns in chess gameplay are most indicative of a player's rating?

Once these patterns are extracted from the dataset and used as features in a machine learning model, it will become clear what moves and patterns in chess are most indicative of chess rating. To measure the effect of bin width on the prediction accuracy, we formulate the final research question as:

3. To what extent is the accuracy of the models affected by the width of the rating categorization window?

4. Methodology

This section will describe the methodology used in this research, from dataset selection to model training and chess rating prediction, to achieve the goal of predicting a players' chess rating based on a single game. First, the dataset and the extracted features will be described, after which each research question and the methods used to answer them will be explained. Figure 2 contains a flowchart summarizing each step and the methods used for this thesis.

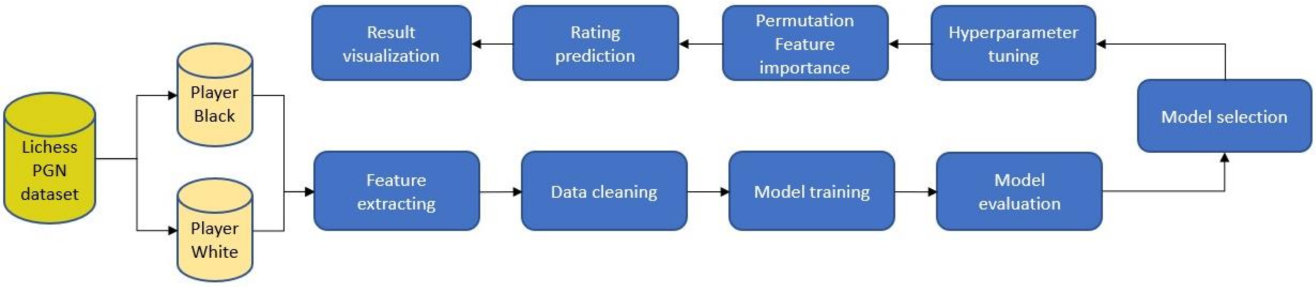


Figure 2 – Methodology Flowchart

4.1 Dataset

The data used in this research consists of two consecutive Portable Game Notation (PGN) files downloaded from Lichess. Each file contains all the games played on Lichess for one month. The two files combine for a total of 240.000 games, which is then split into games played by black and white. Only four features are used from the PGN files to predict the rating of a player. These features are the name of the player, the rating, the colour of the pieces used by the player, and the algebraic notation containing all the moves played. Other features from the PGN files, such as time control (the amount of time each player gets to make their moves in a game) and the outcome of the match, are not used. These features are considered neutral information and can give a false indication of rating. Time control is considered neutral in this thesis because players have different ratings per time control. A player can be rated 1800 for 10-minute games, while only having a rating of 1100 in 2-minute games. The outcome of the match is not used because there always is an outcome no matter what rating the player is. A player can play a very good match and still lose because the opponent is better, while another player can make bad moves but still win.

From the algebraic notation, a total of 30 features are extracted and added to the dataset. Some of these features are measured at four timestamps during a game, based on the length of the game. The four timestamps are after 25, 50, 75 and 100 percent of the total moves played. Table 2 contains a description of the features that are extracted for this research. The features have been extracted based on positions or moves that are described in chess literature. The features contain moves or positions which chess literature considers good, while other features describe moves or positions that should be avoided.

4.1.1 Feature Description

The first feature is ‘Game Length’, which is the total number of moves played in a game. The game length is used to find the position on the board after 25, 50, and 75 percent of the moves have been played. These positions, together with the position at the end of the game, are used to determine the pawn- and mobility-related features.

Pawn positions are evaluated based on a list, which counts the total number of pawns a player has on each file (columns a-h). From this the list isolated pawns, doubled pawns, and tripled pawns are taken as features. The mobility of a player is calculated by taking the total number of legal moves from the player and subtracting the total number of legal moves of the opponent.

The feature called defending centre is used to describe how many knights, bishops, and pawns directly attack or defend the centre of the board after 5 moves. The fifth move is used here based on the principle that the opening moves should be made to defend the centre (Engqvist, 2016). Even though the opening can be considered to consist of more than 5 moves, this thesis interprets this chess principle as the first 5. After the fifth move, the opening can focus on many other areas of the chess board to gain a winning position. “Pieces moved” is used to describe the total number of pieces moved in the first 10 moves. This feature is based on a common rule of thumb in chess, which is not to move the same piece twice during the first 10 moves. This number ranges from 1 to 11, because castling moves two pieces in a single move.

Moves before castling, first knight on edge, and rook on 7th indicate at which move the player performs that move for first time. These features are extracted by looking at all the moves made by a player in the algebraic notation and looking for a notation that describes castling, moving the knight to the edge of the board, or moving the rook to the 7th rank. The feature total knights on edge is a summation of the total amount of times a player moves the knight to the edge of the board.

The final features are blunders, first blunder, mistakes, first mistake, first mate given and first mate opportunity. Blunders, mistakes, and mate opportunities are based on calculations by the stockfish engine with depth set to 8. This means that the stockfish engine looks 8 moves ahead to evaluate a position. As mentioned in the literature review, stockfish evaluates chess positions and who is in a winning position based on centipawns. Stockfish considers white to have a small advantage at the beginning of the game and updates the centipawn advantage or disadvantage after every move. This research uses the change in the stockfish evaluation to determine when a mistake or blunder has been made.

There is no clear centipawn value for mistakes or blunders in chess theory. This thesis considers a mistake a move as a mistake when the move negatively changes the centipawn value by more than 100 points, while a blunder is considered as a move that negatively changes the centipawn value by more than 300 points. The values for mistakes and blunders are based on the value of a pawn (100 points) and the minimum value of a major piece (300 for a knight or bishop).

First mate opportunity and first mate given are determined by the move where the engine first finds a mate possibility within 3 moves. When stockfish finds the possibility to forcefully checkmate the opponent in 3 moves, the first mate possibility will be the current move plus three.

Feature	Description
Game Length	The total number of moves made during a game.
Moves before Castling	The number of moves made by a player before the players *Castles* ("moving one's king two squares toward a rook on the same rank and then moving the rook to the square that the king passed over") (Castling, n.d.)
Isolated Pawns	A position where a pawn has no neighbouring pawn on either side. (This feature is measured four times during the game)
Doubled pawns	A position where a player has two of his own pawns on the same file. (This feature is measured four times during the game)
Tripled pawns	A position where a player has three of his own pawns on the same file. (This feature is measured four times during the game)
Knights or Bishops	A check to see whether a player moved both of the bishops, both of the knights first, or did not move the two bishops or knights in the game.
Defending Centre	The number of pieces (excluding rooks, the queen, and the king) that directly attack or defend the four centre squares after the first 5 moves.
Pieces Moved	The total number of pieces moved after 10 moves. The maximum here is 11 because a player making a castling move, moves 2 pieces in 1 move.
Blunders	The total number of blunders made by a player.
First Blunder	The move where the first blunder is made by a player.
Mistakes	The total number of mistakes made by a player.
First Mistake	The move where the first mistake is made by a player.
First Mate Opportunity	The move at which the player had the first opportunity to win the game within 3 moves.
First Mate given	The move after which the opponent for the first time had the opportunity to win the game within 3 moves.
Mobility	The difference in total number of legal moves compared to the opponent. (This feature is measured four times during the game)
First Knight on Edge	The move after which the player put a knight on the edge of the board for the first time.
Total Knights on Edge	The total amount of times a player put a knight on the edge of the board.
Rook On 7th	The first time a player moves the rook to the 7th rank (this is the 2nd rank for the player playing with the black pieces)

Table 2 - Extracted features for rating prediction

4.1.2 Data cleaning

The last step before implementing the machine learning models is cleaning the data. Even though almost all the games from the Lichess PGN files are in the same format, there are two exceptions, which have been removed from the dataset. First, 1,290 rows (0.27% of dataset) without a player name or rating were deleted. Then, 7,473 rows (1.56% of remaining dataset) were deleted because the algebraic notation contained evaluation after every move. These rows were deleted because they need a different approach for extracting

features.
Finally, 13,773 (2.92% of remaining dataset) games were removed for players which have not played at least 10 games total. As mentioned in Chapter 2, chess rating is calculated over the result of all previous games of a player. It can take many games before a player has a rating that is representative of their relative strength. Therefore, this research uses a minimal threshold of 10 games. This minimizes the number of games where a player is playing with an inaccurate rating. The final dataset, after removing the rows, consists of 457,464 games. The last step for the dataset is to group the players into 10 bins, based on their rating. Table 3 is an overview of the rating range per bin.

Bin	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Rating Range	800-1339	1340-1429	1430-1492	1493-1541	1542-1593	1594-1647	1648-1706	1706-1773	1774-1870	1871-2341

Table 3 - Rating range per bin

4.2 Model selection & Evaluation

The goal of research question 1 is to find the best model for predicting chess rating. To answer this research question, this thesis compares two machine learning models based on their accuracy in predicting chess rating. The machine learning models used in this research are Random Forest (RF) and Support Vector Machines (SVM). The RF model was chosen because it has shown to perform adequately in predicting skill learning in MOBA games (Aung, et al., 2018), and the SVM model was chosen because it has shown to perform adequately in predicting the results of a match in the MOBA game Dota 2 (Anshori, Mar'i, Alauddin, & Bachtiar, 2019). The decision was made not to use deep learning models such as Convolutional Neural Networks, because they do not initially have the option to compute the important features.

This thesis uses bins as target variables for predicting rating. The players are ordered based on their rating and divided into 10 equally large bins, each containing 10% of the dataset. To evaluate the RF and SVM model, this thesis turns the classification problem into 5 binary classification problems, by considering datapoints from the top and bottom N bins. N is a range from 1 to 5, where the final classification problem predicts whether a player belongs to the bottom 5, or the top 5 bins. For feasibility reasons, the models are evaluated on a subset of 80,000 rows. For each task, the worst and best rated players are represented equally with a total of 40,000 rows. Figure 3 is an example of how the first 2 tasks are split in N bins (1 and 2). The model predicts whether a player belongs to the lower rated players (class 1) or the higher rated players (class 2).

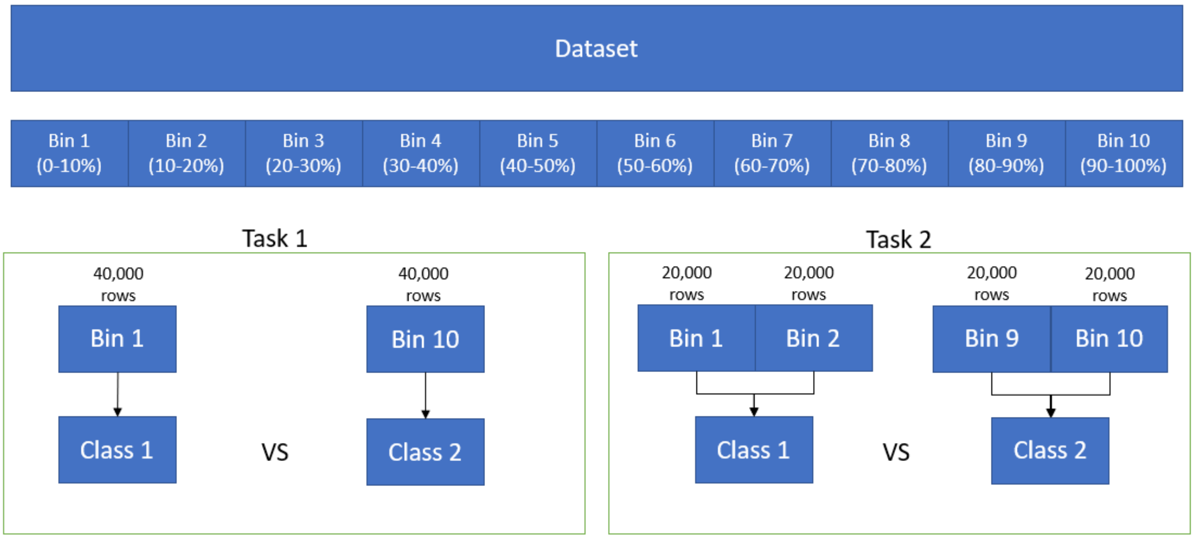


Figure 3 - Dataset split for model selection

The final task contains 8,000 rows from each bin, and the model tries to predict whether a player belongs to class 1 (containing the bottom 5 bins) or class 2 (containing the top 5 bins). For each task the models are run with a 5-fold cross-validation. The models are then compared based on the average classification accuracy for each of the five subsets. The cross-validation method ensures that the model does not test on data that it has seen before. By averaging the results on 5 folds, random outliers in model accuracy are prevented and ensure that new data will be predicted with a similar accuracy. After the first research question is answered and the best performing model is found, the model will be optimized using hyperparameter tuning. The optimized model will then be used to answer the remaining research questions.

Hyperparameter tuning of the best performing model is done in two steps. The first step contains a parameter grid with a wide variety of values. This grid will then be randomly searched through in a 5-fold cross validation. The result from this random search contains the values that contain the highest prediction accuracy. These values will then be used for the final step of hyperparameter tuning, where a new grid is made. The new grid contains only values close to the outcome of the random search. For this grid all possible combinations are used in a 5-fold cross validation on the entire dataset.

4.3 Feature Importance

The RF classifier provides two methods that indicate how valuable a feature is for a prediction. These methods are feature importance based on mean decrease of impurity, and permutation feature importance.
The first method uses mean decrease in impurity (MDI) to calculate what features contribute most to the prediction accuracy. MDI, also called Gini importance (GI), is derived from the training of the RF classifier, and calculates how much information is gained after each split of the decision tree. While this method can be a good indication of feature importance, it is also known to be strongly biased.

Research shows that permutation feature importance allows feature importance to be measured without bias (Altmann, André, Tolosi, Sander, & Langauer, 2010). Therefore, this research uses permutation feature importance to calculate what features are most indicative of chess rating. Permutation feature importance measures the importance of a feature by randomly shuffling all the feature values and measuring the change in accuracy. To control for randomness the permutation feature importance is run 10 times for each feature to calculate the average change in accuracy.

4.4 Bin width & Rating gap

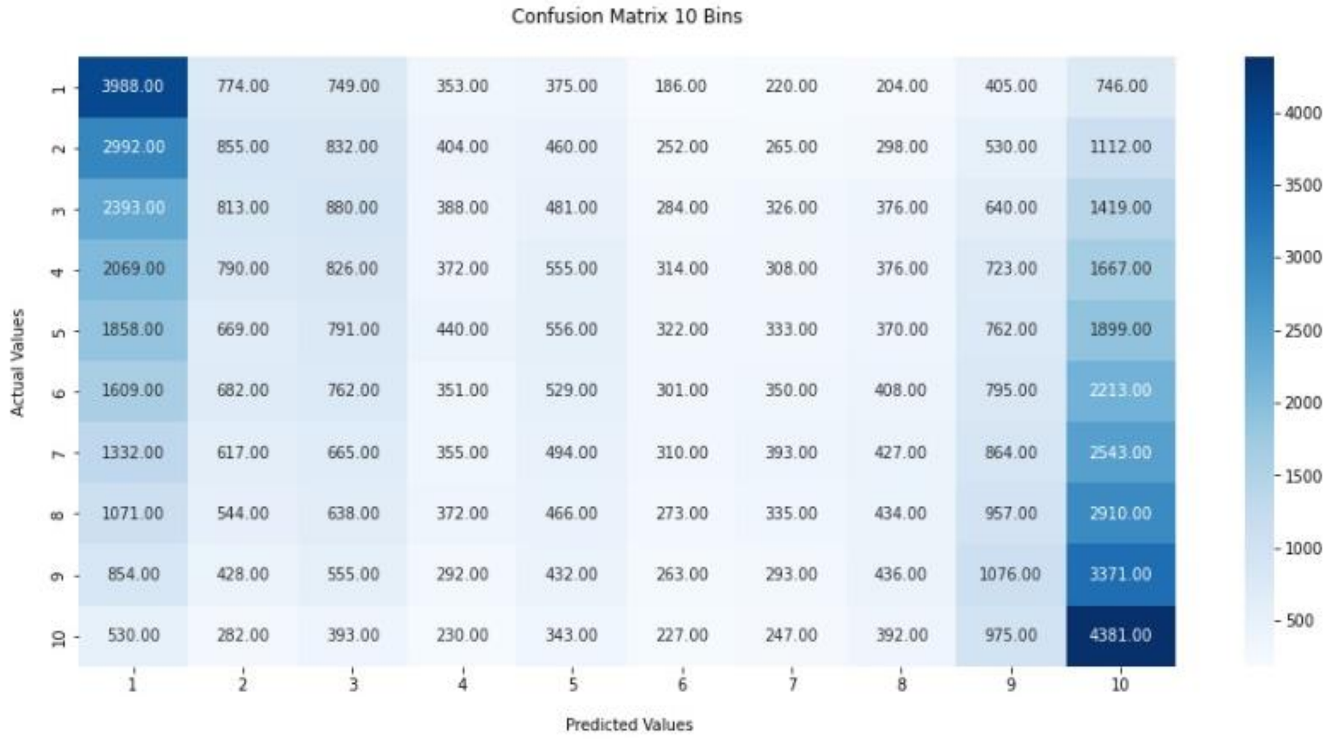
The third and final research question aims to find out to what extent the accuracy is affected by the width of the rating categorization windows. In addition to this, the effect of the rating range between the bins is also measured. The effect of bin width is measured by the same 5 tasks as explained in the model selection, but this time on the entire dataset. The effect of the rating gap is measured by starting with the middle bins (bin 5 and 6) and increasing the bin width for each task. Here the accuracy is expected to increase as more bins are added.

5. Results

5.1 Model Comparison

The two machine learning models that have been tested for this thesis are RF and SVM. The results of the models are compared based on their accuracy in predicting which rating bin a player belongs to. The accuracy is compared to the baseline, which is the accuracy of random guesswork. The models are compared on a subset of the data consisting of 80,000 games, where each bin is equally represented.

Initially the models were tested on predicting rating based on all 10 bins, making it a 10-class classification problem. Each bin contains 10 percent of the entire dataset. Every task for each of the models was performed with a 10-fold validation, where the 80,000 rows were split into 10 different training and testing sets. This way the train and the test data do not contain the same games. Both models were able to predict the rating with an accuracy slightly below 17 percent based on a single game. The predicted values and actual values of the RF model are visualized in the confusion matrix in Figure 4. The confusion matrix shows that most of the predictions are based on either the worst of the best bin and the number of predictions in the middle bins are a lot lower. A 17 percent accuracy is a small improvement of the baseline (which is 10%), but the confusion matrix shows that many predictions are not far off from the actual value. This can be explained by the small difference in the range of the 10 percentiles. For example, the fifth bin ranges from 1542 to 1593 rating scores, while the sixth bin ranges from 1594 to 1648.



	1	2	3	4	5	6	7	8	9	10
1	3988.00	774.00	749.00	353.00	375.00	186.00	220.00	204.00	405.00	746.00
2	2992.00	855.00	832.00	404.00	460.00	252.00	265.00	298.00	530.00	1112.00
3	2393.00	813.00	880.00	388.00	481.00	284.00	326.00	376.00	640.00	1419.00
4	2069.00	790.00	826.00	372.00	555.00	314.00	308.00	376.00	723.00	1667.00
5	1858.00	669.00	791.00	440.00	556.00	322.00	333.00	370.00	762.00	1899.00
6	1609.00	682.00	762.00	351.00	529.00	301.00	350.00	408.00	795.00	2213.00
7	1332.00	617.00	665.00	355.00	494.00	310.00	393.00	427.00	864.00	2543.00
8	1071.00	544.00	638.00	372.00	466.00	273.00	335.00	434.00	957.00	2910.00
9	854.00	428.00	555.00	292.00	432.00	263.00	293.00	436.00	1076.00	3371.00
10	530.00	282.00	393.00	230.00	343.00	227.00	247.00	392.00	975.00	4381.00

Figure 4 - Confusion matrix based on 10 bins

To make the difference between the bins larger, the model comparison has been done on a binary classification problem. This approach was chosen to see how the models perform in an iterative fashion, starting from the top and bottom 10% and expanding the two groups with 10% with each step. The models are compared on their ability to predict whether a player belongs to the weakest or the strongest bin. This task has been split into 5 tasks, where the bin width is increased gradually. Firstly, the weakest and strongest 10 percent are compared and for every task another 10 percent has been added to each group. In example, the second task classifies whether a player belongs to the weakest or strongest 20 percent, until the final task predicts whether a player belongs to the top or bottom 50 percent. Each task has been run 10 times and the models are compared based on their average accuracy for each task. The results of both models are very similar, as can be seen in the boxplot in figure 5.

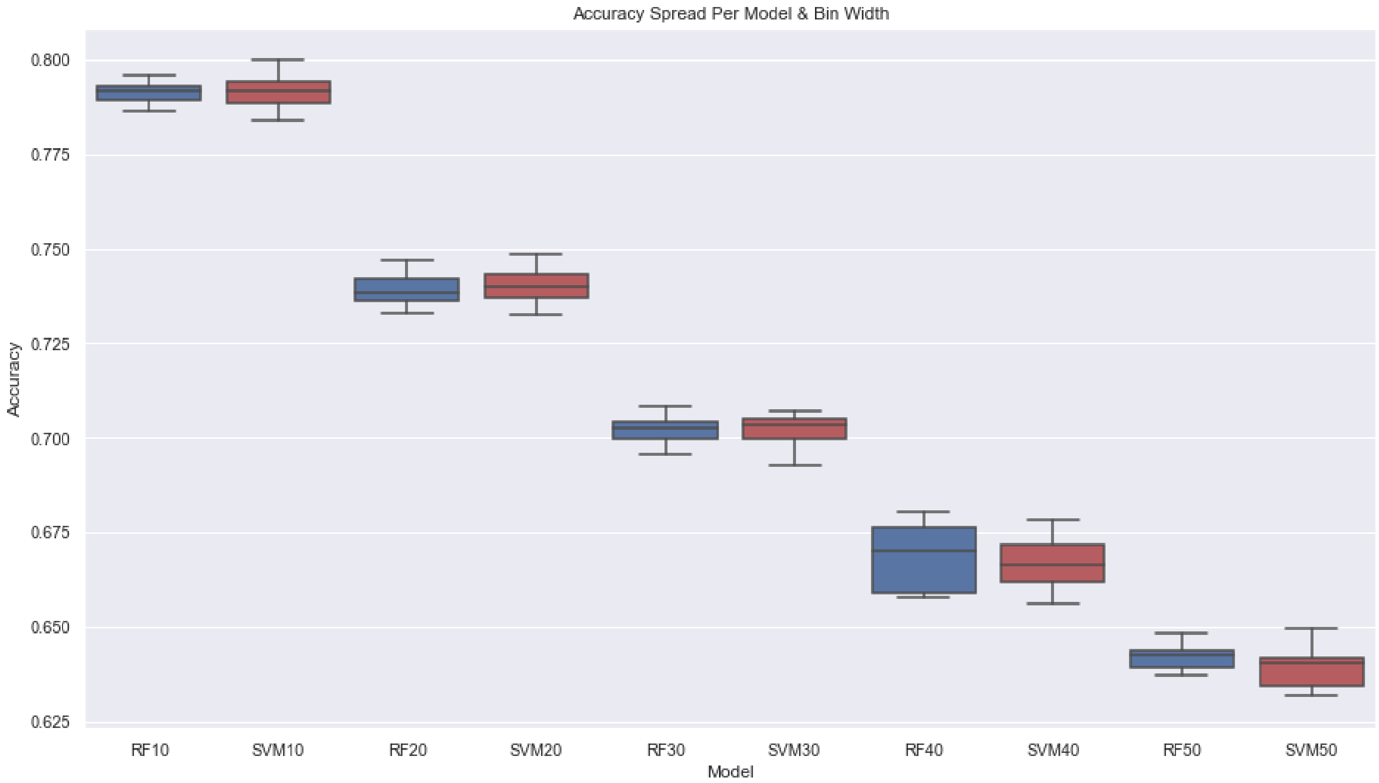


Figure 5 - Accuracy spread per model & bin width

The average accuracy on the lowest and highest rated 10 percent is 79.11% for the RF model and 79.15% for the SVM model. The accuracy goes down when the size of the bins increases, with a final average accuracy of 64.21% for the RF model and 63.96% for the SVM model. The accuracy for the models on the first task is relatively high, where the model can accurately classify the right bin almost 4 out of 5 times. Each iteration, adding another bin to the group, naturally decreases the prediction power. The high accuracy for the first task and the drop in accuracy every iteration after were to be expected, since the gap in rating becomes smaller. While the accuracies of the two models are very similar, there is a big difference in the computation time. The RF model needs 46 seconds to make predictions, while the SVM model needs 37 minutes when running the same task. The computation time is also based on the subset of 80,000 rows.

5.2 Feature Importance Analysis

As explained in chapter 4, this thesis uses permutation feature importance to analyse what features are most indicative of a players' chess rating. Initially this analysis has been performed on the entire dataset, where the rating values have been replaced by the numbers 1 through 10. These numbers are an indication of the percentile bin of the rating. However, the important features and the degree of importance is dependent on the size of the bin. Figure 6 shows the feature importance of the 10-bin classification model.

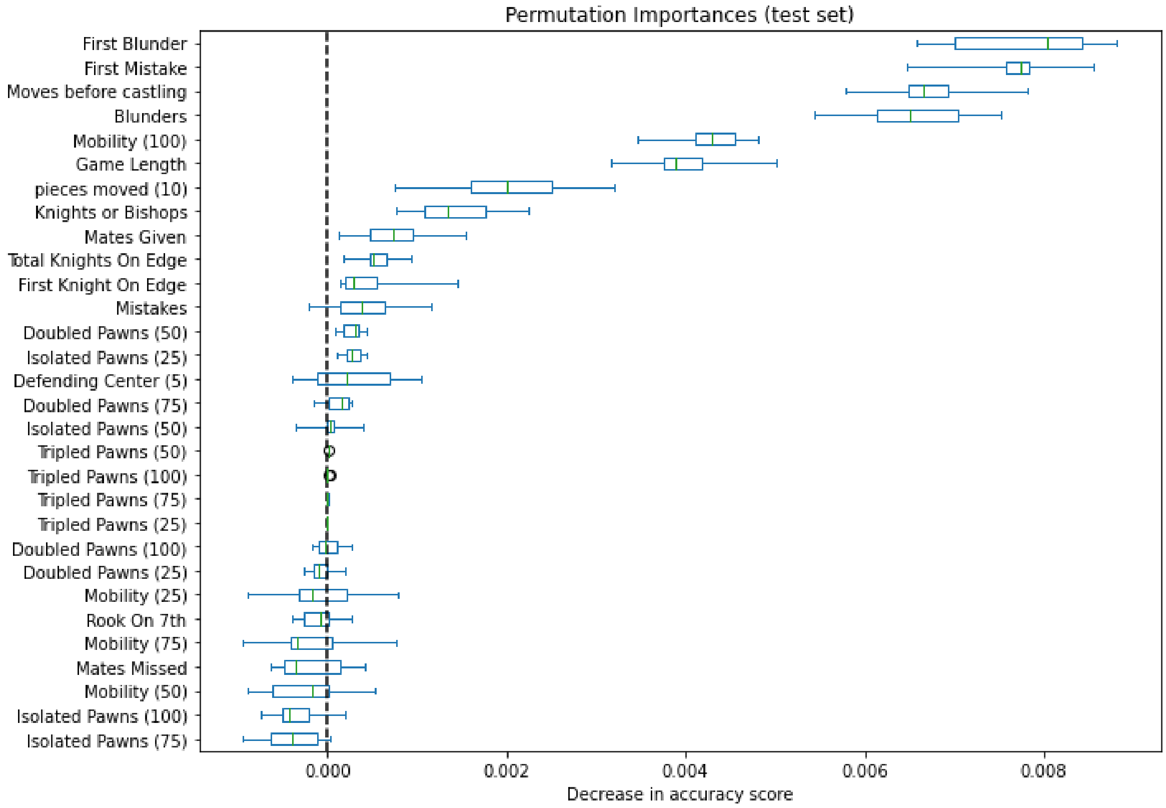


Figure 6 - Feature importance 10 -class classification

The values on the x-axis in the image represent the decrease in accuracy if the values of the feature are randomly shuffled. A value of 1 would mean the accuracy decreases by 100%. The values for this model are between -0.002 and 0.008, which means removing the most important feature would decrease the accuracy by about 0.8%. There are 6 features that stand out from the rest. These are the first blunder, first mistake, number of moves before castling, total amount of blunders, mobility (100), and game length. Their importance for the 10-bin classification model ranges between 0.004 and 0.008 on average. This is based on the RF model with an 17% prediction accuracy, when all 10 bins were considered in a 10-class classification.

These values for the features change based on the accuracy of the model for each task. When the classification task was changed to a two-class classification problem (high vs. low rating), the feature importance order was relatively similar. The six most important features are the same, but in a different order. The most important feature is still the first blunder, followed by blunders, first mistake, game length, moves before castling, and mobility (100). For that model the first blunder has an average impact of around 3.2%, while the prediction accuracy for this model is close to 80%. Table 4 shows the average value per bin for each of the six most important features.

Feature/ Bin	*First Blunder*	*First Mistake*	*Moves Before Castling*	*Total Blunders*	*Mobility (100)*	*Game Length*
1	10.81	6.12	6.32	3.66	-8.51	30.23
2	11.94	6.85	7.39	3.33	-3.65	32.20
3	12.76	7.28	7.94	3.16	-1.94	33.37
4	13.08	7.56	7.97	3.05	-0.82	33.63
5	13.74	7.88	8.22	2.96	0.01	34.60
6	14.04	8.26	8.23	2.84	1.15	35.02
7	14.50	8.55	8.28	2.71	1.74	35.63
8	14.93	8.93	8.35	2.60	2.97	36.22
9	15.50	9.47	8.46	2.51	3.61	37.01
10	15.78	10.43	8.32	2.21	7.12	36.98

Table 4 - Average value per feature and per bin

Figure 6 also shows some features with a negative decrease in accuracy. This could mean that removing these features will improve the accuracy of the model. The feature importance for the binary classification does not have any negative values, with some exceptions that are slightly below 0. Below, the two-class classification is discussed in more depth, primarily focussing on the effect of bin width and the rating gap between the bins.

5.3 Bin width

This section will look at the effect of the bin width on prediction accuracy. Results from Chapter 5.1 show that the increase in bin width, on a subset of the data, causes a decrease in accuracy for the two models before hyperparameter tuning. This chapter uses the RF model after hyperparameter tuning, which is trained and tested on the entire dataset, to measure the effect of the bin width and the rating range between the bins. Table 5 shows the bins that are considered and the accuracy of the model.

Bins considered	1 or 10	1-2 or 9-10	1-3 or 8-10	1-4 or 7-10	1-5 or 6-10
Accuracy	79.29%	74.44%	70.48%	67.21%	64.26%

Table 5 - Prediction accuracy RF model per bin width

The results show that increasing the bin width one bin at a time has a negative effect on the prediction accuracy. This can be explained by the rating getting closer to each other. Table 6 shows the accuracies per bin width are a lot lower when the tasks have no gap in rating. Here predictions start from the closest bins (5 and 6) and iteratively add one bin to the top and bottom half.

Bins considered	5 or 6	4-5 or 6-7	3-5 or 6-8	2-5 or 6-9	1-5 or 6-10
Accuracy	51.65%	54.68%	57.28%	60.36%	64.26%

Table 6 - Accuracy of RF model per bin width without range gap

When the model is run on only the fifth and the sixth bin, the accuracy is 51.65%. In this case the best player from bin 5 (rating of 1593) is only 1 rating point removed from the worst rated player in bin 6 (rating of 1594). Therefore, the answer to research question 3 is that increasing the bin width will lower the prediction accuracy. When looking at different bins, the decrease in accuracy can be explained by the decrease in the difference of the upper bound from the lower half and the lower bound from the upper half. The closer these numbers get to each other, the lower the accuracy becomes. This can be explained by the feature values being very similar for the middle bins. The average values of the most important features for the middle bins are very close to each other. This could be an indication that those groups have an almost indistinguishable playstyle when looking at the features extracted for this thesis.

6. Discussion & Conclusion

Chess players receive a rating that represents their relative strength related to a player pool. This rating is based on the results of all their previously played games. Currently, players joining a new pool will have to play many games to get a rating that is representative of their relative strength within the player pool. Since match making is based on the chess rating, new players will most likely play games against an opponent of disproportional strength. Playing a match with a player of unequal skill directly impacts the players' enjoyment of the game (Zhang, et al., 2022). There currently is no research in predicting chess rating, but the methods used in this thesis are based on methods used in predicting rating in other games. The research goal of this thesis is to solve this problem and fill the literature gap by using machine learning to predict chess rating based on a single game.

A total of 30 features have been extracted from the algebraic notation, based on chess theory and literature. These features are based on chess principles that describe good moves and positions, combined with moves and positions that should be avoided. These are features that can be used to predict chess rating using a machine learning model. Higher rated players are expected to (generally) follow the chess principles and make fewer mistakes or blunders. Results of important feature analysis confirm this assumption by looking at the average values for the important features for each bin. When the methods of this thesis are reproduced on a different data set, the results are expected to be similar when the feature values share the same spread between higher and lower rated players.

6.1 Machine learning model

Three research questions have been formulated to reach the goal of predicting chess rating based on a single game. The first research question aims to find the machine learning model that can most accurately predict chess rating. The machine learning models chosen are RF and SVM, because the literature review showed that these models have performed adequately in similar prediction tasks. The research question was answered by making predictions on 5 different subsets, each representing different bin widths. The prediction accuracy between the two models proved to be almost equal, with neither model performing better on all 5 tasks. However, the RF model took only 46 seconds to make the predictions, while the SVM model needed 37 minutes to perform the same task. Therefore, based on its faster computation time, the RF model was chosen for future tasks.

The boxplot in Figure 2 shows that there can be a large difference in prediction accuracy between each cross-fold validation. The robustness of this thesis is ensured by averaging the results after a 5-fold cross validation. Although, the model comparison could be more precise when the comparison was being done on the entire dataset, the RF model proved to perform well for this thesis. Due to time constraints, it was impossible to test more models and/or settings.

6.2 Important feature analysis

Permutation feature importance was used to find the features that are most indicative of chess rating. This was done by finding the feature importance on the 10-class classification model, as well as for the 5 binary classification models. While there is a small difference in the order of the top six features, the same six feature are at the top for every model. These features are the first blunder, first mistake, moves before castling, blunders, mobility (100), and game length. What stands out here is that three of the top six features are calculated by the chess engine Stockfish. Moves before castling, mobility and game length can be easily computed and identified by a player. However, the features that are generated based on Stockfish evaluations take a lot of computation time and are difficult to extract. For this research the depth limit for Stockfish was set at 8. This means that Stockfish calculates 8 moves ahead after every move to evaluate the position. The features for evaluation were split into 4 subsets, where each subset took a total of 13 hours to compute the Stockfish evaluations. For a more precise evaluation the limit can be set up to 20 moves, but for this thesis there was not enough time to compute the features with the depth set at more moves.

The important features of this thesis are based on the permutation feature importance method, because this method ensures that there is no bias in the analysis of important features. The permutation feature importance is run 10 times for each of the 6 classification tasks, where the results are based on the average of all runs per task. The first task is the feature importance for the 10-class classification, followed by the binary classification tasks with 5 different bin widths. This is done because permutation feature importance is based on randomly shuffling the values of a single column. This method assumes that randomly shuffling a column decreases the prediction accuracy of a model. But when only a single random shuffle has been performed, the feature can have an unexpected and inaccurate influence. The outcome of the feature importance analysis showed which features are most indicative of chess rating but does not contain any information about the values that are representative for each bin. Table 4 in chapter 6.2 shows an overview containing the average value for each bin. This table gives an indication of the difference in the feature values between the rating groups, but a limitation to this method is that it is unknown where the cut-off value is made in the prediction of the RF model. Predictions in the RF model are based on a group of decision trees, where the predicted class of the RF model is based on the most common outcome among all the decision trees. The cut-off values of a decision tree can be visualized, but it is not known what the effect of a single tree is on the final prediction.

6.3 Effect bin width

The last research question looks at the effect of the rating bin width on the prediction accuracy, when trained and tested on the entire dataset. As expected, the model comparing the weakest and strongest bins performed the best with an accuracy of 79.29%. The model comparing the top and bottom half could predict the correct bin with an accuracy of 64.26%. The decrease in prediction accuracy can be explained by the ratings coming closer together. The difference between the highest rated player in the weakest bin and the lowest rated player in the strongest bin is at least 532. This means an error in prediction is off by at least 532 rating points. When comparing the weakest and strongest half of the dataset, the prediction could be wrong by only 1 rating point and still be counted as an inaccuracy. The inaccurate predictions made in the first task, comparing the weakest and strongest bin, can be considered more erroneous than predictions made for the top vs. bottom 50%. Based on the average feature values for the middle bins, an error between these bins can be explained by players having a similar playstyle. This can be seen in the average value of the most important features. However, errors between the top and bottom bin should be concerning, given the differences in rating.

There can be different explanations as to why the accuracy in predicting the worst and best 10 percent doesn’t go above 80%. One explanation is that chess is a very complex game, where there are many exceptions to what is generally considered a good move. Some players decide to play a ‘trap’, which is where they decide to sacrifice a piece or good position and hope that the opponent gives away checkmate by taking this piece or position. This move will be considered a blunder by Stockfish, while the move is played on purpose with the knowledge that it can result in winning the game.

6.4 Considering more games

The results mentioned above are all computed based on a single game. This section aims to explore whether evaluating on more games will increase the model's accuracy. Three separate predictions were run using the RF model, to check whether the extracted features of this thesis can be used to make more accurate predictions. For these predictions the features and the rating were averaged based on 5, 10 and 20 games. With an average of 5 games, the accuracy for the lowest and highest rated bin went up to 91.14%. The accuracy based on the same bins but averaged over 20 games even went up to 96.48%. The accuracy for the 10-class classification problem went up to 25.13% based on the average of 20 games.

6.5 Summary

In conclusion, machine learning methods can be used to predict chess rating based on a single game. The predictions are based on features that are extracted from the algebraic notation. A total of 30 features were used in this research to predict the rating. When ordering the players based on rating and dividing them into 10 bins, the correct bin can be predicted with an accuracy just below 17%. This outperforms the baseline of 10%. When changing the width of the bins and the classification problem was made into a binary classification problem, the RF model was able to predict whether a player belongs to the strongest or weakest half of the players with an accuracy of 64.26%. Players that are further away from the middle rating have a higher accuracy than players that are close to it. Predictions based on the weakest and strongest bin are made with an accuracy of 79.29%.

The goal of the thesis was to predict chess rating based on a single game. This solves the problem of players having to play many games, when joining a new pool, to get a rating that matches their relative strength. The results of this thesis are promising. The extracted features show that they can be used as an indication of rating, without taking the result of the game into consideration. This can help chess organizations with their matchmaking for new players, without relying solely on the outcome of their matches. However, an accuracy of 79.29% is not yet sufficient for direct implementation and should be improved in future research to accurately predict rating based on a single game.

Due to time constraints for this thesis the impact of time control and the colour of the pieces could not me analysed. The assumption can be made that the range of feature values can be explained for each rating bin due to the time constraints of the game. Time constraints of games played on Lichess range from 1 minute to 30 minutes total, per player, to make decisions. For future work it can be interesting to see what the influence of time constraints and other subsets of the data have on the prediction accuracy and the important features. Furthermore, it can be interesting to look at more and/or different features, as there are many more chess principles and theories that have not been used for this thesis.

Bibliography

Allerbest, E. (2021, August 23). *How Chess.com scaled a massive community*. Retrieved from mixergy: mixergy.com/interviews/chess-com-with-erik-allebest/

Altmann, André, Tolosi, L., Sander, O., & Langauer, T. (2010, May 15). *Permutation importance: a corrected feature importance measure*. Retrieved from academic.oup: https://academic.oup.com/bioinformatics/article/26/10/1340/193348

Anshori, M., Mar'i, F., Alauddin, M., & Bachtiar, F. (2019, April 18). *Prediction Result of Dota 2 Games Using Improved SVM Classifier Based on Particle Swarm Optimization*. Retrieved from ieeexplore: https://ieeexplore.ieee.org/abstract/document/8693204/authors#authors

Aung, M., Bonometti, V., Drachen, A., Cowling, P., Kokkinakis, A., Yoder, C., & Wade, A. (2018, October 14). *Predicting Skill Learning in a Large, Longitudinal MOBA Dataset*. Retrieved from ieeexplore: https://ieeexplore.ieee.org/abstract/document/8490431?casa_token=juCmEgEkHb8AAAAA:LwvrNdsJM1ICkCzSl_EDahH3XJQJ-77tvPjfIe8wsj3CZ3SqwWA_5WdUMVCWW2yY4967ZoeOI7K1tA

Bialecki, A., Gajewski, J., Bialecki, P., & Phatak, A. (2022, September 17). *Determinants of victory in Esports - StarCraft II.* Retrieved from springer: https://link.springer.com/article/10.1007/s11042-022-13373-2#Sec16

Campbell, M., Hoane Jr, J., & Hsu, F.-h. (2022, Januari). *Deep Blue*. Retrieved from sciencedirect: https://www.sciencedirect.com/science/article/pii/S0004370201001291

Castling. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Castling#:~:text=Castling%20is%20a%20move%20in,that%20the%20king%20passed%20over.

Dick, S. (2019, July 2). *Artificial Intelligence*. Retrieved from duqduq: https://hdsr.duqduq.org/pub/0aytgrau/release/2

Elo, A. (1986). The Rating of Chessplayers Past & Present. In A. Elo, *The Rating of Chessplayers Past & Present* (pp. 58-64). New York: American Chess Foundation.

Engqvist, T. (2016). *Chess Strategy for Kids.* London: Gambit Publications Ltd.

Glickman, M., & Jones, A. (1999). *Rating the Chess Rating System*. Retrieved from Glicko: http://glicko.net/research/chance.pdf

Lewis, L. (2021, August 27). *Chess World Cup Winner Name, Runner-UUp, And Prize Money*. Retrieved from sportsunfold: sportunfold.com/chess-world-cup-winner/

Maharaj, S., Polson, N., & Turk, A. (2021, September 23). *Chess AI: Competing Paradigms for Machine Intelligence.* Retrieved from arxiv: https://arxiv.org/pdf/2109.11602.pdf#:~:text=Stockfish%20uses%20the%20alpha%2Dbeta,player%20will%20redirect%20the%20game.

McCarthy, J., Minksy, M., Rochester, N., & Shannon, C. (1955, August 31). *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence*. Retrieved from jmc.stanford.edu: http://jmc.stanford.edu/articles/dartmouth/dartmouth.pdf

Steinerberger, S. (2014, November 6). *On the number of positions in chess without promotion.* Retrieved from springer: https://link.springer.com/article/10.1007/s00182-014-0453-7

Structuur. (2022). Retrieved from schaakbond: https://www.schaakbond.nl/knsb/organisatie/structuur#:~:text=Er%20zijn%20ongeveer%20435%20verenigingen,Verder%20zijn%20mensen%20individueel%20lid

Yin, Q., Yang, J., Huang, K., Zhao, M., Ni, W., Liang, B., . . . Wang, L. (2022, August 18). *AI in Human-computer Gaming: Techniques, Challenges and Opportunities*. Retrieved from arxiv: https://arxiv.org/pdf/2111.07631.pdf

Zhang, C., Wang, K., Chen, H., Fan, G., Li, Y., Wu, L., & Zheng, B. (2022, August 15). *QuickSkill: Novice Skill Estimation in Online Multiplayer Games.* Retrieved from arxiv: https://arxiv.org/pdf/2208.07704.pdf