Machine Learning in Chess Move Predictions

Josh Cox

Abstract

In this paper, we explore the effects of technology on chess, with a more specific focus on machine learning. The idea being that a program will then learn and adapt like a human to achieve a result. We investigate different machine learning models, considering their accuracy and interpretability. Furthermore, we discuss different methods that have been used to solve multiple problems in chess, including chess engines, pattern recognition, and chess predictions. We then propose a project using machine learning to advance the understanding of how different chess principles are valued, by players at different rating levels. This will further the learning and understanding of the game of chess for beginners. The approach we discuss involves comparing decision trees, gradient boosting, and the explainable boosting machine (EBM) for predicting the next move in a given chess position and evaluating their accuracy and interpretability. We will then draw conclusions from these results to show whether this study can be used to help beginner players improve at chess.

I certify that all material in this dissertation which is not my own work has been identified.

Signature: Josh Cox

1 Introduction

The game of chess has been played for around fifteen hundred years [1] and is still gaining popularity. In recent years, the volume of new players has increased drastically. Since January 2020, over 102 million users have started playing chess on the website Chess.com alone [2]. Technology being recently introduced has revolutionised the game. The ability to watch people play chess online during the pandemic, along with being able to play the game online, made chess far more accessible, thus introducing new players [3]. Chess databases and engines are some other main advancements that technology has brought to chess. By using a chess database [4], you can view how many times an opening (first moves of a game) has been played, as well as the win chance of each colour. This is a great tool for learning chess and expanding your knowledge. Chess engines are computer programs that use the incredible amount of computing power available to determine the best move in a position. Early versions of these engines were not sophisticated enough to beat top-level humans until 1997 when the engine Deep Blue [5] beat world chess champion Gary Kasparov. Since then, the strength of engines has increased exponentially, with the best reaching a rating of around 3500 Elo [6], easily besting the highest level of human players (around 2800 Elo). An Elo rating in chess determines the level of play of a human or engine, named after the physicist who developed the system, Arpad Elo [7].

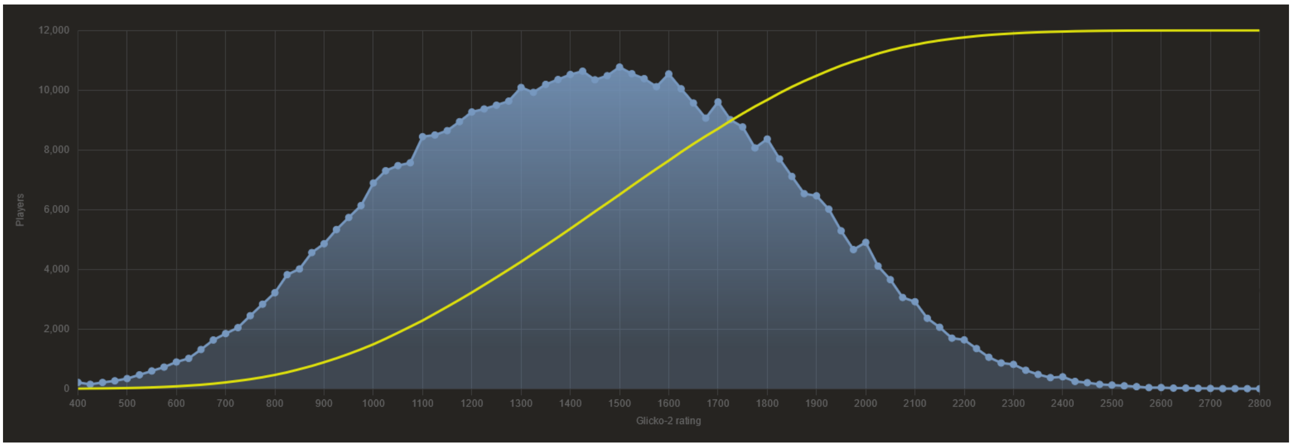


Figure 1. Lichess player ratings for the week of 13 Nov 2023 (10 minute games). Adapted from [8]

The use of these engines in chess game analysis is a huge leap forward for both high-end levels of play and beginners. On some websites, such as Chess.com [9], you can play against engines at varying Elo ratings. Engines that play at lower ratings, while technically having the same skill as human players at this level, adopt a completely different play style. Consequently, if a beginner were to try and learn using these engines, they might not learn the correct principles, as very random non-human-like moves are played. An engine that plays like a human by learning from them has been considered to overcome this [10]. We will investigate the results of similar studies in section 2.5.

Being able to play against these human-like engines is one of the ways that the advancement of technology can help further the teaching of chess to beginners. In this paper, we will consider how technology can be used in different ways to increase the efficiency of learning the game of chess, and its many principles. The approach we will take is to train multiple machine learning models from a database of human chess games, at a variety of different Elo ratings, to predict the next move of a chess game. We will then critically evaluate each model's interpretability and accuracy to decide which model is best suited for predicting in chess. Finally, we will determine whether these results can be used to further the study of chess for beginner players.

1.1 Project Objectives

This project aims to evaluate the use of interpretable machine learning models to significantly advance the ability to learn chess. We will investigate whether a decision tree, while easily understood, is complex enough to accurately predict in chess. A gradient boosting machine, thought to be more accurate, will be trained to determine whether it can be interpreted to provide useful insights into the importance of different factors in chess. Finally, the EBM is said to provide black box like accuracy while maintaining the interpretability of a linear model [11][12]. If a model is a black box, it means the process that was used to reach an outcome cannot be explained [13]. A model that is too complex to be expressed by a linear function is a non-linear model, whereas a linear model, such as logistic regression [14], can be. We will evaluate whether the EBM can be used for predictions in chess. By training these models, and observing the results, the importance of different chess principles in certain positions can be investigated. We will answer questions such as: What is the most common kind of mistake that is made at each rating level? Are some factors more important regardless of the players' ratings? Is there a model that is both accurate and interpretable for predictions in chess? Can the interpretability of a model give valuable insights into the patterns in play styles of differently rated players? The answers to these questions could prove useful in deciding what a new player should prioritise learning when improving at the game.

2 Preliminary Research

2.1 Advantages of technology in chess

The introduction of technology to chess has made a huge impact on how the game is played and studied. Beginners can learn more quickly as they are able to practice more with the introduction of online games. Computer engines and online chess websites, such as Chess.com [9], are usually the main things that are talked about within the topic of computers and chess. However, the introduction of technology has brought many other advantages, like enabling visually impaired people to play the game. Computer vision has been used to identify chess pieces on different board themes [15], allowing them to be able to play online chess games. This particular study was found to be only 60% accurate, with some arguing that Scale-invariant feature transform (SIFT), the technology they used, was inappropriate as the texture between different chess pieces was too similar [16]. There have been several other techniques used in different studies, including robots playing over the board (not online) [17][18].

2.2 How Chess Engines Work

Chess engines are computer programs that determine the best move in a board position [19]. Their play style is extremely unique, as the methods they use to find the best move are different from any human. The most common algorithm used in chess engines is the Minimax algorithm. It is named this as it aims to minimise the opponent's maximum advantage, and it is used mainly in zero-sum two-player games. This is a game where each player can independently choose an action that will equally affect each player in a negative and positive way [20]. In the case of chess, if a player makes a move that gives them an advantage, they are giving their opponent an equal disadvantage. Other examples of zero-sum two-player games are Tic-Tac-Toe and Poker. The Minimax algorithm traverses a tree of all possible game moves, up to a certain depth, then will recursively evaluate each node on each level of the tree (each level alternating between each player's turn), returning either the largest or smallest child node [21]. The problem with this is that as the tree gets deeper, the number of nodes it has to evaluate increases exponentially [22]. Alpha-Beta Pruning is an optimising technique that is applied to the Minimax algorithm to reduce the number of nodes that it needs to evaluate. This works by pruning off any node, and its respective subtree, that leads to a worse position than one already evaluated by the algorithm. By getting rid of these large chunks of nodes, we increase the efficiency of the algorithm, and therefore the speed. Conversely, the approach a human takes to deciding the best move to play in a position, is mainly by pattern recognition, not deep analysis of every move. The main difference between high and low-level players is being able to recognise similar patterns and which strategies worked in them, not being able to calculate a lot of moves ahead. This was determined in a paper [23], which used a neural network to evaluate positions similar to how top-level players do. It would rely on pattern recognition rather than a deep calculation of moves. This differs from most engines, as it introduces the aspect of machine learning, rather than relying on brute force to evaluate every move. Hence, we will be exploring what a neural network is in section 2.4.2.

2.3 Human Behaviour in Chess

Humans and computers play chess in contrasting ways, as they process information and make decisions very differently [24][25]. This is perhaps why creating programs that think the same way humans do can be challenging. Neural networks, explained in section 2.4.2, take inspiration from humans, as they are based on the human brain, using a system of neurons. Although engines play in a different way to humans, they show some similarities in identifying chess concepts. In 2022, one of the top engines, AlphaZero, was studied to see what knowledge it had learned about chess by only playing against itself [26]. They confirmed that human concepts were found in the neural network of the engine. Despite the few apparent similarities, an engine's overall play style is vastly different from a human, leading to projects such as Maia [27]. Maia is an engine that was created with the aim to be indistinguishable from a human, as the authors claim that humans prefer playing humans to computers. They were successful with a good level of accuracy, however, a more in-depth study extended this, with the intention of adapting the engine to a specific player's style [28]. Their models managed to capture "near-perfect stylometry", meaning that they could identify one player out of 400, based purely on their chess games. This could advance how we learn chess, as we could play against an almost identical version of ourselves, and see what mistakes we make. An observation by Francis Mechner, about cognitive skill research, found that the difference between people who play chess and people who don't is not in general memory [29]. In fact, even for the cases of people who can play a full game of chess blindfolded, it was concluded that their level of general memory was "no better than that of the average person". This shows that chess is a skill that is learned through practice, not just memorisation.

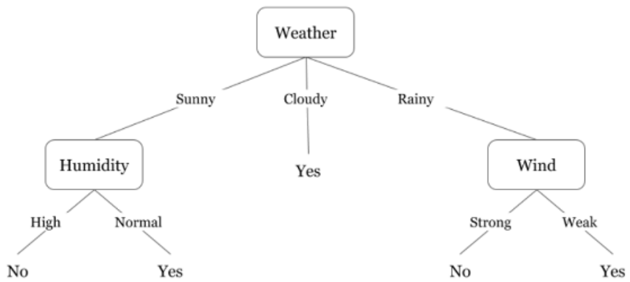
2.4 Machine Learning Models and Interpretability

In this section we will investigate different machine learning models, along with their accuracy and interpretability. Machine learning is when an algorithm is given the desired result without the specification of how to achieve it, along with a data set to learn from [30]. The aim being to try and simulate the way humans learn new skills, not by being told what to do, but by practicing [31]. There are lots of different models, the main ones for chess evaluation being decision trees, neural networks, and ensemble models. Unsupervised machine learning is the idea of clustering unlabelled data together based on their similarity of features. Supervised learning, however, is classifying given data into those labelled groups [32]. The models we will be using are non-linear and supervised. This will allow us to predict the next move in a game, whilst considering its complexity. Interpretability is how easily a human can understand the factors that lead to a result. We are aiming to understand what caused a chess move prediction to be made.

2.4.1 Decision Tree

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

(a) Dataset



(b) Decision tree

Figure 2. Example of decision tree creation for playing golf. Adapted from [33]

A decision tree is a learning algorithm used to classify data items [34]. It is a tree formed of a root node followed by various sub-trees, with the purpose of forming clusters of items with similar features. Each node is a question and has a child node for each possible answer. This forms a hierarchical model that can be grown by adding question nodes [35]. A decision tree has exceptional interpretability, allowing us to see the weighting of each feature used. The tree display is also very easily understood by humans, giving it an advantage over other machine learning models. The above figure shows a decision tree created from a table of features, to decide whether to play golf on a certain day. The factors that determine the outcome are weather, humidity, and wind. Using a decision tree for predictions in chess games could be challenging, as chess can be very complex. While a decision tree is very good for this application in terms of interpretability, the accuracy of the predictions may suffer. We will determine how feasible using a decision tree for chess predictions is in this project.

2.4.2 Neural Network

Another form of machine learning model is a neural network. This is a black box model, unlike the decision tree, and its interpretability is poor. Although this could be a disadvantage, a neural network is capable of extremely accurate results in complex situations. To create a neural network you layer perceptrons, feeding the outputs of some into others (see Figure 4). Figure 3 shows an example of a perceptron that takes three inputs, each with different weightings (importance), and gives one output. Essentially, it is a function that takes the sum of the weights of the inputs and returns an output [37]. There are many examples of these that are used in different neural networks [38]. The function used in a perceptron is called an activation function. Each perceptron (also known as a neuron) in a network can have its own activation function. There are many different network architectures today, with each having an advantage in different applications.

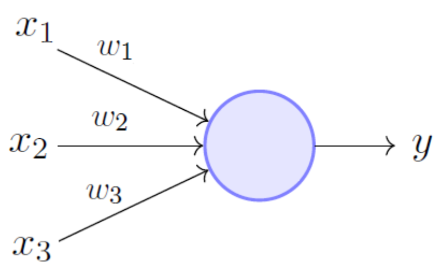


Figure 3. Example of a perceptron. Adapted from [36]

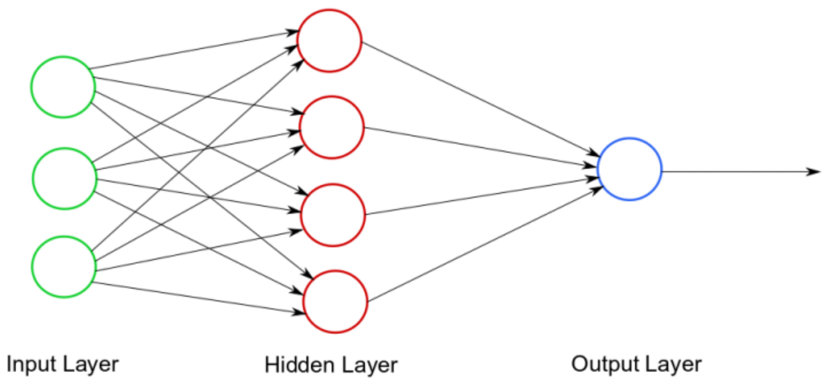


Figure 4. Example of a neural network. Adapted from [39]

In relation to chess, using a neural network has been explored, with some accurate results, to create both computer chess engines and predictors. We will investigate these in section 2.5.

2.4.3 Gradient Boosting and the Explainable Boosting Machine

An ensemble machine learning model is a collective of models [40]. This can overcome the fact that single models, such as a decision tree, can succumb to overfitting. Meaning a model performs well on data that it has seen before but struggles to produce accurate results for unseen input data [41].

Gradient boosting is an ensemble model, that utilises boosting. Boosting is creating iterations (versions) of the model, each time improving the errors [42]. This is done sequentially, and in gradient boosting specifically, the algorithm used to minimise these errors is called the gradient descent algorithm. We will be using gradient boosting, as it has been shown to predict the outcome of a chess game while maintaining a level of interpretability [43], but has not yet been used to predict the next move of a game. The project shows that the model's accuracy was 0.55, giving a correct prediction just over half of the time. This can be seen as a good result as a chess game can end in three ways (win, lose, draw), meaning that a random guess would average an accuracy of 0.33. However, as discussed in the article, most of the games were either a win or a lose, which could have led to some bias. This could have misrepresented the accuracy of the model, meaning it would be only slightly better than a random guess. One reason for low accuracy could be due to the features it was given. It could only make a prediction based on the players' ratings and win/lose game streaks. In our project, we will focus on the principles of chess, mentioned in section 3.1.2, as well as the current board position. This would give a more complex representation of the game, leading to more accurate results.

Ensemble models along with neural networks have the advantage of being able to maintain accuracy in complex applications, but the disadvantage of not being highly interpretable. Whereas other models like the decision tree remain understandable to humans but lack the complexity for certain tasks. One solution to be able to make predictions in chess while maintaining a good level of interpretability could be the EBM. This is a type of generalised additive model (GAM) [44]. This is a model that manages to keep the interpretability of a linear model, while also being able to take into account non-linear data. This gives a higher accuracy than other linear models, while still remaining easy to understand [45]. EBMs usually take longer than other models to train [12] but often result in similar accuracy to a black box model, such as the gradient boosting machine mentioned before.

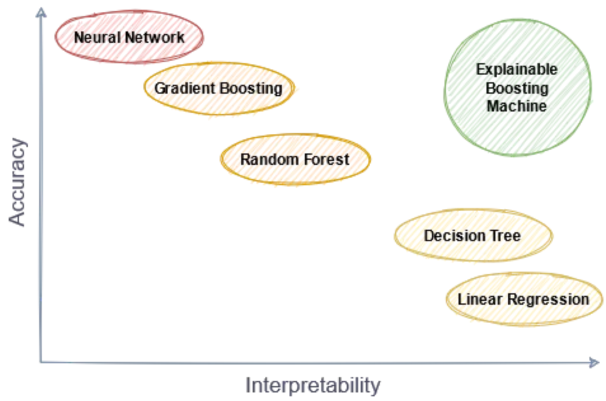


Figure 5. Different models' accuracy against interpretability. Adapted from [11]

2.5 Solving Chess Problems with Machine Learning

The use of machine learning has been considered for solving the chess endgame problem. This is the task of finding the least amount of moves for a player to win a game, in a position where there are not many pieces left on the board (endgame). Hongyue Li carried out an investigation, from which he concluded that both ensemble methods (adaboost and random forest) performed better than a decision tree, however they were found to be less understandable [46]. This aligns with what we covered about each model's accuracy and interpretability in the previous section. A related paper investigating logistic regression and neural networks, along with decision trees concluded that the black box model resulted in the highest accuracy, with logistic regression (LR) being the least accurate [47]. This is likely due to the linear nature of LR, restricting its performance for more complex tasks such as chess games.

Another area where machine learning has been used is to analyse chess data, with an aim to uncover patterns in the game such as long-term strategies. A 2006 study found that the traditional Minimax algorithm with Alpha-Beta Pruning that most engines relied on, while strong for short-term tactics, had a weakness for long-term strategies [48]. They aimed to learn insights into the principles of playing endgames, similar to top-level human play. The approach they took was to use a decision tree classifier to split the endgame into different stages of play. They then applied a linear regression function to approximate the distance-to-win for each endgame stage, to use with the Minimax search algorithm. Above a depth of 10 ply (5 moves) the predictor would start to see successful results.

We have looked at how current engines carry out large amounts of computation to find the best move. They evaluate every move in the position to a certain depth, whereas humans at the higher level are able to look at a position and recognise what types of moves will work, and what to discard [49]. This means that they can discard a large number of moves straight away and focus the evaluation on a select few. This is much more computationally efficient, and studies have been trying to replicate this pattern recognition style of playing using machine learning. A paper written in 2018 shows this, where they trained a neural network to evaluate chess positions, not by relying on the usual deep look ahead, but by pattern recognition. The results of this study were that the best performance would be an engine that incorporates both pattern recognition and deep look ahead for more tactical positions. Neural networks have also been used to try and predict outcomes of ongoing games [50]. An accuracy of 68.9% was achieved by using both the moves played and the metadata of the game. Metadata is information about the data, such as the player names and Elo rankings.

The play style of chess engines is very different to humans. Machine learning has been used to try and solve this problem and create an engine that plays more like a human. Rosemarin and Rosenfeld successfully demonstrated the use of a neural network, as well as a common chess engine algorithm (Monte Carlo Search Tree), to create an engine that was indistinguishable from a human [51]. This would allow human players to play against an engine of any rating, to improve at the game. In the study, an expert would guess if a game was played by their engine or a human. Both were guessed to be engines half of the time, meaning that they could reasonably conclude that they had succeeded in their task. These results could be extended by investigating what moves caused the expert to guess that the game was played by a computer, and trying to eliminate these types of moves from the engine's play style. Regardless of this, studying and learning what the engine plays in certain situations would further a beginner's understanding of the different principles of chess.

3 Project Specification

This project will aim to investigate the importance of different factors in chess, at different Elo ratings. The ability to see what most beginners would play compared to higher-rated players will be more useful than looking at what an engine would play, as discussed before, an engine is very non-human-like. By factors, we mean the different aspects within the game of chess, such as material advantage (who has more/better-valued pieces on the board). This section will provide the detailed approach we will take to answer the questions mentioned in section 1.1, as well as methods to evaluate our results.

3.1 The Approach

We will train a decision tree, gradient boosting machine, and explainable boosting machine for chess predictions. We will do this on a large data set of chess games, played at different rating levels. The data set we will be using is the Lichess open database, which has around 5 billion games currently stored [52], as this is one of the only databases that allows you to access the moves of each game, along with the player ratings. It is also one of the most popular websites for playing chess online, providing a good range of games at different levels. Python will be the programming language we will be using for two main reasons. It is well known with an active community, giving it an advantage over other languages, due to the number of guides and documentation available. Python has a very large number of libraries that will be useful for this project. For example Scikit-Learn [53] for machine learning, Pandas [54] for data science, SHAP [55] for interpretability, and more niche libraries such as python-chess [56]. These reasons make it the clear choice for this project.

3.1.1 Chess Notations

There are two notations we need to cover when talking about representing chess. Portable Game Notation (PGN) [57] and Forsyth-Edwards Notation (FEN) [58]. These are standards for showing details about a game, and specific board positions respectively. PGN holds the moves of the game for both colours, as well as relevant metadata; this would include things like the date of the game, usernames and ratings of players, time control, and the game result. FEN shows where each piece is on the board, whose turn it is, specific moves that can be played in certain positions (castling and en passant), and tracks the number of moves played in the game so far (halfmove and fullmove clock) [58].

3.1.2 Data Representation

The Lichess open database allows you to download all games of each month from Jan 2013 to the present, as a PGN. As for the time control, we will only be using games that are standard ten-minute rapid games, with no time increments. This is to ensure that the project scale is feasible within the time frame. Ten-minute games are one of the standards to play for beginners as it gives you just the right amount of time to think. As well as this, it is one of the most popular time controls played online [59]. Future models could be improved by adding time control as a determining feature.

Deciding how to represent our data is a challenging task as we need to denote the board position so that the next move can be predicted. One of the methods to use would be a bitboard; this is an 8x8 array containing each square on the board, with a value to indicate the colour and type of piece on that square. Figure 6 demonstrates this, where the white pieces are numbers 1-6 (Pawn, Rook, Knight, Bishop, Queen, King), and the black pieces are the respective white piece value, plus 6. 0 is an empty square.

```
[8, 9, 10, 11, 12, 10, 9, 8,
7, 7, 7, 7, 7, 7, 7, 7,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 1, 1,
2, 3, 4, 5, 6, 4, 3, 2]
```

Figure 6. Chess position as a bitboard representation.

While this could work, our aim is to investigate what factors (features) had what influence (weighting) on the prediction. We want to know if a player's lack of central space control mainly led to the decision, as opposed to their king's safety. With just a bitboard this would be difficult to extract. A solution to this would be to include hand-crafted, high-level features. This means we could define the features manually, and how they are calculated before training the model. To create the feature of king safety, we would use the FEN notation to see how many pieces, of the same colour, are surrounding the king, providing it protection. If there are none then the king is unsafe, whereas if there is a wall of pawns in front of it, this could be considered safe. There are a large number of features to consider when evaluating a chess position. To make sure the project stays within scope while maintaining a good amount of useful information about the positions, we will be focusing on a select few: King safety (how many friendly pieces are around the king), Elo rating of each player, central space control (how many friendly pieces have influence over the central squares), material difference (difference in total piece value of each player). Accepted values are (Queen: 9, Rook: 5, Bishop/Knight: 3, Pawn: 1) [60].

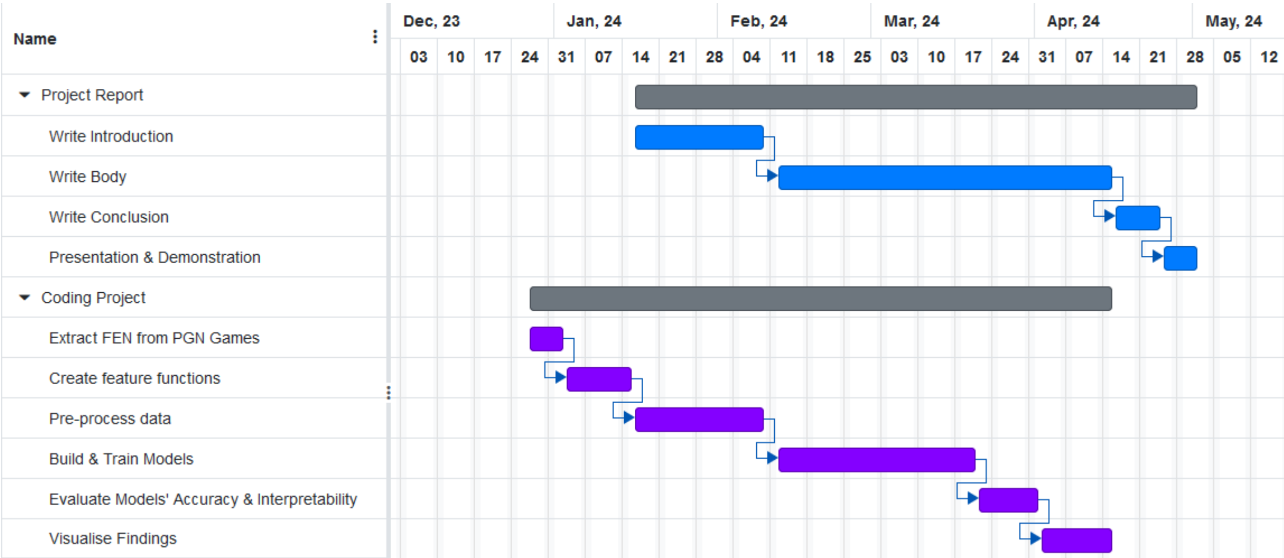
We will create functions to generate numbers to represent each feature's value using FEN. These will be combined with the bitboard to create an ideal representation. This will be in the form of a one-dimensional array (an array with only one row), with the first set of elements being the flattened bitboard, and the following the above-mentioned features. Capturing both static and dynamic position features will be possible with a combination of the two representations.

3.2 Evaluation Strategy

A successful outcome for this project would be to determine a suitable model that can be used to accurately predict the next move in a chess game while maintaining a good level of interpretability. We will then conclude whether our findings can be used to further the understanding of the importance of different chess principles. The accuracy of each model will be evaluated relative to the project. For example, an accuracy of 0.33 could be considered low when looking at the prediction of the game outcome, as it is the same as a random guess (win, lose or draw). However, in the case of predicting a specific move, there are usually lots to choose from, meaning that this accuracy could be acceptable. We will test each model with a variety of positions to come to a valid conclusion about its accuracy. To evaluate the interpretability of each model we will attempt to extract weighting for each of the above discussed input features. This will allow us to determine which features were of higher importance in different situations.

4 Project Management

4.1 Work Plan



4.2 Risks and Legal/Ethical Issues

One risk to consider for this project could be the amount of computation and time needed to train the models. We have chosen a data representation that is suitable for the project aim, but not too complex. One paper uses a 71-bit representation for each square in order to properly capture the dynamics of the positions [49]. While this could provide more accurate results, the amount of information to process would simply be too large for our project scope.

The only legal or ethical issues to consider for this project would be the sensitivity/access rights to the data we will be using. However, this is not an issue as it is publicly available and does not contain any sensitive information. The Lichess website states that you can "use them for anything you like" and you can "download, modify, redistribute them, without asking permission" [52]. Furthermore, we are not looking at a specific user's games, we will be looking at an abundance of games, from a range of different players and ratings. We will only be using information about the game itself, for example, the moves of the game, the time control, the outcome, and the player ratings.

5 Conclusion

In summary, this paper highlighted the profound effect of technology on chess, specifically the use of machine learning techniques to solve known problems. We showed that large steps have been taken towards solving these problems, but that overall, there is still more research to be carried out. Thus, we hypothesised an approach to advance the ability to learn chess as a beginner by investigating the relevance of different chess principles in different situations. We proposed to achieve this by critically evaluating both the accuracy and interpretability of decision trees, gradient boosting, and the EBM when predicting the next move in a chess game. The conclusions we come to could significantly impact the types of decisions beginners make in games, as well as the strategies they use to learn and improve. Any advancement in machine learning accurately imitating human decision making, will help beginners improve more efficiently to the desired standard, but more importantly help maintain that standard when faced with a human opponent.

References

[1] Y. Averbakh, *A History of Chess: From Chaturanga to the Present Day.* Milford, USA: Russel Enterprises Inc, 2012. [Online]. Available: https://books.google.co.uk/books?id=uJBXDwAAQBAJ&printsec=copyright&redir_esc=y#v=onepage&q&f=false

[2] B. Shimel, "Chess grows more popular, with chess.com experiencing server issues due to high user volume," *WUFT News*, Accessed: Oct. 30, 2023. [Online]. Available: https://www.wuft.org/news/2023/02/17/chess-grows-more-popular-with-chess-com-experiencing-server-issues-due-to-high-user-volume/

[3] K. Browning, "Chess (Yes, Chess) Is Now a Streaming Obsession," *The New York Times*, Accessed: Nov. 13, 2023. [Online]. Available: https://www.nytimes.com/2020/09/07/technology/chess-new-streaming-obsession-twitch.html

[4] Chess.com, "Chess Opening Explorer & Database," *Chess.com*, Accessed: Nov. 18, 2023. [Online]. Available: https://www.chess.com/explorer

[5] M. Campbell, A. J. Hoane, and F.-h. Hsu, "Deep Blue," *Artificial Intelligence*, vol. 134, no. 1, pp. 57–83, Jan. 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370201001291

[6] "CCRL 40/15 - Index," *Computerchess*, Accessed: Nov. 13, 2023. [Online]. Available: https://computerchess.org.uk/ccrl/4040/index.html

[7] W. L. Winston, S. Nestler, and K. Pelechrinis, "55. Elo Ratings," in *Mathletics: How Gamblers, Managers, and Fans Use Mathematics in Sports, Second Edition.* Princeton, USA: Princeton University Press, 2022, pp. 524–530. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/9780691189291-058/html

[8] Lichess, "Weekly Rapid rating distribution," *lichess,org*, Accessed: Nov. 18, 2023. [Online]. Available: https://lichess.org/stat/rating/distribution/rapid

[9] Chess.com, "Home," *Chess.com*, Accessed: Nov. 20, 2023. [Online]. Available: https://www.chess.com/home

[10] R. McIlroy-Young, S. Sen, J. Kleinberg, and A. Anderson, "Aligning Superhuman AI with Human Behavior: Chess as a Model System," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* Virtual Event CA USA: ACM, 2020, pp. 1677–1687. [Online]. Available: https://dl.acm.org/doi/10.1145/3394486.3403219

[11] D. R. Kübler, "The Explainable Boosting Machine," *Medium*, Accessed: Nov. 11, 2023. [Online]. Available: https://towardsdatascience.com/the-explainable-boosting-machine-f24152509ebb

[12] "Explainable Boosting Machine — InterpretML documentation." [Online]. Available: https://interpret.ml/docs/ebm.html

[13] N.-T. Vu and K.-U. Do, "Chapter 27 - Prediction of Ammonium Removal by Biochar Produced From Agricultural Wastes Using Artificial Neural Networks: Prospects and Bottlenecks," in *Soft Computing Techniques in Solid Waste and Wastewater Management*, R. R. Karri, G. Ravindran, and M. H. Dehghani, Eds. Elsevier, Jan. 2021, pp. 455–467. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128244630000124

[14] IBM, "What is Logistic regression?" *IBM*, Accessed: Nov. 19, 2023. [Online]. Available: https://www.ibm.com/topics/logistic-regression

[15] B. S. Kong, I. Hipiny, and H. Ujir, "Classification of Digital Chess Pieces and Board Position using SIFT," in *2021 IEEE International Conference on Signal and Image Processing Applications (ICSIPA).* Kuala Terengganu, Malaysia: IEEE, 2021, pp. 66–71. [Online]. Available: https://ieeexplore.ieee.org/document/9576797/

[16] G. Wölflein and O. Arandjelović, "Determining Chess Game State from an Image," *Journal of Imaging*, vol. 7, no. 6, p. 94, Jun. 2021, Accessed: Oct. 14, 2023. [Online]. Available: https://www.mdpi.com/2313-433X/7/6/94

[17] E. Sokic and M. Ahic-Djokic, "Simple Computer Vision System for Chess Playing Robot Manipulator as a Project-based Learning Example," in *2008 IEEE International Symposium on Signal Processing and Information Technology*. Sarajevo, Bosnia and Herzegovina: IEEE, 2008, pp. 75–79. [Online]. Available: http://ieeexplore.ieee.org/document/4775676/

[18] K. I.-K. Wang, "Robust Computer Vision Chess Analysis and Interaction with a Humanoid Robot," *Computers*, vol. 8, no. 1, pp. 11–14, 2019, Accessed : Nov. 20, 2023. [Online]. Available: https://www.proquest.com/docview/2548365803?pq-origsite=primo

[19] M. Méndez, M. Benito-Parejo, A. Ibias, and M. Núñez, "Metamorphic testing of chess engines," *Information and Software Technology*, vol. 162, p. 107263, Oct. 2023, Accessed : Nov. 15, 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0950584923001179

[20] T. E. S. Raghavan, "Chapter 20 Zero-sum two-person games," in *Handbook of Game Theory with Economic Applications*, R. Aumann and S. Hart, Eds. Elsevier, Jan. 1994, vol. 2, pp. 735–768. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574000505800529

[21] J. Madake, C. Deotale, G. Charde, and S. Bhatlawande, "CHESS AI: Machine learning and Minimax based Chess Engine," in *2023 International Conference for Advancement in Technology (ICONAT)*. Goa, India: IEEE, 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10080746/

[22] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*, 3rd ed. Upper Saddle River, USA: Pearson, 2016. [Online]. Available: https://web.cs.ucla.edu/~srinath/static/pdfs/AIMA.pdf

[23] M. Sabatelli, F. Bidoia, V. Codreanu, and M. Wiering, "Learning to Evaluate Chess Positions with Deep Neural Networks and Limited Lookahead," in *7th International Conference on Pattern Recognition Applications and Methods*, Jan. 2018. [Online]. Available: https://www.researchgate.net/publication/322539902_Learning_to_Evaluate_Chess_Positions_with_Deep_Neural_Networks_and_Limited_Lookahead

[24] C. T. Ross, "Unlocking the Secrets of the Brain | The Institution of Analysts and Programmers," *The Institution of Analysts and Programmers*, Accessed: Nov. 13, 2023. [Online]. Available: https://www.iap.org.uk/main/unlocking-the-secrets-of-the-brain/

[25] S. Maharaj, N. Polson, and A. Turk, "Chess AI: Competing Paradigms for Machine Intelligence," *Entropy*, vol. 24, no. 4, p. 550, Apr. 2022. [Online]. Available: https://www.mdpi.com/1099-4300/24/4/550

[26] T. McGrath, A. Kapishnikov, N. Tomašev, A. Pearce, M. Wattenberg, D. Hassabis, B. Kim, U. Paquet, and V. Kramnik, "Acquisition of chess knowledge in AlphaZero," *Proceedings of the National Academy of Sciences*, vol. 119, no. 47, p. e2206625119, Nov. 2022. [Online]. Available: https://www.pnas.org/doi/full/10.1073/pnas.2206625119

[27] MaiaChess, "Maia Chess," *MaiaChess.com*, Accessed: Oct. 29, 2023. [Online]. Available: https://maiachess.com

[28] R. McIlroy-Young, R. Wang, S. Sen, J. Kleinberg, and A. Anderson, "Learning Models of Individual Behavior in Chess," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aug. 2022, pp. 1253–1263. [Online]. Available: http://arxiv.org/abs/2008.10086

[29] F. Mechner, "Chess as a Behavioral Model for Cognitive Skill Research: Review of Blindfold Chess by Eliot Hearst and John Knott," *Journal of the Experimental*

Analysis of Behavior, vol. 94, no. 3, pp. 373–386, Nov. 2010. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2972788/

[30] IBM, "What is Machine Learning?" *IBM*, Accessed: Nov. 02, 2023. [Online]. Available: https://www.ibm.com/topics/machine-learning

[31] M. Kubat, *An Introduction to Machine Learning*. Cham: Springer International Publishing, 2021. [Online]. Available: https://link.springer.com/10.1007/978-3-030-81935-4

[32] J. A. Brown, A. Cuzzocrea, M. Kresta, K. D. Kristjanson, C. K. Leung, and T. W. Tebinka, "A Machine Learning Tool for Supporting Advanced Knowledge Discovery from Chess Game Data," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Cancun, Mexico: IEEE, Dec. 2017, pp. 649–654. [Online]. Available: http://ieeexplore.ieee.org/document/8260705/

[33] A. Saini, "Decision Tree Algorithm - A Complete Guide," *Analytics Vidhya*, Accessed: Nov. 03, 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

[34] B. de Ville, "Decision trees," *WIREs Computational Statistics*, vol. 5, no. 6, pp. 448–455, Accessed: Nov. 03, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/wics.1278

[35] C. Kingsford and S. L. Salzberg, "What are decision trees?" *Nature Biotechnology*, vol. 26, no. 9, pp. 1011–1013, Sep. 2008. [Online]. Available: https://www.nature.com/articles/nbt0908-1011

[36] M. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry*, L. Bottou, Ed. The MIT Press, Sep. 2017, Accessed: Nov. 20, 2023. [Online]. Available: https://direct.mit.edu/books/book/3132/PerceptronsAn-Introduction-to-Computational

[37] M. T. Jones, "A neural networks deep dive," *IBM Developer*, Accessed: Nov. 09, 2023. [Online]. Available: https://developer.ibm.com/articles/cc-cognitive-neural-networks-deep-dive/

[38] S. Sharma, S. Sharma, and A. Athaiya, "ACTIVATION FUNCTIONS IN NEURAL NETWORKS," *International Journal of Engineering Applied Sciences and Technology*, vol. 04, no. 12, pp. 310–316, May 2020. [Online]. Available: https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf

[39] V. ANAND, "Introduction to Neural Network in Machine Learning," *Analytics Vidhya*, Jan. 2022, Accessed: Nov. 07, 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2022/01/introduction-to-neural-networks/

[40] IBM, "What is Boosting?" *IBM*, Accessed: Nov. 11, 2023. [Online]. Available: https://www.ibm.com/topics/boosting

[41] ——, "What is Overfitting?" *IBM*, Accessed: Nov. 18, 2023. [Online]. Available: https://www.ibm.com/topics/overfitting

[42] C. Wade and K. Glynn, *Hands-On Gradient Boosting with XGBoost and scikit-learn: Perform accessible machine learning and extreme gradient boosting with Python*. Birmingham, UK: Packt Publishing Ltd, 2020. [Online]. Available: https://www.google.co.uk/books/edition/Hands_On_Gradient_Boosting_with_XGBoost/2tcDEAAAQBAJ?hl=en&gbpv=0

[43] A. R. PhD, "Predicting chess games results using LightGBM," *Medium*, Accessed: Nov. 18, 2023. [Online]. Available: https://medium.com/@alunariz/predicting-chess-games-results-using-lightgbm-818f30b5a7c3

[44] T. Hastie and R. Tibshirani, "Generalized Additive Models," in *Statistical Science*, vol. 1, no. 3. Institute of Mathematical Statistics, Aug. 1986, pp. 297–310. [Online]. Available: https://projecteuclid.org/journals/statistical-science/volume-1/issue-3/Generalized-Additive-Models/10.1214/ss/1177013604.full

[45] P. Pandey, "Explainable Boosting machines for Tabular data," *Kaggle*, Accessed: Nov. 11, 2023. [Online]. Available: https://kaggle.com/code/parulpandey/explainable-boosting-machines-for-tabular-data

[46] H. Li, "The Application of Machine Learning in Chess Endgames Prediction," in *Proceedings of the 2020 2nd International Conference on Big Data and Artificial Intelligence*. Johannesburg South Africa: ACM, Apr. 2020, pp. 9–14. [Online]. Available: https://dl.acm.org/doi/10.1145/3436286.3436289

[47] M. S. Fayed, "Classification of the Chess Endgame problem using Logistic Regression, Decision Trees, and Neural Networks," Nov. 2021, Accessed: Oct. 14, 2023. [Online]. Available: http://arxiv.org/abs/2111.05976

[48] A. Sadikov and I. Bratko, "Learning long-term chess strategies from databases," *Machine Learning*, vol. 63, no. 3, pp. 329–340, Jun. 2006. [Online]. Available: https://doi.org/10.1007/s10994-006-6747-7

[49] M. Kubat and J. Žižka, "Learning Middle-Game Patterns in Chess: A Case Study," in *Intelligent Problem Solving. Methodologies and Approaches*, R. Logananthara, G. Palm, and M. Ali, Eds. Berlin, Heidelberg: Springer, 2000, pp. 426–433. [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-45049-1_52

[50] R. Dreżewski and G. Wątor, "Chess as Sequential Data in a Chess Match Outcome Prediction Using Deep Learning with Various Chessboard Representations," *Procedia Computer Science*, vol. 192, pp. 1760–1769, 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050921016756

[51] H. Rosemarin and A. Rosenfeld, "Playing Chess at a Human Desired Level and Style," in *Proceedings of the 7th International Conference on Human-Agent Interaction*. Kyoto Japan: ACM, Sep. 2019, pp. 76–80. [Online]. Available: https://dl.acm.org/doi/10.1145/3349537.3351904

[52] Lichess, "lichess.org open database," *lichess.org*, Accessed: Nov. 12, 2023. [Online]. Available: https://database.lichess.org/

[53] Scikit-Learn, "scikit-learn: machine learning in Python," *scikit-learn.org*, Accessed: Nov. 12, 2023. [Online]. Available: https://scikit-learn.org/stable/

[54] Pandas, "pandas - Python Data Analysis Library," *pandas*, Accessed: Nov. 12, 2023. [Online]. Available: https://pandas.pydata.org/

[55] C. Tann and J. Goh, "Welcome to the SHAP documentation," *SHAP*, Accessed: Nov. 12, 2023. [Online]. Available: https://shap.readthedocs.io/en/latest/

[56] N. Fiekas, "python-chess: a chess library for Python," *python-chess*, Accessed: Nov. 12, 2023. [Online]. Available: https://python-chess.readthedocs.io/en/latest/

[57] Chess.com, "Chess PGN (Portable Game Notation)," *Chess.com*, Accessed: Nov. 14, 2023. [Online]. Available: https://www.chess.com/terms/chess-pgn

[58] ——, "FEN (Forsyth-Edwards Notation) - Chess Terms," *Chess.com*, Accessed: Nov. 12, 2023. [Online]. Available: https://www.chess.com/terms/fen-chess

[59] ——, "[UPDATED] 10 Minute Chess Now Rapid Rated, Bullet Ratings Increased," *Chess.com*, Accessed: Nov. 20, 2023. [Online]. Available: https://www.chess.com/news/view/10-minute-chess-now-rapid-rated-bullet-ratings-increased

[60] A. Gupta, C. Grattoni, and A. Gupta, "Determining Chess Piece Values Using Machine Learning," *Journal of Student Research*, vol. 12, no. 1, Feb. 2023. [Online]. Available: https://www.jsr.org/hs/index.php/path/article/view/4356