Win-Loss Prediction Based on an Opening A Player Chooses for a Game of Chess

Akshay Jain, Nidhi Tholar, Sashank Pidur Kuppuswamy, San José State University

October 2021

Abstract

Using data mining techniques, the project seeks to forecast the winning percentage in a chess game. The main purpose of the research is to predict the outcome of a game using the massive quantity of data accumulated over the years whenever a player chooses a particular opening. Furthermore, we have methods which we will suggest next moves to be played.

Introduction

A Game of Chess is played between two players on an 8x8 Square board,16 pieces on each side, and with millions of board positions and possibilities. It was invented fifteen centuries back in India and has been spread all over the world it lots of changes in the game's rules over time until they were finalized around the 1880s which is also the romantic era of chess. There are millions of games played and recorded in various places like books, newspapers, and online databases over time and the players use them to improve and learn the game of chess. In this project, we are trying to create an algorithm that helps chess players to learn about the different types of chess openings and statistics associated with it like winning percentage over the years, use of it at grandmaster (expert at the game) level, etc. A game of chess is stored in a form of a PNG (portable game notation) file. To store and process a chess game we first need to understand the notations used in a PNG file, a sample PNG looks like –

[Event "New York"]
[Site "New York"]
[Date "1918.??.??"]
[Round "?"]
[White "Capablanca, Jose Raul"]
[Black "Marshall, Frank James"]
[Result "1-0"]
[WhiteElo "2589"]
[BlackElo "2632"]
[ECO "C89"]

1.e4 e5 2.Nf3 Nc6 3.Bb5 a6 4.Ba4 Nf6 5.O-O Be7 6.Re1 b5 7.Bb3 O-O 8.c3 d5 9.exd5 Nxd5 10.Nxe5 Nxe5 11.Rxe5 Nf6 12.Re1 Bd6 13.h3 Ng4 14.Qf3 Qh4 15.d4 Nxf2 16.Re2 Bg4 17.hxg4 Bh2+ 18.Kf1 Bg3 19.Rxf2 Qh1+ 20.Ke2 Bxf2 21.Bd2 Bh4 22.Qh3 Rae8+ 23.Kd3 Qf1+ 24.Kc2 Bf2 25.Qf3 Qg1 26.Bd5 c5 27.dxc5 Bxc5 28.b4 Bd6 29.a4 a5 30.axb5 axb4 31.Ra6 bxc3 32.Nxc3 Bb4 33.b6 Bxc3 34.Bxc3 h6 35.b7 Re3 36.Bxf7+ 1-0

Important notations and jargons to learn – A chessboard consists of 8files(columns) which are denoted from 'a' to 'h' and 8 ranks(rows) which are denoted from 1-8. The moves are denoted starting with white pieces and followed by black pieces. Pieces and notations – there are 6 different pieces in a game of chess 1)A pawn – there are 8 pawns with each side, A pawn moves only in one direction and captures diagonally.it can also move one or two steps in the first move and after that, it can only one step. It is denoted with the file name on which it is present for example: 1.e4 denotes on the first move white as pushed the pawn on 'e' file from his current position to e4. 2)A Bishop – there are 2 bishops with each side namely

light squared(white) bishop and dark-squared(black) bishop they move diagonally in any direction and any number of steps. It is denoted with a capital letter 'B' for example: 3.Bb5 denotes white has moved the bishop to b5. 3) A Knight – there are 2 knights witch each side and they move in an 'L' in any direction and a knight is the only piece that can move over other pieces. It is denoted with the capital letter 'N' for example: 4.Ba4 Nf6 denotes black has moved to the knight to f6. 4)A Rook – there are 2 rooks with each side and they move in a straight line both horizontally and vertically. It is denoted with the capital letter 'R' For example: 12.Re1 denotes white has moved to the rook to e1. 5)A Queen- there is one queen per side and is the most powerful piece on the board which can move in all directions (vertically, horizontally, and diagonally) any number of steps.it is denoted with the capital letter 'Q' for example: 14.Qf3 Qh4 white has moved the queen to f3. 6)The King-there is one king per side and it is the most important piece on the board one it is captured the game ends and it can be moved in all direction but only one step. It is denoted with the capital letter 'K' for example: 24.Kc2 denoted white has moved to the king to c2. Elo is rating associated with the player, higher rating indicates a better player(mostly).

Other important notations – castling – it is denoted by 'O-O' or 'O-O-O' depending on the side the king moves.

Checks- it is denoted by a plus sign 'Rae8+', it denoted there is a direct attack to the opposite side king.

Captures- it is denoted by a small letter 'x' eg: Nxe5 knight has captured the piece on e5.

Data

We utilized a data-source[1] that provided thousands of games with respect to an Opening. We picked 15 opening/variation for our analysis. The data files were in text format (.txt). We transformed the data into a dataframe with the help of python string manipulation and regex. We also manually created a python dictionary that contained a mapping of an Opening to the move pattern.

Openings we chose: Four Knights, Caro Kann Classic, Qid4e3 (Queen's Indian,4e3, Sicilian Rossolimo, Sicilian dragon other6, KingsGambit, RuyLopezMarshall, ScotchGambit, GiuocoPiano, Reti2b3, Caro-Kann2c4, Caro-KannAdv, Nimzowitsch-Larsen, Caro-KannEx, Modern.

Missing values: The missing values were high for Elo rating of Players. Since it's an important factor in the analysis, replacing these missing values with mean or some other value, might have harmed the analysis. Hence, we decided to drop the rows containing missing values.

Elo Rating: We considered only those games where Elo Rating of both the players were above 2000

Openings: Though the data files were for a respective opening, there were some errors. We dropped the rows where the opening didn't belong to the 15 openings we chose.

Methods

The Dataset created has multiple features ranging from the games ,player, the date the game was played to the moves played for that game's particular round. The final outcome of the game is noted as white black or draw in the result column. Since the problem's target variable is categorical and the data is available in columns, the possibilities of applying machine learning techniques are considered. The simplest classification technique is considered initially to determine the outcome.

Decision Tree

We wanted to analyze the impact an Opening had for a game's result. As the goal was to predict the Game's result based on an opening, we could apply a Classification technique.

Decision tree is a supervised learning method used for classification. It is used to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The library considered for the implementation is from sklearn[2].

Here we considered following features: Opening, player's Elo rating and the target variable was result of the game. We utilized 80% of the data to train and the other 20% to test. Since Decision Tree doesn't work with categorical data, we encoded the opening name with numbers. We built the decision trees varying in depths to determine the outcome of the data considered. The accuracy we obtained was around 51%. We used k-fold cross-validation to for evaluating the model's performance in an hope to witness improved accuracy, but it remained the same.

Chess is a complex game and a game's result is affected by lot of factors other than opening. The classifier model's accuracy supported this inference. The accuracy of the model showed that it is difficult to predict the game result based only on the opening of the game and player's rating.

Approach 2
This dataset contains a significant amount of information that can be used to predict how the game will end. Regardless of the player's skill level, they tend to play the same set of plays over and over again, and by comparing their games with other players, we can identify the most common moves and use this information to identify better strategies for the middle game. These commonalities are kept in an array of played game indexes, which we're using to discover the closest match between all the games we've considered. Games with similar moves would be grouped together and the remaining unique moves would be regarded as possible moves if a certain threshold was met. For this, Python's built-in library for difflib is used, along with the Sequence Matcher module. An object called the similarity object is returned by the sequence matcher module after it compares two strings and returns the percentage of similarity for each two supplied games. If the intensity on the hardware is not excessive, this technique would have been a perfect solution for providing numerical values to the game moves. Exhausted memory occurred while running this strategy on a simple 30 thousand game set. In the meanwhile, there is still room for development in this method. Significance of the method – Using this method, we are able to determine which openings and moves lead to what kind of final game. For every chess player, the endgame method is the most important aspect of winning the game, and knowing what to expect in the endgame simply by glancing at the opening and starting moves will undoubtedly aid in the player's success.

Approach 3
Frequent Pairs Technique It is necessary to identify frequently recurring consecutive movements from the considered set of opening PGNs in order to provide ideas for the following move to play or take into consideration. In order to forecast the moves, we take into account all of the moves that have been made on the board and the color of the pieces that are being used. It was possible to compare the suggestions produced by the model with those offered by a popular online chess site move suggestions provider, and the results were positive. The screenshots of the results obtained uploaded in the images folder. Significance of the method – "The most powerful weapon in Chess is to have the next move"- Emanuel Lasker (world champion). In chess games, the majority of moves follow a predictable pattern, and having a large database to anticipate outcomes, the results that we achieved are of tremendous benefit to the player.

Comparisons

Decision Trees: We used Decision tree to analyze the impact of Opening moves for a game's result. We performed two approaches. The first approach limited the Features to Elo Rating of players and Opening. The second approach considered first 10-20 moves and Elo Rating of players as features. Both the models had accuracy around 50%. We modified the depth of decision tree and also used K-Fold cross variation in a hope to achieve higher accuracy, but unfortunately the accuracy remained the same.

Generalization of Chess board Method:

Accordingly, we phrase the task as detecting whether the outcome of previous games is "win," "loss," "draw," or none of the above. Because the game data is coded, finding a model that worked well with it was a major concern. More or less all of the machinery. In order

to improve forecasts, models make use of numerical values. Hence The data could be generalized if it was possible to add numerical values to the text. The bag of words method is a good place to start. Converting text into a vector Every instance of this word in the document is treated as though it were a single, distinct entity. It is possible that the chess moves data does not contain only one unique word. The game's flow is determined by the moves that came before them, and this flow shifts as the game goes on to the finish of the story. As a result, we're looking towards fusing the two in order to give our acts more weight. Word to vector conversion algorithms - A positional code is generated using this procedure for each item to determine the current board alignment and generates a 32-character code representing the game's current board position. The advantage of using this strategy is that previous games would focus just on the current move, disregarding any earlier actions that contributed to the current board arrangement. It is possible to compute the best move based on the current alignment of the pieces regardless of when the game begins. Navigating between games could be kept as a pair of key-value pairs in order to decide the optimal next step. The mismatch in data length has also been corrected. There is an additional charge for any enhancements or issues. Every step is recorded. Every game's data grows at an exponential pace, and the only way to keep up is to develop new code for each arrangement and then update the historical game library with it. The potential of this approach is currently being studied.

why it doesn't work - Team members came up with an in-house solution for determining the most common moves and pairings of pieces using the Considered approach. In the end, this method's complexity and verbosity led to the creation of a cluttered and verbose set of data that was difficult to read and comprehend. After considering the game's characteristics, we came up with a straightforward approach to indicate each one.

Zero Shot learning – One shot and zero shot learning approaches are investigated for the practicality of neural network-based methods for predicting outcomes. Zero shot learning's prerequisites - a comprehension of the game's rules and a working knowledge of how a neural network is implemented - severely limited the study of alternative approaches. Also, Since the considered problem is of classification type, other approaches of regression and clustering are ignored. since most of the Classification algorithms are tree based and the simplest of the model being Decision Tree Classifier, the dataset is highly explored using Decision Tree method as the training and testing time is fastest for Decision Tree Classifier.

Example Analysis

Finding out how much of an impact an opening has on a game's outcome is one of our first analyses. We used Decision Trees using Elo Ratings of Players and Openings as features since we needed to perform predictive analysis. The model's accuracy was shown to be low, proving that a game's outcome cannot be predicted just on the basis of its Opening. In the second way, we've tried forecasting what kind of game we can obtain from any given opening, which will help a player understand what kind of game he's getting into and direct him to the best strategies and tactics available in that game. Image of the result have been uploaded in the git-hub repo. For the third method, we devised a way for predicting the next probable move based on the previous moves made on the board. This method will aid the player in searching for possible moves made in that particular location, as knowing the next move to make is crucial to victory. Image of the result have been uploaded in the git-hub repo.

Conclusions

Finally, we can state that there is a lot of data that can be gleaned from games that have been stored for a long time. Our tactics are most suited for newcomers to the game who lack a thorough understanding of the subject matter. For a grandmaster, our algorithm would be of little help, given there are already several AI-based chess programs on the market, as well as room for improvement in our approaches.

References

[1] https://www.pgnmentor.com/files.html#modking
[2] https://scikit-learn.org/stable/modules/tree.html