CSE 519 - Data Science Fundamentals

PROJECT PROGRESS REPORT

How Good is a Chess Player?

1 Objective

To build a model that will predict the Elo score of a chess player. Additionally, we are motivated to guess the type of game depending on the time limit of the game and to predict the quality of the play looking at the final board.

2 Introduction

Chess is a two-player strategy board game played on a checkerboard with 64 squares arranged in an 8×8 grid. Each player begins with 16 pieces: one king, one queen, two rooks, two knights, two bishops, and eight pawns. Each piece type moves differently, with the most powerful being the queen and the least powerful the pawn. The objective is to checkmate the opponent's king by placing it under an inescapable threat of capture. To this end, a player's pieces are used to attack and capture the opponent's pieces, while supporting each other. During the game, play typically involves exchanging pieces for the opponent's similar pieces, and finding and engineering opportunities to trade advantageously or to get a better position. In addition to checkmate, a player wins the game if the opponent resigns, or (in a timed game) runs out of time. There are also several ways that a game can end in a draw.

The Elo rating system is a method for calculating the relative skill levels of players in zero-sum games such as chess. Since its development, the system has been adopted with various modifications by many national chess federations. Today it is impossible to imagine tournament chess without a rating system. This system calculates for every player a numerical rating based on performances in competitive chess. A rating is a number normally between 0 and 3000 that changes over time depending on the outcomes of tournament games. When two players compete, the rating system predicts that one with the higher rating is expected to win more often.

In this project we are concerned with predicting the chess rating score (Elo) of both players in a game, given a record of the match. In addition to predicting the Elo score we also predict the type of the game by looking at the time taken for each move.

3 Data Processing

3.1 Lichess Dataset

Lichess is an open-source internet chess server. The data of all games since 2013 is available in LiChess Database. There are over 800,000,000 games in the database, each tagged with ratings of both players and the speed.

Following data fields are available in the Lichess database for every game:

1. Event
2. Site
3. Date
4. Round
5. White
6. Black
7. WhiteTitle
8. BlackTitle
9. Result
10. BlackElo
11. BlackRatingDiff
12. ECO
13. Opening
14. Termination
15. TimeControl
16. UTCDate
17. UTCTime
18. WhiteElo
19. WhiteRatingDiff
20. Moves

The format of the data in LiChess Database is Portable Game Notation (PGN). PGN is a plain text in computer-processable format for recording chess games (both the moves and related data), supported by many chess programs. PGN is structured "for easy reading and writing by human users and for easy parsing and generation by computer programs." The chess moves themselves are given in algebraic chess notation. Each game contains a header file and a sequence of moves made by each player.

```
[Event "F/S Return Match"]
[Site "Belgrade, Serbia JUG"]
[Date "1992.11.04"]
[Round "29"]
[White "Fischer, Robert J."]
[Black "Spassky, Boris V."]
[Result "1/2-1/2"]

1. e4 e5 2. Nf3 Nc6 3. Bb5 a6 4. Ba4 Nf6 5. O-O Be7 6. Re1 b5 7. Bb3 d6 8. c3
O-O 9. h3 Nb8 10. d4 Nbd7 11. c4 c6 12. cxb5 axb5 13. Nc3 Bb7 14. Bg5 b4 15.
Nb1 h6 16. Bh4 c5 17. dxe5 Nxe4 18. Bxe7 Qxe7 19. exd6 Qf6 20. Nbd2 Nxd6 21.
Nc4 Nxc4 22. Bxc4 Nb6 23. Ne5 Rae8 24. Bxf7+ Rxf7 25. Nxf7 Rxe1+ 26. Qxe1 Kxf7
27. Qe3 Qg5 28. Qxg5 hxg5 29. b3 Ke6 30. a3 Kd6 31. axb4 cxb4 32. Ra5 Nd5 33.
f3 Bc8 34. Kf2 Bf5 35. Ra7 g6 36. Ra6+ Kc5 37. Ke1 Nf4 38. g3 Nxh3 39. Kd2 Kb5
40. Rd6 Kc5 41. Ra6 Nf2 42. g4 Bd3 43. Re6 1/2-1/2
```

Sample PGN Game Data with Minimum Headers

The month-wise data files available in the Lichess database is in bz2 format. We first had to convert the file from bz2 to pgn format using the bz2 python library to get the header fields and moves as shown above. Parsing of the pgn data was done using the chess.pgn python library for further processing.

3.2 Pre-processing

The original raw data obtained was further processed using the following steps:

3.2.1 Drop Unwanted Columns

We dropped columns like Site, Round and Date since it doesn't contribute to the analysis. The columns 'WhiteTitle' and 'BlackTitle' contain a significantly large number of NaN values and hence we dropped them as well.

3.2.2 Extract Information from Moves

Split first four moves of the game into four different columns; namely White1, Black1, White2, Black2. These are used for analysis of most frequent opening moves.

Clock gives us the information about the time remaining for the current player at each move. This is present in the 'Moves' column which is required to predict Game Type and also gives us information about the level of a player.

We also obtain Stockfish Analysis Evaluation from the 'Moves' column. Stockfish is a chess engine that is used to evaluate any move in a game transcript. This field is available for only 15% of the data.

3.3.3 Replace NaN values

Few columns had NaN values in some rows and few columns had values as '-' or '*' instead of NaN. Since such rows were significantly less as compared to the size of the dataset, we dropped these rows.

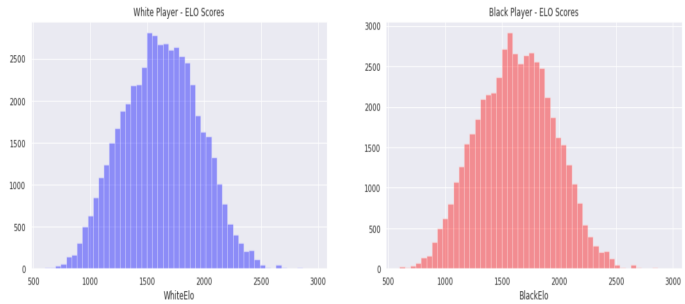
3.3.4 Label Encoding

In order to run Machine Learning algorithms, we used Label Encoding to convert the columns with categorical values to numeric values.

4 Analysis

4.1 Elo Score

Elo Score is used to calculate the relative skill levels of players. From the analysis given below, we can see that the Elo Scores are distributed in the range of 500-3000 and majority of the players have Elo Scores between 1500-2000.



Elo Scores of all Players

This feature is used for tournament sectioning, prize eligibility and pairing purposes in tournaments. It avoids pairing candidates who are most likely to win a tournament during the earlier rounds of the tournament.

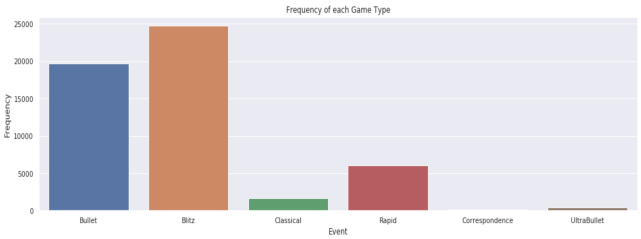
4.2 Game Type

Types of Games are UltraBullet, HyperBullet, Bullet, Blitz, Rapid and Classic.

Rapid: Each player is given less time to consider their moves than a classic tournament time controls allow.

Blitz: Players have three to five minutes to make all of their moves.

Bullet: Players have less than three minutes to make their moves. Ultra and Hyper Bullet are shorter variants of Bullet chess.



Count of each Game Type

The most popular type of games are Bullet and Blitz.

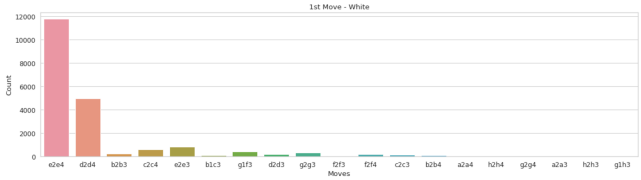
4.3 Moves

This field gives the moves made by each player. It is extracted in the Standard Algebraic Notation. Moves are represented by the name of the piece and the square to which it is being moved.

The first few moves of a chess game, known as the chess opening, are one of the most-studied aspects of the game, largely because of how important they can be.

First Move - White:

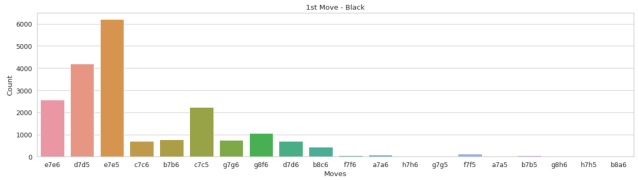
White has a small advantage at the beginning of the game. To maintain this advantage, White should press their advantage to take over the middle of the board as quickly as possible.



Most frequently used First Moves by White

First Move - Black:

Many of Black's opening moves are more defensive in nature and attempt to undermine White's initial advantage.



Most frequently used First Moves by Black

White - 1st Move	Black - 1st Move
e2 e4	e7 e5
d2 d4	d7 d5
e2 e3	e7 e6

Top 3 Moves

4.4 Time Control

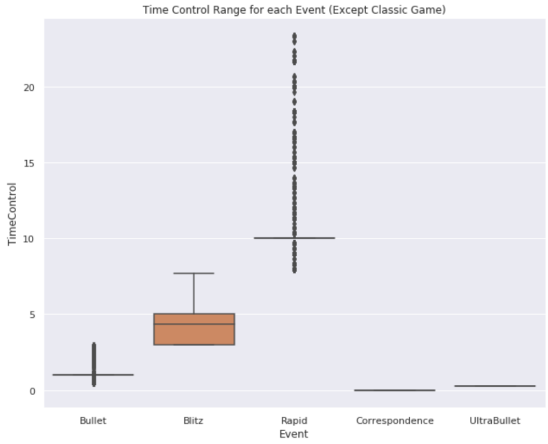
Time Control gives the time limit for the game per player.

Format:

[(Base time) + (Extra Time per move)] seconds

= [(Base Time + (Extra Time * 40))/ 60] minutes

Since there are 40 moves



Time Control Range for each Game Type

4.5 ECO

ECO stands for Encyclopedia of Chess Openings code. It is used to classify the game based on Opening moves.

Code	Freq	Move	Name
A00	3286	b4	Polish Opening
A40	2654	d4	Queen's Pawn
B01	2448	e4 d5	Scandinavian Defense
C00	2118	e4 e6	French Defense
D00	2033	d4 d5	Queen's Pawn Game

Top 5 frequent ECO

4.6 Opening

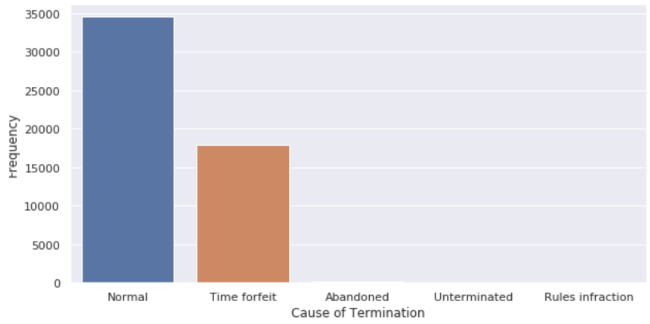
It gives the opening move of the game made by the white player.

Opening Move	Frequency
Modern Defense	1043
Van't Kruijs Opening	1020
Scandinavian Defense: Mieses-Kotroc Variation	958
Queen's Pawn Game: Mason Attack	774
Sicilian Defense	770

Top 5 Opening Moves

4.7 Termination

Termination gives the information about the game ended e.g. Normal, Time forfeit, Abandoned etc.



Frequency of Cause of Termination

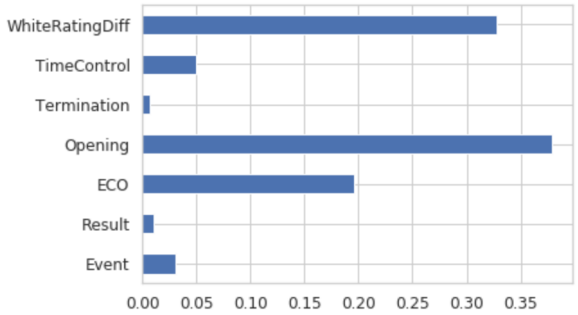
4.8 White / Black Rating Diff

This gives the rating for white/black player based on the Elo ranking.

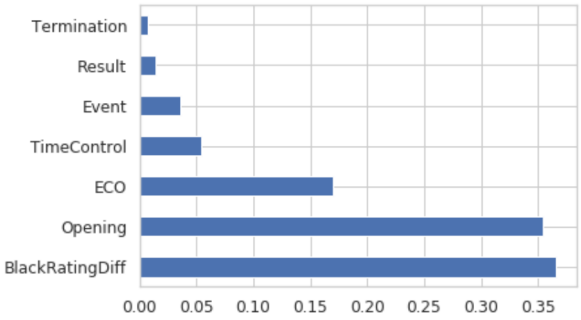
5 Implementation

5.1 Predict Elo Rating of Player from Transcript

To select the best features for building the baseline model, we apply feature importance technique. The features selected with respect to White and Black Elo are as shown in the figures below.



Best Features with respect to White Elo



Best Features with respect to Black Elo

5.1.1 Classification Model

Based on the Elo Score analysis (4.1), we group the Elo Score into three buckets:

- Low Score: Elo Score < 1500
- Mid Score: 1500 <= Elo Score <= 2000
- High Score: Elo Score > 2000

Elo Score Distribution

We have trained the WhiteElo score and BlackElo score on the same training data. The only difference is the RatingDiff column depending on White / Black Elo. The results for the models are given below:

Model	WhiteElo	BlackElo
Logistic Regression	51.4	51.9
XG Boost	57.1	57.6
Light GBM	62.3	62.9

Accuracy of Model (in %)

We can further improve this model by better data analysis and feature extraction which will help us gain better insights for feature engineering.

5.1.2 Regression Model

We have also built a regression model to predict the actual Elo Score of a player. We used the same data as the classification model. The results for the models are given below:

Model	WhiteElo	BlackElo
Linear Regression	34.1	34.7
XG Boost	39.0	38.7
Light GBM	43.6	42.9

Accuracy of Model (in %)

Currently, we have used the data of only 50,000 games. Further, we plan to improve this model by training on the larger available dataset and thoroughly analyzing the data.

5.2 Predict Game Type from Transcript

The Game Type depends mainly on the time information. From the game transcript the most important features needed for this task are TimeControl and Clock. The total time allowed is different for each game type.

Game Type	Time
UltraBullet	< 15 seconds
HyperBullet	< 30 seconds
Bullet	< 3 mins
Blitz	3 - 14 minutes
Rapid	15 - 25 minutes
Classic	25 minutes - 4 hours

Time for each game type

Also, each player has a different time limit to perform each move in these different games. We have the time taken by each player to perform each move extracted from the 'Moves' column in terms of the 'Clock' feature. We use classification to predict the game type based on the clock. The results obtained are as follows:

Model	Accuracy
Linear Regression	48.2
XG Boost	53.6
Light GBM	58.4

This model doesn't have good enough accuracy because there are instances where the clock time indicates a very different value than expected. One of the reasons for such unexpected trends can be when a strong player plays against a weak player, the stronger player finishes his move much faster than the allotted time. Such scenarios causes instability in the model. We plan to further improve it by combining the clock with the TimeControl feature, as well as performing deeper analysis on the clock feature to understand such cases.

6 Current Work

Sub-transcript Level Analysis

We intend to perform sub-transcript level analysis to determine the quality of play. This can be done using various factors.

6.1 From final board position

To determine the quality of play, we analyze the final board position. Here we are evaluating the number of pieces left on the board and their positions. This gives us the idea of the moves taken by the players and hence determining the quality of the game.

To obtain the final board position we iterate through each step of the game stored in PGN format. The final step gives us the Extended Position Description (EPD) representation of the final board position. EPD is a standard for describing chess positions along with an extended set of structured attribute values using the ASCII character set.

Example of Final Board Position in EPD format:
r1b1k1nr/pp1p1ppp/8/2p1p3/8/1Pq1Bn2/P1P2P1P/R4K2 w kq -

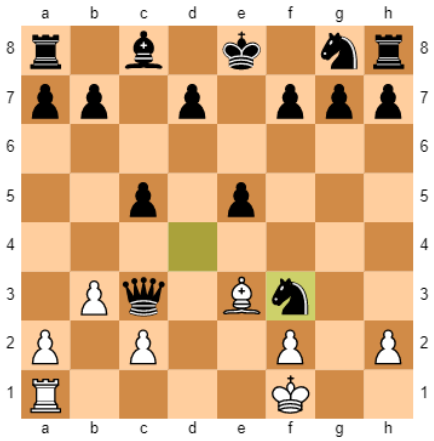


Image Representation of the EPD format

6.2 From time taken to play each move

This can be determined using the Clock feature. This is the anomaly found while predict Game Type. If the player makes a move really fast as compared to the available time, it shows the expertise of the player. Similarly, if a player takes very long to make common moves, it shows the player may be a beginner. When we compare the moves of both players, we can determine the quality of play of that game.

7 Next Steps

In the next part of our project, we focus on improving the data and the initial model to predict the Elo rating of the player more accurately. We will also work on other challenges such as:

- To identify if the player is Human or Computer we have to evaluate the moves of the player. There are two general strategies available to computer chess programs: Brute force search where all possible positions are examined or Strategic AI where only good positions are examined. The computer will always take the best move by examining all the moves possible. Humans tend to make mistakes and may not always choose the best move possible.

- To identify if a player is a beginner or a master, we have to analyze the game duration and move transcript with the expectation that the games between players with huge difference in Elo rating generally finish soon with low chances of upset.

- To predict the games in which the stronger player loses to the weaker one. Based on the current game analysis, we will be able to get the subset of data where the player with high Elo score loses to a player with low Elo score. We hope to find a pattern for this in the subset.