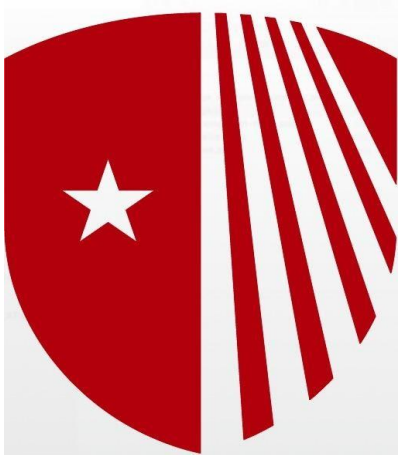
# AMS.561 Introduction to Computational Science



# Title: Chess Data Analysis

Meghana Jayaswamy (112871790)
Nishant Kamat (112818481)
Shwetha Malleshappa (112958664)

Department of Electrical & Computer Engineering
Spring 2020

# 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 of the game and performing classification and regression based on the Elo score

# 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 beer position. In addition to checkmate, a player wins the game if the opponent resigns, or (in a med game) runs out of time. There are also several ways that a game can end in a draw [1].

The ELO rating system is a method for calculating the relative skill levels of players in zero-sum games such as chess. It was developed by the Hungarian physicist Arpad Elo in the 1950's and adopted by the world chess federation (FIDE) in 1970. 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 [2].

The Elo rating 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. The more marked the difference in ratings, greater the probability that the higher rated player will win [3].

In this project we specifically aimed to achieve:

- Conversion of dataset into a comma-separated values file from a portable game notation file for analysis.
- Predicting the Elo rating for white and black players, respectively.
- Predicting the game type.

- Planning to generate different classification models and find their respective accuracy.
- Build a regression model to predict the actual Elo score of a player.

For our team contributions, we worked together on data searching and performing exploratory data analysis. The coding for conversion of PGN data to csv files and feature extraction was done collectively. Feature selection and extraction was also done collectively. Meghana contributed to the prediction of Elo rating of white and black players. Nishant implemented various regression models on the Elo score obtained and Shwetha contributed in the classification of the game type. Everyone contributed in report writing and the making of presentation slides.

# 3. Techniques and Tools

In this project, we used Google Colab as the platform, where python was used as the programming language. The implementation involved few initial techniques which are detailed below.

### 3.1 Data preprocessing:

We have used Lichess dataset for implementing our project. 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 ranks of both players as well as their speed [4]. The format of the data in Lichess database is Portable Game Notation (PGN). PGN is a plain text in data supported by many chess computer-processable format for recording chess games (both the moves and programs).

We have used the dataset available for the month September, 2017.

One of the main challenges was to extract this data and convert it into a readable format for further use. The month wise data files in the database are in bz2 format.

We had to first convert the file from bz2 to pgn format by extracting it using bz2 python library, and the pgn files were then converted to csv files using the chess.pgn library to get the adder fields and moves as shown below.

```
[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**

### 3.2 Feature Extraction:

The next important step after analyzing the data is to extract features relevant to the objective of the project. The features extracted from various headers in the pgn files are listed below.

Following features are extracted using the moves field in the game:

**1. moves**: Every move made in a game. From this, following features were extracted:

1. total_moves: Total number of moves made in a game.
2. white_moves, black_moves: Total number of moves made by white and black players respectively for each game.

**2. clock**: Following features were extracted from clock:

1. time_remaining: Amount of time remaining after the entire game for each player.

**3. SAN (Standard Algebraic Notation):** It contains piece information, end position, capture, check and checkmate information and quality of the move.

1. captures: The number of pieces captured by both white and black players.
2. checks: A Boolean value indicating if checkmate was done during the game.
3. king_castle: Number of king castle moves made by the player.
4. queen_castle: Number of queen castle moves made by the player.
5. pawn_promotes: Number of pawn promotions to special pieces X.

After each move made by the player, we evaluate the board using Stockfish engine. This engine gives us the score by evaluating the positions of the pieces at that time of the game. We extract the following features using Stockfish engine:

1. eval: This gives us the average score of moves for both white and black players.
2. early_game_eval: It gives the average score of the first 25% of the game.
3. middle_game_eval: It gives the average score of 25% - 75% of the game.
4. end_game_eval: It gives the average score of the last 25% (75%-100%) of the game.

### 3.3 Pre-processing

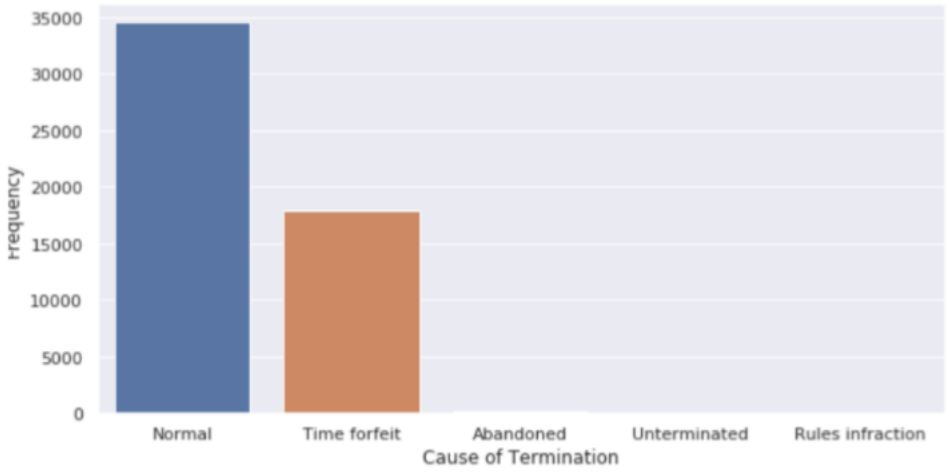
Following with this, the originally obtained raw data was subjected to further processing.

#### 3.3.1 Drop unwanted columns

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

#### 3.3.2 Replace NaN values

Few columns had NaN values in some rows. Since such rows were significantly less as compared to the size of the dataset, we dropped these rows. The rows with Termination = 'Abandoned' were dropped as this data did not add any value to our analysis and very few games were abandoned.




Frequency of Cause of Termination

### 3.3.3 Label Encoding

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

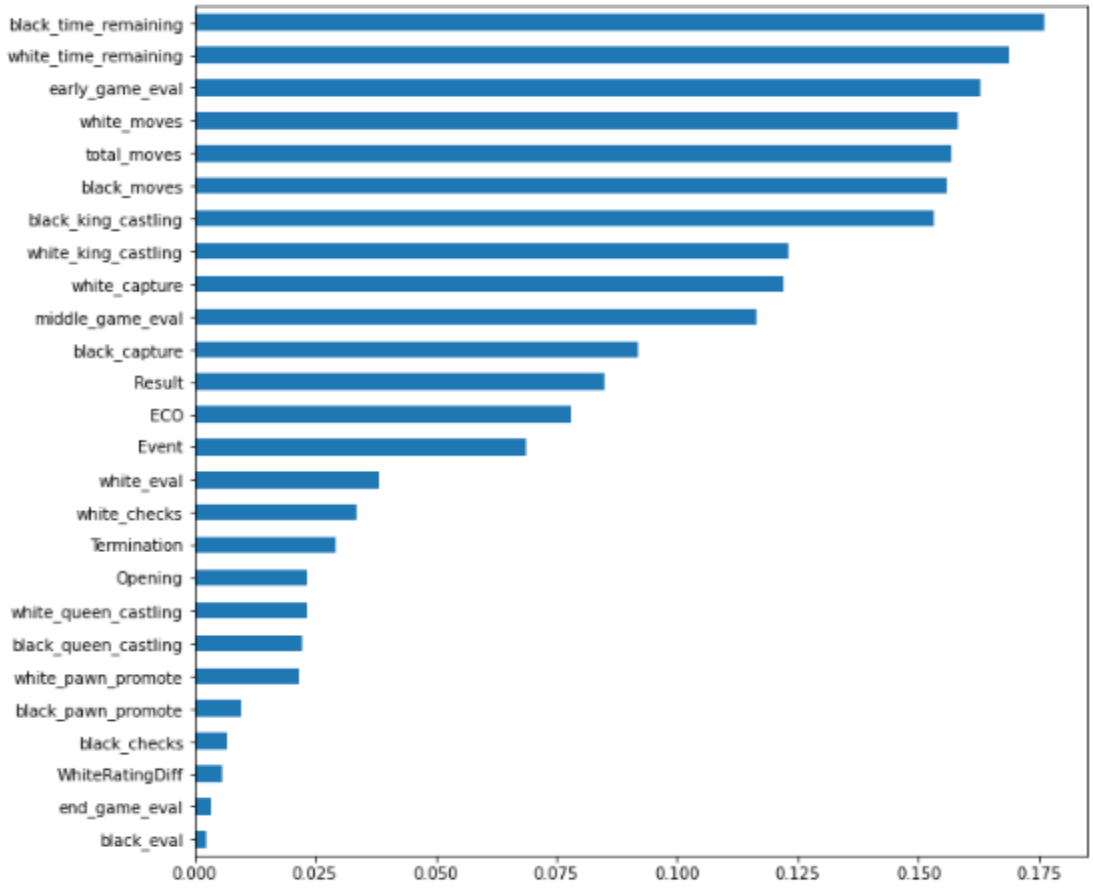
### 3.3.4 Split dataset into white and black datasets

Few features are common to both white and black players but, there are many features like EloRating, eval, etc. which are characteristics for each player. Hence, we create two new dataframes for white and black separately which consists of the features common to both as well as individual characteristic features.

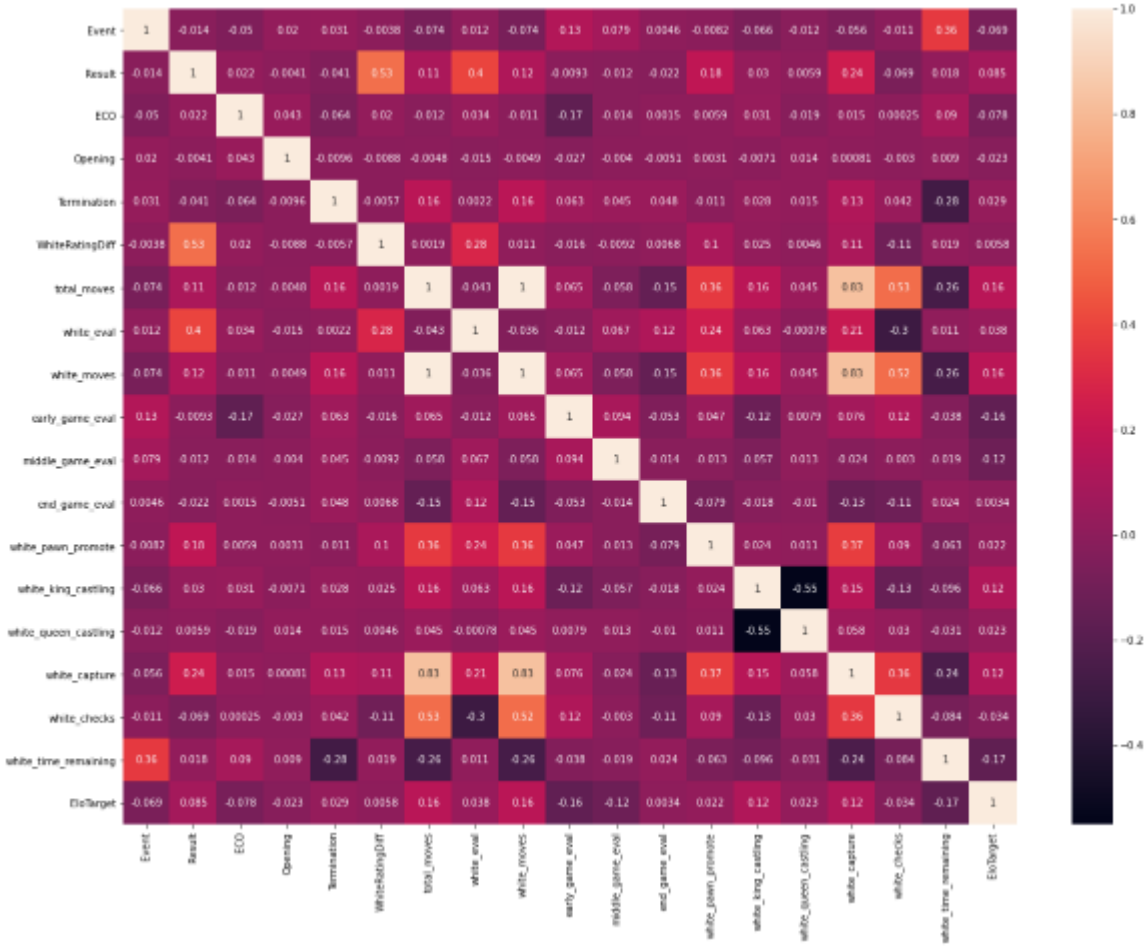
# 4. Implementation and Results

## 4.1. Predict the Elo rating of Player from transcript

The top features which are highly correlated to the features of the white player with respect to White Elo are as shown in the figure below.




**Features with respect to White Elo**




**Correlation matrix for White Elo**

The same was implemented for BlackElo score.

## 4.2. Classification Model

Elo Score is used to calculate the relative skill level of players. Elo scores are distributed in the range of 500-3000 and majority of the players have scores between 1500-2000. 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.

Therefore, we group the Elo Score into three buckets:

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

We have trained the White Elo score and Black Elo score on the same training tuples. The features considered for each Model the features characteristic to the White / Black player respectively and few common features like Termination, Result, etc. The results for the models are given below:

| Model | White Elo | Black Elo |
|---|---|---|
| KNN | 61.55% | 62.11% |
| Decision Tree | 63.29% | 63.89% |
| Random Forest | 69.5% | 68.5% |
| XG Boost | 75.10% | 76.1% |

### 4.3 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 | White Elo | Black Elo |
|---|---|---|
| KNN | 348.5 | 352.6 |
| Linear Regression | 286.55 | 248.7 |
| Random Forest | 285.5 | 258.9 |

**RMSE of Models**

### 4.4 Predict Game Type

Types of Games are Ultra Bullet, Hyper Bullet, Bullet, Blitz, Rapid and Classic.

1. Rapid: Each player is given less me to consider their moves than a classic tournament me controls allow.
2. Blitz: Players have three to five minutes to make all of their moves.
3. Bullet: Players have less than three minutes to make their moves.
4. Ultra and Hyper Bullet are shorter variants of Bullet chess.

From the game transcript the most important feature needed for this task is the 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**

We use classification to predict the game type based on these features. The results obtained are as follows:

| Model | Accuracy |
|---|---|
| KNN | 71.4% |
| Decision Tree | 69.9% |
| Random Forest | 65.6% |
| XG Boost | 80.6% |

Accuracy of different classification models

## 5. Conclusion

In this project we have successfully developed models to predict the Elo Range of the white and black player using Classification and Regression techniques. We have also predicted the Game Type from the transcript with very good accuracy.

# References:


[1] https://en.wikipedia.org/wiki/Chess

[2] https://en.wikipedia.org/wiki/Elo_rating_system

[3] Paras Lehana, Sudhanshu Kulshrestha, Nitin Thakur and Pradeep Asthana - Statistical Analysis on Result Prediction in Chess [https://search.proquest.com/openview/5108034a9fa8212970332620bc1ace84/1?cbl=2026670&pq-origsite=gscholar]

[4] Lichess Game Database [https://database.lichess.org/]

[5] https://python-chess.readthedocs.io/en/latest/pgn.html