Predicting ELO Rating

Predicting ELO Ratings from a game of Chess

Adithya Murali
University of Waterloo
Waterloo, Ontario, Canada
a9murali@uwaterloo.ca

ABSTRACT

Chess is one of the most popular board games across the world. Professional chess players often have an ELO rating, which gives a measure of the player's ability to play chess. A high rating indicates a strong player whereas a low rating indicates a weak player. This rating is assigned to players by FIDE which is an international chess federation. The goal of this project is to estimate the ELO rating of two players by analyzing a game played between them. Linear Regression model and Random Forest model have been used in this project. Spark framework has been used which makes the model scalable. A fairly good estimate of the ratings could be obtained by using the above mentioned models.

KEYWORDS

Chess, ELO, Linear Regression, Random Forest, Spark.

1 INTRODUCTION

The history of chess can be tracked back to 500 AD where a game called *chaturanga*, which is similar to modern day chess was widely played in India [1]. This game subsequently evolved to be modern day chess. Chess is widely considered to be a highly intellectual game which requires excellent analytical and decision making skills. ELO Ratings are used to quantify the strength of a given player. This rating system was developed by Arpad Elo, a Hungarian-American physics professor and an avid chess player (as stated in [2]). It is worthwhile to note that ELO rating is formulated in such a way that it is relative to the pool of players and does not give an objective estimate of a player's strength. The ELO rating system was adopted by FIDE in 1970 and is still in use today [3].

A simplified estimate of a person's rating is given by the below formula [3].

$$ELO\ Rating = \frac{Sum\ of\ opponent's\ rating + 400(Win - Loss)}{Games} \quad (1)$$

However, this is only a close approximation of the actual rating. The actual method used by FIDE to give ELO ratings is slightly more complex and can be found in [4].

The table below shows the category of a chess player based on his rating.

Table 1: Rating-Category (Adapted from [5])

Rating range	Category
2700+	World Championship contenders
2500–2700	most Grandmasters (GM)
2400–2500	most International Masters (IM) and some Grandmasters (GM)
2300–2400	FIDE Masters (FM)
2200–2300	FIDE Candidate Masters (CM), most national masters
2000–2200	candidate masters, experts (USA)
1800–2000	Class A, category 1
1600–1800	Class B, category 2
1400–1600	Class C, category 3
1200–1400	Class D, category 4
below 1200	novices

The highest rated player as of 1st November 2019 is Magnus Carlsen with a rating of 2870 [6].

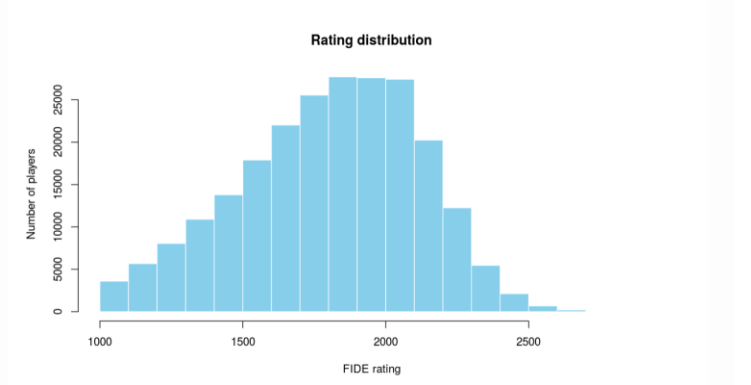

Figure 1: Histogram of FIDE ratings (Adapted from [11]).

Chess engines play an integral part in the proposed model. Deep Blue was the first computer chess engine to defeat the human world champion in 1997 [7]. Chess engines have become much stronger nowadays. Chess engines evaluate a position in chess based on the "centipawn" score assigned to it. A positive centipawn score indicates a stronger position for white whereas a negative centipawn score indicates a stronger position for black. At every move, the engine analyzes the position and calculates a centipawn score. As an example, a move by white player which changes the centipawn score from +600 to -200 is a bad move as

it gives a stronger position to black whereas a move which increase the centipawn score from +600 to +800 is a good move as it further improves white's position. The training dataset used in this project comes with pre-evaluated computer engine scores for each move which reduces the computational load required to train our model.

A game of chess has 3 possible outcomes. It ends in a draw or a win for either white or black.

Summary of Project This project uses the engine score of each move and the result of a game to train a Linear Regression classifier and a Random Forest classifier. We then predict the player ratings using one model on the test set and calculate the accuracy of the model based on mean absolute error metric which is calculated as:

$$MAE = \frac{1}{n}\sum_{i=1}^{i=n} | \hat{y}_i - y_i | \quad (2)$$

Where is $\hat{y}$ the predicted value and y is the actual value. This project was inspired by an online competition held by Kaggle [8].

PySpark is a Python API to interact with Spark framework. Python objects are stored as RDD objects in a distributed fashion by the Spark framework. PySpark was used to develop this project. This project can be found at [12].

2 DATASET

The dataset used in this project can be found in [9]. Training set consists of 25000 games. Test set consists of 25000 games for which the ratings of the players must be predicted.

Each training instance consists of:

- Result.
- White ELO.
- Black ELO.
- Moves in Standard Algebraic Notation (SAN) and Universal Chess Interface (UCI) format.
- Chess engine evaluation for each move of the game using Stockfish chess engine [10]. Each move was evaluated for one second on one core.

Each test instance consists of:

- Result.
- Moves in Standard Algebraic Notation (SAN) and Universal Chess Interface (UCI) format.
- Chess engine evaluation for each move of the game using Stockfish chess engine [10]. Each move was evaluated for one second on one core.

White ELO and Black ELO must be predicted for each test instance which can be submitted to Kaggle for evaluation [8].

3 METHODOLOGY

The features are first extracted from the dataset. Two features are extracted from the dataset. The first feature considered is the game result which can be -1, 0 or 1 which denotes a win for black, draw and a win for white. Secondly, we extract a cumulative game score for all the moves in the game. The formula used to get this score is:

$$\boldsymbol{Cumulative\ score} = \sum_{i=1}^{i=n-1} \boldsymbol{s_{i+1} - s_i} \quad (3)$$

S_i denotes the score at move i and n denotes the total number of moves in the game. A positive cumulative score indicates a good game for white whereas a negative score indicates a good game for black. These features are then fed into a linear regression model and a random forest model.

3.1 Linear Regression

A linear regression model makes a prediction by formulating a linear combination of predictor variables.

$$\boldsymbol{y' = a_0 + b_1 x_1 + \ b_2 x_2 + \cdots + b_n x_n} \quad (4)$$

Where y' is the prediction, x_i is the predictor variable and a_0, bi are the coefficients. These coefficients formulated to minimize the squared error.

$$\boldsymbol{MSE = \frac{1}{n}\sum_{i=1}^{i=n}(y'_i - y_i)^2} \quad (5)$$

There are other functions which can be minimized other than MSE. These regression models such as Lasso and Ridge regression may improve the performance of the ordinary linear regression model but it is not the case in this project from the experimental results.

A linear regression model was trained using the above features. 25,000 training examples was further randomly split into a train set and a test set in the ratio 0.8 and 0.2.

This model gave a mean absolute error of 207.878 for white and a mean absolute error of 212.986 for black on the test set.

The parameters for white were:

- Coefficients = 83.338, -0.004
- Intercept = 2240.786

The parameters for black were:

- Coefficients = -74.054, 0.001
- Intercept = 2249.437

The below image shows the predictions made by the model on the test set.

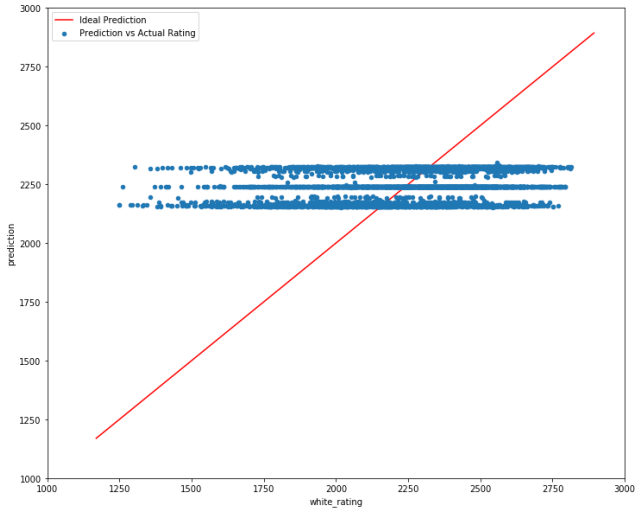

Figure 2: Predictions by linear model (white).

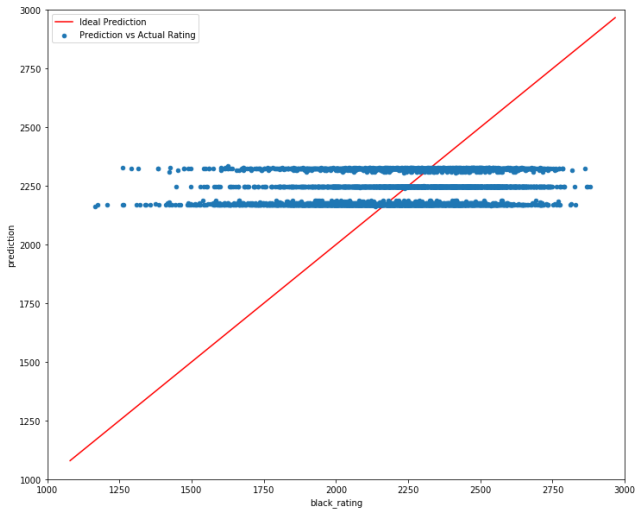

Figure 3: Predictions by linear model (black).

In an effort to improve the model, a new feature which indicates the rating difference between the players was added. This significantly improved the performance of the model which now gave a mean absolute error of 195.023 for white and a mean absolute error of 199.942 for black. The parameters of the model with additional features are:

The parameters for white with additional information were:

- Coefficients = 8.919, -0.002, 0.471
- Intercept: 2244.527

The parameters for black with additional information were:

- Coefficients = 7.331, -0.002, -0.525
- Intercept: 2244.527

The below image shows the prediction made by the model with additional information.

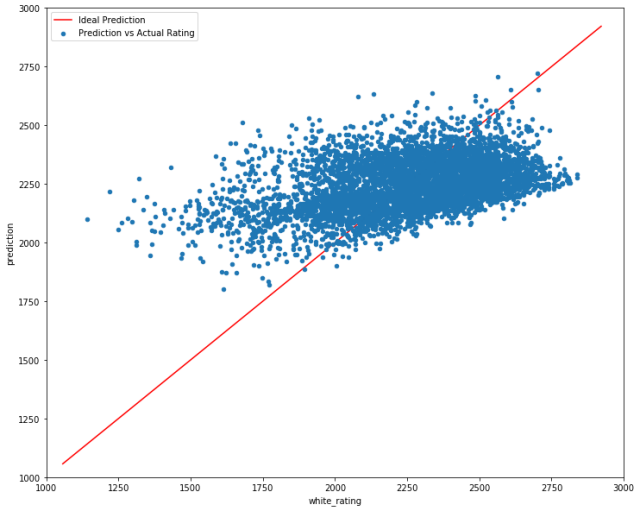

Figure 4: Predictions by linear model with extra features (white).

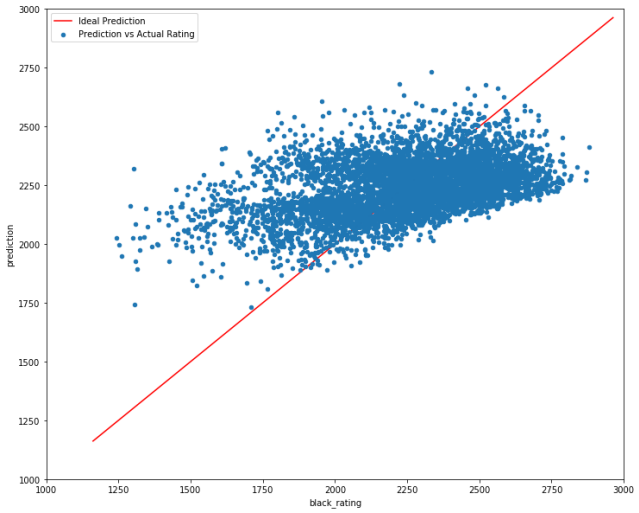

Figure 5: Predictions by linear model with extra features (black).

3.2 Random Forest Regression

A random forest model is an ensemble of several decision trees. Each decision tree is trained on a different subset of the train dataset and the prediction made by the random forest model is the average of all the predictions made by the individual decision trees.

$$y' = \frac{1}{n} \sum_{i=1}^{i=n} y_i \quad (6)$$

Where y' is the final prediction and y_i refers to the individual predicton made by each decision tree.

The mean absolute error for white and black using random forest regression model with 50 trees with a maximum depth of 25 are 204.198 and 208.147 respectively.

The below image shows the predictions made by random forest regression model.

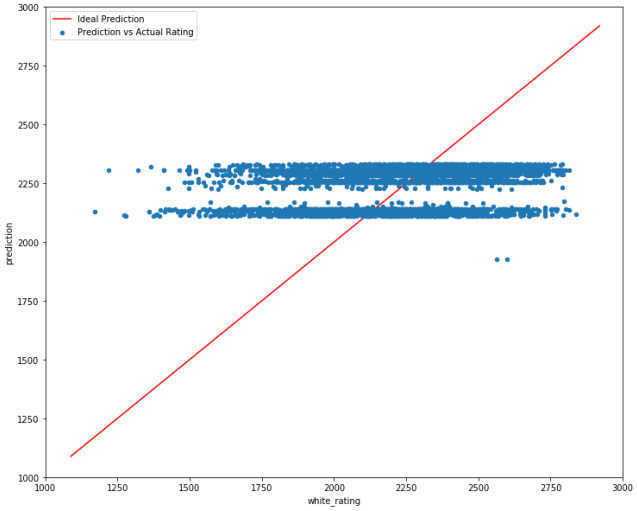

Figure 6: Prediction by random forest model (white).

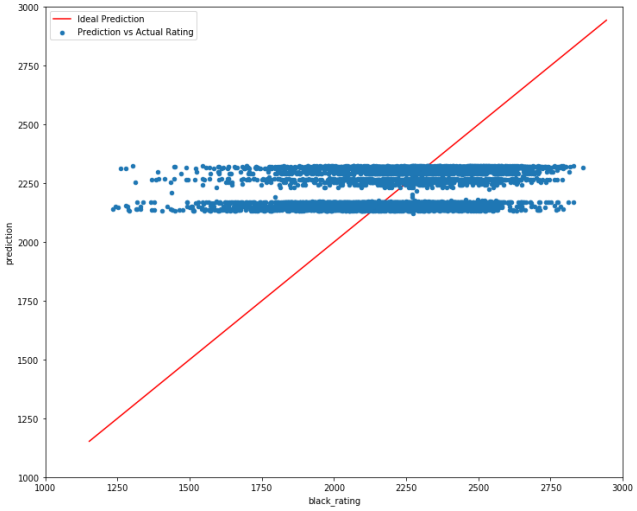

Figure 7: Prediction by random forest model (black).

To improve the model, the same additional information of rating difference between the players is added. This improved the mean absolute error to 182.936 and 182.896 for white and black respectively. The below image shows the predictions made by random forest regression model with additional information.

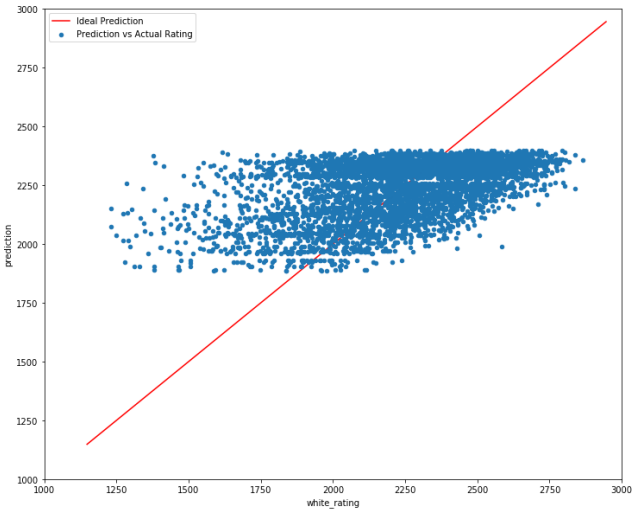

Figure 8: Predictions by random forest model with additional information (white).

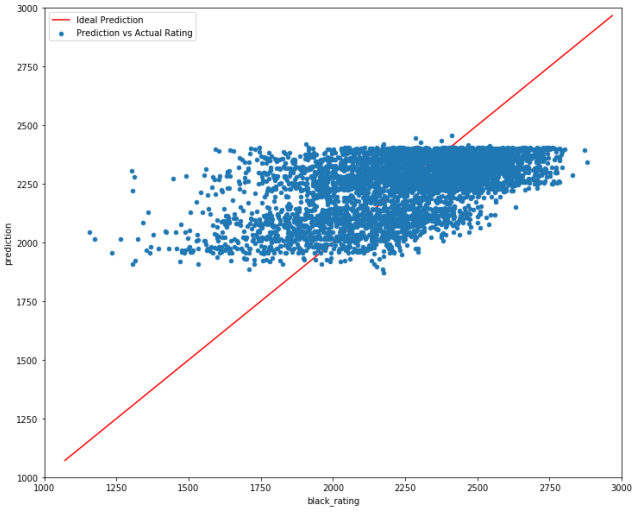

Figure 9: Predictions by random forest model with additional information (black).

It is worth noting that the models with additional feature are not used on the Kaggle test set as it is not possible to extract that feature from the Kaggle test set. The extra feature was introduced to see if it would improve the model and that seems to be the case.

4 RESULTS

Random forest regression gives better results than linear regression. The number of trees and max depth of each tree doesn't significantly affect the quality of the model which suggests that we can reduce the number of trees and its depth to provide a computational boost. Using L1 and L2 regularization on the linear model does not seem to improve the model significantly.

The models without the use of additional information were used to predict the ratings of the 25000 games of the test set and the

results were submitted to Kaggle for evaluation [8].The table below summarizes the performance of the models.

Table 2: Mean absolute error on test set submitted to Kaggle for evaluation.

Model	Mean Absolute Error
Linear Regression	208.81
Linear Regression with regularization parameter = 0.7 and elastic net parameter =0.2	208.77
Random Forest Regression with trees = 5 and max depth = 5	205.30
Random Forest Regression with trees = 50 and max depth = 25	205.00

The top performing model on Kaggle has an impressive mean absolute error of 155.78. A better model could be achieved by extracting more features from the data. One possible improvement to this project could include using a personal chess engine to further analyze the moves made by each player. Grandmasters in chess often make very few inaccurate move and capitalize on good moves. Therefore, we could analyze each move and find if a player has missed an excellent move for a good move. This feature could significantly improve our model at the cost of increase computation and complexity.

Another interesting feature which could be extracted with the aid of data is inaccuracies in opening. Although the total number of moves in chess is large and unexplored, the opening moves in chess have been well studied. Any deviation from the opening theories could indicate that the player is not familiar with nuances of chess opening and conversely, strict adherence to the opening theory suggests that the player is familiar with chess theory and is likely to be an experienced player.

ACKNOWLEDGMENTS

I would like to thank Kaggle for providing the dataset and organizing this competition which served as an inspiration for me to choose this project.

REFERENCES

[1] H.A.Davidson, *A Short History of Chess*, New York, NY: Three Rivers Press, 1981.
[2] T.Graepel and R.Herbrich, "RANKING AND MATCHMAKING -- Grouping Online Players For Competitive Gaming", *Game Developer*, Vol. 13, Issue. 9, pp. 25-34. 2006
[3] https://en.wikipedia.org/wiki/Elo_rating_system
[4] https://handbook.fide.com/chapter/B022017
[5] A.E Elo, *The rating of chessplayers, past and present*, Japan: Arco Pub, 1979
[6] https://ratings.fide.com/toparc.phtml?cod=569
[7] https://www.nydailynews.com/news/world/kasparov-deep-blues-losingchess-champ-rooke-article-1.762264
[8] https://www.kaggle.com/c/finding-elo/
[9] https://www.kaggle.com/c/finding-elo/data
[10] https://stockfishchess.org/
[11] http://nikking.github.io
[12] https://github.com/adithya2208/Project