Chess game rating difference based on game length

1 Introduction

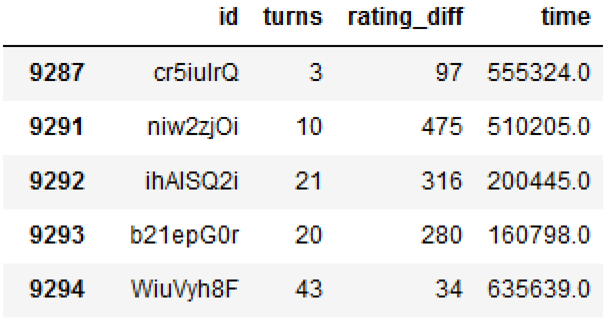
2 Problem formulation

The question I'm trying to answer is given the game length what is the difference in skill between the players. The data points are chess games collected from the chess app lichess. Difference in the players skill will be the label and the length of the game will be the label. The dataset I'm using is from Kaggle [1] and it has a record of over 20000 most recent games taken from users from the top 100 teams on Lichess. However 19% of these games are unrated so the player can't lose any rating. Also the dataset has some outliers and will need some cleaning. In this project I will focus only on the rated games because I believe the games will be played more seriously and provide more accurate data.

3 Methods

3.1 Dataset

The used dataset is from Kaggle [1]. The dataset has a rating for both players ranging from 780 to 2730 and the skill difference will be the difference of these values. For the features we can either use the amount of turns in a game or the duration of the game in milliseconds. The dataset has the start and end times for the games so the time of the game can be derived from these. We will filter out games that last less than 10 seconds and more than 10000 seconds as 10 seconds is most likely immediate concede and 10000 seconds is probably a delayed game or some other error. Every match also has a unique id that we can use to remove any duplicate data points.



	id	turns	rating_diff	time
9287	cr5iulrQ	3	97	555324.0
9291	niw2zjOi	10	475	510205.0
9292	ihAISQ2i	21	316	200445.0
9293	b21epG0r	20	280	160798.0
9294	WiuVyh8F	43	34	635639.0

Fig. 1: Example of dataset and data points

After filtering out the unwanted data points from the set of 20000 data points we are left with about 8300 data points. Most of the filtered data points were unrated games or matches that lasted 0.0 seconds.

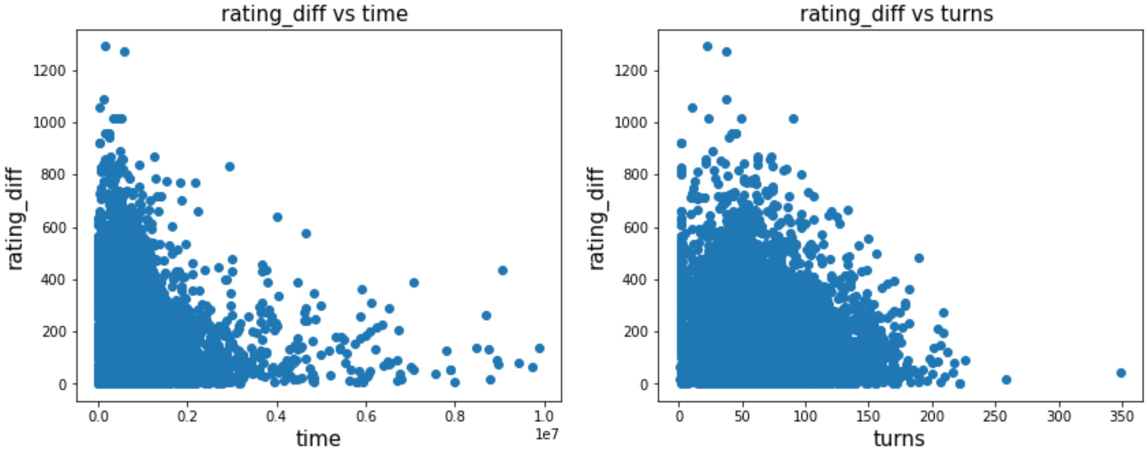


Fig. 2: Scatterplot comparison of the features time and turns

From the scatterplots in figure 2 we can see that when comparing time and turns, time has a better correlation with rating so we will use it as our feature.

3.2 Linear regression model

I will use linear predictor maps as we see somewhat of a linear correlation between the feature and label. The dataset will be divided with a function provided by the scikit-learn package [2] to training set (70%) and validation set (30%). Our loss function will be mean squared error (MSE) as it is often used with linear regression and it is also readily available with the linear regression model in scikit-learn package [2]. We get the mean squared error by summing over all data points the squared difference between the actual value *Y* and the predicted value $\hat{Y}$ given by our hypothesis.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

The loss is calculated for the training and validation sets with different order polynomials. The goal is to get training and validation errors as low as possible. Higher order polynomials may cause overfitting but this can be seen if the validation error is considerably higher than the training error.

4 Results

References

[1] https://www.kaggle.com/datasnaek/chess

[2] https://scikit-learn.org/stable/index.html