GitHub URL

https://github.com/ktadgh/UCDPA_tadghkelly

Abstract

The aim of this project was to use multi logistic regression to predict the result of chess games, based on information available from the PGNs (Portable Game Notation). I would expect players' ELO ratings to be the most valuable predictor, but we also have information on the event type and the round included in the PGN which could also be predictive of the result. This project is based on a trading use-case, so the output I care about is the probability, which could be used to price games (and tournaments, using monte carlo simulation).

Introduction

I chose this project use-case as I work for a bookmaking company so sports modelling is relevant to my work. Chess is also a sport I'm interested in and one with a large amount of data, and an official ELO system which is highly valued by players which makes it uniquely well suited to modelling. The ELO system gives an inherent implied value for expected points, where a win is one point and a draw is half a point. However, I want to predict wins and draws individually so that both outcomes can be priced.

Dataset

As mentioned above, chess games are generally stored as PGNs so I wasn't able to find a relational database with the information I needed. I tried a number of sources, but most didn't include the time control of the game in the PGN. Chess games can be split into four time formats: Blitz, Rapid, Classical and Correspondence. It's reasonable to expect that results behave very differently for different time controls, with Classical games having more predictable results and more draws generally speaking. I wanted to focus on making predictions for Classical games, so I needed a chess database which included the time control of the game. Yottabase (https://www.yottachess.com) was chosen as it's a large free database with all the information I required. The only issue was that games were only available to download by player, so I downloaded games for the current top 20 by classical live rating (https://2700chess.com), converted each to a pandas dataframe with the relevant information, and merged these into one database.

Implementation Process

Database generation

Data Preparation

In the data preparation section I removed all non-classical games from my database. I also removed team games, since I am interested in individual events and I believe that they could skew data due to slightly different incentives for players. I also removed any null values. I replaced the standard strings used in PGNs to signify results with the points associated with the result for the player with the white pieces, using RegEx. Lastly I added a column to the database with white's ELO-implied expected points (https://www.cantorsparadise.com/the-mathematics-of-elo-ratings-b6bfc9ca1dba).

Exploratory Data Analysis

The first thing I wanted to look at was the relationship between the difference between the players' ELO ratings and the points won by the player with the White pieces. I visualised this using a violin plot to get an idea of the distribution for each result (loss, draw and win). Next, I cut the data into bins, so that I could plot the *average* observed points on the x-axis. This showed a clear positive correlation between points and ELO difference (White ELO - Black ELO), but the relationship didn't look linear. Lastly, I did the same but with bins of ELO-implied expected points on the x-axis. This looked more like a clear linear relationship. I calculated the mean squared error and mean absolute error of the ELO-implied expected points versus the actual points, to use as a benchmark to compare my model with.

Multinomial Logistic Regression

To make predictions in the three categories of Win, Draw and Loss, multinomial logistic regression rather than binomial logistic regression was used. I added variables of the squares of the difference, as I thought that the magnitude of the difference might impact the model, regardless of in whose favor. I also added squares of white's ELO-implied expected points and White's ELO, in case those were more closely associated with the result than the original values. The points values were doubled to 0, 1 and 2. The data was split into a train and test set, and a logistic regression model was fit on the training data. The accuracy and log-loss was calculated for the model, accuracy was reasonably high considering there are three possible outcomes, but the log-loss is of more interest.

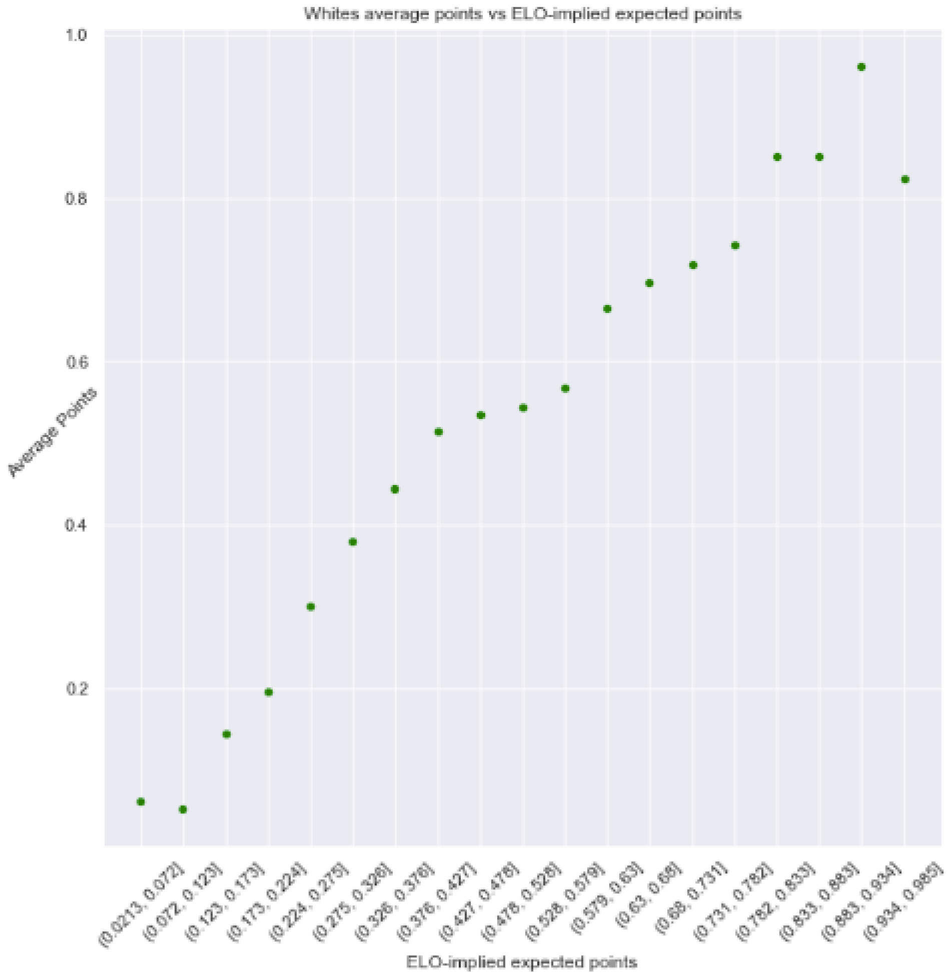
The event type was added as a categorical variable, and rounds were converted to numeric so that they could be added to the model. It would make intuitive sense that both of these could have an impact on the outcome of the game; if the game was part of a swiss tournament for example, it may be the case that a player is more incentivised to draw than they would be in a knockout tournament. Similarly, in the final rounds of a tournament there would likely be more must-win situations, so I would expect more draws in the earlier stages.

Cross-Validation and Hyper-Parameter Tuning

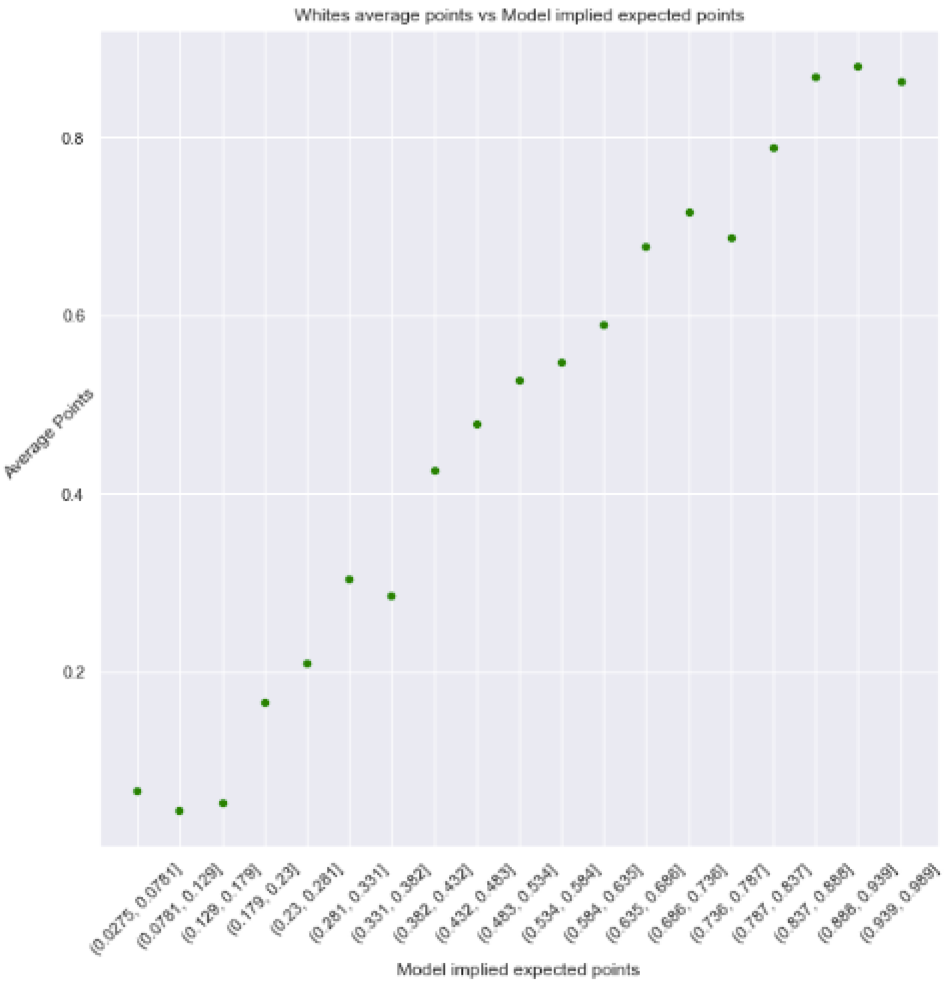
Here round and event type were added to the model, and hyperparameter tuning was conducted using GridSearchCV. Due to the nature of the project, scoring was set to use negative log-loss.

Results

The final model gave a much better log-loss of _ and an accuracy of _



ELO-implied expected points (shown above) had a mean squared error of 0.09902 and a mean absolute error of 0.2396 for the test data.



The model's implied expected points had a mean squared error of 0.09902 and a mean absolute error of 0.2396 for the test data, outperforming the ELO-implied expected points.

Insight

1. Interestingly, looking at the coefficients, it seems like a swiss tournament or a tournament in general makes a decisive result (win or loss) more likely, and a draw less likely.
2. Also, not quite what I would have expected, the chance of white getting a win or a draw seems to increase with the round, while the chance of white losing decreases. This would indicate that the player with the white pieces loses less often later in tournaments. Due to the slight advantage white has, this may not be due to a random (BIAS) in the data, it's possible, for example, that as the standard of play improves so do white's chances.
3. It's interesting that the coefficient of the elo difference (white ELO - black ELO) is positive while the coefficient of its square is negative. When the magnitude of the difference increases the probability of a draw goes down, however when it increases in White's favour it goes up - I would have expected the opposite…
4. It's very cool that the model associates an increase in white's ELO with a *decrease* in the probability of white winning. I take this as a very good sign for the model, meaning it is complex enough to account for the fact that higher rated players will tend to draw more since their play is closer to perfect (and chess, with perfect play, is a draw.)

References

(Include any references if required)