How Do Knights Move?

Predicting the Probability of Winning in Chess

DESCRIPTION

First off, I want to acknowledge that the title above for the project is in reference to a viral YouTube video of reigning world chess champion, Magnus Carlsen. The model itself is not based on how knights move.

The purpose of this project is to build a model using chess data. My hope was to enhance my understanding of the data science process by putting it into practice over a real set of data. My goal was to step outside of the comfort of guided classroom learning and enhance my knowledge through real practical experience. It would have been a much simpler process to model over an organized, curated dataset friendly to model building and variable regularization. However, after watching the World Chess Championship, I was inspired to tackle data I had a true passion for.

Chess is not just any normal game. It's a game of kings with history reaching back many hundreds of years. And up until recently, now a game of data models. With vast databases containing millions of recorded games and with a complex but programmable logical rule system, chess has provided an interesting data problem for data scientists around the world. As only an amateur of data science and chess, I acknowledge that the model I build will not be novel. This project is my first attempt at tackling data outside of a classroom/work environment. I am also a relatively new player to chess. In line with the project requirements, I will only use techniques within the scope of our class and simply use this opportunity to learn through application.

PROBLEM

Our model will attempt to estimate the probability of winning a chess game given variable inputs. This will be a classification problem where we will use logistic regression to fit a model that predicts the probability of winning a game. To maintain scope within lecture module content, I will limit the $\hat{y}$ to either win or not win for white. The target variable is denoted below:

$$\hat{y} = \begin{cases} 0, & \text{if white did not win} \\ 1, & \text{if white did win} \end{cases}$$

For future use case, I could expand my target variable to include cases such as draws and stalemates or even predict the winner's color. Additionally, I could also see if there is any way to predict the win by move (resignation, mate, etc.). Other model enhancements would involve gathering more robust data or applying machine learning models.

DATASET

There were many options for data to use. For instance, there are databases that hold all professional games, but I wanted to scope my project to apply to amateur players. Initially I saw there was a chess dataset available on Kaggle, dating to 2019. This seemed like a good potential option; however, I wanted to practice my extraction capabilities and access the source directly. I found out that there was an API accessible database via Lichess. Lichess is one of the biggest online chess websites and conveniently allow direct access to millions of games. Rather than simply using existing python libraries to pull the data, I decided to recreate the API requests myself to allow greater manual control and more abstract data. The code I created to get this data is provided. Instructions and details are provided in the readme. (Please don't run the API Extraction code unless you want to wait 10+ hours)

In the description of the video you will see tags to each step of my modeling to help with jumping to important sections.

Link to YouTube Presentation: https://www.loom.com/share/886bf5cd68a14449ac6200b75237b43b