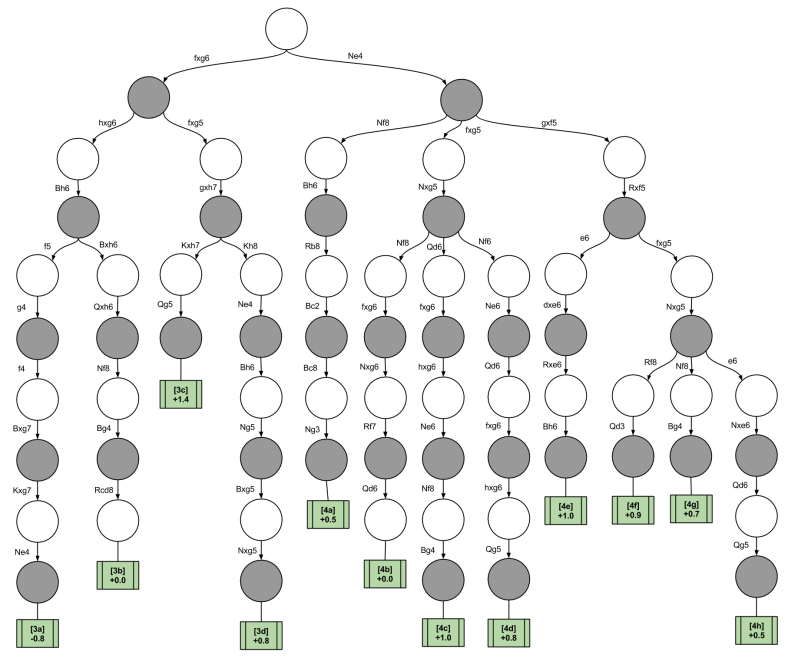
Predicting Chess Matches Based on Opening Moves & Rating

**Progress Report 1**

Throughout the past week, I have primarily researched the algorithms I will be using to complete my project, preprocessed my data, and began looking into examples of applications of the algorithms in Python, specifically JupyterLab and iPython of course.

Conducting research on Decision Trees, I have found that Chess Decision Trees can grow to be quite large quickly.

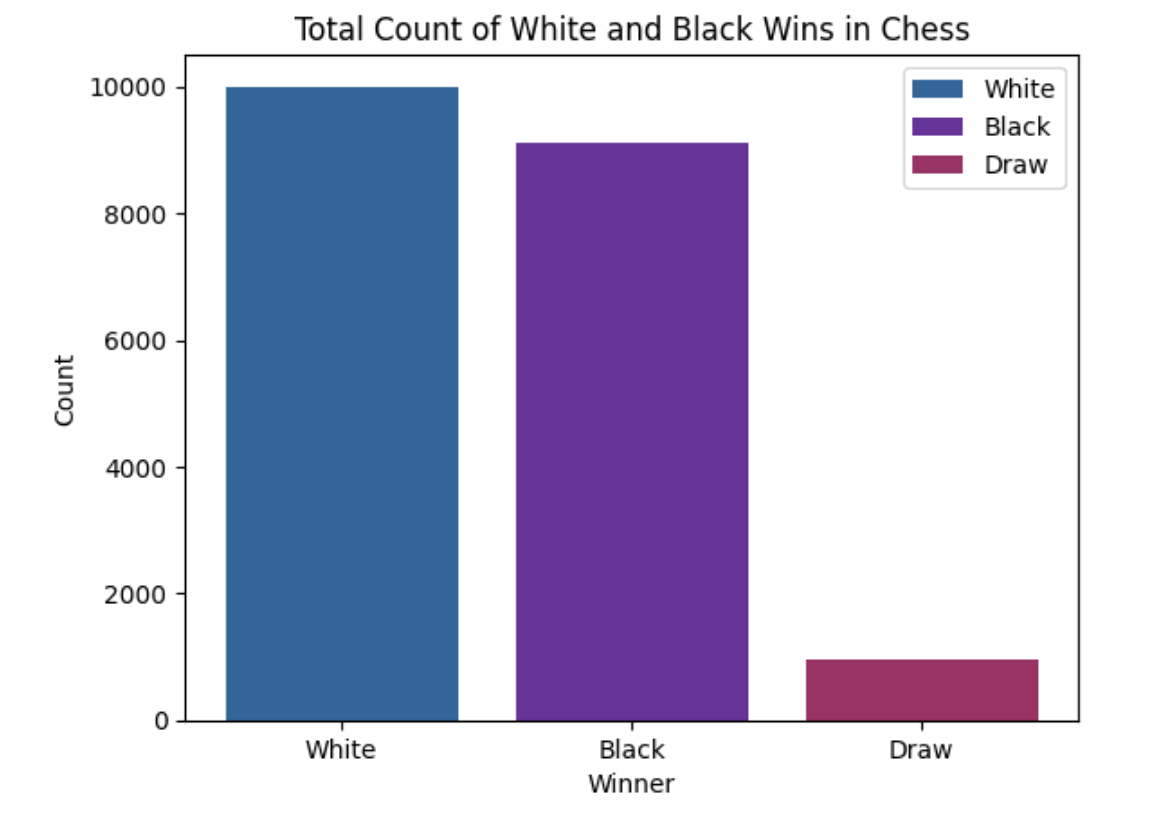


This is a Chess Decision Tree. As one may conclude, this can be tricky or confusing for someone initially researching this algorithm. The Decision Tree nodes alternate between white and black to simulate the turn order in chess, and the leaves represent the end score at that point. In Chess, a positive (+) score means white is ahead, and a negative (-) score means black is ahead. Oftentimes, Chess engines use this notation to indicate who is better off (white or black) in a given position ([What do the Chess Engine Evaluation Numbers Mean?](https://support.chess.com/article/656-what-do-the-computer-evaluation-numbers-mean-like-225)). I’m currently looking into using this evaluation system to predict winners after a specific n-set of moves.

In terms of preprocessing my dataset, I have confirmed that there are no missing values that need to be filled in. As the winner feature in my dataset is categorical, I needed to use encoding to convert it to numerical. Initially, my data used 'white' or 'black' as categorical data to determine the winner of the game. I attempted to use binary encoding to convert 'white' to '0' and 'black' to '1'. However, I later discovered that there was a third categorical datatype for draws, which required me to use one-hot encoding instead to convert the data into numerical.

At this stage, I have not yet decided whether I want to use one-hot encoding to simplify rank distribution to improve predictions. In addition, I have removed any features that are deemed useless, meaning those that do not contribute to the project's goal or the intended prediction. I have decided to keep features such as 'winner', 'white\_rating', 'black\_rating', 'moves', 'opening\_eco', 'opening\_name', and 'opening\_ply', which I believe are relevant to the project's objectives. However, some of these features may also be considered unnecessary in the future, in which case I will drop them accordingly.

Looking at the data, the wins between white and black are close, with a slight bias in favor of white. The number of draws is low, almost rare.



I believe this approach will help ensure unbiased predictions in the future, and draws will not impede the model's prediction ability. Additionally, I'm considering normalizing the rating data to further enhance efficiency if I decide not to use one-hot encoding.

Moving forward, I plan to start implementing the algorithms this week. If these algorithms prove to be inefficient or ineffective, I will conduct further research and explore alternative approaches. I'm excited to start working on the implementation of the algorithms this week, as it's an important step in the project. While I'm hoping that the initial implementation will be successful, I understand that it's not always the case. If I run into any issues, I'll take a step back, re-evaluate, and try to identify the source of the problem. I may need to refine my data preprocessing techniques or explore alternative algorithms to achieve the desired results.