ST1Capstone Presentation Script

# Introduction

Chess is a game that has been played for centuries, and throughout that time strategies have constantly evolved to gain an edge. This project aims to analyse a large quantity of online chess games in order to detect patterns, and to create an algorithm in order to predict the result of a game based on the skill of the competitors and the opening used.

# Dataset Details

The dataset used was sourced from Kaggle, and included over 20000 unique games. There are over 9000 unique players represented in the dataset. The dataset has 16 meaningful columns, which held details on the games rated status, the number of turns, the winner and the method of victory, the time increment of the game, the ratings and names of both players, the moves used, and the details of the openings used.

# EDA Outcomes

The initial graphs of the numerical values of the dataset show that the ratings values have a roughly standardised bell curve, with a sharp peak at the 1500 mark. This indicates that the dataset used has a disproportionate amount of 1500-1550 rated players, which could be caused by a number of factors, such as players rated in that range simply being the most active or most likely to agree to their data being recorded. The turn count peak at roughly 40, and then tapers off, with practically no games ever going past the 200 mark. This matches up with the expected amount of turns in a standard chess game.

-next slide-

While there were outliers in all of the numerical values in the dataset, I chose to only remove extreme outliers in turn count, which I counted as games above 200 turns. A game of chess that goes for over 200 turns is extremely unusual, while players above 2700 rating are still below the level of grandmaster, and thus entirely believable. The graph on the right is turn count after the removal of outliers.

Additionally I chose to remove the ID related columns from the dataset, as the game\_id column was literally just an index column and the white/black\_id columns were people’s usernames, which were not useful for my goals with the data.

-next slide-

Analysing the non-numerical data in the dataset shows that white is favoured to win over black by roughly 4.5%, which matches up with general chess statistics. The most common form of loss is through resigning, while losses through checkmate, time running out or draws together still make up less than half of games.

-next slide-

The most popular categories shown through the opening\_shortname value are the sicilian defense, the French defense, the queen’s pawn game, the Italian game and the queen’s gambit. These are only categories of the opening of a game and can be split into much further categories using the other opening values in the dataset.

-next slide-

For predictive data analysis, my goal was to train an algorithm to predict the winner of a match given the other attributes. I chose to train two separate algorithms for this purpose. The first algorithm, the expanded data algorithm. would exist more purely as a test, using all the other data except for victory\_status, highlighted in both red and yellow, to try and make an algorithm as accurate as possible. The second algorithm, the limited data algorithm is more practical, and uses only the data that can be predicted before a match starts. In practice, this means it excludes the turns and moves attributes, highlighted in yellow. As a note, victory status is excluded because it explicitly states whether the game is a draw, which makes it too easy for the algorithm to predict a draw.

-next slide-

For the expanded data algorithm, The Gradient Boosting classifier algorithm method had a significant advantage over the other algorithms.

-next slide-

On the testing dataset, the expanded data algorithm achieves a 68% accuracy with its predictions. Notably, the confusion matrix shows that the algorithm practically never predicts a draw, but is relatively reliable for predicting a win or loss. The bias towards white can be explained by white’s inherent advantage over black, as shown earlier.

-next slide-

For the limited data algorithm, the SVC algorithm was only slightly better than the gradient boosting classifier algorithm. Comparing the values, it seems the gradient boosting classifier algorithm was significantly impacted by the loss of the data provided previously, while the SVC algorithm remained basically just as accurate as it was before.

-next slide-

The limited data algorithm achieved a 61% accuracy on the testing dataset. Similarly to the expanded data algorithm, it is still seemingly incapable of predicting a draw. However, compared to the expanded data algorithm it is much worse at predicting a black win, only getting it right slightly more than 50% of the time.

-next slide-

For implementation and deployment, I plan to use the limited data algorithm as it gives the tool more of a practical purpose, while the expanded data algorithm exists more as a machine learning exercise. The desktop tool using Tkinter is mostly complete, only missing the decision logic. The web platform tool is partially complete, and is still in the early stages.