HIGH SCHOOL EDITION
Journal of Student Research

Determining Chess Piece Values Using Machine Learning

Aditya Gupta[1], Arnav Gupta[1] and Christopher Grattoni[#]

[1]William Fremd High School, Palatine, IL, USA
[#]Advisor

ABSTRACT

This paper attempts to generate point values for chess pieces, as alternatives to the commonly accepted chess piece values. We use a database of over a million online chess games to heuristically determine the value of a chess piece, by using material imbalances to predict game results. We then explore how piece values change when we analyze material imbalances at various stages of a chess game. As further exploration, we determine what practical values chess pieces and imbalances have at various rating ranges. This creates practical data that players of varying rating can use to aid in chess calculation, as opposed to the rigid values that are typically accepted.

Introduction

Chess has been marked as a game that requires players to display tremendous calculation skills at the highest levels. One major component that aids calculation and analysis of a chess position is the development of chess piece values. The commonly accepted system assigns 1 point to a pawn, 3 points to a knight or bishop, 5 points to a rook and 9 points to a queen [1-5]. This system is rigid throughout an entire game, but only provides a rough state of the game, i.e a well-placed Bishop may be worth more than a passive Rook. Still, they are essential in human calculations when roughly assessing a position.

In this paper, we attempt to derive the practical value of each chess piece by using Machine Learning methods on a chess dataset consisting of millions of online chess games played on the Lichess Platform, an open source chess playing website [7]. Furthermore, we analyze how the value of pieces change when looking at players in only certain rating ranges, thus serving as a more accurate model for players to use when assessing positions.

We attempt to answer the following questions. What are the practical values of Chess Pieces based on what happens in real games? How do the Chess Piece values change when applying Machine Learning methods to various stages of a chess game (are certain pieces more valuable at the end of a chess game, compared to the middle)? How do the Chess Piece values change in various time controls (i.e Bullet, Blitz, Classical)?

We analyze piece imbalances by using statistical methods to determine the effect that various chess imbalances have on the outcome of a chess game. Specifically, we used Logistic Regression to create a Machine Learning model that can accurately predict game outcome based on the piece imbalances in a position, and then use the coefficients of such a model to aid our understanding of the relative value of the chess pieces. We go into a detailed analysis of our approach in Section III. We then analyze our model to determine the values of chess pieces in different game states to further explore questions highlighted in the previous paragraph.

This paper is structured as follows. Section II explores and analyzes previous research done to predict the value of chess pieces. Section III highlights, in detail, the way that we approached this problem, and the methods that our implementation requires. Section IV explains the process to prepare the chess dataset for analysis. Section V shows the results of our chess research, and the various conclusions that we can make from such results.

Literature Survey

Chess Piece values are commonly accepted according to Table 1[1-5].

Table 1. Accepted Chess Piece Values.

Piece	Pawn	Knight	Bishop	Rook	Queen
Value	1	3	3	5	9

These values have primarily been developed with the analysis and expertise of chess masters throughout the ages. Notably, Ralph Betza attempted to determine the ideal and practical value of various chess pieces, including pieces not present in standard Chess, using various factors such as average mobility, color boundness, type of movement, leveling effect, etc. [8]. These ideas can be improved upon by letting computers do the approximations, instead of just humans armed with practical game knowledge.

There have been a few previous approaches to approximate these piece values heuristically, using mathematical models. Jack Good attempted to approximate piece values using assumptions of various parts of the game, although the lack of verifiability in assumptions may have caused inaccurate piece values [6]. Recent approaches take advantage of available chess game datasets [7], applying various data-fitting approaches. One approach attempted to analyze how chess piece imbalances may predict the evaluation offered by engines by generating and analyzing computer engine chess games [9]. The research expressed concerns that the engine evaluations may have influenced the results and piece balances that occurred, although this was addressed by the author [10]. The results from the mentioned research were chess piece values that were remarkably close to the accepted piece values offered in Figure 1. Rasmus Bååth, in his online blog, similarly attempted to fit chess piece imbalances to predict game results, and his results too were similar to accepted values [11].

In previous research in the field, it has been established that chess piece values mostly hover around the accepted piece values offered in Figure 1, although there are outliers, like Grandmaster Larry Kaufman's evaluation of the pieces [12]. Kaufman assigned 1 point to a pawn, 3 ½ points to a knight or bishop, 5 ¼ points to a rook and 10 points to a queen, although his valuations changed slightly throughout the years [12]. As data science tools become more sophisticated, there has been a shift from using expert chess player analysis towards statistical tools that take advantage of large chess game datasets. Such an approach may lead to more practical chess piece values, although care must be taken to use quality data. There may also be certain limitations when these evaluations are generalized to all players and games.

Dataset Preparation

We utilized the Lichess Database in our research [7]. This is a free database of all online games played on the Lichess.org platform. We used the August 2014 version of the database, which contains 1,013,294 games. The database required significant work before it was ready for our analysis.

We first converted the data of the Lichess dataset into a usable Pandas data frame. PGN (Portable Game Notation) refers to a standard plain text format for recording chess games. It stores all the moves of a chess game. The PGN typically also stores other information like the names of the players, the setting of the game, the time control, the game's result, and so on. For our research, the PGN is useful as it allows us to play through chess games by following the moves described in the chess PGN, thus allowing us to determine the piece imbalances that exist. The mode refers to the time control/variant that the game was played in i.e., Bullet, Blitz, Classical. The result column tells us who won the game. The average rating column is the average rating of both players in a chess game. The rating difference column tells us the difference between the White player's rating and Black player's rating. The termination type column shows the way in which the game ended i.e.Time Forfeit, Normal.

Table 2. A Few Elements from Chess Data frame

	PGN	Mode	Result	Average Rating	Rating Difference	Termination Type
0	1. e4 g6 2. d4 Bg7 3. c3 d6 4. Qf3 Nf6 5. h3 O...	Classical	Black Wins	1538	-262	Time forfeit
1	1. d4 Nf6 2. c4 g6 3. a3 Bg7 4. Nf3 O-O 5. Nc3...	Blitz	Black Wins	1492	-220	Time forfeit
2	1. e4 c5 2. Qf3 e5 3. Bc4 Nf6 4. Nh3 h6 5. g4 ...	Blitz	White Wins	1413	213	Normal
3	1. e4 e5 2. Nf3 d6 3. Nc3 f5 4. exf5 Bxf5 5. d...	Blitz	Black Wins	1444	-358	Normal
4	1. e4 e6 2. d4 d5 3. Nc3 Nf6 4. e5 Ne4 5. Nce2...	Bullet	White Wins	1931	117	Normal
5	1. e4 g6 2. d4 Bg7 3. c3 d6 4. Qf3 Nf6 5. h3 O...	Classical	Black Wins	1538	-262	Time forfeit

We then analyzed the make-up of the games in the data to get a better sense of how we need to filter it. The dataset has 843,230 normal games, a game that had at least 1 move played. 169,854 games contained computer evaluations. Additionally, 210 games were empty, meaning that there was less than 1 full move before the game terminated.

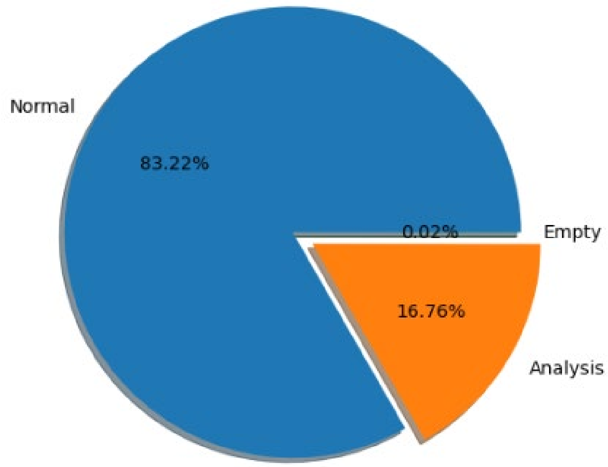

Figure 1. Game Types.

In our analysis, we discarded the games that had computer evaluations or were empty. We then observed the distribution of the game mode. The remaining dataset contained 372,651 Blitz games, 235,828 Bullet games, 232,835 Classical games, and 1,916 Correspondence chess games.

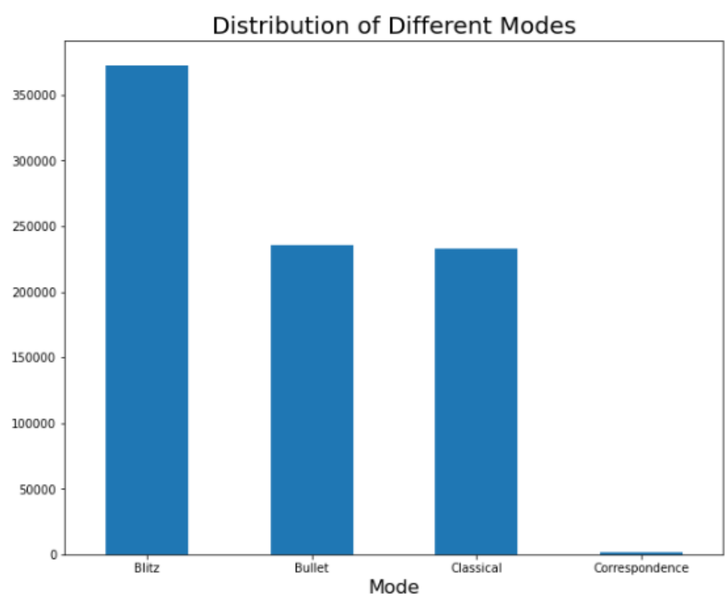

Figure 2. Distribution of Different Modes.

The games have 4 different termination Types: 560,233 games terminated in a “normal” fashion, 282,850 games terminated with a time forfeit, 86 games terminated in abandonment, and 61 games terminated with a rules infraction.

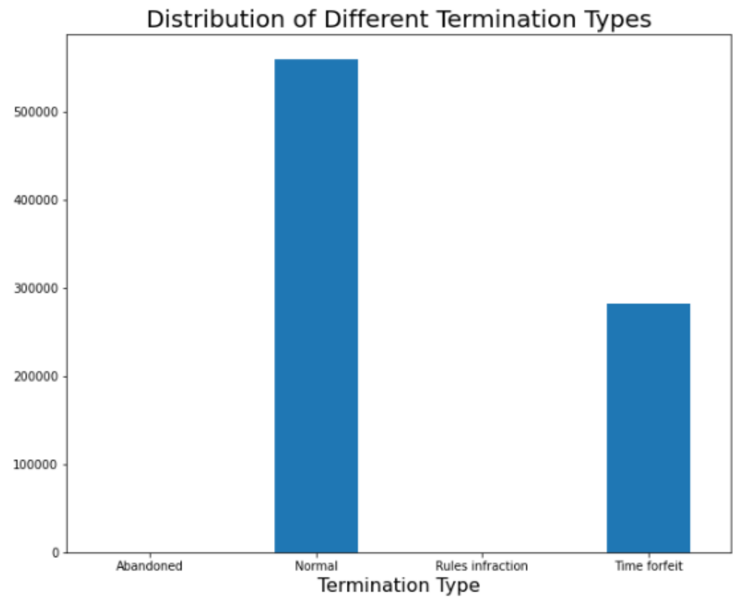

Figure 3. Distribution of Game Termination Types

To prevent different termination types from influencing our results, we consider only games that had a "Normal" termination, i.e. Checkmate, Resignation, etc. We next considered the results of those games. The results had an even distribution with White winning slightly more games, and a small percent of games finishing in draws. To fit the assumption of Logistic Regression that the response variable is binary, we discard games that ended with a draw.

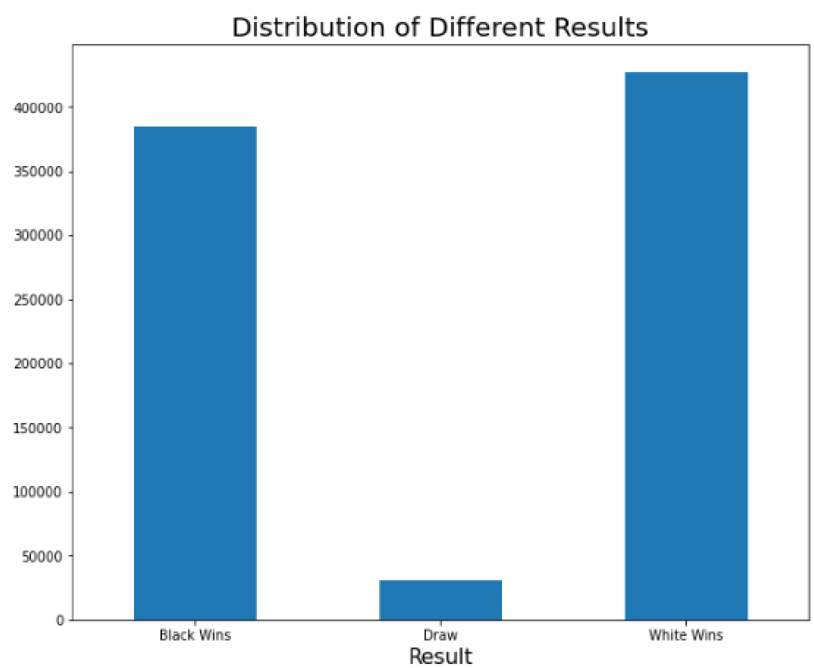

Figure 4. Distribution of Game Results.

We next analyze the ratings of the players. The dataset contains 18,874 beginner games with an average rating below 1250, 385,240 intermediate games with an average rating of 1250-1750, 148,080 advanced games with an average rating of 1750 - 2250 , and 1,437 master games with an average rating above 2250. Since most of our chess games come from the 1250-2250 rating range, the results of our research are most applicable to this rating group.

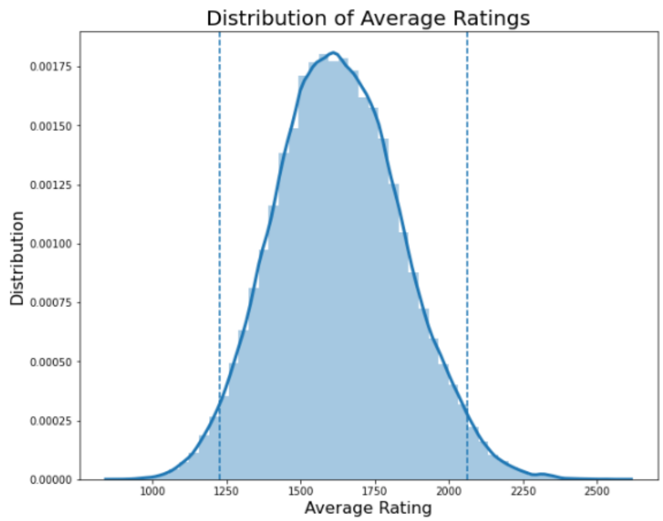

Figure 5. Distribution of Average Ratings.

The final dataset contains 553,631 chess games. In this form, the only information that the dataset contains on how the game played out is the PGN itself. Since it is hard to extract piece imbalances directly from the PGN, we utilize the FEN of the game at various stages of the game [13]. The FEN is a standard notation for describing a particular position of the board. To obtain the FEN of the chess games, we utilized the Python Chess Library[14] and played out the PGN of each game, noting the FEN both halfway through the game and at the end of the game.

A chess FEN position lets us easily check what pieces both the White and Black side have, allowing us to add data columns to account for material differences. For example, we created a column 'Pawn Difference' which is determined by subtracting the number of Black pawns from the number of White pawns. Such an approach was followed for all pieces, independently at the halfway point of a chess game and at the end. Our final dataset contained the following columns:

Table 3. Dataset Columns

Column Name(s)	Purpose
"PGN"	The PGN of the chess game
"Mode"	The time control/variant that the game was played in i.e Bullet, Blitz, Classical
"Result"	The result of the chess game
"Average Rating"	The average rating of both Chess players.
"Half FEN"	The FEN of the chess game after half the moves have been played.

"FEN"	The FEN of the chess game at the end of the chess game.
"White Pawn Half", "White Bishop Half", etc.	The quantity of the respective piece that White had halfway through the game.
"Black Pawn Half", "Black Bishop Half", etc.	The quantity of the respective piece that Black had halfway through the game.
"Pawn Difference Half", "Bishop Difference Half". etc.	The difference in the quantity of the respective chess player halfway through the game. The entry is positive if White had more of the respective piece, negative if Black had more of the respective piece, and 0 otherwise.
"White Pawn Final", "White Bishop Final", etc.	The quantity of the respective piece that White had at the end of the game.
"Black Pawn Final", "Black Bishop Final", etc.	The quantity of the respective piece that Black had at the end of the game.
"Pawn Difference Final", "Bishop Difference Final". etc.	The difference in the quantity of the respective chess player at the end of the game. The entry is positive if White had more of the respective piece, negative if Black had more of the respective piece, and 0 otherwise.

Our dataset now contains the features needed for us to conduct our analysis.

Design and Implementation

To determine chess piece values, we utilized Logistic Regression. This is a helpful technique to fit a regression model when the response variable is binary[15] - in our case, we need to fit the predictive features of a chess game to the result of a chess game: 1 for white winning a game and 0 for white losing a game.

A general Logistic Regression model has the following probability p of the event occurring:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}}$$

Figure 6. The probability p of the event occurring.

In our research, p represents the probability of White winning. We fit such a logistic regression model to various features of our dataset to determine the optimal coefficient $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$ for each feature $x_1, x_2, x_3, \cdots, x_n$. It is important to understand the interpretation of the coefficients $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$ of a logistic regression model to understand how we used them to determine relative piece values. Specifically, a 1 unit increase in x_i will result in β_i increase in the log-odds ratio of the event occurring. Thus, to determine the exact percent difference an increase of 1 unit in a feature x_i will have on the probability p of the predicted event occurring, we would have to manipulate the coefficient β_i associated with the feature by exponentiating β_i and then subtracting 1. For example, a sample $\beta_i = .38$ for a feature x_i would mean that a 1 unit increase in x_i

signifies a $e^{\beta_i} - 1 = e^{.38} - 1 = 1.46 - 1 = .46 = 46\%$ higher chance that the predicted event will occur. Since we are predicting the game results using piece imbalances, our features $x_1, x_2, x_3, \cdots, x_n$ are the piece imbalance features that we created in the previous section. For example, when analyzing the value of pieces in the middle of a chess game, our features $x_1, x_2, x_3, \cdots, x_n$ are the columns "Pawn Difference Half", "Bishop Difference Half". etc. Our approach is to use the coefficients $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$ to determine the relative value that a specific chess piece has at the stage of the game we are analyzing for the various subsets of our data. Since we are interested in determining the relative value of the pieces, and not the exact mathematical predictive power a specific piece imbalance has on the game result, for ease of comprehension, we decided to use the log-odds coefficients $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$themselves, as it led to results that were not only easier to visualize, but also easier to understand.

To implement Logistic Regression, we use the Sci-Kit Learn library[16], which has an inbuilt Logistic Regression model that enables us to fit the desired features to a result of a chess game. We first split the dataset into two parts, the predictive features, and the Result feature, which is a binary feature. We then further split the dataset into two parts: 80% of the dataset became the training data, which is what we use to fit our Logistic Regression model, and the remaining data is used to test the model's accuracy. After fitting the Logistic Regression model to the data, we can then analyze and normalize the coefficients of the predictive model to see what the value of a specific chess piece is.

We conduct the mentioned analysis with the various subsets of our data to see how the value of chess pieces changes at different stages of a chess game, and in various time controls.

We first create a Logistic Model that predicts the results of a chess game based on the piece imbalances that exist at the end of a chess game. We then modify the model to consider imbalances that existed halfway through the chess game, as opposed to at the end of a chess game, giving us better ideas as to how the values of chess pieces change through the various stages of a chess game.

We next conduct similar analysis in various time controls: 'Bullet', 'Blitz', and 'Classical' to see how the predicted values of the chess pieces changes in various time controls. This refined analysis is done by only using games that meet the respective time control when constructing the Logistic Regression model.

The results and discussions of our research are discussed in more detail in section V.

Results and Discussion

We first created a Logistic Regression Model to predict the value that each chess piece has on the outcome of a given chess game. As discussed in Section IV, we can then compare the coefficients $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$ of each piece imbalance for our features $x_1, x_2, x_3, \cdots, x_n$ to determine the relative value of each piece.

Our first Logistic Regression model analyzes the effect of chess piece imbalances in the position at the end of the chess game in all games regardless of time controls. We trained the Logistic Regression model on 80% of the dataset with the following predictive features ['Pawn Difference', 'Knight Difference', 'Bishop Difference', 'Rook Difference', 'Queen Difference'], predicting the ['Result'] column. We analyze the make-up of such a model to get a better sense of both the model and the coefficients that it produces.

Our Logistic Regression model's results on the testing data can be classified in a confusion matrix:

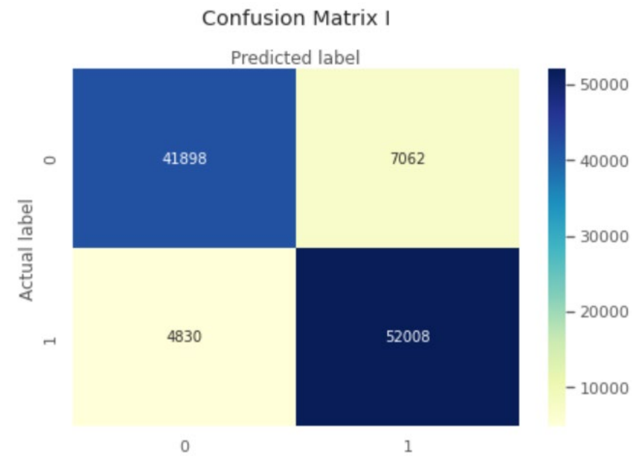

Figure 6. Confusion Matrix I

From this matrix, we can see that the model mostly classifies games correctly based on the piece imbalances. It is important to note that there are many factors that go into deciding the outcome of a chess game, most notably, it is important to realize that one may be up in material but have an otherwise losing position, which may lead to a wrongly classified game in our model. We can see the classification report of this model on the testing data here:

	precision	recall	f1-score	support
White Lost	0.90	0.86	0.88	48960
White Won	0.88	0.92	0.90	56838
accuracy			0.89	105798
macro avg	0.89	0.89	0.89	105798
weighted avg	0.89	0.89	0.89	105798

Figure 7. Classification Report I

We note that the model has an approximately 89% accuracy, which indicates that piece differences at the end of a chess game are generally good indicators of the result. To determine the coefficients $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$ of each piece imbalance for our features $x_1, x_2, x_3, \cdots, x_n$, we can use the coefficients available in the summary of the regression model. These coefficients are [0.74932113, 1.56811632, 1.72501634, 2.38331119, 4.65995062] for each feature [‘Pawn’. ‘Knight’, ‘Bishop’, ‘Rook’, ‘Queen’] respectively. The LLR p-value for these coefficients was significantly small (<.005).

Figure 8. Chess Piece Values At End of Game(Unnormalized)

These coefficients don't resemble the typical chess piece point values that are accepted. Regardless, like the normal chess point values, it is of more importance here what the relative values of the chess pieces are. To show these relative values in a sense that may make more sense to chess players, we normalize these results for various pieces. We first normalize the results for the pawn:

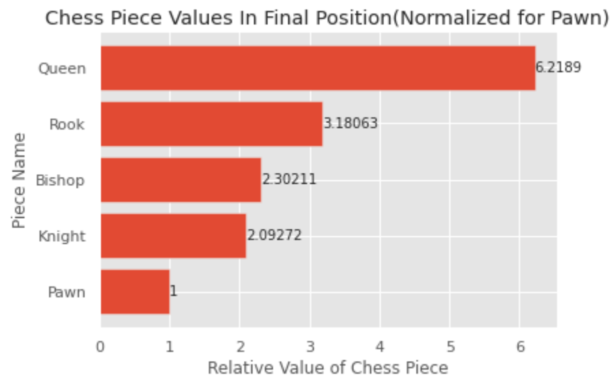

Figure 9. Chess Piece Values At End of Game(Normalized for Pawn)

Normalizing for the pawn still does not resemble the values that are typically accepted. One reason for this may be that pawns tend to become more important as a chess game progresses. At the highest level of chess games, a single pawn may be sufficient for one side to obtain a win. Further research may need to be conducted to see how the value of the pawn itself changes throughout a game and depending on its position on the board.

If we normalize our results to make the knight equivalent to 3 points, we can better compare the relative value of the pieces. This gives us the values [1.43354377, 3., 3.30016909, 4.55956838, 8.9150605] for each feature ['Pawn'. 'Knight', 'Bishop', 'Rook', 'Queen'] respectively.

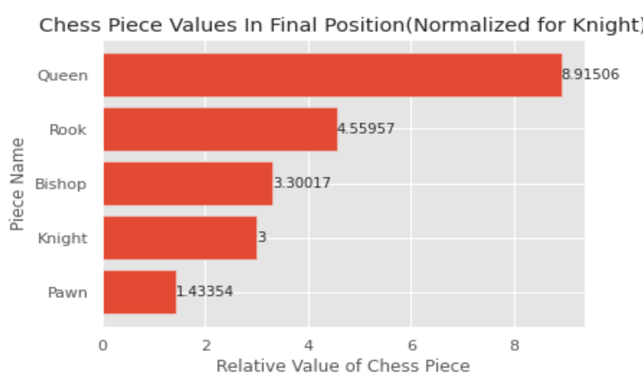

Figure 10. Chess Piece Values At End of Game (Normalized for Knight)

These values are much more typical compared to the commonly used points from Figure 1. These coefficients show us that the value of a Bishop tends to be slightly higher than a Knight, something that is generally agreed upon by chess experts [17]. Modern theory also sometimes suggests that the rook may be worth around 4.5 points, as opposed to the typical 5.0 points i.e. in Artur Yusupov's Build up your chess with Artur Yusupov: The fundamentals [18]. This is supported by our results. We can see that the value of the queen is just slightly less than the generally agreed upon 9.0 points.

Now we will show the results of the same analysis ran on the middle of the games. We do this by utilizing the features ['Pawn Difference Half', 'Knight Difference Half', 'Bishop Difference Half', 'Rook Difference Half', 'Queen Difference Half'] created in Section IV.

Volume 12 Issue 1 (2023)

We once again split our dataset into a 80-20 split for training and testing. Once we have fitted a Logistic Regression model to our training set, we can evaluate its accuracy on the testing set. We create a confusion matrix to see the results of the model's predictions:

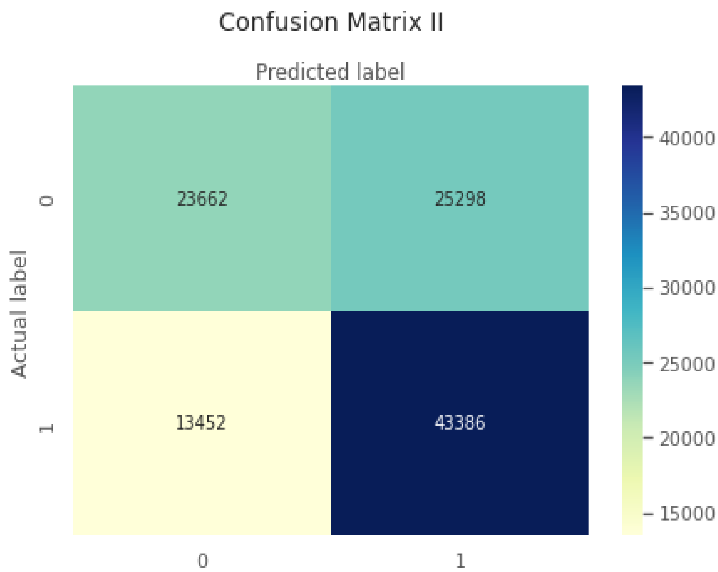

Figure 11. Confusion Matrix II

	precision	recall	f1-score	support
White Lost	0.64	0.48	0.55	48960
White Won	0.63	0.76	0.69	56838
accuracy			0.63	105798
macro avg	0.63	0.62	0.62	105798
weighted avg	0.63	0.63	0.63	105798

Figure 12. Classification Report II

Unlike the previous model, this model classifies many more games inaccurately. In fact, we can see this model has an accuracy of only 63%, which is only slightly better than if the model were to randomly predict the results of a chess game(50%). This may be explained by many possible factors. Unlike piece imbalances that exist in the final position of a game, piece imbalances halfway through may offer other advantages. For example, if white is playing a gambit, they may sacrifice a pawn early game for a positional advantage that will not materialize by midgame. Our analysis would not capture this in its game prediction. For insight, we will still briefly analyze the coefficients of such a model. Like we did previously, we determine the coefficients $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$ of each piece imbalance for our features $x_1, x_2, x_3, \cdots, x_n$.These coefficients are [0.381301, 0.843027, 1.04086, 1.36466, 2.3595] for each feature ['Pawn'. 'Knight', 'Bishop', 'Rook', 'Queen'] respectively. The LLR p-value for these coefficients was significantly small (<.005).

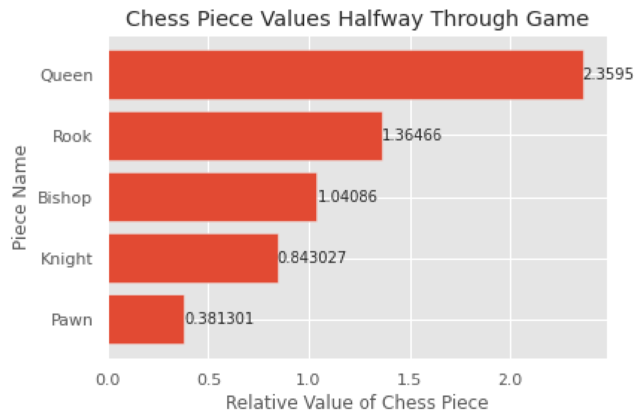

Figure 13. Chess Piece Values Halfway Through Game (Unnormalized)

We again normalize the coefficients to get values that may seem more familiar. We first normalize Pawn = 1 and then to Knight=3.

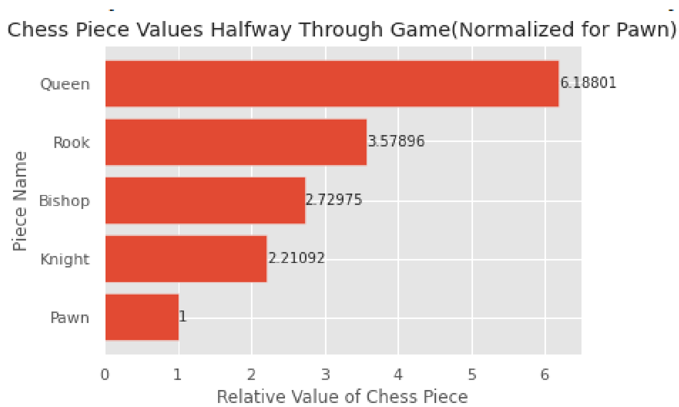

Figure 14. Chess Piece Values Halfway Through Game (Normalized for Pawn)

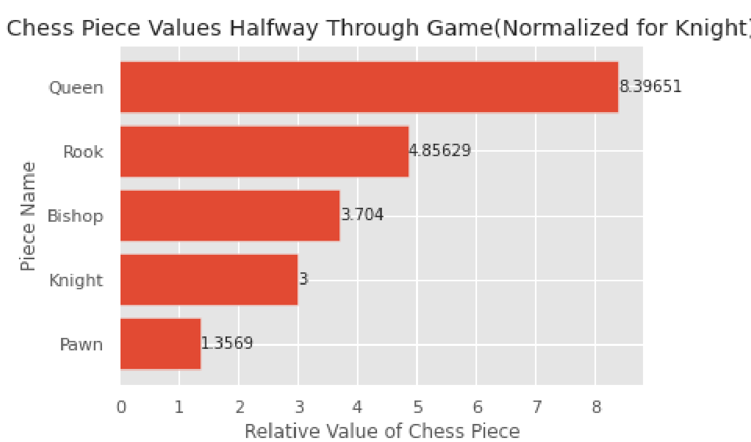

Figure 15. Chess Piece Values Halfway Through Game (Normalized for Knight)

Normalizing the chess piece values for the knight, we get the following chess piece values [1.35690059, 3, 3.70400444, 4.856291, 8.39651183] for each piece ['Pawn'. 'Knight', 'Bishop', 'Rook', 'Queen'] respectively. These values better resemble the commonly accepted values from Figure I. We see that the queen is valued less than it's typical value or the estimated value at the end of a chess game. This can be

HIGH SCHOOL EDITION
Journal of Student Research

explained by the fact that in some games if one player is up a queen, the opponent probably obtained comparable compensation for the material deficit, whether that is in the form of other material imbalances or position advantages.

We next analyze how the value of chess pieces changes as we analyze games in different time controls. Specifically, we take a look at the value of chess pieces in the final positions in 'Bullet', 'Blitz', and 'Classical' time controls, as defined on the Lichess website.

We first start with 'Bullet' time control. The bullet time control refers to games that are quite fast in nature, with less than 3 minutes for each side to play their moves[19].

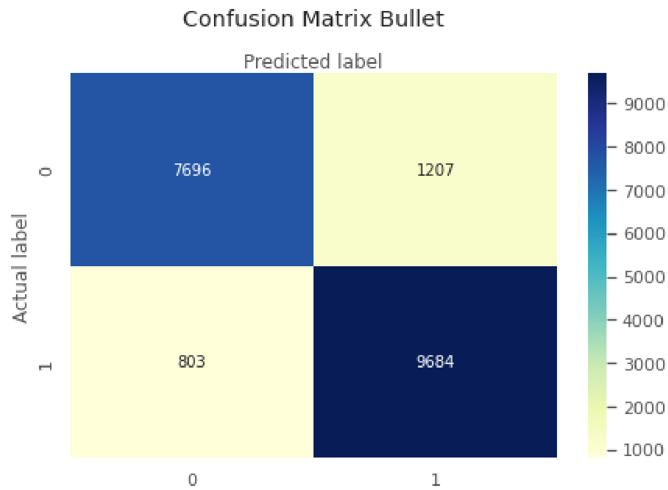

Figure 16. Confusion Matrix for Final Positions in Bullet Games

	precision	recall	f1-score	support
White Lost	0.91	0.86	0.88	8903
White Won	0.89	0.92	0.91	10487
accuracy			0.90	19390
macro avg	0.90	0.89	0.90	19390
weighted avg	0.90	0.90	0.90	19390

Figure 17. Classification Report for Final Positions in Bullet Games

From Figure 16. and Figure 17., we can see that the predictive model has an accuracy of approximately 90% when being tested on the Bullet games in the testing set, similar to the accuracy of the model that tried to predict the result of a chess game from the final position.

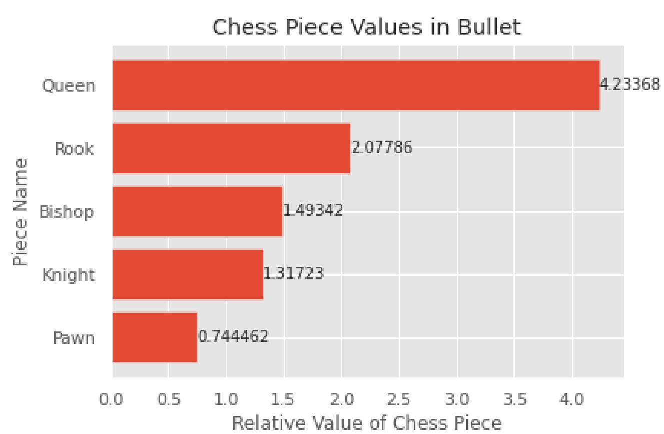

Figure 18. Chess Piece Values for Final Positions in Bullet Games (Unnormalized)

Like we did previously, we determine the coefficients $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$ of each piece imbalance for our features $x_1, x_2, x_3, \cdots, x_n$. These coefficients are [0.74446229, 1.31722762, 1.49341847, 2.07785844, 4.2336753] for each feature ['Pawn'. 'Knight', 'Bishop', 'Rook', 'Queen'] respectively. The LLR p-value for these coefficients was significantly small (<.005).

We again normalize the coefficients to get values that may seem more familiar.

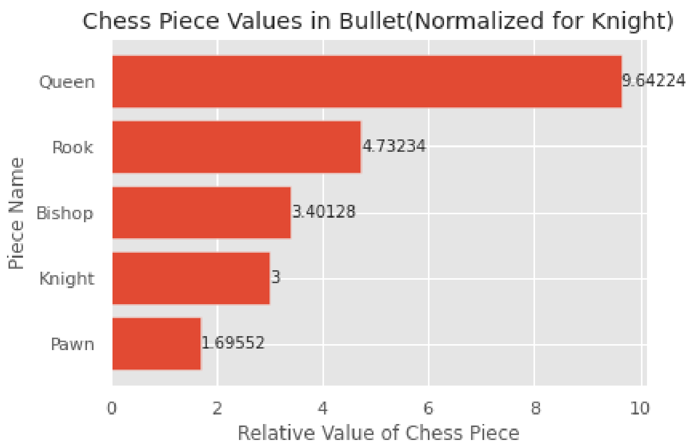

Figure 19. Chess Piece Values for Final Positions in Bullet Games (Normalized for Pawn)

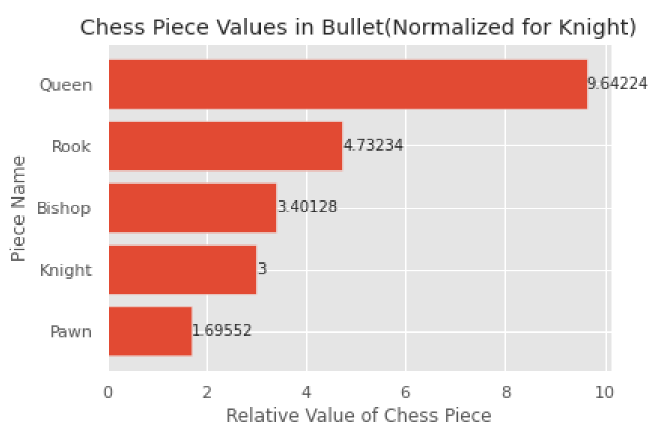

Figure 20. Chess Piece Values for Final Positions in Bullet Games (Normalized for Knight)

Normalizing the value of the Chess pieces in Bullet games for the Knight, we get the following values [1.69552084, 3, 3.40127653, 4.73234481, 9.64224078] for each piece ['Pawn'. 'Knight', 'Bishop', 'Rook', 'Queen'] respectively. These values somewhat resemble the values in Figure 1. Comparing with the values obtained when analyzing the value of the pieces at the end of a game regardless of time control (Figure 14.), we see that the Bishop, Rook, and Queen tend to be more valuable in Bullet Games, compared to other time controls. In fact, the Queen is worth significantly more in Bullet games compared to its value in all games regardless of time control.

Based on these results, we can conjecture that material imbalances, especially of the higher-value pieces, tend to be more consequential in the 'Bullet' time control. Particularly, the Queen's high value may stem from the Queen's ability and reach in a fast-paced chess game, where there is less time for concrete calculation and more scope for tactical play, in which the Queen particularly excels. On the other hand, pawns, which are important at the highest-level games, lose some of their value in a fast-paced chess game.

We next analyze games with the 'Blitz' time control. The blitz time control refers to games that are moderate in speed, with more than 3 minutes, but less than 8 minutes for each side to play their moves[19].

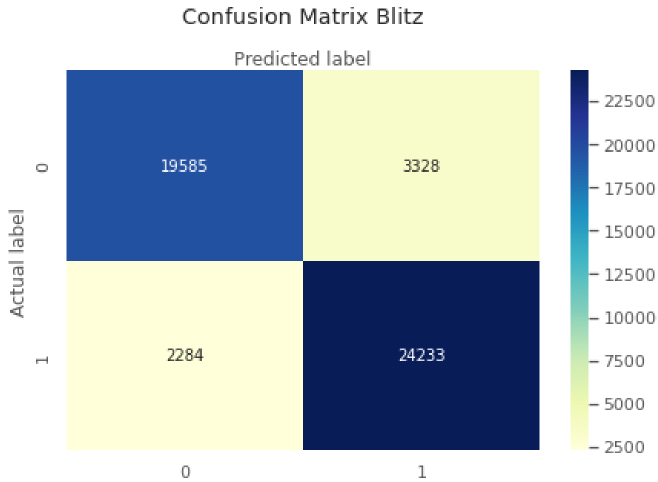

Figure 21. Confusion Matrix Blitz

	precision	recall	f1-score	support
White Lost	0.90	0.85	0.87	22913
White Won	0.88	0.91	0.90	26517
accuracy			0.89	49430
macro avg	0.89	0.88	0.89	49430
weighted avg	0.89	0.89	0.89	49430

Figure 22. Classification Report for Final Positions in Blitz Games

From Figure 21. and Figure 22., we can see that the predictive model has an accuracy of approximately 89% when being tested on the Blitz games in the testing set, similar to the accuracy of the other Logistic Regression models that tried to predict the result of a chess game from the final position.

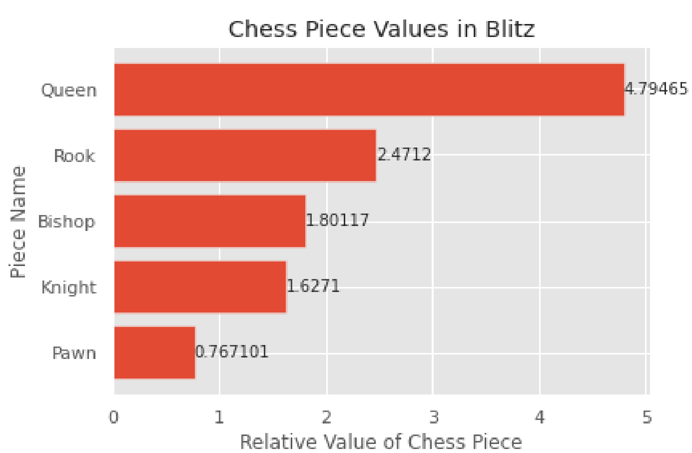

Figure 23. Chess Piece Values for Final Positions in Blitz Games (Unnormalized)

Like we did previously, we determine the coefficients $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$ of each piece imbalance for our features $x_1, x_2, x_3, \cdots, x_n$. These coefficients are [0.76710072, 1.6271021, 1.80116821, 2.47119884, 4.7946502] for each feature [‘Pawn’. ‘Knight’, ‘Bishop’, ‘Rook’, ‘Queen’] respectively. The LLR p-value for these coefficients was significantly small (<.005).

We again normalize the coefficients to get values that may seem more familiar.

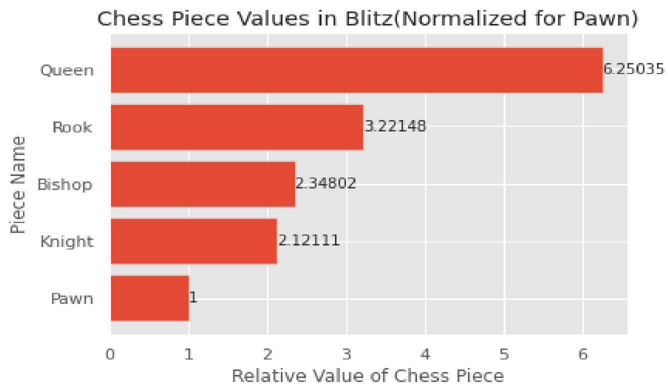

Figure 24. Chess Piece Values for Final Positions in Blitz Games(Normalized for Pawn)

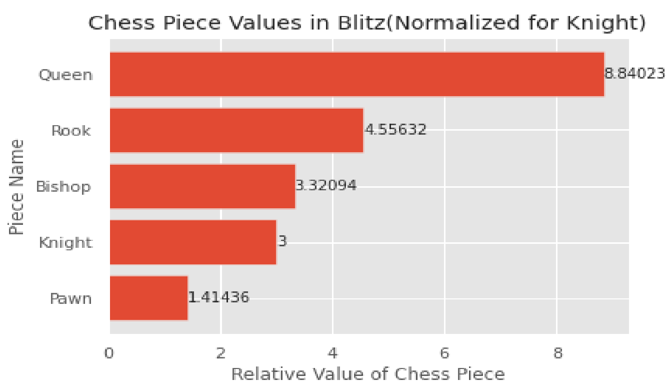

Figure 25. Chess Piece Values for Final Positions in Blitz Games (Normalized for Knight)

Normalizing the value of the Chess pieces in Blitz games for the Knight, we get the following values [1.41435633, 3., 3.32093765, 4.55631919, 8.84022618] for each piece [‘Pawn’. ‘Knight’, ‘Bishop’, ‘Rook’, ‘Queen’] respectively. These values closely resemble the values in Figure 1. We see that the value of the Queen slightly diminishes from the value it has across all time controls, but closer inspection reveals that this happens because the high value of the Queen in Bullet brings up the overall average across the time controls. Additionally, the high similarity between the point values in the Blitz time control and across the values found across all time controls may be caused by the large number of Blitz games in the dataset, which may cause Blitz games to have a disproportionate influence on the general piece values.

We finally analyze the ‘Classical’ time control. The time control refers to games that are slow in speed, with more than 8 minutes for both sides to play their moves [19].

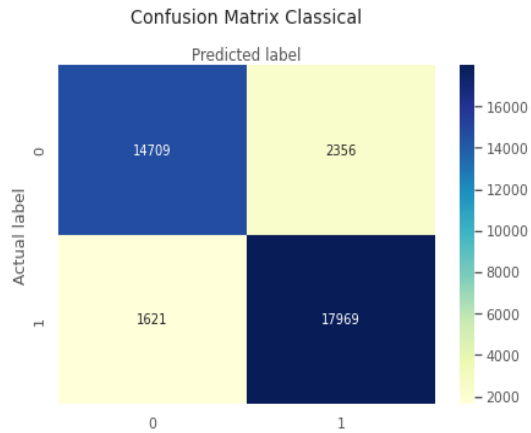

Figure 26. Confusion Matrix for Final Positions in Classical Games

	precision	recall	f1-score	support
White Lost	0.90	0.86	0.88	17065
White Won	0.88	0.92	0.90	19590
accuracy			0.89	36655
macro avg	0.89	0.89	0.89	36655
weighted avg	0.89	0.89	0.89	36655

Figure 27. Classification Report for Final Positions in Classical Games

From Figure 26. and Figure 27., we can see that the predictive model has an accuracy of approximately 89% when being tested on the Classical games in the testing set, similar to the accuracy of the other Logistic Regression models that tried to predict the result of a chess game from the final position.

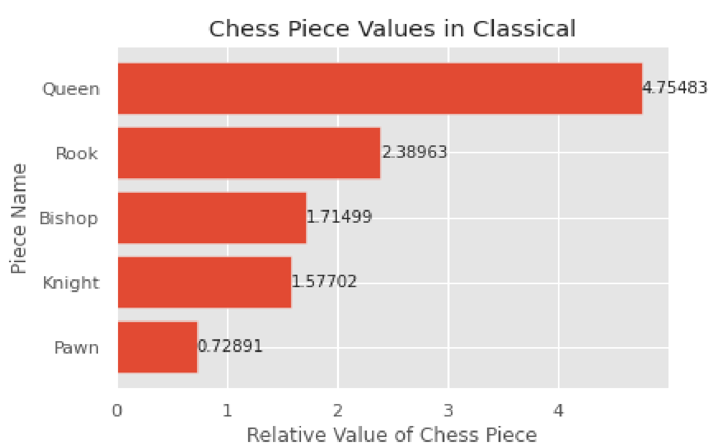

Figure 28. Chess Piece Values for Final Positions in Classical Games (Unnormalized)

Like we did previously, we determine the coefficients $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$ of each piece imbalance for our features $x_1, x_2, x_3, \cdots, x_n$. These coefficients are [0.72890957, 1.57701573, 1.71499235, 2.38963125, 4.75482938] for each feature [‘Pawn’. ‘Knight’, ‘Bishop’, ‘Rook’, ‘Queen’] respectively. The LLR p-value for these coefficients was significantly small (<.005). We can see this in Figure 32.

We once again normalize the coefficients to get values that may seem more familiar.

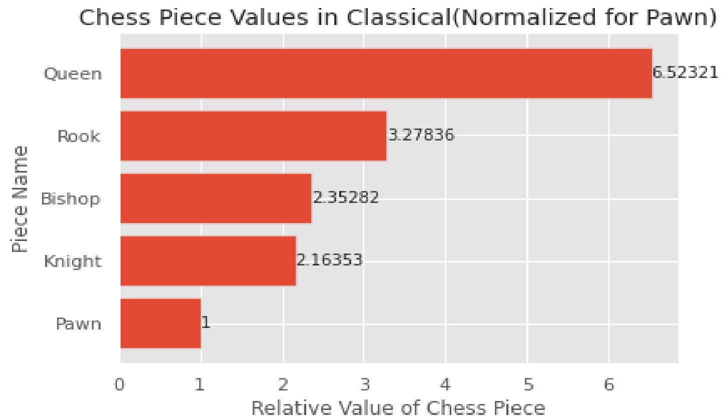

Figure 29. Chess Piece Values for Final Positions in Classical Games (Normalized for Pawn)

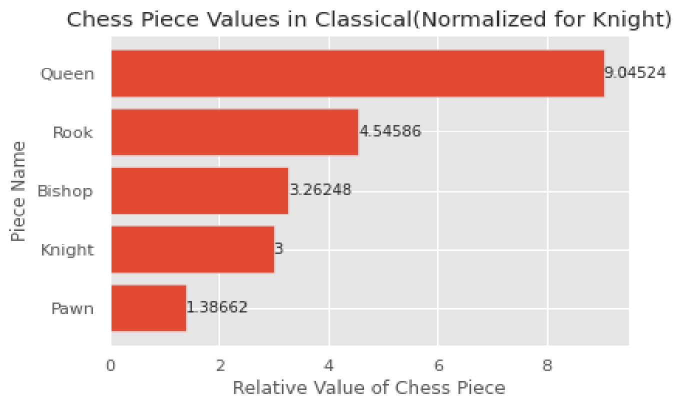

Figure 30. Chess Piece Values for Final Positions in Classical Games (Normalized for Knight)

Normalizing the value of the Chess pieces in Classical games for the Knight, we get the following values [1.38662454, 3., 3.26247669, 4.54586064, 9.0452415] for each piece ['Pawn'. 'Knight', 'Bishop', 'Rook', 'Queen'] respectively. Rounding the values for the Knight, Bishop, Rook, and Queen, we get the values [3.0, 3.25, 4.5, 9.0]. These values are suggested by Artur Yusupov in his Build up your chess with Artur Yusupov: The fundamentals[18]. Perhaps surprising, our analysis values the pawn more than the analysis done by Yusupov. One reason for this may be that, in practical games between relatively strong players, small imbalances like the pawn may be enough to decide the outcome of the chess game. Our results in conjunction with the expert analysis done by Yusupov suggest that the piece values in Table 4. may have merit to them.

Conclusion

In this paper, we used the statistical technique of Logistic Regression to predict the results of a chess game based on piece imbalances. We use the coefficients of such a model to determine the relative values of chess pieces. We vary our analysis on different time controls and positions of a chess game to analyze how the values of chess pieces change.

Our results indicate that predicting the value of chess pieces from the final position using these techniques leads to accurate models with values close, yet not exactly the same to the ones in Figure 1. When we normalize these relatives such that the Knight is equivalent to 3 points, we get the following piece values.

Table 4. Piece Value Results for All Positions and Time Controls

Position and Time Control	Piece Values: [Pawn, Knight, Bishop, Rook, Queen]
Final Position - All Time Controls	[1.43354377, 3., 3.30016909, 4.55956838, 8.9150605]
Halfway Position - All Time Controls	[1.35690059, 3, 3.70400444, 4.856291, 8.39651183]
Final Position - Bullet Time Control	[1.69552084, 3, 3.40127653, 4.73234481, 9.64224078]
Final Position - Blitz Time Control	[1.41435633, 3., 3.32093765, 4.55631919, 8.84022618]
Final Position - Classical Time Control	[1.38662454, 3., 3.26247669, 4.54586064, 9.0452415]

Our research indicates that attempting to predict the results of a chess game by the piece imbalances that exist halfway through a game using Logistic Regression leads to a model that is not accurate. Thus, the coefficients obtained from that model may not be of particular importance, and further research may need to be conducted to analyze this discrepancy. We conjecture that this happens because unlike piece imbalances that exist in the final position of a game, it is more likely that piece imbalances that exist only halfway through a game offer the opponent some other form of advantage. This would align with the fact that half of the chess game is still yet to be played. Additionally, further research may need to be conducted to see how the value of the pawn itself changes throughout a game and depending on its position on the board. Further research may consider how the piece values change based on various positions and stages for a chess game. Another field of exploration would be to consider how the values of chess pieces change as the average ratings of the opponents change, or when there are large differences between the ratings of the opponents in a chess game.

Our research helps provide answers to the questions asked in Section I: What are the practical values of Chess Pieces based on what happens in real games? How do the Chess Piece values change when applying statistical methods to various stages of a chess game (are certain pieces more valuable at the end of a chess game, compared to the middle)? How do the Chess Piece values change in various time controls (i.e Bullet, Blitz, Classical)? Such values and research may help chess players better calculate and assess a position in the game of chess.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

References

[1] Capablanca, Jose; de Firmian, Nick (2006), Chess Fundamentals (Completely Revised and Updated for the 21st century), Random House, ISBN 0-8129-3681-7

[2] Seirawan, Yasser; Silman, Jeremy (1990), Play Winning Chess, Microsoft Press, ISBN 1-55615-271-X

[3] Soltis, Andy (2004), Rethinking the Chess Pieces, Batsford, ISBN 0-7134-8904-9

[4] Silman, Jeremy (1998), The Complete Book of Chess Strategy:Grandmaster Techniques from A to Z, Siles Press,
ISBN 978-1-890085-01-8

[5] Polgar, Susan; Truong, Paul (2005), A World Champion's Guide to Chess, Random House, ISBN 978-0-8129-3653-7

[6] Good, I. J. (1968); A Five-Year Plan for Automatic Chess

[7] Lichess Database. Available online: https://database.lichess.org/

[8] Betza, Ralph; "Ideal and Practical Values". The Chess Variant Pages.

[9] Ubdip (2022); "Finding the Value of Pieces"; Lichess Blog.

[10] Ubdip (2022); "Comments on piece values"; Lichess Blog.

[11] Bååth, Rasmus; "Big Data and Chess: What are the Predictive Point Values of Chess Pieces?". Sumsar

[12] Kaufman, Larry (March 1999), "The Evaluation of Material Imbalances", Chess Life, archived from the original on 2006-06-29, retrieved 2022-11-23

[13] "Forsyth-Edwards Notation (FEN)", Chess.com, https://www.chess.com/terms/fen-chess

[14] Python Chess. Available online: https://python-chess.readthedocs.io/en/latest/#

[15] Wikipedia contributors. "Logistic Regression." Wikipedia, 30 Oct. 2022, en.wikipedia.org/wiki/Logistic_regression.

[16] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[17] Giovanni Di Luca, "Knight Vs Bishop: Which Chess Piece Is More Valuable?", ChessPulse.com

[18] Yusupov Artur. (2008). *Build up your chess with Artur Yusupov: The fundamentals*. Quality Chess.

[19] "Time Controls in Chess - Chess Terms." *Chess.com*, https://www.chess.com/terms/chess-time-controls.