Checkmate by Algorithm: Analyzing Online Chess Games through the Lens of Machine Learning

Thanya Nguyen
The University of Tennessee
Computer Science
Knoxville, United States
Thatngu1@vols.utk.edu

Mourtalla Diop
The University of Tennessee
Computer Science
Knoxville, United States
mdiop@vols.utk.edu

***Abstract*— Chess, an esteemed strategic board game, is played globally in both digital and face-to-face formats. The comprehensive "Chess Game Dataset (Lichess)" hosted on Kaggle encompasses more than 20,000 recorded chess matches, providing intricate details such as player ratings, winning parties, initial strategies, and time increments. This analytical project leverages the dataset to conduct an in-depth examination of the Vienna Opening. Utilizing advanced machine learning algorithms, it aims to accurately predict game outcomes, delineate the characteristics of winning strategies, and implement linear regression analysis to establish a correlation between time controls and the number of moves executed.**

I. INTRODUCTION AND MOTIVATION

This is not complete.

A. [What is the dataset or datasets you chose for your project, and why did you choose it/them?]

While conceptualizing our project, our longstanding friendship played a significant role, especially as we explored our shared interests in marathon running, coding, and video gaming. We made a collective decision to delve deeper into researching chess, given our active involvement in the Chess Club at the University of Tennessee, Knoxville (UTK). In this club, Thanya holds the position of president, and Mourtalla has been an engaged member for two years. As integral members of the UTK Chess Club, our shared passion and understanding of chess is evident. Initially, we contemplated focusing our project on marathon running, an activity we both enjoy and plan on working towards together in the future.

The data set we have chosen for this project is called “Chess Game Dataset (LiChess)” from Kaggle’s Dataset website. We would like to further our understanding of chess openings to potentially sharpen our performance skills, get a better understanding of stronger algorithms, and analyze other people’s performance to learn from their blunders or brilliances.

B. [What is the overarching goal of you’re trying to achieve with ML on this dataset?]

The overarching goal we are trying to achieve on this dataset is getting a better understanding of player’s performance and tendencies while playing online. We hope that we can reveal further information on the “stronger” openings being played, what time controls have better odds of winning or any information that could give us an upper hand when playing in the future. Any new information gained on the trends and tendencies found in the dataset could benefit our understanding and future performance.

The machine learning approaches we are using are validation accuracy, training set, testing set, heatmaps, and linear regression. Heatmaps were used to display validation accuracy though different iterations and C values. Heatmaps were also chosen because they were one of the more recent topics covered in the Introduction to Machine Learning course, compared to Decision Trees, which were an initial idea to pursue but was not due to too many paths possible and the fear of a convoluted decision tree.

C. [What is the dataset or datasets you chose for your project, and why did you choose it/them?]

My partner and I, who are both avid chess players and my partner is the club president, selected the chess dataset for our project. This dataset is in line with our individual interests and offers a chance to learn more about the variables affecting the results of chess games. We chose to use chess data over other choices, such a running dataset, because it does not only align with our interests but also provides a distinctive educational opportunity in the field of strategic board games.

D. [What is the overarching goal of you’re trying to achieve with ML on this dataset?]

The main objective of using machine learning on the chess dataset is to find trends and insights that help improve our comprehension of the variables affecting game results. Using machine learning techniques, our goal is to find important characteristics and connections in the data that influence a player's ranking. This investigation extends beyond the conventional chess viewpoint and offers a way to glean important data that may enhance strategic play and add to the body of knowledge within the chess community. The project's goal is to improve our understanding of chess dynamics by using machine learning as a tool to decode the game's intricacies.

II. DATASET

In Kaggle’s Chess Game Dataset, it holds 16 types of data, but the ones that we are using in this research project are: [turns, victory_status, winner, increment, white_rating, black_rating, moves, opening_eco, and opening_name].

This dataset comprises more than 20,000 online games, detailing the victors, and the specific openings used in each game, all of which are crucial for understanding the behaviors of online chess players. It contains patterns that are valuable for machine learning algorithms to discern new trends, as well as to examine the correlation between various openings and the likelihood of winning for both white and black players.

The first thing we wanted to visualize was the wins as black vs. white and how they depend for each rating. For example, is black more likely to win in the 1000 rating compared to the 2600 rating? Is there an advatage in color we see in various rating brackets. The first step was to divide the ratings into bins that were 200 point increments. For example, bin 0 would be from 0-200, bin 1 would be 200-400, so on and so forth. After seperating all the players into their specific bin, we then counted every time a player won playing as black or white and placed them into Figure 1.

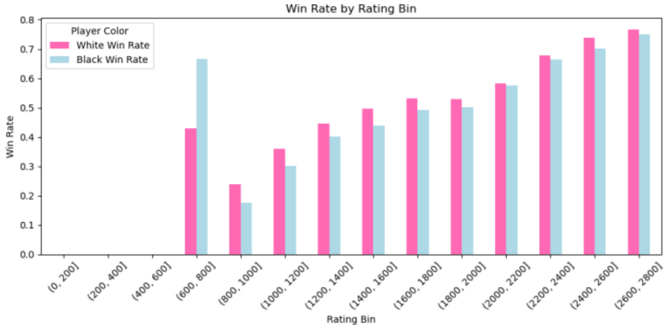

Figure 1: Bar chart of win rate as black vs. white separated by bins.

Looking at Figure 1, it shows the most inconsistency at the lower rating from ratings 600-800 with a higher win rate as black than white. The dataset does not have players rated less than 600 and does not have many total wins in groups 800-1000 than 600-800. As you proceed on the x-axis, the win rate as white is gradually slightly higher than black’s win rate and the total number of wins recorded also increases as well. Bin 2600-2800 holds the highest frequency of games recorded.

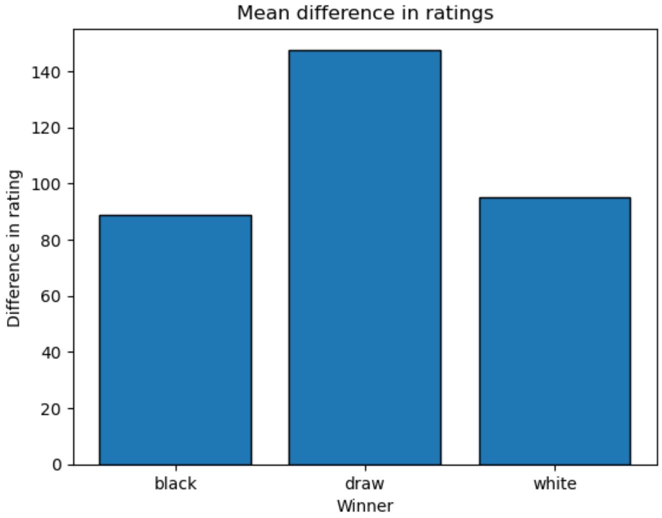

Figure 2: Bar graph of Mean Difference in Ratings based on Winner.

The mean rating difference between chess game victors and losers is depicted in the histogram. The average rating difference for games won by either black or white is shown by each bar. In the 60-80 rating range, the histogram shows that white players have a higher win percentage than black players. It's crucial to remember that the dataset is missing players with ratings lower than 600, which could have an impact on the pattern that has been seen. There is a progressive change in favor of a marginally greater win rate for white players down the x-axis. There is a concentration of games in the higher rating brackets, as seen by the highest frequency of games recorded in the rating bin of 80–100. Based on the investigation, it appears that different rating ranges have different winning dynamics like black or white. The trends in the 80–100 rating range that have been seen may be affected by the absence of lower-rated players in the sample. The highest frequency observed in the 80–100 range may indicate a concentration of challenging games.

III. MACHINE LEARNING APPROACHES AND METHODOLOGY

A. *Are you comparing two different algorithms?*

In our machine learning method appriach, the first approach was to use a Support Vector Machine, very similar to a previous lab done in this course. The idea was to sift through the data for simplicity. We are more interested in the Vienna Opening (with any variation). We extracted the data entries who have openings played in the Vienna, called Vienna dataframe, which were 154 entries and was able to use a C and a Max_iter to find the highest accuracy score. We split the Vienna dataframe into a training and testing groups and computed the accuracy score into a heatmap.

Before being able to do that, we had to convert the columns, “Victory status” and “winner” to numerical values. For victory status, we converted mate to 1, resign to 2, out of time to 2, and

Identify applicable funding agency here. If none, delete this text box.

draw to 3. For our “winner” column we converted white to 1, black to 2, and draw to 3. For the “time increment” column, grouped the times to the bigger categories: bullet, blitz, rapid, and classical. We also truncated the incremental seconds in the data for simplicity. For example, “5+3” which represents a 5-minute game with 3 second increment per turn will be simplified as “5”, or simply a 5-minute game. In all the data entries, we also calculated the “rating difference” between the player as white and black, this would be able to show how important the difference in rating is when playing. A positive rating difference would mean that white is a higher rating than black, and a negative rating difference signified that black is higher rated than white.

B. *Are you comparing performance across hyperparameter values?*

We computed three heatmaps, which tried to predict different targets with different hyperparameters.

Heatmap 1 tried to predict the winner color (white, black, or draw) given the features: white’s rating, black’s rating, number of turns, victory status (checkmate, resignation, etc.), time control, and rating difference. Heatmap 2 tried to predict the winner color, like in Heatmap 1, but was given less hyperparameters this time. White’s rating and Black’s rating was not given, but the rating difference was still a given feature. The features used in Heatmap 2 are: number of turns, victory status, time control, and rating difference. Lastly, Heatmap 3 tried to predict the victory status (checkmate, resignation, draw, etc.) given the features: number of turns, winner color, time control, and rating difference.

C. *How are you defining whether the project is successful?*

The way we determined if this project was successful is if we are given a diverse heatmap of many different shades and colors.

D. *Some Common Mistakes*

Originally, the idea was to utilize decision trees, but more success was found with using support vector machines because Thanya was too scared to use decision trees because of how many different approaches that could’ve been taken for decision trees. Too much free will with too many different hyperparameters to consider. Not considering some of the hyperparameters felt like it could’ve skewed the data to how accurate it would be.

For determining the heatmaps, encoding the winner and victory status may have skewed the heatmaps because I have chosen “white” to be signified as 1, “black” as 2, and “draw” as 0. Before this assignment, I had “draw” signified as 3 instead of 0 because I felt that my heatmaps did not display a wide variety of colors desired. This tweak changed the heatmap from having many empty squares to less empty squares.

Are you comparing two different algorithms?

We were mostly concerned with linear regression, a particular approach that is employed to predict a continuous target variable by considering one or more independent factors. It shows how to design and evaluate a linear regression model for example predicting 'white_rating' based on the 'winner_encoded' in figure 6, characteristic, rather than comparing two distinct procedures.

Are you comparing performance across hyperparameter values?

No, a performance comparison among hyperparameter settings is not included. For both the training and testing datasets, it constructs a linear regression model and assesses its performance using mean squared error (MSE) and R-squared. It does not, however, conduct a methodical search or comparison of various hyperparameter values.

E. *How are you defining whether the project is successful?*

The capacity of the linear regression to correctly forecast this project's features—such as white player ratings based on the winning encoding feature, for example—will decide its success. Metrics like Mean Squared Error (MSE) and R-squared (R2) scores, which show the model's predictive ability on both the training and testing datasets, are the main tools used to evaluate success.

F. *Some Common Mistakes*

It is important to stress the adherence to important assumptions and potential problems in the linear regression when working on the project. Fundamental to this whole process is the assumption that independent and dependent variables have a linear relationship. Unreliable predictions could result from the model's inability to correctly identify the underlying patterns in the data if this premise is disregarded or broken.

IV. RESULTS

Figure 1 shows the heatmap of the validation accuracy with the Max_iter and C values. The heatmap is mostly dark but has multiple repeating values. The first two rows in the heatmap are the same values of 0.75 and 0.69. The highest validation accuracy in this heatmap is 0.75 which is displayed 11 times in Figure 1. While the lowest validation accuracy is 0.4375. A potential reason for the multiple values may be because black’s rating and white’s rating are included in the training set. The most interesting features in Figure 1 are when Max_Iter is 1000 and 10000 because they all have the same values. The average validation accuracy for Figure 1 is 0.6822.

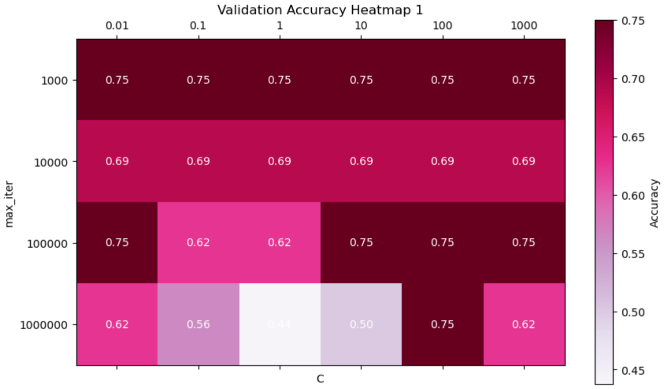

Figure 1: Validation Accuracy Heatmap 1 – Predicting the winner color

Figure 2 shows a more diverse heatmap with different values and intensities. The highest validation accuracy was depicted in three instances with the value 0.88. These values were located at (100,1000), (1000,1000), and (1000,100000). The most repeated value is still 0.75 at 6 different instances. Whereas the lowest validation accuracy is shown twice with the values 0.375. The validation accuracies whose Max_Iter is 10000 is most interesting because there are not repeated values. A potential reason that caused the diversity in values may be from excluding the ratings of the players in the training set. This heatmap is the most visually appealing and is very diverse as intended. This model is deemed successful because of its diversity in values and color. The average validation accuracy for this heatmap is 0.7188.

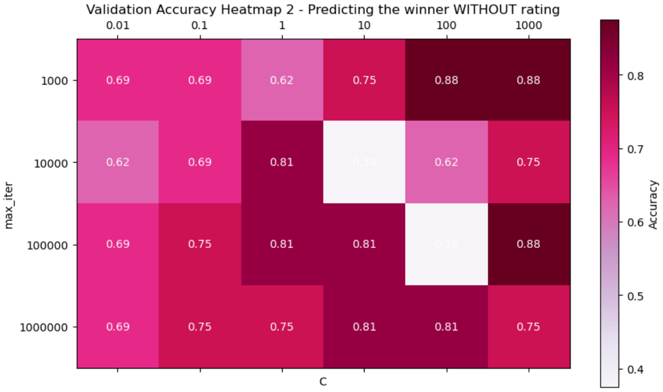

Figure 2: Validation Accuracy Heatmap 2 – Predicting the winner color without black's rating and white's rating.

Lastly, Figure 3 shows a heatmap of the validation accuracies when trying to predict the state of the victory instead of the winner color while still excluding the white's rating and black's rating. Figure 3 shows smaller value numbers overall compared to Heatmap 1 and 2. The highest validation accuracy in Figure 1 appears twice with the value 0.88 while the lowest value appears three times at the value 0.375. Interestingly, the lowest values all share the same Max_Iter of 1000 but different C values. Due to the low values in the heatmap, this machine learning model was probably unable to predict the victory status as well as it should. The average for Figure 3 is 0.6484.

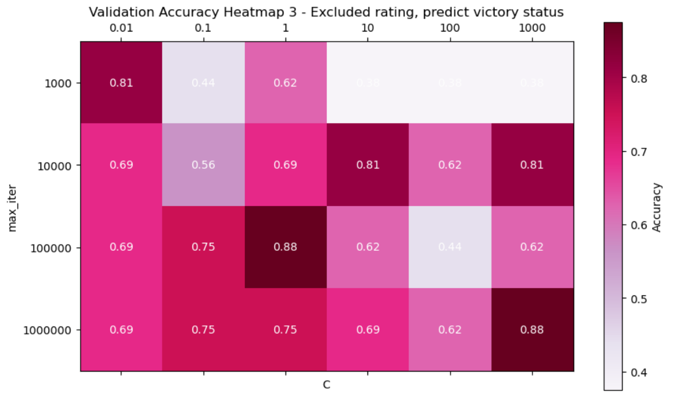

Figure 3: Validation Accuracy Heatmap 3 – Predicting the state of victory without the black's rating and white's rating.

Figure 4, the link between the number of turns in chess games and the accompanying player ratings is visually represented by the scatter plot. The model's forecast based on this link is shown by the linear regression line (in green). Higher player ratings are generally correlated with more turns, as the scatter figure illustrates. The general linear relationship in the data is captured by the linear regression line. We can see how effectively the model generalizes to new data by looking at the scatter plot, which overlays the training and testing data points. The model predicts strong generalization if it performs well on both training and testing data. Inconsistencies between testing and training data could be a sign of either underfitting or overfitting. The positive coefficient suggests that better player ratings are generally correlated with more turns. The intercept gives the expected grade when there are no turns; however, this may not make sense in this situation.

We also have;
Coefficient: 1.1409071265183761
Intercept: 1526.4751605614613
Train MSE: 83763.49251027362
Train R2: 0.01721090379824386
Test MSE: 81935.46249376345
Test R2: 0.014828814895568976

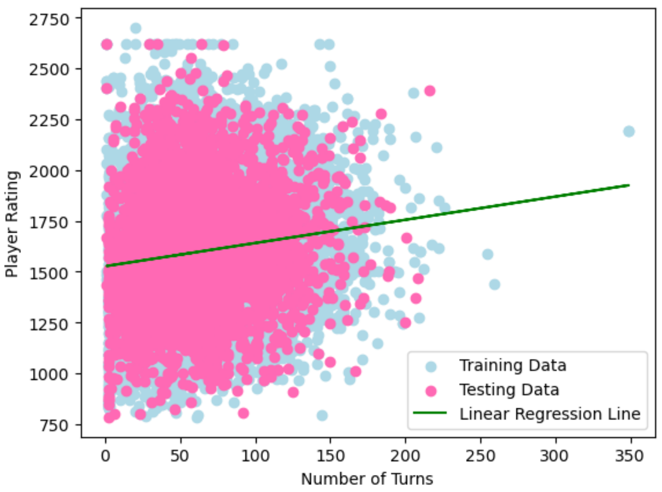

Figure 4: linear regression between number of turns of chess games and player rating.

Figure 5, the scatter plot shows how the number of turns in a chess game correlates with the time increment. The model's forecast based on this link is shown by the linear regression line (in green). The scatter plot indicates a possible pattern in which time intervals correspond to numbers of turns. The general linear relationship that the model predicts is represented by the linear regression line. The coefficients of the linear regression model shed light on how variations in the time increment affect the estimated number of turns. This tells us how the expected number of turns changes with each unit increase in the time interval. The model's performance is assessed using Mean Squared Error (MSE) and R-squared (R2) on both the training and testing datasets. We can see how effectively the model generalizes to new data by looking at the scatter plot, which overlays the training and testing data points. Both training and testing data should show strong performance from a solid model, indicating strong generalization. Inconsistencies between testing and training data could be a sign of either underfitting or overfitting. The coefficient shows how many turns change on average for every unit increase in the time increment. Though it might not have a useful meaning in this situation, the intercept gives the expected number of turns when the time increment is zero.

We also have.

Coefficient: -0.12442874490155253, Intercept: 62.118769957431915

Train MSE: 1122.3366226457367, Train R2: 0.004077381616892528

Test MSE: 1125.6071187317684, Test R2: 0.0010663829533220648

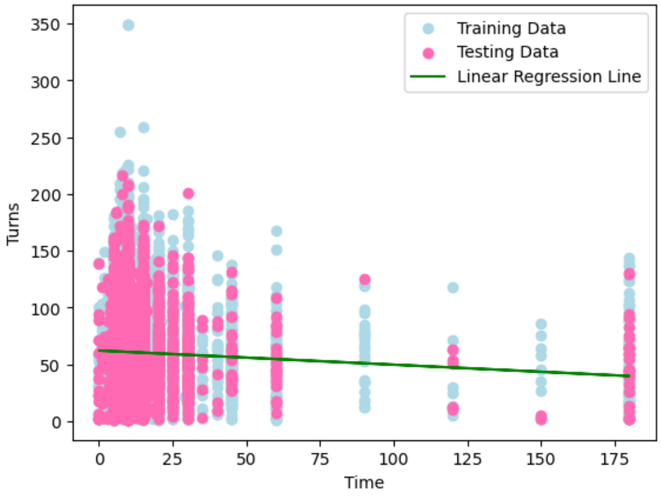

Figure 5: linear regression between formatted time increment and the number of turns in chess games.

Figure 6,the association between the appropriate white player ratings and the encoded winning variable ('white': 1, 'black': 2, and 'draw': 3) is visualized using a scatter plot. Based on this association, the model's prediction is represented by the linear regression line (in green). The scatter plot displays the distribution of white player ratings according to the winner encoding. The general linear relationship that the model predicts is represented by the linear regression line. The intercept gives the predicted white player rating when the winner encoding is zero, though this may not have a useful interpretation in this situation. The coefficient shows the average change in the predicted white player rating associated with each unit increase in the winner encoding.

We also have;

Coefficient: -45.371563797470685, Intercept: 1665.6527779545395

Train MSE: 84526.1629277854, Train R2: 0.008262564278681994

Test MSE: 82319.65616748798, Test R2: 0.010209367768122424

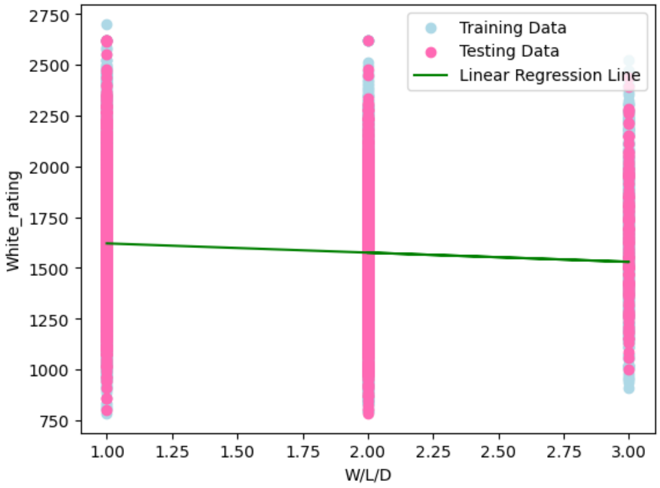

Figure 6: linear regression of White Player Rating vs Winner Encoding

.

V. DISCUSSION, CONCLUSION, AND FUTURE WORK

After looking at the results, there is no correlation on which color, black or white, would win more. The machine learning model was able to predict the winner color the best when the rating was not provided as a feature. The machine learning model had a harder time trying to predict the state of the victory without black and white's ratings given. The machine learning model was able to predict the winner color given each player's rating. From the 3 heatmaps, the heatmap with the highest validation accuracy is the one that Heatmap 2, the one that predicted the winner color without black’s rating and white’s rating as a feature.

If there was more time, I would try to make a decision tree based off the players moves in the Vienna Opening. This may take a long time and the decision tree might be very convoluted, but it would be interesting to see what the tree would look like and what are common moves or common mistakes players make. I would also be interested to see how many inaccuracies player’s make when the games have a fast time control compared to a slower time control. A good idea that could be pursued is creating an 8x8 heatmap that represents each square on the chess board and seeing where a certain piece, like a knight, is placed most on the board. This would be interesting to see the Vienna opening specifically, so I am able to see which spots on the board have the most attention and traction.

This project opened some new research questions like what would happen to the validation accuracies scores if the Max_Iter values or the C values were different. What would the heatmap look like for different openings such as the Sicilian Defense? Queen’s Gambit? Will the winners be easily predicted for other openings/defenses like how it was for the Vienna Opening? Is the Vienna Opening the reason why machine learning was able to predict so highly? Why was the player’s rating given effecting the machine learning’s predictions to the winner? Was the player’s rating given perhaps overfitting and incapable of making accurate predictions? Can these findings help players determine if the Vienna Opening is a sturdy opening to learn?

The other project idea I would’ve liked to pursue that I didn’t get the chance to do is something running related. There were marathon running datasets I could’ve used that would’ve been interesting to see the trends and to predict things. Another hobby I am also interested in is solving Rubik’s Cubes. Applying machine learning to analyze record. solve times or seeing the most efficient way of solving the cube would be very interesting. Overall, I (Thanya) am very happy that I had the opportunity to use chess as the subject for our project. Given another chance, I'd still choose chess, but with expanded ideas. Chess, with its vast scope, offers numerous opportunities for applying machine learning in various strategies. Pursuing a more advanced approach to this project would be what I am most interested in.

Looking at the results for the linear regression, our main objective was to investigate the association between time increment, the number of turns in chess games, white player ratings, and encoded game outcomes (win/loss/draw) in the linear regression analysis we performed on the chess dataset. Our familiarity with the game and its strategic character led us to select the chess dataset. Our prediction algorithm was able to identify patterns that affect white player ratings thanks to the encoded game outcomes, which gave our model a numerical representation.

We discovered some intriguing findings after preprocessing the data and training the linear regression model. The model's coefficients allowed us to measure the effect of time increment and turn count in addition to highlighting the effect of game results on white player ratings. This thorough approach gave rise to a more sophisticated comprehension of the variables influencing player ratings. Regression lines were added to the scatter plots to further illustrate the model's performance on the training and testing datasets. The model's accuracy and capacity for generalization were quantitatively evaluated by the mean squared error (MSE) and R-squared (R2) values. It's important to highlight that although the model offered insightful information, it also ran into issues with the assumption of linear correlations and the possible impact of outliers.

Our linear regression research clarified the complex interactions that exist in chess between player ratings, time increment, number of turns, and game outcomes. The findings add to our knowledge of the intricate dynamics present in chess games and suggest directions for future research, such as examining the effects of time increments or nonlinear relationships. It is important to recognize the analysis's limitations, including its assumptions and possible sources of inaccuracy, to improve the model going forward.

To sum up, the linear regression analysis shed light on the correlation between player ratings and game outcomes. Chess enthusiasts can understand the quantitative effects of various game outcomes on player performance thanks to the interpretability of the model. Future research may examine more complex models, consider new features, and apply cutting-edge methods to improve prediction accuracy. To further improve the model's performance, a deeper exploration of feature engineering and domain-specific insights is recommended.

For future work, investigating nonlinear correlations within the dataset may reveal hidden patterns for future research that linear regression could miss. To handle nonlinearity and enhance predictive accuracy, more sophisticated machine learning methods like logistic regression or even more intricate models like decision trees or neural networks could be used. The study concentrated on the ratings of white players, but a more thorough investigation might look at the dynamics of black player ratings as well. A more thorough knowledge of the chess rating system can result from examining the asymmetry in player ratings based on game outcomes and other factors. Finally, taking into account the temporal aspect of the dataset through trend analysis across time or examining the effects of changing playing styles and strategies could offer a dynamic viewpoint on the state of chess.

VI. CONTRIBUTION OF TEAM MEMBERS

A. *Thanya Nguyen*

Thanya Nguyen worked on the bar chart with the win rate by rating bins, made the Hello Kitty PowerPoint, the Validation accuracy heatmaps, and wrote her part in the paper.

B. *Mourtalla Diop*

Mourtalla Diop worked on the bar chart with the winners vs. difference in rating, the three linear regression models, the paper, and uploading the zoom video.

VII. REFERENCES

Kaggle data set “Chess Game Dataset (Lichess)” - https://www.kaggle.com/datasets/datasnaek/chess/data