WORCESTER POLYTECHNIC INSTITUTE

Final Report

Leveraging Chess Game Data for Player Performance Prediction

DS 502 – Statistical Methods for Data Science

Presented By:

Sheroz Shaikh & Steffi Dorothy

Presented To:

Prof. Fatemeh Emdad

05/01/2024

TABLE OF CONTENTS

Team Members:

Sheroz Shaikh - Skilled in Data Preprocessing & Cleaning, and Machine Learning Model Building. Experienced in handling data preprocessing tasks such as data cleaning, data transformation, and feature engineering and developing and implementing machine learning models including regression, classification, and clustering algorithms.

Steffi Dorothy - Skilled in performing statistical analyses, hypothesis testing, and interpreting results to derive actionable insights from data as well as creating insightful visualizations and effectively communicating findings to technical and non-technical stakeholders.

Background

AI (Artificial Intelligence) has had a significant impact on chess playing and strategy since the release of some of the top chess engines. Virtually all top Grandmasters, who have International Chess Federation (FIDE) ratings above 2700, use AI tools extensively to analyze their games and consequently play moves from chess engine recommendations. AI is fundamental in chess engines because it allows the computer to evaluate positions and make informed decisions based on its analysis. Chess engines use various algorithms and techniques, including machine learning, to evaluate positions and carry out the next move. They analyze vast amounts of data to produce very solid and accurate position choices, allowing them to play much faster than a human could. This makes it a valuable tool for chess players who want to study and improve their play. Moreover, chess engines can work at professional levels, making them challenging opponents for even the strongest human players.

AlphaZero is one of the best examples of a chess engine developed by DeepMind, an AI research company acquired by Google. It became famous for its quick mastery of the game after only a few hours of self-play training. Machine learning is the idea that a machine can understand and interpret data, while artificial intelligence, under the best circumstances, can decide what data points are important and which are not, and determine the best action from that evaluation. The fundamental principle established by computer scientists when they sought to create a machine that could play chess like a human was to make it understand the context and quality of the data.

Earlier, programmers fed a machine learning algorithm with vast chess match data, hoping it would play like a grandmaster. However, the algorithm consistently sacrificed its queen early, as it observed this tactic in many games where players won after such a sacrifice. Unfortunately, it failed to grasp the strategic context and indiscriminately adopted the "lose the queen, win the game" approach. This instance underscores the crucial need for algorithms to not only receive extensive data but also understand its nuanced application, highlighting the importance of context and quality in training data.

There are no less than 10^{111} positions in chess. There are 318,000,000,000 ways for the first four moves of a chess game to go down. There is not a single human on the planet who can account for every single possibility in chess, but AI is getting close.

Description of the problem

Our project aims to analyze a large dataset of chess games obtained from the Lichess online chess platform to predict player performance by analyzing strategic patterns and understanding factors influencing game outcomes. The target problem revolves around developing a predictive model that can accurately forecast the likelihood of a player winning or losing a chess game based on various game attributes and player characteristics. We seek to uncover strategic insights by identifying recurring patterns and tactics employed by players.

Description of the dataset

We will primarily utilize the "Chess Game Dataset (Lichess)" available on Kaggle, which contains a vast collection of chess games played on the Lichess platform. This data set provides comprehensive information about each game, including player ratings, game outcomes, moves played, time controls, and more. The dataset contains around 20K samples of games collected from a selection of users on the site Lichess.org with 16 features. The dataset comprises a collection of chess games, where each row corresponds to a specific game, and each column represents a distinct attribute of the game. It includes various features such as game identifiers, timestamps for when the game was created and last moved, the total number of moves made, the outcome of the game (whether it ended in a checkmate, resignation, timeout, or draw), the usernames and Elo ratings of the players, the sequence of moves made during the game, and information about the opening used, including the ECO (Encyclopedia of Chess Openings) code, name, and number of moves played in the opening phase. This comprehensive dataset allows for detailed analysis of chess games, including player performance, game outcomes, and opening strategies, facilitating insights into the dynamics of chess gameplay. Lichess

provides a vast database of over 12 million games that users can effortlessly search through. This database comprises more than 390 million unique positions. In addition to that, it also includes a curated collection of two million master games from over-the-board tournaments since 1952. This dataset is highly credible as it includes metadata, such as subtitles, tags, overview descriptions, and cover images. It also uses appropriate file formats and has file descriptions. Moreover, it is licensed by CC0 1.0 Universal Public Domain Dedication, and it has a public kernel and task. The combined usability score for all these features is 8.2.

We plan to apply a combination of descriptive statistics, exploratory data analysis (EDA), and machine learning techniques to achieve our project goals. Our initial steps will involve data preprocessing to handle missing values, and outliers, and feature engineering to create new variables that capture meaningful aspects of the game data. We will conduct an in-depth exploratory analysis to uncover trends, distributions, and correlations within the dataset. This will involve visualizations to gain a comprehensive understanding of the data. Additionally, we will employ advanced techniques to identify strategic patterns, opening moves, and tactical maneuvers employed by players across different skill levels. This will involve clustering analysis, sequence mining, and association rule mining to uncover hidden insights.

Motivation

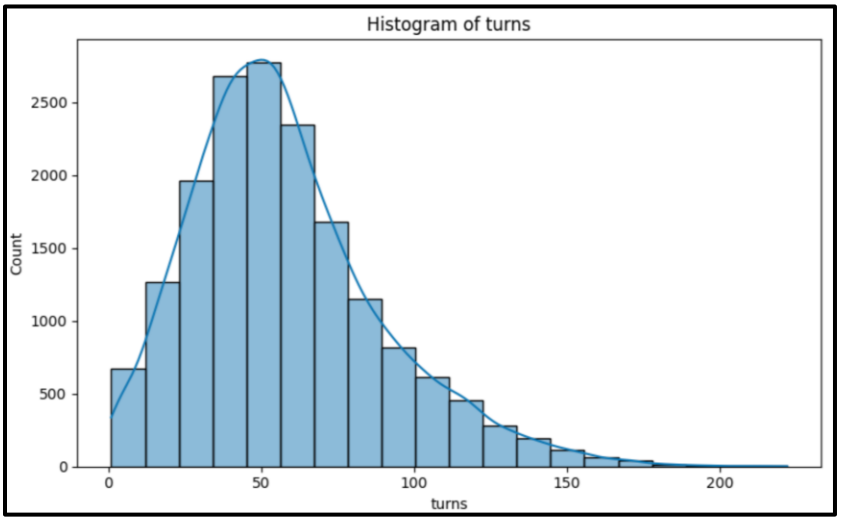
Our team was motivated to explore data analysis applications in understanding intricate systems, such as chess. We recognized that by analyzing substantial amounts of data, we could gain insights into the strategies and patterns that underlie successful gameplay. We believed that this data-driven approach could provide a new level of understanding of the complexities of the game, allowing us to make more informed decisions and improve our gameplay. With this goal in mind, we set out to gather and analyze a comprehensive dataset of chess matches, hoping to uncover hidden insights that could take our understanding of the game to the next level.

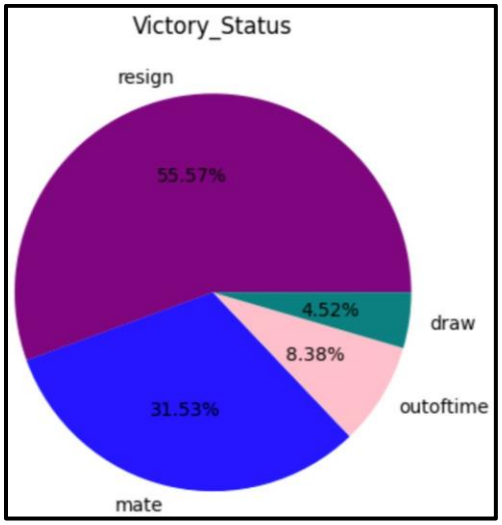
Preprocessing

After conducting a thorough analysis of the dataset containing over 20,000 records, it was discovered that there were no instances of missing or null values. The preprocessing in this project involved extensive data manipulation to ensure the dataset was ready for machine learning tasks. First, the data was cleansed by dropping duplicates and creating bins for 'turns' using 'pd.cut', allowing us to categorize games based on the number of moves. Further, categorical features like 'game_type', 'ECO_Names', 'rating_level', and 'turns_binned' were transformed into factors, facilitating one-hot encoding for machine learning. The column 'winner' was transformed into a numeric variable, providing a binary outcome for model training. Additional preprocessing included creating a new feature 'game_time_duration' by subtracting 'created_at' from 'last_move_at', followed by filtering out zero-duration games. 'Rating_diff' was calculated to represent the absolute difference between 'white_rating' and 'black_rating', with further categorization into 'Low', 'Mid', and 'High' based on defined thresholds. Data was then split into training and test sets using an 85/15 ratio. For feature scaling, we used MinMaxScaler for 'initial_time' and 'rating_diff', and StandardScaler for 'turns', 'white_rating', and 'black_rating', ensuring consistent data scaling. Categorical data was encoded with OneHotEncoder to create dummy variables. These preprocessing steps, encompassing data cleaning, feature engineering, scaling, and encoding, were crucial to preparing the dataset for robust model training and evaluation. This indicates that the dataset is complete and accurate and can be used confidently for further analysis and decision-making.

Exploratory Data Analysis

In the current phase, we have successfully executed various functions such as descriptive statistical analyses, data aggregation, and grouping techniques. These functions proved to be highly beneficial in providing us with new insights into the data. We utilized powerful data visualization libraries such as matplotlib, pyplot, seaborn, and pandas libraries to visualize the data and discover hidden patterns and trends. By leveraging these data visualization techniques, we were able to uncover valuable insights that would have been difficult to identify otherwise.





Participants in a game make a series of moves commonly called "turns." An analysis of these turns reveals that they follow a Gaussian distribution with a positive skew, which is indicative of a normal data distribution. This observation can be attributed to the inclusion of various substantial datasets. It is worth noting that the number of turns that are less than six amounts to 284, which makes up only 2.0% of all games. In a game of chess, there are four main ways in which the game can end. These are: when the time on the chess clock runs out, when a player voluntarily forfeits by resigning, when a player is checkmated, or when the game ends in a draw. To keep track of how each game ends, the data includes a field called 'victory_status'. An analysis of this field has been conducted, resulting in a pie chart that shows the percentage of each outcome. The pie chart reveals that the most common way in which games end is by resignation.

Methods And Error Metrics

For the predictive modeling aspect, we will experiment with various machine learning classification algorithms such as logistic regression, decision trees, random forests, gradient boosting, LDA, QDA and KNN. We will evaluate the performance of these models using appropriate metrics such as accuracy, precision, recall, MCC, and F1-score. Additionally, we will employ techniques such as cross-validation to assess model stability and generalization to unseen data.

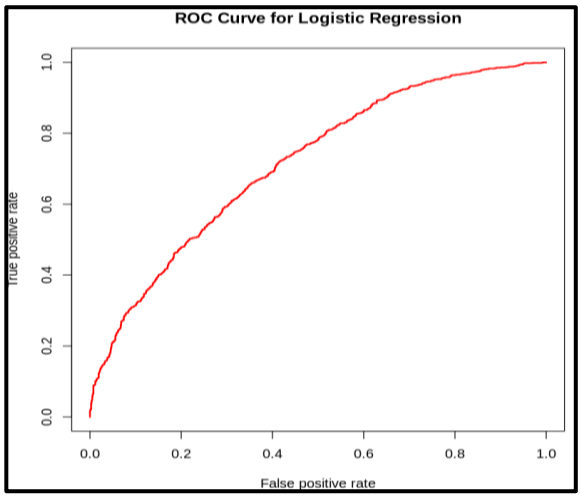
Furthermore, we plan to leverage data visualization techniques in Python and R, to create insightful visualizations that help us understand patterns and trends in the data. These visualizations will aid in interpreting model results and communicating our findings effectively in the final report.

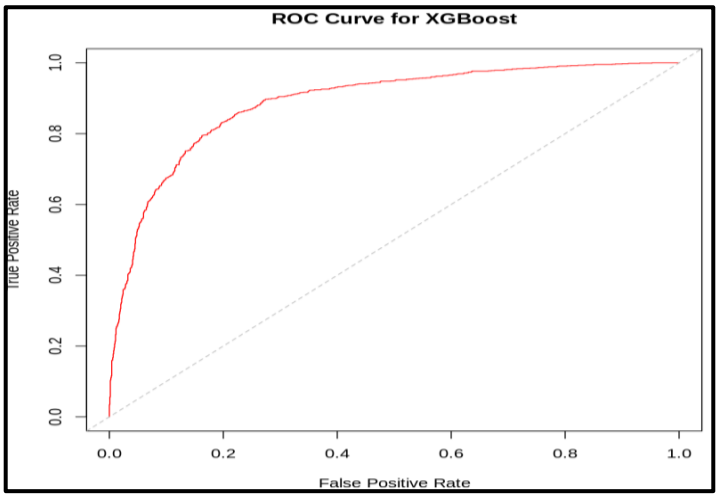
Overall, our project aims to provide valuable insights into player performance in chess games using statistical analysis and machine learning techniques. We believe that understanding the factors influencing game outcomes can have practical implications for both players and chess enthusiasts, such as improving training strategies or developing personalized recommendations for players to enhance their performance by providing actionable insights. The task involved several modeling techniques to predict the winning chances in a chess game, given features like 'turns', 'white_rating', 'black_rating', 'initial_time', and 'rating_diff'. Logistic regression, linear discriminant analysis, and quadratic discriminant analysis were among the initial approaches used to understand the relationships between these variables and the binary outcome. In logistic regression, significant factors included 'white_rating', 'black_rating', and 'turns', with the coefficients indicating the strength and direction of these relationships. The linear discriminant analysis also emphasized the importance of 'white_rating' and 'black_rating', while 'turns' had a lower impact.

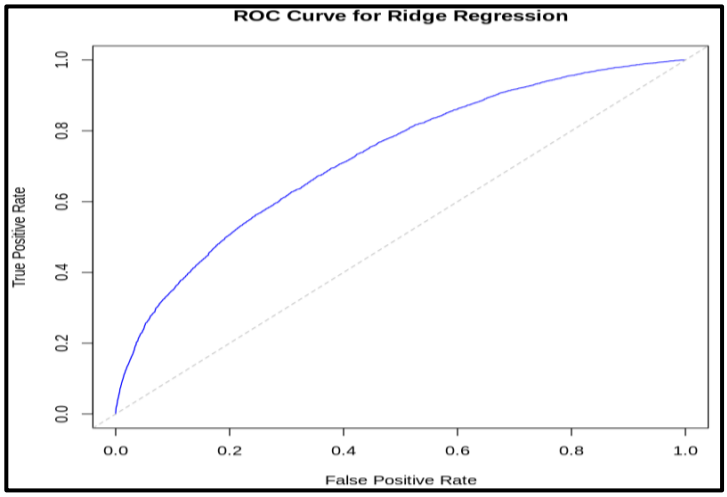
Subset selection analyses provided a deeper exploration into optimal combinations of predictors. Both forward and backward selection processes revealed that 'black_rating', 'white_rating', and 'turns' were consistently chosen, with 'rating_diff' showing a less significant impact. Ridge regression and lasso regression, with their emphasis on regularization, presented sparse models, with a focus on minimizing overfitting while maintaining predictive power. Ridge regression considered all predictors, while lasso regression produced a sparser model, focusing on the most impactful variables, thus promoting model simplicity without sacrificing accuracy.

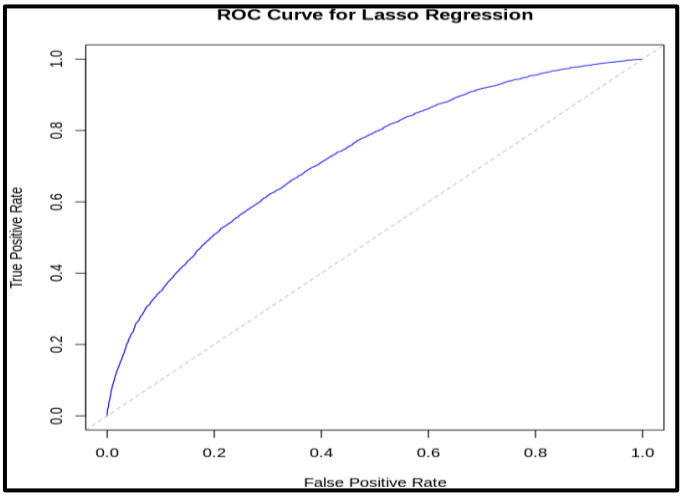
The ROC (Receiver Operating Characteristic) curve is useful because it visually demonstrates the trade-off between sensitivity (true positive rate) and specificity (false positive rate), allowing for an effective evaluation of a classifier's performance at various threshold

levels and enabling the selection of an optimal threshold for the desired balance between true positives and false positives. Below are some ROC Curves from our model.



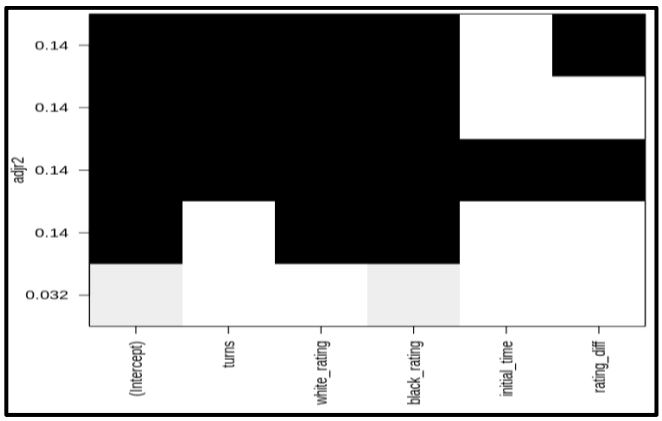


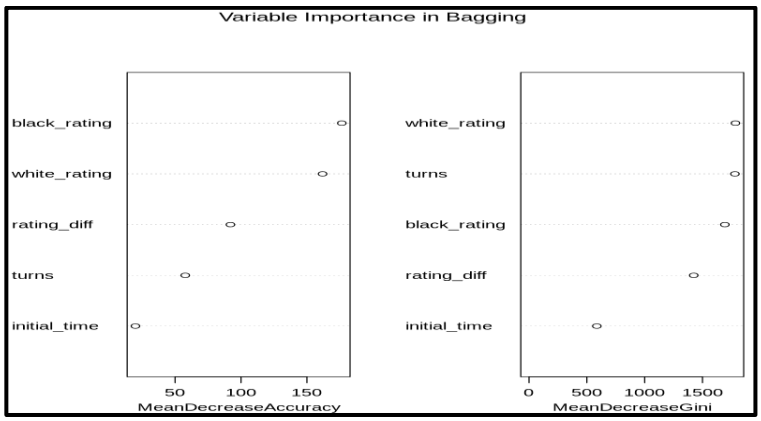




Random Forest, bagging, and boosting models like XGBoost were explored for more complex and robust predictions. The Random Forest model, with 500 trees and 2 variables tried at each split, yielded an Out-of-Bag (OOB) error rate of 30.61%, indicating moderate accuracy. Bagging, with all variables considered at each split, resulted in a slightly higher OOB error rate of 32.15%, suggesting a potential for overfitting. XGBoost, with its focus on boosting and tree-based learning, revealed the relative importance of each feature, with 'turns' having the highest gain, cover, and frequency, indicating its significant role in prediction. Despite variations across these models, a consistent theme emerged: the critical role of player ratings, the number of turns, and the limited impact of initial time and rating differences on prediction outcomes.

Results





The first image, which illustrates adjusted R-squared values against various variables, has the y-axis labeled as 'adjR2', indicating the adjusted R-squared value - a measure of how well the independent variables explain the variance in a regression model. The values range from 0.032 to 0.14, and the x-axis contains labels for several variables including 'turns', 'white_rating', 'black_rating', 'initial_time', and 'rating_diff'. The black bars suggest these variables have higher adjR2 values, indicating a more substantial impact on the model's outcome compared to other variables.

The second image focuses on variable importance in bagging by XGBoost. It has two key metrics: 'mean decrease accuracy' and 'mean decrease gini', which show the importance of each variable in a bagging model. For 'mean decrease accuracy', the 'black_rating' and 'white_rating' variables have the highest values, exceeding 150, while 'rating_diff' is around 100, 'turns' is approximately 60, and 'initial_time' has the lowest importance. For 'mean decrease gini', 'white_rating', 'turns', and 'black_rating' are the top variables, with values over 1500, while 'rating_diff' is slightly below 1500, and 'initial_time' is significantly lower at around 600. These results suggest that in the bagging model, 'black_rating' and 'white_rating' are the most prominent features for accuracy, whereas for the Gini metric, 'white_rating' and 'turns' are among the most influential.

Conclusion

The analysis of the chess dataset has revealed how the opening moves impact game outcomes and player ratings. The dataset analysis shows that having a higher player rating does not guarantee victory. The analysis provides detailed insights into the individual player and team performance. It highlights the importance of strategic moves and player skills in chess games. Overall, the analysis of the chess dataset offers valuable information for understanding player dynamics and outcomes.

Future works can include analysis and implementations such as:

- ✓ Player Profiling: Develop player profiles based on playing style, preferred openings, and strategic tendencies.
- ✓ Recommendation Systems: Implement recommendation systems to suggest optimal moves or openings based on game context and player preferences.
- ✓ Community Detection: Identify communities of players with similar playing styles or strategic preferences using network analysis techniques.
- ✓ Integration with External Data: Integrate external datasets such as weather data or event calendars to analyze the impact of external factors on game outcomes.

The dataset presents several comments, concerns, and limitations that warrant consideration such as:

- ✓ The Eco ratings provided for players may not always accurately reflect their skill levels. The ratings are based on a player's performance in previous games, but factors such as player improvement, fluctuations in performance, and the context of individual games may not be fully captured. This could impact the accuracy of analyses related to player performance and game outcomes.

- ✓ While the dataset includes various attributes related to chess games, it may have limitations in terms of the feature set available for analysis. Additional features such as time spent on each move, player demographics, or contextual information about the games (e.g., tournament type, event location) could provide deeper insights but are not included in the dataset.
- ✓ There could be potential sampling bias in the dataset, as it may primarily consist of games played on a specific online chess platform (e.g., Lichess). This could lead to a skewed representation of player demographics, playing styles, and game contexts, affecting the generalizability of findings to the broader population of chess players.
- ✓ The dataset includes information about the openings used in each game, categorized according to the Encyclopedia of Chess Openings (ECO) classification system. However, there may be inconsistencies or inaccuracies in the classification of openings, as the classification relies on human judgment and interpretation, which can vary among annotators.
- ✓ The dataset may lack contextual information about the games, such as the time and location of play, the level of competition, or the players' strategic goals and motivations. Understanding the broader context of the games could provide valuable insights into the factors influencing player behavior and game outcomes but this is not available in the dataset.

Addressing these comments, concerns, and limitations is crucial for ensuring the validity and reliability of analyses conducted using the dataset. It is essential to approach the data with caution, critically evaluate the findings, and consider potential biases and uncertainties in the interpretation of results. Additionally, supplementing the dataset with additional contextual information and conducting sensitivity analyses can help mitigate some of these limitations and enhance the analysis's robustness.

References

Riccardo D. The Ultimate Checkmate: AI and Chess Engines. Codemotion. Retrieved Feb 12, 2024, from *https://www.codemotion.com/magazine/ai-ml/the-ultimate-checkmate-ai-and-chess-engines/*

Jacob Y. How AI Powers Chess Engines and Creates Grandmasters. PureStorage. Retrieved Feb 12, 2024, from *https://blog.purestorage.com/perspectives/how-ai-powers-chess-engines-and-creates-grandmasters/*

Statistical analysis of the Lichess dataset (chess game) using Python. Retrieved Mar 12, 2024, *https://medium.com/@mahmed31098/statistical-analysis-of-the-lichess-dataset-chess-game-using-python-f63eeaf7b592*