Decoding Chess: Analysis of 20,000 Games

Abhiram Naredla, Devarsh Shah, Veer Patel

School of Information, University of Texas at Austin

I310D: Intro to Human-Centered Data Science

Professor Abhijit Mishra

06 December 2023

Introduction

With the recent chess boom in the past years, the game of chess has skyrocketed in popularity and online chess has provided a very convenient and accessible platform for newcomers. Our project explores the game of chess in an effort to identify the underlying patterns, strategies, and variables that have an impact on a game's result. Our project attempts to provide useful insights into winning strategies for both the white and black pieces, while also examining the relationship between particular chess openings and game outcomes, with an emphasis on meeting the needs of casual and beginner-level chess players.

Dataset

We used a dataset we found on Kaggle titled "Chess Games Dataset (Lichess)" (Link to the dataset: Chess Game Dataset (Lichess) | Kaggle). It comprises over 20,000 rows taken from chess games played on the second most popular free online chess server, Lichess.org using the Lichess API.

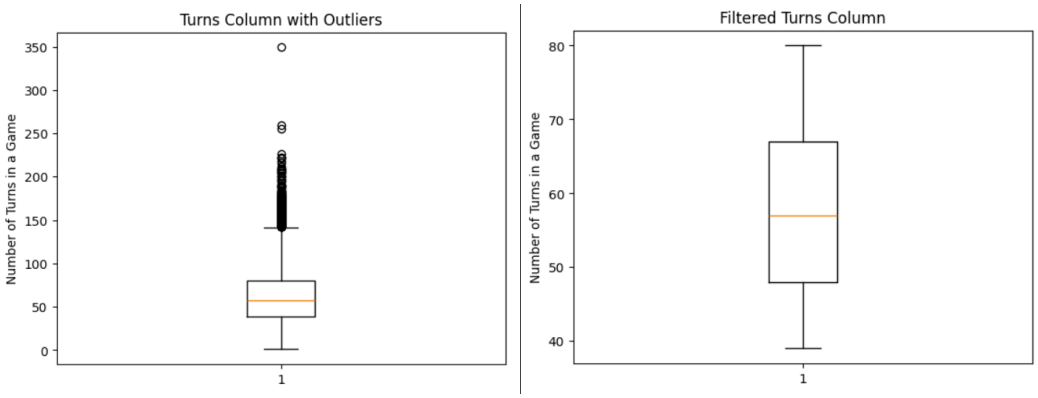
Data Description

The raw dataset consists of 16 columns but we only used the following relevant fields for our data analysis and machine learning models:

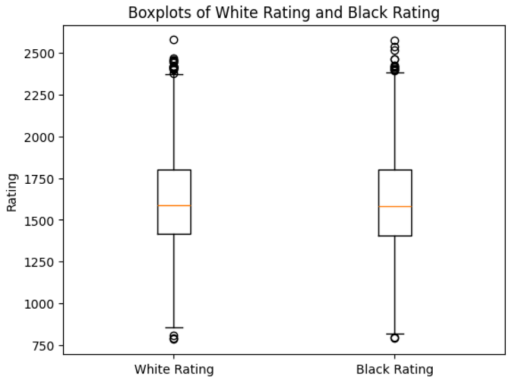
1. Turns - Total number of turns in the game.
2. White Rating - Standardized Lichess rating of the player with the white pieces.
3. Black Rating - Standardized Lichess rating of the player with the black pieces.
4. Opening Name - Official name of the chess opening
5. Winner - Which player won the game (white or black pieces).

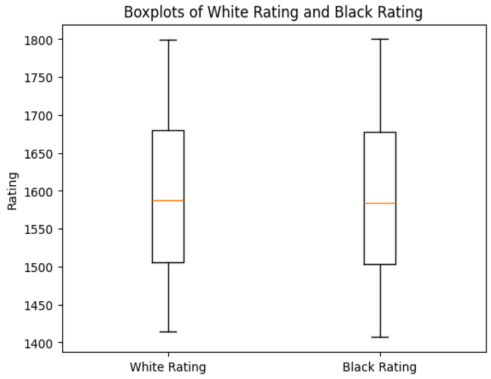
Method

1. Exploratory Data Analysis: We started by examining our dataset to uncover patterns, outliers and potential areas of interest. We used descriptive statistics and visualizations like box plots and bar graphs to find correlations between the features, and potential outliers in the data.
2. Data Cleaning Steps:
    a. opening_name column - This field initially had 1477 unique variations, which would've been too much for our model to handle since our dataset merely consists of 20,000 rows. We brought this value down to 169 by only accounting for the type of opening and not any variations of the opening. For instance: "Slav Defense: Exchange Variation" is just Slav Defense in our model.
    b. turns column - This field initially had a substantial amount of outliers, as depicted by the box plot below. We decided to filter this data by only accounting for values between its 25th and 75th percentile, to mitigate the impact of outliers in our analysis and ensure a valid representation of the central tendency of this column.

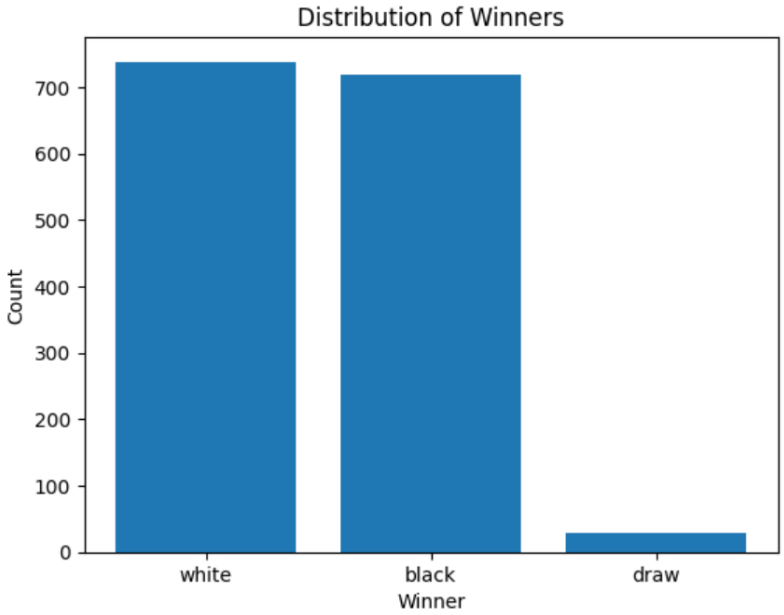


c. white/black rating columns - These fields also had a significant amount of outliers, as depicted by the following boxplots so we decided to filter them the same way as “turns” column (only include data points between its 25th and 75th percentile).





d. winner column - After thorough cleaning of the dataset, this field had a negligible amount of “draw” values, which are instances where the game resulted in a tie. To account for this, we decided to remove all “draw” values from the dataset to maintain the accuracy of our analysis.



3. Encoding the “opening_name” column: In order for our machine learning algorithm to be trained, we needed to change the categorical variable, that is the chess opening name, and convert it into a binary matrix. We used one-hot encoder to identify all the unique opening names and create binary columns for each unique opening name.
4. Developing a suitable Machine Learning algorithm: In order to accurately predict the outcome of sample chess games based on selected features, we decided to engineer a machine learning model that would take number of turns and player rating as input variables and return the predicted winner of the game. We selected four models to train with the resampled data: Logistic Regression, MLP Classifier, Random Forest, and K-Nearest Neighbour (an explanation of the models and rationale can be found in the Analysis Tools section).
5. Obtaining Results and Evaluation Metrics: In order to assess the overall performance of our ML models, we decided to use accuracy_score from the scikit-learn library due to its robust implementation and compatibility with various machine learning models. The accuracy score is the measure of the number of correct predictions made by our model in relation to the total number of predictions made. The accuracy of our ML models were as follows:

```
Accuracy of the Logistic Classifier = 0.60
Accuracy of the MLP Classifier = 0.49
Accuracy of the k-Nearest Neighbour Classifier = 0.55
Accuracy of the Random Forest Classifier = 0.58
```

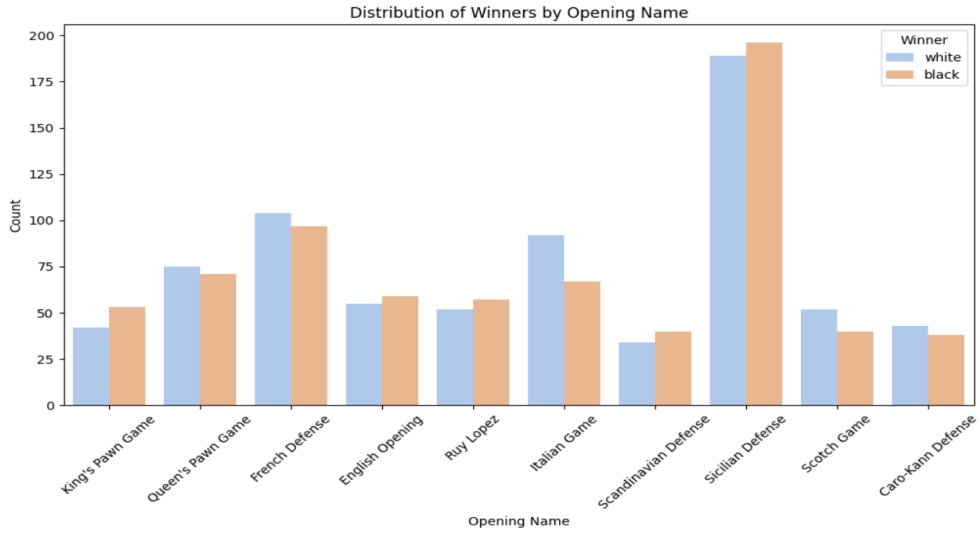
6. Testing the algorithm: Among the models tested, the logistic regression model emerged as the most effective one, having the highest accuracy score in predicting the winner of

the sampled chess games. Our model takes turns, white_rating, black_rating, and opening_name as input variables and outputs the predicted winner of the chess game with an accuracy of 60 percent. A sample test of our model is as follows:

```
# Sample input data for testing
turns = 13
white_rating = 1500
black_rating = 1191
opening_name = "Slav Defense"
```

```
Predicted Winner: white
```

7. Data Visualization: We used the following bar plot, that represents the win rates associated with different chess openings for both white and black pieces, to exhibit how some of the most popular chess openings perform for both the white and black pieces. Entry-level chess players could use this information to choose and stick to specific openings to achieve the best results possible. For instance, a new chess player can clearly tell from this bar chart that the “italian game” opening is very strong when played using the white pieces and can practice this opening to eventually achieve the best results.



Analysis Tools

1. Logistic Regression

   Logistic regression is a model which in simple terms, shows the probability of an event occurring. It is generally used when the dependent variable is binary (in our case, winner is a binary value). It uses a logistic or sigmoid function to find the probability. The solver for optimization in our model is a limited-memory Broyden-Fletcher-Goldfarb-Shanno Algorithm (LBFGS).

2. Neural Network (MLP Classifier)

   A multilayer perceptron is an artificial neural network which includes neurons arranged in at least three layers or rows including an input layer, a hidden layer, and an output layer. The learning in this model is achieved by a process called backpropagation. The weight optimization is handled by a limited-memory Broyden-Fletcher-Goldfarb-Shanno Algorithm (LBFGS) again. We used this algorithm instead of a stochastic gradient descent-based optimizer because we have a relatively small dataset and LBFGS works better for smaller datasets.

3. Random Forest Classifier

   A random forest model is a machine learning algorithm that makes a number of decision trees whilst the model is being trained. Each tree is independently trained on a random subset of the dataset, this process being called bootstrapping and further aggregation being referred to as bagging. In classification tasks such as ours, the class selected by the highest number of decision trees is given as the output/prediction. This classifier helps

reduce overfitting and provides a more robust model than one obtained from using a single decision tree.

4. k-Nearest Neighbor Classifier

   A k-NN model is a machine learning model that uses proximity or distance, usually in terms of Euclidean distance, to make a prediction regarding the grouping of a data point. The value of k for our model is 5 which means that the closest 5 neighbors to the data point are to be considered when making predictions.

Results

In this project, we used data analysis and machine learning methods to gain useful information for beginner-level chess players. The findings can be useful in selecting chess openings for our target audience, for example some openings like the Italian Game and Scotch Game have higher win rates for the player with the white pieces whilst other openings such as the Sicilian Defense have better win rates for the player using the black pieces. Thus, a new chess player could utilize these results by practicing specific chess openings to achieve the best win rate.

Conclusion and Limitations

Despite the valuable knowledge gained from our meticulous analysis, it is important to acknowledge the limitations of our data analysis that may influence the reliability of our findings. Some limitations that we identified are as follows:

- Limited Dataset size: Since our initial dataset consisted of less than 20,000 rows, we were constrained to a significantly small amount of data, especially after data cleaning. We believe this was a major reason for the low accuracy of our machine learning model.
- Unequal Representation of Chess Openings: Over 3,000 chess openings and specific variations exist, but our project only deals with 169 of these openings. This results in certain less popular openings not being accounted for which is a limitation in our findings.
- Exclusion of Draws: In our model, we excluded all draws so as to focus only on decisive outcomes but drawn games can also offer valuable insights into the game of chess.
- Assumption of a Single Strategy: One assumption of our project is that there is a degree of similarity across all players, however individual playing styles have a significant influence on the game as well. Furthermore, a chess opening is only the first part of a game and the middle and end game portion of a match have a major impact on the outcome of the match as well. Thus, our findings are limited in this sense.

Appendix

After our in-class presentation, where we presented our method, findings and limitations, we were asked the following questions:

1. Explain how the K-Nearest-Neighbour Classifier works?

As mentioned in the Analysis Tools section, the K-Nearest-Neighbour Classifier uses distance to make a prediction regarding the grouping of a data point.

2. How did you encode the categorical variable from your dataset so that you could reliably use it in your machine learning model?

We used One Hot Encoding to convert our categorical column (opening_name) into binary values so that we could use this in our machine learning model. One Hot Encoding is a technique in which each unique category in the categorical variable is represented by a binary column, and only one of these columns is marked as "1" for each observation, while the rest are marked as "0".

References

Banoula, M. (2023, November 7). An introduction to logistic regression in Python. Simplilearn.com. https://www.simplilearn.com/tutorials/machine-learning-tutorial/logistic-regression-in-python#:~:text=Logistic%20regression%20is%20a%20statistical,one%20or%20more%20independent%20variables.

Bento, C. (2022, January 5). Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis. Medium. https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141

“Chess Game Dataset (Lichess).” Kaggle, 4 Sept. 2017, www.kaggle.com/datasets/datasnaek/chess/data.

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

What is the k-nearest neighbors algorithm? | IBM. (n.d.). https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20algorit

hm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point.

Other Links

Github link: https://github.com/ShahDevarsh0209/Chess-Data-Analysis/tree/main

Slideset: I 310D: Final Presentation(Fobs)