ML Group 86

Introduction & Background

Lately there has been a lot of attention on forecasting the results of chess matches with the growing popularity of machine learning methods. Past studies, such as Aluna Riz's research with LightGBM have demonstrated how gradient boosting algorithms can effectively predict chess outcomes by analyzing player data and game related characteristics [1]. This research furthers our knowledge that factors like player ratings, game length and opening moves influence winning outcomes. In this project's dataset obtained from Kaggle there are over 20,000 chess game records that include details such as player ratings, game outcomes, and time controls for each game played. The varied information available allows for an examination of gameplay dynamics and the meaning of ratings, furthering insight into the prediction of success or failure in competitive chess [2]. https://www.kaggle.com/datasets/datasnaek/chess

Problem Definition

Problem: Finding the factors that most significantly impact the outcome of chess games, these could be player experience/ranking, game duration, or opening strategies, and creating a model to predict game results.

Motivation: Through identifying the determinants of success and creating predictive models, chess enthusiasts, players, and coaches can benefit by analyzing these game trends and utilizing prediction in regards to strategic planning matches, or personalized training.

Methods

Preprocessing

- Data Cleaning
    - Cleaning that should already be done from Kaggle: Handling missing data, removing duplicates, fixing errors (spelling, format)
    - Handling Outliers
- Dimensionality Reduction
    - We are not going to need all the features presented for each model

- Feature Engineering
  - Creating Interaction Features
  - Binning
- Sampling Data
  - If we want to look at smaller subsections of the 20,000 games
- Data Transformation
  - Normalization
  - Log transformation if large values dominate

ML Supervised Methods

- Random Forest:
  Reason: group of decision trees, which makes it highly effective at handling complex, non-linear relationships in the data. It is also robust to overfitting because of its averaging nature, which makes it a strong candidate for chess prediction, where numerous features (e.g., moves, positions, strategies) interact in complex ways. It can handle categorical and numerical data well and often performs well without extensive parameter tuning.
- Neural Networks (NN):
  Reason: Neural Networks, particularly deep learning models, are capable of learning highly complex patterns and making decisions based on features that might not be explicitly coded (e.g., intricate chess strategies). They are especially useful when provided with large datasets, which is common in chess with thousands of game records. NN models can handle non-linear and high-dimensional data effectively, making them suitable for a complex game like chess. [2]
- Support Vector Machines (SVM):
  SVMs are effective in high-dimensional spaces and are particularly useful when the decision boundary between classes (e.g., win, loss, draw) is complex. With the right kernel, SVMs can capture non-linear relationships in the data, which is essential for chess predictions. They also work well with smaller datasets compared to neural networks. [3]

(Potential) Results + Discussion

In this project, we’ll use accuracy, F1 score, and ROC-AUC to evaluate model performance. Accuracy provides a quick snapshot of overall correctness, F1 score balances precision and recall to handle imbalanced data, and ROC-AUC measures the model’s ability to distinguish between outcomes. Our goals are to achieve high predictive accuracy while minimizing bias, ensuring fairness across player ratings and game outcomes. Ethical considerations include avoiding overfitting to specific player pools and ensuring data privacy. We expect Random Forest to perform well due to its robustness, while Neural Networks might excel with larger datasets. Overall, we anticipate discovering insights into the impact of openings and player ratings on game outcomes, contributing to more strategic chess analysis.

References

[1] P. Aluna Rizzoli, “Predicting chess games results using lightgbm,” *Medium*, 16-Oct-2023. [Online]. Available: https://medium.com/@alunariz/predicting-chess-games-results-using-lightgbm-818f30b5a7c3. [Accessed: 02-Oct-2024]

[2] V. Kumar, D. Singh, G. Bhardwaj, and A. Bhatia, “Application of Neurological Networks in an AI for Chess Game,” *2020 Research, Innovation, Knowledge Management and Technology Application for Business Sustainability (INBUSH)*, Greater Noida, India, 2020, pp. 125-130, doi: 10.1109/INBUSH46973.2020.9392188.
Keywords: Technological innovation, Modulation, Games, Programming, Pattern recognition, Task analysis, Sustainable development, Neurological network, Reinforcement learning, Supervised, Unsupervised.

[3] N. L. T. Tra, P. T. Cong, and N. D. Anh, “Design A Chess Movement Algorithm and Detect the Movement by Images Classification Using Support Vector Machine Classifier,” *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*, Ho Chi Minh City, Vietnam, 2018, pp. 335-340, doi: 10.1109/GTSD.2018.8595604.
Keywords: Robot kinematics, Support vector machines, Classification algorithms, Image processing, Artificial intelligence, Cameras, Chess program, Image processing, MATLAB, Support vector machine.

Contribution Table

TASK TITLE	TASK OWNER
Project Team Composition	All
Project Proposal	
Introduction & Background	Randall
Problem Definition	Randall
Potential Dataset	Anay & Drew
Methods	Anay & Drew
Potential Results & Discussion	Apollo & Nithil

TASK TITLE	TASK OWNER
Gantt Chart	Drew
Video Recording	All
GitHub Page	Randall

Gantt Chart

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION
Project Team Composition	All	8/26/2024	9/14/2024	18
Project Proposal		9/23/2024	10/4/2024	12
Introduction & Background	Randall	9/23/2024	10/4/2024	12
Problem Definition	Randall	9/23/2024	10/4/2024	12
Potential Dataset	Anay & Drew	9/23/2024	10/4/2024	12
Methods	Anay & Drew	9/23/2024	10/4/2024	12
Potential Results & Discussion	Apollo & Nithil	9/23/2024	10/4/2024	12
Gantt Chart	Drew	9/23/2024	10/4/2024	12
Video Recording	All	9/23/2024	10/4/2024	12
GitHub Page	Randall	9/23/2024	10/4/2024	12
Midterm Report		10/5/2024	11/8/2024	34
Model 1 (M1) Design & Selection	All	10/5/2024	10/11/2024	7
M1 Data Cleaning	Nithil	10/5/2024	10/11/2024	7
M1 Data Visualization	Drew	10/5/2024	10/11/2024	7
M1 Feature Reduction	Apollo	10/5/2024	10/11/2024	7
Fall Break	All	10/12/2024	10/15/2024	4

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION
M1 Implementation & Coding	Randall & Anay	10/16/2024	10/22/2024	7
M1 Results Evaluation	All	10/23/2024	10/24/2024	2
Model 2 (M2) Design & Selection	All	10/25/2024	10/29/2024	5
M2 Data Cleaning	Nithil	10/25/2024	10/29/2024	5
M2 Data Visualization	Drew	10/25/2024	10/29/2024	5
M2 Feature Reduction	Apollo	10/25/2024	10/29/2024	5
M2 Coding & Implementation	Randall & Anay	10/30/2024	11/4/2024	6
M2 Results Evaluation	All	11/5/2024	11/6/2024	2
Midterm Report	All	11/6/2024	11/8/2024	3
Final Report		11/9/2024	12/3/2024	25
Model 3 (M3) Design & Selection	All	11/9/2024	11/13/2024	5
M3 Data Cleaning	Nithil	11/9/2024	11/13/2024	5
M3 Data Visualization	Drew	11/9/2024	11/13/2024	5
M3 Feature Reduction	Apollo	11/9/2024	11/13/2024	5
M3 Implementation & Coding	Randall & Anay	11/14/2024	11/18/2024	5
M3 Results Evaluation	All	11/19/2024	11/20/2024	2
M1-M3 Comparison	All	11/21/2024	12/3/2024	13
Video Creation & Recording	All	11/21/2024	12/3/2024	13
Thanksgiving	All	11/27/2024	12/1/2024	5
Final Report	All	11/21/2024	12/3/2024	13