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.

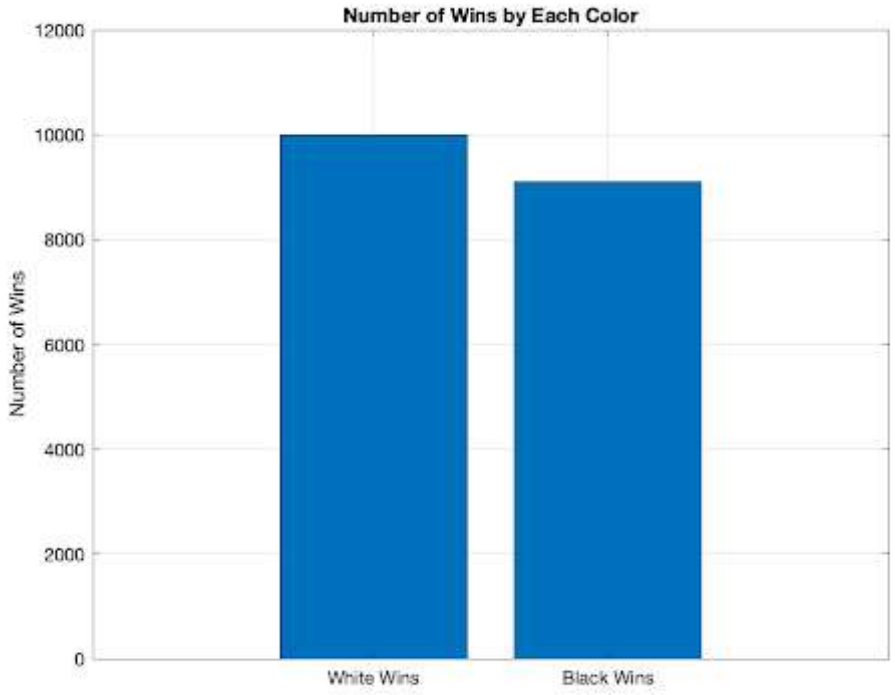
Motivation: Through identifying the determinants of success and creating predictive models, chess enthusiasts, players, and coaches can benefit by analyzing these game trends and utilizing prediction in regards to strategic planning matches, or personalized training.

Color Advantage

There is a general consensus in the world of chess that going first as the White piece is an advantage. By going first the player has the initiative and should try to extend it into the middlegame, while Black should strive to neutralize White's initiative and attain equality. Because White begins with the initiative, a minor mistake by White generally leads only to loss of the initiative, while a similar mistake by Black may have more serious consequences.
(https://en.wikipedia.org/wiki/First-move_advantage_in_chess#:~:text=Joseph%20Bertin%20wrote%20in%20his,while%20Black%20should%20strive%20to)

Reasons going first provides an advantage:

- **Initiative**: White makes the first move, which allows them to dictate the pace and direction of the game. This initiative can lead to more aggressive and advantageous positions.
- **Control of the Center:** Many opening strategies emphasize controlling the center of the board. By moving first, White can more effectively establish control over key central squares, which can lead to better piece activity and development.
- **Psychological Pressure:** Starting the game can create psychological pressure on Black. White's first move can set a tone for the game, and Black must respond to White's choices, which can sometimes lead to defensive play.
- **Fewer Defensive Moves:** White generally has the opportunity to develop pieces and create threats without needing to respond to an opponent's threats first. This can allow White to build more dynamic positions.
- **Opening Preparation:** Many chess players study and prepare specific openings as White, taking advantage of their first move to create familiar positions that they are comfortable playing. This preparation can lead to stronger play compared to responding as Black, where the focus is often on countering White's strategy.
- **Statistics:** Historically and statistically, games played with White tend to result in a higher win rate compared to those played with Black, reinforcing the perceived advantage of playing first.

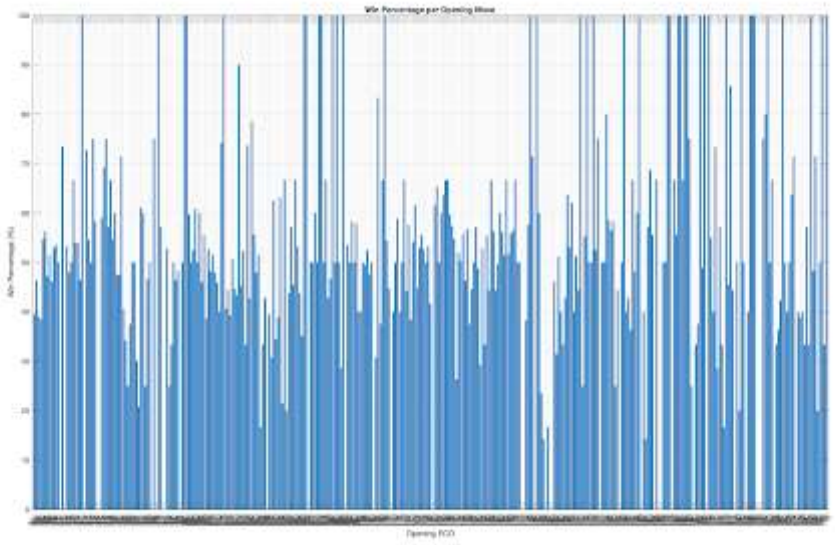


Similar to historical data, the dataset being examined had White winning more often than Black. Although this could be a factor that is implemented into our overall model, it should not have a strong weight in decision making for which player will win. Below are the following reasons for this.

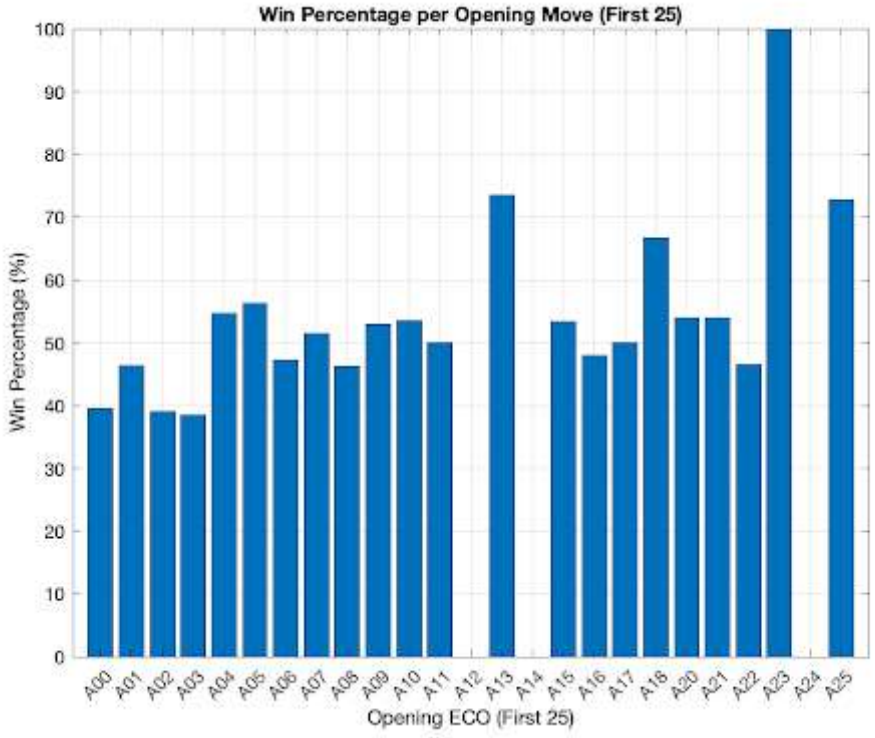
- **Correlation vs. Causation:** The slight advantage of White is a general trend, but it doesn't dictate the outcome of every individual game. Many other factors, such as player skill, specific opening strategies, and in-game decisions, play a significant role in determining the outcome.
- **Player Ratings:** Player ratings are often a more significant indicator of expected performance than color. Higher-rated players tend to win more frequently, regardless of whether they are playing as White or Black. Including player ratings in the model can provide a more nuanced understanding of each player's capability.
- **Game Context:** The context of each game matters—such as player experience with specific openings, styles of play, and psychological factors. A player who is adept at handling certain positions may perform better as Black against a specific opponent, negating the typical advantage of White.
- **Data Distribution:** If the model heavily weights the color advantage, it may not generalize well to different contexts or player pairings. Overfitting to the color advantage could lead to less accurate predictions in matchups where other factors are more influential.
- **Variation in Outcomes:** While White may have a statistical advantage, the margin is not large. Many games end in draws, and the specific dynamics of each game can vary widely. The model should capture this variability by considering a range of features rather than overly focusing on color.

Opening Move Impact on Winning

The opening move aspect of chess plays hand in hand with White having an advantage since they have the initiative and can drive the behavior of the game in the early stages to some degree. Although having this initiative is beneficial, opening moves vary in their effectiveness. There are some that are regarded as always good choices and some that can lead to negative impacts by leaving value pieces vulnerable. For the dataset being examined there were 365 openings. For each opening the corresponding win percentage was plotted.



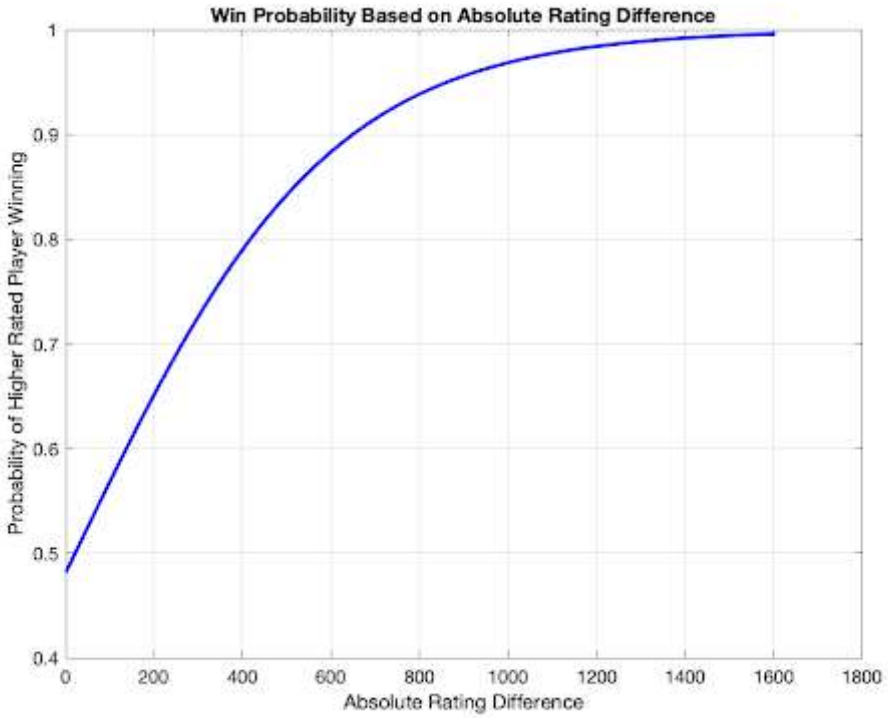
This figure shows the results for all opening moves. The x-axis is the Opening Move Code and the y-axis is the Win Probability. See below for an extracted version that more clearly shows the results for a few Opening Moves.



Based on these results there are some opening moves that seem to increase the chances of winning while others decrease the chance of winning. It is important to note that this was completed for a general data visualization. It can be improved by incorporating how many games each type of opening was used for. The results can be very skewed if certain openings were only used once or a few times. There are also many other factors that impact the outcome of a chess game that need to be combined with these results. This data can be compared to what is typically considered the best openings and see if the trends match.

Rating Difference

When looking at the dataset during preprocessing, player rating was thought to be a key driver of win probability. By examining the absolute rating difference between opponents, we can better understand how relative skill levels impact game results. This analysis utilized a logistic regression model to explore the relationship between rating differences and the likelihood of the higher-ranked player winning.



The resulting plot illustrates how this probability varies across different levels of rating disparity. It indicates that as the rating difference between players increases, the probability of the higher-rated player winning increases rapidly at first but then levels off, reaching a plateau where additional rating differences have diminishing impact on the win probability. Beyond a certain skill gap, the stronger player is almost guaranteed to win, and further rating differences don't significantly change this likelihood.

Methods

Data processing method:

- We used pandas to create a dataframe of the dataset. This allowed us to properly use specific columns of data (winner, black_rating, white_rating, opening_ply, turns) to train and test. To prepare the winner column for machine learning, we applied LabelEncoder as a data preprocessing method. This tool converts the categorical values in the winner column (such as

'white', 'black', and 'draw') into numerical labels (0,1,2), making them compatible with our predictive models.

- To prepare the dataset we explored utilizing SMOTE
  - SMOTE (Synthetic Minority Oversampling Technique) was explore in order to generate synthetic samples for the minority class, this preprocessing step was done in aim to help the model learn from a more balanced dataset, increasing accuracy on minority classes like when games are draws

Machine Learning Algorithms

- We implemented the Gradient Boosting Classifier, this is an ensemble model which combines multiple weak learners (decision trees) to create a strong predictive model. Gradient Boosting was choosing for its ability to handle non-linear relationships, making it well suited for our data set (e.g., variations in player ratings, game runs, and opening plays)
- We implemented Random Forest, another ensemble model that builds multiple decision trees and aggregates the result of those trees usually through a majority vote. We selected the model because it has a low chance of overfitting and is quite effective in identifying more important features.

Supervised Learning Method:

- Our task of predicting the winner (black, draw, or white) based on historical game features utilized supervised learning classification. We used labeled data to train the model where each game record has a known winner.
- With our Gradient Boosting algorithm operating under supervised learning by training on labeled data seeking to minimize a loss function iteratively, we saw a model that can generalize unseen data to make accurate predictions.
- With our Random Forest algorithm operating under supervised learning, we trained on labeled data. This involved building multiple decision trees to create a model that can aggregate their predictions to generalize to unseen data and make accurate predictions.

Results and Discussion for Gradient Boosting

Quantitative Metrics

- Best Parameters: After hyper parameter tuning, we found that the best parameters for the Gradient Boosting model were:
  - learning_rate: 0.2
  - max_depth: 5

    - n_estimators: 150
- Accuracy: The model achieved an overall accuracy of 0.84 (84%), indicating that it correctly classified the winner for 84% of the test cases in the suite. This is a solid performance, showing that this model is generally effective in predicting outcomes.
- Balanced Accuracy: With a balanced accuracy of 0.61, the models performance across all classes, especially the monitory draw class, is lower than the overall accuracy. This metric accounts for imbalance between classes, showing that this model struggles more with less frequent outcomes.
- ROC-AUC Score: The score of 0.76 suggests a good discriminatory power in distinguishing between classes, but it still could be improved.

Classification Report

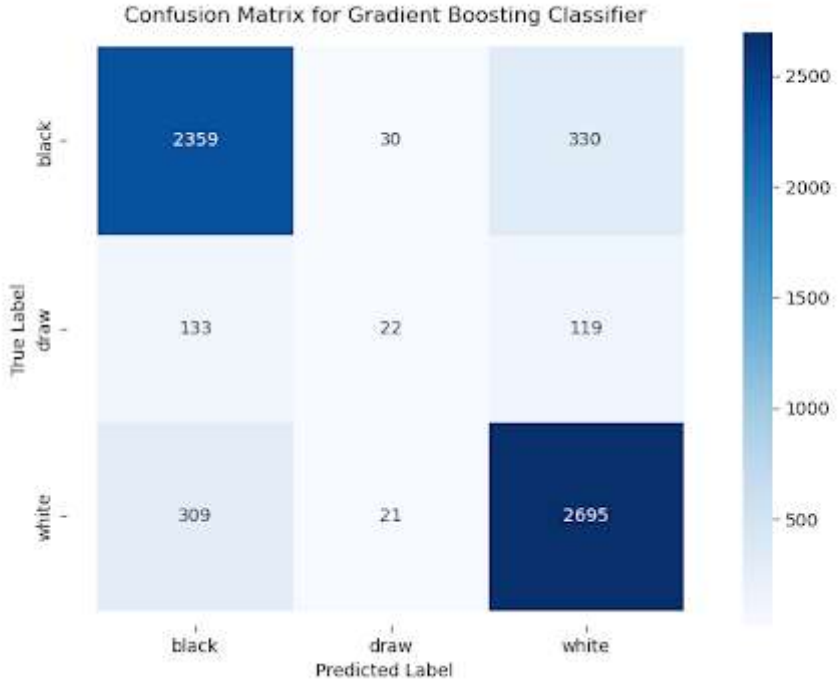
- Class-wise Performance:
    - Black: Our model achieved a precision of 0.84, recall of 0.87 and F1-score of 0.85. Through these metrics we can see indicated that the model is highly accurate in predicting black outcomes, which is expected due to their large presence in the dataset.
    - Draw: Precision of 0.31, recall of 0.08, and F1-score of 0.85. These metrics show that the model struggled with the draw class. The low recall in particular indicates that the model rarely correctly predicts draws. This is due to the class imbalance within the dataset, as draws are far less common than black or white outcomes.
    - White: For the white class, there was a precision of 0.86, recall of 0.89, and F1-score of 0.87. These metrics are similar to the black class, showing high performance.
- Macro and Weighted Averages
    - Macro Average: Precision, recall, and F1-score were all lowing (near 0.62) in the macro average, which treats each class equally. This metric draws attention and is affected by the poor performance of the draw class.
    - Weighted Average: The weight average accounts for the support of each class, leading to a higher average (near 0.84), this is more reflective of the model's strong performance with the majority classes.

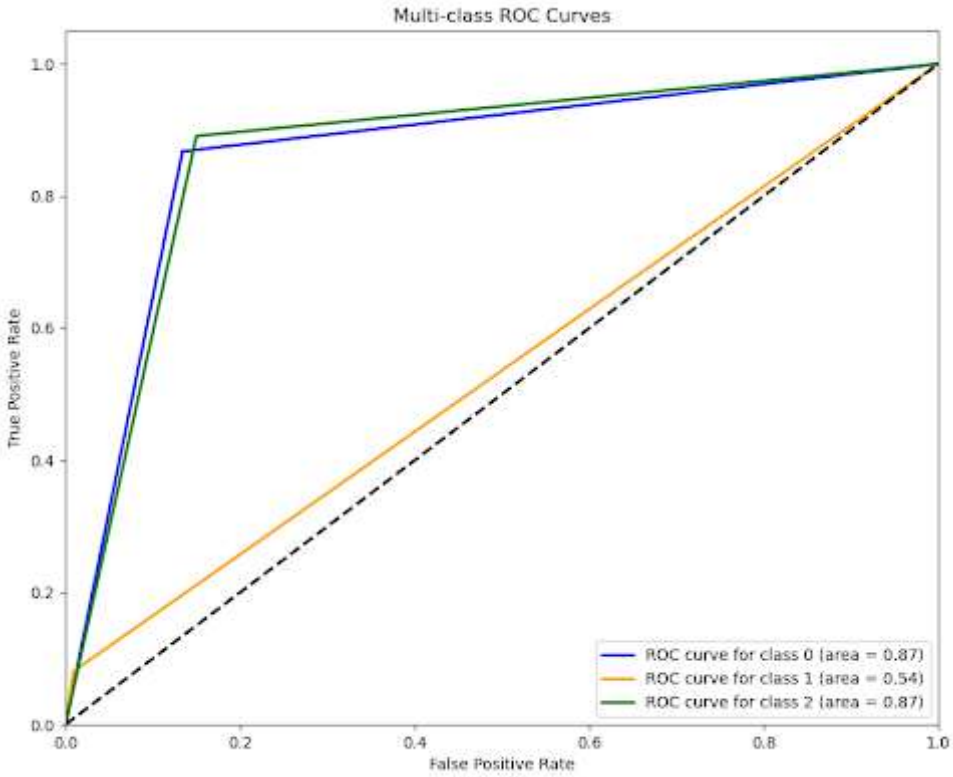
Analysis of Model Performance

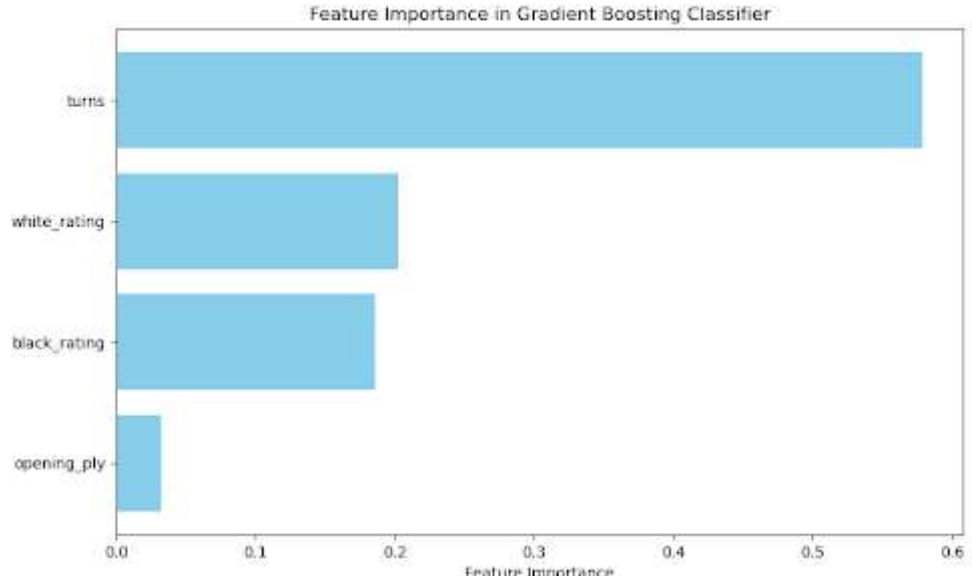
- Strengths: The Gradient Boosting Model performed well on the majority classes (black and white), with high precision, recall and F1-score. The high accuracy overall showed the model's effectiveness in handling complex, non-linear patterns.
- Challenges: The model's low performance regarding the draw class indicates a difficulty in predicting the minority class due to class imbalance. Despite SMOTE preprocessing and parameter tuning, the model still tends to favor the majority classes. The low performance for

minority classes suggest that additional strategies may be needed to improve predictions for classes with a lower presence.

Data Visualizations







\

Next Steps

Class Weight Adjustment: We could explore adjusting class weights within Gradient Boosting to give more emphasis to the draw class, seeking to balance the model's attention across classes more effectively.

Results and Discussions for Random Forest

Quantitative Metrics

Accuracy: The accuracy is 68% which means it classified the winner for 68% of cases with 6,018 samples.

Classification Report

- Class-wise Performance:
    - Black: The precision, recall, and f1-score were all .67 which is a strong performance in predicting all outcomes where black was the winner.
    - Draw: The precision, recall, and f1-score were .35,.11,and .17 respectively. A low recall indicates a large number of false negatives and a low f1-score means the model was not good in predicting draws.
    - White: The precision, recall, and f1-score were .70, .74, and .72 respectively. The f1-score indicates a strong performance in predicting all outcomes where white was the winner.

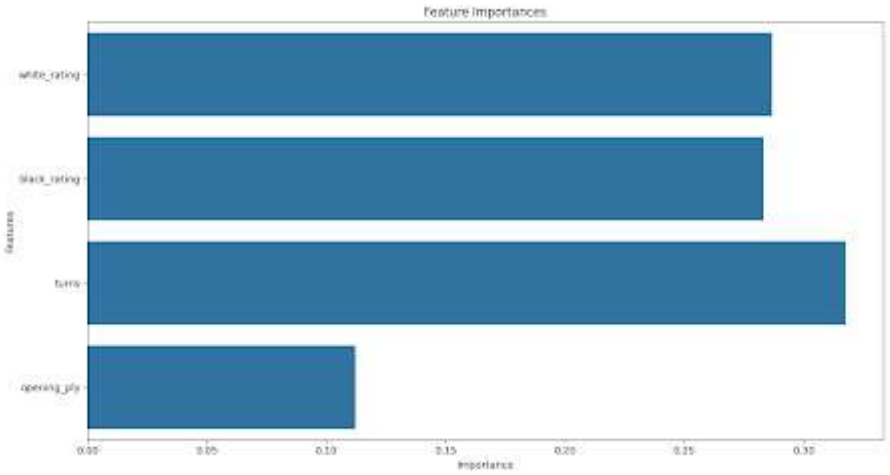
- Macro Average: Precision, recall, and F1-score were 0.57, 0.51, and 0.52 in the macro average. This metric treats each class equally, regardless of its frequency in the dataset. The lower values highlight the model's difficulty in predicting the minority "draw" class.
- Weighted Average: The weighted average, which accounts for the number of instances in each class, shows higher values for precision, recall, and F1-score (0.67 each). This metric better reflects the model's stronger performance with the majority classes.

Analysis of Model Performance

- Strengths: The Random Forest model performed well on the majority classes (black and white), with balanced precision, recall, and F1-scores. Additionally, the model was effective in identifying the most influential features for predicting the outcome.
- Challenges: The model was not good at performing with minority classes as seen with draw instances. This leads to class imbalance as draw is underrepresented compared to the instances where black or white won.

Visualizations

This visualization gives the relative importance of white_rating, black_rating, turns, and opening_ply in determining the win outcome



\

Next Steps for Model 3

Exploring Neural Networks:

Neural Networks, particularly deep learning architectures, could be applied to capture intricate patterns in chess games. Neural Networks can model non-linear relationships and complex interactions between features (such as player ratings, game length, and opening moves). A possible

approach could involve training a multi-layer perceptron or experimenting with a recurrent neural network (RNN) if we include sequential game data (Each move during the turn). Given the dataset's size, this model may require careful tuning to avoid overfitting, especially since neural networks are more prone to memorizing rather than generalizing from training data.

Applying Support Vector Machines (SVM):

SVM's are effective in high-dimensional and non-linear spaces, this approach could enhance classification, especially when tuned with the appropriate kernel (e.g., radial basis function for non-linearity). SVM is also suitable for handling imbalanced data, which could improve predictions for underrepresented classes like draws. While SVMs are computationally intensive, they are robust against overfitting and could provide a reliable baseline model for smaller, balanced data subsets.

Results and Discussion for Neural Networks

- Quantitative Metrics
  - Accuracy: The Neural Network model achieved an overall accuracy of 64%, correctly classifying the winner in 6,018 test cases. This indicates that the model is moderately effective in predicting game outcomes based on the input features.
  - Balanced Accuracy: The balanced accuracy of 0.72 shows that the model performs well across all classes, especially considering the imbalance in the dataset. This metric highlights the improvements in handling minority classes, such as draws.
  - Macro Average: The precision, recall, and F1-score for the macro average are 0.67, 0.72, and 0.69, respectively. These metrics treat all classes equally, regardless of their frequency, and reflect the model's strong generalization.
  - Weighted Average: The weighted average for precision, recall, and F1-score are all 0.64, which incorporates the support of each class and aligns closely with the overall accuracy.
- Classification Report

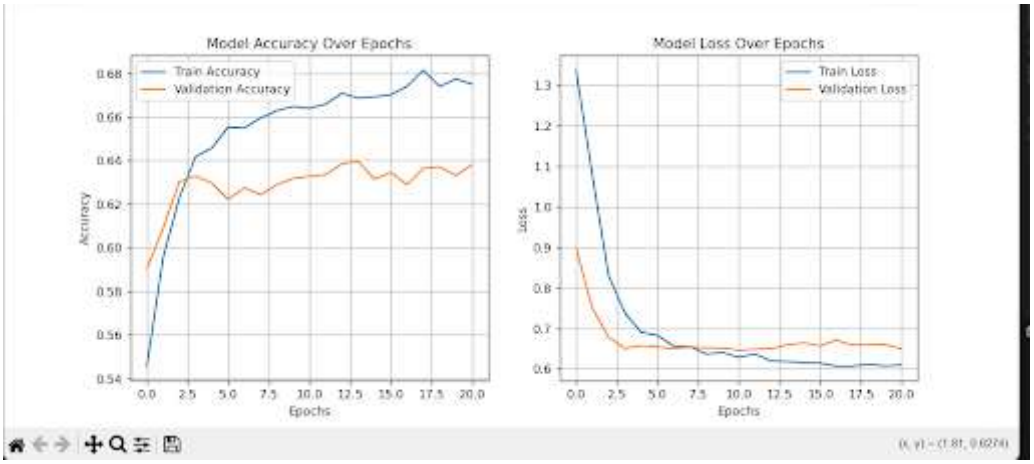
```
Classification Report:
              precision    recall  f1-score   support

       black       0.61      0.60      0.61      2719
        draw       0.75      0.91      0.82       274
       white       0.65      0.65      0.65      3025

    accuracy                           0.64      6018
   macro avg       0.67      0.72      0.69      6018
weighted avg       0.64      0.64      0.64      6018

Balanced Accuracy: 0.72
```

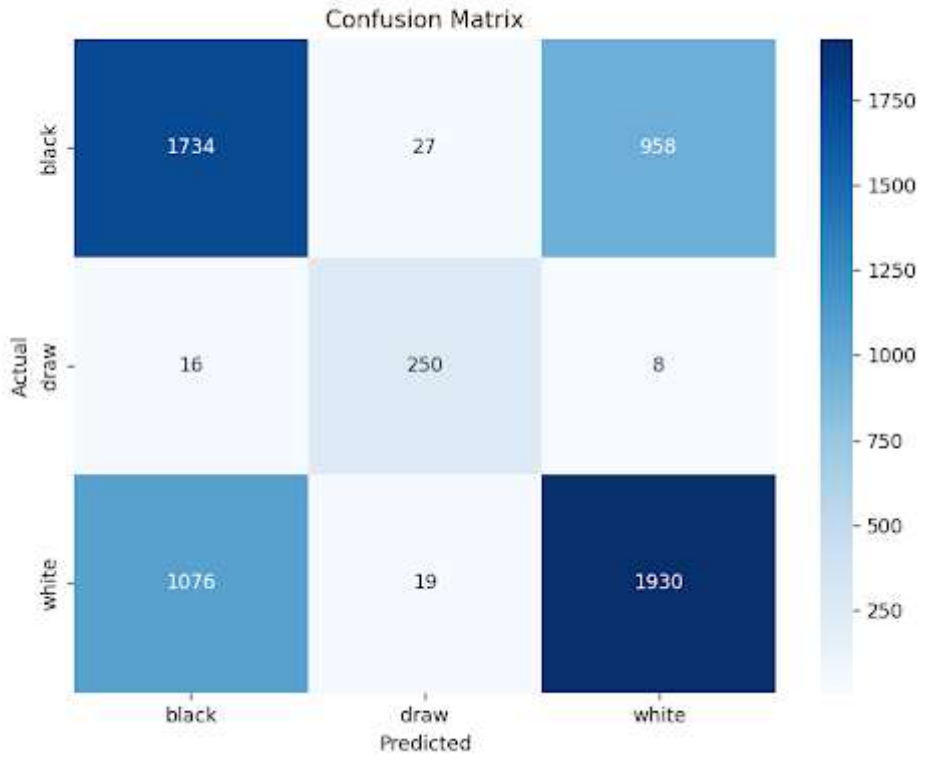
  - Black Class:
    - Precision: 0.61

        - Recall: 0.60
        - F1-Score: 0.61
          The model performed well in predicting outcomes where black won. However, the slightly lower recall indicates that some black wins were misclassified.
    - Draw Class:
        - Precision: 0.75
        - Recall: 0.91
        - F1-Score: 0.82
          The model excelled in identifying games that ended in a draw. The high recall suggests that the model correctly identified most draw, making it the strongest class prediction. This improvement is attributed to class weighting during training.
    - White Class:
        - Precision: 0.65
        - Recall: 0.65
        - F1-Score: 0.65
          The model demonstrated stable performance for the white class, with balanced precision and recall. However, the performance was slightly lower compared to the draw class.
- Analysis of Model Performance
  Strengths:
    - The Neural Network model effectively handled class imbalance, as evidenced by the strong performance in the minority draw class. Class weighting during training played a significant role in improving recall for this class.
    - The balanced accuracy of 0.72 demonstrates the model's ability to generalize across all classes, making it robust for this task.
    - The inclusion of features such as rating_diff and rating_ratio likely improved the model's ability to capture relationships between player ratings, leading to better predictions.

Challenges:

- The performance for the majority classes (black and white) is slightly lower than expected, particularly in recall. This indicates that some games were misclassified, likely due to overlapping patterns in the dataset.
- While the model achieved good overall accuracy, the black and white classes could benefit from further feature engineering or advanced architectures to better separate their patterns.
- The model may still struggle with overfitting on some features due to the relatively small dataset and complexity of the task.





Description of the Plots

- Model Accuracy Over Epochs
  - Train Accuracy: The blue line represents the model's accuracy on the training dataset over the course of the epochs. It shows a steady increase during the first 10 epochs and then stabilizes around 0.68, indicating that the model effectively learns from the training data.
  - Validation Accuracy: The orange line represents the model's performance on unseen validation data. It improves rapidly in the first few epochs before plateauing around 0.64, indicating that the model generalizes reasonably well but is limited by inherent data patterns or complexity.

Insight: The gap between training and validation accuracy is small, suggesting minimal overfitting. However, the plateau indicates that the model has likely reached its capacity under the current architecture and features.

- Model Loss Over Epochs
  - Train Loss: The blue line indicates the model's training loss, which decreases consistently and stabilizes around 0.6, showing that the model effectively minimizes its error during training.
  - Validation Loss: The orange line represents the loss on the validation dataset. It decreases quickly during the first 5 epochs before leveling off around 0.65, indicating that the model has reached its optimal point for validation data.

Insight: The parallel trends in training and validation loss suggest a well-regularized model with no significant overfitting. The stable validation loss reinforces the accuracy plot's conclusion that the model has reached its learning capacity for this configuration.

- Confusion Matrix
  - Black Class: The model correctly predicted 1,734 out of 2,719 black wins, with most errors misclassifying black as white (958 instances).
  - Draw Class: The model performed strongly, identifying 250 out of 274 draws correctly, with minimal misclassifications (16 as black and 8 as white).
  - White Class: For white wins, the model correctly predicted 1,930 out of 3,025 cases but often confused them with black wins (1,076 instances).

Insight: The model handles the minority draw class well but struggles to distinguish between black and white wins, likely due to overlapping patterns in features.

Next Steps for Model 3

Exploring Neural Networks:

Neural Networks, particularly deep learning architectures, could be applied to capture intricate patterns in chess games. Neural Networks can model non-linear relationships and complex interactions between features (such as player ratings, game length, and opening moves). A possible approach could involve training a multi-layer perceptron or experimenting with a recurrent neural network (RNN) if we include sequential game data (Each move during the turn). Given the dataset's size, this model may require careful tuning to avoid overfitting, especially since neural networks are more prone to memorizing rather than generalizing from training data.

Applying Support Vector Machines (SVM):

SVM's are effective in high-dimensional and non-linear spaces, this approach could enhance classification, especially when tuned with the appropriate kernel (e.g., radial basis function for non-linearity). SVM is also suitable for handling imbalanced data, which could improve predictions for underrepresented classes like draws. While SVMs are computationally intensive, they are robust against overfitting and could provide a reliable baseline model for smaller, balanced data subsets.

Conclusion

For our project the objective was to be able to predict the winner of a chess game (Black, White, or Draw) based on player ratings and opening moves. To do this we analyzed a comprehensive dataset composed of chess games, we focused on key factors including first-move advantage, opening strategies, player ratings, etc. We employed three machine learning models and evaluated them: Gradient Boosting Classifier, Random Forest Classifier, and a Neural Network.

Summary of Findings

Data Analysis and Preprocessing

- Color Advantage: As consistent with historical trends, games where white moved first resulted in a higher win rate. However, when included in predictive models, this advantage did not notably address enhancing prediction accuracy due to its relatively small impact compared to other features.
- Opening Moves: With 365 unique opening moves in the dataset, some opening moves did show higher win probabilities. Still, the impact of specific openings on predicting was limited, partly due to the varying frequency of each opening's occurrences.
- Rating Difference: Our most significant predictor was the absolute difference in player ratings. A logistic regression analysis showed that as the rating difference increases, the higher-rated player's chance of winning rises rapidly before eventually plateauing, showing a diminishing return after a certain skill difference.

Model Performance Comparison

- Gradient Boosting Classifier
  - Accuracy: Achieved the highest overall accuracy being 84%
  - Balanced Accuracy: Found 61%, reflecting challenges in predicting the less frequent outcomes (Draws).
  - Strengths: Did exceptional at predicting wins for the majority classes (black and white), with high precision (0.84 for Black and 0.86 for white) and recall

    - Challenges: Struggled with the minority class (draws), evidenced by the low recall rate at 8% for draws even though techniques like SMOTE were used to address the class imbalance.
- Random Forest Classifier
    - Accuracy: Overall accuracy of 68%
    - Strengths: Did well at predicting the majority classes, having precision scores of 0.67 for black and 0.70 for white.
    - Challenges: Like gradient boosting, it had difficulty accurately predicting draws, having a low recall rate of 11% for the draw class.
- Neural Network
    - Accuracy: Reached an overall accuracy of 64%
    - Balanced Accuracy: This was notably higher at 72%, showing better performance across all classes.
    - Strengths: It effectively handled class imbalance. Excelling in its ability to predict draws, getting a precision of 75% and a recall of 91% for the draw class.
    - Challenges: The model's performance for the majority classes was slightly lower being 61-65% for Black and White. Occasionally misclassifying Black and White wins, suggesting overlapping patterns in this data.

Conclusion and Recommendations

Best Performing Model:

The Gradient Boosting Classifier was the top performer in overall accuracy (84%) and it was especially strong in predicting outcomes where either black or white wins, but its performance on predicting draws was limited due to class imbalance.

Neural Network's Strength in Predicting Draws:

While our Neural Network had a lower overall accuracy, its high balanced accuracy at 72% and great performance when predicting draws make it a valuable model when the accurate prediction of all classes is crucial.

Final Thoughts

Our project was able to highlight the complexities in predicting chess game outcomes. While we saw that player ratings and openings are significant indicators, the complicated nature of chess

requires models that are able to handle imbalance and capture the subtle patterns in the data. We recommend Gradient Boosting Classifier for scenarios when overall accuracy and when there is importance placed on all outcomes, then Neural Network is preferable.

Next Steps

Data Augmentation: We could acquire more data, particularly for underrepresented classes like draws, to provide a more balanced training set for the models. Further Address Class Imbalance: Implement advanced techniques possibly like adjusted class weights to improve draw predictions in models like Gradient Boosting and Random Forest.

Contribution Table

Name	Tasks Completed
Nithil	Introduction/Background and Problem Definition, Completed Random Forest and Gradient Boosting.
Anay	Methods and Result + Discussion, Completed Random Forest Model and Gradient Boosting.
Randall	Methods and Problem Definition, Completed Gradient Boosting and Random Forest.
Apollo	Github Pages / Result + Discussion, Data Preprocessing / Github Pages / Next Steps
Drew	References + Final Review, Data Preprocessing / Data visualizations / Data Analysis

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

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION
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
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

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION
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