# Group 10 Project Report

**Qirui Zheng A17317403**

**Samantha Lin A16758004**

**William Kam A16940420**

**Aryan Kanuparti A18093864**

**Derek Klopstein A17008856**

**Daniel Gitelman A16861627**

**Editor:**

## Abstract

Chess is an old game played by many people worldwide. One thing that is unequivocally true about chess is that it is really hard to master. As such, being able to provide quick dos and don'ts for beginners in chess helps make it more accessible. Therefore, we intend to evaluate the moves in chess from a hypothesis test point of view and see the importance of making some moves over others. In this article, we attempted to analyze various features of a chess dataset to get a better idea of what differentiates chess players across all levels. These include: how the opening moves performed by black or white correlates to win result, the frequency of different moves and their correlation to skill level, and what move or sequences of moves cause players to resign versus resulting in them being checkmated. Leveraging these findings, we trained various models to explore whether patterns that show whether a player is going to win based off their opening moves are learnable.

**Keywords:** chess, hypothesis testing, decision trees, neural network, deep learning

## 1 Introduction

Chess is a board game played between two players. A chess board consists of 64 squares, 8 rows by 8 columns. Looking at the board from the white side, the columns are labeled characters "A" to "H" left to right, and rows are labeled from 1 to 8 bottom to top. Each player has 16 pieces that fill up their first two rows, 1-2 for white and 7-8 for black. These consist of 1 king, 1 queen, 2 rooks, 2 bishops, 2 knights, and 8 pawns for each player. A player wins if they can checkmate the other king. A king is in check if an opposing piece can take the king on their turn, so the king must move to a square that is not threatened by the opponent's pieces. Checkmate is found if the king has no "safe" space to move to.

Figure 1: Chess Board

### 1.1 Dataset

The data we used in this article is from an online chess game platform called Lichess. The data collected from the games is available on Kaggle (https://www.kaggle.com/datasnaek/chess). Some key features of this dataset are the rating of each player, all of the moves from both players in a match, the total number of moves in the match, and the name of the opening move by each player.

## 2 Prior Work

There is a similar analyses has already performed in Predicting Chess Opening Through Modelling Of Chess by Debarpan Bose Chowdhury and Banashree Sen (1).The authors then discuss various factors that influence a chess move, including the current position, personal preferences of players, average turn, memory retention, and transposition. They propose an algorithm that combines these factors to predict the opening moves of players. The algorithm involves analyzing probabilities of moves appearing in games, calculating the percentage of retention of each move over time, and relating average turn to move probabilities. In this analyses the authors also utilized data using the lichess database. Their algorithm consistent of finding the probability of a move appearing in a game, find the percentage of retention of each move using Ebbinghaus' forgetting curve, calculating P1‘ and P2‘ using the distribution curve then using P1‘ and P2‘ to predict moves. When trying to predict the first five moves of a game the authors' model achieved an accuracy of 54 but decrease as the number of moves to be predicted increases. Based of their model the authors' believe that their model would allow players the ability to predict their opponent's moves. This study is a bit similar to ours but differs in the fact that we are trying to predict the winners based off the opening moves not the opening moves themselves.

## 3 Analysis performed for the Project

To start our analysis we first began by data cleaning. We decided to pick which columns were useful first and also look for missing values and found none in our data set. We then one hot encoded opening play and feature engineered first ten moves by only getting the first ten moves from the moves column. Our features ended up being white rating, black rating, turns, victory status, moves, opening eco, opening name, and opening play where we tried to predict the winner.

After deciding on our preliminary features we then decided to create a pairplot to visualize relationships between our features and the winner column. In our pairplot we found distributions of how are data looks like plotted against our target variable but couldn't find much insight so we decided to create histograms for each feature variable.

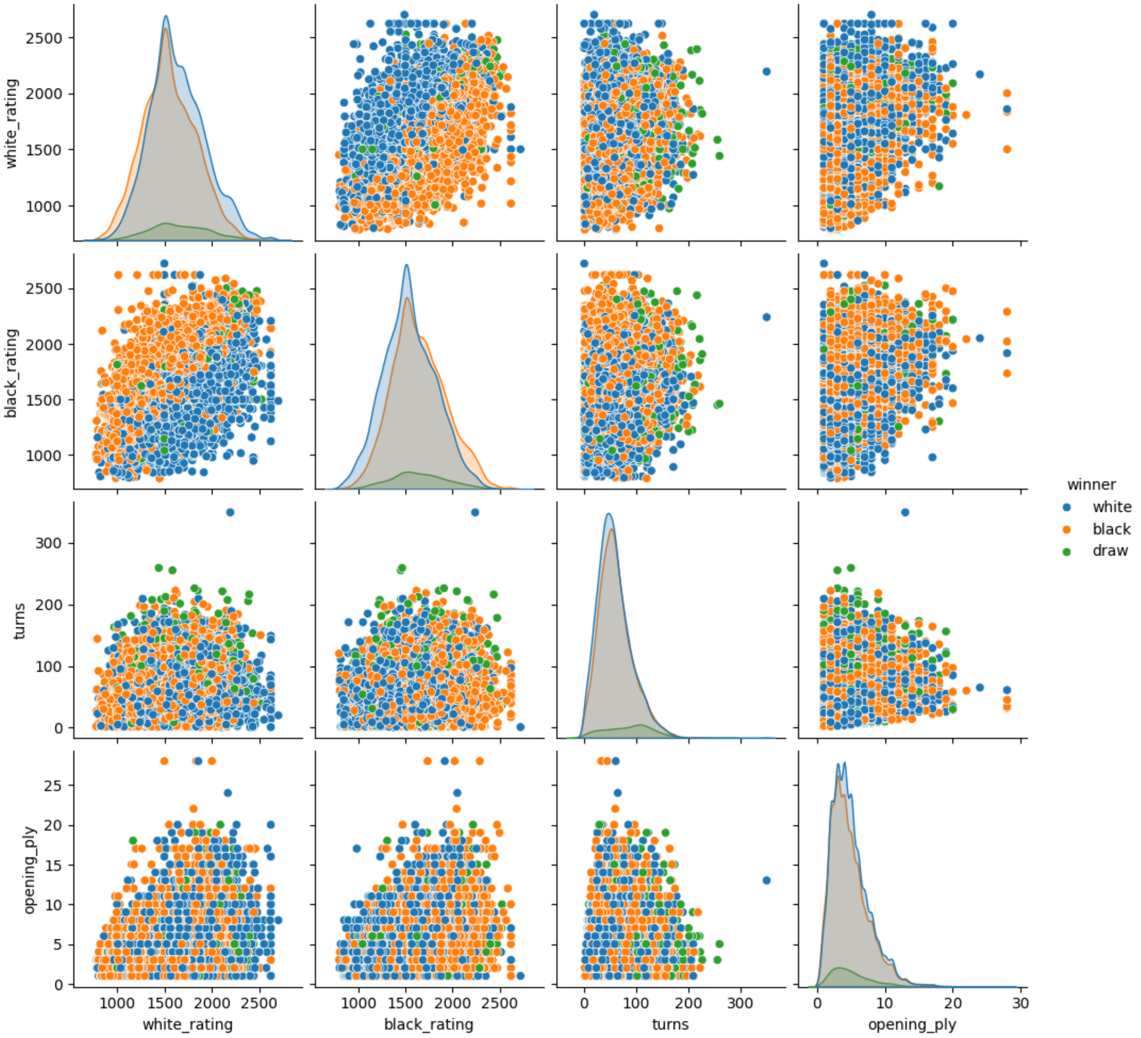
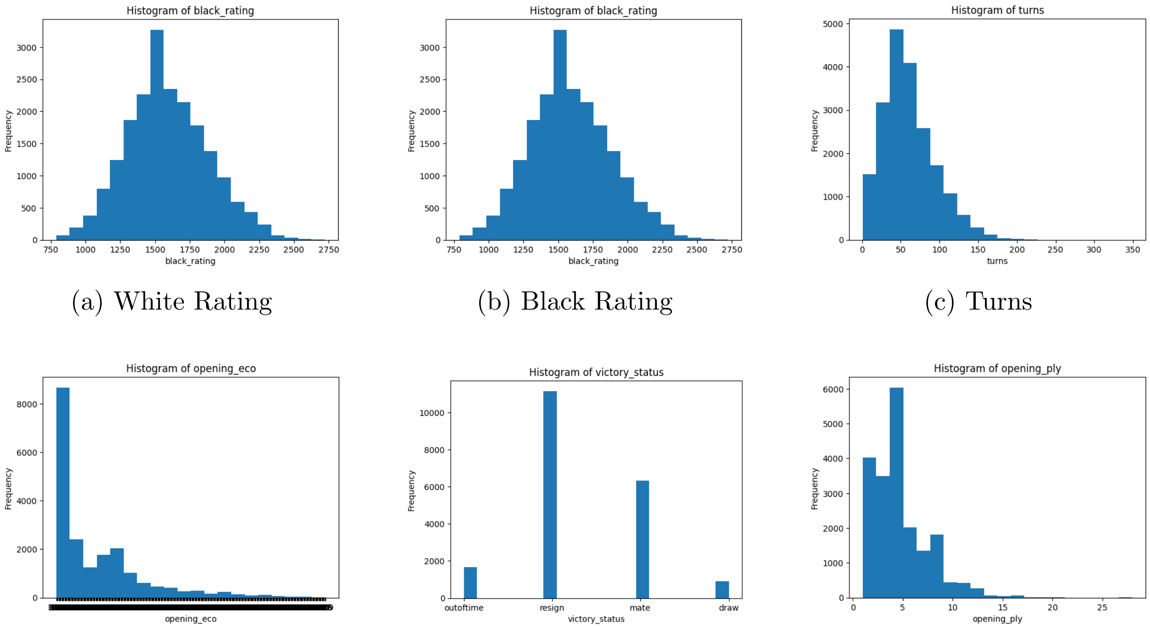


Figure 2: Pairplots

Our histograms of black rating and white rating show that they both have a uniform distribution. For our turns column the most likely number of turns per game is fifty. Victory status indicates how a player won and the histogram indicates that most players resign. Both opening eco and opening name indicate that one opening eco move and name has been utilized in this data much more frequently than any openings. Lastly our opening play indicates the number of moves in the opening phase and the most frequent number is between four and five.



(a) White Rating

(b) Black Rating

(c) Turns

(d) Opening Eco

(e) Victory Status

(f) Opening Play

Figure 3: Histograms

After gaining some more knowledge and understanding of our dataset we continued by performing some preliminary hypothesis tests. Firstly, we want to analyze the relationship between the color that a player is playing and the win result/overall rating. To do so we decided to perform a t-test for Independence. This test compares the means of two independent groups to determine if there is a statistically significant difference between them. In our case, we can move forward with the following hypothesis:

- Null Hypothesis ($H_0$): There is no significant difference in ratings between white and black players.

- Alternate Hypothesis ($H_a$): There is a significant difference in ratings between white and black players.

We obtained quite a large P-value and thus failed to reject the null hypothesis. In other words, we found no significant difference in ratings between white and black players. Intuitively this aligns the logic that all players play black or white and never only one color throughout their games. Despite this, we wanted to be thorough to ensure that playing as either black or white does not entail advantages or disadvantages in terms of victory.

Next, we took a look at whether or not the opening move has a significant impact on the victory of the game. To do this we can perform a Chi-square Test of Independence to determine whether there is a significant association between these two categorical variables: opening move and win result.

- Null Hypothesis ($H_0$): The opening move made does not significantly affect victory status.
- Alternate Hypothesis ($H_a$): The opening move made does significantly affect victory status.

This test yielded some statistically significant results with a P-value close to zero. We can safely say that the opening move has a relationship with who the victor of the game will be. This is interesting to note and begs the question, what are the best opening moves that increase the likelihood of victory?

Following some analysis, we produced this heatmap that displays the ten most frequent opening moves and their associated victory status outcome percentages.

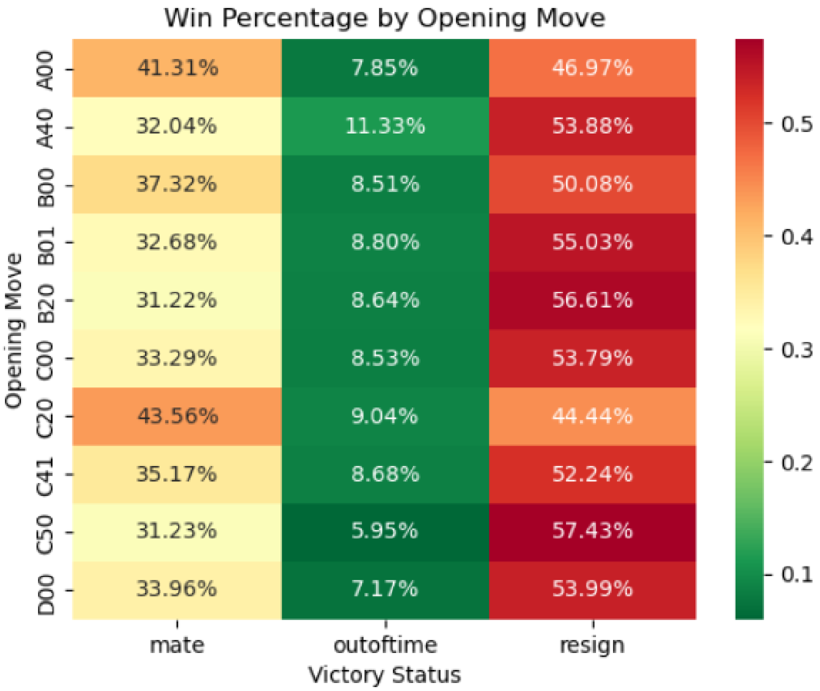


Figure 4: Heatmap of 10 Most Frequent Opening Movies

From here, we want to find out the differences between these top ten move distributions in the hopes of potentially finding an objectively superior first move. To do so, we perform the Kruskal-Wallis non-parametric test that can be used to determine whether there are statistically significant differences between the distributions of three or more independent groups. In this case the groups would be the ten most popular opening moves seen in the figure above,

- Null Hypothesis ($H_0$): There is no significant difference in win percentages among the top 10 opening moves.
- Alternate Hypothesis ($H_a$): There is a significant difference in win percentages among the top 10 opening moves.

This test did not yield statistically significant results, with a fairly high P-value. This leads us to believe that there is no clear-cut best first move to make that boasts a significantly higher win percentage than the others. Ultimately, this makes sense considering the nature of countless possible outcomes in chess. However, as of right now, the only thing we can conclude with certainty is that there is the opening move made is correlated to the result of the game, unlike player color.

## 4 Modeling: Baseline (Decision Tree)

With our main question in predicting the winner of a chess game, we utilize a baseline model that can be used to evaluate later model performances. Based on heuristic, the most important factor that directly lead to winning or losing a chess game is the sequence of steps taken. Thus, only the moves column is used, to avoided the highly variate moves taken towards the end of the game only the first 10 moves is considered, and each step is separated into their individual features. The winning class is binary encoded as 1 stands for black as the winner, 0 for white. The end resulting data to train upon is 10 feature each containing a step. The decision tree split is evaluated using the Gini criterion

$$Gini(split) = \sum_{i=1}^{n} p_i \cdot (1 - p_i).$$

Where $p_i$ is the probability of being in the positive class in this case, black as the winner. There was significant overfitting observed as the test accuracy achieved 0.94, while test accuracy is only 0.59. The hyper parameter of tree depth is then fine tuned by using grid search with the most optional depth of 70. The model is then trained again yielding a test accuracy of 0.58 and train accuracy of 0.74.

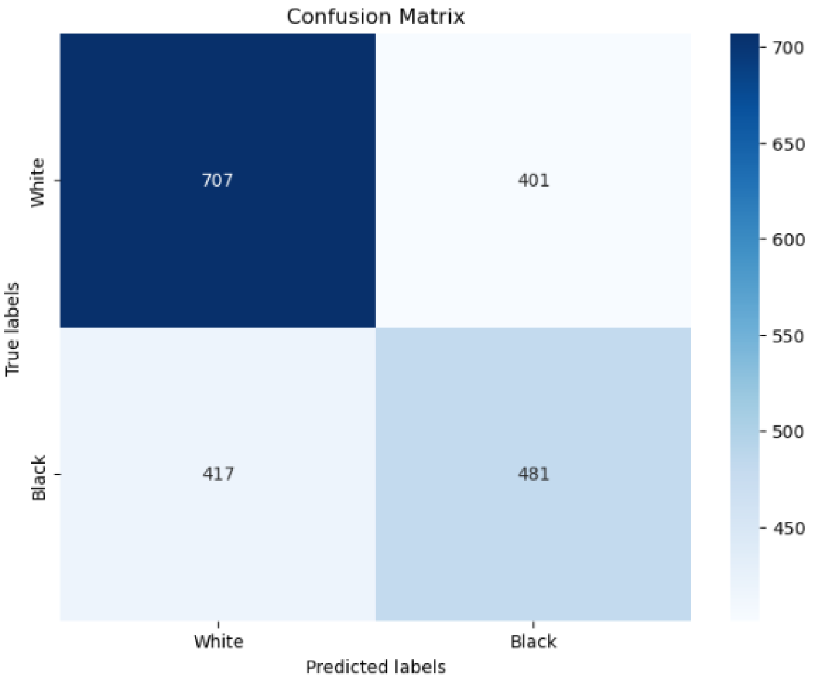


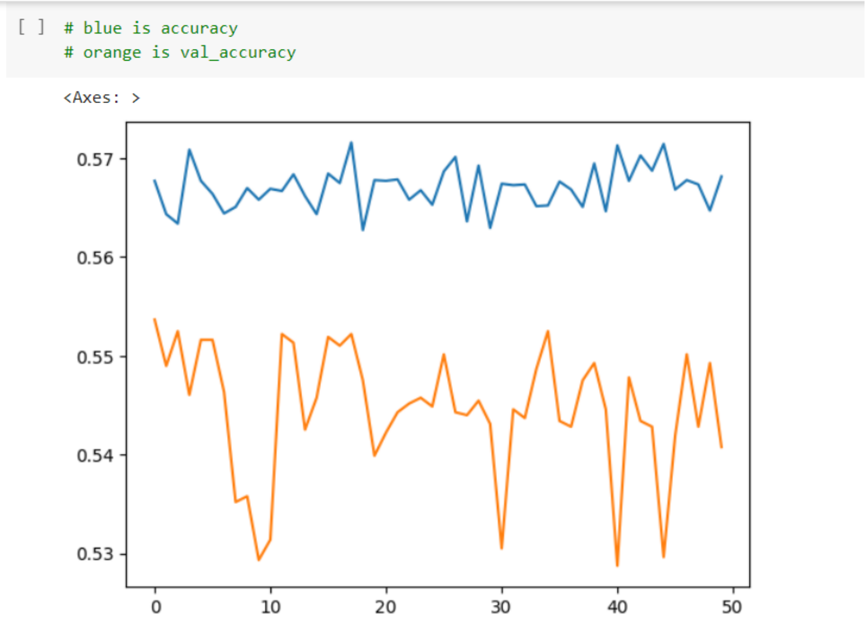
Figure 5: Confusion Matrix

As a single decision tree is prone to overfitting, we used the ensemble method of a random forest classifier where multiple decision trees are fitting to the data and take the majority vote from all the different decision tress. An increase in accuracy to 0.60 is found, however high training accuracy is also observed to be 0.90. The factor of overfitting heavily effects model performance. Since this is just our baseline model, we later used state of the art modeling method to improve upon the actual model performance.

## 5 Modeling: Neural Network

We performed a neural network on our dataset without the first ten move column since we couldn't include it in our model. We've chosen to build the model with 1 input layer, 5 hidden layers (nodes: 35,28,20,12,4), and 1 output layer. We've also chosen sigmoid functions as our activation functions for all layers because we're doing a binary classification task. Finally, we fitted our model using a validation split of 0.2 and ran it on 50 epochs.

However, after running the model, we can see that our accuracy wasn't performing better than the baseline model (decision tree). We were only able to get a 0.57 accuracy in our training data, and a 0.54 accuracy in our validation data. After seeing the results of our accuracy we checked if there was a sign of an overfitting in our model, so we plotted a graph to see where the best fit would occur.



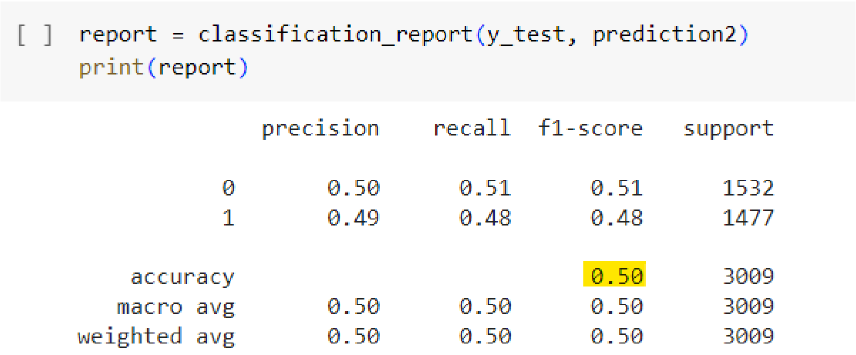
```
[ ] # blue is accuracy
    # orange is val_accuracy

<Axes: >
```

Figure 6: Training and Validation Loss Graph

This plot shows that there is no sign of over fitting since the two lines of accuracy never touched each other, and the validation accuracy is below the training accuracy.

We then further investigated in a tuner to try to find a better neural network that might work for this task. However, after we ran the model, we found out that the accuracy of this model performed even worse, with an accuracy of only 0.5.



```
[ ] report = classification_report(y_test, prediction2)
    print(report)

              precision    recall  f1-score   support

           0       0.50      0.51      0.51      1532
           1       0.49      0.48      0.48      1477

    accuracy                           0.50      3009
   macro avg       0.50      0.50      0.50      3009
weighted avg       0.50      0.50      0.50      3009
```

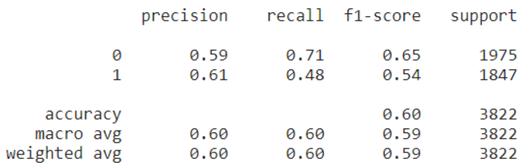
Figure 7: Classification Metrics

That being said, we've concluded that a neural network might not be a good way to approach this task, and we might need some deep learning in order to find some patterns within our chess matches.

## 6 Modeling: Transformer

Because the neural network failed to accurately learn to predict the winner from opening moves, we wanted to test out an additional deep learning approach, transformers. Transformers are most commonly used for language modeling, and use attention, which allows these models to learn the relationships any given word has to each other word in the input. The opening moves of a chess game, can be represented as a sentence consisting of the moves in standard chess notation, so we hypothesized we might be able to apply this model to learn how to predict a winner from opening moves. In addition, previous studies, (3) have found that an attention based language model can learn to play chess, so it seems feasible that an attention based model could also learn to predict the winner from opening moves.

We chose to use Google's pre-trained Bidirectional Encoder Representations from Transformers (BERT) (2) as a base language model to learn representations of the opening moves, which are then fed into a linear layer to classify the sequence as winning or losing. We then fine tuned the model with our dataset, achieving an accuracy of 60.2% on the dataset, just slightly short of the baseline decision tree. It achieves a similar score in macro average recall and precision. The detailed performance can be viewed below, in Figure 8.



```
              precision    recall  f1-score   support

           0       0.59      0.71      0.65      1975
           1       0.61      0.48      0.54      1847

    accuracy                           0.60      3822
   macro avg       0.60      0.60      0.59      3822
weighted avg       0.60      0.60      0.59      3822
```

Figure 8: Metrics of BERT on Validation Data

Observing the predictions of the BERT model by confusion matrix (9), it is clear that similarly to the predictions of the decision tree, the model most often correctly predicts the winner of games white won more often. The model performed poorly on games where black won, also similar to the decision tree. This suggests a pattern in the data that the machine learning models are picking up on.

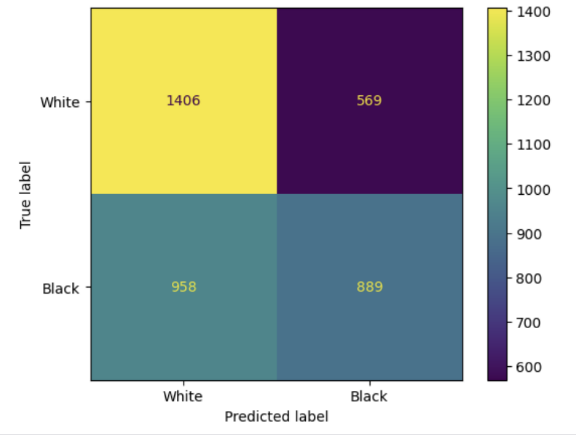


Figure 9: Confusion Matrix of BERT on Validation Data

## 7 Discussion

From the results of analysis we can uncover several different interesting insights. First, regardless of any other factors involved in a chess game, we were able to find that the opening moves of a game have a statistically significant impact on the rest of the data.

We also uncovered unexpected results from the models trained to predict the winners. While neural networks did poorly, no better then guessing at random, both transformers and decision trees performed almost equally as well at predicting the winner. They both achieved statistical significance over what the mean correct prediction rate would be given the models were guessing at random. This shows that even without any external factors, such as the rating of different players, patterns in opening moves can still be learned by machine learning models to predict winners.

Below, in Table 1, we include a comparison of the models on the validation sets and the metrics they achieved. Clearly, decision trees and transformers are the standouts over neural networks for this kind of modeling. This shows that deep learning, as well as classical machine learning methods, can work to model complex data, like chess moves. Interestingly, both of the standout models predicted the winner of games where white won far more effectively then ones where black won. This suggests that white is often able to gain more decisive leads in the opening moves that result in victory.

| Metric | Decision Tree | Neural Network | Transformer |
|---|---|---|---|
| Accuracy | .59 | .50 | .60 |
| Macro Average Precision | .59 | .50 | .60 |
| Macro Average Recall | .59 | .50 | .60 |

Table 1: Model Performance Comparison

## 8 Conclusion

Chess is undoubtedly a very complex and intricate game and we definitely have a deeper understanding of it after completing hypothesis testing and machine learning research. Through our exploration, we discovered a statistically significant correlation between the opening moves in a chess game and its eventual outcome. Certain initial moves displayed a stronger association with victory, indicating their strategic value. However, our analysis revealed that there isn't a definitive 'best' opening move that ensures success. Furthermore, we didn't find a clear-cut best opening move that guarantees success, highlighting the complexity and variability of chess itself.

We also found that transformer based language models and decision trees can serve to predict winners based on opening moves with statistical significance, and about the same accuracy. However, the Google's pre-trained model was pre-trained to understand natural language, not chess notation, and we had a lack of computational power and data needed to train our own version that was. For future work, the effectiveness of this architecture could be explored when built with chess notation specifically in mind.

In addition, there still remains the possibility of exploring why the successful models often predicted games where white won correctly more often then games where black won. Analysis could be performed on what moves players playing white make that result in a discernible lead to victory, providing valuable insight.

In summary, our project illuminates the significance of various aspects of chess. Moving forward, we hope to refine our methodologies and explore new avenues to deepen our understanding of chess strategy and the intricate game play dynamics through statistical analysis and ML modeling.

## References

[1] Debarpan Bose Chowdhury and Banashree Sen. Predicting chess opening through modelling of chess opponents. *Webology*, 18(6):6748–6757, 2021.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[3] Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Mguni, Yali Du, and Jun Wang. Chessgpt: Bridging policy learning and language modeling. 2023.