BradleyChess

A Capstone Proposal
Presented to the
Faculty of CST 499 at
California State University, Monterey Bay

By
Abraham Borg, Sarom Thin and Mehar Rekhi
Fall 2022

EXECUTIVE SUMMARY OF PROPOSAL

Bradley Chess Interactive Dashboard to Predict Best Moves

By

Abraham Borg, Sarom Thin and Mehar Rekhi

Bachelor of Science in Computer Science

California State University Monterey Bay, 2022

Chess is a highly complex game. However, specific trends and playstyles become dominant and repeatedly appear in play throughout different periods. These trends are things like popular openings. There is a lot of data that can be gleaned from every chess game, and we can analyze this data and make predictions about the outcome of a game. The team will use Dash and Plotly to create a chess analysis dashboard web application. The dashboard web app will present an analysis of several million chess games. It will show charts and figures that describe the playstyle of grandmasters through the years, starting from the 1970s to the present day. The dashboard will also feature an interactive chess board that will have an indicator showing the best move.

The chess data will be used to train a model to predict the best move in response to an opponent’s current move. This recommendation will be based on several parameters. For example, the program will consider the current pieces left on the chess board. This project will benefit the chess and data science enthusiasts and players at the Fresno Chess Club. These players will be able to analyze their games and other players' games at the club. The analysis is essential to players because it will allow them to offload the analysis to the computer instead of their minds. They will see the trends in their playstyle and use that information to improve their chess skills.

Table of Contents

PAGE

LIST OF FIGURES

PART I: BACKGROUND AND APPROACH

INTRODUCTION

PROJECT NAME AND DESCRIPTION

The name of the project is BradleyChess. The target audience is chess players, data science enthusiasts, and members of the Fresno Chess Club. The team will create a web application with an interactive dashboard that will provide an analysis of chess games. End users can play a chess game and observe game records gathered since the 1970s, where millions of games have been recorded up until now.

Analysis of millions of chess games reveals patterns in play that can be utilized to predict the best chess move. For example, our dataset has many openings for White and Black. However, the total number of typical opening moves for White is much smaller compared to Black. It must be due to the nature of the game since White always moves first; therefore, Black usually plays reactively. As mentioned before, users of the BradleyChess tool will be able to play a game. During the process, they can see the next best possible move. In addition, there will be graphs available to view specific statistics during the game. The dashboard will also have a section where users can select what analysis they wish to view. In addition, the user can enter their games into the database.

Allowing the player to add their games to the database will be helpful for chess players to compare their games with expected trends - this will be useful for data scientists as they can observe how raw data is transformed into comprehensible information. Finally, testing will be done by the Fresno Chess Club.

ISSUE: COMPLEX DATA RECORDS AND LACK OF SIMPLE ANALYSIS TOOLS

The dataset that will be used contains millions of records. However, this raw data contains overwhelming information that can be difficult to read. There may also exist irrelevant information that would not contribute to any meaningful analysis, which can confuse observation. There also lacks an analysis tool that matches this team's project vision. The analysis tools available are complicated machine learning algorithms and complex chess engines. The team seeks to create something much more straightforward and based on probabilities of winning rather than deep analysis and logic.

SOLUTION: SIMPLIFYING DATA AND MACHINE LEARNING

To tackle these issues. The team has decided to use Python as the choice of language. The reasoning is that this programming language has libraries specifically for data preparation, creating visualizations, and machine learning. Examples include Pandas and SciKit-Learn. Pandas is a specialized dictionary that can assist in reorganizing and minimizing raw data to query information more efficiently. SciKit-Learn is the tool that trains a system to make chess move predictions. There are different methods to train a system, and for this case, the team will evaluate kNN classification and other algorithms like decision trees to identify the best model. There are also full-stack frameworks that can work seamlessly with Python. For example, Plotly will be used to create beautiful graphs, and Dash can inject visualization interfaces on the frontend page.

EVIDENCE

The rise of chess tournaments worldwide has been increasing over the years. Of course, before the tournaments, people would have to practice their skills beforehand. With our project, the user will be able to gain access to data records of previous games, analyze different moves, and learn more from the past. Using this tool will help the user decide the different amount of

moves they can have in their arsenal, whether it be at the start of the game or the end. This tool will aid those needing an extra edge over their future opponents. This program is necessary because the average player cannot analyze millions of games. There is too much data, and it is difficult to access that data in any meaningful way. There are simple things that can be synthesized by our program that will be immediately useful. These data points are things like the most common openings per decade. Another useful metric is how often the player with the most pieces left at the end game won. Many more data metrics and visualizations will be available to the application user.

PROJECT GOALS AND OBJECTIVES

GOALS

The goals of this project are to:

- Have users compare their games with current trends.
- Users are able to recollect different information about various games through the years.
- Users can add their game data and store it.
- Aid the user in deciding which is the best move in response to the opponent's move.

OBJECTIVES

- Simplify raw data by using Pandas. Only extract information that will be useful for analysis.
- Use SciKit-Learn to train a system in predicting the next best move.
- Create graphs using the Plotly library and include those visualizations on the dashboard.
- Inject data and visualization into the frontend using Dash.
- Create an interactive chess board on the frontend. The board will change based on user inputs as the game progresses.
- Moves will be suggested to the user. These chess moves are what the the machine learning algorithm has found will be *most likely* to win the game.
- Users will be able to add their game to the dataset.

ENVIRONMENTAL SCAN

This project is based on the foundational work completed by Abraham, Mehar, and Sarom in the data science course, https://github.com/abecsumb/DataScienceProject. In that project, the objective was to predict whether a player would win the game based on parameters like the openings and the ELO rating of each player. The model used for this was kNN classification, and this model predicted the correct outcome about 70% of the time.

The team previously used a simple machine learning approach similar to this project, https://www.kaggle.com/code/andriizelenko/chess-classification. Our model was almost 15% better at predicting a win than the model shown in the link. Also, there are many chess engines and AI projects centered around chess prediction and analysis. One example is this, https://tinyurl.com/2z2pp4as and this https://www.chess.com/forum/view/general/machine-learning-in-chess. The chess engines are very complicated, and so are some machine learning models already out there. This project will be different because we don't seek to develop an excellent chess engine. What we will make is a chess prediction program. The logic and best analysis are outsourced to the best players and chess engines. Our program will take all the available chess data and suggest the move most likely to win the game. This approach is like the greedy method; we make the best move at each step based on available data and analysis. Any algorithms and logic are limited to checking if a suggested chess move is legal and other basic checks. The chess meta (the best-understood way to play the game) has changed over the years, so this chess program will consider that when making predictions. The chess program will also consider the ELO ratings of each player. Based on our research, no other program is available that uses a purely statistical approach to play the game. Other chess programs are very complicated, and other chess machine learning programs

try to do too much and are also very complex. Our chess program will prioritize simplicity and limit the metrics used for predictions to the most crucial information. Finally, it bears repeating that this program will seek to make the best decision at each move based on current legal moves and their likelihood to win based on historical data and analysis of chess games. In other words, our program will not plan and analyze future positions. This simple approach is what sets our project apart.

PART II: SOCIAL AND LEGAL CONSIDERATIONS

STAKEHOLDERS

The Fresno Chess Club are the stakeholders for this project. This group will have a web application that will help them improve their chess skills. There are many metrics and visualizations in chess that can help players and BradleyChess will give the members of the Fresno Chess Club an interactive data dashboard that can generate visualizations and other data based on user input. The application will also use machine learning to suggest the best moves. This approach to playing chess will open up new possibilities and an unorthodox way to play the game. The reason for this is because other chess programs try to win the game through powerful algorithms and so the play is familiar to the top players (the top players in the world study the playstyles of the best chess engines - like Stockfish). BradleyChess seeks to give the player the best advice, and that can lead to games that haven’t been seen before. The chess club would lose the ability to explore millions of games through an interactive dashboard, and the ability to generate their desired visualizations. And most of all, a new way to think about the game. Chess should not be about trying to win and place an efficient game, we believe that the point of chess is to play a beautiful game.

ETHICAL CONSIDERATIONS

Automation is growing everyday, and with the emergence of machine learning these systems are gaining the ability to learn like humans. They are able to learn at an extremely fast rate as well. Automation has been an ethical concern due to job displacement. As repetitive tasks are seen to be taken over by systems, leading to certain jobs becoming obsolete. Machine learning adds to this concern because this could mean that systems can expand to other jobs that are not just repetitive. This issue can also extend to chess. There was an event where the greatest chess player of all time was defeated by a computer. Also, in these past few decades professional chess players have nearly imitated opening strategies created by computers, which resulted in other players minimizing new creative strategies.

These are the perceived disadvantages. However, introducing artificial intelligence has actually brought some good, and the Bradley Chess application aims to take those benefits and amplify it. The goal of the application has always been to create an interactive dashboard for players to easily observe characteristics of a game. It is not to seek out in creating a system that is unbeatable in a game of chess. Afterall, the target audience is for any enthusiasts of chess or data science. So, technically speaking it is for everyone of all levels. Chess has been known to be a complex game, this application will use machine learning to do the hard thinking first. Then process that information to make it digestible and easy for newcomers. Also, thanks to machine learning the application can recreate high-level plays anywhere, so the experience doesn't have to be restricted to tournaments. So, with these results it will actually help a community grow, and not get displaced by systems.

LEGAL CONSIDERATIONS

A feature of the application will be to insert game results into a database. This means that a login will most likely be used to keep track of users. Thus data privacy needs to be considered. Since a small team will be engineering this project. The handling of credentials will be outsourced. The plan is to use OAuth2 to manage login. OAuth2 is actually a popular standard that uses another website such as google, facebook and others to access data from another hosting website. That other host could be the Bradley Chess application. When using OAuth2, the aforementioned websites will create an access token for users that will serve as a password for the host website. This means that the host will never know the true passwords or personal information of the user. These companies have a reputation for having robust security as well.

Another concern that this application may face is copyright. The dataset that the team will start out with is from an open source project licensed by MIT. So there shouldn't be any legal concern in the beginning. However, users will eventually be able to push their game to the database. There may be professional players who do not want to share their private games. From research there isn't any copyright law that players can claim for a match. That being said, we want to respect users. So there will be options to save a game as a private match and not be shown to the public.

A final concern that this application may run into is cheating. This application will have the feature to guess the next best move. Having the ability to predict may lead others to use the tool for malicious purposes. For example, gaining unfair advantages in an online tournament. Tournaments may have high rewards on the line, and of course require every participant to follow the rules to ensure fairness. However, if there are any fraudulent players caught in action then the tournament organizer can take legal actions to get compensation. Unfortunately,

cheating can't be prevented as the Bradley Chess is just a tool that people can use freely. Fortunately, tournaments understand that it's not the fault of the chess engines, but the fault of users if they decide to cheat. That is why online tournaments have developed anti-cheating systems themselves to prevent such malicious use.

PART III: PROJECT SCOPE

TIMELINE/BUDGET

DETAILED TIMELINE

Date	Description
Tuesday, 10/25: Meeting with Fresno chess club	Describe the basic functionality and get feedback on features they'd like to see.
Wednesday, 10/26: LA Tech Professionals networking	Networking event for computer science professionals. Discuss project and get feedback.
Thursday, 10/27: San Diego Python Meetup	Get feedback on SARSA Python code. Discuss project.
Thursday, 10/27: Fresno DS Meeup (virtual meeting)	Give BradleyChess project status report and get feedback.
Saturday, 10/29: San Diego Data Science Competition/Collaboration	Data science professionals discuss their projects and any competitions they've entered and collaborate and share advice.
Sunday, 10/30: Team meeting	What is the status of the dashboard? Is the pseudocode of the SARSA algorithm implementation complete?
Wednesday, 11/02: React workshop	The interactive chessboard that will be part of the Plotly Dash dashboard will be implemented using React. We can get tips on how to do that (we already have a pretty good idea on how to do it).
Wednesday, 11/02: Week 1 Milestone items due	see milestone table

Date	Description
Thursday, 11/03: Fresno DS Meetup	Give progress report and showcase finished SARSA algorithm.
Friday & Saturday, 11/04 - 05: Irvine Chess tournament	Get additional feedback from chess players. Have chess players use the application to play against each other. Get feedback on questionnaire
Sunday, 11/06: Team meeting	How many visualizations do we have? Is the React chessboard working? Is the Python code for the SARSA algorithm complete?
Monday, 11/07: Reinforcement Learning book group	Group will discuss reinforcement learning and Q learning in particular.
Tuesday, 11/08: DS and ML study group	share project ideas and get tips and help.
Wednesday, 11/09: Anaheim chess club meeting	get additional feedback from chess players and do additional testing for application
Wednesday, 11/09: Week 2 Milestone due	see milestones
Thursday, 11/10: ML coding group	an opportunity to present SARSA algorithm and application and get feedback from data science professionals.
Sunday, 11/13: team meeting	do we have enough visualizations? Is the user able to query the database for analytics and visualizations. Is the SARSA algorithmn complete?
Wednesday 11/16: milestone week 3 due.	
Thursday 11/17: Fresno data science meeting	give progress report, get any feedback
Sunday 11/20: team meeting	video outline due, discuss what that will look like.
Wednesday 11/23: week 4 milestone due	
Sunday 11/27: team meeting	capstone project should be almost complete.
Wednesday 11/30: week 5 milestone due	
Thursday 12/01: meeting with Fresno data science meetup	Show completed application.

Date	Description
Friday 12/02: last meeting with chess club	show completed application.
Sunday 12/04: team meeting	video presentation due.

MILESTONE

Week # and milestones	Notes
Week 1: Dash app with basic visualizations, a chessboard, and a text field for chess move suggestion. Database complete with backend functional. SARSA impletementation plan will be complete.	The chess logic does not need to work at this point and the chess pieces don't need to be interactive. The backend and database complete. The SARSA algorithm can be pseudode at this point.
Week 2: Dash chessboard logic and interactive pieces will be complete, additional visualizations with user input. SARSA algorithm python code due.	The python code should have its own chess logic, use the python chess library for that. There should be 5 or more visualizations and analytics at this point. The user will be able to select which visualization to view from a dropdown.
Week 3: Dash will be complete and all chess logic is functional and chessboard is interactive. The SARSA agent has been trained and the algorithm can provide move suggestions that are shown to the user on the Dash	The visualizations are complete and the user will have the option to query the chess data for chess games by year and ELO (show the first 10 entries only).
Week 4: Testing of components and other functionality. Additional features included games of professional chess players. Make the GitHub repo visible so the data science club can provide feedback.	At this point the chess club will be able to use the application and provide feedback. The data science club can provide any feedback on the SARSA algorithm.
Week 5: Modifications or additions to Dash dashboard based on user input. Additional testing of code. Modifications to SARSA algorithm based on Fresno data science meetup members.	The feedback from both focus groups will be used to finish up the application. The machine learning component will be completely done by this week.
Week 6: Capstone video draft	
Week 7: Capstone Festival	

RESOURCES NEEDED

The resources needed are a PC with at least 16GB ram (32 is better), Plotly Dash, Numpy, Pandas, and PyChess.

RISKS AND DEPENDENCIES

RISKS

Hurdles that this team can run into is that Flask will be used to handle the back end of the application. No one on this team has experience with this framework. However, since the machine learning portion will be built using python, then it was deemed that using a backend framework that utilizes the same language would be necessary. Afterall, Flask uses python for the development of applications. Fortunately, a positive characteristic of Flask is that it's lightweight. Meaning it does not require a lot of dependencies to kickstart an application. Another aspect that gives the team confidence to proceed is that React, a front end framework, will be used. Along with a library called plotly. Now, with these two features the team is experienced in using them. So combining Flask, React and plotly will equate to a dash application the team is seeking out to create, and all three can be implemented with one language like python for simplicity.

DEPENDENCIES

The core of this project will be dependent on the implementation of Machine Learning and using certain chess libraries. The Capstone course has set a small timeframe for the project to be completed and deliverable. So it would be ideal to avoid programming all the logic of a chess board game, because that can easily represent an entirely different software development

life cycle. Fortunately there exists certain libraries that can inject a chess board with all the logic built in. This can work well with python too.

As for Machine Learning there is the question of where the code will exist. It was uncertain if it was going to be placed in the front end or back end. Fortunately, this team has done preliminary research of the libraries and frameworks that are planned to be used. It appears that Dash, which exists on the front end, will provide a convenient way to add machine learning code. There is strong documentation on how to achieve this, so it is promising that the answer will be answered soon.

PRELIMINARY USABILITY TEST PLAN

Usability testing will be completed by the Fresno Chess Club that consists of players between 1200 ELO and 2500 ELO. These players will test the Dash application on a laptop and use the application to play others players at the club and also a chess bot on Chess.com. The players will fill out a questionaire about their experience using the application. The Fresno Data Science meetup will have be able see the application on GitHub and also a virtual presentation. They will provide feedback on the machine learning implementation.

FINAL DELIVERABLES

The final deliverable for this project will be a Plotly Dash dashboard web that will allow users to play a game of chess and at each every turn the application will provide the user with a move suggestion. Data analytics and visualizations will also be provided to the user in another section of the application.

APPROACH AND METHODOLOGY

The team has chosen the agile methodology over the waterfall method. This team consists of three members and the project that will be undertaken can easily be split into three

sections. The first is the frontend that will involve React, Dash, and Plotly. The second is the back end which will consist of mysql and Flask. Then finally the machine learning logic that will utilize chess libraries and more. The great aspect of this project is that it is modular. Meaning that each role can implement their features independently. Eventually, to complete the final deliverable all sections will need to fit together. However, in terms of the micro features no section will be held back by the other. Giving each section the ability to be tested without relying on other parts to be completed. This is exactly what the agile methodology is. Create features, test them individually, make adjustments and then repeat.

TEAM MEMBERS

NAMES:

Abraham Borg, Sarom Thin, Mehar Rekhi

DIVISION OF LABOR:

Abraham: machine learning - SARSA implementation, communications with Fresno Data Science Meetup and the Fresno Chess Club.

Lom: database, backend and support for Dash.

Mehar: Dash dashboard and visualizations