Chess AI Project Proposal

DS 3500 Final Project

By: Adrian Monaghan, Maxwell Arnold, Nicholas Gjuraj, and Brandon Onyejekwe

1. **Motivation.**

Chess is a very popular game which has been played by millions of people for centuries. However, a computer could not beat the world champion in chess until IBM's Deep Blue won in 1997. Today, the advancement of technology has allowed some of the brightest minds to create programs that can easily beat the greatest human chess players in the world. However, creating an artificial intelligence to beat an average professional chess player is still challenging. This project will allow us to learn and utilize skills that will help us for years to come. Creating a chess AI is easy, but creating a great chess AI is challenging. Therefore, the project of creating an excellent AI will impress recruiters and graduate admissions committees alike.

2. **Goals and objectives**.

The project code will allow users to play a game of chess against each other or against an AI. The goal of our AI is to be able to regularly beat an above average chess player. Our code will probably not be useful to future developers, but it may serve as a model for any aspiring individuals attempting the same project. This project has been done many times and been perfected by the top of academia. However, the methods we use to generate and train the AI could give potential insights into solving more challenging problems or constructing a chess AI that plays with a unique style.

3. **Data sources**.

Since the project is self-contained, we do not need data sources in the traditional sense. However, it should be noted that we will be using this website page as a method for finding educational resources for the project. In addition, we could use this website API in order to test our AI on real players. Here are our data sources:

4. **Platform architecture**.

We plan to use Python to create a chess game that can be played by users (ie. Professor Rachlin). For the actual game of chess and logic, we are going to make a board object, likely storing values in a 2 dimensional, 8-by-8 array. The board will control the logic for determining if a move is valid or invalid relative to the other pieces on the board and utilizing the rules of chess. There will be a piece class which will be a base class to all other pieces on the chess board. Every chess piece object will be able to determine if a move is valid based on the possible relative movements of the

specified piece. If a move is valid for the piece and for the board, then the move will be made. Then we will implement this board with an artificial intelligence that will be able to read the board and determine the best move, and move the piece. By being able to read the chess board it will be able to attempt to determine the best possible move given the current state of the board. The chess AI will be trained using an evolutionary algorithm to determine best values of the hyperparameters. If time permits, we would like to create a bot to play on a competitive chess website for humans, such as chess.com, and find out the elo score of our artificial intelligence.