Background:

Poker is a game of luck, nuance, intricacies, and strategy. While you cannot control the cards you get and the board that's dealt, understanding how players think and patterns within the game can be complex and very important in making smart and valuable decisions. WinningEdge™ is a strategy optimization tool for online, low-stakes Texas hold'em players. The application will provide real time probabilities and suggest action based on an ML algorithm trained using historical data. In using WinningEdge™, players will be able to make better informed decisions with the goal of winning and being profitable.

User Profile:

The primary user of this software will be online, low-stakes, heads-up texas hold'em poker players. The players will want to use the tool to determine optimal strategy including bet/fold decisions at the river stage of the game as well as bet magnitude. The users will require the tool to be easy to use and require minimal input and respond rapidly such that it will inform their decisions and strategies in real time. The players will be of a variety of technical skill levels and such the software should be designed for users of minimal technical skill.

Another important user is the technician. They are responsible for maintaining and updating the software. Some possible updates include adding more data, creating potential plots and retraining the model with the extra data. They would be very technical and have a deeper understanding of the code and repository structure.

Data Sources:

The main data source that we are using are the ~30,000 raw text files (.txt) that were bought and provided by 888poker.com. In these files, the data is stored as

Use Cases:

Use Case 1: River Evaluator

Objective: The objective of the player's interaction would be to gain insights on how they should play the final stage of the poker game, the river, based on their cards, the board, and previous players actions, bet sizes, and stack sizes.

Interaction: The first action would be to input their cards and position (big blind/small blind) during the preflop as well as their opponents and personal stack size (current money they have). If the player does not fold, the user will play the preflop and upon seeing the flop cards, would input the flop cards and previous opponent and personal actions during the preflop round. (Ex, [('r',0.5)],('r',1),('k',0)] for raise $0.5, raise $1, call). If the player still does not fold, they will then input similar arguments for the turn card and turn actions. The player will then input the

river card, which will cause our code to run and output the likelihood that they will win against a random opponent's hand, as well as the recommended action and bet size based on the ML algorithm. The user will then be informed of the recommendation and will play the river round with this information in mind.

Use Case 2: Future Stage Evaluator

Objective: The objective of the player's interaction would be to gain insights on how they should play every stage of the poker game, including preflop, flop, turn, and river, based on their cards, the board at each stage, and both players actions, bet sizes, and stack sizes.

Interaction: Similar to Use Case 1, the first action would be to input their cards during the preflop, their position, as well as their opponent's and personal stack sizes. During the preflop stage, the evaluator would suggest the recommended action and corresponding bet size. The player would then play the preflop, and then input into the system the actions of the round. If both players did not fold, the same method would continue for the flop, turn, and river rounds, in which the player would input the newest cards to the board, the evaluator would output the recommended action and bet size, and then ask what each player's actions were during that stage.