Brian Holliday

Professor Burdis

Final Project

22 December 2024

Spades Trick Predictor

**Introduction**

Spades is a trick-taking card game in which usually played with four players. In Spades, what makes the game unique is that Spades are the trump suit of card. This means that if a card that has Spades as it’s suit will always win against a card of another suit, even if the card is a higher rank within its own suit. The object of the game is to correctly pick the number of rounds (tricks) that your team will win. You will only win the points for the round if you correctly predict the number of books you will make throughout the round. A reference for the rules of Spades can be found [here](https://www.wikihow.com/Play-Spades). Please note this is a simulated game and not every aspect of the game is adhered to, specifically the rules about bags and not leading rounds with Spades. We are only interested in simulating rounds of gameplay.

**Methodology**

There was limited data out there that would help answer this problem. For this problem, I had to simulate rounds of Spades and create synthetic data. The synthetic data was created by a Python program. In this program, I recorded the cards played in each trick, the players hands in each round, and the number of tricks won by each player in each round.

I set up a Players class to store played data. This includes the player’s name, the cards in each player’s hand, and the number of tricks that each player won in each round. We did not keep score for each player, since recording the winner for each round was not necessary for the simulation.

The Spades Game class track all aspects of the game in each round. This includes the deck used by the players in the game, the cards played by the players in each trick, the starting hand for each player in each round, and the number of tricks won by each player in each round.

For the gameplay, players will randomly play cards from there hand within the rules. For example, if clubs lead in a round, if the player has a club, they will random select a club from there hand, otherwise they will randomly select a card from their hand. This is simulated through the Python “Spades Game.py” script.

**Analysis**

A Random Forest was selected to evaluate the data. The Random Forest did the best with the training data out of the models that were evaluated. It had the best R^2 score. We were particularly interested in the mean absolute error for this problem, since we are trying to correctly predict the number of books that were achieved by each player in each round. We want to minimize our prediction for the number of tricks each player earned in the round. We chose a Random Forest with 100 estimators. We were able to get a mean absolute error score of 0.94. This means that we were about 1 trick off on our prediction on average. For our R^2 score on our training data we got 0.91 and 0.49 on our testing data. This means that our model was good at accounting for the randomness in the data that it was trained on, but did not perform as well on new data.

Most of all what we learn in particular through our exploratory data analysis and in the models that we trained is that all Spades cards and high-level face cards are particularly influential in predicting the number of tricks that a player achieved throughout a round. The analysis for this project can be found in the “Spades Analysis.ipynb” Jupyter Notebook.

**Conclusion**

More data is needed to answer this problem particularly from real players with winning strategies. Picking cards at random tells us where each card ranks in the game, but it does not account for cutting behavior and team strategies that are essential to the game, such as underbidding with a large lead. Picking cards at may have introduced excessive randomness in the game for our model when introduced to new data.