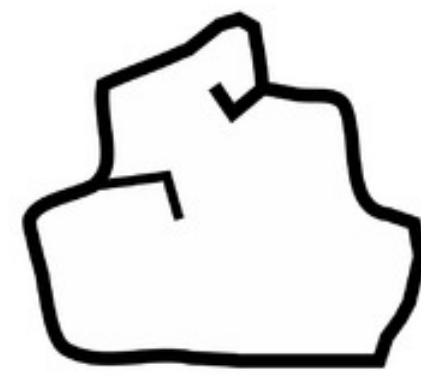


RPS BOT: Learning Human Patterns In A Random Game



rock



paper



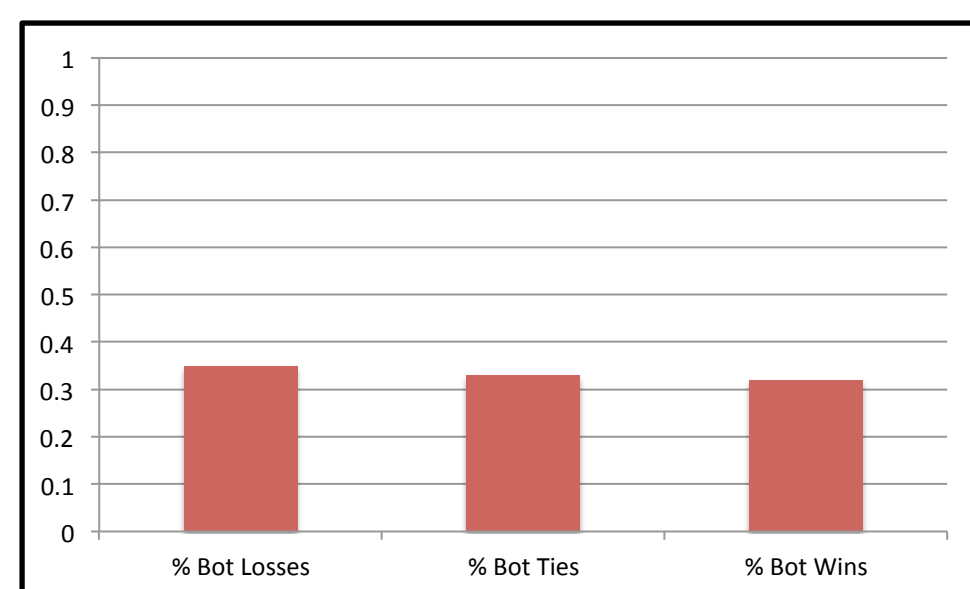
scissors

Tristan Breeden
'16

Problem Statement and Approach

The Nash Equilibrium for Rock Paper Scissors is to play randomly between the three options.

My baseline does this: for any given game, it selects a random play. After ~500 trials, I achieved the following result:



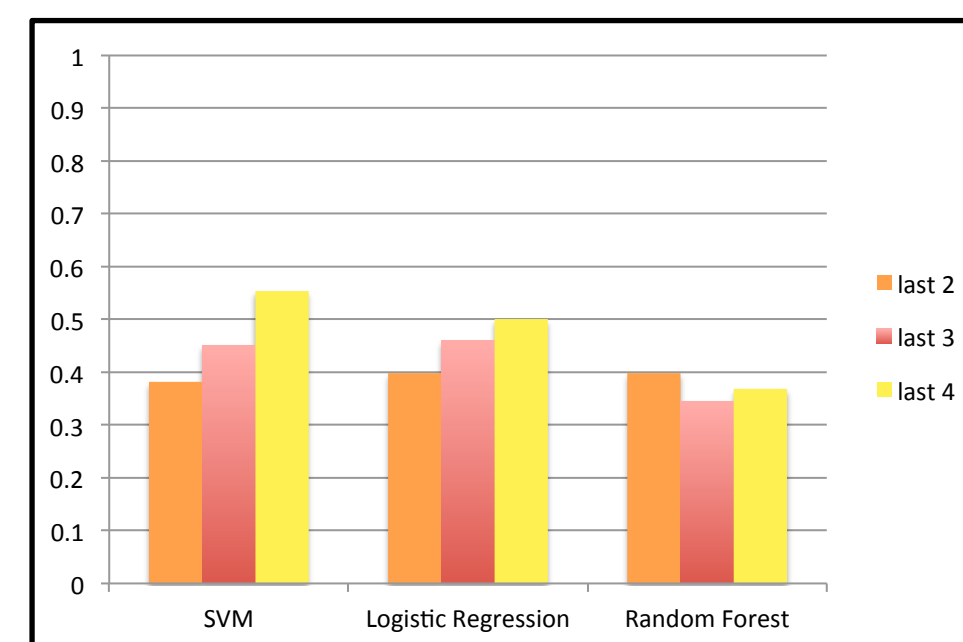
Baseline Results

My goal is to create a bot that improves upon that strategy by finding patterns in human play that tell us what will be played next in the game.

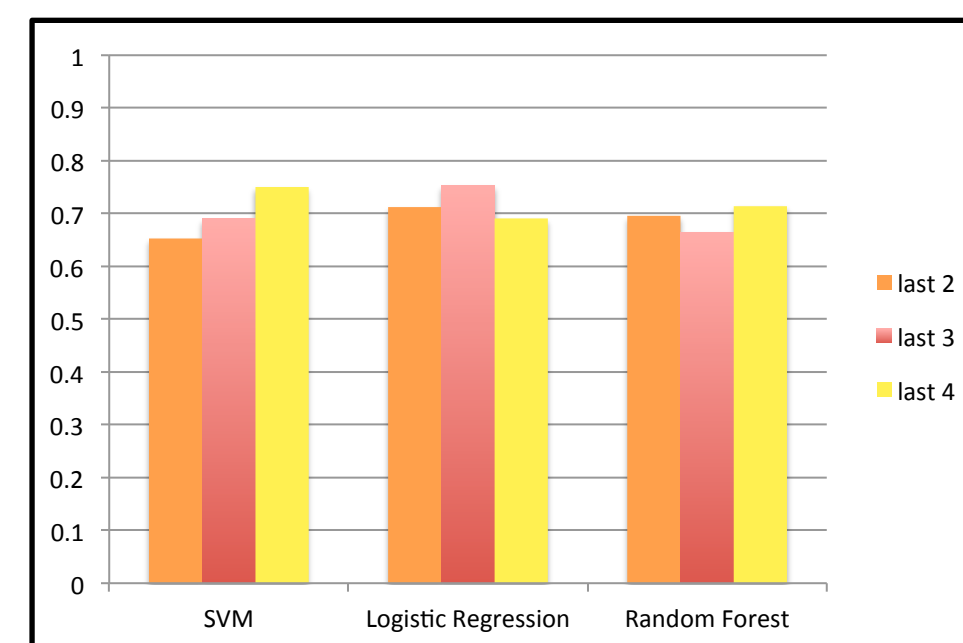
So far, I have used various classifying algorithms from the scikit learn library to predict a human opponent's next play. This was done using features that track the last one, two, three and four plays made by the human.

Results were obtained through my own games against the bot (approximately 500 per classifier).

Initial Results



Percentage Of Bot Wins



Percentage Of Bot Non-losses

Using a SVM classifier, I can achieve a win rate of 55%, a significant increase from my 32% baseline win rate.

Logistic regression and the Random Forest Classifier also yielded improved win rates, but not as good as the SVM classifier.

Analysis And Final Work

Looking more in depth at the data from my results, it seems that the bot is picking up on these common RPS strategies:

- Play 'Paper' as your first throw, since most non-competitive players start with 'Rock'
- Play the throw that would have lost to the human opponent's last throw, as inexperienced players often deliver the throw that beat their last one
- Play 'Paper' more commonly than the other throws, since it is statistically the less used throw

I initially chose these three classifiers in order to sample a variety of unsupervised learning methods offered by the scikit learn library. Over the course of the next week, I need to continue examining the pros and cons of each of them in order to see which one is best suited for my data set and features.

Additionally, I need extract more advantageous features. For instance, losing or winning a game seems to affect my playing. I would like the bot to pick up on that, so it can be even more accurate. I am considering interlacing the bot's play history with the human players history, in order to extract more information from it.