Blackjack Simulation

We chose to analyze blackjack data comprised of 6 players with 150,000 hands of blackjack each totaling 900,000 rows of data. Knowing blackjack is a game of chance and the rules favor the house by forcing the player to act without knowing the dealer’s hand, we felt there had to be a better way to even the odds. We setout to do just that by using neural network and multiple linear regression.

The ultimate goal is for the dealer to bust and the player to stay or hit when appropriate ultimately extending their game play and hopefully winning more money.

We built a simulation in Python Pandas to test and lower the odds of busting by staying with a hand of 12. For the simulation there are 4 decks of cards totaling 208 cards shuffled randomly. The set value for face cards is 10 and the Ace is valued at 11 or 1.

With the 2-card sum of 11, the player can take another card without the risk of busting due to the highest non-interchangeable card having a value of 10. This hand has a high possibility of getting blackjack versus busting.

With a 2-card sum of 12, the player should stay. If he takes another card, there’s a high chance he will bust. If he stays, there’s a chance the dealer will bust because the dealer must have a hard 17 or hit until he reaches a soft 17. increasing the likelihood of the dealer busting.

900,000 simulations were run through the simulation producing random hands that resulted in 415,343 wins, 55,442 pushes, and 429,215 losses.

Therefore, when a player stays at 12, he has a 46.15% chance to win, a 6.16% chance for a push, and a 47.69% chance to lose.

We compared this with the blackjack data. When a player stayed at 12, he won 32.17% of the time, lost 62.11% of the time and pushed 5.71% of the time.

This comparison shows the exact opposite effect. When hands are randomly generated and the player stays at 12, his changes of winning are much greater than losing. This was most definitely not the case with the blackjack dataset. The player lost much more than won.

This difference would suggest that the player is most likely influenced several factors such as how the other players are the table are playing, the overall excitement of playing the game and the possibility of winning, and taking chances he wouldn’t normally take in a Casino environment, basically the Go Big or Go home mentality.

Neural Network

We also built a neural network using the Blackjack dataset.

We are using ML and logistic regression to be specific to predict the chances of - A) a player not going for the next bet based on the card counting principles and B) a player winning a hand based on the cards being delivered. We are using regularization to tune the data and compare the results with the co-relation derived for the same data set using Pandas and Matplotlib. \*\* Assuming our dataset could be a high volume one, given the randomness of the data, we have used AWS - S3 as the single source to access the data set for all components of our entire project.

[6:47](https://uncchadatapt0-ccd2271.slack.com/archives/GUJUBKL7R/p1583495272008100)

For the component flows - We have used ZEPL notebook to clean up the data and then perform ML and MATPLOTLIB using Pandas on Jupyter notebook.