Blackjack Sim 2021

By Afnan Rehman

Abstract

Let's be honest, casino games are tough. They are usually stacked against you and the odds of winning big are generally known to be infinitesimal at best. However, a lot of fun research has been done over the years into approximating this rate of success or failure through the art of simulation.

Blackjack is one of the most well-studied games in this regard. The possibilities are huge, impossible to keep track of via memorization or iterative method, and usually only kept track of though simulated chance. By producing the best probability of a good outcome, you can try and work the system as favorably as you can towards yourself and ensure better chances of winning, or at he very least not going bust in a spectacular fashion.

This paper explores simulation of Blackjack and analysis of results using usual house rules and how success rate can be affected by different strategies. As someone who is not familiar with casino games in the slightest, I found this to be an interesting and fruitful learning experience on how these games work and how chance and statistics can be used in various ways to game the system (or game in this case).

Problem Background

Blackjack is one of the most well-known casino games. It involves card decks and a lot of chance, but there are some known strategies out there to mitigate your losses and maximize your odds of winning.

Blackjack is an interesting game in that it seems relatively simple at its surface. At its core, the player is dealt two cards, as is the dealer. Each card has a point value to it, and whichever has the higher point value but less than 21 total, wins. However, the game becomes more nuanced when the rules are examined.

Looking up the rules, the player decides whether to "hit" or "stand", in other words to ask for another card or not, respectively. The player may continue to hit until they decide to stop, or they go bust. The "blackjack" is a hand with a ten-card and an ace and is the best possible hand because you are already at the limit of 21. A player with a blackjack automatically wins unless the dealer also has one, in which case it is a tie.

The dealer on the other hand, once the player finishes, must play by a certain set of rules and continue to hit (draw cards) until their total is at or above 17 at which point, they stop. The dealer can also go bust, resulting in an automatic win for the player.

Assuming both the dealer and the player make it without busting, the round then moves to a comparison of raw numbers. Whoever has the highest value hand wins, and in the event of a tie, all the bets and hands are reset.

This process is further complicated by two additional rules: the split and the ace value. Regarding splitting, this is allowed when a player's initial hand has two of the same value cards. The player may

split the hand into two if they can place a bet equivalent to their current bet on the newly formed second hand. The player then plays for both hands. This can be beneficial in certain scenarios and may be harmful at high or low card levels.

The ace value is another complication because it is able to change. Typically, the point value of an ace is set to 11. However, the player may choose to set this to a 1 if it benefits them or keeps them from busting after a hit. This is another choice that again increases the complexity of the game for regular calculation.

This paper will go through a description of my simulator, as well as the performance and assumptions or limitations I may have placed in the process of creating it. I will include plots and generated results data in the paper as well as in the appendix to showcase the results we were getting and discuss the performance of various strategies employed. After that I will tie it up with a conclusion on the results and findings.

Main Findings

The game itself has many different possible paths to take given the nature of different rulesets, stratagems that depend on the money you have available, and even location-based differences. With that in mind, there were several limitations and assumptions I placed on the simulation. First, that each "game" would only use 1 deck, instead of the possible up to 8 that I have seen suggested. Second, the simulation assumed the game would be held between only one player and one dealer. Third, in the case of a split being made, I focused on the left split and disregarded the right split.

In addition to the above, there was no consideration for money being made or doubling down. I simply focused on the win, loss, or tie for the player and utilized know card values to make decisions with the objective of winning. Surrenders were not taken into consideration as my research has shown that is not a common rule at Blackjack tables.

The coding for this simulator was done in Python 3.8 using Jupyter Notebooks. This allowed me to segment the code and show result tables easily. If you would like to run the program, the notebook file has been included in the project submission and can be ran using an install of Jupyter notebooks (easy if using Anaconda). In addition, a HTML of the notebook contents is included that can be opened in any browser for your convenience.

My first strategy going into creating this simulator was to hit unless the first dealer card was more than a 7, and to always split if there was a chance. This strategy was strangely consistent, possibly due to the PRN not being an extremely accurate representation of reality. I tried to use the Python random package, which is the best quick solution, but with some exploration others may also work better. In addition, the results were also quite bad, with around a 33% win-rate over many runs (about 4.5 million, see fig 1).

	Win%	Loss%	Tie%	rounds_played
0	0.330769	0.627356	0.0418747	920485
1	0.330524	0.62756	0.041916	920627
2	0.3308	0.627292	0.0419083	920581
3	0.330432	0.627498	0.0420703	920292
4	0.330374	0.627648	0.0419771	920383

Figure 1 Win/Loss on Ultra Basic Strategy

The results here show each simulation sequence (total of 5). Within each sequence I played through 100,000 generated and randomly shuffled 52 card decks and each "round" played is from the dealing of cards until the final comparison of values. Based on these numbers, we see that each deck seems to last around 9.2 rounds, and win rate is steady at approximately 33% of the games played, losses at 62.7% of games played, and about 4.2% of games resulting in a tie.

Another strategy I created was a bit more complex and involved decisions on hits or stands on various dealer card values. I still did not touch splitting or dynamic ace values in this version of the strategy, but I did implement the following rules:

- If the dealer upcard (the first dealer card) is a 7 or higher, the player will hit until a total of 17 or greater is reached.
- If the dealer upcard is 4, 5, or 6, the player stops drawing once they get to 12 or higher.
- If the dealer upcard is 2 or 3, the player stops hitting at 13 or higher.

These rules were taken from the basic guide I have been using to reference rules and play for Blackjack. They seem to be general guidelines for basic strategy, so I hoped for a received a higher win rate, as seen below:

	Win%	Loss%	Tie%	rounds_played
0	0.430965	0.484653	0.0843821	362802
1	0.431253	0.484235	0.0845118	362766
2	0.433471	0.481235	0.0852943	362744
3	0.431797	0.483158	0.0850448	362703
4	0.431009	0.483143	0.0858483	362663

Figure 2 Basic strategy sans splitting and ace changes

As we can see, the success rate shot up to about 43%, which is starting to look fairer whilst maintaining the dealer's edge in the game.

I then added a rule to switch the ace value from 11 to 1 if a "soft hand" is detected. A soft hand in Blackjack is a combination of an ace with a card other than a ten-card (card with value of 10 such as a face card). Usually this would be a decision made by the player, but I set it to always make this choice if the conditions are met.

	Win%	Loss%	Tie%	rounds_played
0	0.432479	0.483797	0.0837234	362718
1	0.434067	0.481672	0.0842609	362695
2	0.433205	0.481082	0.0857125	362701
3	0.433599	0.481163	0.0852378	362609
4	0.43298	0.482669	0.0843513	362638

Figure 3 Basic strategy with soft hand switch

As we can see here, this rule did slightly improve things by anywhere from 0-0.2%, chipping away further at the dealer's edge.

The final exploration I attempted was to explore how the dealer's hand played into the probability of a win or loss. To do this I had to redo the simulation output to also include the dealer's first card along with the round result. I felt the first card was important as it is a major component for the player strategy, and the only information available on the opponent while the player is making their decision.

What I found was an interesting but understandable distribution where a high dealer card more often led to a loss than a win, and low dealer cards were more likely to result in a win.

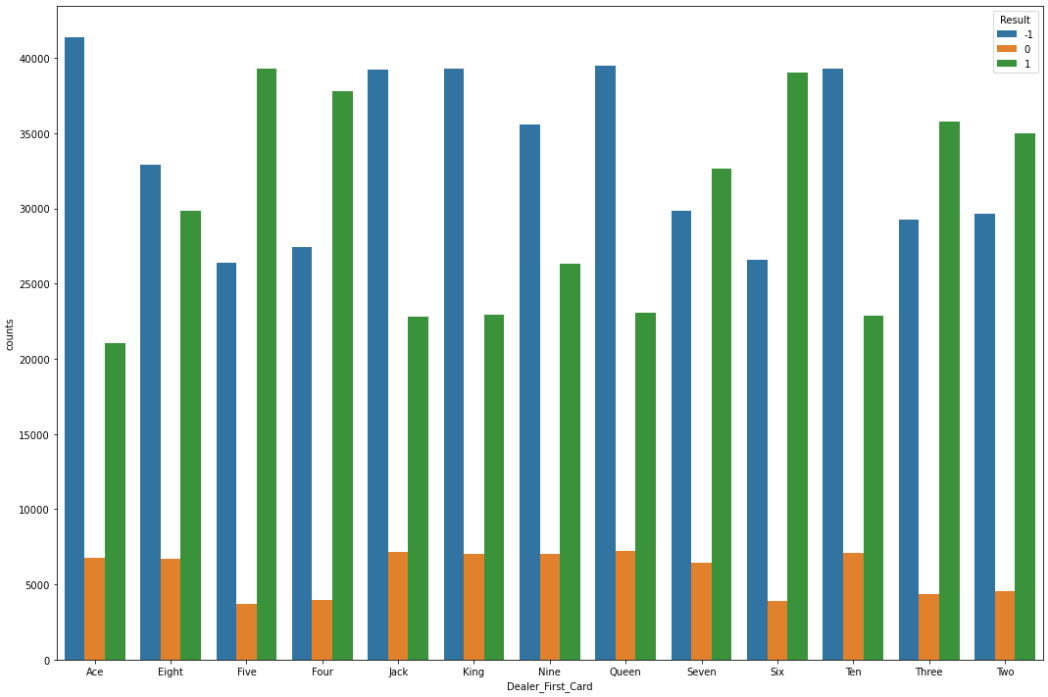

Figure 4 Distribution of win/loss using Basic Strategy

This makes sense since a high initial card would more likely result in a high overall value for the dealer's hand. In the case of an ace card, we see the disparity is most pronounced, since many of those could have also been a dealer blackjack, leading to an automatic loss for the player. Whereas if you have the first dealer card being lower, such as a five, you can feel better about your chances.

Overall, we can see how this strategy can produce “good enough” results for the casual player. With careful tuning we can attempt to gain the last 4-6% win-rate over the dealer and close the gap, although that will require some deep dives into the realm of casino strategy and card counting.

Conclusions

In conclusion, Blackjack is harder than it looks! It was interesting to dive deep into this game especially as someone completely unfamiliar with most gambling and card games.

Beyond learning a lot about the rules of Blackjack specifically, I was able to learn more about how casino games are generally tilted towards the casino. This makes sense given that the casino must make money, although it is interesting to see how rules and probability are tweaked to give the dealer just enough of an edge to make money over the long term whilst leaving customers feeling satisfied and not cheated.

On the technical side of things, I had to further flex my brain into optimization and code cleanliness so as not to sit for hours on end waiting for the simulation to finish and final results to be tallied. Honestly, this part took almost half the total time spent on the project. However, it was a necessary step not only to achieve results now but also to further develop this simulator effectively and efficiently in the future.

Speaking of the future, one of the nice things about the complexity, however, is that it leaves a good amount of room for future development work. There are many established as well as new strategies with different rulesets that could be reanalyzed using this method. With a higher amount of computing power, one could also attempt to randomize the choices on all the potential decision points, training a truly optimized model for how to best play blackjack.

One idea for the future is to try to at least randomize one or two decisions and try to manually converge on an optimal solution. For example, splitting 80% of the time vs 100% of the time when a matching pair is held, and seeing how that would affect the win rate.

Appendix and Citations

Blackjack – Card Game Rules. Bicycle Playing Cards. (n.d.). https://bicyclecards.com/how-to-play/blackjack/.

Wikimedia Foundation. (2021, June 21). Blackjack. Wikipedia. https://en.wikipedia.org/wiki/Blackjack#Rules_of_play_at_casinos.