TP1: Tunisian Delight

Due: Friday, November 03, 2023 (12:00pm)

Learning Objectives

The purpose of this assignment is to learn about random numbers, probability, and Monte Carlo simulation. Along the way, you will also have a bit of fun by designing one of your own new Tunisian card games.

Background

The number of tourists visiting a popular Tunisian casino has been declining lately. The casino's management has enlisted you to help evaluate some new candidate card games that might help boost tourist traffic to their gaming tables. Your task is to determine which of these new card games are the most attractive or the least attractive for players, so that the casino knows which new card game(s) to deploy.

Here are the new card games that are being considered:

1. Sahara Ace: The player draws one card. If the card is an ace, then the player wins 10 Tunisian dinars. Otherwise, the player loses.

2. Tunisian Twins: The player draws two cards. If the two cards are identical (i.e., suit, colour, and rank), then the player wins 50 Tunisian dinars. Otherwise, the player loses.

3. Medina Biggie: The player draws two cards. If the second card drawn has a strictly higher rank than the first card drawn, then the player wins 2 Tunisian dinars. Otherwise, the player loses.

4. Desert Hearts: The player draws three cards. If at least one card is Hearts, then the player wins an amount equal to the number of drawn cards that are Hearts, in Tunisian dinars. Otherwise, the player loses.

5. Oasis Runny: The player draws five cards. If there is a subset of these cards that makes a run of three or more consecutive card values (e.g., 5-6-7 or 10-J-Q), then the player wins 5 Tunisian dinars. Otherwise, the player loses.

6. Your Game Here: Please make up your own proposed Tunisian card game here. It should involve at least 2 cards, but at most 10 cards, and have simple rules and payouts.

Technical Requirements

In this assignment, you will implement a Monte Carlo simulation in Python to evaluate and compare six different Tunisian card games. For each game, you will simulate 100,000 plays of the game to estimate the probability of winning.

You will also need to design and implement the code to model a card dealer who is working with a well-shuffled single deck of cards. You can use the example code for an infinite-deck card dealer as a starting point, but you will need a more realistic dealer model to complete your Monte Carlo experiments.

When you are finished, please submit your solution in electronic form to your Google Classroom space. Your submission should include the source code for your simulation program, a brief user manual describing how to use your simulator, and a description of the results generated using your program.

Remember that assignments are to be done individually and submitted on or before the stated deadline. Late submissions will incur a penalty of 2 marks per day (or portion thereof) beyond the deadline.

Grading Rubric

The grading scheme for the assignment is as follows:

5 marks for the design and implementation of a proper Monte Carlo simulator (i.e., main iterative loop, random number generation, statistical output, etc).

3 marks for a suitable implementation of a single-deck card dealer (i.e., functionality, correctness, efficiency).

6 marks for the correct implementation of the Tunisian card games indicated above (1 marks each, total of 6).

4 marks for your table of simulation results estimating the likelihood of winning each game in both infinite-deck and single-deck scenarios. Augment this table with the expected winnings (in Tunisian dinars) per play, for each dealer model. Provide a few sentences summarizing your results and observations.

2 marks for a clear and concise user manual (at most 1 page) that describes how to configure, and use your simulation program. Make sure to clarify where and how the testing was done, what works, and what does not. Be honest!

Up to 4 bonus marks will be awarded for a concise mathematical analysis of the probability of winning each of the Tunisian card games, as validation for your simulation results.