CS5180 – Ex4

Sudhendra Kambhamettu
NUID: 002786797

Q1.

The first-visit Monte Carlo algorithm calculates the average return for each state over all the episodes only considering the first visit for each state in a given episode. It uses this to update the value function towards the average return.

We can modify this algorithm to use incremental implementation for sample averages as follows –

As described in the Section 2.4, generally the algorithm for incremental updates for sample averages look as follows –

$$Q_{n+1} = Q_n + \frac{1}{n}(R_n - Q_n)$$

Now, we can use the same method by adjusting the **value update step** for each state to incorporate the incremental formula. Which would look as follows –

$$V(S_t) = V(S_t) + \frac{1}{n}\big(G - V(S_t)\big)$$

This means, for each episode and for each state visited, you would update the state's value based on the incremental average of the returns following the first visit to that state in each episode. Mathematically over the episodes –

$$\therefore V_n(S_t) = V_{n-1} + \frac{1}{n}\big(G_n(t) - V_{n-1}(S_t)\big)$$

Q2.

1. In the blackjack task, using every-visit Monte Carlo (MC) instead of first-visit MC is unlikely to yield significantly different results for two main reasons. Firstly, the deterministic nature of blackjack, driven by the outcomes based on the player's and dealer's cards, ensures that the sequence of actions within an episode exhibits less variability compared to other tasks. This inherent structure of the game limits the difference that the choice between first-visit and every-visit MC could make. Secondly, the nature of blackjack gameplay, where episodes rarely involve revisiting the same state within a single game due to the game's rules and objectives, makes the first-visit and every-visit methods functionally equivalent. Essentially, the uniqueness of each state within a blackjack episode means there aren't multiple opportunities to revisit the same state, rendering the distinction between first-visit and every-visit MC methods moot for this particular task.
2. A. For first-visit MC, we only consider the return from the first time the non-terminal state is visited within an episode. However, since there is a single non-terminal state and we observe it from the beginning, the first-visit estimation will be based on the total return observed from that single episode. With $\gamma = 1$, the return from the first visit (and since it's the only state visited before transitioning to the terminal state) is the total reward accumulated until the episode ends, which is 10.

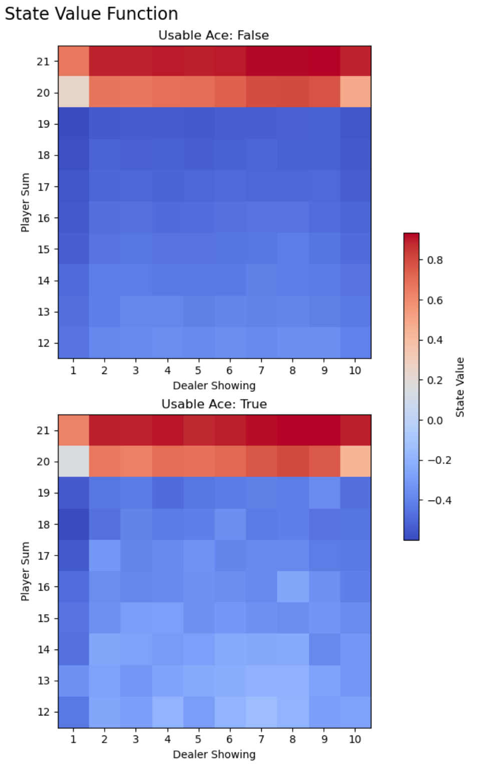
   B. Every visit estimator is the average of all the 10 returns i.e.,

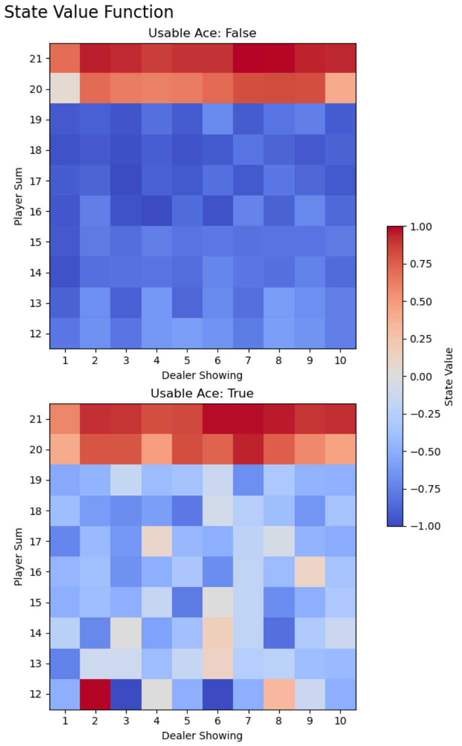
$$G = \frac{1+2+3+4+5+6+7+8+9+10}{10} = 5.5$$

Q3.

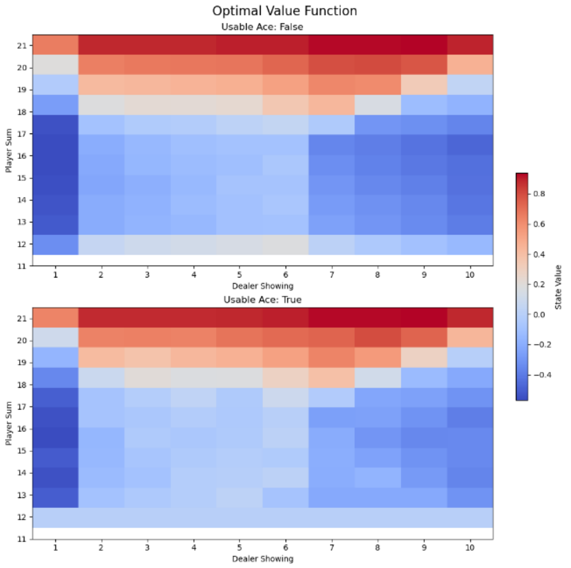
Plot

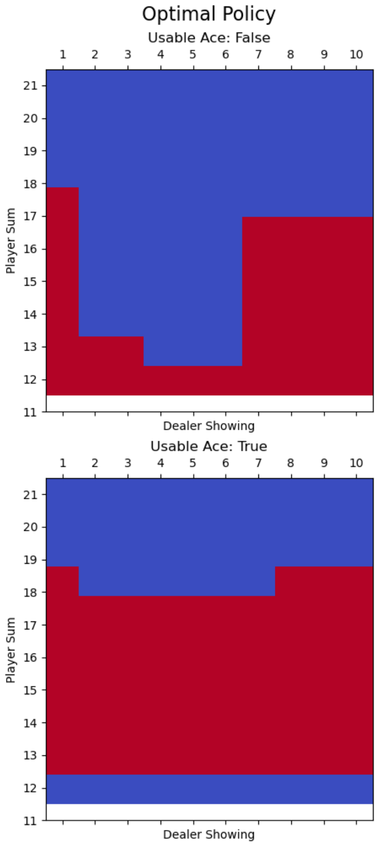
Part a





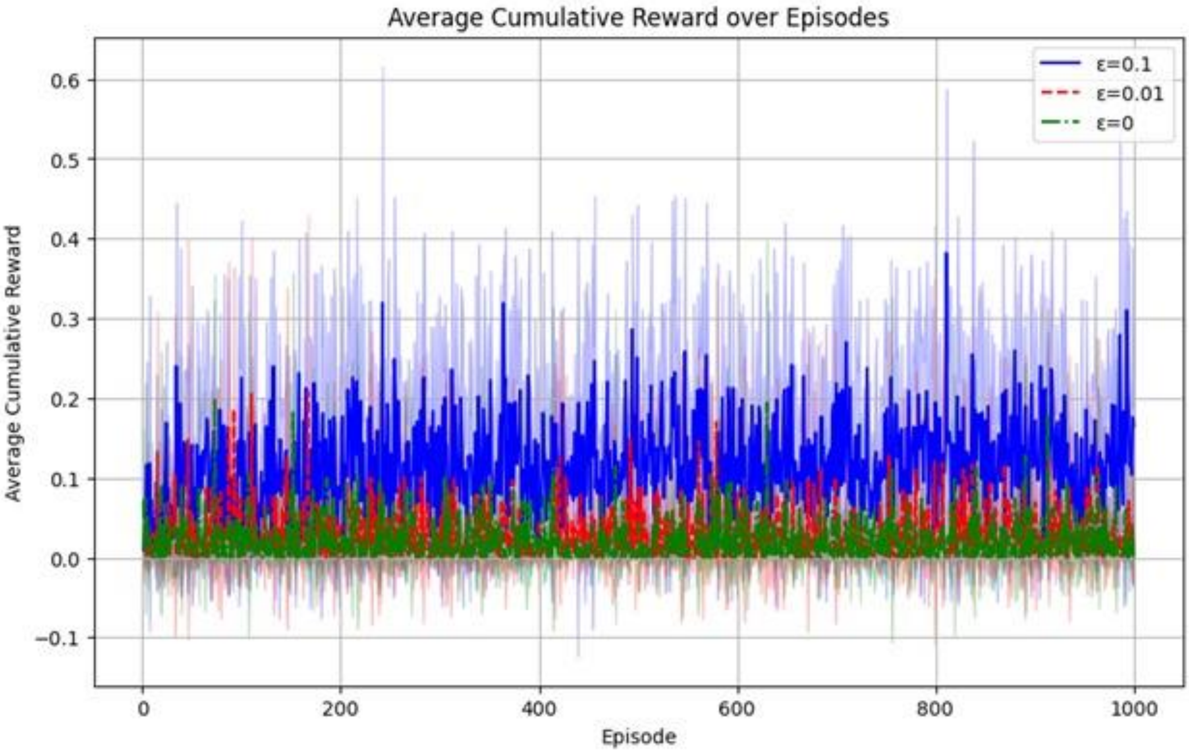
Part b





Q4.

Plot



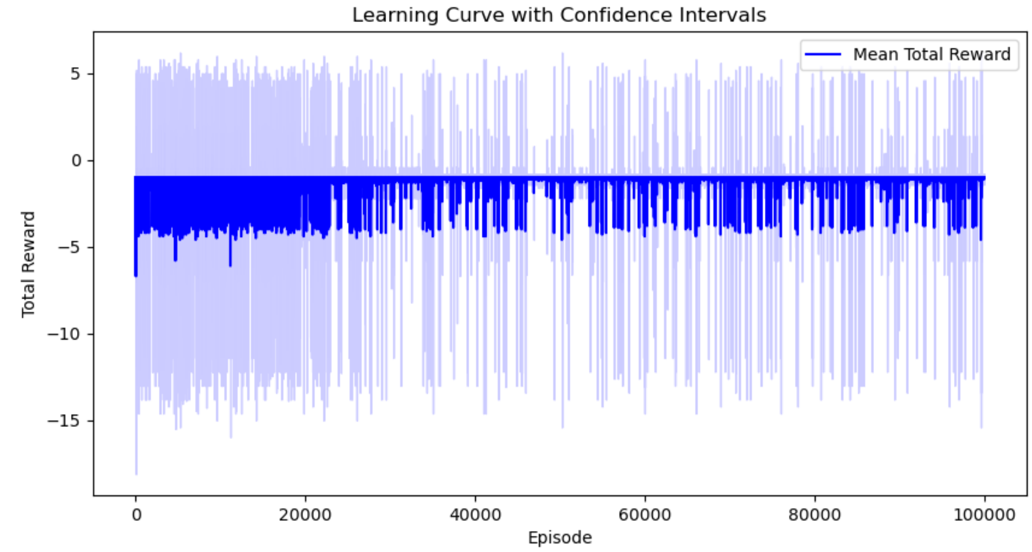
Written

The significance of exploration is illustrated by the outcomes for ε = 0 in the context of the Monte-Carlo ES (Exploring Starts) technique. To guarantee that every state-action combination is sampled in Monte-Carlo ES, exploratory beginnings are employed. The policy acts greedily, choosing just the most well-known action that is currently available when ε = 0. There is no unpredictability in the action selection process. The agent would never investigate behaviors that are not optimum, even if they could result in superior long-term rewards, if exploring starts were not there. The findings for the ε = 0 option are less successful than those for the ε > 0 settings, highlighting the need to explore starts in order to keep the policy from becoming trapped in a less-than-ideal deterministic policy. This is due to the fact that insufficient exploration prevents the agent from discovering possibly better behaviors that might result in greater rewards.

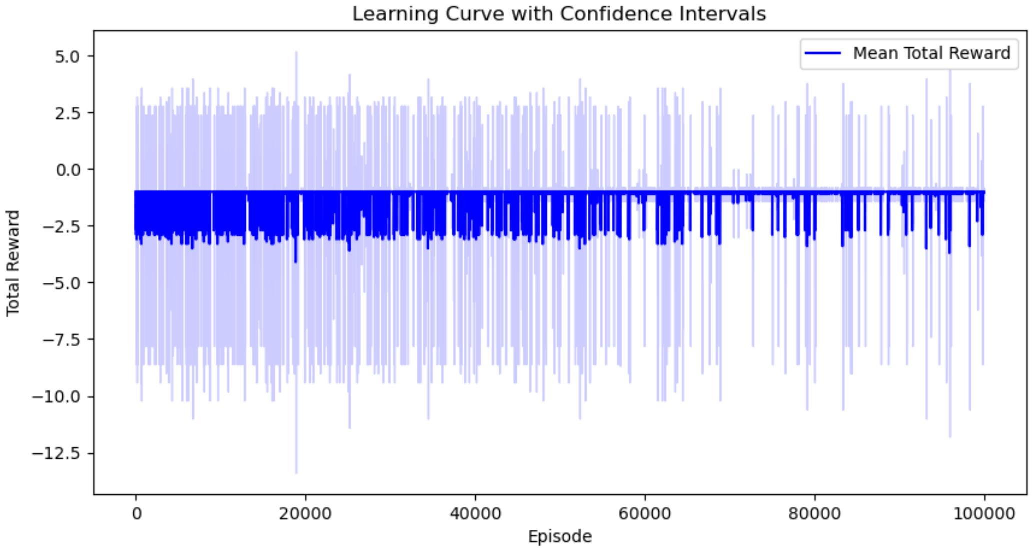
Q5.

Part a

Track 0

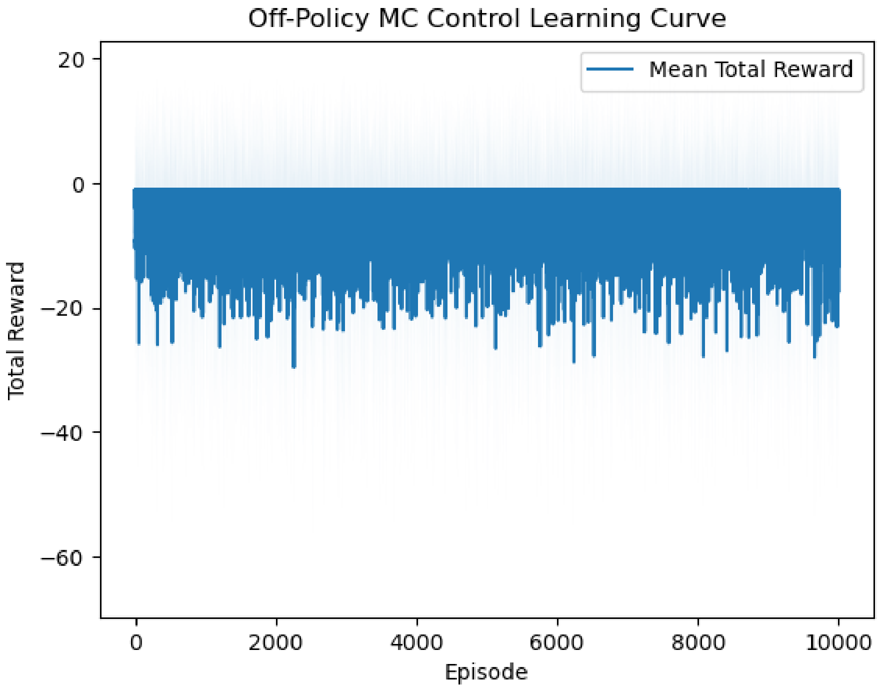


Track 1

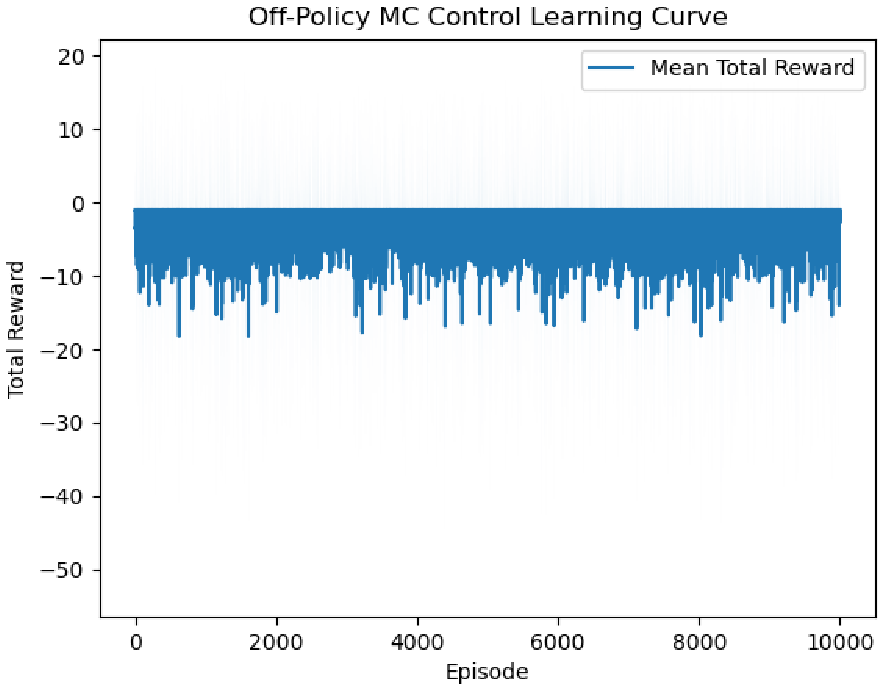


Part b

Track 0



Track 1



Written

Based on the learning curves obtained for the track 0 and track 1 on-policy (part a) and off-policy (part b) methods:

1. Since the policy being followed is also the one being improved, the on-policy technique displays a consistent learning curve. The number of episodes grows with a progressive improvement in the learning curve.

2. The learning curves for the off-policy approach appear to have more variance and slower convergence. This is because it is more difficult to appropriately evaluate the policy that is being improved because it differs from the behavior policy.

3. The on-policy and off-policy approaches appear to converge for track 0, but the off-policy approach exhibits greater variation in the rewards for each episode, suggesting a more exploratory behavior policy.

4. The off-policy technique appears to outperform the on-policy method for track 1, indicating that the behavior policy of the off-policy method may be better suited to investigate the unique structure of track 1.

Off-policy methods can potentially yield a better-performing policy since they separate the exploration process from the policy improvement process. However, they can also lead to higher variance in learning and may require more episodes to converge. The differences in performance between the two tracks likely arise from their structural differences, which may affect how exploration and exploitation are balanced by the learning algorithm.