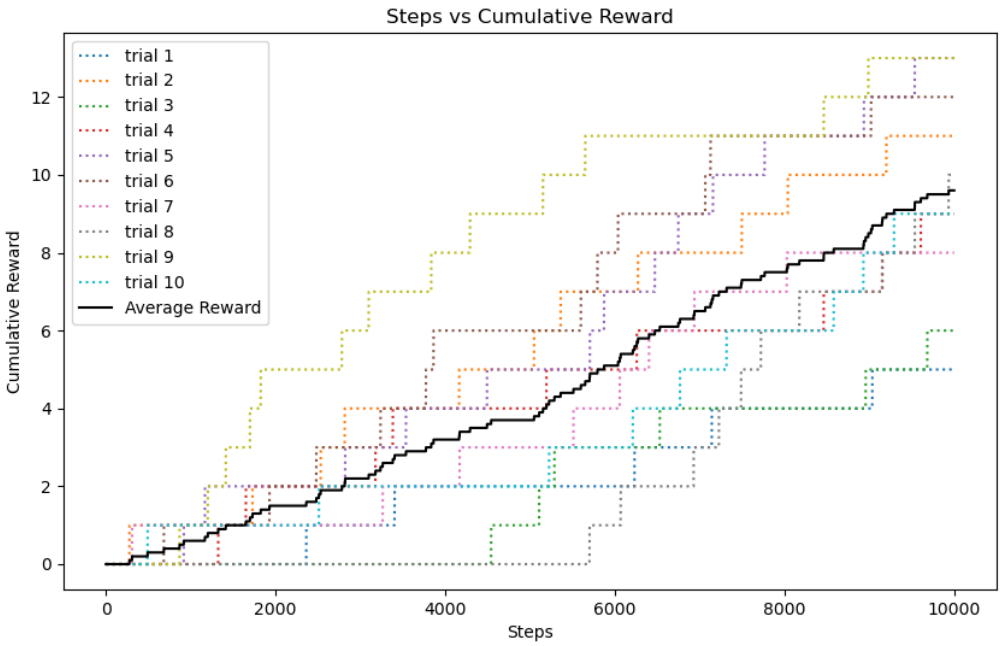
Q3.

Written

Ideally, the manual policy will outperform random policy since the agent would be me (i.e., the person themselves). Knowing a goal state will make a human agent perform a lot better than the agent with a random policy.

But in cases where the human agent isn't aware of the goal state, random policy might have a better edge in reaching the goal since it can explore a lot further.

Plot



Q4.

Written

Better Policy – Better policy uses a multi-stage epsilon greedy algorithm for balancing the tradeoff between exploration and exploitation.

For exploitation the better policy calculates distance between all the possible next states that will emerge from all possible actions and picks the action which leads to the next state which is the closest to the goal state by distance. Additionally, the policy also computes 'legal actions' which are the actions from the current state of the agent which won't lead to the agent crossing the boundary or bumping into the walls.

Further the better policy prefers actions that will lead the agent towards the goal state by checking if it's current state's X, Y co-ordinates are close to the goal state & explores with a 60% probability to move towards UP & RIGHT and 40% towards DOWN & LEFT.

This policy inevitably will lead the agent towards the goal giving a generally better performance.

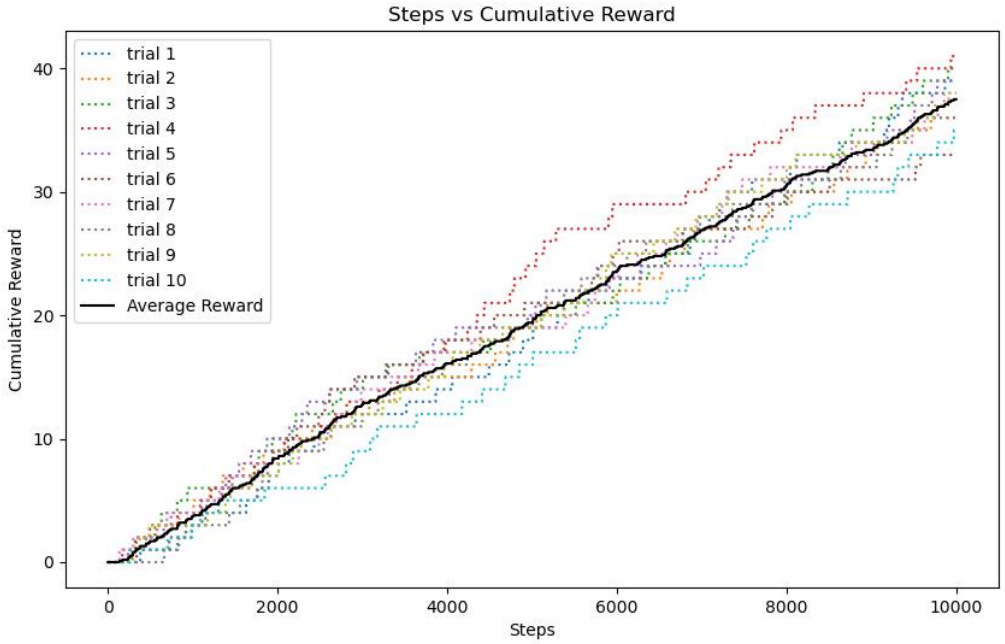
Worse Policy –

The worse policy contrary to the better policy actively outputs actions which will lead the agent away from the goal state and when not aligned with the goal, the policy prefers moving left and down over moving right and up. This preference is more likely to increase the distance from the goal, especially in a grid where the goal is randomly placed.

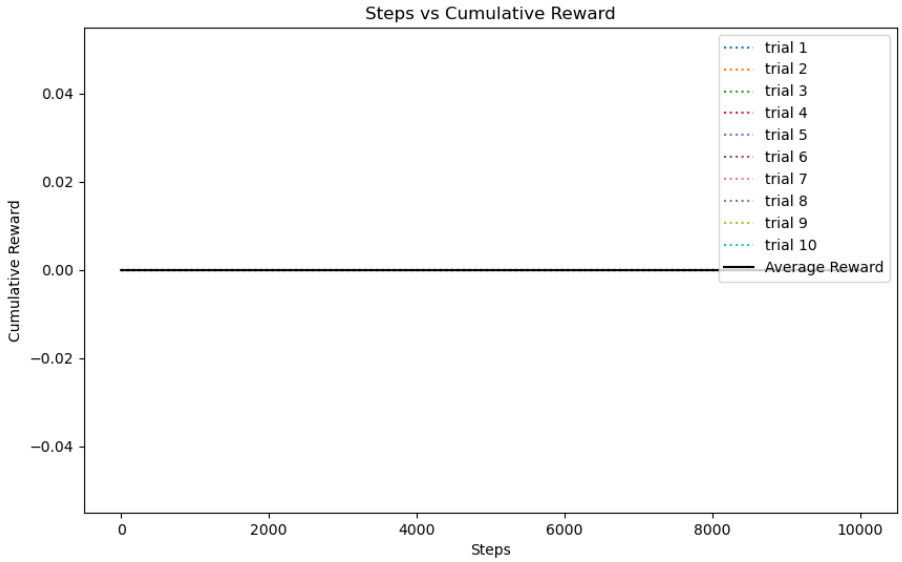
Therefore, leading to a generally worse policy.

Plot

Better policy



Worse policy



Q5.

Written

The learned agent uses a learned policy which accounts for exploration and exploitation as follows.

Exploration: Initially since the agent is not aware of the goal state it's imperative that the agent explores to find the goal state. Once, the first reward is known to the agent, the policy marks the goal state for the trial by using a Boolean flag.

Exploitation: Once, the policy is aware of the goal state after it's initial exploration the learned policy uses epsilon greedy strategy with Q-learning. At every step the Q value of each state in the environment is updated in the Q-table, using the Q-learning update rule. Then the policy picks the best action among all the possible actions from the current state which will lead to the next state with the highest Q-value. Over the 10000 steps, the policy forms a good understanding i.e., a close to accurate Q-values for each state therefore exploiting the path which maximizes the reward.

Another key observation was that the learning agent performs significantly better when run for more than 100,000 steps or more.

Plot