CAP 5636: Assignment 4 Explanations

Problem 1:

Implement an explicit policy for the mountain car environment without using any learning algorithm. Explain in detail your reasoning behind your policy and run several test episodes to measure its performance.

Explanation: In this issue, the action variable can have 3 values 0,1 or 2, identifying to 'left', 'no push' and 'right'. The obs variable can have 2 values, position and speed. The explicit policy used in this question uses the approach that if the speed if negative then return 0. By doing so, it knows that it has to turn left or it returns 2 so it moves right.

PSEUDO CODE:
if speed < 0:

return 0
elif SPEED >= 0:
return 2

Problem 2:

Implement an explicit policy for the cartpole environment without using any learning algorithm. Explain in detail your reasoning behind your policy and run several test episodes to measure its performance.

Explanation: The point of the problem is to balance a pole on the cart. The action variable can have two values 1 and 0. The observation number can be positive or negative . If the speed registered is positive then move right and if the speed registered is negative move left.

PSEUDO CODE:

If state<0: return 0 elif state<=0 return 1
action=explicitPolicy(obs[2])

Problem 3:

Apply the cross-entropy method to mountain car. Explain how many episodes are needed to learn a good policy. Explain which reward you use (original, modified).

Explanation: The Cross-Entropy method is a technique every now and again utilized for uncommon occasion simulation and optimization. The essential strides of cross-entropy comprise of creating irregular information tests and keeping up a circulation of good examples as per some scoring system to produce new examples from. Cross-entropy can be connected to support learning by learning (improving) an esteemed work. The cross-entropy method for MountainCar issue took 150Episodes and 49 iterations to accomplish a normal score of 90.21.