Implement a Basic Driving Agent

QUESTION: *Observe what you see with the agent's behavior as it takes random actions. Does the* ***smartcab*** *eventually make it to the destination? Are there any other interesting observations to note?*

When smartcab agent was tuned to take random actions at any state I observed following trends:

1. *Smart cab reached the destination point 'luckily'.*
2. *It made actions that had no sense.*

Inform the Driving Agent

QUESTION: *What states have you identified that are appropriate for modeling the* ***smartcab*** *and environment? Why do you believe each of these states to be appropriate for this problem?*

OPTIONAL: *How many states in total exist for the* ***smartcab*** *in this environment? Does this number seem reasonable given that the goal of Q-Learning is to learn and make informed decisions about each state? Why or why not?*

I identified the following states appropriate:

1. ***Next waypoint*** *direction.*
2. ***Traffic light*** *color.*
3. ***Oncoming*** *cars.*
4. *Cars that come from* ***left*** *direction.*
5. *Cars that come from* ***right*** *direction.*

*Because****: Next Waypoint*** *teaches car to choose the right direction.* ***Traffic light*** *teaches car to follow the driving rules (move only on green light).* ***Oncoming, Left, Right*** *teach car to avoid obstacles. Even with low traffic, car crash can lead to serious effects.*

There are N states in total exist for the smartcab in this environment. This number seem reasonable given that the goal of Q-Learning is to learn and make informed decisions about each state because it covers all the rules of the imagined world. Other states like:

1. *Deadline number*
2. *Time*
3. *Amount of steps to destination point*

Do not fit the model because they add a lot of new states in Q-table.

Implement a Q-Learning Driving Agent

QUESTION: *What changes do you notice in the agent's behavior when compared to the basic driving agent when random actions were always taken? Why is this behavior occurring?*

The main change I've noticed is that Q-Learning Driving Agent always goes to destination point purposefully. The second change I've noticed is that Q-Learning Driving Agent makes better decisions after each attempt. This behavior occurring because Agent uses its previous experience while making decision.

Improve the Q-Learning Driving Agent

QUESTION: *Report the different values for the parameters tuned in your basic implementation of Q-Learning. For which set of parameters does the agent perform best? How well does the final driving agent perform?*

QUESTION: *Does your agent get close to finding an optimal policy, i.e. reach the destination in the minimum possible time, and not incur any penalties? How would you describe an optimal policy for this problem?*

I've tuned parameter "Learning Rate" (alpha). I set discount factor equal to 0.9.

Observation 1 *(Learning Rate = 1, Number of trials = 100, Discount factor = 0.9):*

Agent did ~50 actions with negative reward.

Observation 2 *(Learning Rate = 0.5, Number of trials = 100, Discount factor = 0.9):*

Agent did ~45 actions with negative reward.

Observation 3 *(Learning Rate = 0.1, Number of trials = 100, Discount factor = 0.9):*

Agent did ~50 actions with negative reward.

Observation 4 *(Learning Rate = 0.4, Number of trials = 100, Discount factor = 0.9):*

Agent did ~40 actions with negative reward.

Observation 4'Th set of parameters makes the agent perform best.

With each attempt my agent gets closer to finding optimal policy. I think that optimal policy is:

1. *Do not avoid red light.*
2. *Avoid other cars*