In your report, mention what you see in the agent's behavior. Does it eventually make it to the target location?

Number of trials=100. Action=forward. Number of successful trials=9.

The driving agent does reach the destination in some of the trials.

Given (x1, y1) as starting point and (x2, y2) as destination, the cab only reaches its destination if y1 and y2 are same or x1 and x2 are same.

When I ran the simulation with Action=left, there were much lesser trials which were successful.

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?*

I chose all the possible states to update the agent.

The most appropriate states are:

#	Light	Oncoming	Left	Right	Action
1	Green	None	None	None	Forward / Left / Right
2	Green	None	None	Left	Right
3	Green	None	None	Right	None
4	Green	None	None	Forward	None
5	Green	None	Left	None	None
6	Green	None	Right	None	Forward / Right
7	Green	None	Forward	None	None
8	Green	Forward	None	None	Forward / Right
9	Green	Left	None	None	None
10	Green	Right	None	None	Forward / Right
11	Green	Right	None	None	Right
12	Green	Forward	None	None	None
13	Green	! (None)	! (None)	! (None)	None
14	Red	None	None	None	Right / Left
15	Red	None	None	Right	None
16	Red	None	None	Left	None
17	Red	None	None	Forward	None
18	Red	None	Right	None	Forward / Right
19	Red	None	Left	None	Right
20	Red	None	Forward	None	None
21	Red	Forward	None	None	Right
22	Red	Right	None	None	Right
23	Red	Left	None	None	None
24	Red	! (None)	! (None)	! (None)	None

(Note: The directions would be with respect to the traffic we are talking about, ie, if Left='Right' would mean traffic coming from the left is turning to its right)

Light, Oncoming, Left and Right were chosen as at every junction these states must be evaluated to prevent getting penalised when an action is to be taken. It is important that in most cases the agent does follow the traffic rules and doesn't penalised provided it reaches the destination well in time.

So the agent is not penalised when it moves in the forward direction. All these states have been documented according to the driving rules. If the agent follows these rules the cab will reach the destination safely.

I have not considered the deadline in the states as it varies. For longer routes the expected time to cover the distance will be larger than for shorter distances, especially if the cab is required to drive extremely safely without any penalties whatsoever. In case the passenger prefers reaching before the deadline and does not mind incurring some penalties, then some of the states mentioned above can be violated.

The states numbered 13 and 24 have not been used during the simulation as it will increase the time to reach the destination many times. In this environment, it is okay to bear a little penalty, if the cab reaches the destination before the deadline.

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 agent reaches the destination most of the times. Also, the agent does seem to take random actions, as compared to the basic agent which only went in 1 direction that was specified.

The actions are changing because the action having minimum penalty or greatest reward is chosen at every step. Thus the actions don't remain the same, as it was in the case of basic agent.

The agent incurs penalties, which means that sometimes the agent violates the traffic rules, after implementing Q learning. The agent does stop at the red lights as the penalties incurred there are greater than for not reaching deadline.

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?

The table below gives a summary of parameters tuned:

#	Gamma	Alpha	Success Rate (Best of 5 runs)	
			Non Decaying Gamma	Decaying Gamma
1	0.1	0.6	0.74	0.69
2	0.1	0.8	0.7	0.78
3	0.2	0.8	0.72	0.73
4	0.3	0.9	0.67	0.71
5	0.4	0.8	0.68	0.68
6	0.5	0.8	0.7	0.74
7	0.6	0.9	0.69	0.7

In the case of decaying gamma, the agent performs best when learning rate (alpha) is 0.8 and when gamma (discount) is 0.1.

In case of non-decaying gamma, the agent performs best when the learning rate (alpha) is 0.6 and gamma (discount) is 0.6.

Thus the agent is performing better when the discount is decaying over time with the success rate of 0.78, reaching the destination 8 of the last 10 trials.

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?

No, the agent does incur penalties.

An optimal policy is one where the agent follows most of the driving rules (if possible, all) and reaches in time. So there needs to be a balance between the number of times the rules were violated and how many times the agent reaches in time.

For this problem, the optimal policy would be to have the reward positive. So, even though sometimes penalties were incurred, because the agent reaches before deadline it receives an immediate +10 reward and the overall reward would be positive.