# Smart Cab

**Implement a basic driving agent**

Q1: In your report, mention what you see in the agent’s behavior. Does it eventually make it to the target location?

**Identify and update state**

Q2: Justify why you picked these set of states, and how they model the agent and its environment.

**Implementing Q-Learning**

*Q3:* What changes do you notice in the agent’s behavior?

**Enhance the driving agent**

Q4a: Report what changes you made to your basic implementation of Q-Learning to achieve the final version of the agent. How well does it perform?

Q4b: 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?

**Implement a basic driving agent**

The driving agent needs to make a decision about where to go. I will consider 4 different ‘strategies’:

* stubborn: agent always makes a left turn
* random: agent takes a random action out of ‘left’, ‘right’, ‘forward’, or ‘stay’ (no action taken)
* guided: agent always follows the planner if light is green
* qlearner: agent uses reinforcement learning (Q-learning) to find the ‘best’ action to take

Figure 1 shows paths followed by an agent following each of the strategy, whereas Figure 2 shows the cumulative reward, respectively. I fixed the code so that they all start from the same location and have the same destination.

Figure 1

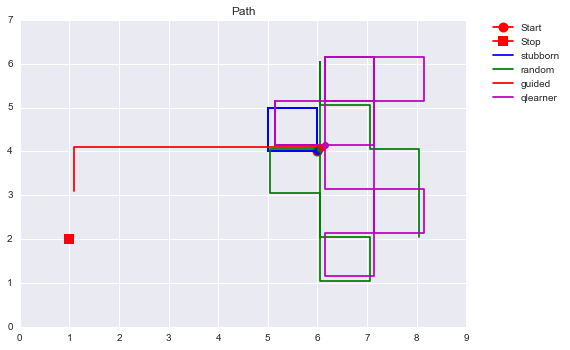
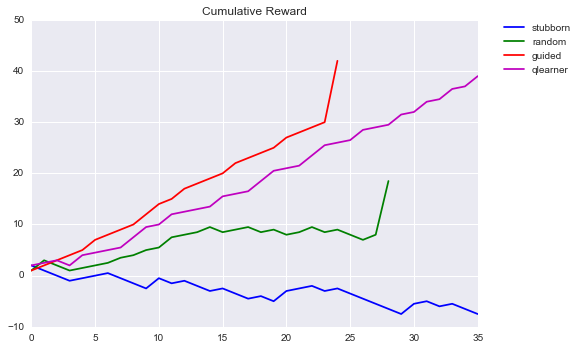


Figure 2



As we can see the stubborn driver keeps looping around without going anywhere. The random driver was actually lucky in this case and ended up at the destination. As a preview, the Q-learner is exploring the space but is not reaching its destination. More comments on this are given below.

For this section I will consider the strategy ‘guided’, which essentially follows the planner:

# Guided

action= None

if inputs['light']=='green':

action = self.next\_waypoint

Running several trials I observe that it normally reaches the destination in time.

**Identify and update state**

The state should be a function of what type of inputs we are sensing from the environment. While distance to the destination, etc. could be useful information I assume these remain unknown and we can only sense the traffic light and oncoming traffic. The simplest definition of state (I can think of) is the color of the traffic light. This makes the ‘state space’ limited to 2.

One could also add a state of representing generically the intersection:

* free: when there is no incoming traffic
* busy: if there is any car, left, right, or incoming

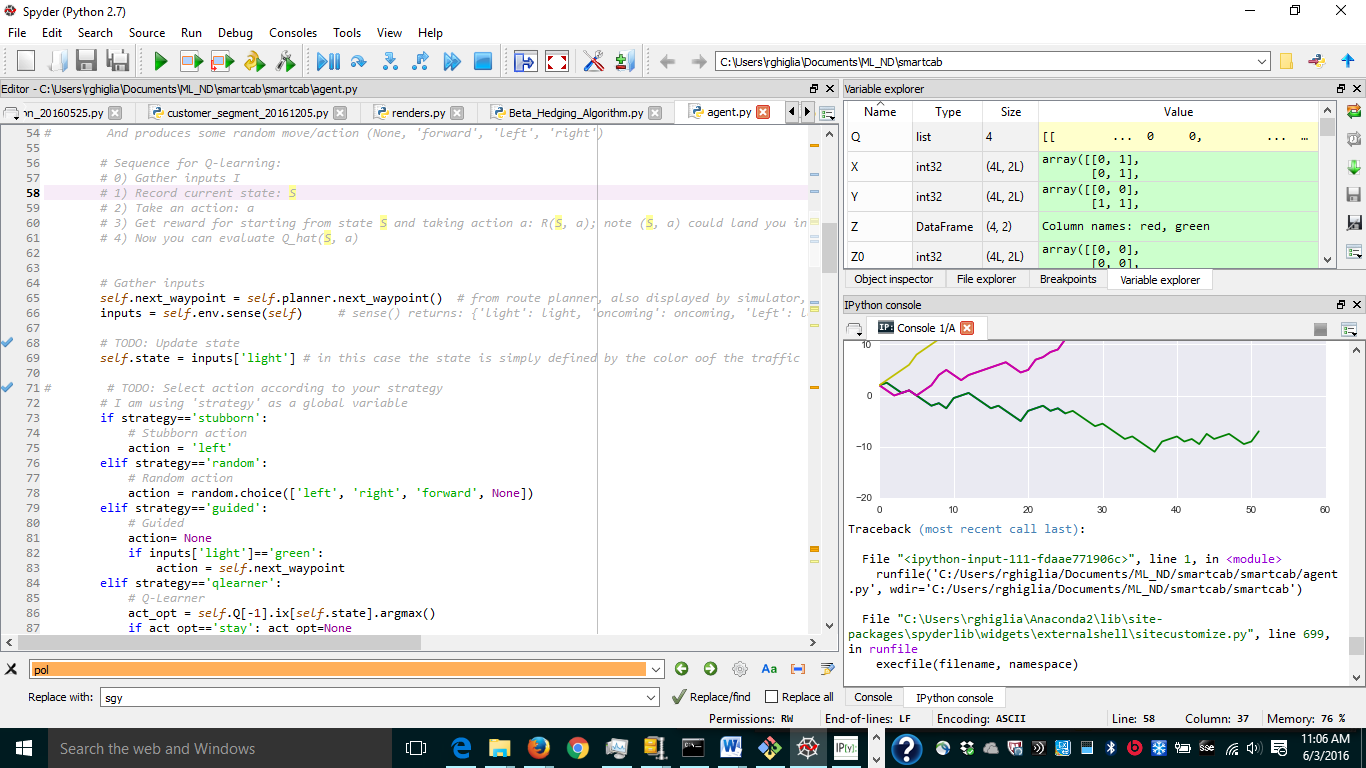
This would make the state space be 4-dimensional: red/green x free/busy. If one wanted to use all the information available the state space would be 2 x 4 = 8-dimensional.

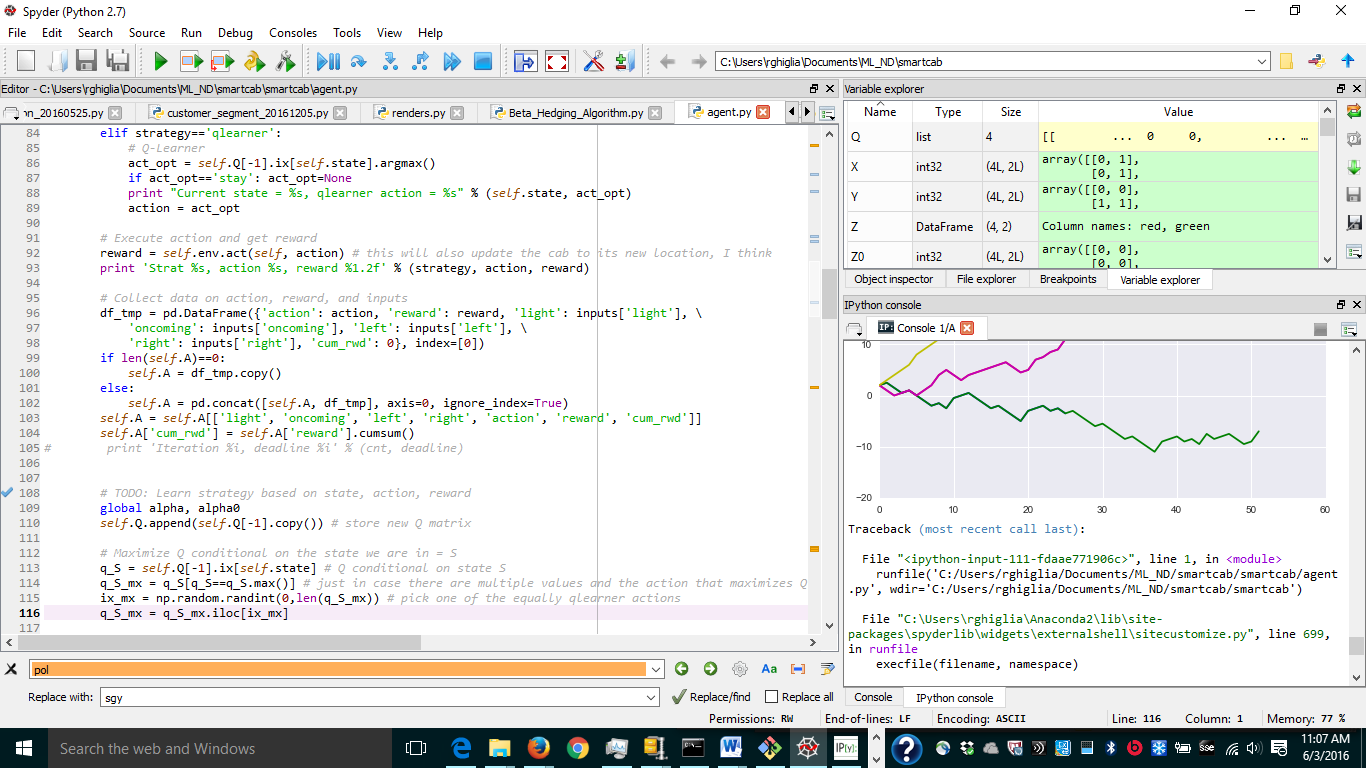
To start, I picked the simplest description of the environment: red or green. This will help building intuition in the algorithm.

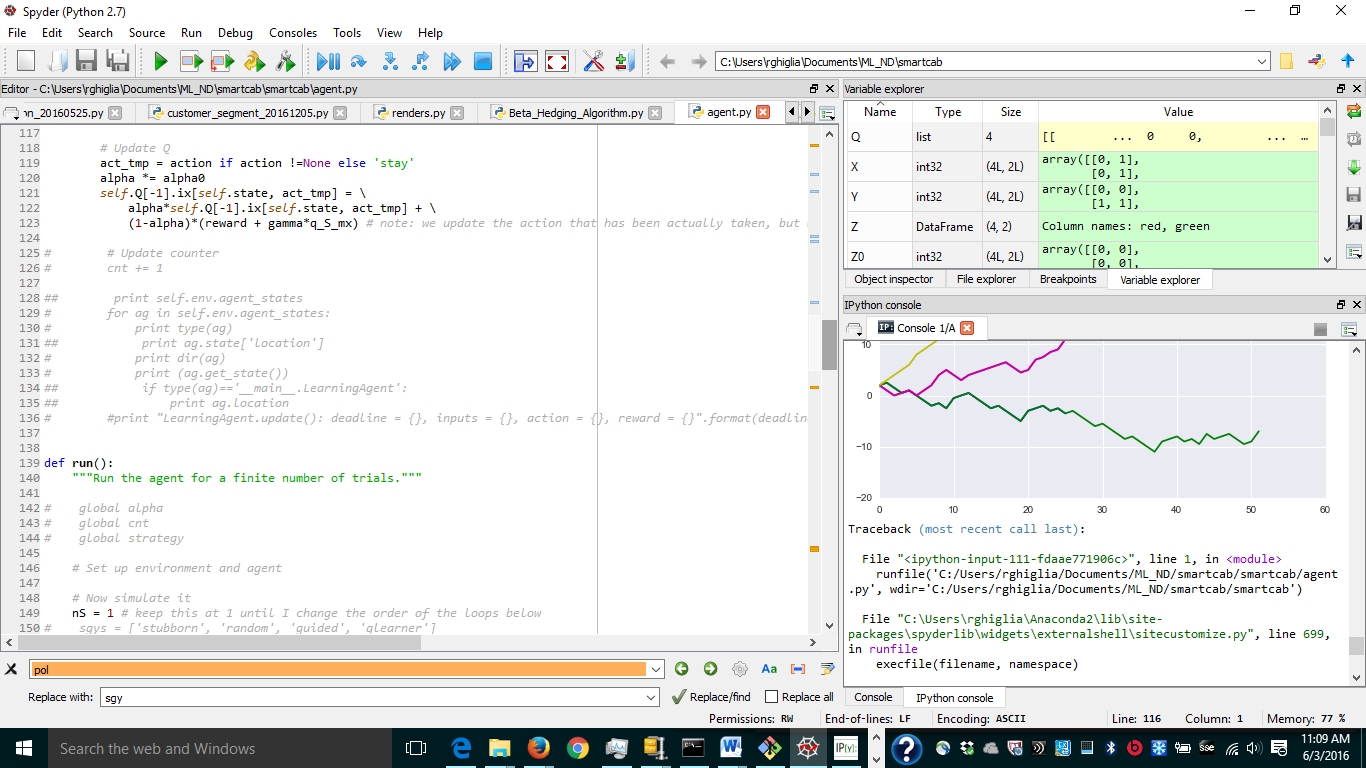
These inputs do not describe the agent though. It seems to me none of the available inputs are descriptive of the agent himself. Examples of that would be the distance to the target. Strictly speaking the latter would also not be describing the agent independently but rather the relation of the agent to the environment. A state that would be strictly relative to the agent could be something like the agent’s health affecting his ability to drive or something. (The questions asks about the ‘agent and its environment’; I not sure if I should interpret it as those in isolation or in combination. The above explanation tries to address both interpretations).

**Implementing Q-Learning**

The ‘guided’ strategy in sense is a bit like cheating assuming that the planner has more global (i.e. extended) information than the agent and can therefore be more effective. The ‘Q-learning’ strategy estimates an optimal combination of state and action by trial and error. It takes actions, gets a reward and incorporates (updates) that information into an estimate of how best to proceed.







As such, the agent wanders around a lot more and the success rate is a lot lower than the ‘guided’ strategy. It is hard to describe the actual behavioral changes without collecting more data. It seems that in the ‘guided’ strategy the agent tries to go on more straight-line paths.

NEED TO REWRITE LOOPS AND GATHER STATS

EXTEND CODE TO INCLUDE MULTIPLE SCENARIOS

PLOT FAN OF PATHS CONDITIONAL ON STRATEGY WITH SAME STARTING AND ENDING POINT

IMPLEMENT A CROSS-GAME LEARNING WHERE YOU INITIALIZE THE Q MATRIX FROM PREVIOUS RIDES

CHANGES TO INITIAL IMPLEMENTATION:

* VARY PARAMETERS: ALPHA AND GAMMA
* USE CROSS-GAME INFO
* EXTEND STATE SPACE?

DISCUSS Q-LEARNING RULE IN AN APPENDIX?