Implement a Basic Driving Agent

*QUESTION 1: 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?*

*Answer – Yes, the **smartcab** is able to make it to the destination but only few times. The interesting observations to note here is that even in the last trials the success rate is totally random i.e. **smartcab** is not able to learn anything for its experience or previous trials .*

The following are the report for the success ratio for the random agent across multiple trials i.e. in this case we run it for 20 times and measured each time its success ratio. To convert following success ratio into success percentage, multiply it by 100.

	success ratio when using random action
1	
2	0.19
3	0.23
4	0.19
5	0.29
6	0.16
7	0.19
8	0.17
9	0.23
10	0.25
11	0.14
12	0.17
13	0.27
14	0.14
15	0.22
16	0.26
17	0.23
18	0.21
19	0.16
20	0.25
21	0.2

Inform the Driving Agent

*QUESTION 2: 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?*

Answer - all_actions = self.env.valid_actions   # i.e. [None, 'forward', 'left', 'right']

traffic_light = ['red','green']   # or, traffic_light = self.env.TrafficLight.valid_states

oncoming, left, right, waypoint = all_actions, all_actions, all_actions, all_actions

state = (traffic_light, oncoming, left, right, waypoint)

Our state comprises of five things:

(1). Traffic light :- 'red' or 'green' i.e., on green light, a left turn is permitted if there is no oncoming traffic making a right turn or coming straight through the intersection. And, on red light, a right turn is permitted if no oncoming traffic is approaching from your left through the intersection. Smartcab uses the state of the traffic light for its direction of movement. Its value can be one of the following : ['red', 'green']

*(2). Oncoming, Left, Right :- **Smartcab** uses it to determine whether there is a vehicle at the intersection for each of the oncoming directions. For each action, the **smartcab** may either idle at the intersection, or drive to the next intersection to the left, right, or ahead of it. There values can be one of the following : [None, 'forward', 'left', 'right']*

*(3). Waypoint :- Since, the **smartcab** has been assigned a route plan based on the passengers' starting location and destination. And, the route is split at each intersection into waypoints, and assuming that the **smartcab**, at any instant, is at some intersection in the world. Therefore, the next waypoint to the destination, assuming the destination has not already been reached, is one intersection away in one direction (North, South, East, or West). And, the waypoint will suggest the **smartcab** that in which direction it should move to reach the Destination. And, if the action taken by the **smartcab** matches with the Waypoint then a positive reward will be given for that particular action in the given state to the **smartcab**. Its value can be one of the following : ['forward', 'left', 'right']*

*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?*

*Answer – In this environment, there are (2*4*4*4*3 = 384) states in total that exist for the **smartcab**. Yes, 384 states seem reasonable given we have total 100 trials and there is high probability that majority of the states will be updated during Q-Learning i.e. updating Q-Table for each state and help in making informed decisions about each state.*

Implement a Q-Learning Driving Agent

QUESTION 3: 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?

Answer – When the primary agent was always taking random actions (i.e. only exploring) then its success ratio of reaching to the destination was small and it was almost never able to find the optimal path to reach the destination but when instead of taking random action every time, it started to take combination of random action and argmax (a) of Q(s,a) i.e. chose that action for which Q(s,a) has maximum value (both exploitation and exploration), it significantly increases the success rate of reaching to the destination and also it was able to find some optimal path for reaching to the destination. Now, this behavior starts to occur due to Q-table i.e. agent starts to learn from the previous action it has taken and the corresponding reward it gained due to that action for a particular state and it saves all these information in Q-table and when next time it encounters any new situation it updates the Q-table and use it further to find which action to take so that its action returns the maximum reward.

Improve the Q-Learning Driving Agent

QUESTION 4: 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?

Answer – Different values for the parameters tuned in our basic implementation of Q-Learning are as follows :-

Note that, here value of success ratio is between 0 and 1 i.e. from worst to best respectively. Also, the optimal path find ratio is between 0 and 1. And to find these in percentage, multiply them with 100.

	alpha	gamma	success ratio	optimal path found ratio	epsilon function
2	0.1	0.1	0.96	0.05	1/math.pow((self.total_num_of_trials)(1/1.1))
3	0.1	0.1	0.98	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
4	0.1	0.1	0.97	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
5	0.1	0.2	0.98	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
6	0.1	0.2	0.95	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
7	0.1	0.2	0.96	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
8	0.1	0.3	0.98	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
9	0.1	0.3	0.95	0.04	1/math.pow((self.total_num_of_trials)(1/1.3))
10	0.1	0.3	0.98	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
11	0.1	0.4	0.97	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
12	0.1	0.4	0.94	0	1/math.pow((self.total_num_of_trials)(1/1.3))
13	0.1	0.4	0.98	0.03	1/math.pow((self.total_num_of_trials)(1/1.5))
14	0.1	0.5	0.96	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
15	0.1	0.5	0.97	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
16	0.1	0.5	0.92	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
17	0.1	0.6	0.99	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
18	0.1	0.6	0.93	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
19	0.1	0.6	0.95	0.03	1/math.pow((self.total_num_of_trials)(1/1.5))
20	0.1	0.7	0.85	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
21	0.1	0.7	0.93	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
22	0.1	0.7	0.93	0	1/math.pow((self.total_num_of_trials)(1/1.5))
23	0.1	0.8	0.92	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
24	0.1	0.8	0.88	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
25	0.1	0.8	0.92	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
26	0.1	0.9	0.93	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))

Note here that the self.total_num_of_trials is not a constant variable rather its value will increase from one with each iteration with a trial. And at the beginning of each trial, its value will be reset to one.

27	0.1	0.9	0.61	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
28	0.1	0.9	0.91	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
29	0.2	0.1	0.96	0.08	1/math.pow((self.total_num_of_trials)(1/1.1))
30	0.2	0.1	0.96	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
31	0.2	0.1	0.99	0.04	1/math.pow((self.total_num_of_trials)(1/1.5))
32	0.2	0.2	0.98	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
33	0.2	0.2	0.96	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
34	0.2	0.2	0.97	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
35	0.2	0.3	1	0.05	1/math.pow((self.total_num_of_trials)(1/1.1))
36	0.2	0.3	0.98	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
37	0.2	0.3	0.96	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
38	0.2	0.4	0.98	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
39	0.2	0.4	0.97	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
40	0.2	0.4	0.96	0	1/math.pow((self.total_num_of_trials)(1/1.5))
41	0.2	0.5	0.97	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
42	0.2	0.5	0.97	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
43	0.2	0.5	0.92	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
44	0.2	0.6	0.99	0	1/math.pow((self.total_num_of_trials)(1/1.1))
45	0.2	0.6	0.94	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
46	0.2	0.6	0.89	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
47	0.2	0.7	0.75	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
48	0.2	0.7	0.92	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
49	0.2	0.7	0.8	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
50	0.2	0.8	0.94	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
51	0.2	0.8	0.93	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
52	0.2	0.8	0.89	0	1/math.pow((self.total_num_of_trials)(1/1.5))

53	0.2	0.9	0.92	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
54	0.2	0.9	0.93	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
55	0.2	0.9	0.92	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
56	0.3	0.1	0.95	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
57	0.3	0.1	0.95	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
58	0.3	0.1	0.97	0	1/math.pow((self.total_num_of_trials)(1/1.5))
59	0.3	0.2	0.97	0.05	1/math.pow((self.total_num_of_trials)(1/1.1))
60	0.3	0.2	0.97	0	1/math.pow((self.total_num_of_trials)(1/1.3))
61	0.3	0.2	0.98	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
62	0.3	0.3	0.98	0	1/math.pow((self.total_num_of_trials)(1/1.1))
63	0.3	0.3	0.96	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
64	0.3	0.3	0.94	0	1/math.pow((self.total_num_of_trials)(1/1.5))
65	0.3	0.4	0.89	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
66	0.3	0.4	0.97	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
67	0.3	0.4	0.97	0	1/math.pow((self.total_num_of_trials)(1/1.5))
68	0.3	0.5	0.93	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
69	0.3	0.5	0.87	0.04	1/math.pow((self.total_num_of_trials)(1/1.3))
70	0.3	0.5	0.94	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
71	0.3	0.6	0.97	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
72	0.3	0.6	0.89	0	1/math.pow((self.total_num_of_trials)(1/1.3))
73	0.3	0.6	0.95	0.03	1/math.pow((self.total_num_of_trials)(1/1.5))
74	0.3	0.7	0.93	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
75	0.3	0.7	0.85	0.05	1/math.pow((self.total_num_of_trials)(1/1.3))
76	0.3	0.7	0.92	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
77	0.3	0.8	0.85	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
78	0.3	0.8	0.93	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))

79	0.3	0.8	0.89	0.03	1/math.pow((self.total_num_of_trials)(1/1.5))
80	0.3	0.9	0.97	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
81	0.3	0.9	0.74	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
82	0.3	0.9	0.85	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
83	0.4	0.1	0.99	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
84	0.4	0.1	0.97	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
85	0.4	0.1	0.9	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
86	0.4	0.2	1	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
87	0.4	0.2	0.96	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
88	0.4	0.2	0.99	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
89	0.4	0.3	1	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
90	0.4	0.3	0.99	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
91	0.4	0.3	0.96	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
92	0.4	0.4	0.97	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
93	0.4	0.4	0.98	0	1/math.pow((self.total_num_of_trials)(1/1.3))
94	0.4	0.4	0.94	0	1/math.pow((self.total_num_of_trials)(1/1.5))
95	0.4	0.5	0.91	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
96	0.4	0.5	0.97	0.04	1/math.pow((self.total_num_of_trials)(1/1.3))
97	0.4	0.5	0.96	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
98	0.4	0.6	0.95	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
99	0.4	0.6	0.95	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
100	0.4	0.6	0.94	0.03	1/math.pow((self.total_num_of_trials)(1/1.5))
101	0.4	0.7	0.96	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
102	0.4	0.7	0.92	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
103	0.4	0.7	0.86	0	1/math.pow((self.total_num_of_trials)(1/1.5))
104	0.4	0.8	0.89	0.06	1/math.pow((self.total_num_of_trials)(1/1.1))

105	0.4	0.8	0.83	0	1/math.pow((self.total_num_of_trials)(1/1.3))
106	0.4	0.8	0.91	0	1/math.pow((self.total_num_of_trials)(1/1.5))
107	0.4	0.9	0.87	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
108	0.4	0.9	0.85	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
109	0.4	0.9	0.81	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
110	0.5	0.1	1	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
111	0.5	0.1	0.88	0	1/math.pow((self.total_num_of_trials)(1/1.3))
112	0.5	0.1	0.97	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
113	0.5	0.2	0.99	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
114	0.5	0.2	0.99	0	1/math.pow((self.total_num_of_trials)(1/1.3))
115	0.5	0.2	0.93	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
116	0.5	0.3	0.97	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
117	0.5	0.3	0.99	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
118	0.5	0.3	0.96	0	1/math.pow((self.total_num_of_trials)(1/1.5))
119	0.5	0.4	0.99	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
120	0.5	0.4	0.96	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
121	0.5	0.4	0.93	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
122	0.5	0.5	0.97	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
123	0.5	0.5	0.95	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
124	0.5	0.5	0.91	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
125	0.5	0.6	0.88	0.05	1/math.pow((self.total_num_of_trials)(1/1.1))
126	0.5	0.6	0.94	0	1/math.pow((self.total_num_of_trials)(1/1.3))
127	0.5	0.6	0.9	0	1/math.pow((self.total_num_of_trials)(1/1.5))
128	0.5	0.7	0.92	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
129	0.5	0.7	0.88	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
130	0.5	0.7	0.89	0.03	1/math.pow((self.total_num_of_trials)(1/1.5))

131	0.5	0.8	0.9	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
132	0.5	0.8	0.91	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
133	0.5	0.8	0.89	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
134	0.5	0.9	0.88	0	1/math.pow((self.total_num_of_trials)(1/1.1))
135	0.5	0.9	0.67	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
136	0.5	0.9	0.77	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
137	0.6	0.1	0.99	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
138	0.6	0.1	0.98	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
139	0.6	0.1	0.96	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
140	0.6	0.2	0.99	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
141	0.6	0.2	0.96	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
142	0.6	0.2	0.99	0	1/math.pow((self.total_num_of_trials)(1/1.5))
143	0.6	0.3	0.97	0.05	1/math.pow((self.total_num_of_trials)(1/1.1))
144	0.6	0.3	0.99	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
145	0.6	0.3	0.96	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
146	0.6	0.4	0.97	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
147	0.6	0.4	0.94	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
148	0.6	0.4	0.97	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
149	0.6	0.5	0.95	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
150	0.6	0.5	0.93	0	1/math.pow((self.total_num_of_trials)(1/1.3))
151	0.6	0.5	0.98	0.04	1/math.pow((self.total_num_of_trials)(1/1.5))
152	0.6	0.6	0.9	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
153	0.6	0.6	0.9	0	1/math.pow((self.total_num_of_trials)(1/1.3))
154	0.6	0.6	0.95	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
155	0.6	0.7	0.89	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
156	0.6	0.7	0.93	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))

157	0.6	0.7	0.92	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
158	0.6	0.8	0.84	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
159	0.6	0.8	0.86	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
160	0.6	0.8	0.8	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
161	0.6	0.9	0.76	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
162	0.6	0.9	0.7	0	1/math.pow((self.total_num_of_trials)(1/1.3))
163	0.6	0.9	0.72	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
164	0.7	0.1	0.94	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
165	0.7	0.1	0.98	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
166	0.7	0.1	0.96	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
167	0.7	0.2	0.99	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
168	0.7	0.2	0.98	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
169	0.7	0.2	0.99	0.03	1/math.pow((self.total_num_of_trials)(1/1.5))
170	0.7	0.3	0.96	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
171	0.7	0.3	0.97	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
172	0.7	0.3	0.96	0.04	1/math.pow((self.total_num_of_trials)(1/1.5))
173	0.7	0.4	0.94	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
174	0.7	0.4	0.97	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
175	0.7	0.4	0.98	0	1/math.pow((self.total_num_of_trials)(1/1.5))
176	0.7	0.5	0.98	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
177	0.7	0.5	0.93	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
178	0.7	0.5	0.96	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
179	0.7	0.6	0.93	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
180	0.7	0.6	0.96	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
181	0.7	0.6	0.94	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
182	0.7	0.7	0.86	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))

183	0.7	0.7	0.89	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
184	0.7	0.7	0.91	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
185	0.7	0.8	0.82	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
186	0.7	0.8	0.89	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
187	0.7	0.8	0.78	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
188	0.7	0.9	0.77	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
189	0.7	0.9	0.62	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
190	0.7	0.9	0.8	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
191	0.8	0.1	0.97	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
192	0.8	0.1	0.93	0.04	1/math.pow((self.total_num_of_trials)(1/1.3))
193	0.8	0.1	0.95	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
194	0.8	0.2	1	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
195	0.8	0.2	0.98	0.05	1/math.pow((self.total_num_of_trials)(1/1.3))
196	0.8	0.2	0.93	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
197	0.8	0.3	0.96	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
198	0.8	0.3	0.96	0	1/math.pow((self.total_num_of_trials)(1/1.3))
199	0.8	0.3	0.95	0	1/math.pow((self.total_num_of_trials)(1/1.5))
200	0.8	0.4	0.95	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
201	0.8	0.4	0.96	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
202	0.8	0.4	0.95	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
203	0.8	0.5	1	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
204	0.8	0.5	0.93	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
205	0.8	0.5	0.97	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
206	0.8	0.6	0.97	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
207	0.8	0.6	0.99	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
208	0.8	0.6	0.93	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))

209	0.8	0.7	0.95	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
210	0.8	0.7	0.91	0.04	1/math.pow((self.total_num_of_trials)(1/1.3))
211	0.8	0.7	0.9	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
212	0.8	0.8	0.84	0.03	1/math.pow((self.total_num_of_trials)(1/1.1))
213	0.8	0.8	0.82	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
214	0.8	0.8	0.8	0	1/math.pow((self.total_num_of_trials)(1/1.5))
215	0.8	0.9	0.66	0	1/math.pow((self.total_num_of_trials)(1/1.1))
216	0.8	0.9	0.78	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
217	0.8	0.9	0.71	0	1/math.pow((self.total_num_of_trials)(1/1.5))
218	0.9	0.1	0.96	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
219	0.9	0.1	0.99	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
220	0.9	0.1	0.95	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
221	0.9	0.2	0.98	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
222	0.9	0.2	0.97	0	1/math.pow((self.total_num_of_trials)(1/1.3))
223	0.9	0.2	0.98	0.05	1/math.pow((self.total_num_of_trials)(1/1.5))
224	0.9	0.3	0.96	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
225	0.9	0.3	0.96	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
226	0.9	0.3	0.98	0.03	1/math.pow((self.total_num_of_trials)(1/1.5))
227	0.9	0.4	0.94	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
228	0.9	0.4	0.96	0	1/math.pow((self.total_num_of_trials)(1/1.3))
229	0.9	0.4	0.96	0	1/math.pow((self.total_num_of_trials)(1/1.5))
230	0.9	0.5	0.98	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
231	0.9	0.5	0.95	0	1/math.pow((self.total_num_of_trials)(1/1.3))
232	0.9	0.5	0.98	0.04	1/math.pow((self.total_num_of_trials)(1/1.5))
233	0.9	0.6	0.99	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
234	0.9	0.6	0.96	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))

235	0.9	0.6	1	0.01	1/math.pow((self.total_num_of_trials)(1/1.5))
236	0.9	0.7	0.88	0.04	1/math.pow((self.total_num_of_trials)(1/1.1))
237	0.9	0.7	0.93	0.03	1/math.pow((self.total_num_of_trials)(1/1.3))
238	0.9	0.7	0.86	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
239	0.9	0.8	0.92	0.02	1/math.pow((self.total_num_of_trials)(1/1.1))
240	0.9	0.8	0.89	0.01	1/math.pow((self.total_num_of_trials)(1/1.3))
241	0.9	0.8	0.78	0.02	1/math.pow((self.total_num_of_trials)(1/1.5))
242	0.9	0.9	0.76	0.01	1/math.pow((self.total_num_of_trials)(1/1.1))
243	0.9	0.9	0.81	0.02	1/math.pow((self.total_num_of_trials)(1/1.3))
244	0.9	0.9	0.79	0	1/math.pow((self.total_num_of_trials)(1/1.5))

Now, we have defined the value of epsilon as ,

Epsilon (ϵ) = 1/math.pow(total_number_of_trials_until_now, 1/(either 1.1 or 1.3 or 1.5))

Here, we tried different functions for epsilon like , f(x) = 1/log(x) or, f(x) = 1/sqrt(x) and several others but included in the above chart, only few combinations due to the length of the chart. Now, among these we found the combinations of (alpha, gamma, epsilon) which give almost 100% success ratio are as follows :-

(0.2, 0.3, 1/math.pow(total_number_of_trials_until_now, 1/1.1)) ,

(0.4, 0.2, 1/math.pow(total_number_of_trials_until_now, 1/1.1)) ,

(0.4, 0.3, 1/math.pow(total_number_of_trials_until_now, 1/1.1)) ,

(0.5, 0.1, 1/math.pow(total_number_of_trials_until_now, 1/1.1)) ,

(0.8, 0.2, 1/math.pow(total_number_of_trials_until_now, 1/1.1)) ,

(0.8, 0.5, 1/math.pow(total_number_of_trials_until_now, 1/1.1)) ,

(0.9, 0.6, 1/math.pow(total_number_of_trials_until_now, 1/1.5)) .

The final driving agent performs very well as it's success ratio is above 90% most of the time.

QUESTION 5: 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?

Answer - Yes, primary agent is able to find an optimal policy but only few times and probability of finding optimal policy is more in the end trials. Because, more weightage to exploration and less weightage to exploitation assigned in the initial trials and the reverse in the end trials. Now, I have defined the optimal policy for the agent as if it is able to not getting any negative reward or penalty for its actions during its journey to the destination and it is able to reach to the destination in less than or equal steps as the Manhattan distance between the destination and the initial location of the primary agent, which is calculated by the function compute_dist(start, destination) in the Agent.py .