# Project Report – SmartCab

## Implement Basic Driving Agents

***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?*

Considering that there is no deadline and the agent acts randomly the it can eventually make to the destination or not depends on luck. Since everything is random, the agent does not respect any traffic laws and does not learn in any way.

## 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?*

I chose the states based on the combination of inputs. There are 5 inputs to be taken into consideration: *"light", "oncoming", "left","right","next_waypoint"*. The combinations of these inputs will generate states that contribute to the agent getting to the destination. When implemented with the Q-learning algorithm, the agent will "associate" following the "*next_waypoint*" to positive rewards and disrespecting traffic laws to negative rewards.

***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?*

The total number of states is **384**. This is due to the fact that each state is a combination of inputs and there are a total of 384 possible combinations. This number seems a reasonable number of states to learn because we will do around 100 trials with around 2000 interations in total. Also, considering the low number of dummy agents in the problem most of the states will not be encountered since three inputs are related to encountering dummy agents in the intersection.

## 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 driving agent begins to improve in relation to the basic driving agent with random actions. It starts to respect traffic laws and follow the "next_waypoint" input. This behavior occurs because the driving agent now takes into consideration the Q-values to make decisions.

These Q-values are updated with the rewards gained in each state. So the learning agent will almost always follow the action with the highest Q-value. When there are actions with the same Q-value, it chooses a random one between them.

## 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?*

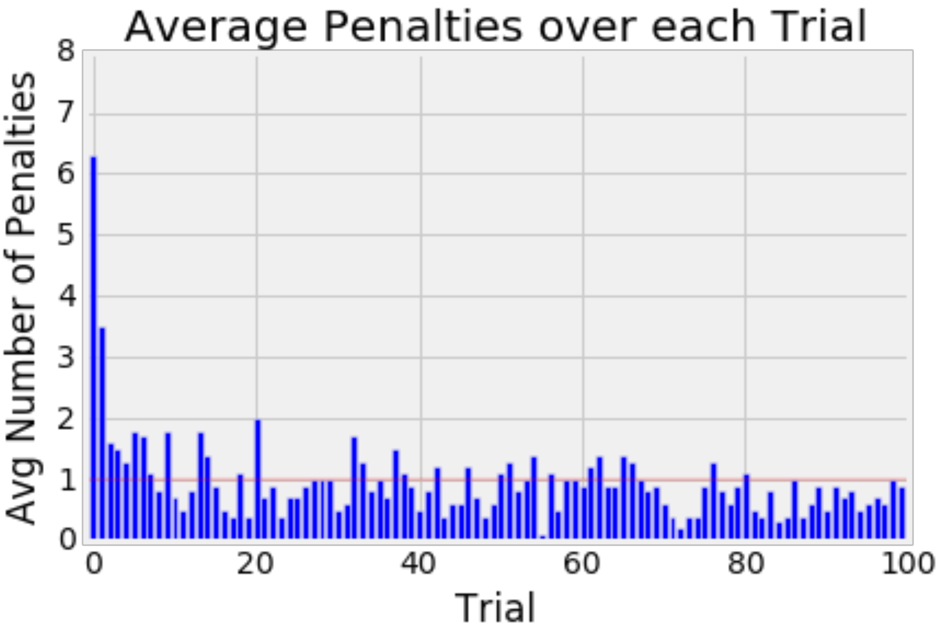
There were 3 parameters to be tuned: $\alpha$, $\varepsilon$, e $\gamma$. The $\alpha$ represents the learning rate in which the agent will learn. The $\varepsilon$ is the exploration rate in which the agent will choose not to follow the highest Q-valued action and choose between 2 random best actions. The $\gamma$ is the discount rate that represents how much of the future reward is to be counted in this action. I chose to tune the parameters in 10 different ways. Those ways and the results are in the table bellow.

| alpha | Gamma | Epsilon | Success Rate |
|---|---|---|---:|
| 0.5 | 0.8 | 0.1 | 0,96 |
| 0.8 | 0.8 | 0.1 | 0,946 |
| 0.3 | 0.8 | 0.1 | 0,962 |
| 0.5 | 0.3 | 0.1 | 0,958 |
| 0.5 | 0.8 | 0.3 | 0,866 |
| 0.5 | 0.5 | 0.05 | 0,956 |
| 0.5 | 0.8 | 0.05 | 0,95 |
| 0.5 | 0.5 | 0.1 | 0,97 |
| 0.8 | 0.5 | 0.05 | 0,96 |
| 0.3 | 0.1 | 0.1 | 0,962 |

The highest success rate was 97% on average and with the parameters [0.5, 0.5, 0.1]. The results show an average learning rate, an average discount rate and a 10% exploration rate. In 100 trials the agent found his destination an average of 97 times.

***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?*

The optimal policy for this problem is to follow the “GPS” or “next_waypoint” input to get to the destination and do not break any traffic laws. Describing the optimal policy this way, my learning agent is very close to finding an optimal policy.



During 10 rounds of testing with 100 trials each, I calculated the number of penalties that occurred in each trial. The plot above shows that the number of penalties is at most 2 penalties per trial after a couple of trials. After the 80th trial it averages to less than 1 penalty per trial.

These penalties that still occur will only disappear if the model is trained with a higher number of trials. The number of dummy agents only amounts to 3 so it's impossible for the learning agent to visit every state with only 100 trials.