UDACITY

PROJECT SPECIFICATION

Train a Smartcab to Drive

Getting Started

CRITERIA	MEETS SPECIFICATIONS
Understand the World	Student provides a thorough discussion of the driving agent as it interacts with the environment.
Understand the Code	Student correctly addresses the questions posed about the code as addressed in Question 2 of the notebook.

Implement a Basic Driving Agent

CRITERIA	MEETS SPECIFICATIONS
Basic Agent • *Agent accepts inputs* • *Produces a valid output* • *Runs in simulator*	Driving agent produces a valid action when an action is required. Rewards and penalties are received in the simulation by the driving agent in accordance with the action taken.
Basic Agent Simulation Analysis	Student summarizes observations about the basic driving agent and its behavior. Optionally, if a visualization is included, analysis is conducted on the results provided.

Inform the Driving Agent

CRITERIA	MEETS SPECIFICATIONS
Identify States	Student justifies a set of features that best model each state of the driving agent in the environment. Unnecessary features not included in the state (if applicable) are similarly justified. Students argument in notebook (Q4) must match state in agent.py code.
State Space	The total number of possible states is correctly reported. The student discusses whether the driving agent could learn a feasible policy within a reasonable number of trials for the given state space.

CRITERIA	MEETS SPECIFICATIONS
Update Driving Agent State	The driving agent successfully updates its state based on the state definition and input provided.

Implement a Q-Learning Driving Agent

CRITERIA	MEETS SPECIFICATIONS
Q-Learning Agent • *Agent updates Q-values* • *Picks the best action* • *Agent correctly implements a "tie-breaker" between best actions* • *Student correct implements eplison decay and exploration (agent takes a random action with epsilon probability)* • *Student implements required 'learning' flags*	The driving agent: (1) Chooses best available action from the set of Q-values for a given state. (2) Implements a 'tie-breaker' between best actions correctly (3)Updates a mapping of Q-values for a given state correctly while considering the learning rate and the reward or penalty received. (4) Implements exploration with epsilon probability (5) implements are required 'learning' flags correctly
Q-Learning Agent Simulation Analysis	Student summarizes observations about the initial/default Q-Learning driving agent and its behavior, and compares them to the observations made about the basic agent. If a visualization is included, analysis is conducted on the results provided.

Improve the Q-Learning Driving Agent

CRITERIA	MEETS SPECIFICATIONS
Improved Q-Learning Agent • *Improvements reported*	The driving agent performs Q-Learning with alternative parameters or schemes beyond the initial/default implementation.
Improved Q-Learning Agent Simulation Analysis • *Improvements reported*	Student summarizes observations about the optimized Q-Learning driving agent and its behavior, and further compares them to the observations made about the initial/default Q-Learning driving agent. If a visualization is included, analysis is conducted on the results provided.

CRITERIA	MEETS SPECIFICATIONS
Safety and Reliability	The driving agent is able to safely and reliably guide the *Smartcab* to the destination before the deadline.
Define an Optimal Policy	Student describes what an optimal policy for the driving agent would be for the given environment. The policy of the improved Q- Learning driving agent is compared to the stated optimal policy. Student presents entries from the learned Q-table that demonstrate the optimal policy and sub- optimal policies. If either are missing, discussion is made as to why.
[OPTIONAL] Future Rewards	Student correctly identifies the two characteristics about the project that invalidate the use of future rewards in the Q-Learning implementation.

Student FAQ