Meets Specifications

congrats!!. Hope a you learned a lot. Reinforcement Learning is an evolving field, lot of research is going on. Here are few resources for further study.

[Q-Learning with Neural Networks](https://medium.com/emergent-future/simple-reinforcement-learning-with-tensorflow-part-0-q-learning-with-tables-and-neural-networks-d195264329d0)  
Last year, I attended Deep RL Boot camp in University of California, Berkeley. Lecture videos, in case you are interested are [here](https://sites.google.com/view/deep-rl-bootcamp/lectures)

Keep up your good work!

**Getting Started**

**Student provides a thorough discussion of the driving agent as it interacts with the environment.**

All required files are found. Good observation about the rewards.

**Student correctly addresses the questions posed about the code as addressed in Question 2 of the notebook.**

correct

**Implement a Basic Driving Agent**

**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.**

Awesome

Agent takes random actions when Learning is false.

**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**

**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.**

Nice analysis and justification of the set of features you've chosen to include in your default agent!

**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.**

Good job here. The calculation for the number of possible states is simply the combination of all outcomes of each feature multiplied together.  
Your calculation of dimension is correct. 96 states is a reasonable number of states to learn in a with a good number of training trials and a good epsilon decay rate.

**The driving agent successfully updates its state based on the state definition and input provided.**

**Implement a Q-Learning Driving Agent**

**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**

**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**

**The driving agent performs Q-Learning with alternative parameters or schemes beyond the initial/default implementation.**

**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.**

Nice job of getting this cab up to A+/A ratings!

**The driving agent is able to safely and reliably guide the *Smartcab* to the destination before the deadline.**

Excellent work achieving a grade of A+ on safety  There is room to improve reliability to A+

**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.**