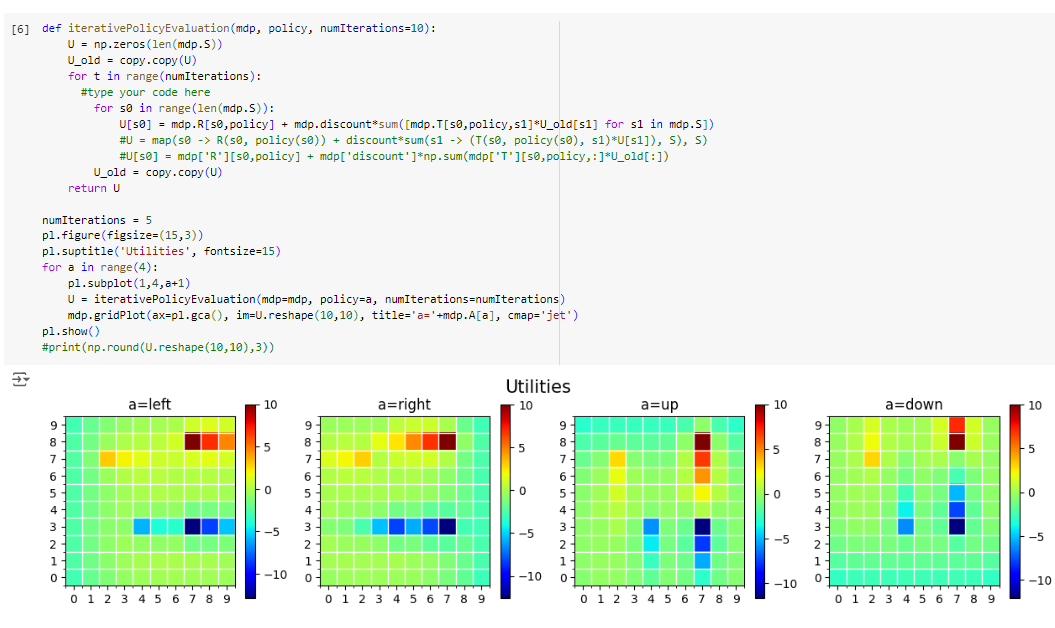
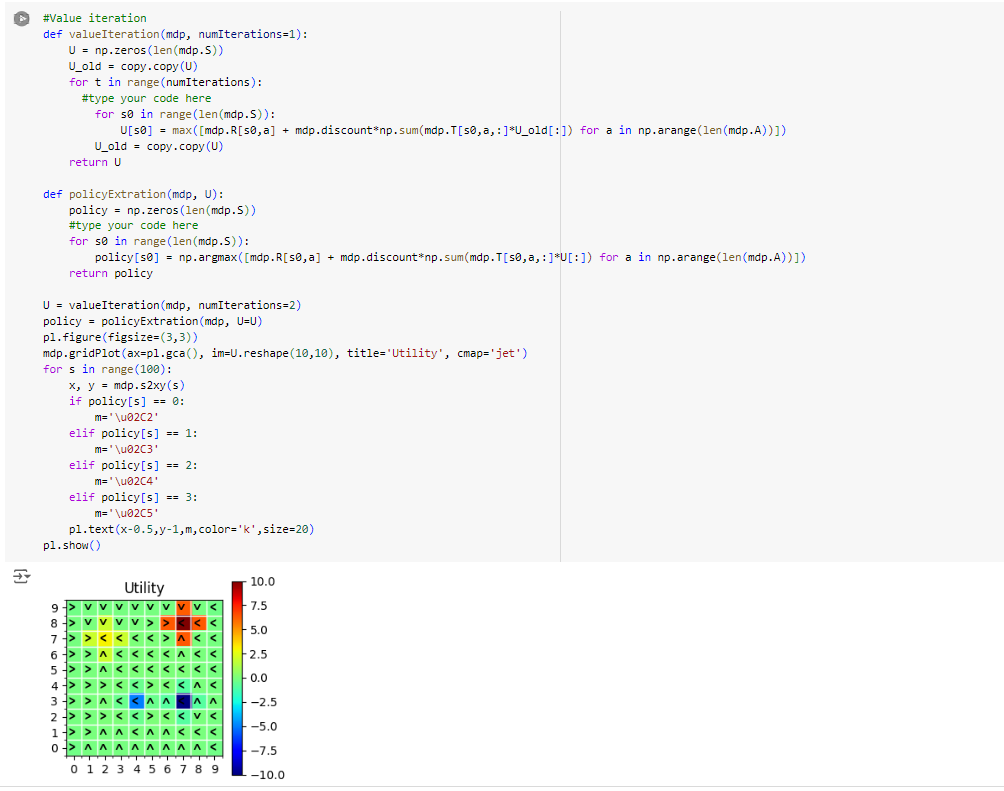
GitHub repository link [https://github.com/](https://github.com/ThaminduSulakshana/DL-Lab-IT21206078/tree/b4bef0d728fab344f7ab9fcf97365b5562c2ef8b/lab-6)IT21343520/lab-8

# Question 1: Markov Decision Process and Q-Learning.

## Markov\_Decision\_Process.ipynb







## Gridworld.ipynb

## 

# Question 2: Model-Based vs Model-Free Reinforcement Learning.

Implement the Model-Based Approach (Value Iteration)

A screenshot of a computer

Description automatically generated

Implement the Model-Free Approach (Q-Learning) A screenshot of a computer code

Description automatically generated

Measure Execution Time and Convergence

A white rectangular object with text

Description automatically generated

A screenshot of a graph

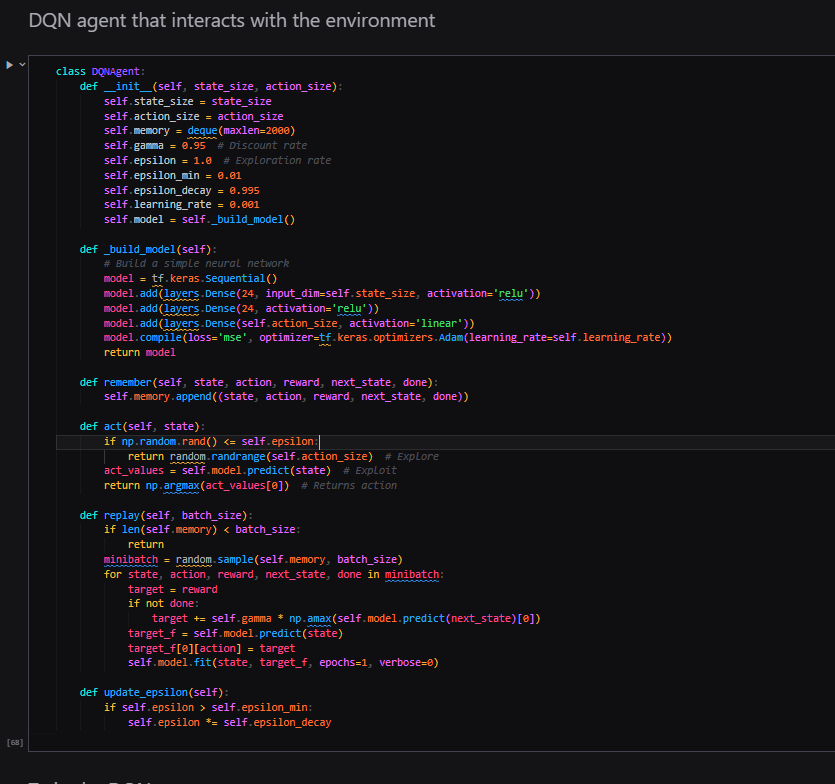
Description automatically generated

**Model-Based algorithms** use a model of the environment to plan and simulate future states, making them faster to converge but dependent on having accurate knowledge of the environment's dynamics. In contrast,

**Model-Free algorithms** learn directly from interaction with the environment, without needing a model, which makes them more flexible but slower to converge as they rely on trial and error to learn optimal actions.

# Question 3: Introduction to Deep Q-Learning (DQN)

DQN agent that interacts with the environment.



Train the DQN agent.

A screenshot of a computer program

Description automatically generated

Plotting Results for Different Epsilon Values.

A screen shot of a graph

Description automatically generated

The Deep Q-Learning (DQN) implementation demonstrates the integration of Q-Learning with deep neural networks to effectively approximate Q-values in complex environments. By utilizing an epsilon-greedy strategy for action selection, the model balances exploration and exploitation, adjusting its learning behavior based on different epsilon values (0.1, 0.5, 0.9). The architecture consists of a neural network with two hidden layers, which is suitable for the CartPole environment. Additionally, experience replay is employed to stabilize training by sampling past experiences. The implementation reduces training duration by limiting episodes to 100, showcasing key reinforcement learning concepts and allowing for quick iterations. The results illustrate how varying exploration strategies impact the agent's performance and learning efficiency, providing a foundation for further exploration in reinforcement learning.