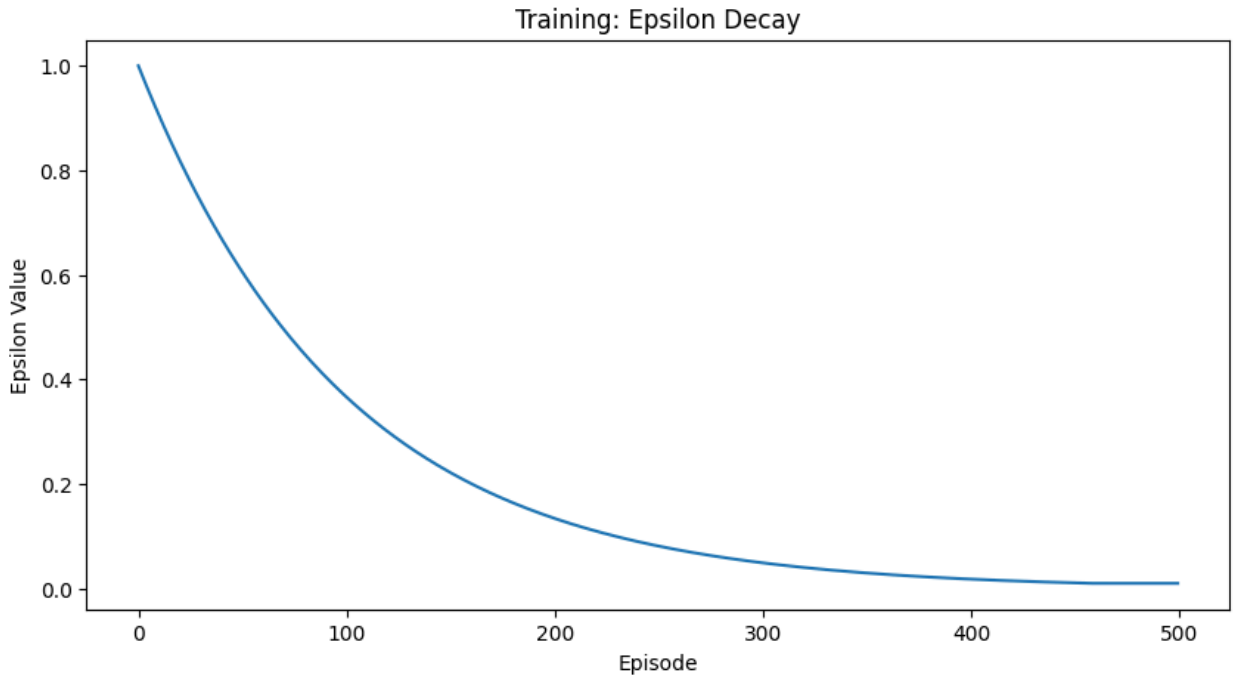
Applying Q-Learning to the Stock Trading Problem

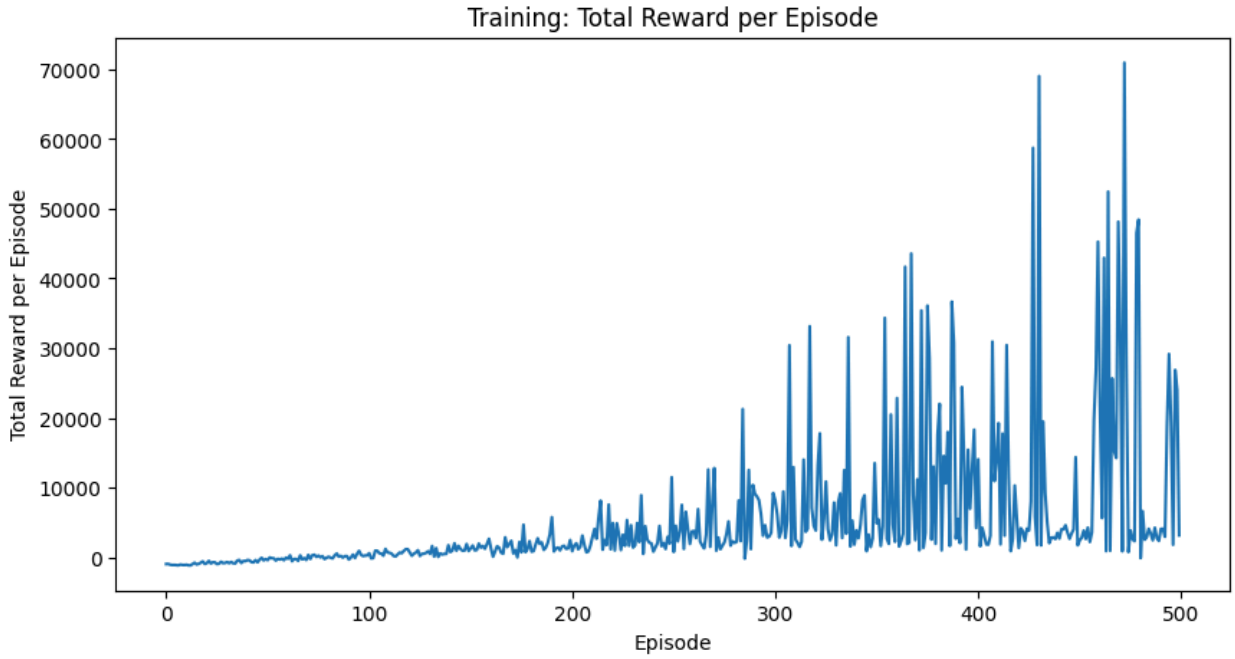
Training Phase

Epsilon Decay:



The epsilon decay plot shows the exploration rate (ϵ) decreasing exponentially over 500 episodes. Initially, the agent explores the environment extensively with a high epsilon value (close to 1). Over time, as learning progresses, epsilon decreases and the agent increasingly exploits the learned policy by taking more greedy actions.

Total Reward per Episode:



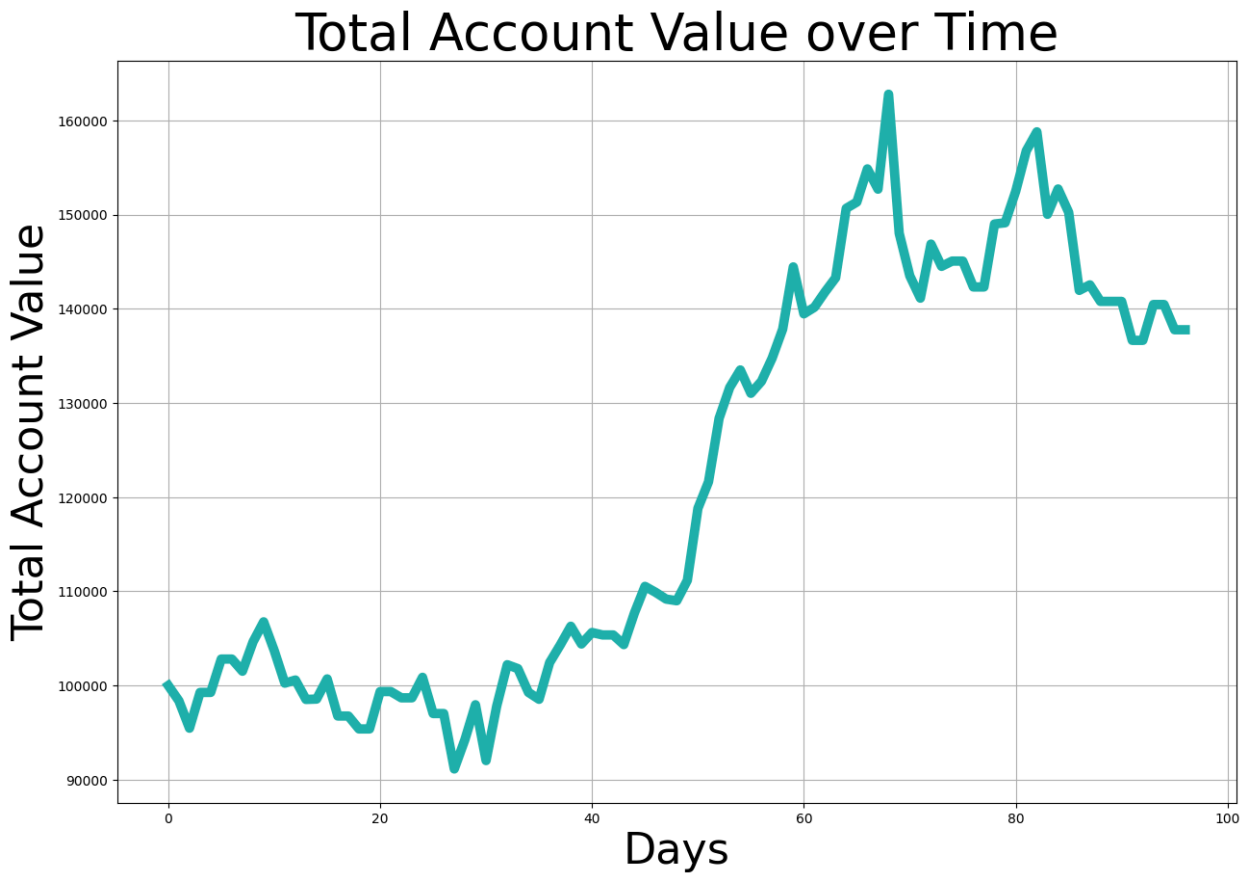
The total reward per episode increases significantly as training progresses. Early episodes show low rewards due to random exploration and lack of knowledge about the environment. However, as the agent learns an optimal policy, rewards improve, with noticeable spikes in later episodes. This indicates that the agent has successfully learned a strategy to maximize returns.

(The **number_of_days_to_consider** hyperparameter is set to **8** for both training and testing the agent, as this value produced the best results.)

Evaluation Phase

Account Value Over Time:

The evaluation plot demonstrates the agent's performance when trained and tested using a greedy policy (i.e., selecting actions with maximum expected rewards). The total account value starts at around $100,000 and shows consistent growth over time, peaking above $160,000 before experiencing some fluctuations.



Despite some volatility, the overall trend is positive, indicating that the Q-learning algorithm has enabled the agent to make profitable trading decisions. The evaluation results highlight that the trained agent can perform well in a simulated trading environment, achieving substantial growth in account value while adapting to market dynamics.

Conclusion

The Q-learning algorithm demonstrates its effectiveness in solving the stock trading problem by enabling an agent to learn optimal trading strategies through trial and error. The results show promising potential for applying reinforcement learning techniques in financial decision-making tasks.