SOC’25 Progress Report (Weeks 1-4)

In Week 1, I focused on brushing up my fundamentals, although the content was relatively basic for me since I already had prior knowledge of Python, NumPy, Matplotlib, and basic probability. Therefore, I applied most of my attention toward understanding the fundamentals of machine learning. I explored the key concepts like supervised vs unsupervised learning, overfitting, underfitting, and the importance of train/test splits. I also completed the Coursera ML course and watched all its videos to get a clearer idea of the overall landscape of machine learning.

In Week 2, my learning pace picked up as I explored practical tools like Pandas and linear programming. I brushed up ways to handle and manipulate datasets using Pandas and practiced operations like filtering, grouping, and handling missing data. I also dove into linear programming, learning the mathematical formulation. Alongside this, I formally started my reinforcement learning (RL) journey. I learned the core concepts of RL, including the agent-environment loop, rewards, actions, and states. The difference between RL and traditional supervised learning became clearer through video resources and introductory readings. This week gave me a solid introduction to RL and also included my first assignment, where I implemented linear regression from scratch using NumPy and evaluated it on a housing price dataset using RMSE. I visualized predictions also.

Week 3 was focused entirely on Multi-Armed Bandits, which I now understand to be a foundational concept in reinforcement learning. I studied the exploration vs exploitation dilemma in detail and implemented three key algorithms: Epsilon-Greedy, UCB1 (Upper Confidence Bound), and Thompson Sampling. I created a complete simulation environment using Python classes to represent bandit arms, and I coded the algorithms from scratch to measure their performance in terms of cumulative regret. I ran experiments for multiple arm configurations and compared the regrets of each algorithm. My results showed that Thompson Sampling consistently outperformed the others in terms of regret minimization, with a flatter curve that indicated faster convergence to the best arm. I also visualized the learning behavior of each algorithm over time, which helped me intuitively understand how they differ.

In Week 4, I explored Markov Decision Processes (MDPs), which generalize bandits to sequential decision-making over states. Through two structured videos given by mentors, I learned how MDPs define environments using a combination of states, actions, rewards, and transition probabilities. I understood how agents interact with MDPs to learn optimal policies using value-based approaches. Although this week was more theoretical till now than the previous ones, it laid a solid foundation for understanding dynamic programming methods like value iteration and policy iteration, which I plan to explore and code in assignment.