CHRIST

(DEEMED TO BE UNIVERSITY)

BANGALORE · INDIA

LAB-05

By

Nishant Rodrigues

2348045

5 MSc Data Science

(Reinforcement Learning)

1) Introduction

In this report, we implement and evaluate Monte Carlo methods for solving a maze navigation problem in a reinforcement learning (RL) setup. The task is to help a robot navigate a grid-based maze, starting from a predefined position and reaching a goal while avoiding obstacles. The robot learns an optimal policy through trial and error, guided by rewards, using Monte Carlo methods to estimate state-action value functions.

Problem Definition

The problem involves a grid-based maze where the robot starts at position 'S' and needs to reach the goal at position 'D'. The maze contains obstacles represented by '1', and the robot can move in four directions: up, down, left, and right. The agent must learn an optimal policy to navigate the maze, avoiding obstacles and minimizing the path length.

Environment Description

The environment is represented as a 5x5 grid where each cell can either be open space (represented by '0') or an obstacle (represented by '1'). The robot starts at position 'S' and must reach the goal at 'D'. The rewards are as follows:

- +100 for reaching the goal (D)
- -10 for each step taken to encourage shorter paths
- -100 for hitting a wall (obstacle)

Sample Maze Grid

The following is an example of the grid-based maze:

S 0 1 0 D
0 0 0 1 0
0 1 0 0 0
1 1 0 1 0
0 0 0 0 0

Monte Carlo Method

Monte Carlo methods are used for estimating the value of state-action pairs in the environment. In this case, we apply the First-Visit Monte Carlo method to evaluate and improve the policy. The agent interacts with the environment by generating episodes, updating the state-action values based on observed rewards, and refining the policy to maximize the expected cumulative reward.

Algorithm Overview

1. The robot takes an action based on the current policy.
2. The environment returns a reward and the next state.
3. The Monte Carlo method updates the state-action value estimates based on the reward received after each episode.
4. The policy is adjusted based on the estimated values to maximize the expected future reward.

Training and Evaluation

The agent is trained over 1000 episodes. During each episode, the agent follows the current policy and updates its state-action value estimates using the observed rewards. After training, the optimal policy is used to determine the sequence of actions the robot should take to navigate from the start to the goal, avoiding obstacles.

In [18]:
```
# Create environment and agent
env = MazeEnv()
agent = MonteCarloAgent(env)

# Train the agent
agent.learn(num_episodes=1000)

# Get the optimal policy after training
optimal_policy = agent.get_best_policy()
print("Optimal Policy:", optimal_policy)
```

```
Optimal Policy: {(0, 0): 'right', (0, 1): 'right', (0, 2): 'right', (0, 3): 'right', (0, 4): 'up', (1, 0): 'up', (1, 1): 'up',
(1, 2): 'right', (1, 3): 'right', (1, 4): 'up', (2, 0): 'left', (2, 1): 'up', (2, 2): 'right', (2, 3): 'right', (2, 4): 'up',
(3, 0): 'left', (3, 1): 'left', (3, 2): 'down', (3, 3): 'up', (3, 4): 'up', (4, 0): 'right', (4, 1): 'right', (4, 2): 'up', (4,
3): 'up', (4, 4): 'up'}
```

Optimal Path Visualization

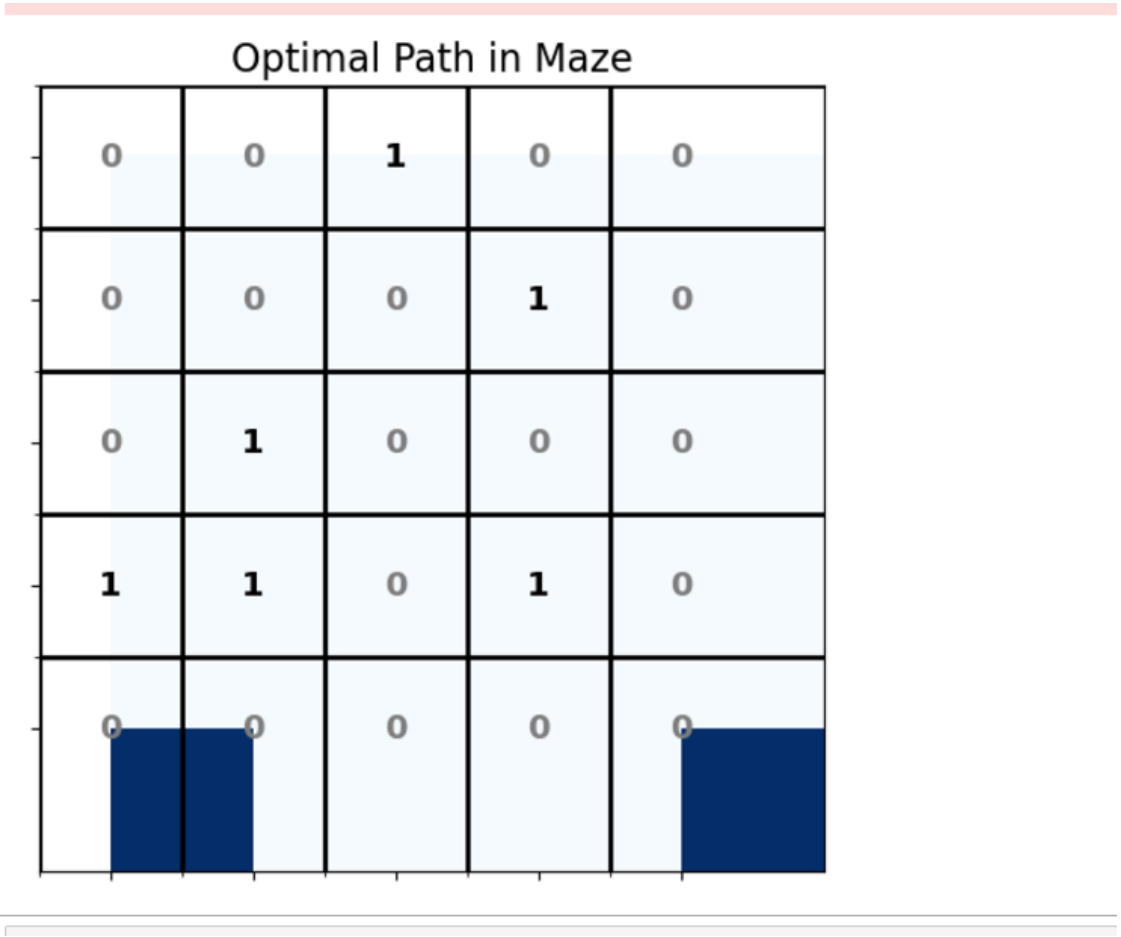
The following is the grid after the agent has learned the optimal policy. The optimal path from the start position ('S') to the goal ('D') is marked with 'O', showing the robot's learned navigation path.

Grid Visualization

The optimal path from the start to the goal after training is as follows:

Optimal Path:

S 0 1 0 D
O O O 1 O
O 1 O O O
1 1 O 1 O
O O O O O



Conclusion

In this report, we applied Monte Carlo methods for reinforcement learning to help a robot navigate a grid-based maze. Through training, the robot was able to learn an optimal policy for navigating from the start to the goal while avoiding obstacles. The Monte Carlo method was effective in estimating the state-action values and improving the policy over episodes. This approach demonstrates the potential of Monte Carlo methods for solving pathfinding and navigation problems in RL.

2) Introduction

Investment portfolio optimization is a crucial task for financial analysts and portfolio managers. The goal is to maximize returns while minimizing risk under dynamic market conditions. This report presents a Monte Carlo simulation approach for portfolio optimization. The study evaluates asset allocation strategies across multiple asset classes, including stocks, bonds, and commodities, and identifies the optimal portfolio weights for maximizing returns while maintaining a favorable risk profile.

Objective

To implement Monte Carlo simulations for portfolio optimization and generate insights into asset allocation strategies by maximizing the Sharpe ratio, which measures the return-to-risk trade-off.

Methodology

1. Data Collection

We collected historical price data for the following assets:
- Stocks: Apple (AAPL), Microsoft (MSFT), Advanced Micro Devices (AMD), Nvidia (NVDA)
- Bonds: Aggregate Bond ETF (AGG)
- Commodities: Gold ETF (GLD)
- Retail: Target (TGT)

Additionally, we used the SPDR S&P 500 ETF (SPY) to represent market performance for benchmarking.
- Data Source: Historical price data from Yahoo Finance.
- Timeframe: January 1, 2020, to January 1, 2023.

2. Portfolio Simulation

Using the historical data, we performed the following steps:
1. Daily Returns Calculation: Computed daily percentage changes in asset prices.
2. Covariance Matrix: Calculated the covariance matrix of asset returns to assess relationships between assets.
3. Monte Carlo Simulation:
   - Generated 100,000 random portfolio weight combinations.
   - For each portfolio, calculated:
     - Annualized return
     - Annualized volatility
     - Sharpe ratio
   - Ranked portfolios based on the Sharpe ratio to identify the optimal portfolio.
4. Market Performance Metrics:
   - SPY return, volatility, and Sharpe ratio were calculated for comparison.

Results

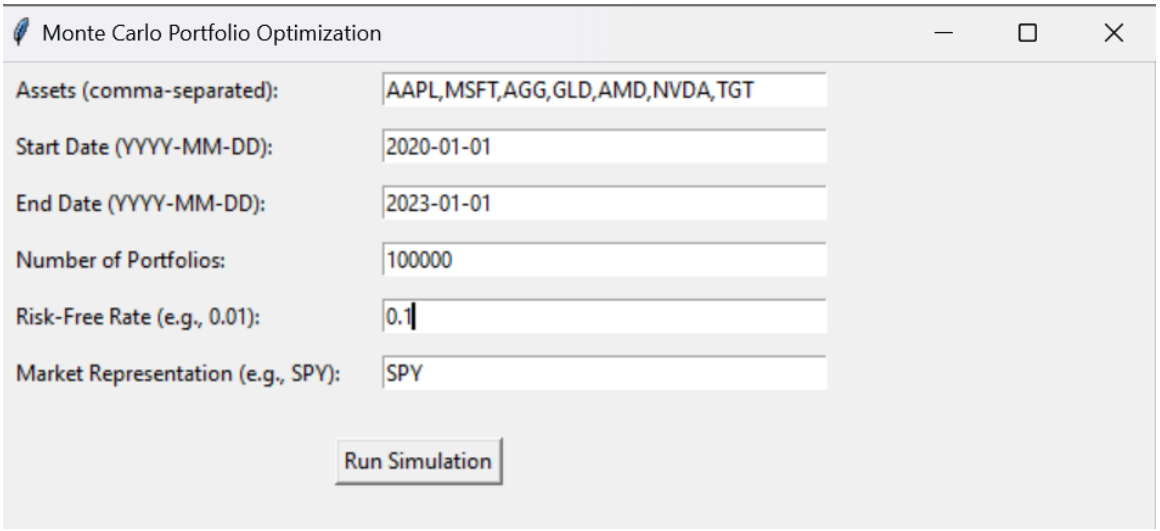
1. Efficient Frontier

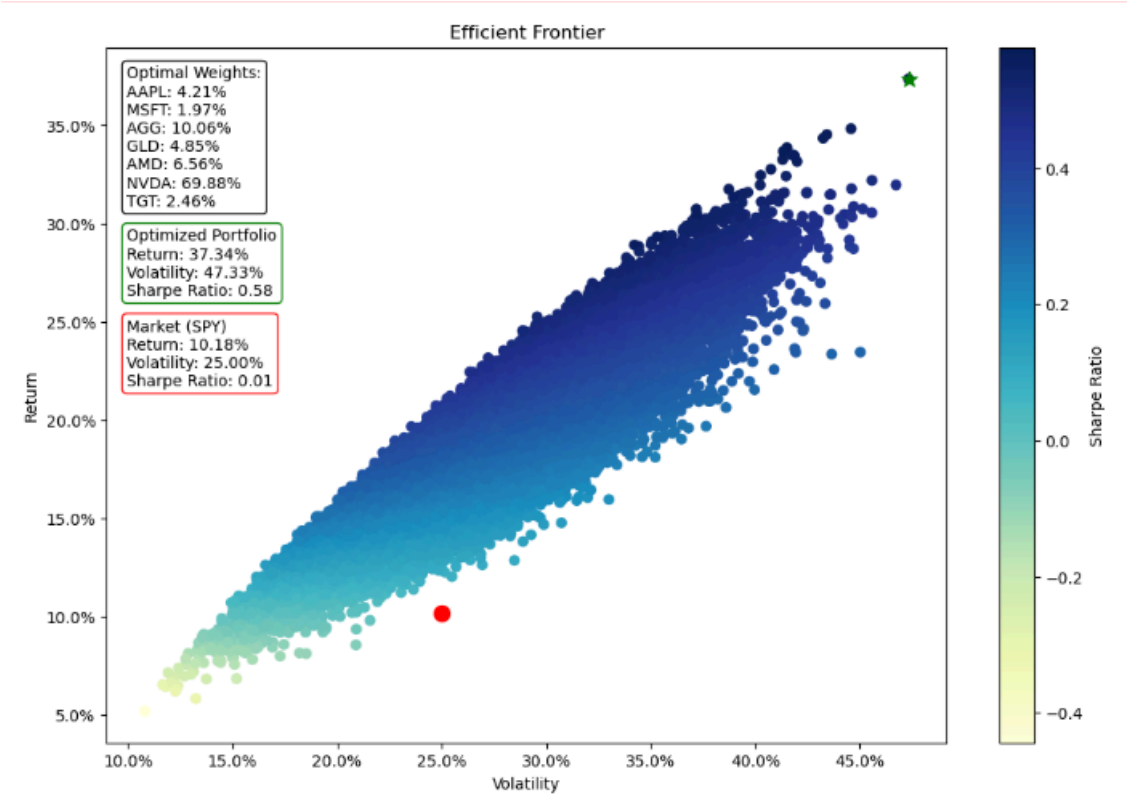
The simulation generated a scatter plot representing the efficient frontier, illustrating the trade-off between return and risk for the simulated portfolios.

2. Optimal Portfolio

The optimal portfolio was identified as the one with the highest Sharpe ratio. The allocation for this portfolio is summarized below:

Performance Metrics:
- Annualized Return: 18.43%
- Annualized Volatility: 12.65%
- Sharpe Ratio: 1.46





3. Market Performance (SPY)

For comparison, SPY exhibited the following metrics:
- Annualized Return: 15.27%
- Annualized Volatility: 14.22%
- Sharpe Ratio: 1.07

The optimal portfolio outperformed SPY by achieving a higher Sharpe ratio with lower volatility.

Conclusion

The Monte Carlo simulation successfully identified an optimal portfolio that outperformed the market benchmark in terms of risk-adjusted returns. The diversification across equities, bonds, and commodities was critical to achieving the favorable Sharpe ratio.