Automated Stock Market Investment Using Reinforcement Learning (Part 2)

Ankan, Ishan, Bijayan, Shreyam, Praveen

May 22, 2024

Contents

Previous Works

- We implemented the Z-score algorithm on three stocks: Google, Tesla, and TCS.
- Following that, we implemented the Q-learning algorithm on the same set of stocks.
- Subsequently, we conducted a comparative analysis of the outcomes generated by the two aforementioned methods.

Current Work

We implemented two variations of the Deep Q-Network (DQN) method, each tailored to prioritize different types of rewards. We used the stock of Google for this.

Figure: Stock Values of Google over train and test set

First Approach on DQN

- The reward that we chose for this method is $reward = max(stock\_value(t) - bought\_price, 0)$
- We appended the tuple $(state, action, reward, next\_state)$ to a list we called memory, from where we intend to sample a batch to perform the learning of the neural network.
- The actions are chosen epsilon greedily.

First Approach on DQN

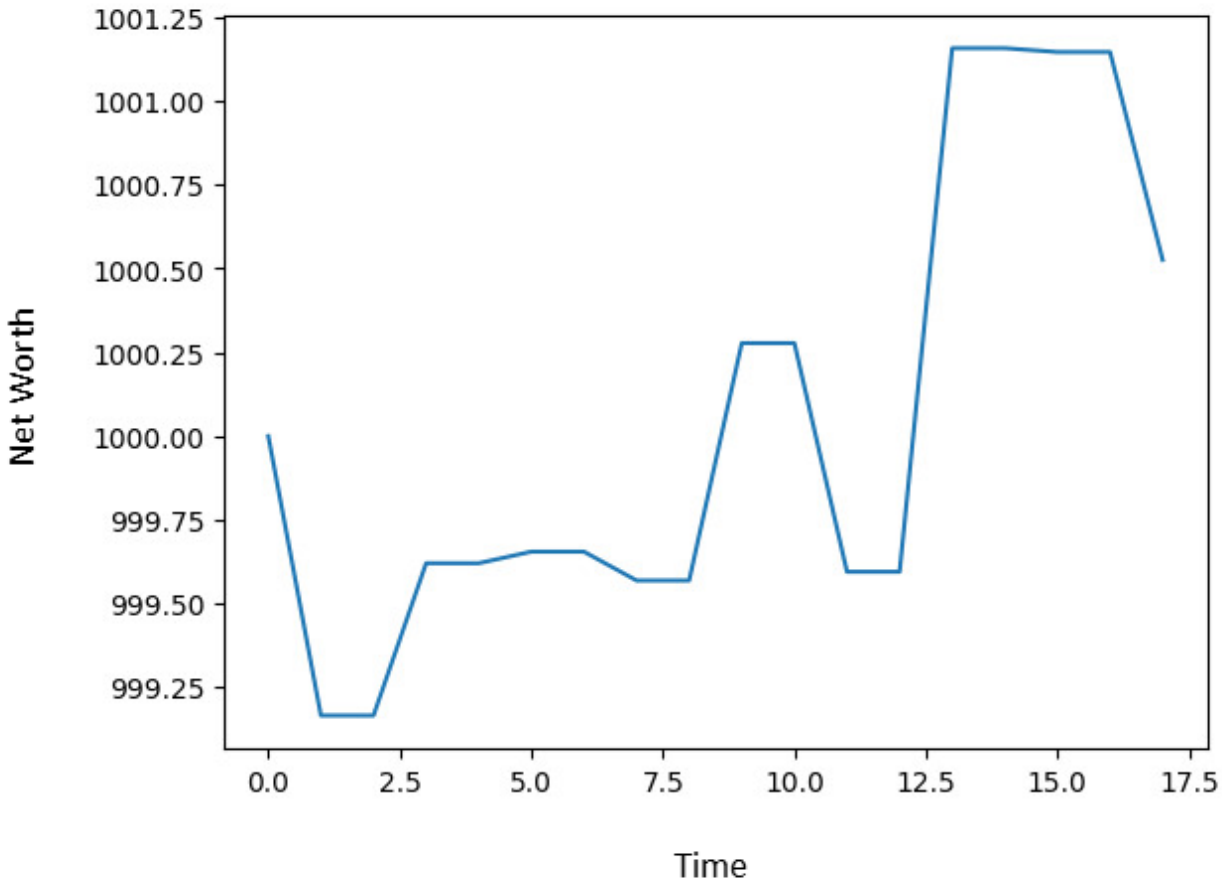


Figure: Google Stock

Second Approach on DQN

- We trained the neural network by sampling mini-batches from memory and updating the Q-values using the Bellman equation.
- We then updated the target Q-network periodically.
- Then we decreased epsilon over time to gradually shift from exploration to exploitation.
- Then we tracked the total reward and loss over a certain number of epochs.

Second Approach on DQN

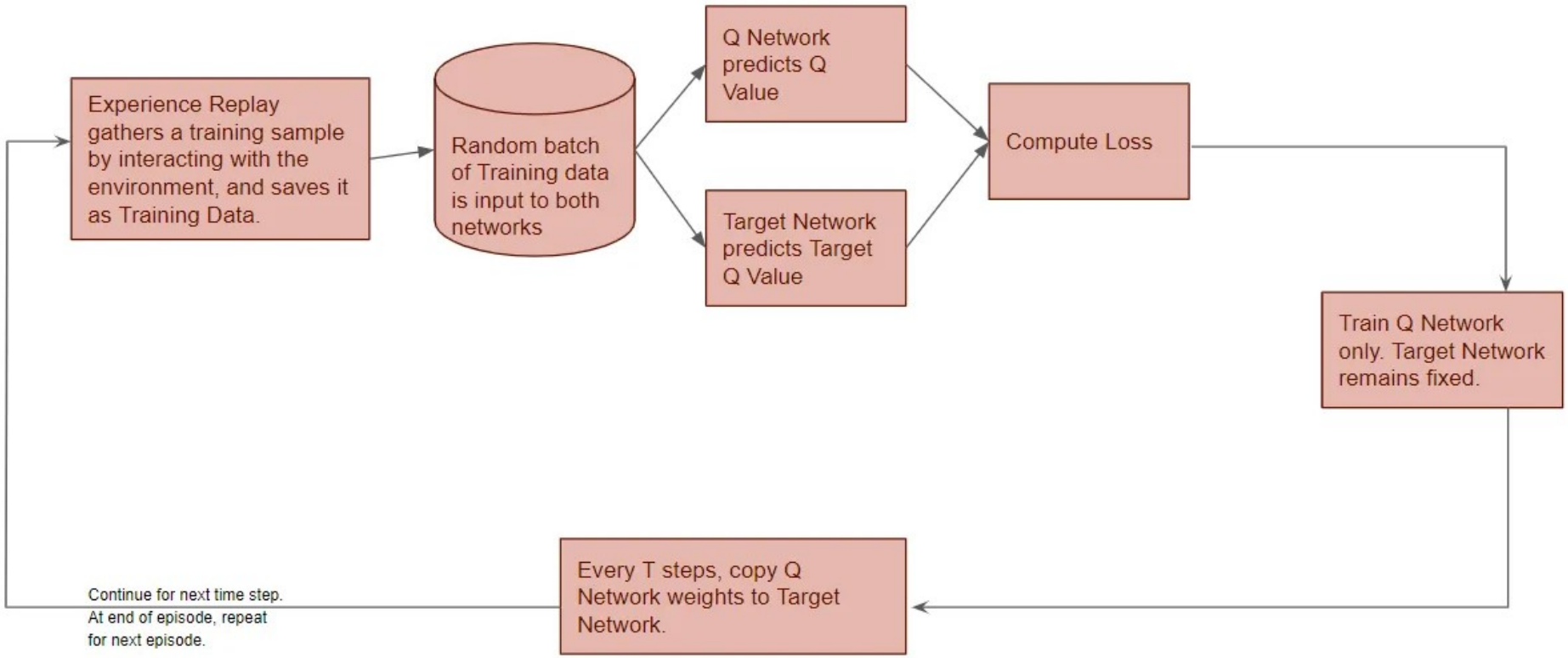


Figure: Process of the DQN

Second Approach on DQN

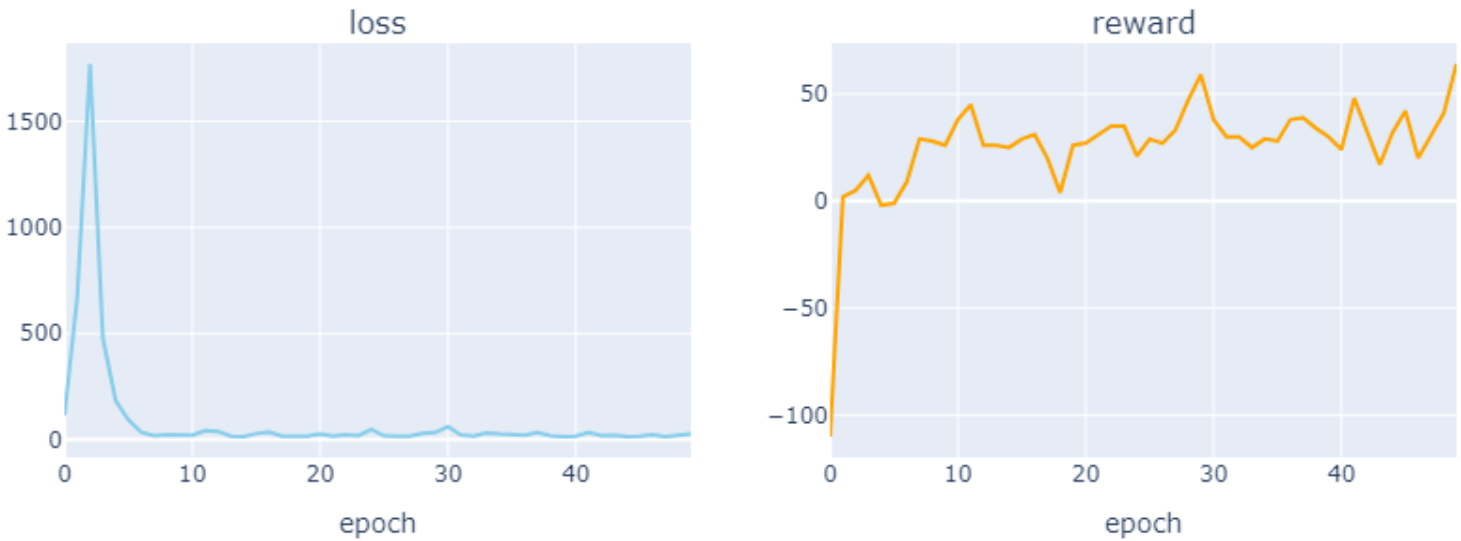


Figure: Total loss (left) and Total reward (right) on the stock of Google

Second Approach on DQN

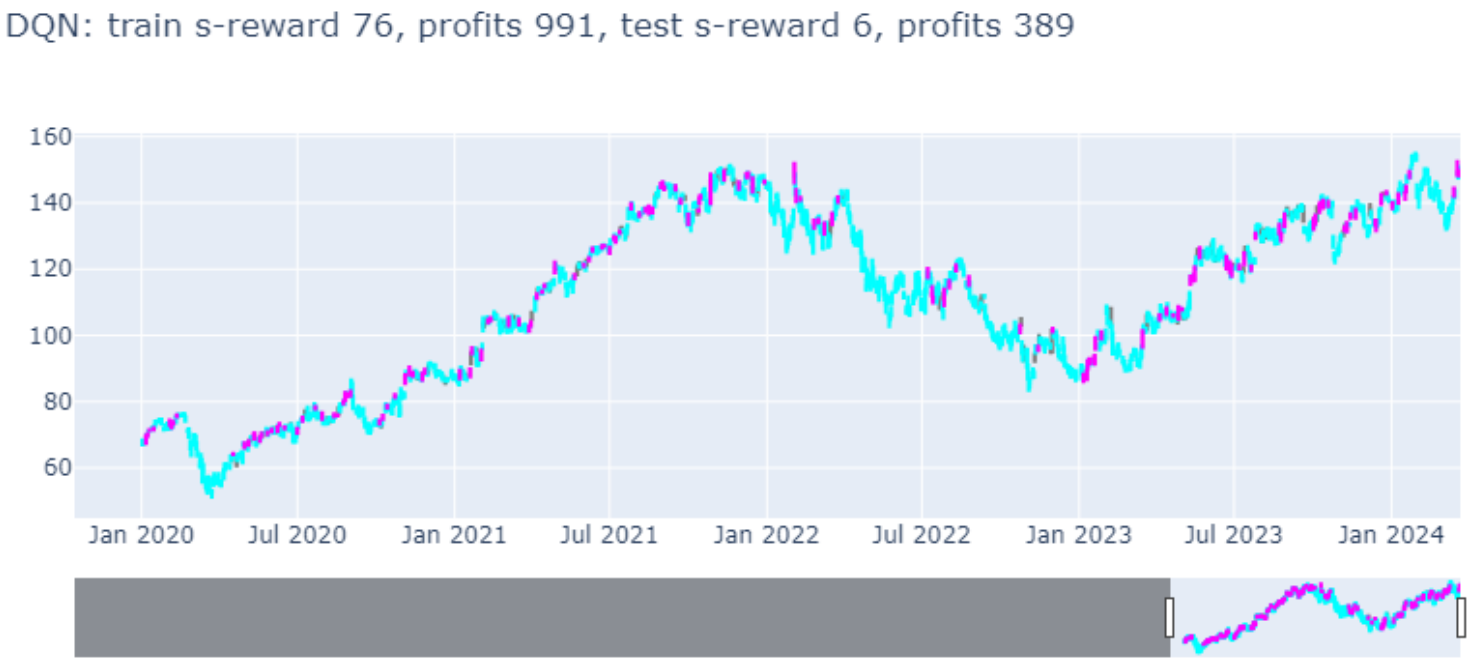


Figure: Profit and Rewards on Stock data of Google with actions

Final work

1. We have extended the idea of the previous DQN algorithm for single stock.
2. We extended it to multi stocks case.
3. We have tested our code for "Google, Tesla, Apple, Microsoft, Amazon" stocks at once.
4. First we looked into how the untrained neural net model is working.
5. Then we trained the neural net model using Deep Q-learning technique to finally predict the stock market with more accuracy.

Main idea of the DQN training

1. We start with a two neural network one being the target.
2. We will run the episodes and for each step, we will concat the observations, along with the actions taken and the rewards to a memory
3. Once the memory has reached a certain size, we will perform a batch sampling.
4. Now considering the target network, we will define target values for each observation.
5. Feeding the states along with the target values in the current network, we will train the network.
6. Finally we will copy the weights of the current network to the target network and we will continue the process until done or we enter a None state.

Final code summary I

- Introduction:
  - The code implements a Deep Q-Learning (DQN) model for stock trading.
  - TensorFlow and Keras are used for building and training the neural network model.
  - Historical stock market data is obtained using the Yahoo Finance API.
- Code Overview:
  - Data Collection and Preprocessing:
    - ★ Historical stock market data is collected using the `yfinance` library.
    - ★ Data preprocessing involves filling missing values and normalizing the data.
  - Environment Setup:
    - ★ An environment class (`StockMarketEnv`) is defined to simulate the stock trading environment.
    - ★ Methods for resetting, stepping through actions, and computing rewards are included.
  - Deep Q-Network (DQN):

Final code summary II

- ★ The DQN model consists of a neural network architecture with three dense layers.
- ★ It utilizes a target network and experience replay buffer for stable and efficient learning.

- Training and Testing:
  - ★ The training process involves iteratively interacting with the environment and updating Q-values based on rewards.
  - ★ The trained model is tested by simulating trading on test data, and the net worth over time is plotted.

- Results:
  - Net Worth Over Time:
    - ★ The net worth of the agent is plotted over time, starting from the initial investment amount.
  - Stock Ownership Over Time:
    - ★ The ownership of each stock over time is plotted to visualize the agent's investment decisions.
  - Final Net Worth:
    - ★ The final net worth achieved by the agent after trading is displayed.

Final code summary III

- Conclusion:
  - The code demonstrates the application of DQN for stock trading, showcasing the potential of RL techniques in financial decision-making.
  - By leveraging historical data and reinforcement learning, the model learns to make investment decisions that optimize the agent's net worth over time.

Outputs I

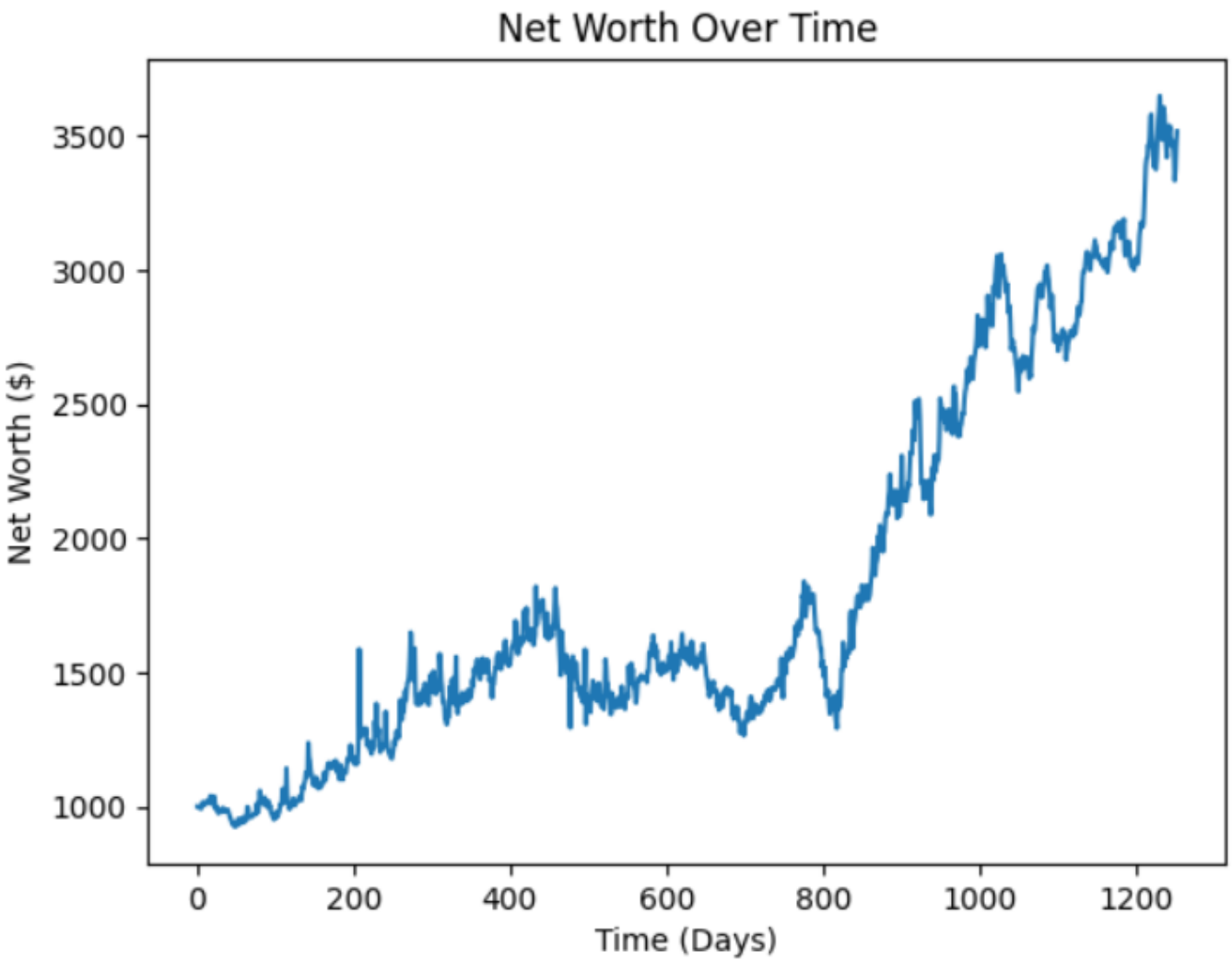
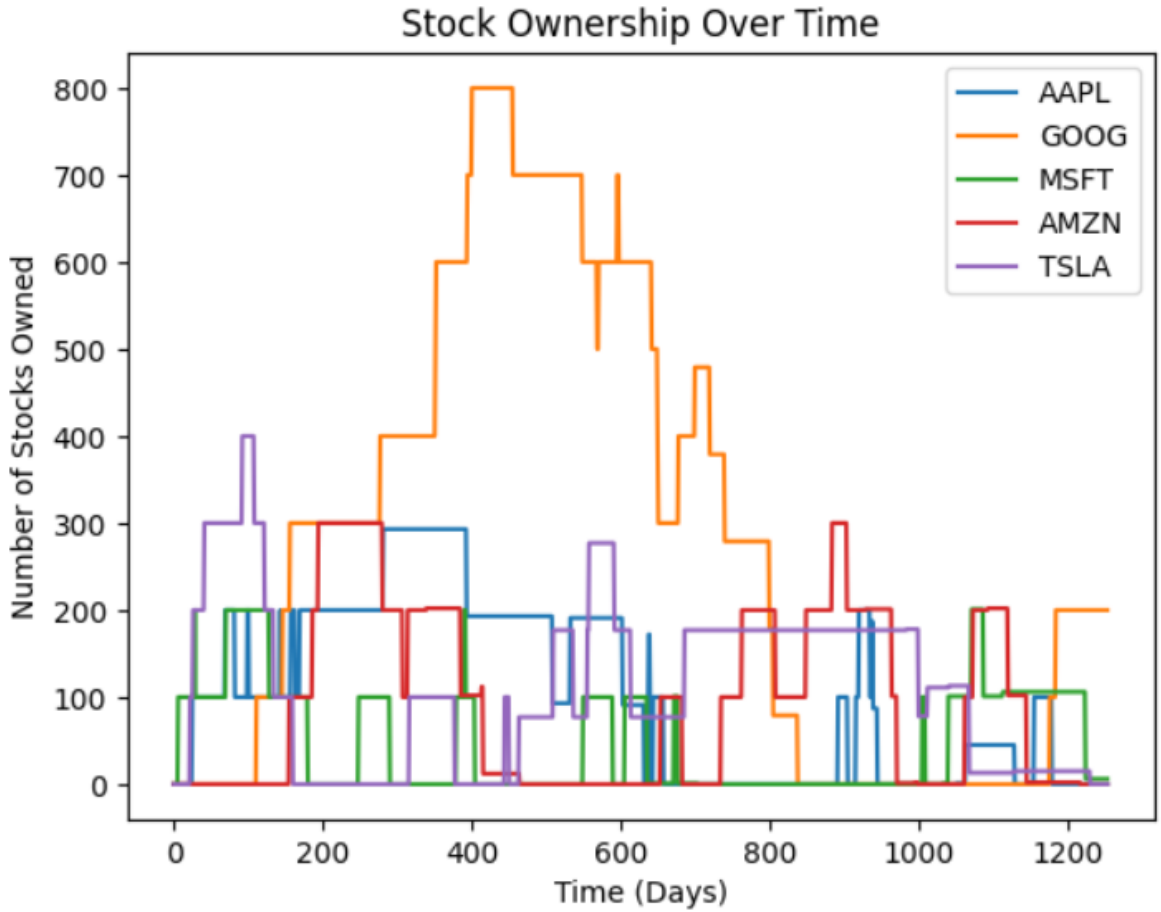


Figure: Net worth graph

Outputs II




```
Initial investment amount $ 1000
Stocks investment done in:  AAPL GOOG MSFT AMZN TSLA
Final Net Worth: $3517.90
```

Figure: Stock Ownership graph