Assignment 3 - CSE 574 - Introduction to Machine Learning

Adeline Grace George

December 10, 2021

1 Assignment Overview

The goal of the assignment is to learn the trends in stock price, to perform a series of trades over a period of time and to end with a profit. Each trade could be either a "buy"/"sell"/"hold". The initial investment capital is $100,000 and the performance of the model is measured as a percentage of the return on investment. The model is implemented using Q-Learning algorithm for Reinforcement Learning.

2 Dataset

The given dataset is a historical stock price for NVIDIA over the last 5 year period. It consists of 1258 entries from 10/27/2016 to 10/26/2021. The features include information such as the price at which the stock opened, the intraday high and low, the price at which the stock closed, the adjusted closing price and the volume of shares traded for the day.

3 Python

Jupyter Notebook has been used for implementation.

3.1 Reinforcement Learning Model using Q-Learning Algorithm

Reinforcement learning (RL) is an area of machine learning that focuses on how an agent might act in an environment in order to maximize some given reward. Reinforcement learning algorithms study the behavior of subjects in such environments and learn to optimize that behavior.

3.1.1 Markov Decision Process

An MDP consists of the following components:

- **agent:** the key decision maker
- **environment:** the surroundings where the agent navigates after interacting with it
- **state:** the current situation in the environment that the agent is positioned in
- **action:** choice of moves that the agent can take given it's current state
- **reward:** a feedback given to the agent (positive, negative or neutral) depending on the action it has taken

In reinforcement learning, the agent interacts with the environment it is placed in to make decisions about the next course of action to take from it's current state, receiving valuable feedback aka, rewards which in turn helps it navigate the environment better.

3.1.2 Q-Learning

The objective of Q-learning is to find a policy that is optimal in the sense that the expected value of the total reward over all successive steps is the maximum achievable. So, in other words, the goal of Q-learning is to find the optimal policy by learning the optimal Q-values for each state-action pair.
$q*(s,a) = E\left[R_{t+1} + \gamma maxq*(s',a')\right]$ The Q-table is updated iteratively as the agent takes actions. The agent has two choices:

- **exploration:** the act of interacting and studying the environment to find out information about the environment
- **exploitation:** the act of making use of information that is already known (Q-table values) about the environment to maximize return

The key to designing a good Q-learning model is striking a balance between exploration and exploitation - it's a trade-off. To get this balance between exploitation and exploration, we use what is called an **epsilon greedy strategy**. With this strategy, we define an exploration rate $\varepsilon = 1$ that we initially set to 1. This exploration rate is the probability that our agent will explore the environment rather than exploit it. With $\varepsilon = 1$, it is 100 percent certain that the agent will start out by exploring the environment. As the agent learns more about the environment, at the start of each new episode, will decay by some rate that we set so that the likelihood of exploration becomes less and less probable as the agent learns more and more about the environment. The agent will become "greedy" in terms of exploiting the environment once it has had the opportunity to explore and learn more about it.

3.1.3 Environment

The environment provided is a **Stock Trading Environment**. The Q-Learning Algorithm is used to train the agent to interact with the given environment to enable it to make wise choices by choosing between the three possible actions **- buy, sell and hold** such that the end goal is achieved. The end goal is to **maximize the return on investment** of the initial capital ($100000) in this case. The reward that the agent receives after taking an action depends on the trend of the stock price after the choice was taken (positive reward for an increase in price and a negative reward otherwise)

3.1.4 Visualization

In Jupyter Notebook

3.1.5 Results

Average reward per 100 episodes: 100 : 4.86418075425284 200 : 22.498126672167437
300 : 57.65796007970613 400 : 82.15277411278612 500 : 105.28622957113092
600 : 127.13443832241181 700 : 108.12634787569155 800 : 127.65090373827552
900 : 132.56639526626262 1000 : 139.32893040693943
Final Stock Value: 149981.190679