RL in Automated Trading

Aadi Krishna Vikram (211020402)
Ashish Agrawal (211020414)
Ankit Ranjan (211020413)
Sahil Pradhan (211020441)

Date: 15/03/2023

Dr. Shyama Prasad Mukherjee International Institute of Information Technology, Naya Raipur

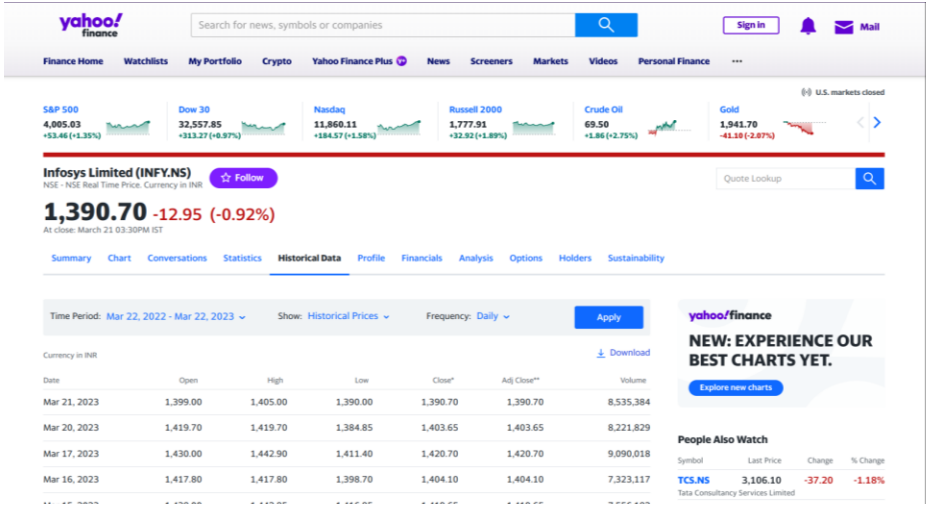
Content

- Introduction to Problem Statement
- Reinforcement Learning
- Proposed Methodology
- Reward System
- Q-Learning
- Demonstration
- Conclusion
- Future Scope

Introduction to Problem Statement

- **Exploring the use of reinforcement learning in automated trading**

- **Whether it is any beneficial or not ?**



Environment Dynamics

- **States**
  - **Opening position**
  - **Closing position**

- **Action**
  - **Sell**
  - **Buy**
  - **Hold**

Reward System

- **+ ve Reward** : buying a stock at a lower price and selling it at a higher price
- **- ve Reward :** selling a stock at a lower price than the purchase price

- **Transaction Costs** : - 0.1% net from all trades

*Transaction costs: Trading involves transaction costs such as commissions, fees, and slippage. The reward function should take these costs into account to avoid excessive trading.

*Also includes Stop Loss Feature

Reinforcement Learning

- **‘Science of decision making’**
- Agent interacts with environment and receives feedback as rewards or penalties based on the actions it takes.
- **Goal ->** Learn a policy that maximizes the expected cumulative reward over time.

Methodology

- We have trained our agent using Q – Learning algorithm .
- Some of the methods defined are :-

Getstate

Buy( )

Act( )

Sell( )

Train( )

Methodology Contd..

We have trained our agent using Q – Learning algorithm .

- The Q-learning algorithm updates the Q-table based on the observed rewards.
- The algorithm selects an action based on the current state and the values in the Q-table.
- The reward for the selected action is then observed, and the Q-table is updated based on the observed reward.

Q-Learning

- Model Free -> Dynamics of the environment are not known

- Off-Policy RL Algorithm

- Enables an agent to learn optimal actions in a Markov decision process (MDP) by estimating the expected long-term reward for each action taken in a given state.

Contd.

It :

- Trains Q-function, an action-value function that contains, as internal memory, a Q-table that contains all the state-action pair values.
- Given a state and action, our Q-function will search into its Q-table the corresponding value.

$$\underbrace{Q(S_t, A_t)}_{\text{New Q-value estimation}} \leftarrow \underbrace{Q(S_t, A_t)}_{\text{Former Q-value estimation}} + \underbrace{\alpha}_{\text{Learning Rate}}[\underbrace{\underbrace{\underbrace{R_{t+1}}_{\text{Immediate Reward}} + \underbrace{\gamma max_a Q(S_{t+1}, a)}_{\text{Discounted Estimate optimal Q-value of next state}}}_{\text{TD Target}} - \underbrace{Q(S_t, A_t)}_{\text{Former Q-value estimation}}}_{\text{TD Error}}]$$

Contd.

- When the training is done,we have an optimal Q-function, so an optimal Q-table.
- And if we have an optimal Q-function, we have an optimal policy,since we know for each state, what is the best action to take.

$$\pi^*(s) = \arg\max_a Q^*(s,a)$$

Note - In the beginning, the Q table is initialised as 0 and as it explore the environment and update our Q-table it will give us better and better approximations.

Q-Learning Algorithm

Input: policy π, positive integer $num\_episodes$, small positive fraction α, GLIE $\{\epsilon_i\}$
Output: value function Q ($\approx q_\pi$ if $num\_episodes$ is large enough)
Initialize Q arbitrarily (e.g., $Q(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$, and $Q(terminal\text{-}state, \cdot) = 0$) Step 1
for $i \leftarrow 1$ *to* $num\_episodes$ **do**
    $\epsilon \leftarrow \epsilon_i$
    Observe S_0
    $t \leftarrow 0$
    repeat
        Choose action A_t using policy derived from Q (e.g., ϵ-greedy) Step 2
        Take action A_t and observe R_{t+1}, S_{t+1} Step 3
        $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$ Step 4
        $t \leftarrow t + 1$
    until S_t *is terminal*;
end
return Q

Conclusion

- We Implemented Q-Learning for the Model
- From our initial testing, the Model had a decent run
- RL can be used in automated day trading

Future Scope

- **We can include more Parameters in our decision making like the current affairs of the company, the social media engagement etc.**

- **Can add more attributes in the dataset**

- **Try Multiple Reinforcement Learning algorithms**

Thank You

Dr. Shyama Prasad Mukherjee International Institute of Information Technology, Naya Raipur