

Stock Transaction Game

Group Members

Rich Jiang rijiang

Pavan Kumar Vaddineni, pvaddine

Chaitanya Mandava, cmandava

Wanning Ding, wding04

Link to the github repository

<https://github.com/rjiang1/StockTrade.git>

Game Rules

States

The state of the game at each turn is players current portfolio that consists of company name, number of shares of that company owned by player, player's current cash balance and the current day's information of the stock AAPL.

In addition, stock information includes

- High - The highest price the stock traded for that day
- Low - The lowest price the stock traded for that day
- Close - The final price of the stock that day
- Volume - The amount of stocks everyone trades in the stock market on that day
- 20_day - The average price of the stock over the course of the last 20 days
- 50_day - The average price of the stock over the course of the last 50 days

State Representation

The player states are represented as class Player which has attributes portfolio and cash.

The portfolio is stored by dictionary i.e. {companyName: # of shares} and current cash balance is stored in a variable.

The stock states are represented as class Stock which has attributes ticker and price.

Valid Actions

- Buy - Enter B/b to buy a stock and then the player has to enter number of stocks he/she want to buy, one cannot buy stock worth more than his/her cash balance.
- Sell - Enter S/s to sell a stock and then the player have to enter number of stocks he/she want to sell, one cannot sell stocks whose count is more than what he/she actually owns.
- Do Nothing/ Hold - Enter H/h, the player will be shown next day apple stock.
- Exit - Enter E/e the player will be exited out of prediction game and his/her portfolio, profit/ loss will be shown.

Game Over

The game is over after the n days player's prediction (n is what the player gave as input at first) or when the player enters the action Exit.

Winner/score Determined

TotalCash will be calculated as cash Balance + total shares cost that one holds according to $n+1$ day apple stock open price. And the player ends up in profit [win] i.e., $\text{totalCash} > 10000$ (starting balance) or loss i.e., $\text{totalCash} < 10000$.

Problem extension

We varied the size of the problem instance in our experiments by letting the player set the number of days he/she wants to predict the stock transaction. For the actions, we have added an exit action in the first version of the experiment, which allows players to exit the game at any time.

As of now, we implemented this game as a single player game because AI part isn't implemented yet. However for the different numbers of players, we prepared to extend the problem to two, three players or more players. The several players all predict according to the same stocks' information and the player who ends up in the highest profit win the game. For the different numbers of stocks, we can add more stocks' information except AAPL, like GOOG, MSFT, etc in the future. Then when the players do the buy, sell or hold action, players need to not only set the number of shares but also choose which stock to trade.

Language and Libraries

Python, numpy, pandas, pytorch/tensorflow/keras

Example

How many days of stock you want to predict?5
Player your Shares : {'Apple': 0} your Cash: 10000

Day 1 apple stock: Open-->127.14250200000001

High-->127.485001

Low-->123.832497

Close-->125.010002

Volume-->155552400.0

4 things you can do
1. Buy(B/b) 2. Sell(S/s) 3. Hold(H/h) 4. Exit(E/e)

what you want to do?!!!!

s

How many shares you want to sell--> 5
Sorry you dont own that many stocks, Try again!!!

Player your Shares : {'Apple': 0} your Cash: 10000

Day 1 apple stock: Open-->127.14250200000001

High-->127.485001

Low-->123.832497

Close-->125.010002

Volume-->155552400.0

4 things you can do
1. Buy(B/b) 2. Sell(S/s) 3. Hold(H/h) 4. Exit(E/e)

what you want to do?!!!!

b

How many shares you want to buy--> 100
Sorry you cant afford that many stocks, Try again!!!

Player your Shares : {'Apple': 0} your Cash: 10000

Day 1 apple stock: Open-->127.14250200000001

High-->127.485001

Low-->123.832497

Close-->125.010002

Volume-->155552400.0

4 things you can do
1. Buy(B/b) 2. Sell(S/s) 3. Hold(H/h) 4. Exit(E/e)

what you want to do?!!!!

h

Player your Shares : {'Apple': 0} your Cash: 10000

Day 2 apple stock: Open-->126.012497

High-->126.442497

Low-->124.577499

Close-->124.807503

Volume-->187630000.0

4 things you can do
1. Buy(B/b) 2. Sell(S/s) 3. Hold(H/h) 4. Exit(E/e)

what you want to do?!!!!

h

Player your Shares : {'Apple': 0} your Cash: 10000

Day 3 apple stock: Open-->127.58000200000001

High-->131.0

Low-->126.0

Close-->129.03999299999998

Volume-->225702700.0

4 things you can do
1. Buy(B/b) 2. Sell(S/s) 3. Hold(H/h) 4. Exit(E/e)

what you want to do?!!!!

h

Player your Shares : {'Apple': 0} your Cash: 10000

Day 4 apple stock: Open-->132.759995

High-->134.800003

Low-->130.529999

Close-->134.179993

Volume-->152470100.0

4 things you can do
1. Buy(B/b) 2. Sell(S/s) 3. Hold(H/h) 4. Exit(E/e)

what you want to do?!!!!

b

How many shares you want to buy--> 2

Player your Shares : {'Apple': 2} your Cash: 9734.48001

Day 5 apple stock: Open-->137.58999599999999

High-->137.979996

Low-->127.0

Close-->131.399994

Volume-->200119000.0

The player lost the game and lost: 1.1300050000008923.
Game was over.