Project Phase I

by

Saurabh Katkar

Collaborators: None

Mode of Execution:

- Place the agent_skatkar.py file in the same folder as the agents.py and the simulate_agents_phase1.py file
- Add 'from agent_skatkar import Agent_skatkar' at the beginning of simulate_agents_phase1.py file.
- Add 'agents.append(Agent_skatkar("agent_skatkar"))' to add the user-created agent in the simulate_agents_phase1.py file.
- This code was executed in IPython using the Anaconda environment.

Overview:

Given the price, value and the condition of the product, determine the profit achieved if and when the product is sold for the cost of value after buying it for the cost of price.

The 'will_buy' method:

Based on the parameters used and the algorithm implemented the 'will_buy' method will determine the likelihood of the product getting bought for the given value.
The 'simulate_agents' class then calculates the profit by comparing the price and the value attributes of the products.

The strategy implemented in the 'will_buy' method:

Initial strategy:

The initial algorithm implemented took into account all the three attributes, namely price, value and probability and assigned the condition for probability based on the condition for the price-to-value ratio.
This was achieved using multiple if statements and it was found to achieve a higher profit margin than the given three strategies.

Optimized(Final) strategy:

Instead of taking multiple if statements that looks at certain cut-off margins to assign probability, in this case each value for the price/value ration is taken into consideration and is assigned the probability greater than that ratio.
This is because a small price-to-value ratio depicts that the product has a higher resale value than its original price, thus one could afford to buy such a product with a probability that is lower that the product with a high price-to-value ratio.

Simulation Results:

Upon execution, the optimized strategy gives a better result in most situations than our initial strategy. Comparing the two results we get the following output:

****START OF OUTPUT****

```
run simulate_agents_phase1.py
----------------------------------------------
FAIR MARKET
----------------------------------------------
Agent_hp:                  $130,502.60
Agent_ratio_0.75:                  $118,270.09
Agent_ratio_0.50:                  $136,310.69
Agent_ratio_0.25:                  $99,585.95
Agent_buy_all:             $24,891.66
Agent_agent_skatkar_optimized:             $171,979.38
Agent_agent_skatkar_initial:             $160,237.48
```

```
----------------------------------------------
JUNK YARD
----------------------------------------------
Agent_hp:                   $27,473.65
Agent_ratio_0.75:                   $-28,842.74
Agent_ratio_0.50:                   $45,543.23
Agent_ratio_0.25:                   $67,055.71
Agent_buy_all:              $-155,068.14
Agent_agent_skatkar_optimized:                $85,311.13
Agent_agent_skatkar_initial:              $65,796.89

----------------------------------------------
FANCY MARKET
----------------------------------------------
Agent_hp:                   $207,261.45
Agent_ratio_0.75:                   $231,146.78
Agent_ratio_0.50:                   $209,695.38
Agent_ratio_0.25:                   $124,640.58
Agent_buy_all:              $183,507.53
Agent_agent_skatkar_optimized:                $254,593.96
Agent_agent_skatkar_initial:              $234,472.77
```

******END OF OUTPUT******

Conclusion:

Thus, due to a higher profit margin, the optimized strategy was used in the code instead of the initial strategy and this strategy used for the 'will_buy' method gives a higher profit margin than the given algorithms.