Market Making through Reinforcement Learning

Contributors: Sanjeev Kumar, Thejeswar Reddy Narravala, Han Chiang

Problem Statement: Implementation of Market Order through Q-learning and SARSA

Introduction:

A market maker (MM) is a firm or individual who actively quotes two-sided markets in a security, providing bids and offers (known as asks) along with the market size of each.

For instance, a market maker in XYZ stock may provide a quote of $10.00-$10.05, 100x500. This means that they bid (they will buy) 100 shares for $10.00 and also offer (they will sell) 500 shares at $10.05. Other market participants may then buy (lift the offer) from the MM at $10.05 or sell to them (hit the bid) at $10.00. Market makers provide liquidity and depth to markets and profit from the difference in the bid-ask spread.

There are two types of order,

1. Limit order
2. Market order

What is Limit order?

A limit order is an order to buy or sell a stock at a specific price or better. A buy limit order can only be executed at the limit price or lower, and a sell limit order can only be executed at the limit price or higher. A limit order is not guaranteed to execute. A limit order can only be filled if the stock's market price reaches the limit price. While limit orders do not guarantee execution, they help ensure that an investor does not pay more than a pre-determined price for a stock.

They are required to strictly keep their inventory between -1 and 1. The problem is when to optimally quote either a bid or ask, or simply wait, each time there is a limit order book update. For example, sometimes it may be more advantageous to quote a bid to close out a short position if it will almost surely give an instantaneous net reward, other times it may be better to wait and capture a larger spread.

Example:

An investor wants to purchase shares of ABC stock for no more than $10. The investor could submit a limit order for this amount and this order will only execute if the price of ABC stock is $10 or lower.

What is market order?

A market order is an order to buy or sell a security immediately. This type of order guarantees that the order will be executed, but does not guarantee the execution price. A market order generally will execute at or near the current bid (for a sell order) or ask (for a buy order) price. However, it is important for investors to remember that the last-traded price is not necessarily the price at which a market order will be executed.

Main Objective

Find the time independent optimal policy for Market Making.

Implementation Details for Limit Order

The agent uses the liquidity imbalance in the top of the order book as a proxy for price movement and, hence, fill probabilities.

At each non-uniform time update, t, the market feed provides best prices and depths $\{p^a{}_t, q^a{}_t, p^b{}_t, q^b{}_t\}$. The state space is the product of the inventory, Xt $\in\{-1,0,1\}$, and gridded liquidity ratio $R^t = \lfloor (q^a{}_t / (q^a{}_t + q^b{}_t))*N \rfloor \in [0,1]$, where N is the number of grid points and $q^a{}_t$ and $q^b{}_t$ are the depths of the best ask and bid. $R^t \to 0$ is the regime where the mid-price will go up and an ask is filled. Vice versa for $R^t \to 1$. The dimension of the state space is chosen to be 3·10=30. A bid is filled with probability $\epsilon_t := R^t$ and an ask is filled with probability $1-\epsilon_t$. The rewards are chosen to be the expected total P&L. If a bid is filled to close out a short holding, then the expected reward $r_t = -\epsilon_t(\Delta p_t + c)$, where Δp_t is the difference between the exit and entry price and c is the transaction cost. For example, if the agent entered a short position at time s<t with a filled ask at $p^a{}_s = 100$ and closed out the position with a filled bid at $p^b{}_t = 99$, then $\Delta p_t = 1$. The agent is penalized for quoting an ask or bid when the position is already short or long respectively.

Implementation Details for Market Order

A market order is an order to buy or sell a security immediately. Therefore, it is independent of fill probability or any ratios.

- When in Short position, if the agent wants to exit then one would choose market buy action and the exit price is ask price.

- When in Long position, if the agent wants to exit then one would choose market sell action and the exit price is bid price.
- When in Flat position, the agent cannot enter into the market with Market buy or Market sell.

The Reward Function Matrix:

		States		
Actions		Short	neutral	Long
Limit Orders	Buy	$-\varepsilon(\delta p+c)$	$-\varepsilon * c$	0
	Hold	0	0	0
	Sell	0	$-(1-\varepsilon) * c$	$(1-\varepsilon)*(\delta p-c)$
Market Orders	Buy	$-(\delta p+c)$	0	0
	Sell	0	0	$(\delta p-c)$

- δp = Exit price – Entry price
- c = transaction cost
- ε = bid_fill probability
- 1- ε = Ask fill probability

Under the ε-greedy policy,

Q-Learning computes the difference between Q(s,a) and the maximum action value.

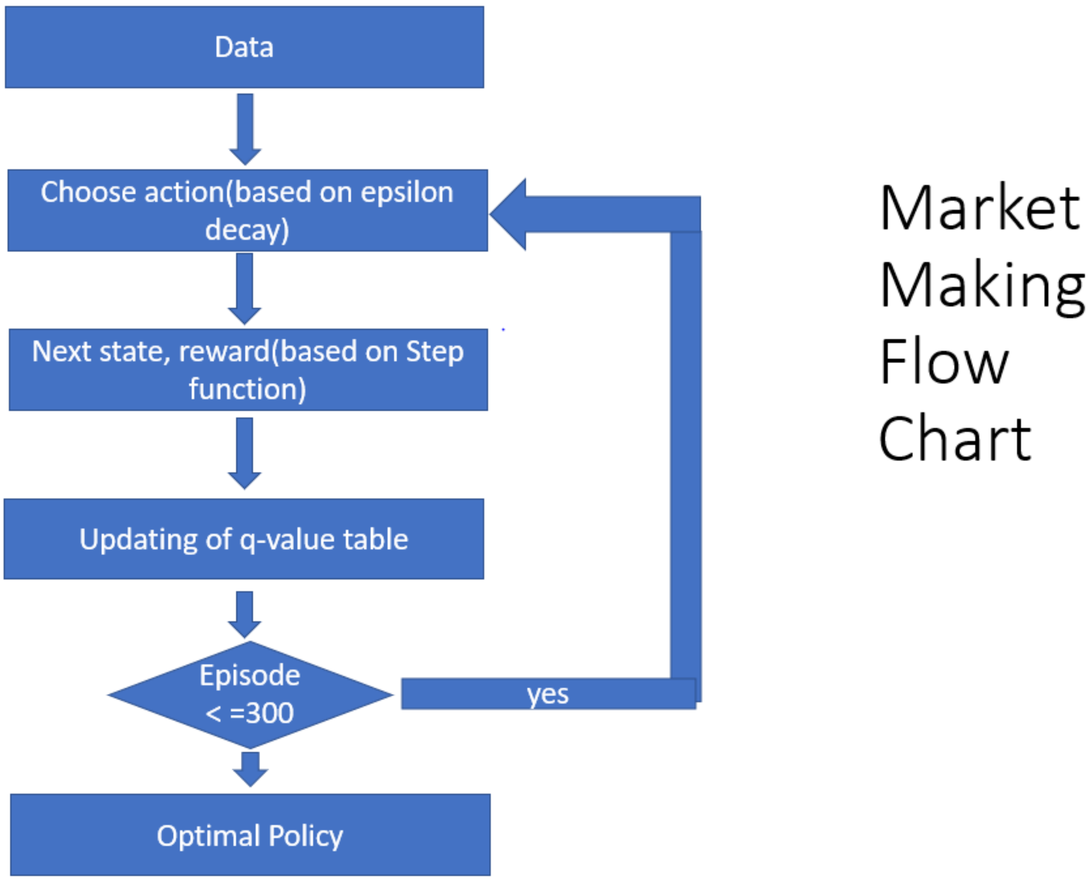
$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\Bigg(\underbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}_{\text{new value (temporal difference target)}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \Bigg)}^{\text{temporal difference}}$$

SARSA computes the difference between Q(s,a) and the weighted sum of the average action value and the maximum.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$$

We decrease the value of epsilon with each epoch - epsilon must approach zero as the number of episodes increases in order to ensure that the q-value function converges to the optimum

The Flow chart of the Python code reinforcement learning works



Code Execution Instruction:

1. Choose the ML_in_Finance_MarketMaking_MarketOrder_Project.ipynb
2. Provide the google drive Authentication
3. Set the dataset file location
4. Set the Alpha (step size) = 0.005 / 0.05
5. Set the transaction cost = 0 / 0.05
6. Run the Code

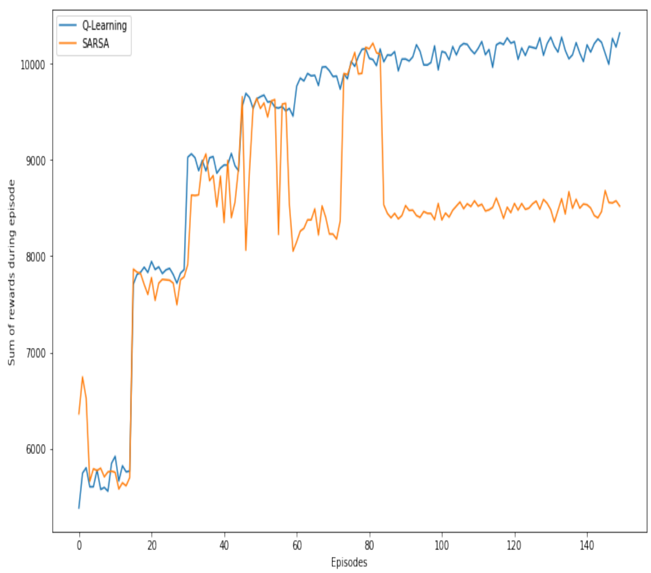
Results:

Amazon dataset Results:

With C= 0 & Alpha = 0.05

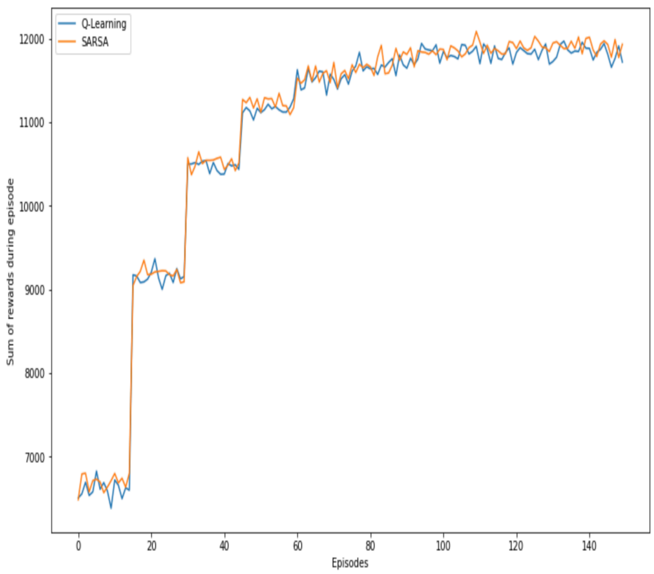
Optimal Policy

```
SARSA
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
          flat    b    b    b    b    b    b    b    b    b    h
         short    b    b    b    b    b   mb    b   mb   mb   mb
          long    h    s    s    s    s    h    s    s    s    s
Q-learning
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
          flat    b    b    b    b    b    b    b    b    b    h
         short   mb    b    b    b    b   mb    b    b   mb   mb
          long    h    s    s    s    s    s    s    s    s    s
```



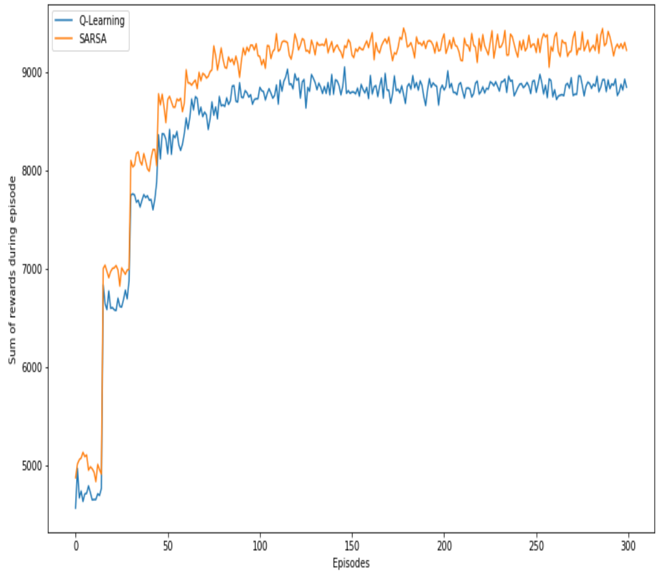
With C= 0 & Alpha = 0.005

```
SARSA
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
          flat    b    b    b    b    b    b    s    s    s    s
         short    b    b    b    b    b    b    b    b    b    b
          long    s    h    s    s    s    s    s    s    s    s
Q-learning
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
          flat    b    b    b    b    b    s    s    s    s    s
         short    b    b    b    b    b    b    b    b    b    b
          long    s    h    s    s    s    s    s    s    s    s
```



With C= 0.05 & Alpha = 0.005

```
SARSA
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
          flat    b    b    b    b    b    b    b    b    b    h
         short    b    b    b    b    b    b    b    b    b    mb
          long    s    s    h    s    s    s    s    s    s    s
Q-learning
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
          flat    b    b    b    b    b    b    b    b    b    b
         short    b    b    b    b    b    b    b    b    b    mb
          long    h    h    s    s    h    s    s    s    s    s
```



Amazon dataset Conclusion:

Important findings
When C= 0 and Alpha = 0.05

In the Optimal Policy If Ask_fill probability is greater than 0.5 and agent is in short position, the algorithm chooses market buy as the best action. As it is appropriate to choose the market buy option when there is less chance of bid_fill in the short position.
Same trend is appeared in the Q-Learning as well. Here, both the SARSA and Q-Learning are converging but not to the same value.
Therefore, Alpha is changed to 0.005 to observe if there is any impact on convergence,

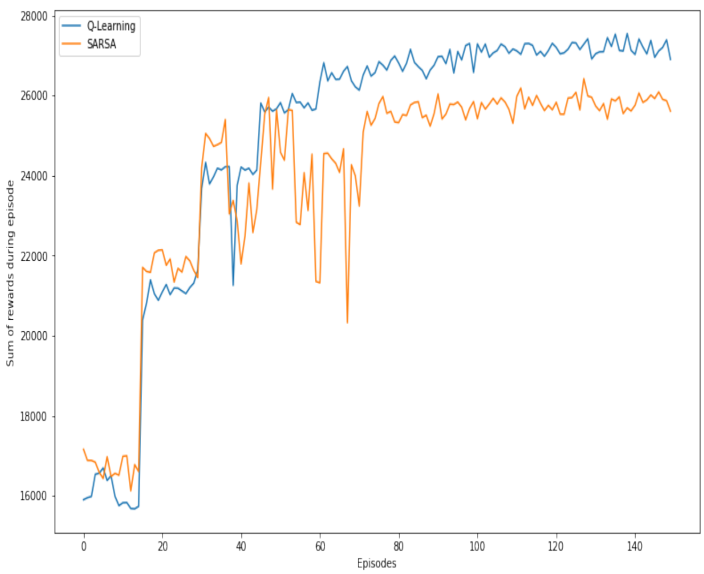
When C= 0 and Alpha = 0.005
There is no market buy or market sell action taken in the optimal policy of SARSA and Q-Learning. when Ask fill probability is greater than 0.5 and agent is in short position the algorithm chooses to buy as the best action.

When C = 0.05, and Alpha = 0.005
In the Optimal Policy, when Ask_fill probability is equal to 1 and agent is in short position the algorithm chooses market buy as the best action. Same trend is appeared in the Q-Learning as well.

Google dataset Results:

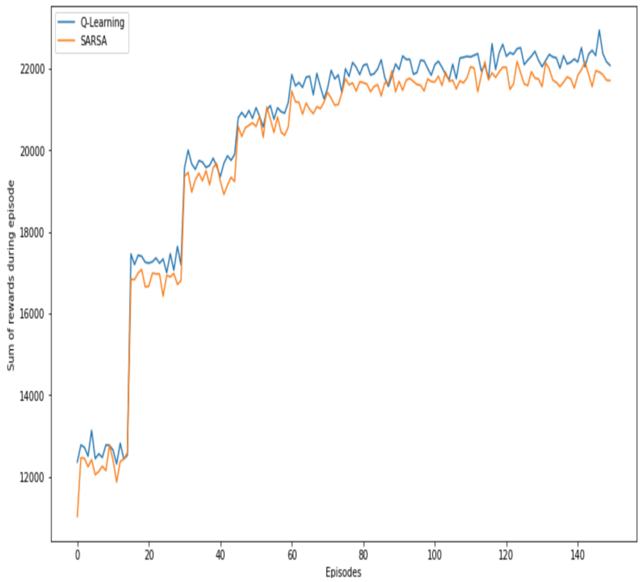
With C= 0 & Alpha = 0.05

```
SARSA
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
          flat    b    b    b    b    b    b    b    b    b    b
         short    b    b    b    b    b   mb   mb   mb   mb   mb
          long    h    s    s    h    s    s    s    s    s    s
Q-learning
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
          flat    b    b    b    b    b    b    b    b    b    h
         short    b    b    h    b    b   mb   mb    b   mb   mb
          long    h    h    s    s    s    s    s    s    s    s
```



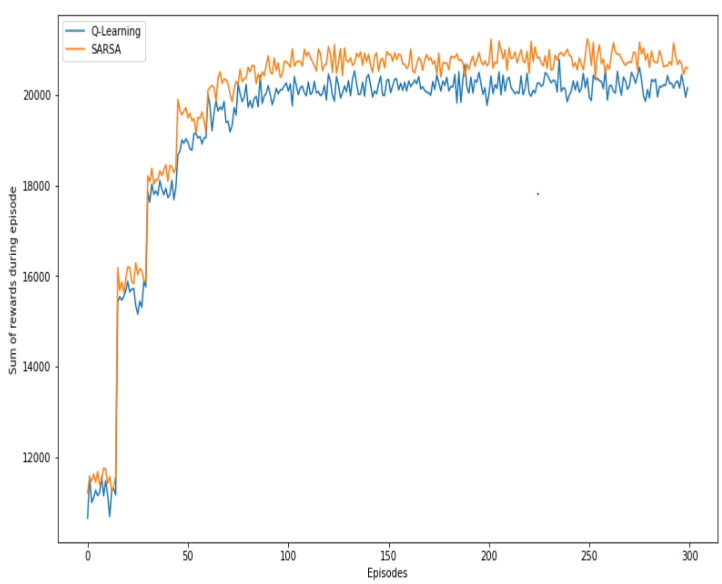
With C= 0 & Alpha = 0.005

```
SARSA
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
          flat    b    b    b    b    b    b    b    s    s    h
         short    b    b    b    b    b    b    b    b    b    h
          long    s    h    s    s    s    s    s    s    s    s
Q-learning
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
          flat    b    b    b    b    b    b    s    b    s    s
         short    b    b    b    b    b    b    b    b    b   mb
          long    s    h    s    s    s    s    s    s    s    s
```



With C= 0.05 & Alpha = 0.005

```
SARSA
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
         flat     b    b    b    b    b    b    b    h    b    s
        short     b    b    b    b    b    b    b    b    b    h
         long     s    s    h    s    s    s    s    s    s    s
Q-learning
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
         flat     b    h    s    s    h    s    s    s    s    s
        short     b    b    b    b    b    b    b    b    b    h
         long     s    s    s    s    h    s    s    s    s    s
```



Google dataset Conclusion:

Important findings

When C= 0 and Alpha = 0.05
Same trend as Amazon dataset.

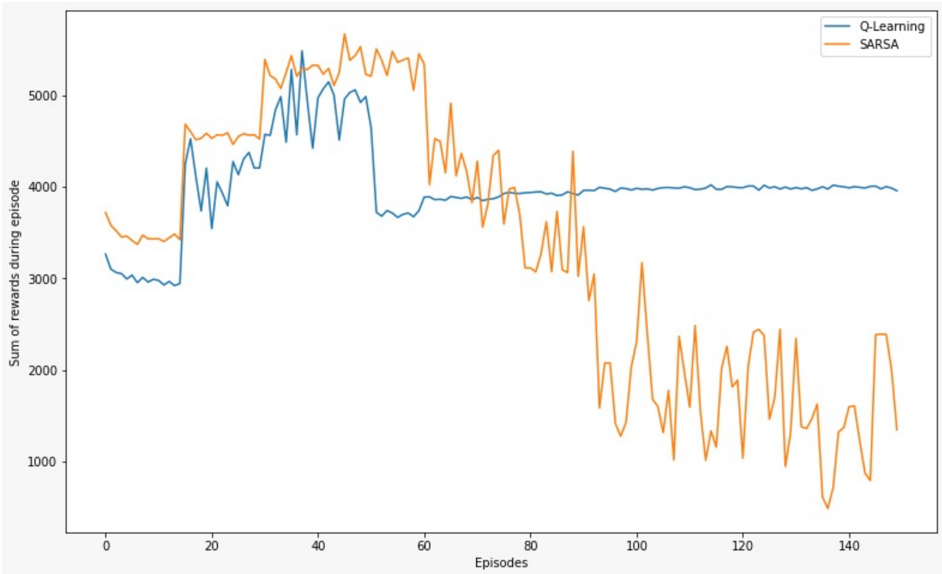
When C= 0 and Alpha = 0.005
There is no market buy or market sell action taken in the optimal policy of SARSA but in Q-Learning.There is no market buy or market sell action taken in the optimal policy of SARSA but in Q-Learning, except in Q-Learning when Ask fill probability is equal to 1 and agent is in short position the algorithm chooses market buy as the best action.

When C = 0.05, and Alpha = 0.005
In the Optimal Policy, when Ask_fill probability is equal to 1 and agent is in short position the algorithm chooses hold as the best action. Same trend is appeared in the Q-Learning as well.

Microsoft dataset Results:

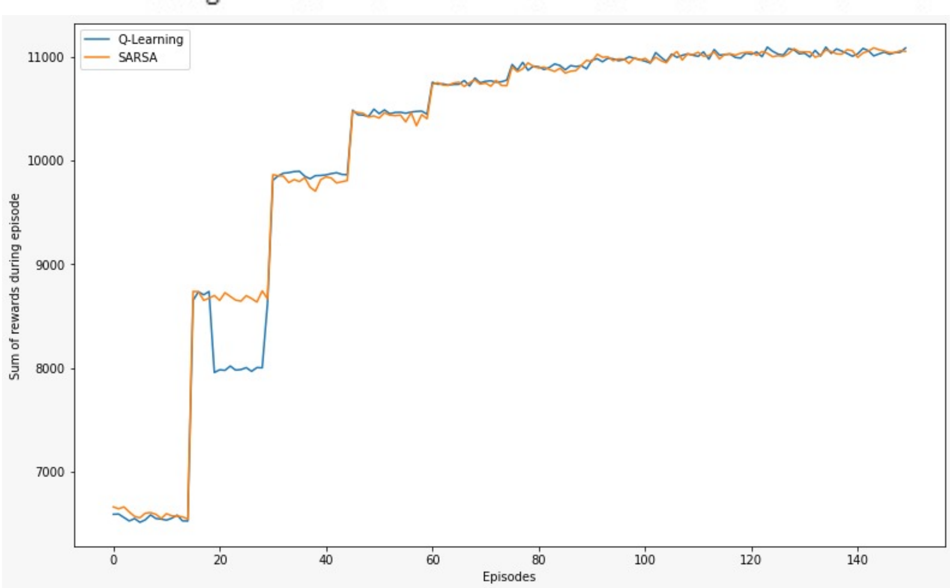
With C= 0 & Alpha = 0.05

```
SARSA
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
         flat     b    b    b    b    b    b    h    h    h    b
        short     b    b    b    b    b   mb   mb   mb   mb   mb
         long     s    h    h    h    s   ms   ms   ms    s    s
Q-learning
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
         flat     b    h    b    b    b    b    b    b    b    h
        short     b    b    b    b   mb   mb   mb   mb   mb   mb
         long    ms   ms   ms   ms   ms    s    s    s    s    s
```



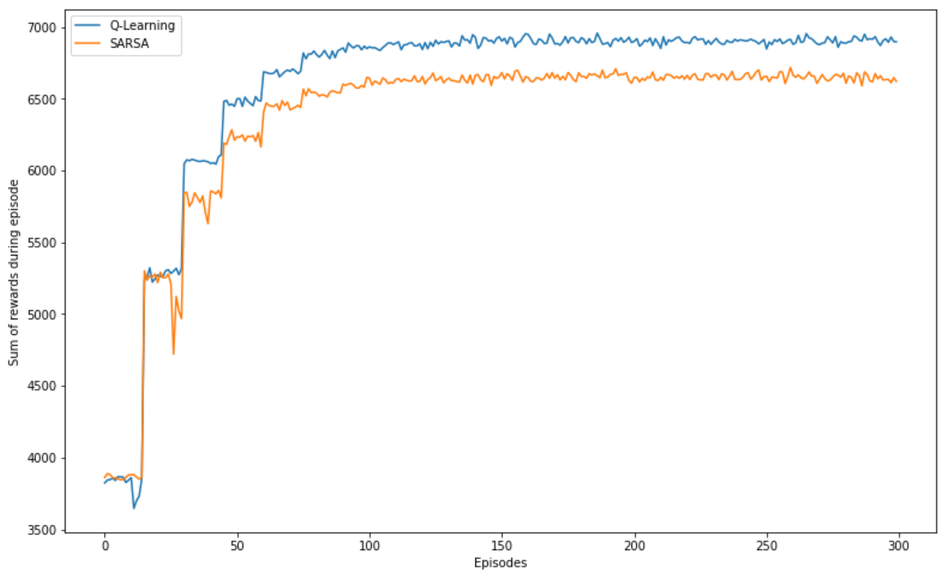
With C= 0 & Alpha = 0.005

```
SARSA
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
         flat     b    b    b    b    b    s    s    s    s    h
        short     b    b    b    b    b    b    b    b    b    mb
         long     ms   s    s    s    s    s    s    s    s    s
Q-learning
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
         flat     b    b    b    b    b    b    s    s    s    s
        short     b    b    b    b    b    b    b    b    b    b
         long     s    s    s    s    s    s    s    s    s    s
```



With C= 0.05 & Alpha = 0.005

```
SARSA
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
         flat     b    b    b    b    b    b    b    h    b    s
        short     b    b    b    b    b    b    b    b    b    h
         long     s    s    h    s    s    s    s    s    s    s
Q-learning
ask fill prob: 0.00 0.11 0.22 0.33 0.44 0.56 0.67 0.78 0.89 1.00
         flat     b    h    s    s    h    s    s    s    s    s
        short     b    b    b    b    b    b    b    b    b    h
         long     s    s    s    s    h    s    s    s    s    s
```



Microsoft dataset Conclusion:

Important findings

When C= 0 and Alpha = 0.05

In the Optimal Policy If Ask_fill probability is greater than 0.5 and agent is in short position, the algorithm chooses market buy as the best action. As it is appropriate to choose the market buy option when there is less chance of bid_fill in the short position. Additionally, when in long position and ask_fill probability is less than 0.5 the algorithm chooses market sell as the best action.

In Q-Learning, Policy If Ask_fill probability is greater than 0.5 and agent is in short position, the algorithm chooses market buy as the best action.

When C= 0 and Alpha = 0.005

There is no market buy or market sell action taken in the optimal policy of SARSA but in Q-Learning, except in SARSA when Ask fill probability is equal to 1 and agent is in short position the algorithm chooses market buy as the best action and when Ask fill probability is equal to 0 and agent is in short position the algorithm chooses market sell as the best action

When C = 0.05, and Alpha = 0.005
In the Optimal Policy, when Ask_fill probability is equal to 1 and agent is in short position the algorithm chooses hold as the best action. Same trend is appeared in the Q-Learning as well.

Future Work

1. Implementation through Deep Reinforcement learning
2. Implementation of queue positions, cancellations, limit order, Market order placement at different levels
3. Monte Carlo Evaluation of Market Making (Through Permutation Principle)

Bibliography:

Wikipedia, Investopedia, ML in Finance book