CitiBike Redistribution with Reinforcement Learning

Team Adelaide 2020 MMAI

MMAI 845 - Reinforcement Learning & Application

Agenda

- CitiBike Overview
- Environment & Problem
- Solving the problem
- Comparing Results
- Best method
- Next Steps

*"I left my bike beside a wall the other day, and it fell over. It was **two tyred**."*

CitiBike Overview

Convenient & popular bike sharing program in NY with over 800 stations

Uneven bike distribution at CitiBike locations in New York

Not having enough bikes or having too many bikes is costly

How many bikes should we remove or add per hour from a location?

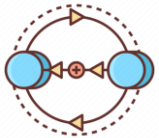
Environment

Location

W 82nd & Central Park West
(Central Park, New York City)

Reward function

Based on stock threshold per hour, movement of bikes



State Transition

3 months of usage history
for every hour of the day

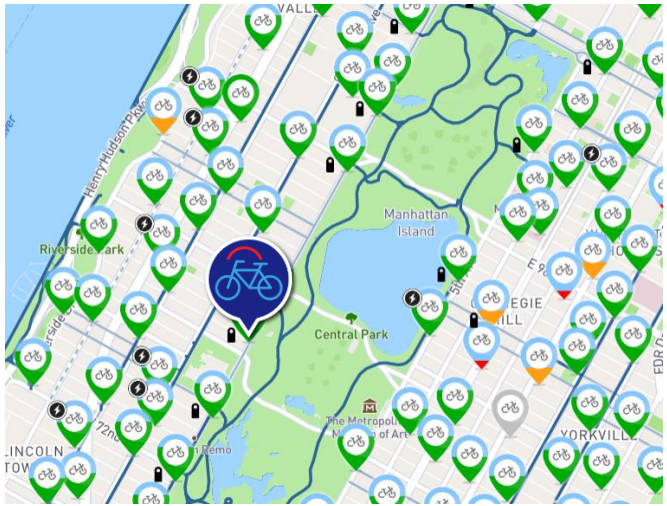
Action space

Add/remove bikes or do nothing
per hour at the station

Problem Specification

Location overview:

- Central Park Location: 0 - 45 capacity
- Overstock: 45 and above
- Understock: 0 and below
- Starting stock: 20 bikes every episode
- 1 Episode = 24 steps (24 hours)



Location

State Transition

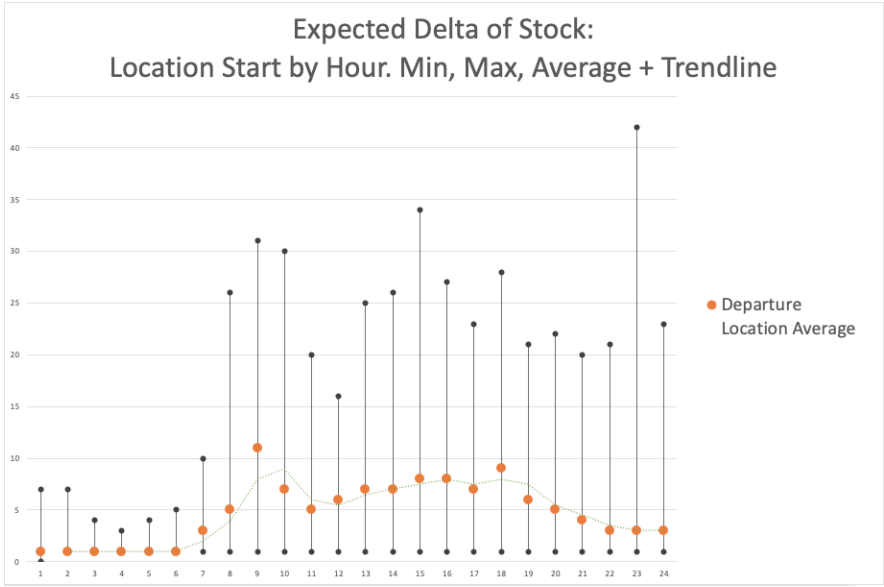
Action space

Reward function

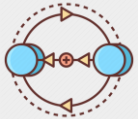
Problem Specification

Expected Stock Generation:

- Difference between arrival and departure of bikes is the per hour net expected stock.
- Historical mean and standard deviation to generate a random number
- Both arrival and departure at given hour



Location



State Transition

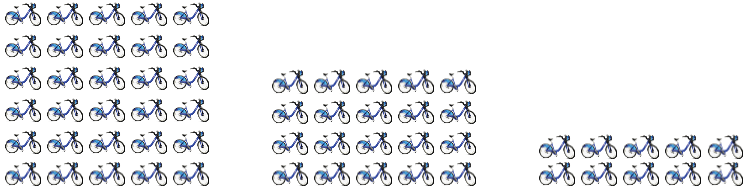
Action space

Reward function

Problem Specification

Action Set:

- Expected stock is random, action space needs to reflect it
- Small number of bikes in action set isn't effective
- 7 possible actions per hour: add, remove or do nothing



Location

State Transition

Action space

Reward function

Problem Specification

Rewards

- Threshold set to 5-40, ensure minimum number of bikes and parking spots
- -0.5 reward for every bike moved per hour to minimize movement
- -30 reward applied if threshold not met at the end of hour
- 0 if threshold met at the end of hour and end of day

Location

State Transition

Action space

Reward function

Solving the problem

RL Algorithm

Q-learning (off policy)

VS

SARSA (on policy)

Settings

- Epsilon = 0.1/0.01
- Discount Factor = 0.9/0.1
- Episodes = 100 – 20k

Evaluation

- Session success rate
- Average rewards/ session
- Stock history

Approach – “Ablation Study”

- Start with a base method: Q learning
- Compare w/ random policy & “do nothing” agent
- Compare base method with SARSA
- Tune hyperparameters one at a time for both

Q learning – Base Method

Algorithm:
Q Learning

Epsilon : **0.1**

Discount factor: **0.9**

Learning rate: **0.01**

Sessions: **4**
[100, 1k, 5k, 10k]

Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes

Average Rewards & Session Success Rate

Q learning – Base Method

Algorithm:
Q Learning

Epsilon : **0.1**

Discount factor: **0.9**

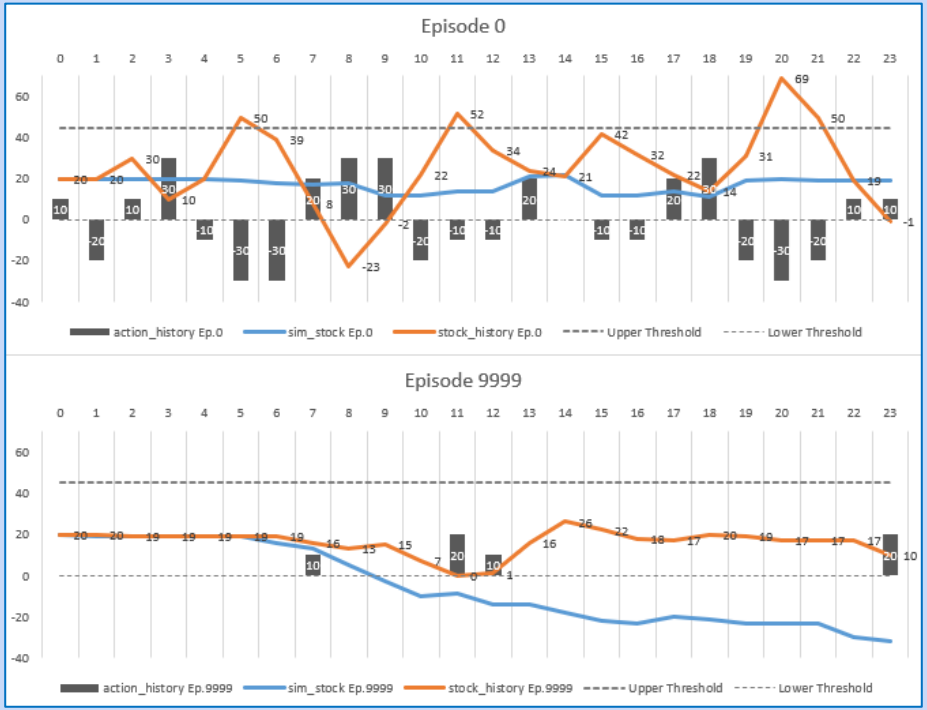
Learning rate: **0.01**

Sessions: **4**
[100, 1k, 5k, 10k]

Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes

Stock History



Q learning – No actions

Algorithm:
Q Learning

Epsilon : **0.1**

Discount factor: **0.9**

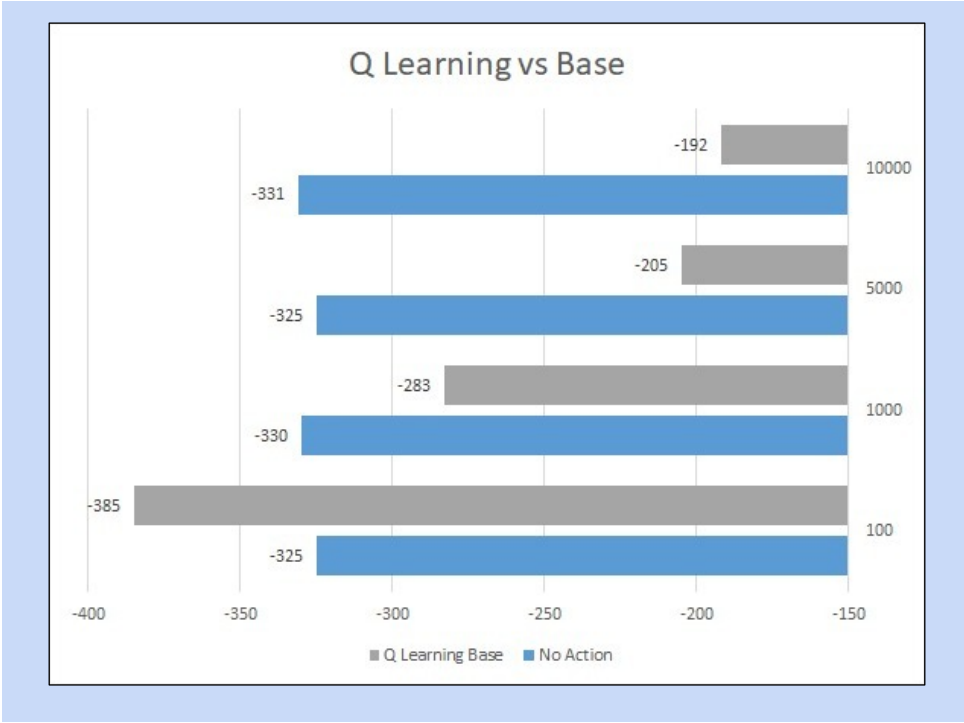
Learning rate: **0.01**

Sessions: **4**
[100, 1k, 5k, 10k]

Actions:
0

Threshold
5 - 40 bikes

Average reward per session



Q learning – No actions

Algorithm:
Q Learning

Epsilon : **0.1**

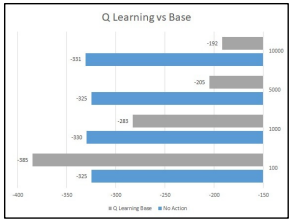
Discount factor: **0.9**

Learning rate: **0.01**

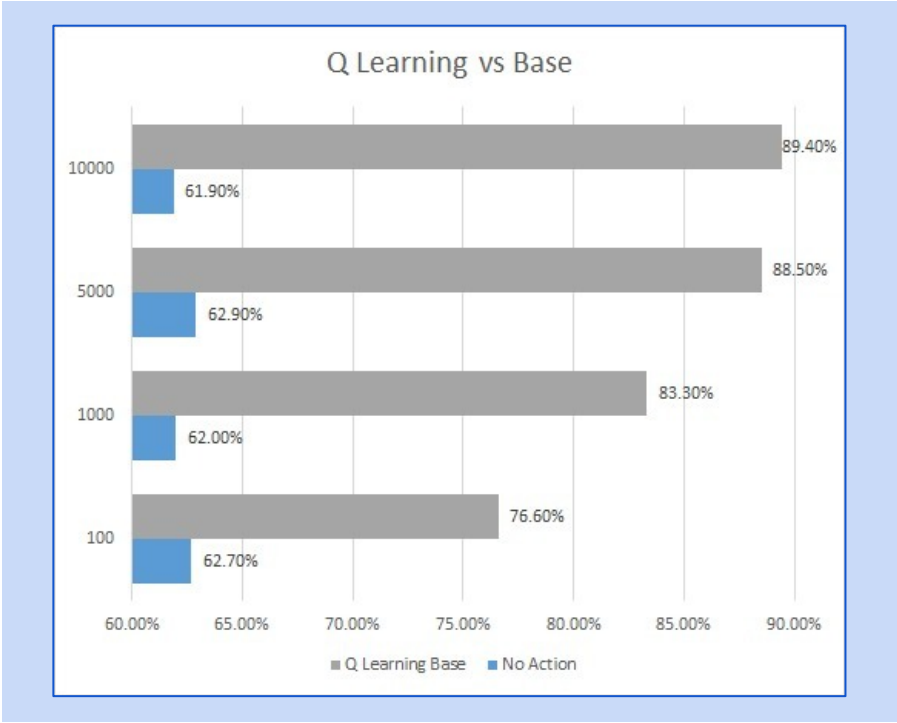
Sessions: **4**
[100, 1k, 5k, 10k]

Actions:
0

Threshold
5 - 40 bikes



Session success rate



Q learning – No actions

Algorithm:
Q Learning

Epsilon : **0.1**

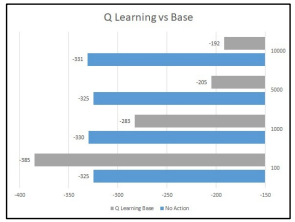
Discount factor: **0.9**

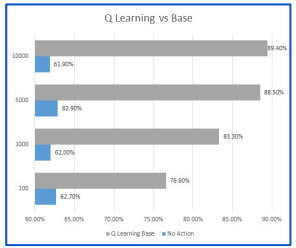
Learning rate: **0.01**

Sessions: **4**
[100, 1k, 5k, 10k]

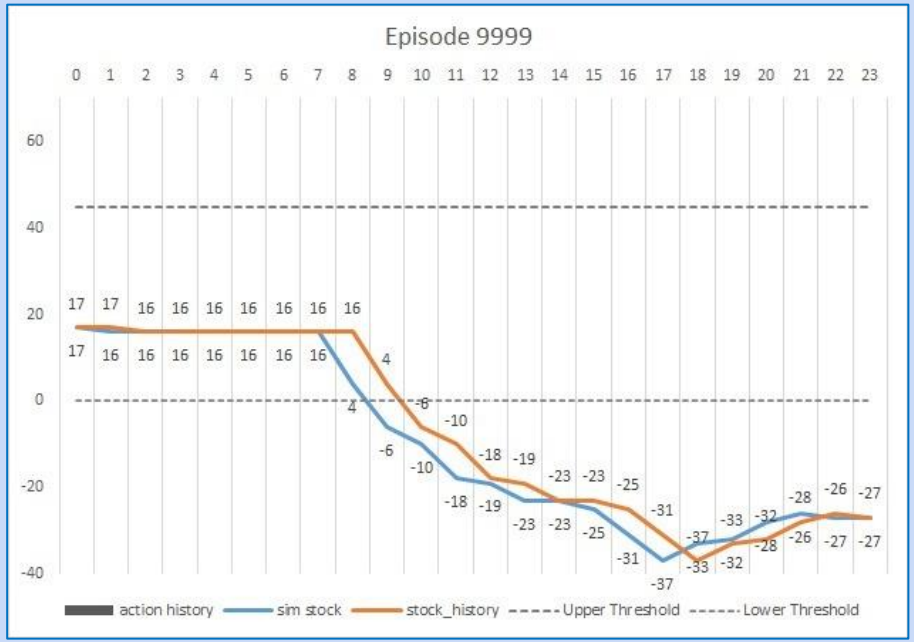
Actions:
0

Threshold
5 - 40 bikes





Stock history



Q learning – Random Policy

Algorithm:
Q Learning

Epsilon : **0.9**

Discount factor: **0.9**

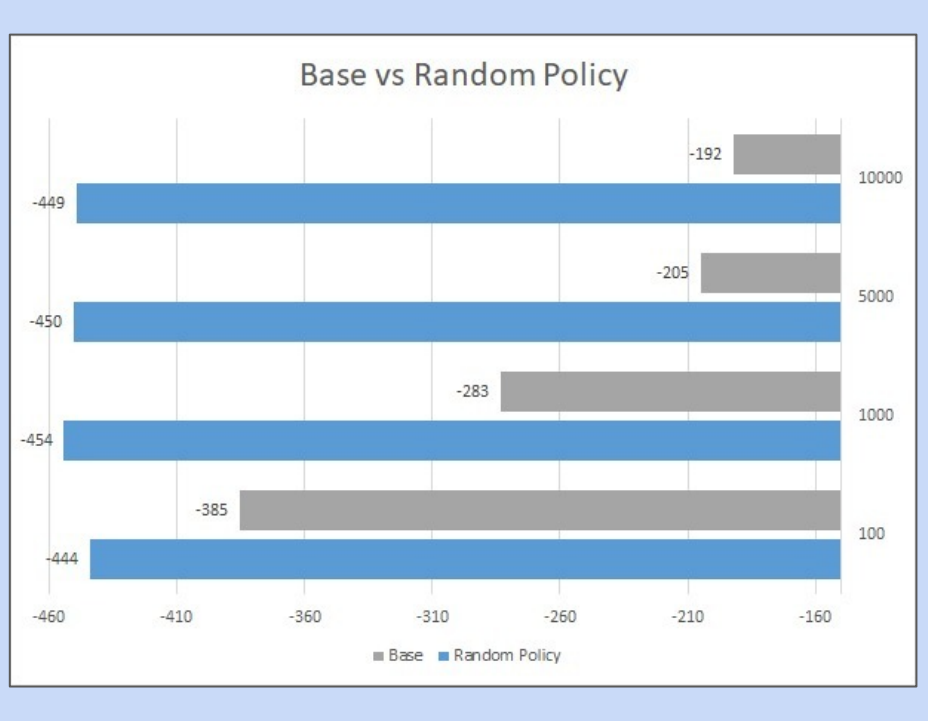
Learning rate: **0.01**

Sessions: **4**
[100, 1k, 5k, 10k]

Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes

Average Rewards per session



Q learning – Random Policy

Algorithm:
Q Learning

Epsilon : **0.9**

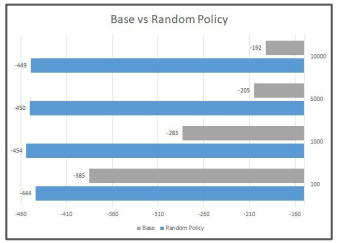
Discount factor: **0.9**

Learning rate: **0.01**

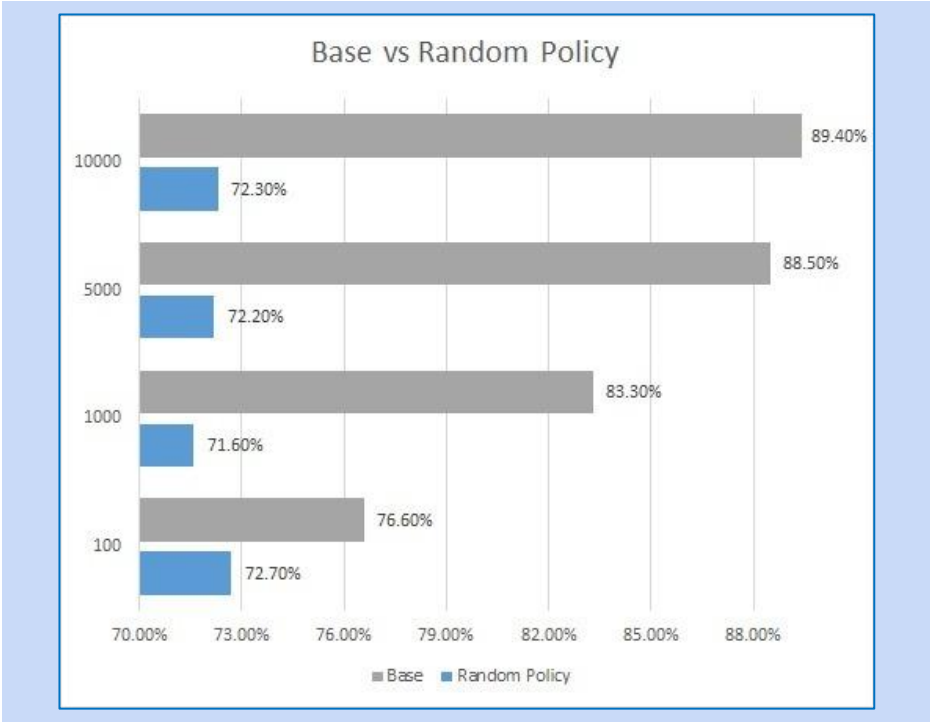
Sessions: **4**
[100, 1k, 5k, 10k]

Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes



Session Success Rate



Q learning – Random Policy

Algorithm:
Q Learning

Epsilon : **0.9**

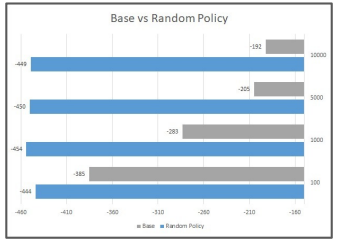
Discount factor: **0.9**

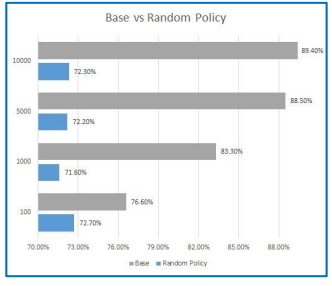
Learning rate: **0.01**

Sessions: **4**
[100, 1k, 5k, 10k]

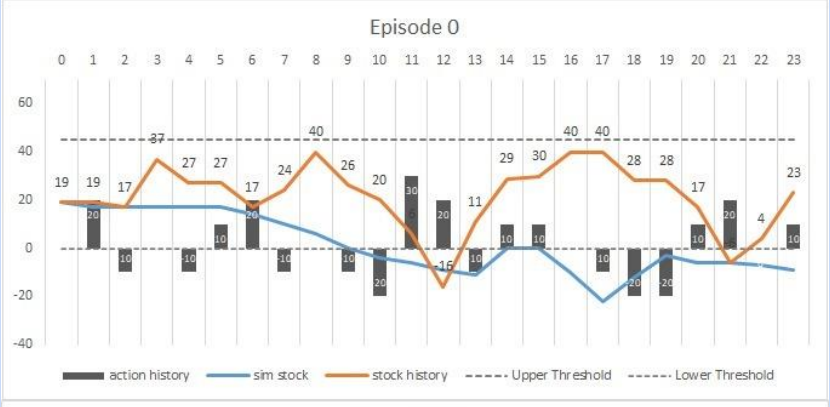
Actions:
+/- 0,10, 20, 30

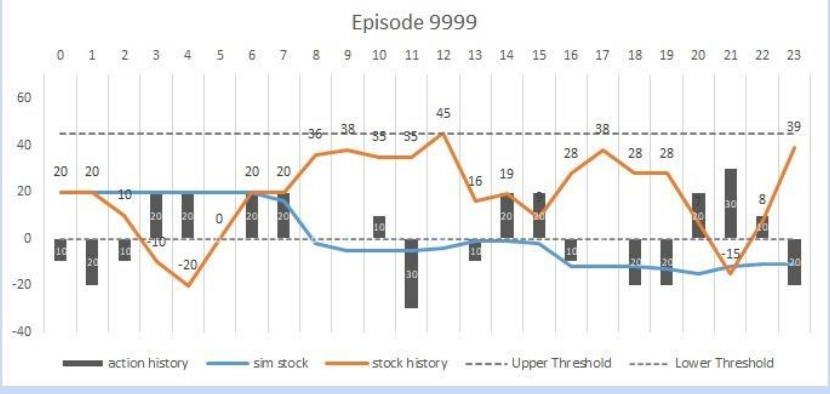
Threshold
5 - 40 bikes





Stock history





Q learning vs SARSA

Algorithm:
Q Learning vs SARSA

Epsilon : **0.1**

Discount factor: **0.9**

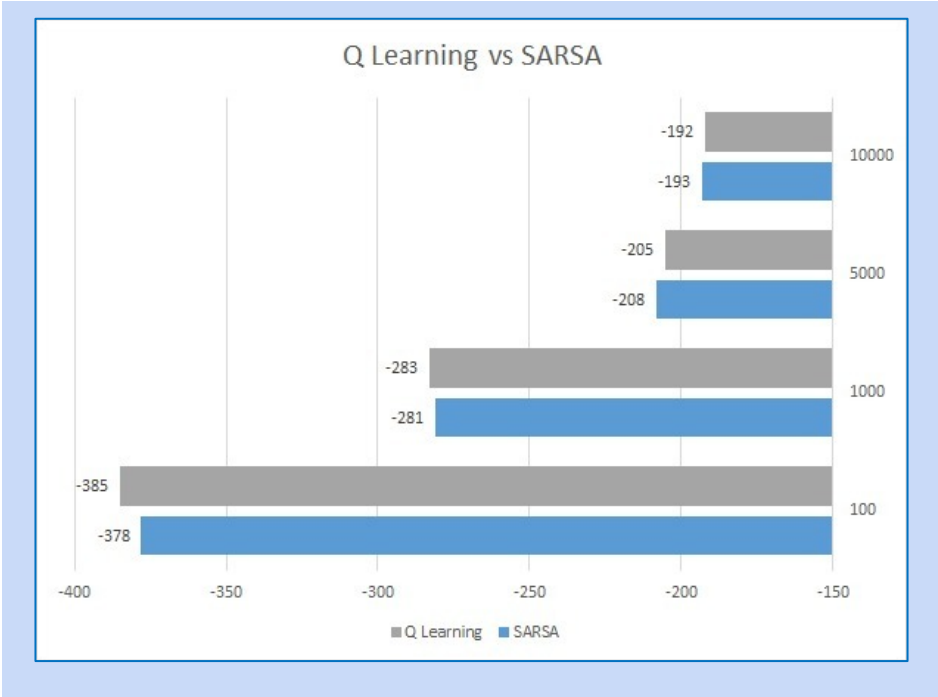
Learning rate: **0.01**

Sessions: **4**
[100, 1k, 5k, 10k]

Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes

Average reward per session



Q learning vs SARSA

Algorithm:
Q Learning vs SARSA

Epsilon : **0.1**

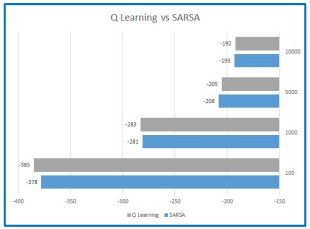
Discount factor: **0.9**

Learning rate: **0.01**

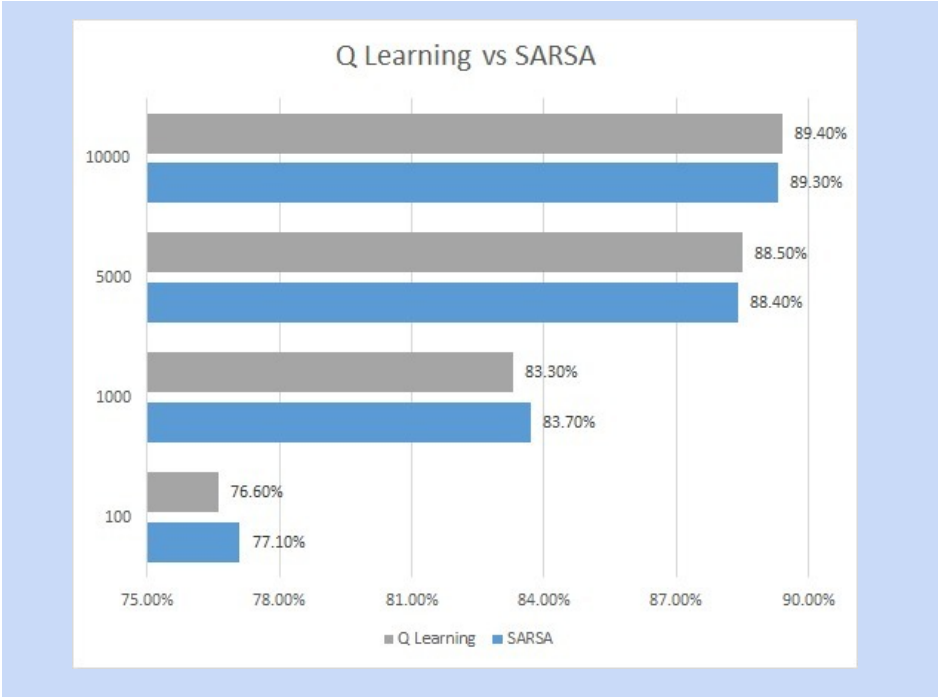
Sessions: **4**
[100, 1k, 5k, 10k]

Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes



Session Success Rate



Q learning vs SARSA

Algorithm:
Q Learning vs SARSA

Epsilon : **0.1**

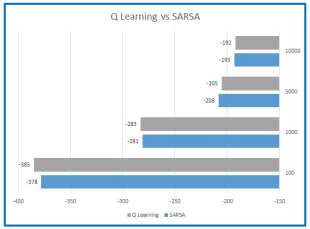
Discount factor: **0.9**

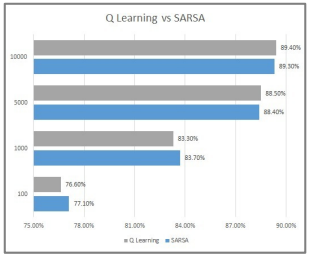
Learning rate: **0.01**

Sessions: **4**
[100, 1k, 5k, 10k]

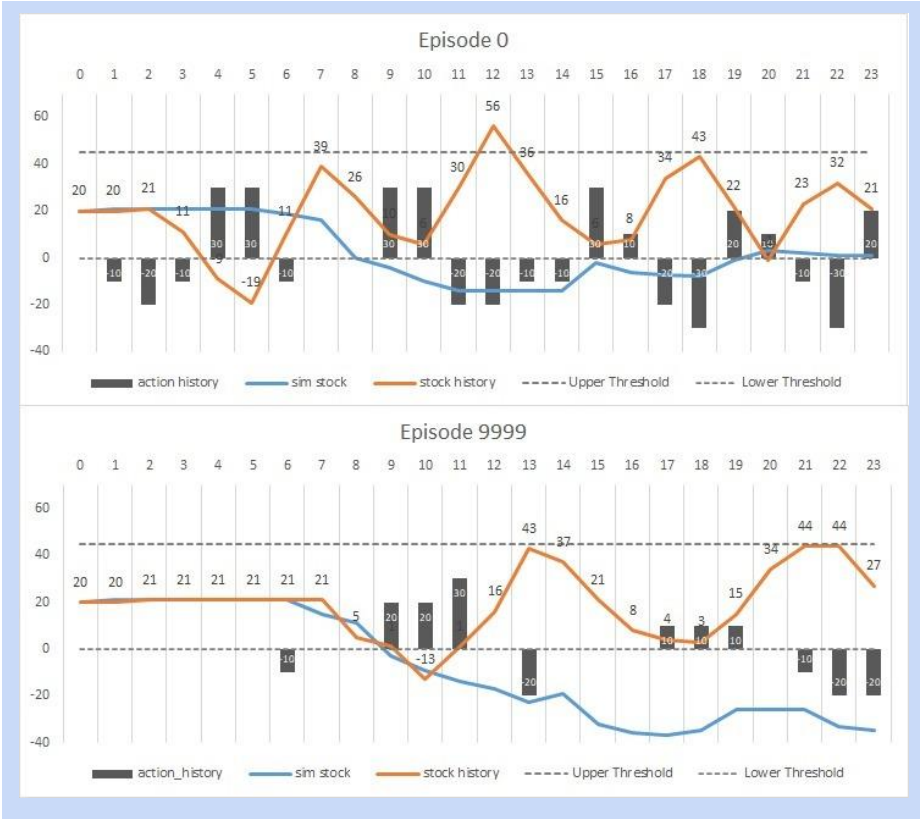
Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes





Stock History



Q Learning Hyperparameter: Epsilon

Algorithm:
Q Learning

Epsilon : **0.01**

Discount factor: **0.9**

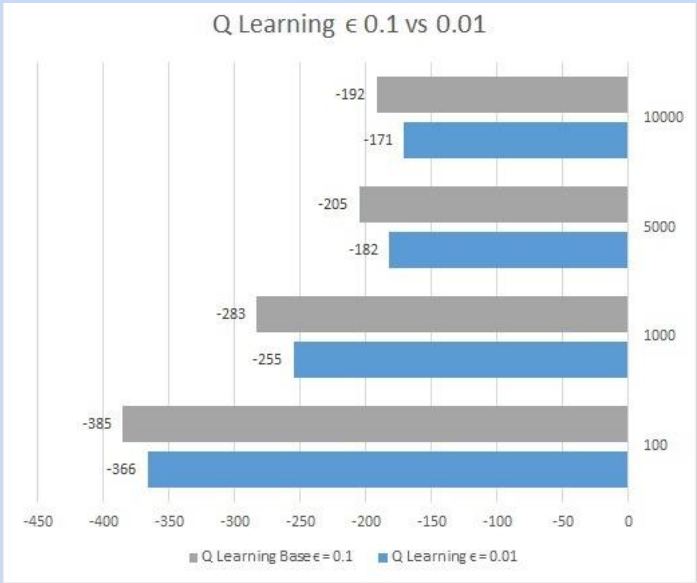
Learning rate: **0.01**

Sessions: **4**
[100, 1k, 5k, 10k]

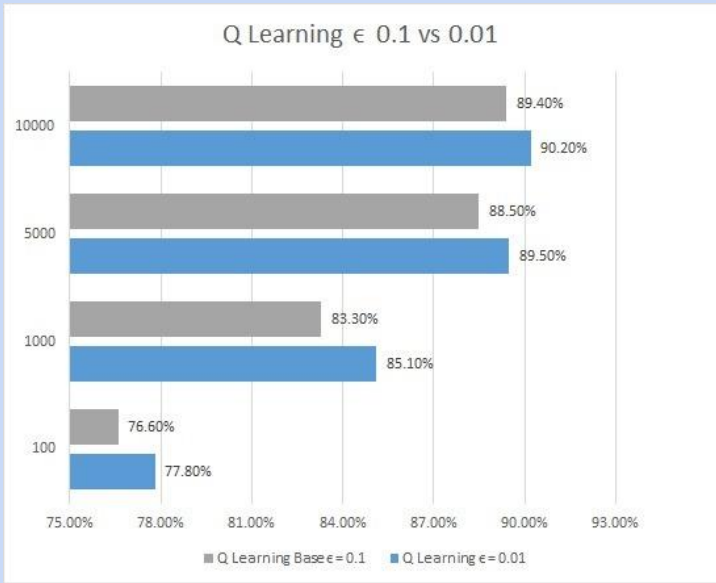
Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes

Average reward



Session Success



SARSA Hyperparameter: Epsilon

Algorithm:
SARSA

Epsilon : **0.01**

Discount factor: **0.9**

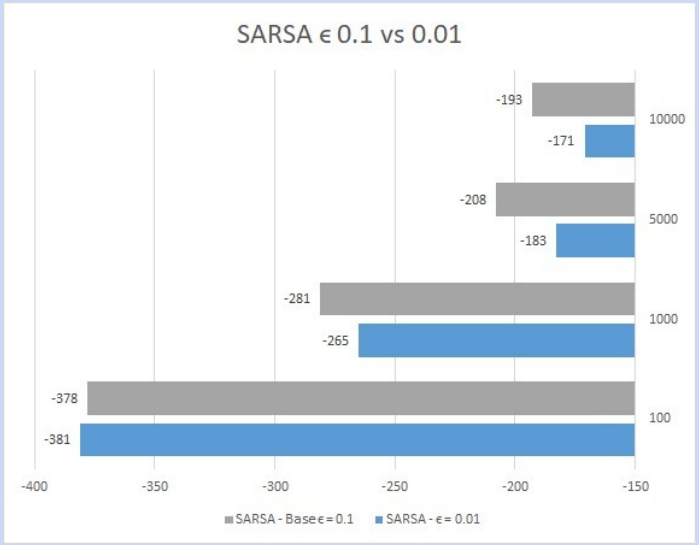
Learning rate: **0.01**

Sessions: **4**
[100, 1k, 5k, 10k]

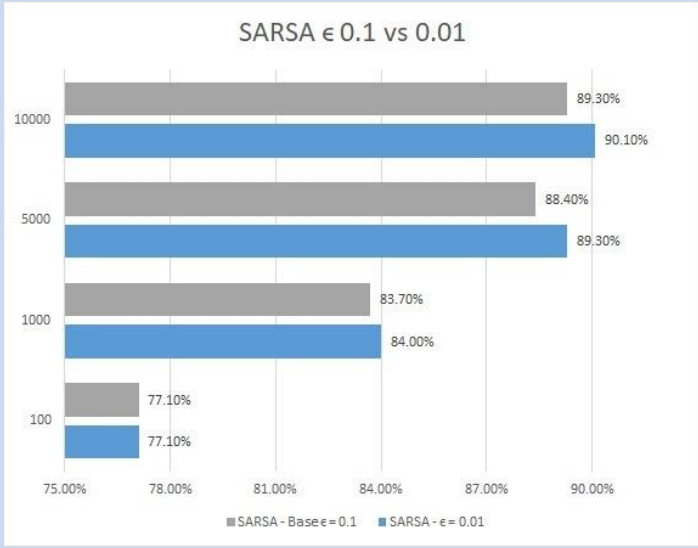
Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes

Average reward



Session Success



Hyperparameter: Discount Factor

Algorithm:
Q Learning vs SARSA

Epsilon : **0.01**

Discount factor: **0.1**

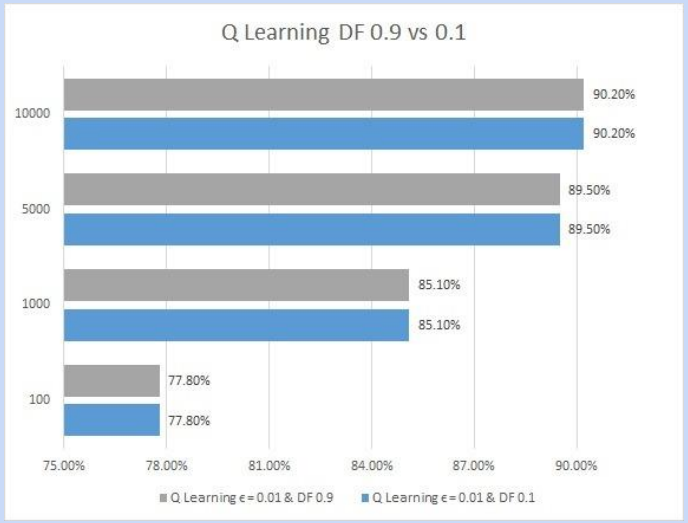
Learning rate: **0.01**

Sessions: **4**
[100, 1k, 5k, 10k]

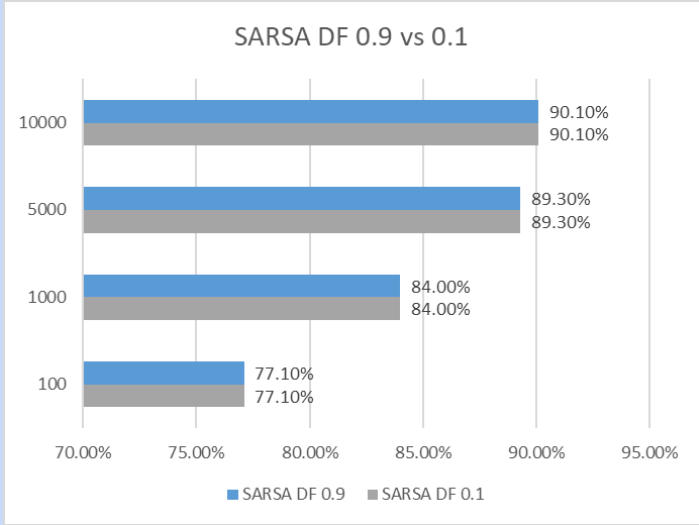
Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes

Session Success



Session Success



Training time – Q Learning

Algorithm:
Q Learning

Epsilon : **0.01**

Discount factor: **0.9**

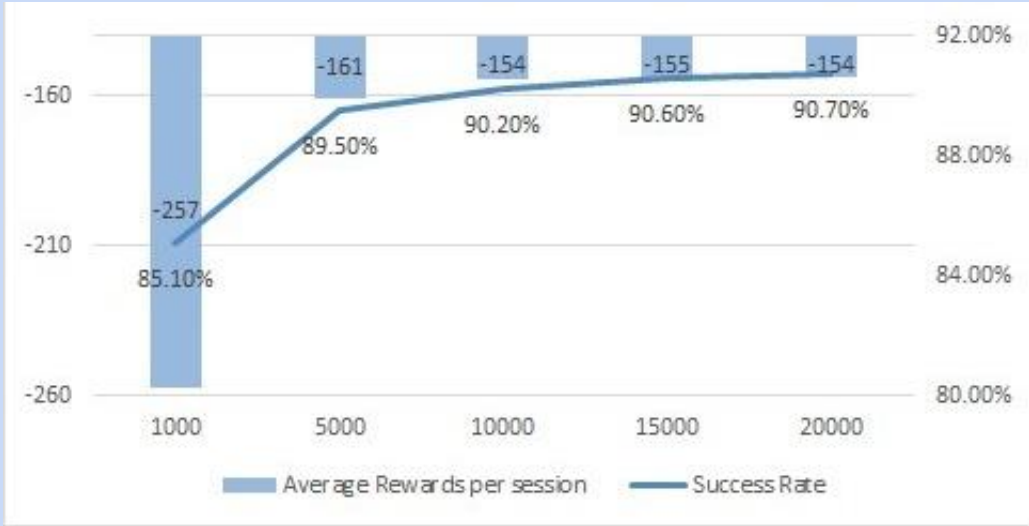
Learning rate: **0.01**

Sessions:
[1 - 20k]

Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes

Avg reward & Session Success



Training time – SARSA

Algorithm:
SARSA

Epsilon : **0.01**

Discount factor: **0.9**

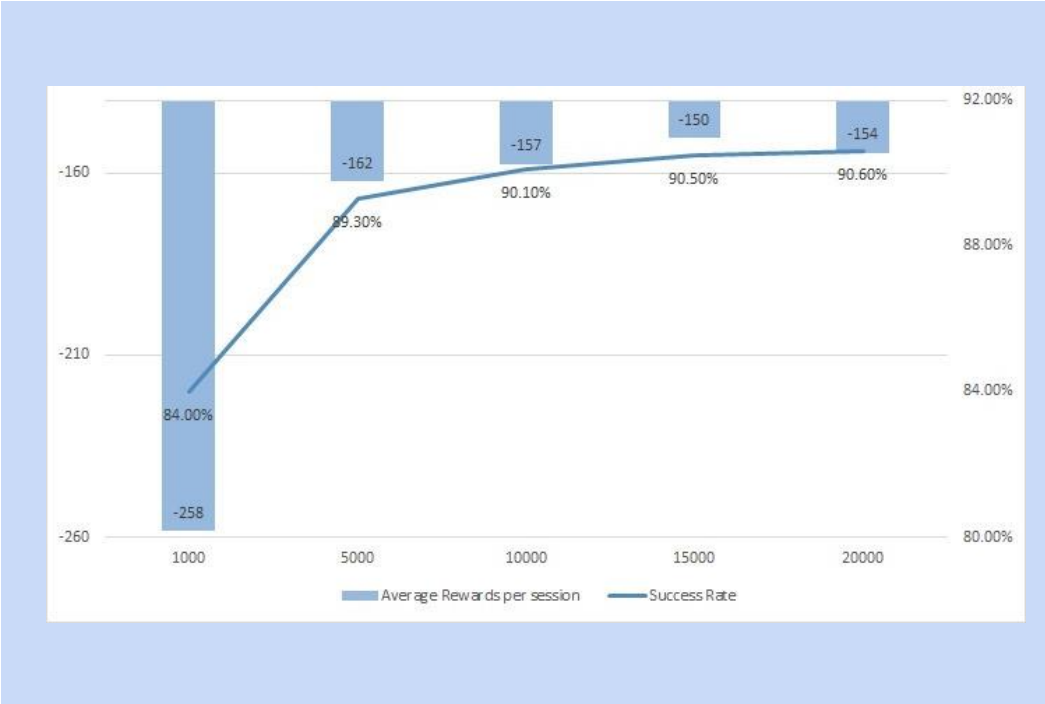
Learning rate: **0.01**

Sessions:
[1 – 20k]

Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes

Avg reward & Session Success



Best Method

Epsilon	DF	LR	Sessions	Success
0.1	0.9	0.01	10,000	89.40%
0.01	0.9	0.01	10,000	90.20%
0.01	0.1	0.01	10,000	90.20%
0.01	0.9	0.01	20,000	90.70%

Q Learning

Epsilon	DF	LR	Sessions	Success
0.1	0.9	0.01	10,000	89.30%
0.01	0.9	0.01	10,000	90.10%
0.01	0.1	0.01	10,000	90.10%
0.01	0.9	0.01	20,000	90.60%

SARSA

Best Method

Epsilon	DF	LR	Sessions	Success
0.1	0.9	0.01	10,000	89.40%
0.01	0.9	0.01	10,000	90.20%
0.01	0.1	0.01	10,000	90.20%
0.01	0.9	0.01	20,000	90.70%

Q Learning

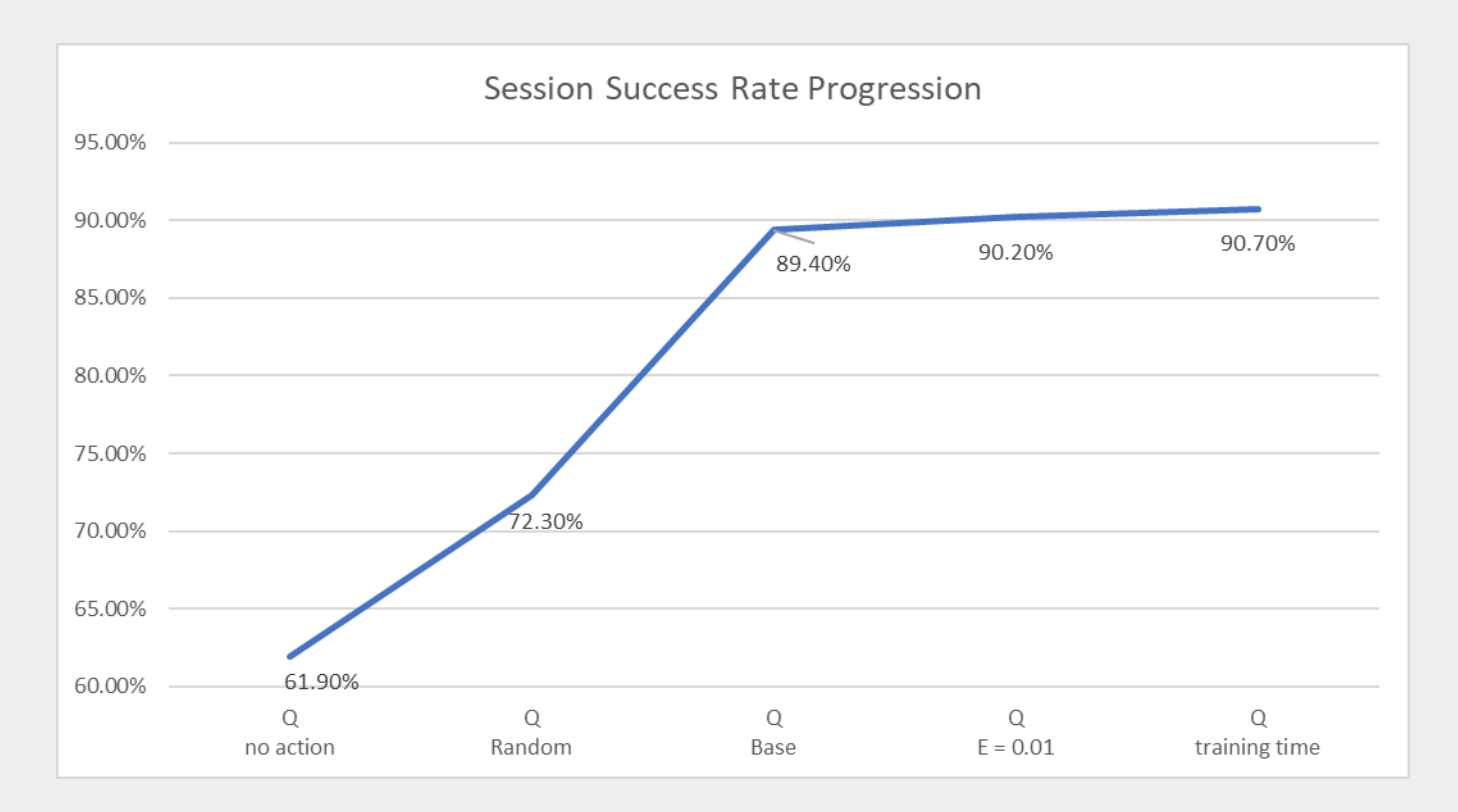
Epsilon	DF	LR	Sessions	Success
0.1	0.9	0.01	10,000	89.30%
0.01	0.9	0.01	10,000	90.10%
0.01	0.1	0.01	10,000	90.10%
0.01	0.9	0.01	20,000	90.60%

SARSA

Optimal Policy- Q table

Hour \ Stock	2	6	10	14	20	25	30	36	42
3	10	20	0	10	0	0	0	-10	-20
6	10	0	0	0	0	0	0	0	-20
10	0	0	0	0	0	0	0	0	-10
11	0	0	0	0	0	0	0	0	-30
13	-10	0	0	0	0	0	0	0	20
16	-20	0	0	0	0	0	0	0	20
18	-20	0	0	0	0	0	0	0	10
21	10	0	0	0	0	0	0	0	-10

Recap


Session Success Rate Progression
95.00%
90.00%
85.00%
80.00%
75.00%
70.00%
65.00%
60.00%
61.90%
72.30%
89.40%
90.20%
90.70%
Q
no action
Q
Random
Q
Base
Q
E = 0.01
Q
training time

Next Steps

Expected Stock:

- No free reset
- Different starting stock
- Scale this to a full year

Reward function

- Include time of day
- Different threshold based on hour

RL Algorithms

- DQN
- Monte Carlo

Thank You!

Appendix

Team Adelaide

Experiments

Algorithm	Action	Episodes	Threshold	Reward	Learning Rate	Epsilon	DF
Q- Learning Base	[-30,-20,-10,0,10,20,30]	100 - 1K - 5K - 10K	5 and 40	0.5, - 30	0.01	0.1	0.9
Q Learning No action	[0]	100 - 1K - 5K - 10K	5 and 40	0.5, - 30	0.01	0.1	0.9
Q-Learning random	[-30,-20,-10,0,10,20,30]	100 - 1K - 5K - 10K	5 and 40	0.5, - 30	0.01	0.9	0.9
SARSA base	[-30,-20,-10,0,10,20,30]	100 - 1K - 5K - 10K	5 and 40	0.5, - 30	0.01	0.1	0.9
Q-Learning new E	[-30,-20,-10,0,10,20,30]	100 - 1K - 5K - 10K	5 and 40	0.5, - 30	0.01	0.01	0.9
SARSA new E	[-30,-20,-10,0,10,20,30]	100 - 1K - 5K - 10K	5 and 40	0.5, - 30	0.01	0.01	0.9
Q-Learning new DF	[-30,-20,-10,0,10,20,30]	100 - 1K - 5K - 10K	5 and 40	0.5, - 30	0.01	0.01	0.1
SARSA new DF	[-30,-20,-10,0,10,20,30]	100 - 1K - 5K - 10K	5 and 40	0.5, - 30	0.01	0.01	0.1
Q-Learning new session	[-30,-20,-10,0,10,20,30]	1K - 10K - 15K - 20K	5 and 40	0.5, - 30	0.01	0.01	0.9
SARSA new session	[-30,-20,-10,0,10,20,30]	1K - 10K - 15K - 20K	5 and 40	0.5, - 30	0.01	0.01	0.9

Simulated success vs no action

Hyperparameter: Discount factor

Algorithm:
Q Learning vs SARSA

Epsilon : **0.01**

Discount factor: **0.1**

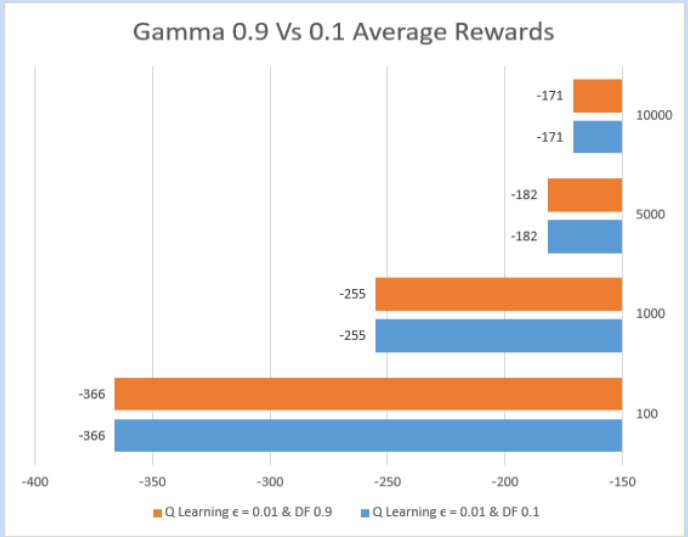
Learning rate: **0.01**

Sessions: **4**
[100, 1k, 5k, 10k]

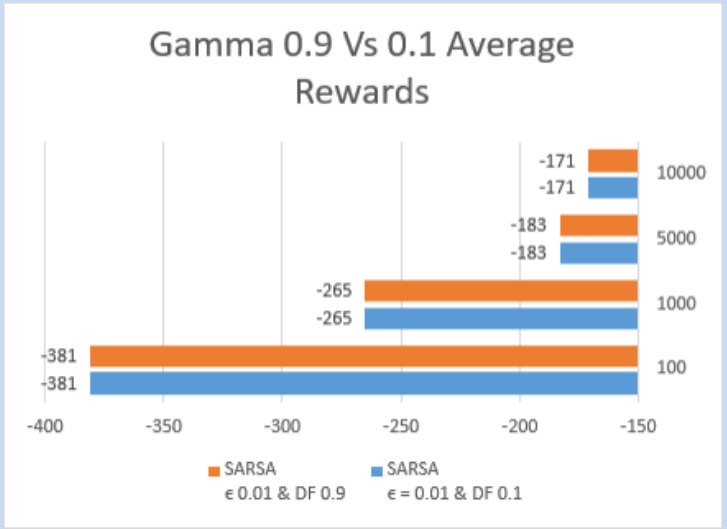
Actions:
+/- 0,10, 20, 30

Threshold
5 - 40 bikes

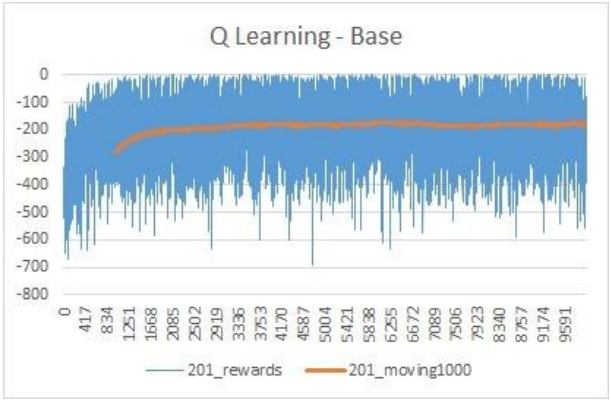
Average Reward

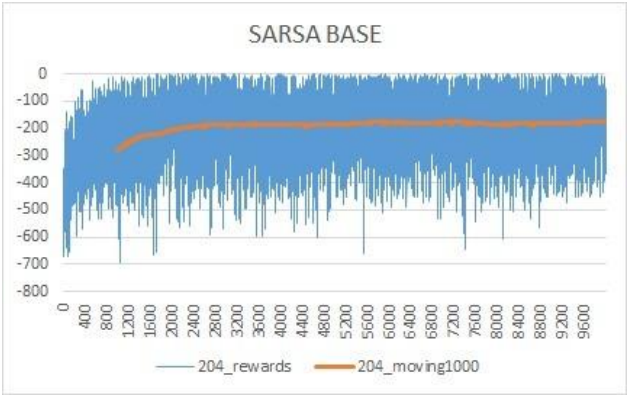


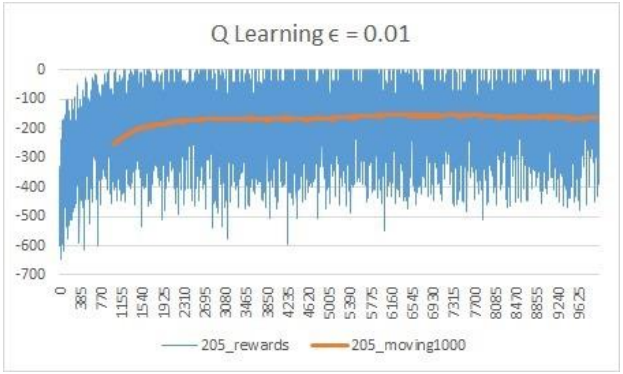
Average Reward

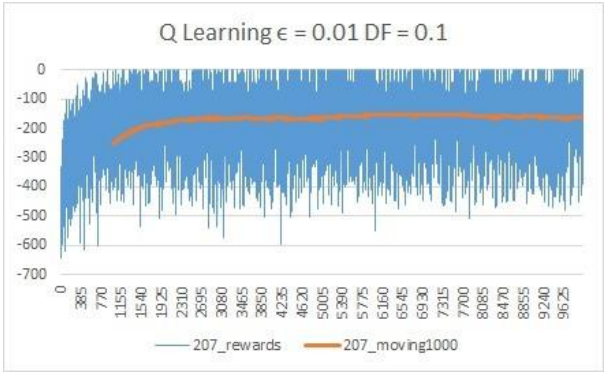


Rewards History









Rewards History