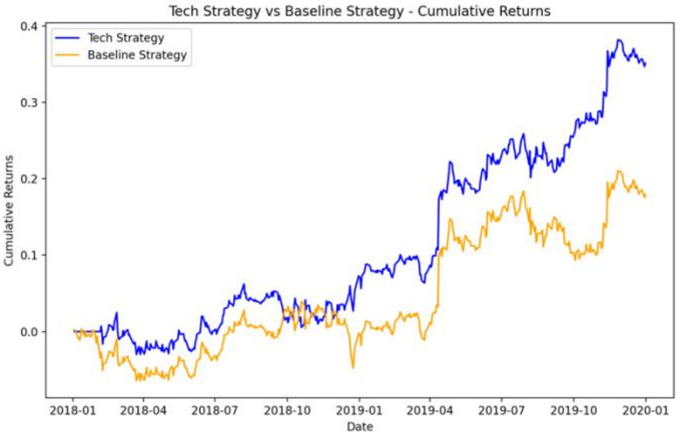
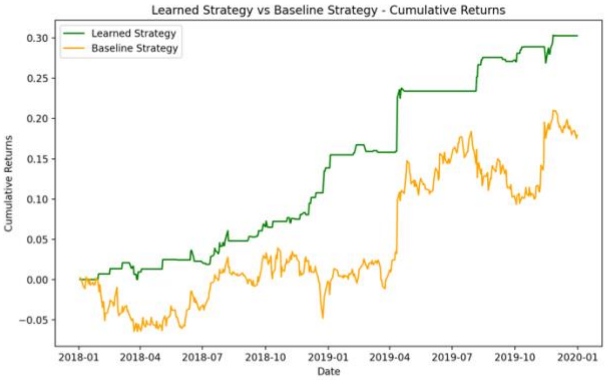
To frame the trading problem as a learning problem for my Q-learner I used three indicators to help represent a state. These were the moving average convergence/divergence oscillator (MACD), the relative strength index (RSI), and the Bollinger Band Percentage. Along with a record of the learner's current holdings of the given stock I quantized the state by first normalizing the values of each technical indicator. This allowed me to easily assign each indicator value across the given time frame into one of 6 bins numbered 1 to 6. From there I was able to quantize each state to a single number using the following formula: holdings + BBP*6**2 + RSI*6 + MACD.

- Experiment 1: Compare your in sample learned strategy with your in sample technical trading strategy from the previous project.



Cumulative return: 0.35115

Cumulative return: 0.3028



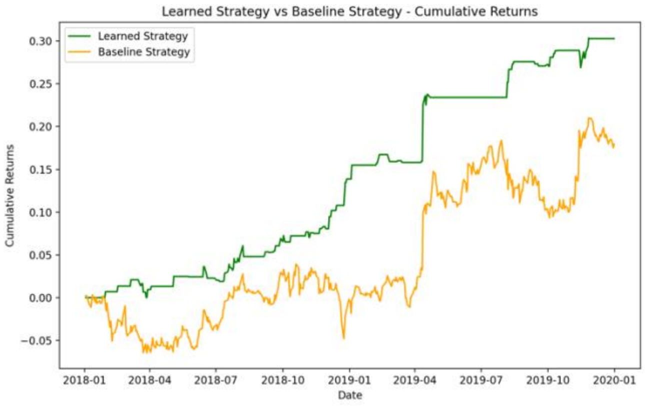
The technical indicators used to inform the states for my Q learner used the following parameters: the Bollinger bands used a lookback of 9 and 2 standard deviations to find the upper and lower bands. The signal line for MACD and overbought and oversold lines for RSI were not needed due to the data being assigned to one of 6 bins allowing the bins to take up some arbitrary meaning as interpreted by the learner. The experiment shows that the two strategies perform similarly when looking at the final cumulative return; however, the Q-learner strategy never dips below the baseline and appears to be a more reliable strategy that I imagine will perform better than the technical strategy in out of sample date ranges. I expect the Q-learner to outperform my technical strategy because it has all the same information, since the same technical indicators are baked into each state and is able to refine its decisions over many trips and therefore converge to a better strategy. The technical indicator can compete with in sample data because it was made to perform well on that specific data so its cumulative return is deceivingly high and not a reflection of how it would perform on other stocks and timeframes, whereas I expect the Q learner to adapt easily.

- Experiment 2: Provide a hypothesis regarding how changing the floating cost value should affect the Q-Learner.

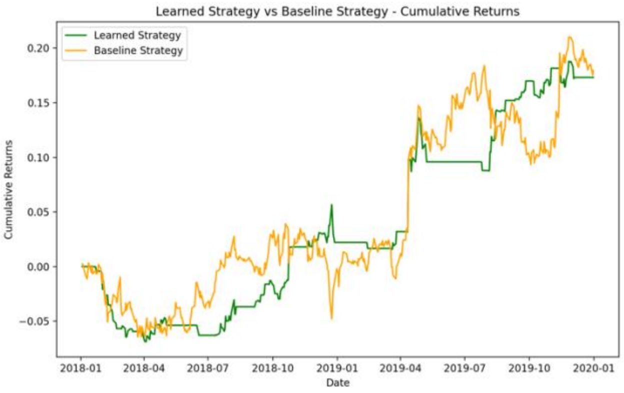
Hypothesis: Increasing the floating cost will cause the Q-Learner to make less trades and consequently less return.

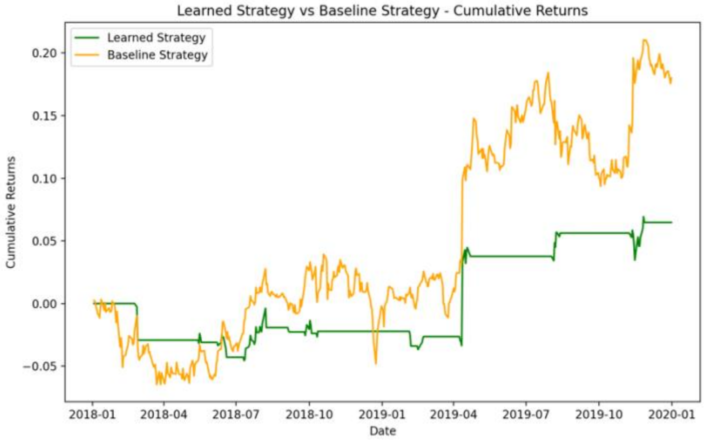
This is because the floating cost is tied to how I calculate reward, so increasing this cost will reduce the reward for buying and selling stock. I expect the Q-Learner to reduce its number of trades in order to incur less fees. This in turn would reduce the cumulative return as the trader will take advantage of price fluctuations less frequently.

Floating cost: 0.0



Floating cost: 0.0005





Floating cost: 0.005

This experiment aligns with my hypothesis. It can be seen that by increasing the floating cost from 0 to 0.0005 there was a reduction in trades and a resulting lower cumulative return. The hypothesis can be seen even more clearly when looking at a floating cost of 0 compared to 0.005. It can be seen that with the higher floating cost the Q-Learner stayed flat on DIS for most of the time, only choosing to trade a couple of times. This clearly shows the increased hesitancy to make trades due to decreased rewards. The reduced cumulative return is also evident as the higher floating cost resulted in a cumulative return of just 0. 0.064741.

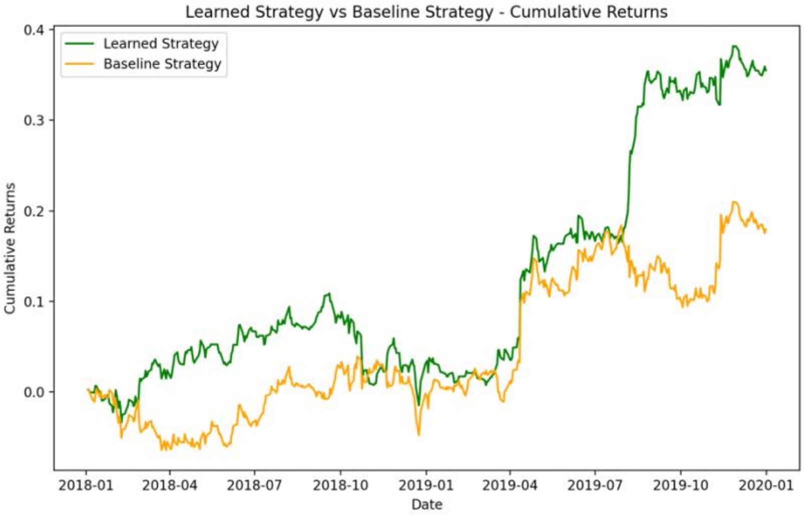
- For Experiments 1 and 2 note whether you used Tabular or Double Q.

I used a Tabular Q learner for experiments 1 and 2

- Experiment 3: Try various combinations of Tabular vs Double Q and Dyna vs not.

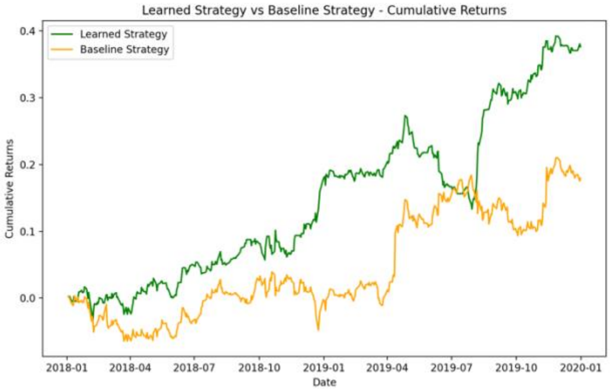
Tabular with dyna:

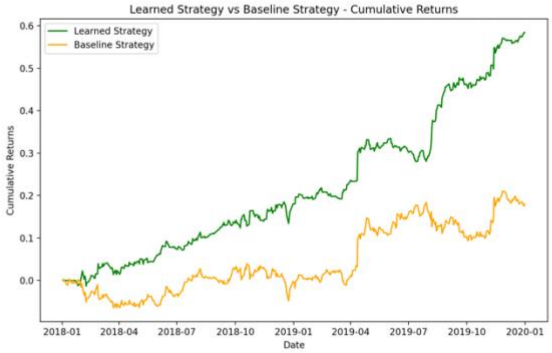
50 hallucinations, 50 trips: 0.3552 (cumulative return)



200 hallucinations, 50 trips: 0.37615

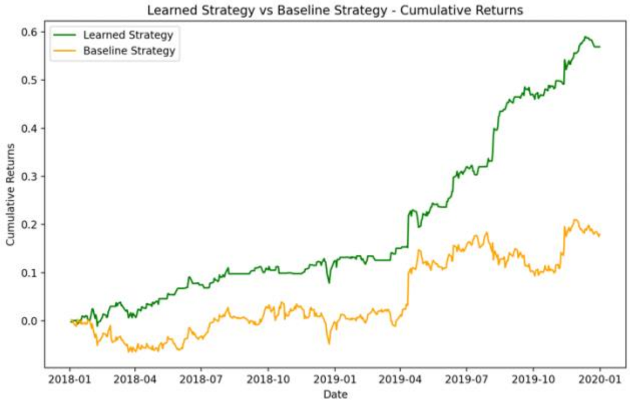
500 hallucinations, 50 trips: 0.58375





From experimenting with a Tabular learner and various levels of dyna it can be seen that more dyna can help, especially with a higher number of hallucinations. The plots above show that the cumulative return increased from 0.35 to 0.37 to 0.58 by increasing the dyna hyperparameter from 50 to 200 to 500.

Double Q-Learner: 0.5689
(cumulative return)

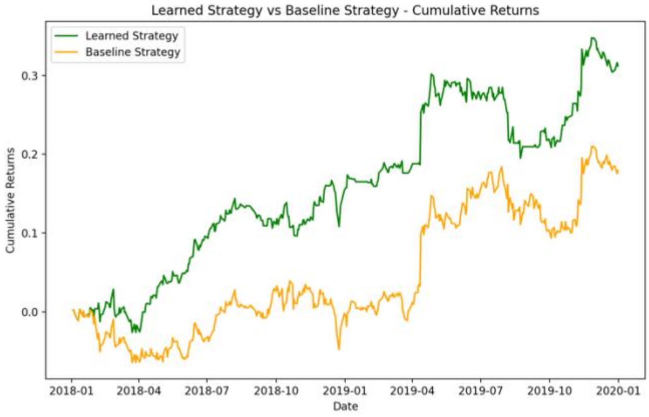


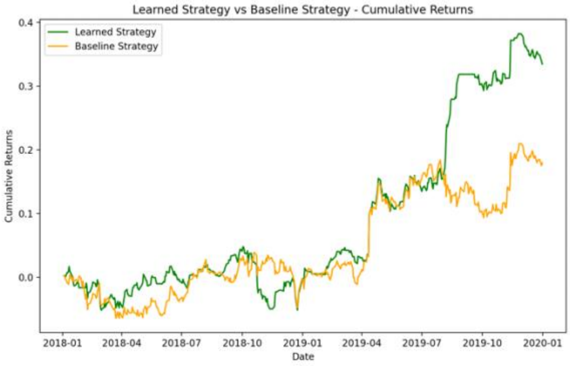
Double Q-Learner with dyna:

200 hallucinations, 200 trips

0.31205 (cumulative return)

500 hallucinations, 200 trips: 0.33475

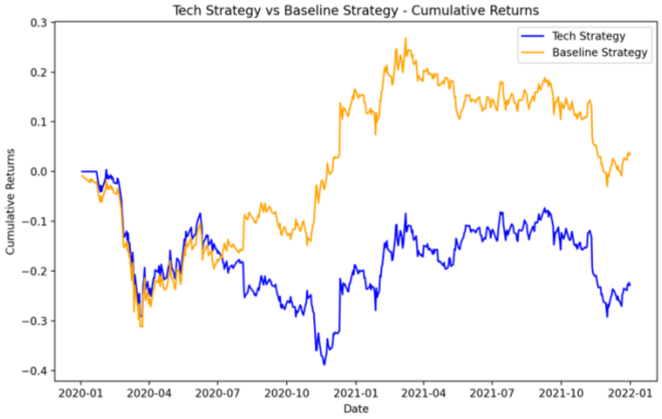


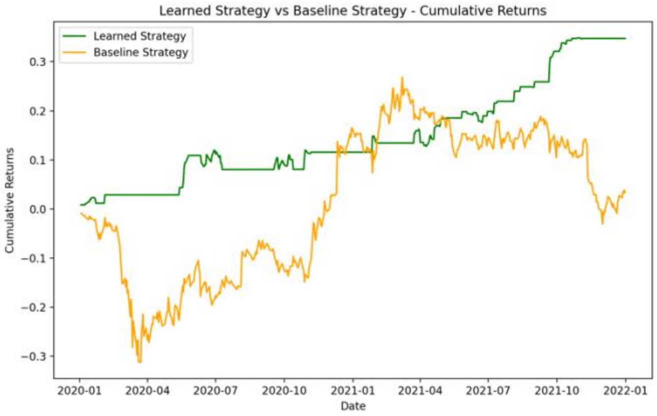


As for a Double Q-Learner it can be seen that it was very effective with a cumulative return of 0.57; Adding dyna, didn’t seem to help with cumulative return falling to around 0.3; however, this is with fewer trips to train the Q-learner.

Through trying various combinations it seems like a Double Q-Learner or a Tabular Q-Learner with a high number of dyna hallucinations produces the best trading strategy.

- Briefly note your out-of-sample results and how this compares to your previous project's out-of-sample results.





It can clearly be seen the Q-Learner greatly outperforms my technical indicator strategy from the previous project with cumulative returns of -0.2286 compared to 0.34755.