PROJECT 8: Strategy Evaluation
CS7637

Ashish Panchal
apanchal33@gatech.edu

Abstract—The project attempts to explore and study the implementation of different technical analysis indicators in forming trading strategies, manually and with Strategy learner, which outperform the Benchmark

1 INTRODUCTION

The strong form efficient market hypothesis suggests all the information in a market is depicted in a stock's price, i.e even with a lack of private and public information them, it is hypothesized that we can predict the future market behavior only with technical parameters, i.e only on historical price and volume of trades. In this project we attempt to explore trading strategies, created manually and with a help of predictive learning engine -Q learning.

Following Two initial hypothesis will be analyzed and validated, 1. With right TA indicators and parameter optimization a profitable strategy can be manually created. 2. Learning Engine can suggest better trading strategy than manual.

The following section explore 5 indicators used in this study, detailing how they create buy and sell signal in the manual strategy. The subsequent sections explore the use of these indicators in a Strategy learning engine, along with there transformation for prediction. The two strategies are contrasted in the in followed experiment, answering the two initial Hypothesis. Additionally, the effect of varying "impact" over the return from a strategy learner is studied.

The hypothesis validation and experimentation is done with 'JPM' stocks through 2008-01-01 to 2009-12-31 for making the strategies, which are tested upon 20010-01-01 till 2011-12-31. With a Benchmark of 1000 stocks holding for the respective period. Benchmark represents 1000 shareholding of JPM with a $100,000 initial cash. Holding is constrained to 1000,0 and -1000 shares, along with additional viability for +/-2000 shares trade at a time

2 INDICATOR OVERVIEW

This section briefly describes the Five Indicators used to create the Strategies.

1 **Bollinger Bands** : Finds over and under sold stocks, as Buy and Sell indicators, identified as below and above 2 x rolling standard deviations, **2σ**, from the rolling mean μ respectively, with a window **n**. BB% as described below can be used to identify for these signals. Optimal **n of 20 days** was found with BB**% > 0.866** and BB**% < 0.134 as Sell and Buy** thresholds, explained the next section.

$$\mu_t = \frac{1}{n}\sum_{k=0}^{n} \textbf{Adjusted close}_{t-k} \quad , \quad \sigma_t = \left[\frac{\sum_{k=0}^{n}(\textbf{Adjusted close}_{t-k} - \mu_t)^2}{n}\right]^{\frac{1}{2}}$$

$$Upper\ Bound = \mu + 2\sigma$$

$$Lower\ Bound = \mu - 2\sigma$$

$$\textbf{BB\%} = \frac{\textbf{Price} - \textbf{Lower Bound}}{\textbf{Upper Bound} - \textbf{Lower Bound}}$$

2 **Simple Moving Average**: depicts a non-myopic view of the trend and possible corrections of the prices. High price to SMA ratio, could indicate a Sell signal and Low for Buy signal. Explained as Below for a rolling window **n**, at time **t**. Optimal **n of 40 days,** signal **Sell** when **Price/SMA > 1.005** and **Buy Price/SMA< 0.994** .

$$SMA_t^{\{n\}} = \frac{1}{n}\sum_{i=0}^{n-1}(Adjusted\ close\ Price_{t-i})$$

$$\frac{Price}{SMA} = \frac{Adjusted\ close\ Price_t}{SMA_t}$$

3 **Stochastic Oscillator:** depicts momentum in a choppy market, comparing current price with historical prices, indicates **Sell** when STC (%D) is high and **Buy** when low. With an Optimal **rolling period n of 180 days and 44 averaging days -d**, these are reflected as **%D > 0.714** and **%D<0.284** respectively.

$$\%K = \frac{Adjusted\ Close_t - LLow_{t:t-n}}{HHigh_{t:t-n} - LLow_{t:t-n}} \times 100$$

$$\%D = \frac{1}{d}\sum_{i=0}^{d} \%K_{t-d}$$

4 **Pivot Point P**: further elevating the understanding of momentum with associated risk, over high, low and close prices. Generating trading signal like %BB. With optimal **n of 40 days,** signal **Sell** when **P > 1.107** and **Buy P/SMA< 0.839**.

Calculated as below, t time stamp, LLow as lowest low price, HHigh and Avg Close as highest high and Avg_Close price during the last n days.

$$P_{t-1} = \frac{HHigh_{t-1:t-n} + LLow_{t-1:t-n} + Avg\_Close_{t-1:t-n}}{3}$$

5 **Chaikin Money Flow**: explains the movement of volume along with price. As a direction metric, a high CMF value indicate Sell Signal and low as Buy signal. Calculated as below. With **Optimized n of 10 days and normalization of 3 days**, generated with CMF **> 3.75 and < -3.75** respectively.

1. Money Flow Multiplier: Identify the general position of close with respect to high and low prices of the day.

$$MFM = \frac{[(Adj\ close - Adj\ low) - (Adj\ high - Adj\ close)]}{Adj\ high - Adj\ low}$$

2. Money Flow Volume : calculate buying and selling pressure per day.

$$MFV = MFM \times volume$$

3. CMF : Aggregation money flow volume of the period

$$CMF_t = \frac{[\sum_{k=0}^{n} MFV_{t-k}\ ]}{[\sum_{k=0}^{n} Volume_{t-k}]}$$

2.1 Optimization procedure

The optimization of the indicators was performed based on search methods Along with contextual constraints. For manual Strategy two parameters for the indicators were optimized, 1. Lookback window, and trading signal thresholds.

Look Back Window n: for each indicator a range of lookback windows was created with contextual constraints sampled randomly, creating sets of indicators. Buy and sell signals were created using Manual strategy over the Training period of JPM, creating portfolio values to calculate sharp ratio of each set. The set with highest sharp ratio was further used for testing. For BB%, a smaller range ,10-25 days, created to explain short term volatility. SMA with 30-65 days, to explain medium term trend. STC with 10-180 days window for lower choppy momentum and $1/5^{th}$ to $1/6^{th}$ of n as normalizing period. Pivot point P and CMF with 10-20 and 60-90 days, to explore price-volume movement and momentum risk over short to medium term.

Trading thresholds: Like Lookback windows, thresholds were picked randomly over a range of max and min indicator bounds over the training period.

For the Strategy Learner, these indicators were discretized into bins, as described in Sector 4

3 MANUAL STRATEGY

Combining Indicators – Trading Strategy: Each Indicator creates a Buy, no action and Sell signals as 1,0 and -1. Individually they may create false signals, resulting in losses. A combination signal could overcome these as defined below:

Sell: if Average of trade signal for all 5 indicator is less than or equal to -0.6.
Buy: if Average of trade signal for all 5 indicator is more than or equal to 0.6.

Reason: minimize loss generating trades, and decrease transactions count, where 2 or more indicators contradict, or there is a low confidence, created by 2 similar indicators, resolved with a 3^{rd} non-similar indicator.

Strategy steps: **1**. For the first trade signal: if signal **s** is 1, buy 1000 shares, if **s** is -1, sell 1000 shares. **2** For subsequent signals, if current position is same as **s**, or **s is 0** then hold the position, else if **s** is opposite to current position, then transact 2000 shares to Sell or buy when s is -1 or 1 respectively. **3** on the last day, exit the position to realize the second last trade.

3.1 Comparison of Manual Strategy with Benchmark.

In-sample Comparison fig 1.0, The Manual strategy give 92% Cumulative return, compared to 1% of benchmark, with a much lower risk.

Observation: While the benchmark in in-sample period majorly show a negative trend, a reversal is observed in 2009-05, with trend smoothening, showing lower gains, though above the benchmark.

	Data	port_value	cumilative Return	Sharp Ratio	daily Return std	Avg daily Return
0	insample	Benchmark	0.012300	0.162	0.017021	0.000174
1	insample	Manual_stratergy	0.928417	2.106	0.010284	0.001364
2	outsample	Benchmark	-0.083400	-0.287	0.008482	-0.000153
3	outsample	Manual_stratergy	0.167960	0.804	0.006531	0.000331

Table 1 — Manual Strategy Vs Benchmark performance

Out-sample Comparison As shown in fig 2.0, The manual strategy is still able to gain a cumulative return of 16.8% compared to the loss of 8.3% in benchmark.

Comparison of "In" with "Out"sample : The trading signal count in the out sample is very low compared to insample signals in 1^{st} 17 months, though similar to its last 6 month, generating low returns.

Reason: the rolling windows and trade thresholds of each indicator were optimized to identify trading opportunities during a strong trend market, unlike the choppy – low difference out-sample market. As the stock price behavior changed, the previously identified window and thresholds become less relevant.

These results support Hypothesis 1, TA factors outperformed benchmark.

Solution: Re-evaluate the indicators on rolling bases, when the avg return over a period show high deviation from previous period.

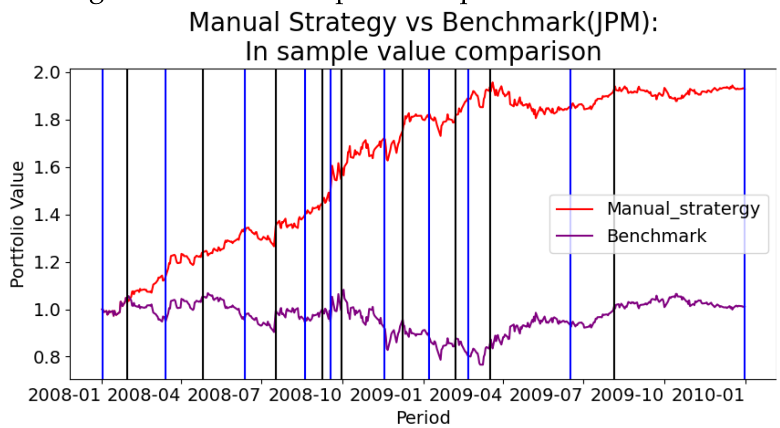

Figure 1 — Manual Strategy vs Benchmark performance – In sample. Sell Signal – Black, Buy Signal -Blue bar.

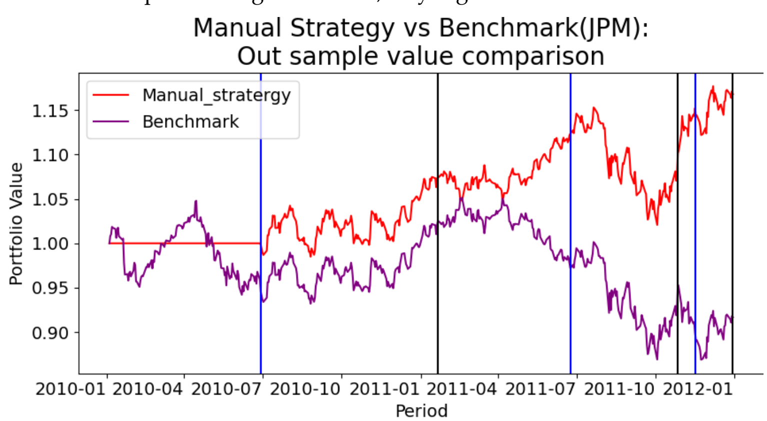

Figure 2 — Manual Strategy vs Benchmark performance – Out sample. Sell Signal – Black, Buy Signal -Blue bar.

4 STRATEGY LEARNER :

While the Manual Strategy outperformed the Benchmark, it is created on individual indicators, therefore it might have missed some trading opportunities based on their interplay between for different market conditions. With increased complexity, Automated computational models such as Q-Learning or Random Forest, etc could identify their relationship with market behavior. This report explores Q-learning, a temporal deviation, off- policy, model-free, bootstrapped reinforcement learning methods, i.e An agent without initial knowledge of the state transition probability of the market and with no information on the daily returns for its actions. It updates the utility/long-cumulative-rewards of experienced states based on the subsequently observations and rewards, which may not follow the current policy, where a actions is picked which maximizes the reward of a previously visited state.

4.1 Converting Trading problem to learning problem

The following definitions convert a Trading problem to a learning problem.

States : The 5 Indicators at time t together define the market's state. As these are continuous, Individual indicators are discretized into bins, as continuous values being infinit, which may not be visited ever again, creating sparse rewards distributions. Market states for each indicators can be discretized in 2 ways :

1. Percentile binning: indicator values are binned into equal division of instances bucket. BB, Price/SMA and CMF are binned on 5 equal percentiles buckets each, creating 125 states, these bins were identified using the optimization procedure as describes in the previous section, constrained to maximum 200 states from these 3 indicators. Any out of bound values, are mapped to the extreme bins.

2. Contextual binning: bins created based on contextual knowledge of the indicator's distribution, it reduces space complexity and make more reliable models. For STC [min(inbound-STC),0.7, max(inbound -STC)] and for Pivot point [min(inbound-P), 0.7,1.2, max(inbound -P)] ads upto 750 states.

Encoding of states, each state is described as bin number for the individual indicator i.e 1-BB, 5-SMA,1-STC,0-PP,3-CMF , when concatenated : 15103. These is very big number to create an array index, while a 5d array can also be created with increased query time, therefore each possible state is mapped to a integer number in 0-750 range and is used to create an array of shape 750 x 3(i.e #actions)

Actions: at each state there can be 3 actions: Do nothing, Buy and Sell, as 0,1,2.

Reward R: daily return for the position in holding, as a result of a previous action is given as a reward for time t, as shown below. Where **th** is the current total holding – based on previous action, **dr** is daily return of the day, and **impact** is percentage slippage from decision to action in the daily return

$$\mathbf{R_t \ = \ th_t * dr_t * (1 - impact)}$$

4.2 Steps of the model: for n iterations:

1. Set initial state as s_t, set th_t = 0, r_t=0
2. Identify ∈- greedy action a_t, for the s_t and identify next state s_{t+1}
    a. If random value > ∈, take greedy action else take a random action
3. Calculate r_t from daily return and current th_t
4. Update Q value for s_t based on s_{t+1} and r_t using the following equation
    a. $\mathbf{Q(s_t, a_t)_{t+1} = (1 - \alpha) * Q(s_t, a_t)_t + \alpha * (r_t + \lambda\, max_{a_t}(Q(s_{t+1})_t))}$
5. Identify action a_{t+1} based on s_{t+1} , $\mathbf{argmax_{a_{t+1}}(Q(s_{t+1})_{t+1})}$
6. Update th based on action, that is
    a. If $a_{t+1} = 1$ and th=0 , then th +=1000 , if th<0, then th +=2000
    b. Else if $a_{t+1} = 2$ and th=0 , then th -=1000 , if th>0, then th -=2000
    c. Else hold the position
7. Decay ∈ : ∈= ∈*decay_factor
8. Go to step 3 till ench of epoch.

For Prediction, encode the state, query the Q-learning agent, and take the Greedy action at that state. Calculate th as above, and create array of **th**, which will define the entry and exit position. For calculating Cumulative return.

4.3 Hyper Parameters: 3 Out of 5 Hyper parameters were tuned for this model

Learning rate $\boldsymbol{\alpha}$: defines the weight of change in Q value of the previously visited state based on the new observations state and is used for averaging the Q value for a state action pair over the period of time. A high α gives more value to new state observed and make substantial changes, unlike small α. By **optimizing** α over 1st 16 months of the insample and cross validated over its last 6 months, **$\boldsymbol{\alpha}$ of 0.5** was chosen, giving equal importance to new and old experiences

Discount Factor $\boldsymbol{\gamma}$: quantifies the importance of future reward, ranging from 0-1. 1 corresponds to hypermetropic rewords and 0 as 1 step myopic rewards. On optimization, $\boldsymbol{\gamma\ of\ 0.8}$ was selected emphasizing on long term returns of the actions. Optimized as described above.

Epsilon Decay rate 'rard' : is used to balance exploration and exploitation. High values, result in slower decay and the model gets more opportunities to explore, before exploiting. A rard of **0.99** was identified as optimal.

Exploration-Exploitation rate ϵ : used to choose the action randomly or greedily at any time, by validating if a random float between 0-1 is more than for it take greedy action and less than for random action. It was set at 0.9

Epochs: the number repeatedly presented experience. Adjusting learning for slow learning rate and epsilon decay rate, the data was presented 1000 times.

5 EXPERIMENT 1 1.5 PG

Setup: This section explores the 2nd Hypothesis, as per the Introduction. The experiment compares Manual Strategy and Strategy learners over in and out sample period for JPM.

5.1 Assumptions, Initial Hypothesis and Parameters.

Assumptions: Impact and commission is fixed for the complete period. Python and numpy random module are uniformly random. Benchmark and holding constrain definition as described above is valid. Only JPM is evaluated with str value of $100000, with only 3 positions, short, long, cash As Q-learner is a model-free and $\in$-greedy model, its exploitation-outcome is dependent on random actions taken for exploration, and therefore without a fixed seed, the model **may give relatively different** strategy, this can be minimized with more experience.

Hypothesis: With right TA indicators and parameter optimization we can identify a profitable strategy manually. Learning Engine can suggest better trading strategy than manually created. **Parameters**: All the 5 parameters were used as described in the previous sections, along with impact of 0.005 and commission of $9.95 for each transaction.

Outcome -: Fig 3. Depicts the Strategy Learner transaction yielded >3x cumulative return over insample, higher than Benchmark and Manual Strategy. And outperform it the Benchmark in outsample and but not Manual overall, overfit on insample data, along with the state space representation, with about outsample 40% states not in insample. Hence, we failed to prove Hypothesis 2.

Solution 1. longer training period 2. Rolling period of training. 3. Reduced states

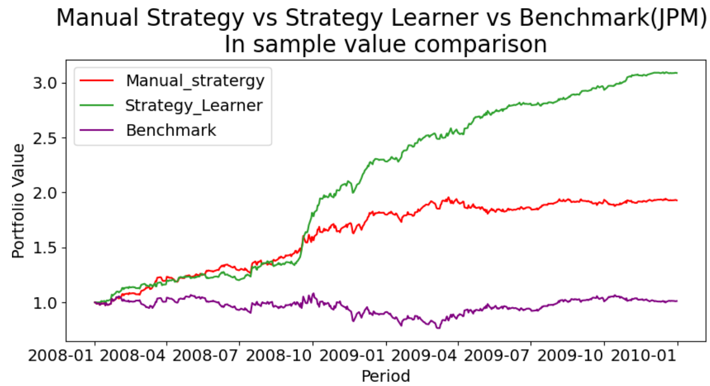

Figure 3— Manual Strategy vs Strategy learner - Insample

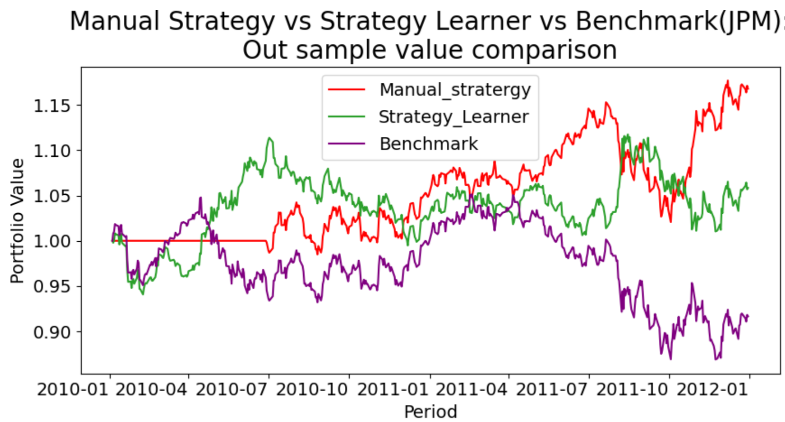

Figure 4— Manual Strategy-v-Strategy Learner outsample

6 EXPERIEMENT 2 – 1.5 PG

Hypothesis: Changing impact effects trading result, and cumulative return.

Setup: Train the strategy learner over insample JPM, with 3 different impact values [0,0.005,0.03] and predict the actions respectively for in sample trading.

Assumption and Parameters are same as Experiment 1, except impact values as stated above and commission is fixed to $0.

Outcome: Fig 5, Table 2. With increase in impact the cumulative return, of SL reduced drastically, from 2.63 at 0% to -0.56, as it reached 3%, a drop of 309%. where 0.5% impact resulted in 207% returns. As the cost of transaction increases, the number of transactions may reduce and therefore the opportunities may reduce as well, for both recovery and gaining profit. Which may result in reduced ability to get out of a bad position, leading into further losses.

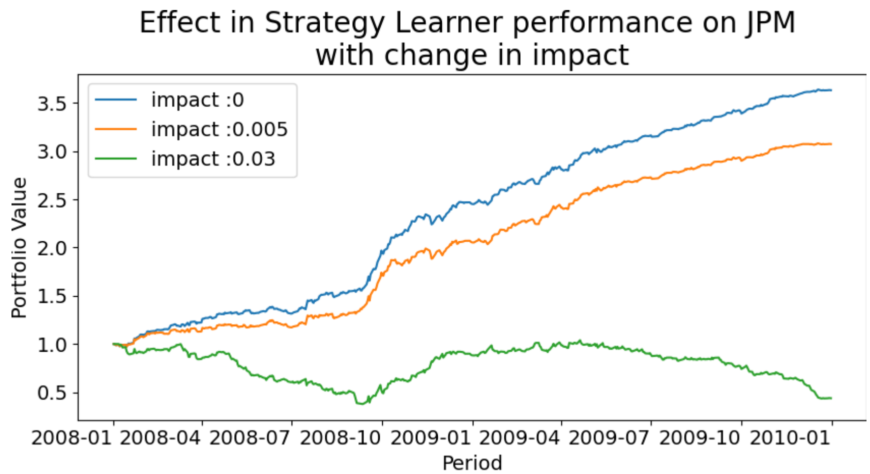

Figure 5— Comparison of the strategy Learner portfolio value at different impact values.

In addition to the cumulative return the sharp ratio is observed to be reduced from 4.928 to -0.881, from 0% impact to 3%, via 3.910 at 0.005, which stands at 80% of 0 impact. With just 0.5% impact, the return to risk reduces by 20%. Which indicates as the number of transactions reduce, the model's aims for high change in value per action. This is visible at impact =0.03 where the daily return deviates ~3 times more than the lower values of impact.

	impact	cumilative Return	Sharp Ratio	daily Return std	Avg daily Return
0	0.0	2.631500	4.928	0.008400	0.002608
1	0.005	2.073154	3.910	0.009266	0.002282
2	0.03	-0.561911	-0.881	0.024234	-0.001346

Table 2 —Strategy learner performance metrics at different impact values.

SUMMARY

The project explored technical indicator-based strategies, manually as well as with a Strategy learner, including the process of optimization, steps in manual strategy along with Q-learner Strategy learner, discussing the procedure to convert the trading problem to a learning problem. Empirically, we found that both performed better than the Benchmark, over in and out sample. Additionally, change in long term market trend, was found to highly impact the model accuracy, and so rolling training model or short-term trend TA indicators may yield better results. Lastly the effect of 'impact' over the Strategy learner behavior was explored, depicting its increases results in reduction of trades and therefore loss of opportunities along with value.