Machine Learning for Trading

Project 8: Strategy Learner

Tharun Saranga (tsaranga3)

Framing the Trading Problem:

The goal of this project is to create a learner that would trade on a given stock to maximize the profits. The learner after training on the provided data, it gives the list of trades to trade on the test data. The learner has to learn from the stock data by adding additional information such as technical indicator values to augment its information about that particular stock. Bag Learning Technique with Random Forests has been used in this project. This trading problem is converted to learning problem by providing the learner the data about the stock and labels showing the trend of the future price. The learner learns to predict the future trend whether the price increase or decreases. When the learner is tested on new data, it predicts the future trend and gives us a set of trades to maximize the portfolio.

The learner is a Bag Learner for Random Forest with leaf-size = 5.

During the training phase in addEvidence() function the prices are augmented with the indicators and a label with values that tell the future trend. +1 if the trend is positive and 0 if the trend is negative. This is used to train the Bag Learner with label as the value to be predicted.

During the testing phase in testPolicy() function the data augmented with the indicators is passed as test data to the query method of the Bag Learner which provides the predicted trend of the stock. Based on this predicted trend, a set of trades are created with +1000 for buy if positive trend is predicted and -1000 for sell if negative trend is predicted and 0 for hold.

The same indicators are used for this project that are used in the Manual Strategy Project.

1. Simple Moving Average (SMA)
2. Bollinger Bands
3. Moving Average Convergence Divergence (MACD)
4. True Strength Indicator (TSI)

The learning method used for this project does not require any discretization. Random Forest Learner can learn continuous values and for any range. Hence, no data alterations are required for this method of learning.

Implementation of Strategy Learner for Trading Problem:

BagLearner Parameters:
Leaf_size = 5
Bags = 100

Implementation of addEvidence:

- The prices dataFrame is added with additional columns of indicators.
- All the four indicators used for the Manual Strategy are used here.
- The prediction label if the price is increasing or not is created by peeking into the future.
- Additional column with shifted prices is created.
- If the future price is greater than current price plus impact, then a True label is Assigned for that price else it is assigned False
- The future price column is dropped.
- The X training values is the dataFrame without label column.
- The y training values is the dataFrame with only label column.
- These X, y are used to train the BagLearner with the above parameters.

Implementation of testPolicy:

- The prices from the given time range is extracted into a dataFrame.
- To this dataFrame the indicators are added as columns same as during the training phase.
- The text X values are created from this dataFrame and passed to predict function of the BagLearner.
- The predicted y values are the labels that inform if the price increases in the future or not.
- These predicted labels are used to create a trade dataFrame the consists of the optimal trade actions.
    - If we have no stocks on hand and the label is True, then we buy the stock (+1000).
    - If we are in long position and label is True, then we Hold the stock (0).
    - If we are in long position and label is False, then we sell the stocks (-1000).
    - If we have no stocks on hand and the label is False, then we buy the stock (-1000).
    - If we are in short position and label is False, then we Hold the stock (0).
    - If we are in short position and label is True, then we sell the stocks (+1000).
- Using the above conditions, a trading dataFrame is created that has the predicted optimal trades to take with +1000 and -1000 for buy and sell respectively.

Experiment 1:

The goal of this experiment is to compare the performance of the Strategy Learner and Manual Strategy. Manual Strategy and Strategy Learner both use the same Indicators, starting cash and impact value. Manual Strategy gives trades according to pre-defined conditions while Strategy Learner after training on data learns to predict and gives optimal trades for test data. Both are tested on same time-period data from JPM stock. The following are the parameters used for Manual Strategy and Strategy Learner.

Time Period: 2008-01-01 to 2009-12-31
Stock: JPM
Indicators: SMA, Bolliger Bands, MACD, TSI
Starting Cash: $100,000
Impact: 0.0
Commission: 0.0
Leaf-Size for Bag Learner: 5
Bags: 100

Based on the trades obtained from the Manual Trader and Strategy Learner the portfolio values calculated using market simulator are plotted to compare between them. We assume that the Strategy Learner will outperform the Manual Trader.

Results:

COMPARISION OF PORTFOLIO	BENCHMARK	MANUAL STRATEGY	STRATEGY LEARNER
CUMULATIVE RETURNS	0.012299999999999978	0.16999999999999993	2.9913
STANDARD DEVIATION OF DAILY RETURN	0.017004366271213767	0.010629233283209359	0.005747905599259141
AVERAGE DAILY RETURNS	0.00016808697819094035	0.00036753012478717655	0.0027663790300554185
SHARPE RATIO	0.15691840642403027	0.5488975264363396	7.640157778458395

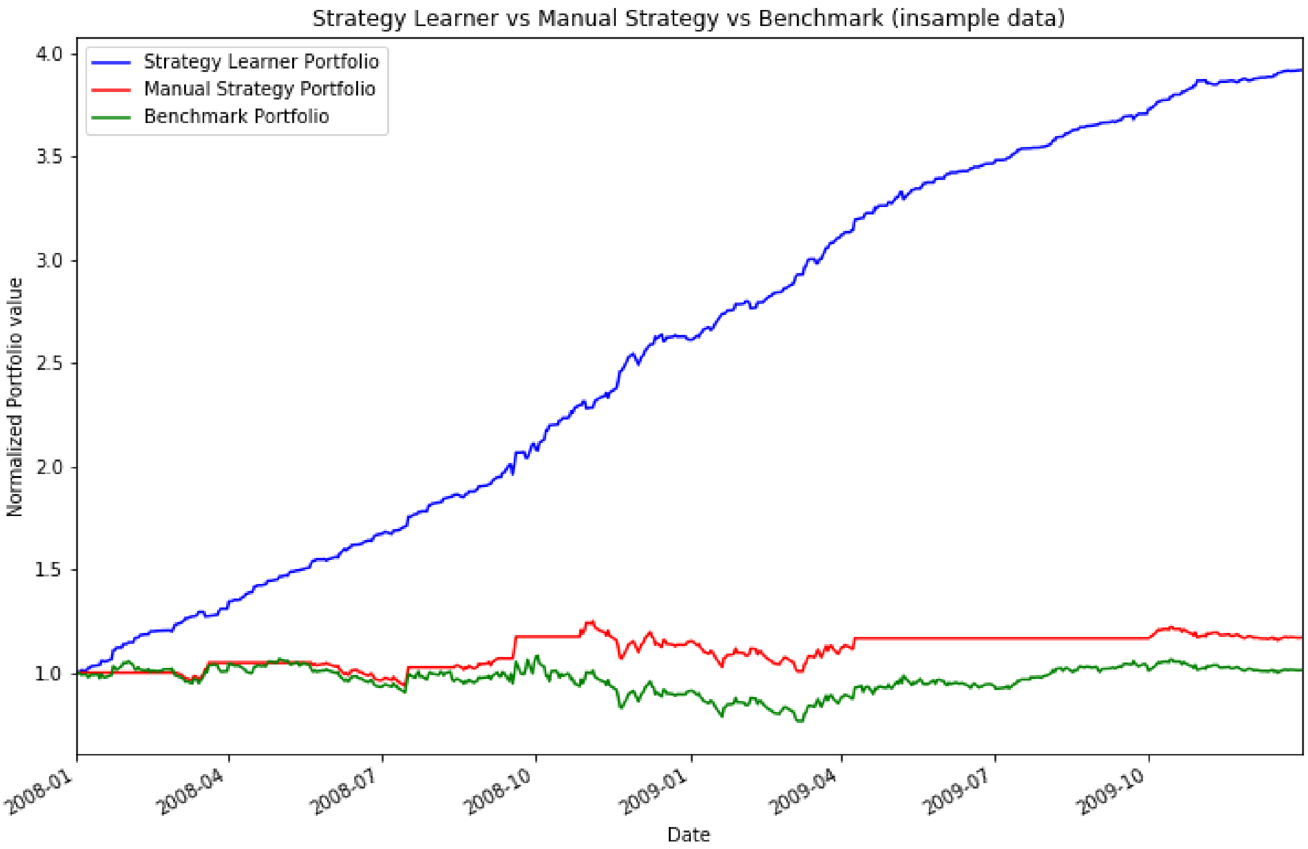

Figure 1: Comparison between Strategy Learner vs Manual Trader vs Benchmark

The Strategy Learner performed better the Manual Trader and the Benchmark by a large margin. This is because the Learner is based on Random Forests which give the best prediction after learning from the data while the Manual Trader only applies a small set of stationary conditions which might not be optimal but give some profit. The Strategy Learner / Random Forest on the other hand creates thousands of conditions after learning from the data. This makes the Strategy Learner perform better than Benchmark and Manual Strategy.

This result is expected to be same every time with in-sample data. Since the learner already learned the predictions from the in-sample data, it will always perform better with in-sample data. The exact values might differ if we retrain the Strategy Learner as training the Random Forest has inherent randomness associated with it. Apart from that the same relative result is expected from the Strategy Learner for the in-sample data.

Experiment 2:

The goal of this experiment is to investigate how the value of "Impact" affects the Strategy Learner's performance. The hypothesis is that as the value of impact increases, the performance of the learner decreases. This can be observed by calculating the portfolio values with different impact values.

The experiment is conducted on the JPM stock with the Strategy Learner using the following parameters.

Time Period: 2008-01-01 to 2009-12-31
Stock: JPM
Indicators: SMA, Bolliger Bands, MACD, TSI
Starting Cash: $100,000
Impact: 0.001, 0.005, 0.01
Commission: 0.0
Leaf-Size for Bag Learner: 5
Bags: 100

Results:

Effect of Impact	**Impact = 0.001**	**Impact = 0.005**	**Impact = 0.01**
Cumulative Return	2.8299373769088927	2.105309061980719	1.466912211276783
Standard Deviation of Daily Returns	0.0059209008835383525	0.0073162022980021064	0.008683111024571585
Average Daily Returns	0.0026852825686780506	0.002277345362685785	0.001830741531856357
Sharpe Ratio	7.1995022545418275	4.941325485806884	3.3469710068817204

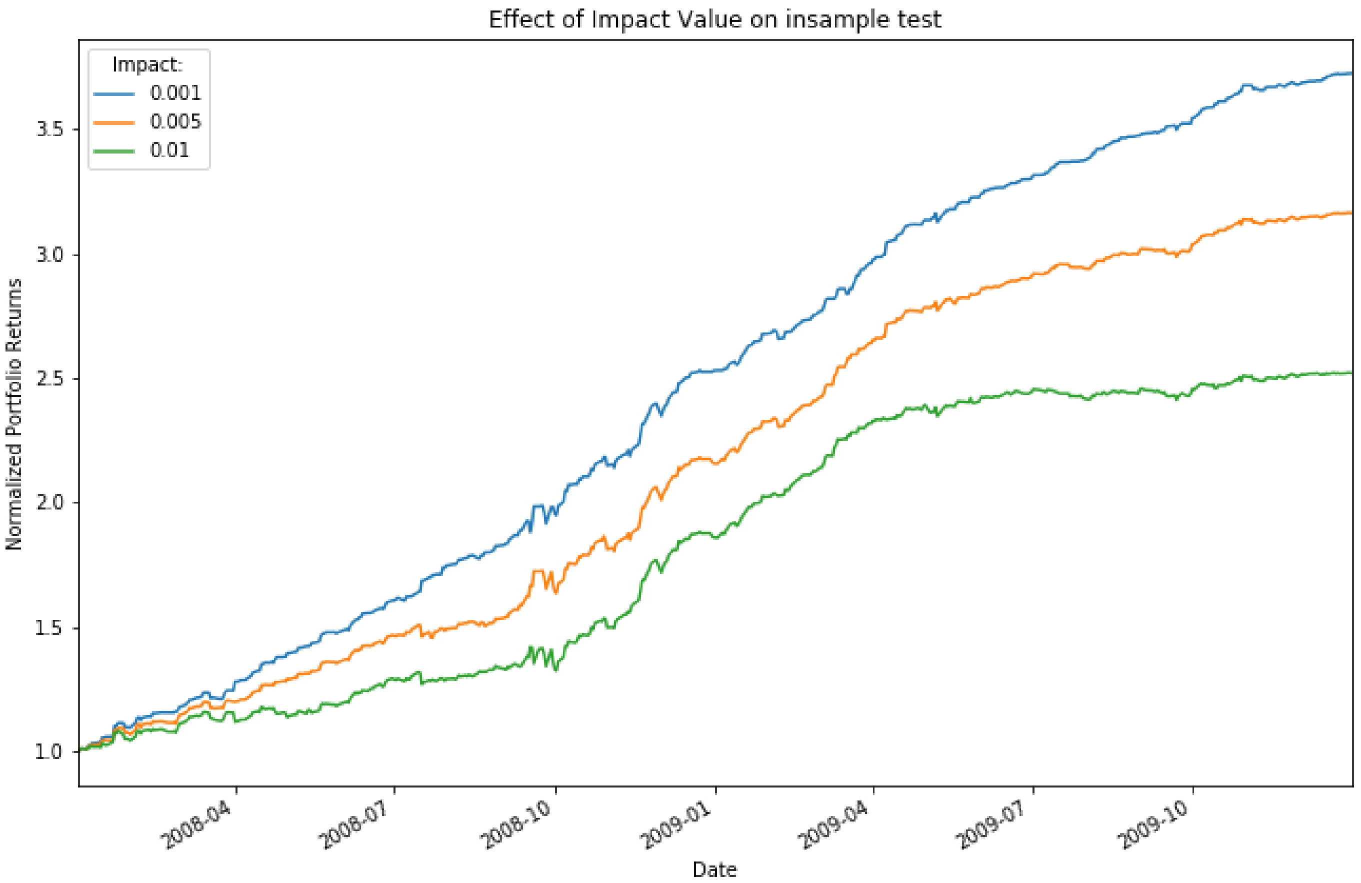

Figure 2: Strategy Learner Portfolios for different impact values

From the table we can observe that Sharpe Ratio and Cumulative Returns decrease as the impact value increases. It is also evident from the graph that higher impact values results in lower cumulative return.