Strategy Learner

Georgia Institute of Technology
CS7646 Machine Learning for Trading
Spring 17
James Chan

Overview

Abstract

The rise of machine learning in A.I. has proven to be an indispensable tool for Wall Street professionals attributing to its remarkable performance in forecasting, optimization, and decision-making. In CS7646 we aim to apply both supervised learning as well as reinforcement learning towards stock trading.

Technology Stack

Python, Numpy, Pandas, Matplotlib.

Table of Content

1. Technical Indicators
2. Manual Trader
3. Random Forest Trader
4. Q-learning Trader

Technical Indicators

A manual strategy based on technical indicators is first devised to establish a baseline comparison for the machine learning based strategy.

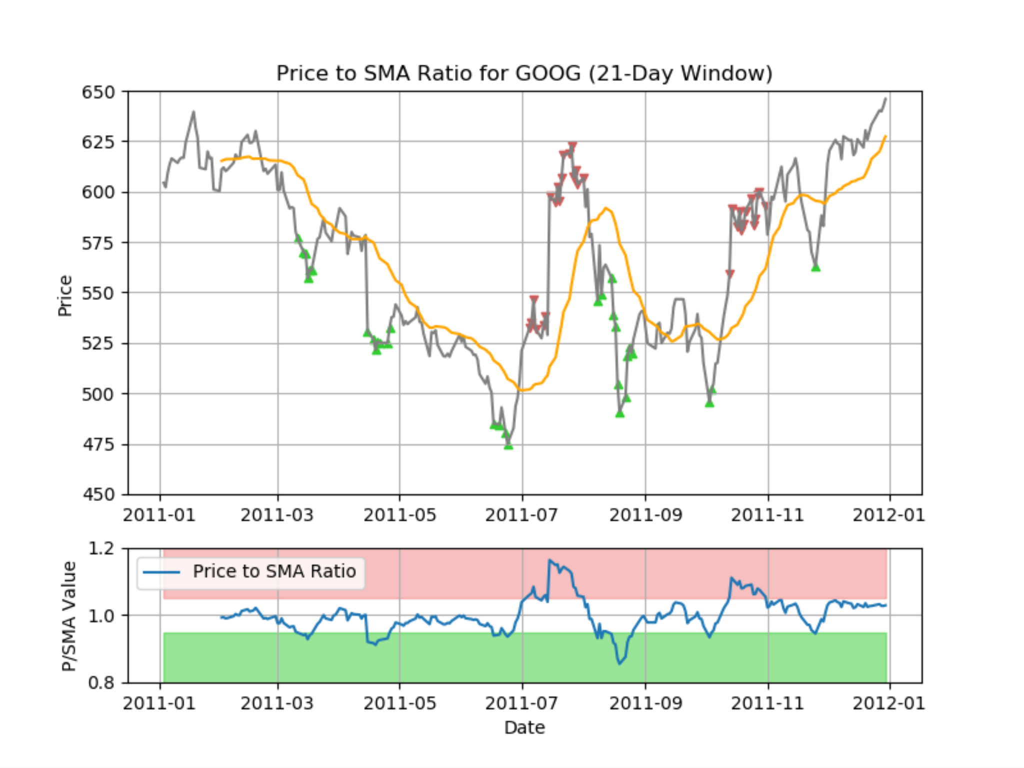
Indicators Used:

- Price/SMA Ratio (21-day window)
- Bollinger Bands® (21-day window)
- Volatility (21-day window)

Price/SMA Ratio

Price/SMA Ratio is defined as the price of the stock divided by its simple moving average. When Price/SMA Ratio goes over a pre-defined upper threshold (e.g., 1.05), it represents a sell opportunity. When the ratio goes below the pre-defined lower threshold (e.g., 0.95), it represents a buy opportunity.

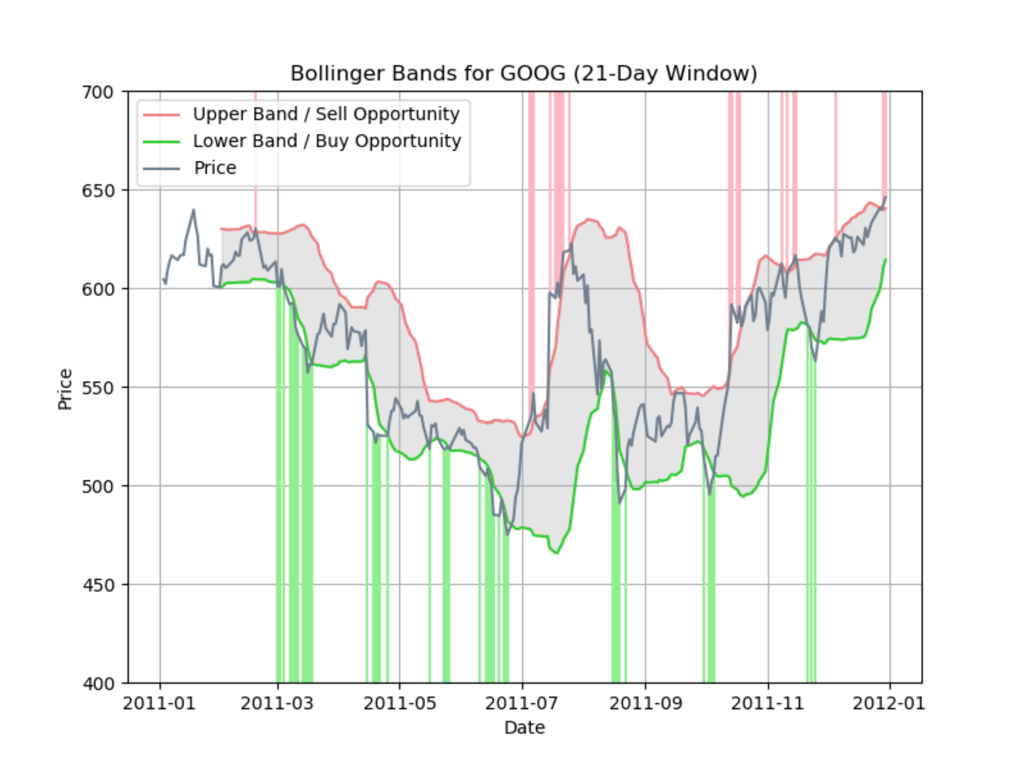
$$Price/SMA\ Ratio_t = \frac{Price_t \times n}{\sum_{i=0}^{n} Price_{t-i}}$$



Bollinger Bands®

Bollinger Bands are defined as the upper and lower boundaries established by adding twice the rolling standard deviation to either side of a stock’s simple moving average. When the price of a stock surpasses the upper band, it is considered a sell opportunity. When it drops below the lower band, it is considered a buy opportunity.

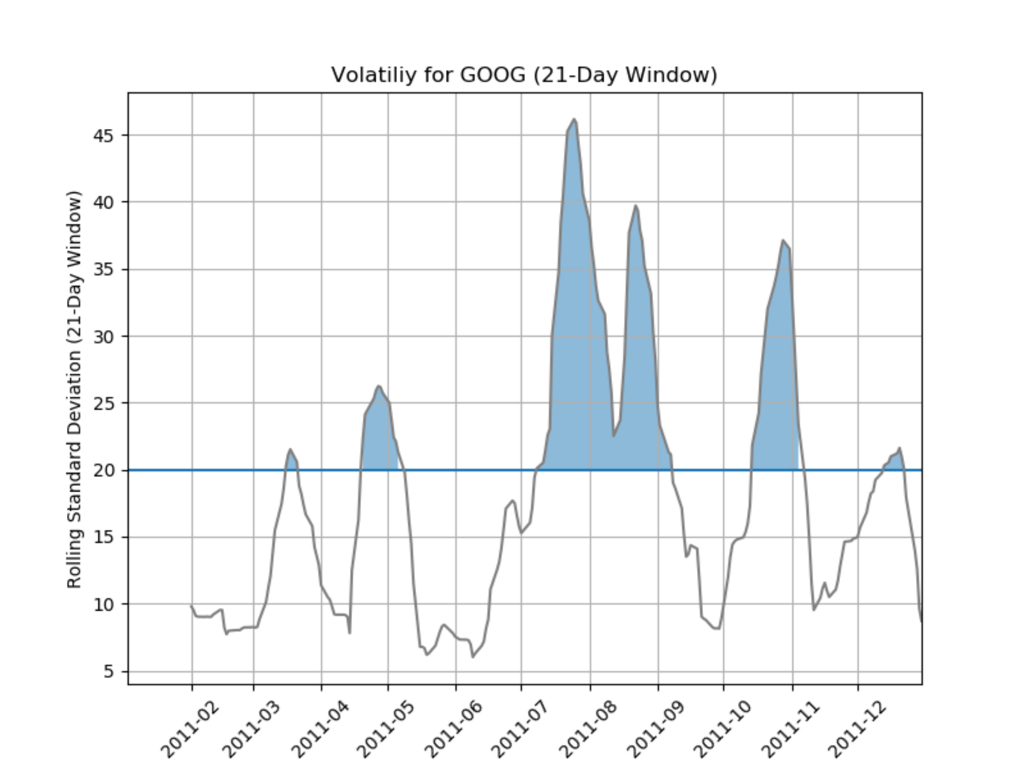
$$Bollinger\ Bands_t = \frac{\sum_{i=0}^{n} Price_{t-i}}{n} \pm \sqrt{\frac{\sum_{i=0}^{n}(Price_{t-i} - \overline{Price})^2}{n}}$$



Volatility

Volatility is defined by the rolling standard deviation of the price of a stock. If the volatility is high, it signals the market is experiencing large swing, which often translates to profit opportunity for traders.

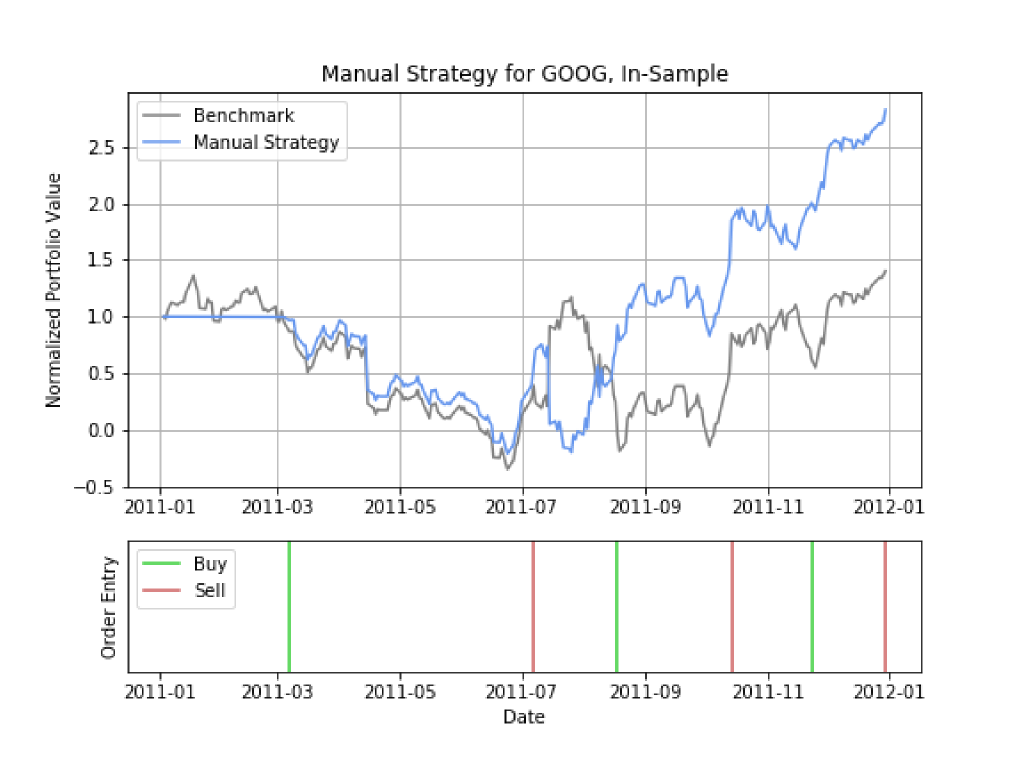
$$Volatility_t = \sqrt{\frac{\sum_{i=0}^{n}(Price_{t-i} - \overline{Price})^2}{n}}$$



Manual Trader

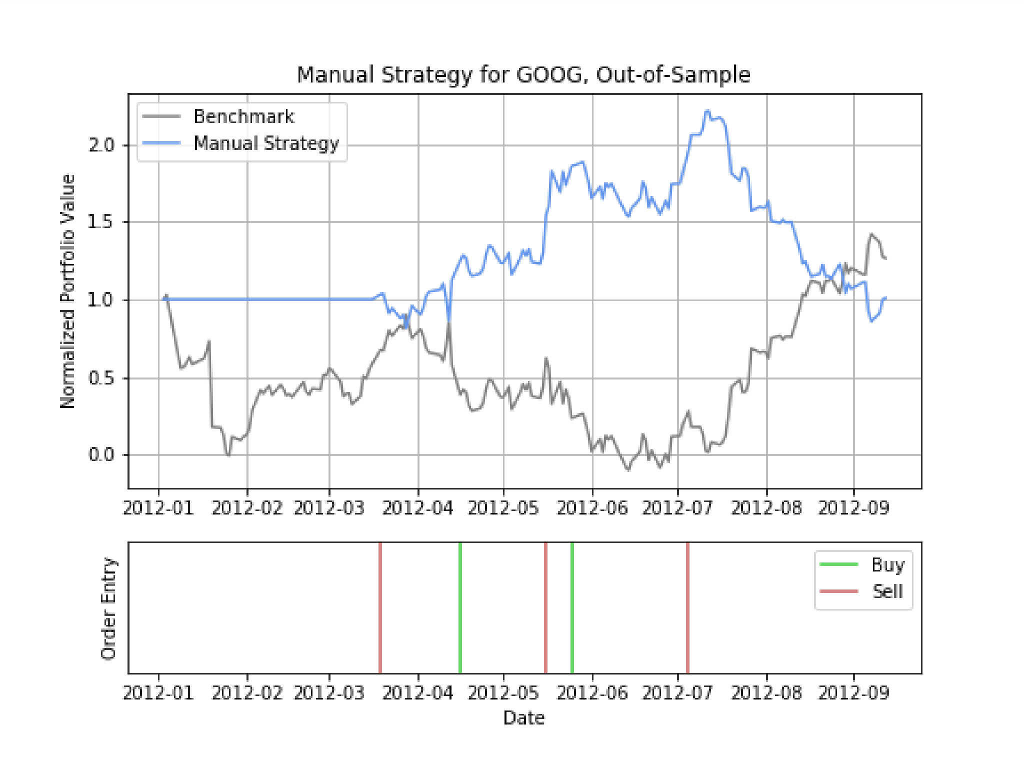
A manual strategy is first devised to establish a baseline comparison for the machine learning strategy. In manual strategy, price/sma ratio, Bollinger Bands, and volatility are used as the sole inputs for informing the trader. Their parameters are selected through manual tuning. Benchmark is defined as having $100,000 of available cash at the start of a portfolio and buying 1000 shares on the first trading day and holding it indefinitely.

As shown below, manual trader beats the benchmark slightly in-sample after some manual tuning. Our traders has a max holding restriction of ±1000 shares.



Manual Trader

Below shows the out-of-sample performance of manual trader. It has some room for improvement. Next, we will go ahead and apply two machine learning algorithms to it in an attempt to capture a closer relationship between these technical indicators and the price movement.

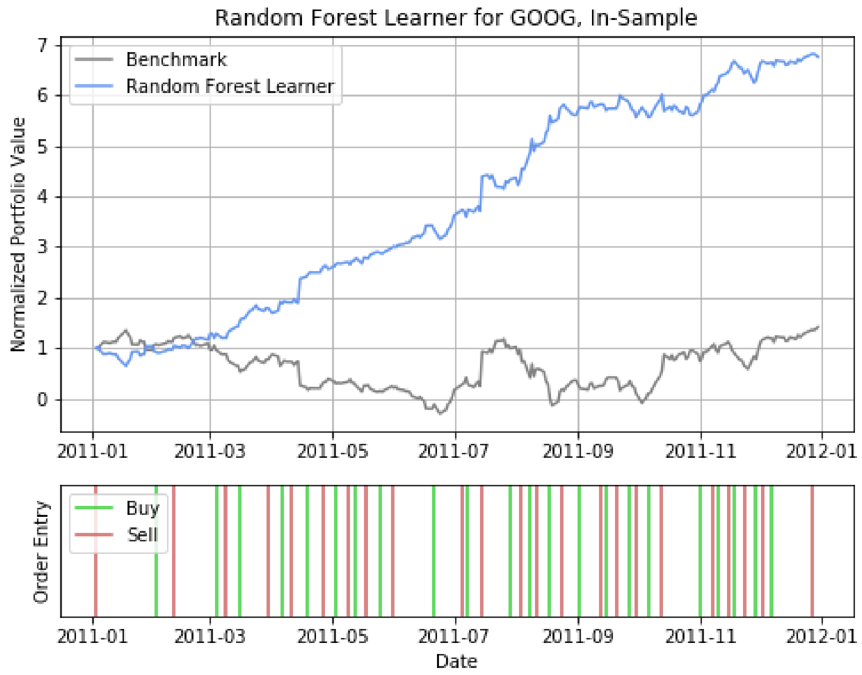


Random Forest Trader

Random Forest excels in noisy environment and is robust to outliers. This is advantageous to a market environment with large spikes in either the up or the down directions.

Here, we use N-day return to label our the target values of our samples. The input features are the same indicator values used in manual trader.

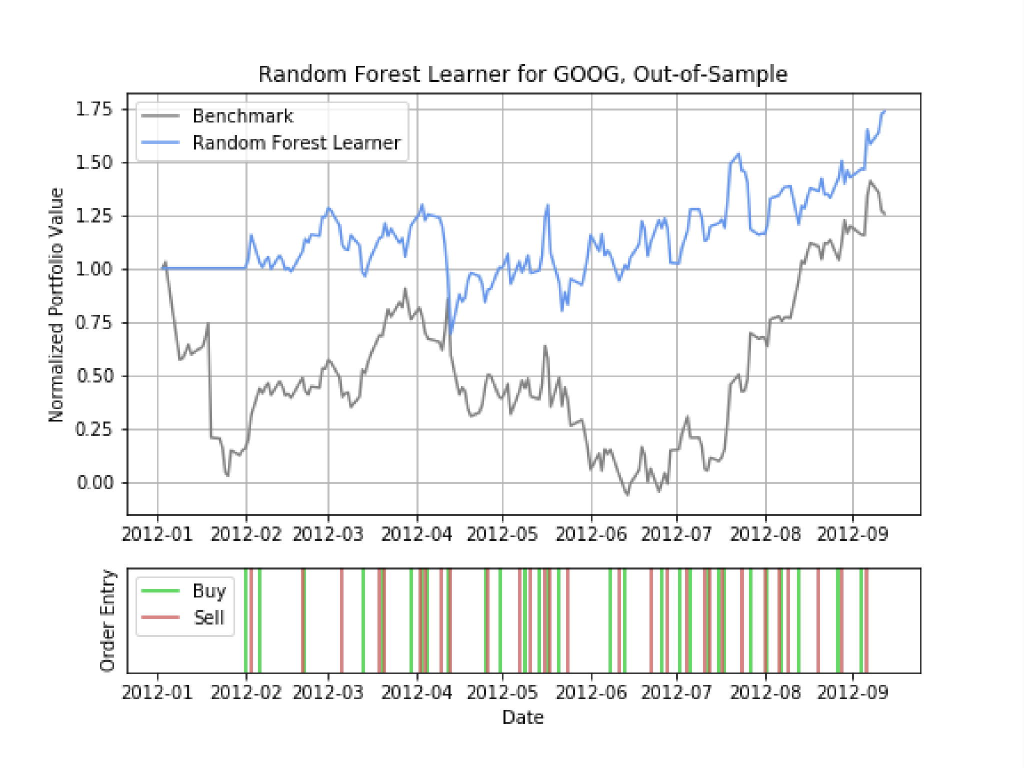
As evident below, the in-sample performance is quite remarkable, but of course, this is because in-sample peeks into the future.



Random Forest Trader

The out of sample performance fluctuates between runs. Below figure shows a run which outperforms the benchmark. The reasons for the fluctuation are primarily as follows:

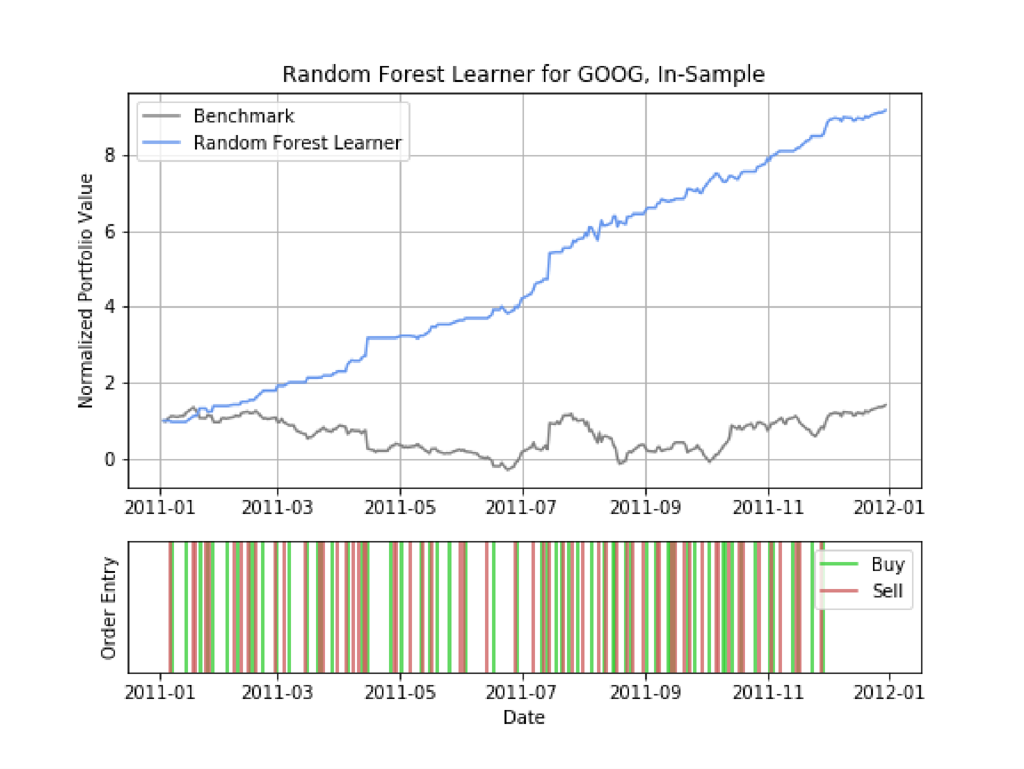
1) Random Forest is a randomized algorithm
2) Three technical indicators alone may lack sufficient predictive power.
3) Size of the training data is on the lower end for an ML problem.
4) N-day labeling assumes static property, which is generally weaker than one that factors in time-series property.



Q-Learner Trader

Reinforcement learning algorithms not only estimate the best possible long term reward but also recommend the optimal action at any given state. In our q-learner, we use the same three indicators(p/sma, Bollinger bands, and volatility) as our states. Actions are Buy, Sell, or Do Nothing. Rewards are the returns of the next day.

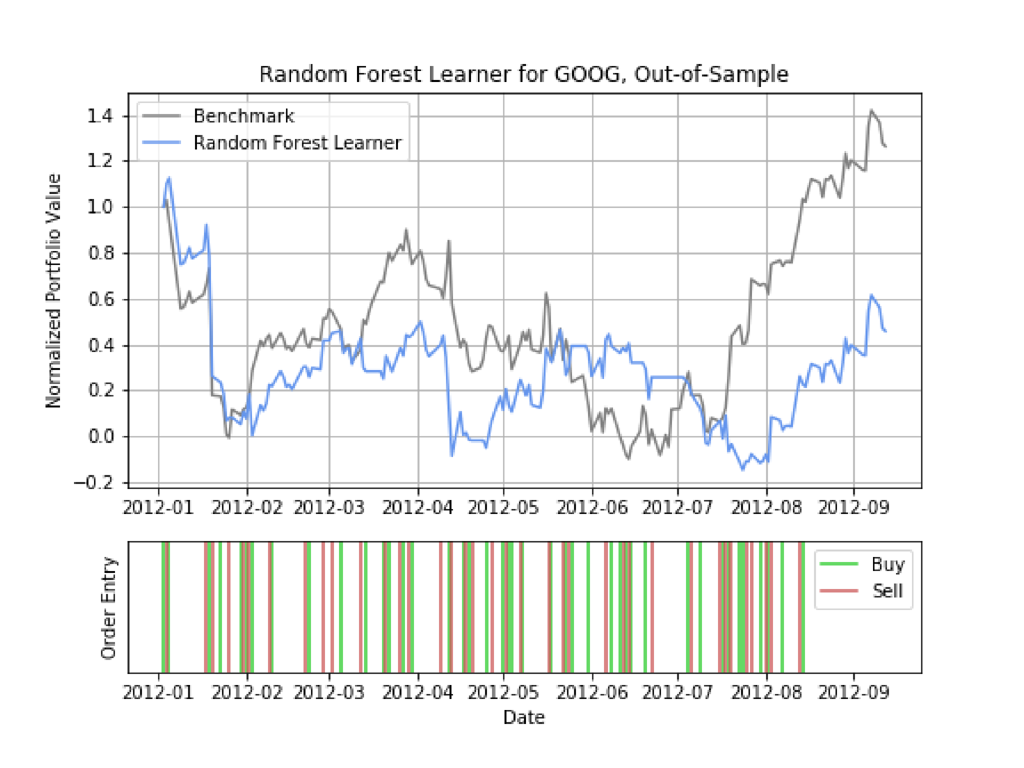
As shown in figure below, the in-sample performance is remarkable as we would expect because we are testing on training data.



Q-Learner Trader

The out of sample performance has room for improvement in this implementation of the Q-learner. Here are some potential reasons why it underperforms the benchmark.

1) 1-day return as reward is too “greedy” and causes overfittage
2) Three technical indicators alone may lack sufficient predictive power.
3) Size of the training data is on the lower end for an ML problem.



Conclusion

The stock market is an excellent domain for applied machine learning. In order to create a truly advantageous trading system, a strategy developer would also want to take the following aspects into consideration.

- Data integrity
- Trading latency
- Ensemble learning
- Survivorship bias
- Non-stationary property of time-series property.
- Event-driven factors.

Related Projects:

- Recurrent Network Trader
  https://github.com/jameschanx/RNN_Stock_Trader
- Sentiment Trader
  https://github.com/jameschanx/Sentiment_Stock_Trader-NLP
- Deep Learning Trader
  https://github.com/jameschanx/Deep_Learning_Stock_Trader

Proposed Projects:

- Vision-based learning (i.e., satellite imagery)