

A simple pullback strategy in Nifty Index spot and its backtesting

Vipul Tanwar

Introduction

The strategy I am presenting here tries to make profits by identifying the small retracements in prices against the ongoing direction of momentum. For example, if the market is bullish, we try to take a buy position when there is a short term downside movement and vice versa.

Hence our problem boils down to: How to identify these **short-term retracements** against **the momentum**?

As I've highlighted the two terms above, it is clear that we need to have two signals, which try to identify the short-term retracements and long-term momentum in the price-time series given. For this purpose I am using two simple and traditional signals called Simple Moving Averages (SMA) and Exponential Moving Averages (EMA)

As we know, that the EMA weighs the near past values more in than the far past values, it is a good indicator that can be used in order to summarize the movements of recent days. Whereas since the SMA equally weighs the rolling values provided, It gives a better picture of a long term as compared to the EMA

For a better contrast of short and long term, I have used the rolling window in EMA to be smaller than the SMA rolling window

Buy and Sell Logic

It is a long only strategy where we generate a **buy signal** when the following conditions are met:

- 1) EMA(15) 6 periods ago was more than SMA(20)
(Explains bullish momentum)
- 2) In the current window, the EMA(15) has crossed SMA(20) more than 0.7%
(Explains short term price decline)

We Generate a **sell signal** when the EMA(15) again goes above SMA(20) which explains continuation of momentum after retracement

The numbers, which can be tweaked around here, are:

- 1) Short term interval
- 2) Long term interval
- 3) EMA-SMA crossover threshold limit (0.7%)
- 4) Previous interval to check directional momentum (6)

Position Size Logic

For risk management regarding the size of position, we have used the Kelly's Criterion for optimal bet size, which incorporates the probability of winning (W) and the payoff ratio (R) in the following model:

$$\text{Kelly \%} = W - [(1-W)/R]$$

But here we are unaware of how the strategy is going to perform and what are going to be our W and R values, hence we let betting start with a seed conservative value and let the model run and try to optimize itself with the few trades.

Hence at any time t the backtesting engine calculates the W for all the trades happened between $0-t$ time and also how much total win/total loss ratio have we got so far and feeds the values to the model for getting optimum position size for current bet

It also incorporates a minimum position size to take even if we are on a losing streak and the model is outputting zero/negative values for order size. We do this to get out of drawdowns. Also there is a cap on maximum fraction of capital which we can bet in order to not lose all our money and go shirtless

Margin Call is received when we reach 25% of our initial capital (1Cr) and the strategy stops

The numbers, which can be tweaked around here, are:

- 1) Initial seed value of position
- 2) Initial fraction of payoff assumption (0.6 currently)
- 3) Maximum Value that can be put at risk (50% currently)
- 4) Minimum Value to be put at risk (1% currently)

Assumptions

- 1) No liquidity constraints and zero slippage from order price
- 2) Transaction cost of 2bps incorporates the interest on margin
- 3) Initially the WinAmount/LoseAmount for the strategy is going to be 60%
- 4) The empty rows for prices in the given dataset are filled with previous days prices

Strategy Performance

After running the above-mentioned strategy on provided NIFTY EOD data for almost 1950 working days, we get the following results:

Portfolio Initial Value: 1,00,00,000
Portfolio Final Value: 11,72,39,705

Performance Metrics:

Strategy CAGR	36.02%
Buy&Hold CAGR	35.34%
Volatility	14%
Sharpe Ratio	1.81
Information Ratio	0.35

Chart 1: Positions of buys and sells with price:

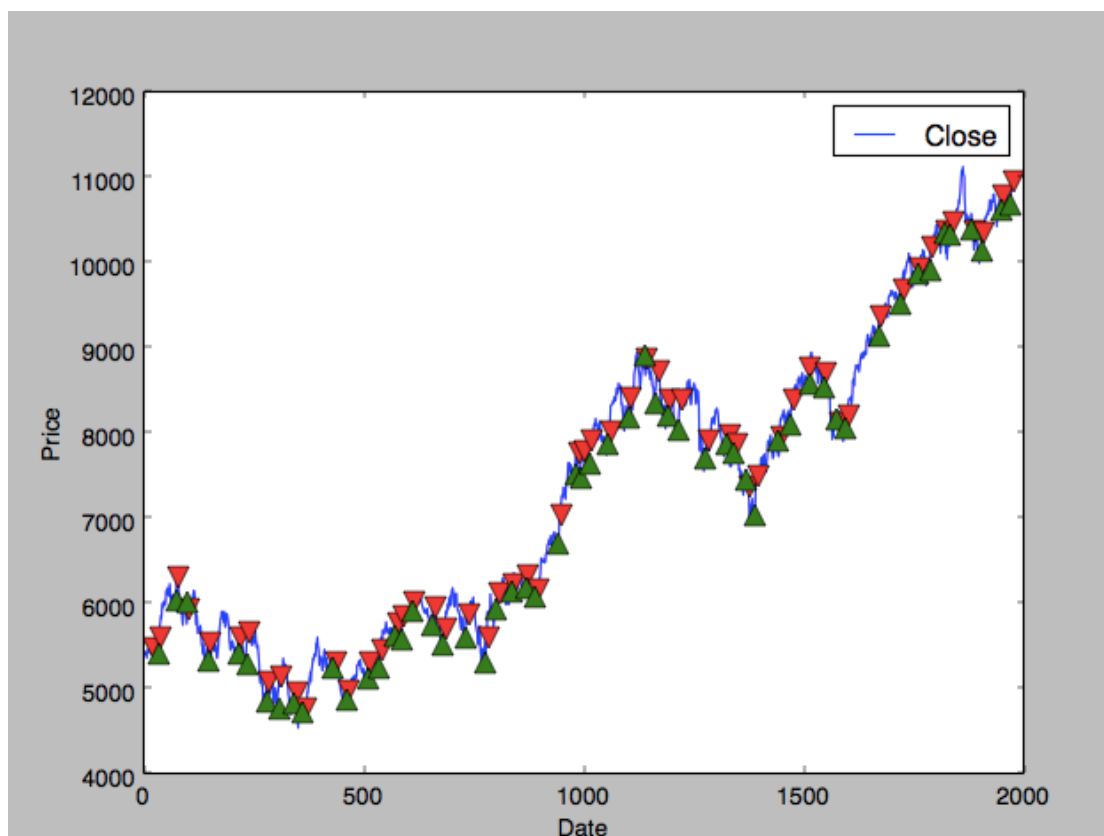
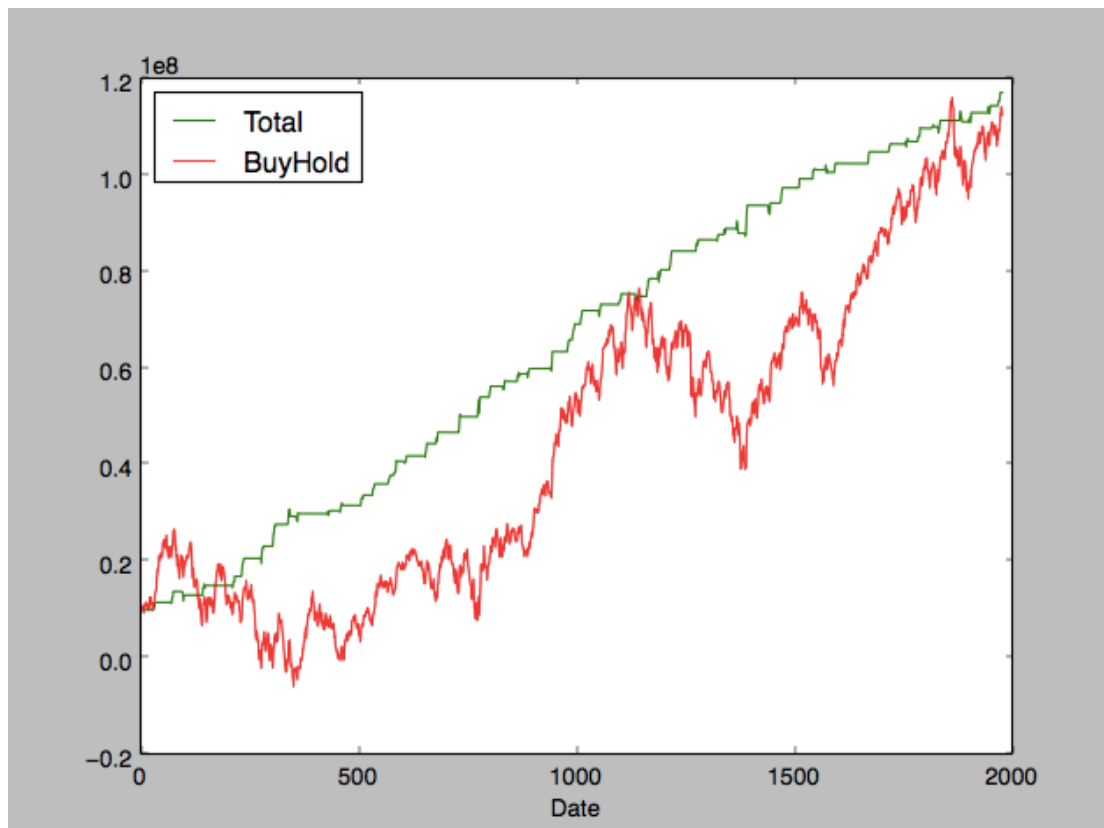


Chart 2: PNL chart of our strategy with respect to Buy&Hold:



Further possibilities

- 1) The signal can be generated using a blend of multiple indicators
- 2) The indicators can be used as features to some learning algorithm which would try to predict the pullbacks more accurately
- 3) The backtesting framework could be created as a binary to be given to the quants to code just the strategy part
- 4) More performance metrics could be added such as maximum drawdown value, maximum drawdown period Sortino ratio, Roy's ratio etc

Thanks
