ML4T Project 6 Report
Jie Lyu
jlyu31 (903329676)
jie.lyu@gatech.edu

# Indicators

### EMA (Exponential Moving Average)

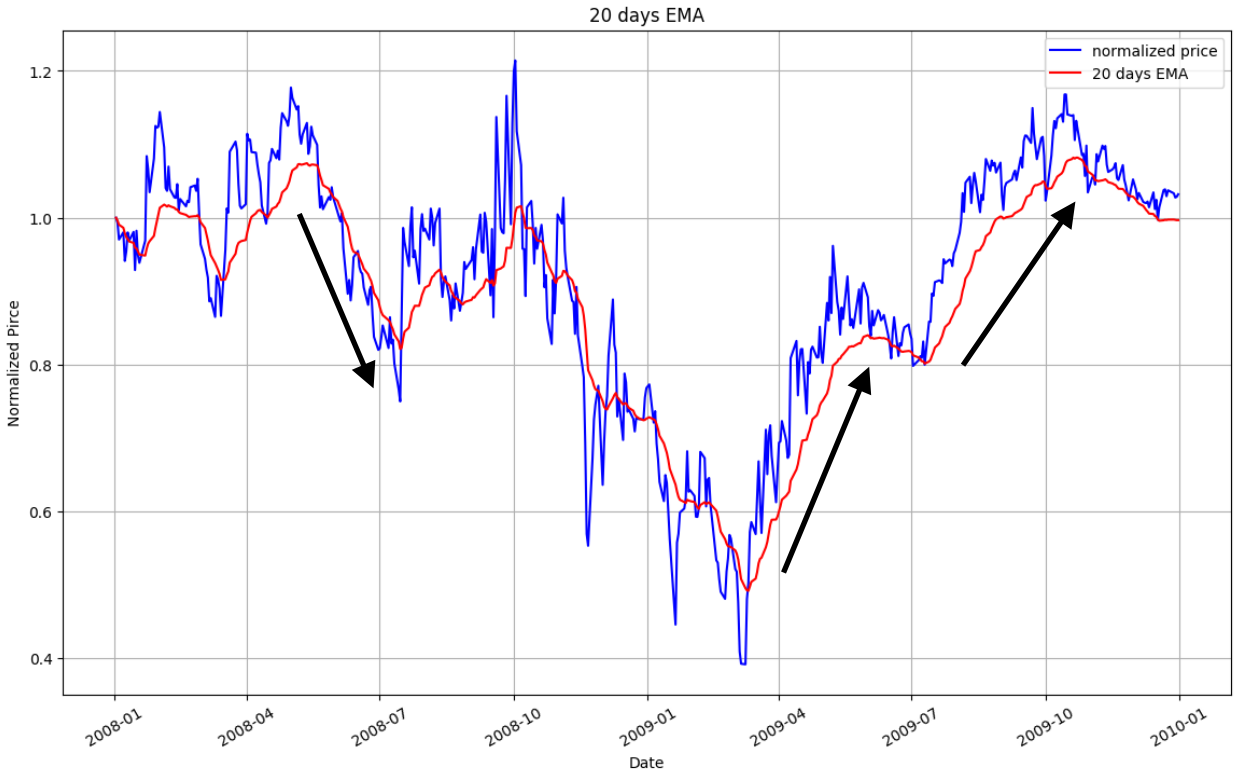
Why it works

EMA is one of the indicators here because it is simple to calculate, easy to use both for eye balling and computational analysis, and powerful. I chose EMA because I have seen a lot of professionals uses it for quick and easy eye balling. The idea behind it is simple and meaningful as opposed to a lot of "black box" indicators that are way too complicated for humans to make sense. The idea of a weighted average places more weight on recent prices, making it more responsive. Another reason I like it is because you can adjust the time window, or lag period, to get different useful informations. This feature is also implemented in my code.

How to use it

Signature: def ema(sd, ed, symbol, plot = False, window_size = 20)
Example: ema_20 = indicators.ema(sd, ed, symbol, plot=True, window_size=20)
Logics: price < ema, BUY; price > ema, SELL. Adjust window_size to get different responsiveness

Chart



**MACD (Moving Average Convergence Divergence)**

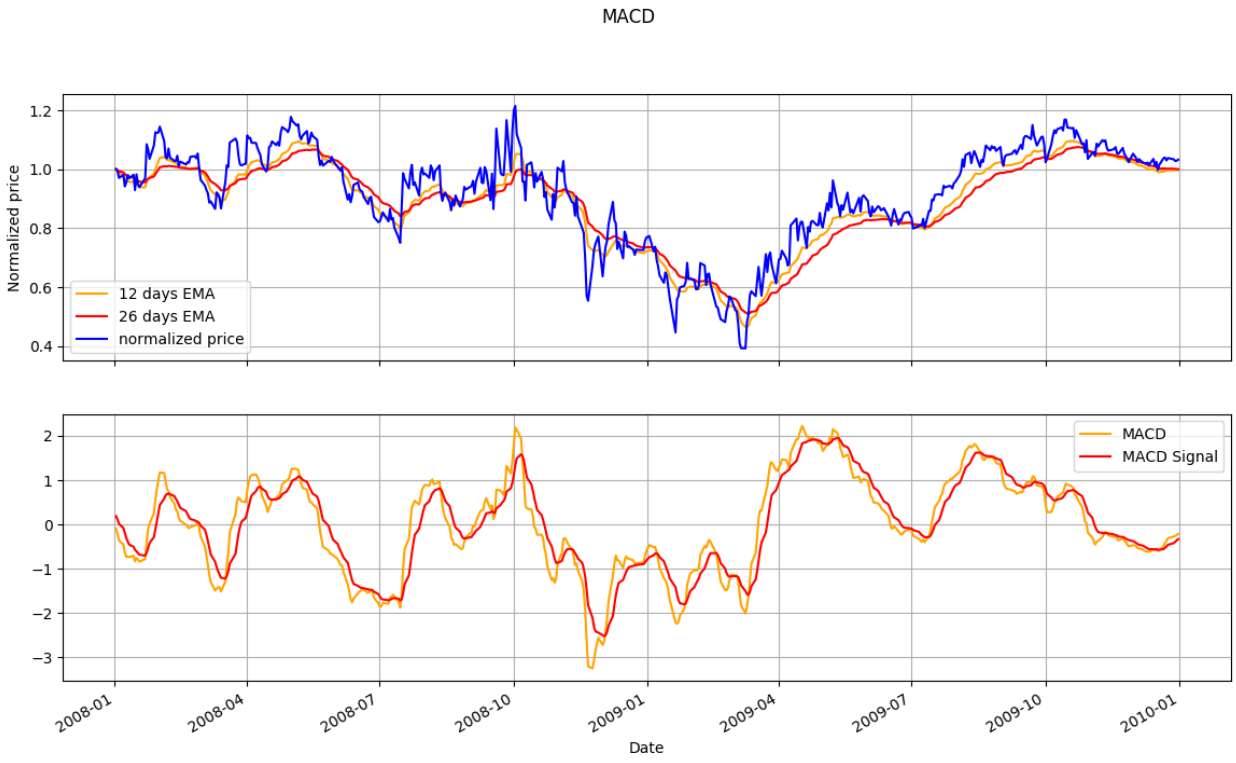
Why it works

MACD automates and advances the concept of EMA. It calculates the difference between a quick EMA and a slow EMA (in my implementation quick has a window size 12, and slow 16), and applies another EMA to smooth it. It is easier and better to incorporate MACD in an algorithm than EMAs because it is extracting information from different EMAs, and gives you a convergence and divergence indications. The logic is simple yet the way it is calculated is rigorous.

How to use it

Signature: macd(sd, ed, symbol, plot = False)
Example: macd_raw, macd_signal = indicators.macd(sd, ed, symbol, plot=True)
Logics: macd_signal > macd_raw, SELL; macd_signal < macd_raw, BUY

Chart



**TSI (True Strength Index)**

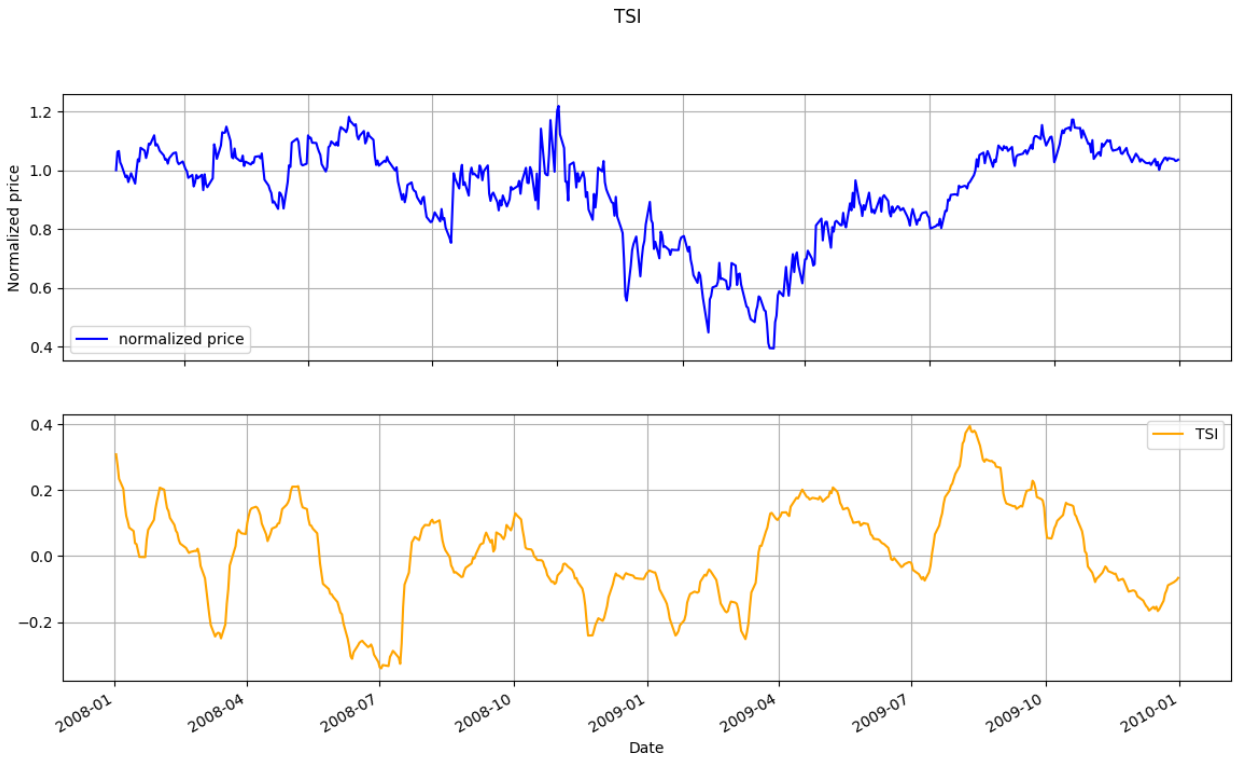
Why it works

TSI is versatile. You can use it in many ways. It double smoothes (two EMAs) both signed price changes and absolute sign changes, and then take the ratio. The output can be interrupted in many ways. The sign tells you which direction it is moving, and the magnitude tells you how strong the move is. It is also possible to tell overbought/oversold and divergence with a bit of calculation.

How to use it

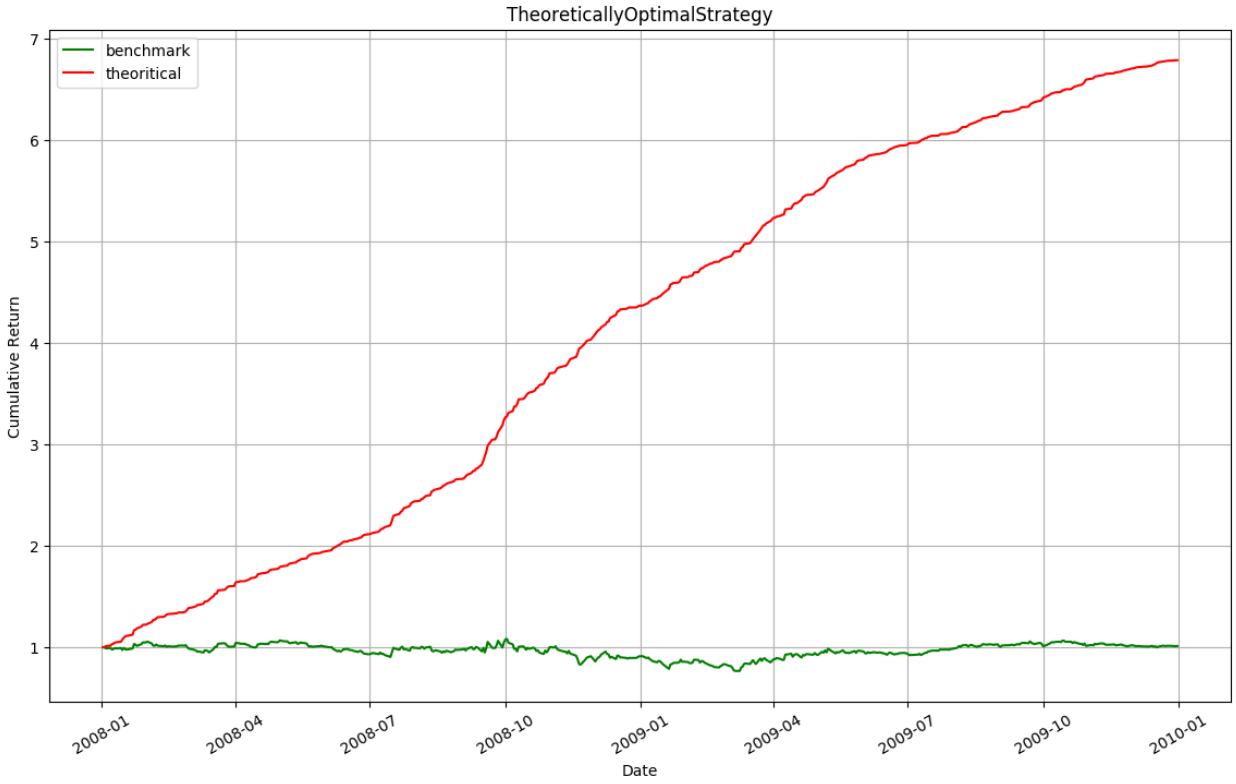
Signature: tsi(sd, ed, symbol, plot = False)
Example: macd_raw, macd_signal = indicators.macd(sd, ed, symbol, plot=True)
Logics: tsi < 0, SELL; tsi > 0, BUY

Chart



# TheoreticallyOptimalStrategy

My strategy is to look at the price tomorrow (because I am able to do that). If price drops tomorrow, I go short -1000 shares. If increases, I go long 1000 shares. Because there is no commission and impact, I trade daily for max profit. Using this strategy, I am able to make profit from every single price change.




```
[TheoreticallyOptimalStrategy]
Cumulative return: 5.7861
Stdev of daily returns: 0.004547823197908003
Mean of daily returns: 0.0038167861508578197

[Benchmark]
Cumulative return: 0.012299999999999978
Stdev of daily returns: 0.017004366271213763
Mean of daily returns: 0.00016808697819094035
```

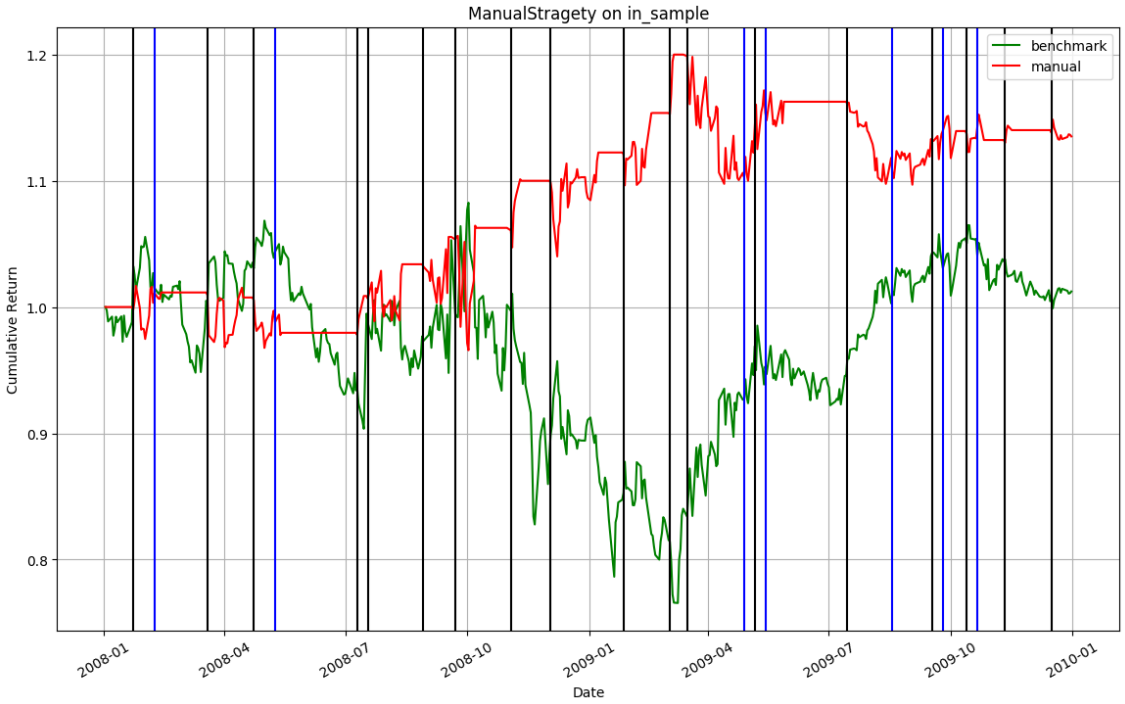
# ManualStrategy

I am using all three indicators implemented in part 1. They way I get a single decision for each day from all three indicators is to use a voting system.

At the start of a day, each indicator votes for a position: long, short or out, using the logics described in part 1. (the special case is when evidence is not compelling, an indicator will vote for out to be safe) The way I tune the model to get the best in sample performance is to give different weights for the votes. For example, I am gave MACE 10 votes when it asks for short, because from my training I know it is accurate when it senses a price drop. I gave it 2 votes for long because it didn't perform that well on long. I also set different thresholds for an indicator to trigger long or short votes. For example, I let TSI vote when the center line is crossed. However, because of the commission, I don't want it to keep voting when the TSI is wondering around the center line. I set the threshold to 0.1 so that there is a cushion zone for "out" votes. The strategy can be seen on the chart indicated by vertical lines.

Furthermore, I had a maximum frequency set to 3 days because of the commission. I also set the trigger of "out" condition more often because I am a safe trader myself, and I want my algorithm to make confident decisions. The "out" sessions are indicated by horizontal movements.

Finally, I did not tune my model for long on sample data, because I don't want to overfit it.



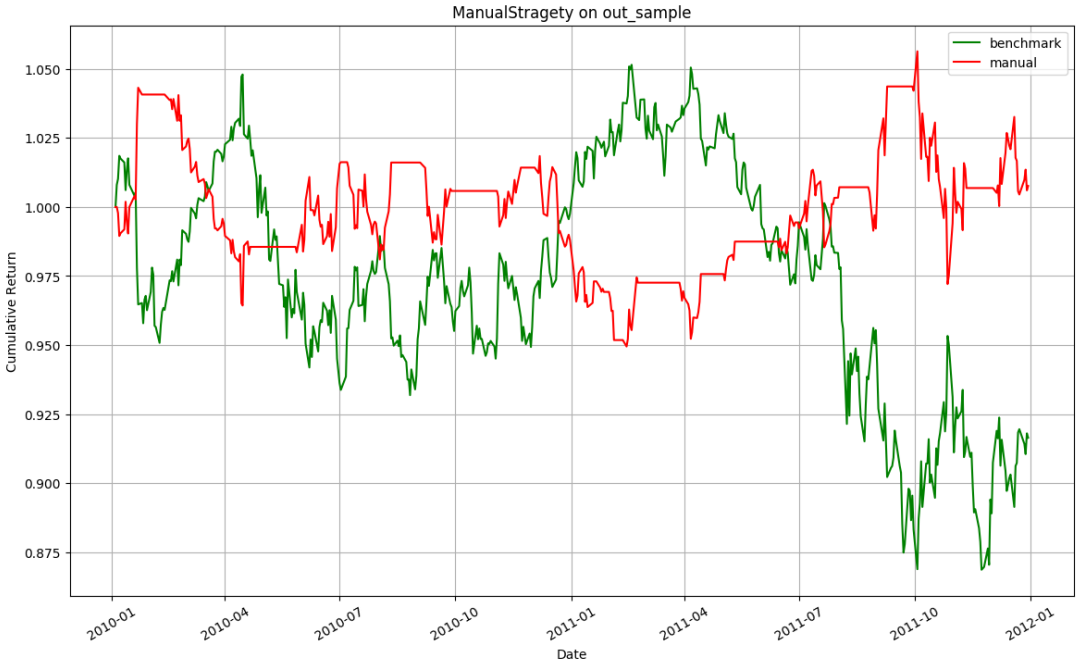

```
[ManualStrategy]
Cumulative return: 0.13506150000000083
Stdev of daily returns: 0.011081094564512495
Mean of daily returns: 0.00031257893375043135

[Benchmark]
Cumulative return: 0.01232493334014717
Stdev of daily returns: 0.017041247068174326
Mean of daily returns: 0.00016875916214640192
```

# Comparative Analysis

My out sample performance is roughly the same, just a little bit higher than my in sample performance. I was not surprised by this. Usually when we use ML optimizers, out sample will perform a lot lower than in sample, because after many iterations of training the model will inevitably overfit the in sample data (since out sample data is never seen by the model). However, since I didn't really tune my model to fit with in sample data, it is reasonable to produce a higher out sample result.

| | **In Sample (2008-01 ~ 2009-12)** | **Out Sample (2010-01 ~ 2011-12)** |
|---|---|---|
| **Stock** | Cumulative return: 0.0123<br>Stdev of daily return: 0.017<br>Mean of daily return: 0.000169 | Cumulative return: -0.0836<br>Stdev of daily return: 0.0085<br>Mean of daily return: -0.000138 |
| **Manual Strategy** | Cumulative return: 0.135<br>Stdev of daily return: 0.011<br>Mean of daily return: 0.000313 | Cumulative return: 0.0763<br>Stdev of daily return: 0.058<br>Mean of daily return: 0.000032 |

JPM does better in in sample than out sample with a positive cumulative return 0.0123. My strategy managed to yield a 0.135 cumulative return with a less stdev of daily return. I think it is due to my large "out" session or the horizontal session on my graph. Going to out sample, my strategy managed to keep a positive cumulative 0.0763 return while JPM goes down with a -0.0836 negative cumulative return. I think the reason why my model did better is that in the out sample period, there is a big price drop, and my model does shorting well.




```
[ManualStrategy]
Cumulative return: 0.007629500000001288
Stdev of daily returns: 0.005816216292546471
Mean of daily returns: 3.196853493101958e-05

[Benchmark]
Cumulative return: -0.08357911003280027
Stdev of daily returns: 0.008500158322332451
Mean of daily returns: -0.00013742923038916516
```