Backtesting System (Simple Guide)

fabz

November 23, 2025

Overview

This is a simple guide to understand my backtesting system. I get stock prices, look at how they move, and use various methods to help me know when to buy or sell.

1 DataLoader

The DataLoader is like a robot that goes and gets stock prices from the internet. It makes sure the dates are in order and fills in any missing prices with the last known price.

- I reset the index to (0,1,2,...).

- Columns that were in multiple tuples are flattened into single names.

- Adjusted close prices renamed to `adj_close`.

- Missing values are filled using forward-fill (`ffill()`).

2 Return Computation

I check how much the stock price changed from yesterday to today:

- Simple return:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

- Log return:

$$r_t = \ln \frac{P_t}{P_{t-1}}$$

I ignore the first day because I don't have a yesterday for it.

3 Indicators

Simple Moving Average (SMA)

SMA is like looking at the last N days and taking the average price:

$$SMA_t = \frac{P_t + P_{t-1} + ... + P_{t-N+1}}{N}$$

It smooths out the bumps so I can see the general trend.

Exponential Moving Average (EMA)

EMA is a special moving average that pays more attention to today's price than yesterday's:

$$EMA_t = \alpha \cdot P_t + (1 - \alpha) \cdot EMA_{t-1}$$

where $\alpha = \frac{2}{N+1}$

- Small window = reacts fast to changes.

- Big window = reacts slowly and smooths the line.

Rolling Window

A rolling window is just the last N days I look at to calculate SMA or EMA.

EWM

EWM is a pandas tool that automatically does the EMA math for me.

4 Signal Generation

Signals tell me what to do based on EMA:

- If price goes above EMA, it is a signal to buy (1).

- If price goes below EMA, it is a signal to sell (-1).

- If price is equal to EMA, I hold (0).

- Signals are shifted by 1 day to avoid looking into the future.

Formula

$$signal_t = \begin{cases} 1 & \text{if } P_{t-1} > EMA_{t-1} \\ -1 & \text{if } P_{t-1} < EMA_{t-1} \\ 0 & \text{otherwise} \end{cases}$$

Example Workflow

1. Load data and compute returns

```
loader = Data_loader("AAPL")
df = loader.load()
df = loader.returns(df)
```

2. Compute EMA

```
df = Indicators.EMA(df, window=10)
```

3. Generate EMA signals

```
df = Signal.generate_emasignal(df, 'EMA_10')
```

4. Inspect signals

```
sig_obj = Signal()
sig_obj.inspect(df, 'EMA_10')
```

Summary

SMA and EMA help me see trends in prices. EMA is smart and reacts faster to recent changes. Signals help me decide when to buy, sell, or hold, and are shifted to prevent using future data.