

You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)

An algorithm to find the best moving average for stock trading

A simple algorithm for finding the best moving average for every stock or ETF

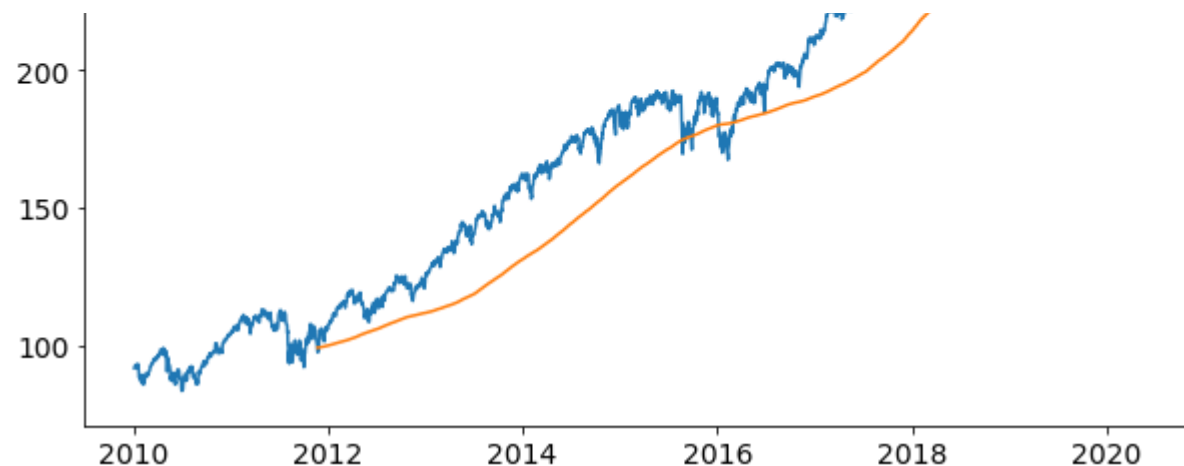


Gianluca Malato

Follow

Jun 23 · 7 min read ★





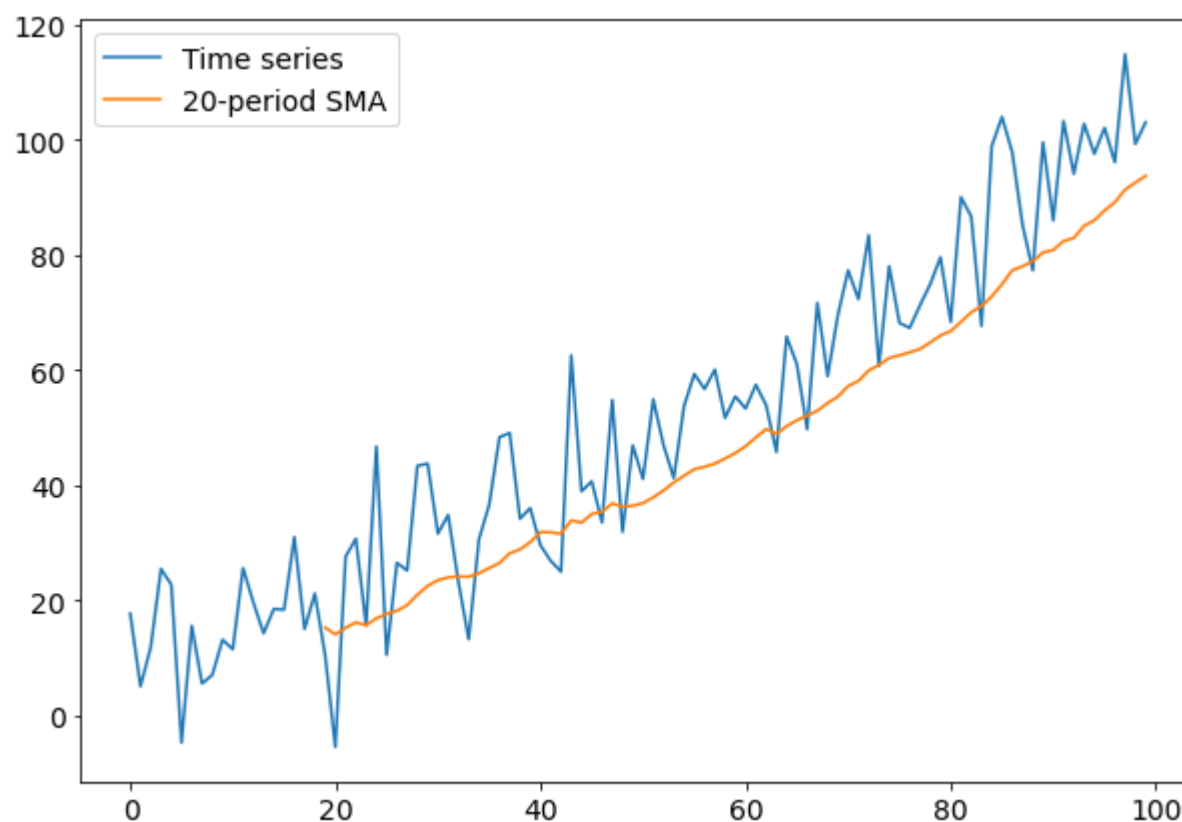
Moving averages are one of the most used tools in stock trading. Many traders actually use only this tool in their investment toolbox. Let's see what they are and how we can use Python to fine-tune their features.

What is a moving average?

In a time series, a moving average of period N at a certain time t , is the mean value of the N values before t (included). It's defined for each time instant excluding the first N ones. In this particular case, we are talking about the Simple Moving Average (SMA) because every point of the average has the same weight. There are types of moving averages that weigh every point in a different way, giving more weight to the most recent

data. It's the case of the Exponential Moving Average (EMA) or the Linear Weighted Moving Average (LWMA).

In trading, the number of previous time series observations the average is calculated from is called *period*. So, an SMA with period 20 indicates a moving average of the last 20 periods.



A time series with a 20-period Simple Moving Average

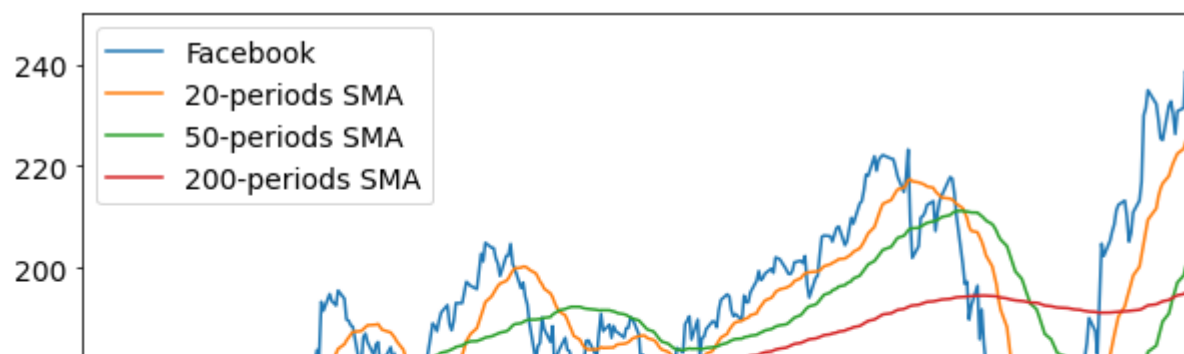
As you can see, SMA follows the time series and it's useful to remove noise from the signal, keeping the relevant information about the trend.

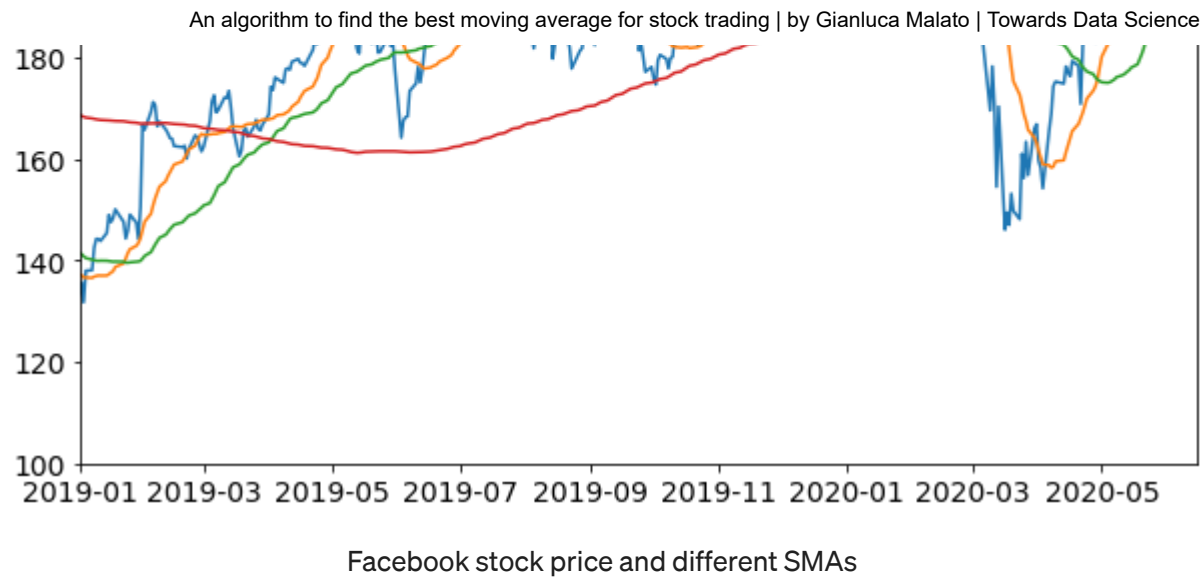
Moving averages are often used in time series analysis, for example in ARIMA models and, generally speaking, when we want to compare a time series value to the average value in the past.

How are the moving averages used in stock trading?

Moving averages are often used to detect a trend. It's very common to assume that if the stock price is above its moving average, it will likely continue rising in an uptrend.

The longer the period of an SMA, the longer the time horizon of the trend it spots.





As you can see, short moving averages are useful to catch short-term movements, while the 200-period SMA is able to detect a long-term trend.

Generally speaking, the most used SMA periods in trading are:

- 20 for swing trading
- 50 for medium-term trading
- 200 for long-term trading

It's a general rule of thumb among traders that if a stock price is above its 200-days moving average, the trend is bullish (i.e. the price rises). So they are often looking for stocks whose price is above the 200-periods SMA.

How can I choose the SMA period?

In order to find the best period of an SMA, we first need to know how long we are going to keep the stock in our portfolio. If we are swing traders, we may want to keep it for 5–10 business days. If we are position traders, maybe we must raise this threshold to 40–60 days. If we are portfolio traders and use moving averages as a technical filter in our stock screening plan, maybe we can focus on 200–300 days.

Choosing the investment period is a discretionary choice of the trader. Once we have determined it, we must try to set a suitable SMA period. We have seen 20, 50 and 200 periods, but are they always good? Well not really.

Markets change a lot during the time and they often make traders fine-tune their indicators and moving averages in order to follow volatility burst, black swans and so on. So there isn't the right choice for the moving average period, but we can build a model that self-adapts to market changes and auto-adjust itself in order to find the best moving average period.

The algorithm for the best moving average

The algorithm I propose here is an attempt to find the best moving average according to the investment period we choose. After we choose this period,

we'll try different moving averages length and find the one that maximizes the expected return of our investment (i.e. if we buy at 100 and after the chosen period the price rises to 105, we have a 5% return).

The reason of using the average return after N days as an objective function is pretty simple: we want our moving average to give us the best prediction of the trend according to the time we want to keep stocks in our portfolio, so we want to maximize the average return of our investment in such a time.

In practice, we'll do the following:

- Take some years of daily data of our stock (e.g. 10 years)
- Split this dataset into training and test sets
- Apply different moving averages on the training set and, for each one, calculate the average return value after N days when the close price is over the moving average (we don't consider short positions for this example)
- Choose the moving average length that maximizes such average return
- Use this moving average to calculate the average return on the test set

- Verify that the average return on the test set is statistically similar to the average return achieved on the training set

The last point is the most important one because it performs cross-validation that helps us avoid overfitting after the optimization phase. If this check is satisfied, we can use the moving average length we found.

For this example, we'll use different stocks and investment length. The statistical significance of the mean values will be done using Welch's test.

An example in Python

Short-term investment

First of all, we must install *yfinance* library. It's very useful for downloading stock data.

```
!pip install yfinance
```

Then we can import some useful packages:


```
import yfinance
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import ttest_ind
```

Let's assume we want to keep the SPY ETF on S&P 500 index for 2 days and that we want to analyze 10 years of data.

```
n_forward = 2
name = 'SPY'
start_date = "2010-01-01"
end_date = "2020-06-15"
```

Now we can download our data and calculate the return after 2 days.

```
ticker = yfinance.Ticker(name)
data = ticker.history(interval="1d", start=start_date, end=end_date)
data['Forward Close'] = data['Close'].shift(-n_forward)

data['Forward Return'] = (data['Forward Close'] -
data['Close'])/data['Close']
```

Now we can perform the optimization for searching the best moving average. We'll do a *for* loop that spans among 20-period moving average and 500-period moving average. For each period we split our dataset in training and test sets, then we'll look only at those days when the close price is above the SMA and calculate the forward return. Finally, we'll calculate the average forward return in training and test sets, comparing them using a Welch's test.

```
result = []
train_size = 0.6

for sma_length in range(20,500):

    data['SMA'] = data['Close'].rolling(sma_length).mean()
    data['input'] = [int(x) for x in data['Close'] > data['SMA']]

    df = data.dropna()

    training = df.head(int(train_size * df.shape[0]))
    test = df.tail(int((1 - train_size) * df.shape[0]))

    tr_returns = training[training['input'] == 1]['Forward Return']
    test_returns = test[test['input'] == 1]['Forward Return']

    mean_forward_return_training = tr_returns.mean()
    mean_forward_return_test = test_returns.mean()

    pvalue = ttest_ind(tr_returns, test_returns, equal_var=False)[1]

    result.append({
```

```
'sma_length':sma_length,  
'training_forward_return': mean_forward_return_training,  
'test_forward_return': mean_forward_return_test,  
'p-value':pvalue  
)
```

We'll sort all the results by training average future returns in order to get the optimal moving average.

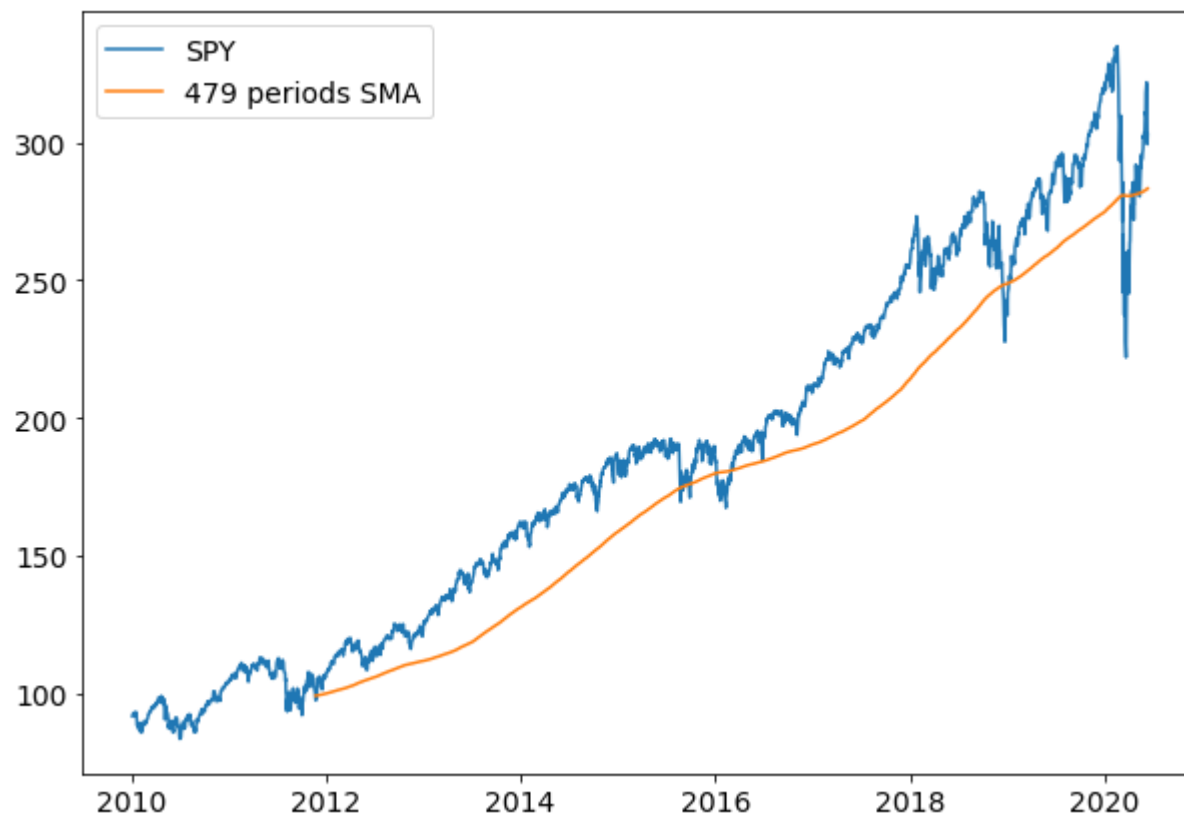
```
result.sort(key = lambda x : -x['training_forward_return'])
```

The first item, which has the best score, is:

```
result[0]  
{  
  'p-value': 0.38049738892303864,  
  'sma_length': 479,  
  'test_forward_return': 0.0005892464070523504,  
  'training_forward_return': 0.0010916823654946123  
}
```

As you can see, the p-value is higher than 5%, so we can assume that the average return in the test set is comparable with the average return in the training set, so we haven't suffered overfitting.

Let's see the price chart according to the best moving average we've found (which is the 479-period moving average).



It's clear that the price is very often above the SMA.

Long-term investment

Now, let's see what happens if we set $n_{forward} = 40$ (that is, we keep our position opened for 40 days).

The best moving average produces these results:

```
result[0]
{'p-value': 7.21116993435927e-05,
'sma_length': 473,
'test_forward_return': 0.011214348392669857,
'training_forward_return': 0.020733959450136234}
```

As you can see, the p-value is lower than 5%, so we can assume that the training phase has introduced some kind of overfitting, so we can't use this SMA in the real world. Another reason could be that volatility has changed too much and the market needs to stabilize before making us invest in it.

Finally, let's see what happens with a Gold-based ETF (ticker: GLD) with 40-days investment.

```
result[0]
{'p-value': 0.4773670114844698,
'sma_length': 136,
'test_forward_return': 0.012172903018859512,
'training_forward_return': 0.0100723311868193}
```

p-value is quite high, so there's no overfitting.

The best moving average period is 136, as we can see in the chart below.



Conclusions

In this article, we've seen a simple algorithm to find the best Simple Moving Average for stock and ETF trading. It can be easily applied every trading day in order to find, day by day, the best moving average. In this way, a trader can easily adapt to market changes and to volatility fluctuations.

All the calculations shown in this article can be found on GitHub here:
https://github.com/gianlucamalato/machinelearning/blob/master/Find_the_best_moving_average.ipynb.

Sign up for The Daily Pick

By Towards Data Science

Hands-on real-world examples, research, tutorials, and cutting-edge techniques delivered Monday to Thursday. Make learning your daily ritual. [Take a look](#)

Your email

Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

Data Science

Stock Market

Trading

Algorithmic Trading

Artificial Intelligence

Learn more.

Medium is an open platform where 170 million readers come to find insightful and dynamic thinking. Here, expert and undiscovered voices alike dive into the heart of any topic and bring new ideas to the surface. [Learn more](#)

Make Medium yours.

Follow the writers, publications, and topics that matter to you, and you'll see them on your homepage and in your inbox. Explore

Share your thinking.

If you have a story to tell, knowledge to share, or a perspective to offer — welcome home. It's easy and free to post your thinking on any topic. Write on Medium

[About](#)

[Help](#)

[Legal](#)