**Quant model reflection (EUR\_USD)**

* Notice: Self-experiment: Oanda API returns different data when compare with yfinance, therefore, we use data from Oanda in this doc as we will be trading on Oanda

**SMA (50,200****) strategy, 2014-2019, tc/each = 0.00007pip**

* We can see that before mid 2016, it does generate promising result, but at the end of 2019, it down performs the market by -1.25%
* This may happen because more traders start to use this strategy in the recent years, also the commission fee ate up most of the profit.
* Remember, using this strategy, we are short or long for the entire time, there is no neutral position
* Chart, line chart

  Description automatically generated

Granularity = daily

|  |  |  |  |
| --- | --- | --- | --- |
| **SMA pairs** | **Outperform** | **Return** | **Number of trades** |
| (50,200) | +19% | +7.1% | 6 |
| (20,100) | +26.6% | +8.4% | 18 |
| (20,50) | +35.6% | +16.2% | 28 |
| Optimise Daily bar  (37,51) | +81.6% | +62.4% | 29 |

* If we short the entire time, +12% return, if we hold, we will lose -12.1%

**Mean Reversion (Bollinger band) strategy, 2014-2019, tc = 0.00007pip,** **grand = D**

Graphical user interface, chart

Description automatically generated

In which we only got a negative return, of -3.3%, although we have an outperformance of +14.3%. Using forward testing (2020-2021), we got +0.9% returns with -3.3% down performance.

* Granularity = 4 hours: +2% returns, +19% outperform / forward test (-0.8%, -2.8%)
* Granularity = 8 hours: +3% returns, +20% outperform/ forward test (-0.7%, -2.84%)

Therefore, using the default (commonly used) setting (SMA = 30, dev = 3) we earn a little to nothing returns

I then use brute force optimisation, SMA (5 to 400), dev (1 to 8) with different granularity.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Granularity | SMA | Dev | Returns | APR | CAGR | Trades/yr | APR(2020-21) |
| **1 day** | 22 | 1 | +27.9% | +4.65% | +4.2% | 32.7 | -0.3% |
| **15 mins** | 153 | 6 | +5.5% | +0.92% | +0.9% | 11 | -0.5% |
| **1 hour** | 179 | 2 | +21.6% | +3.6% | +3.31% | 72 | -0.05% |
| **6 hours** | 73 | 1 | +39.2% | +6.5% | +5.67% | 50 | -7.3% |
| **12 hours** | 36 | 1 | +39% | +6.5% | +5.6% | 44.5 | -4.7% |

The reasons why Dev always = 1 may be because EUR\_USD is as stable as S&P500

Summary with Mean Reversion strategy:

* The default setting (SMA = 30, dev =3) does not work
* Even when we use optimisation for each granularity, but we failed on forward testing
* So, Mean Reversion strategy **does not** performance well with Forex Pair: EUR\_USD. Maybe I should avoid using this strategy in the future, especially when dealing with Forex.

**RSI Strategy**

* This index has a feature of quicker response compare with another index

Chart, line chart, scatter chart

Description automatically generatedChart, line chart, scatter chart

Description automatically generated

Period = 6: - 1.1% return, +16.9% outperformance, 49 full trades

Period = 12: –26.1% returns, -8.7% down performance, 7 full trades

Using default period (6,12) with another granularity (15mins, 30mins) performs bad

Optimise parameter with period 2 to 500

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Granularity | Period | Returns | APR | CAGR | Trades/yr | APR(2020-21) | Hit ratio |
| **1 day** | 3 | +26.6% | +4.4% | +4% | 28 | +0.3% | 0.50097 |
| **15 mins** | 33 | +23.9% |  |  | 24 | -13.9% | 0.485 |
| **30 mins** | 22 | +43% | +7.2% |  | 41 | -3.2% | 0.489 |
| **1 hour** | 14 | +43.4% |  |  |  | -8.3% | 0.493 |
| **4 hours** |  |  |  |  |  |  |  |
| **8 hours** |  |  |  |  |  |  |  |

Did not finish the table above because Oanda API has difficulty access hourly bar (unstable)

**Summary:**

* Even we find the optimised period in back testing, we **failed** on forward testing
* Therefore, using this strategy alone is not a good practice.
* From the graph above, it is noticed that the strategy **work well on stable trend** (small up/down)

**Future improvement:**

* For Low granularity, use period (12 to 50) to avoid excessive trade
* For large granularity, use smaller period.
* Try using **Bottom Divergence/Top Divergence**

**MACD strategy**

* This index is known to has a lagged effect when market change
* Chart, line chart, scatter chart

  Description automatically generated
* EMA\_short = 12, EMA\_long = 26
* -3% returns but +14.5% outperformance, full trades = 85

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Granularity | Returns | Outperformance | Trades | Hit ratio |
| 15 mins | -61% | -42.6% | 5299 | 0.4767 |
| 30 mins | -29% | -10% | 2486 | 0.4829 |
| 1 hour | -13.7% | +5% | 1237 | 0.4847 |
| 4 hours | -10.4% | +9.13% | 317 | 0.4854 |
| 8 hours | -31% | -13.1% | 191 | 0.4884 |
| 1 day | -2.1% | +15.36% | 59 | 0.495 |

* Hit ratio increases as granularity increases

Using optimisation with EMA\_short(2-50), EMA\_long(3,85,1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Granularity | short | long | Returns | Trades | Hit ratio |
| 15 mins | Did not outperform shorting entire time | | | | |
| 30 mins | Did not outperform shorting entire time | | | | |
| 1 hour | Did not outperform shorting entire time | | | | |
| 4 hours | 49 | 47 | +38.8% | 137 | 0.4999 |
| 8 hours | 35 | 4 | +47.8% | 271 | 0.4973 |
| 1 day | 9 | 3 | +33.4% | 85 | 0.492 |

**Summary:**

* It is interesting to see when using optimisation, EMA\_short < EMA\_long performs better…
* Future development: try using **Bottom Divergence/Top Divergence**

**Summary of using traditional strategy (SMA, Mean Reversion, RSI and MACD)**

* Does not perform well, although SMA strategy with (SMA20,50) has a +16% returns. But it is simply not good enough
* One of the main reasons may be that EUR\_USD is the most stable pair (less market inefficient to find), therefore my “simple” strategy is not good enough to earn a decent profit.
* Loading data from csv file will be much faster than loading from web API
* Instead of just long/short, try using Top/bottom divergence when using RSI and MACD
* Using optimisation has little effect when it comes to forward testing

**Machine Learning**

**Using sklearn.linear\_model.LogisticRegression( C = 1e6, max\_iter = 100000, multi\_class = “ovr”)**

This is a very simple machine learning strategy, it is short or long the entire time

Lags = 5

Training set -> 2012-01-01 to 2019-01-01

Test set -> 2019-01-01 to 2021-12-31

Graphical user interface, chart, scatter chart

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Granularity | Returns | Outperform | Trades | Hit ratio |
| 1 day | -6% | -6.4% | 351 full | 0.4963 |
| 15 mins | - 95.2% | -94.2% | 26067 full | 0.5099 |
| 30 mins | -62.2% | -70.58% | 11658 full | 0.5143 |
| 1 hour | -55.23% | -43.8% | 5781 full | 0.5122 |
| 4 hours | -16.1% | -15% | 988 full | 0.4990 |
| 8 hours | +14.4 % | +14.76% | 806 full | 0.5031 |

Granularity = daily, return = (return, outperformance)

Lags = 1: (1.013311, 0.013091)

Lags = 2: (0.89385, -0.105078)

Lags = 3: (0.928536, -0.068982)

Lags = 4: (0.965817, -0.037456)

Lags = 5: (0.939871, -0.064137)

Lags = 6: (1.004783, -0.002274)

Lags = 30: (0.973195, -0.0293)

Summary:

It produces promising returns when tc = 0. Seems like simple machine learning using lags(returns) does not work (hit ratio always around 0.5 no matter what parameters are used). In the future, try using another indicator lags instead of price lag.

Machine Learning with SMA, using lags = 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Granu.** | **Returns/outperform** | **Hit\_ratio** | **Trades (2,0)** |
| **SMA = 5** | **daily** | **(1.074128, 0.07151)** | **0.5** | **128/670** |
|  | 8h | (0.890772, -0.102449) | 0.4895 | 520/1868 |
|  | 4h | (1.043999, 0.050214) | 0.5044 | 316/4442 |
|  | 1h | x | x | x |
|  | 30m | xxx | xxx | xxx |
| SMA = 10 | daily | (1.074491, 0.071873) | 0.5175 | 303/495 |
|  | 8h | (0.983505, -0.009716) | 0.4853 | 314/2074 |
|  | 4h | (1.002822, 0.009038) | 0.4975 | 244/4754 |
|  | 1h | (x | x | x |
|  | 30m | xxx | xxx | xxx |

Machine Learning with RSI, using lags = 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Granu.** | **Returns/outperform** | **Hit\_ratio** | **Trades (2,0)** |
| RSI = 6 | Daily | (0.999, -0.004) | 0.4832 | 409/395 |
|  | 30m | (0.328, -0.659) | 0.5070 | 10908/26493 |
| RSI = 12 | Daily | (1.002, -0.002) | 0.4994 | 415/389 |
|  | 30m | (0.345, -0.643) | 0.5056 | 10570/26831 |

Machine Learning with MACD, using lags = 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Granu.** | **Returns/outperform** | **Hit\_ratio** | **Trades (2,0)** |
| ewm(12,26) | Daily | (1.222828, 0.22021) | 0.5238 | 216/582 |
|  | 8h | (0.9527, -0.040521) | 0.4904 | 472/1916 |
|  | 4h | (0.888236, -0.105548) | 0.5025 | 326/4451 |
|  | 30m | (0.824162, -0.163409) | 0.5107 | 3119/34281 |

Machine Learning using combination features (returns, MACD, SMA = 5, SMA = 20)

|  |  |  |  |
| --- | --- | --- | --- |
| **Granu.** | **Returns/outperform** | **Hit\_ratio** | **Trades (2,0)** |
| Daily | (1.029449, 0.022392) | 0.4994 | 293/510 |
| 1h | (0.537032, -0.453502) | 0.5123 | 5633/13084 |
| 30mins | (0.244538, -0.742921) | 0.5148 | 13766/23634 |

If we make all prediction right, we will get a (9.894972, 8.887915) returns

Seems like ML with MACD produce a slightly better results, RSI needs to do more testing. When using ML with another index, seems like it is a 50/50 win

**Combination strategies**

Test set (beginning of 2019 to end of 2021)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Gran** | **Returns** | **Hit ratio** | **Trades (2,0,1)** |
| ML(MACD), SMA (20,50) | Daily | (0.93303, -0.06920) | 0.5095 | 226/563 |
| ML(MACD), SMA (50,200) | Daily | (1.063176, 0.06) | 0.5134 | 246/498 |
|  | 1h | (0.790856, -0.198138) | 0.5056 | 1290/17368 |
| SMA (20,50), MACD | Daily | (1.004632, 0.002394) | 0.5114 | 18/771 |
| SMA (20,50), RSI (6) | Daily | (0.947139, -0.0551) | 0.4809 | 18/771 |
| SMA (20,50), Bollinger(d2) | Daily | (0.983123, -0.019116) | 0.4889 | 3/786 |
| MACD, RSI (6) | Daily | (0.977169, -0.025069) | 0.4936 | 12/777 |
| MACD, Bollinger(d2) | Daily | (0.946811, -0.055427) | 0.4903 | 1/788 |
| Bollinger(d2), RSI (6) | Daily | (1.096617, 0.094379) | 0.4885 | 13/775/1 |
| MACD, RSI (6), Bollinger(d2) | Daily | (0.946811, -0.055427) | 0.4903 | 1/788 |
| MACD, RSI (6),  SMA (20,50) | Daily | (0.962902, -0.039336) | 0.4872 | 12/777 |

**Using DNN Machine Learning (see DNNModel.py) (**2018-12-12 to 2020-12-30)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features (lags =5)** | **Granu.** | **Returns** | **Hit ratio** | **Trades** |
| "Dir", "SMA", "Boll", "MACD", "Min", "Max", "Mom", "Vol" | Daily | (0.936674, -0.063704) | 0.4330 | 8/783 |
|  | 1 h | (0.982807, -0.006654) | 0.5014 | 366/18337 |
|  | 15 mins | (0.580218, -0.409027) | 0.50578 | 6988/67752 |
| "Dir", "SMA", "Boll", "Min", "Max", "Mom", "Vol" | Daily | (0.959688, -0.04069) | 0.3042 | 1/790 |
|  | 1h | (0.864001, -0.12546) | 0.49489 | 394/18309 |
|  | 15 mins | (0.490079, -0.499165) | 0.5072 | 8061/66678 |
| "Dir", "SMA", "Boll", "MACD", "Mom" | Daily | (0.978802, -0.021576) | 0.2967 | 1/790 |
|  | 1h | (0.897414, -0.092047) | 0.49869 | 536/18167 |
|  | 15mins | (0.59974, -0.389505) | 0.50442 | 5899/68840 |
| "Dir", "SMA", "Boll", "MACD | Daily | (0.966872, -0.033507) | 0.2891 | 1/790 |
|  | 1h | (0.961209, -0.028252) | 0.50243 | 667/18036 |
|  | 15mins | (1.003661, 0.014416) | 0.4948 | 375/74364 |
| “Dir”,”SMA” | Daily | (0.999812, -0.000567) | 0.3522 | 1/790 |
|  | 1h | (0.988121, -0.00134) | 0.4953 | 264/18439 |
|  | 15mins | (0.98418, -0.005065) | 0.4910 | 748/73991 |

Lower granularity actually performance better

Without trading cost, using 15mins granularity will actually generate a good profit (feature: "Dir", "SMA", "Boll", "Min", "Max", "Mom", "Vol") Chart, line chart

Description automatically generated

Live trading using MACD and SMA (10,30)