

# MetaTrader5 と Python による自動売買

富山県立大学 情報基盤工学講座  
1515010 大谷和樹

指導教員：奥原浩之

## 1 はじめに

MetaTrader5 (MT5) と Python で、様々なインジケータを利用して自動売買を行うシステムの開発を行っている。研究の背景や目的などについては、しっかりとした文にまとめることができていないため、今後まとめていく必要がある。

## 2 TA-Lib

前回の研究紹介をした際に、使用するインジケータの計算は Python で簡単にテクニカル指標が計算できる「pyti」というライブラリを使用することにしていた。しかし、うまく pyti のプログラムが動かなかったため、pyti と同じように Python でテクニカル指標が計算できる「TA-Lib」というライブラリを使用することにした。

```

# 単純移動平均(SMA: Simple Moving Average)
output = np.c_[output, talib.SMA(close)]
cols += ['SMA']

# 加重移動平均(WMA: Weighted Moving Average)
output = np.c_[output, talib.WMA(close)]
cols += ['WMA']

# 指数移動平均(EMA: Exponential Moving Average)
output = np.c_[output, talib.EMA(close)]
cols += ['EMA']

# 2重指数移動平均(DEMA: Double Exponential Moving Average)
output = np.c_[output, talib.DEMA(close)]
cols += ['DEMA']

# 3重指数移動平均(TEMA: Triple Exponential Moving Average)
output = np.c_[output, talib.TEMA(close)]
cols += ['TEMA']

# 三角移動平均(TMA: Triangular Moving Average)
output = np.c_[output, talib.TMA(close)]
cols += ['TMA']

# Kaufmanの適応型移動平均(KAMA: Kaufman Adaptive Moving Average)
output = np.c_[output, talib.KAMA(close)]
cols += ['KAMA']

# MESAの適応型移動平均(MAMA: MESA Adaptive Moving Average)
for arr in talib.MAMA(close):
    output = np.c_[output, arr]
cols += ['MAMA', 'FAMA']

# トレンドライン(Hilbert Transform - Instantaneous Trendline)
output = np.c_[output, talib_HT_TRENDLINE(close)]
cols += ['HT_TRENDLINE']

# ボリンジャーバンド(Bollinger Bands)
for arr in talib.BBANDS(close):
    output = np.c_[output, arr]
cols += ['BBANDS_upperband', 'BBANDS_middleband', 'BBANDS_lowerband']

# MIDPOINT over period
output = np.c_[output, talib.MIDPOINT(close)]
cols += ['MIDPOINT']

```

図 1: TA-Lib でインジケータを計算するプログラム (一部)

上の図1からわかるように、TA-Libを使うこと1行のコードでインジケータの計算を行うことができる。

### 3 進捗状況

今回は、TA-Lib を使ったインジケータの計算が実際に行うことができるのかを確かめるために、まずは過去データを用いてインジケータの計算を行った。計算結果が以下の図2である。初めの方の空白部分は計算に使うための情報が揃っていないため計算が行われていなかった部分である。

[illegible]

図 2: 過去データを用いたインジケータの計算結果

ことができることが確認できたため、今度は MT5 を使って実際に自動売買をすることができるか確認した。平松くんが作成していたプログラムの中で、売買ルールに関係している部分を自身が作ったものに書き換え、実際に動かした。今回は単純移動平均 (SMA) を用いたゴールデンクロス・デッドクロスで売買を決定している。その際の売買の結果が以下の図 3 である。

2020.07.29 11:39	usdjpy	66767093	sell	0.1	104.861		2020.07.29 11:48.21	104.870	-90	-0.01%
2020.07.29 11:59	usdjpy	66769421	sell	0.1	104.912		2020.07.29 12:19.37	104.916	-40	-0.00%
2020.07.29 12:30	usdjpy	66772193	sell	0.1	104.910		2020.07.29 12:39.48	104.944	340	+0.03%
2020.07.29 13:44	usdjpy	66773308	sell	0.1	104.921		2020.07.29 13:52.57	104.933	120	+0.01%
2020.07.29 13:00	usdjpy	66774724	sell	0.1	104.930		2020.07.29 13:03.22	104.935	50	+0.00%
2020.07.29 13:07	usdjpy	66775421	sell	0.1	104.934		2020.07.29 13:16.29	104.966	320	+0.03%
2020.07.29 13:22	usdjpy	66776285	sell	0.1	104.943		2020.07.29 13:25.53	104.940	-70	-0.01%
2020.07.29 13:27	usdjpy	66777210	sell	0.1	104.945		2020.07.29 13:47.47	104.998	530	+0.05%
2020.07.29 13:48	usdjpy	66779219	sell	0.1	104.949		2020.07.29 13:51.12	105.054	105	+0.01%
2020.07.29 13:53	usdjpy	66779564	sell	0.1	104.946		2020.07.29 13:44.82	104.992	-50	-0.00%
2020.07.29 14:14	usdjpy	66781937	sell	0.1	105.019		2020.07.29 14:22.23	104.995	-240	-0.02%
2020.07.29 14:29	usdjpy	66783651	sell	0.1	104.974		2020.07.29 14:33.55	104.979	50	+0.00%
2020.07.29 14:39	usdjpy	66784560	sell	0.1	104.935		2020.07.29 14:50.51	105.027	920	+0.09%
2020.07.29 15:12	usdjpy	66790174	sell	0.1	105.015		2020.07.29 15:15.41	105.015	0	0.00%
2020.07.29 15:26	usdjpy	66790119	sell	0.1	105.087		2020.07.29 15:50.53	105.078	90	0.01%
2020.07.29 15:41	usdjpy	66791674	sell	0.1	105.077		2020.07.29 16:08.02	105.040	-40	-0.04%
2020.07.29 16:23	usdjpy	66796768	sell	0.1	104.976		2020.07.29 16:30.31	105.061	850	+0.08%
2020.07.29 16:37	usdjpy	66799496	sell	0.1	105.016		2020.07.29 16:43.35	105.062	50	+0.00%
2020.07.29 16:46	usdjpy	66800618	sell	0.1	105.022		2020.07.29 16:55.24	105.045	230	+0.02%
2020.07.29 16:56	usdjpy	66801484	sell	0.1	105.038		2020.07.29 17:00.08	105.006	-320	-0.03%
2020.07.29 17:08	usdjpy	66803100	buy	0.1	104.964		2020.07.29 17:20.25	104.987	230	0.02%
2020.07.29 17:45	usdjpy	66801224	sell	0.1	104.942		2020.07.29 18:10.25	105.071	1,290	0.12%
2020.07.29 18:15	usdjpy	66812912	sell	0.1	105.065		2020.07.29 18:46.09	105.029	-360	-0.03%
2020.07.29 19:00	usdjpy	66817392	sell	0.1	105.070		2020.07.29 19:03.16	105.055	-150	-0.01%
2020.07.29 19:05	usdjpy	66818044	buy	0.1	104.987		2020.07.29 19:09.07	105.000	-380	-0.04%

図 3: SMA を用いた売買の結果

元々の平松くんのプログラムと同じように SMA 単体だけでは勝てないことが見てわかる。

## 4 おわりに

使用するライブラリを変更することにした。こちらのライブラリは実際に動くことが確認できたので、今後は使用するインジケータの選別とそれを用いた売買ルールの設定・動作確認を急ぐと共に、研究の肝である機械学習での売買ルールの選別方法を考える部分に取り掛かりたい。

## 参考文献

- [1] <https://qiita.com/ryoshi81/items/983c06e0bf859b280eba>  
Accessed: July 30, 2020.
- [2] [https://www.mql5.com/ja/docs/integration/python\\_metatrader5/mt5copyratesrange-py](https://www.mql5.com/ja/docs/integration/python_metatrader5/mt5copyratesrange-py),  
Accessed: July 30, 2020.
- [3] <https://tkstock.site/2019/11/21/python-quandl-get-stock-price-data/>  
Accessed: July 30, 2020.

これにより実際に TA-Lib を用いてインジケータの計算を行う