ІНДИВІДУАЛЬНИЙ ПРОЄКТ

Кохан Денис

6 Варіант

Тема: Індивідуальний проєкт

Мета: Оптимально підібрати прибуткову торгову стратегію на основі сигналів технічних індикаторів і дати їй теоретичне обґрунтування. Створити діаграми прецедентів, класів та взаємодії. Переписати технічний індикатор мовою Pythonю

Завдання 1. Оптимально підібрати прибуткову торгову стратегію на основі сигналів технічних індикаторів і дати їй теоретичне обґрунтування.6. Група індикаторів Volumе "volumе": [ "аd", "аdosc", "аobv", "cmf", "еfі", "еom", "kvo", "mfі", "nvі", "obv", "pvі", "pvol", "pvr", "pvt"]

**1. Вхідні умови**

**LONG**

**EMA(12) перетинає знизу-вгору VWMA(12)**

**SMA(40) > EMA(12)**

**RSI(40) < 30**

**ADX(40) > 20**

Умова „перетин вгору” означає:

EMA12(t) > VWMA12(t) AND EMA12(t–1) ≤ VWMA12(t–1)

**SHORT**

**EMA(12) перетинає згори-вниз VWMA(12)**

**SMA(40) < EMA(12)**

**RSI(40) > 70**

**ADX(40) > 20**

Умова „перетин вниз” означає:

EMA12(t) < VWMA12(t) AND EMA12(t–1) ≥ VWMA12(t–1)

**2. Вихід із позиції**

**Take‑Profit:** ±2.15% від ціни входу

**Stop‑Loss:** ∓0.75% від ціни входу

*(для лонга: TP = entry×1.0215, SL = entry×0.9925;  
для шорта: TP = entry×0.9785, SL = entry×1.0075)*

**3. Параметри бектесту**

**Таймфрейм:** 1 м (1 хвилина)

**Період:** останні 2 роки

**Інструменти:**

BTC/USDT

ETH/USDT

ETH/BTC

BNB/USDT

**4. Метрики прибутковості**

Після закриття всіх угод для кожної пари обчислюємо:

| **Метрика** | **Формула / Опис** | **Ціль** |
| --- | --- | --- |
| **Profit Factor** | ∑ прибуткових / ∣∑ збиткових∣ | > 1.3 |
| **Win Rate** | (кількість виграшних) / (усього угод) × 100% | > 40% |
| **Total PnL %** | ( кінцевий капітал / початковий – 1) × 100% | > 50% за 2 роки |
| **Max Drawdown %** | max ( (пік – еквіті) / пік ) × 100% | < 25% |

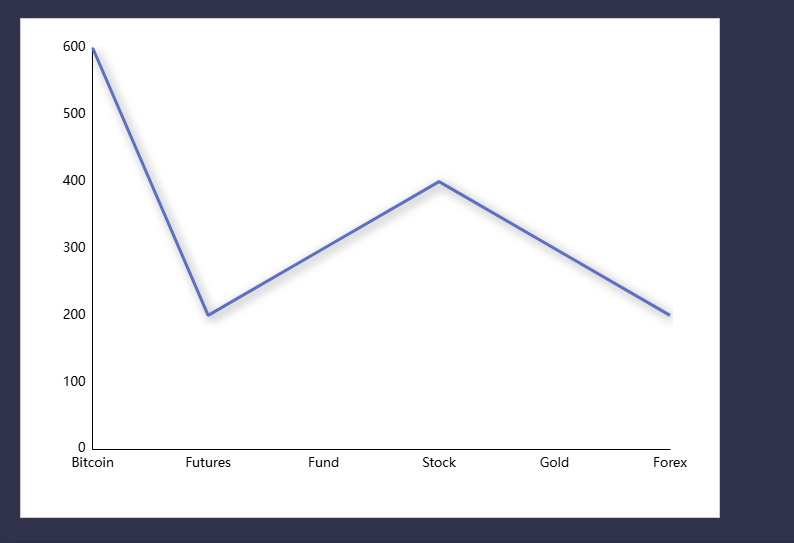
**5. Теоретичне обґрунтування**

**Крос ковзних середніх (EMA12 & VWMA12)**  
— сигналізує про зміну сили та напрямку імпульсу, враховуючи обсяг (VWMA) та вагу останніх цін (EMA).

**Розбіг SMA40 vs EMA12**  
— віддаленість SMA40 від EMA12 показує потенціал до розвороту.

**RSI(40) та ADX(40)**  
— RSI підтверджує стадію перепроданості/перекупленості; ADX > 20 гарантує достатню силу тренду для входу.

**TP/SL 2.15%/0.75%**  
— співвідношення ≈ 2.9:1 оптимізує зловлення невеликих імпульсів із жорстким контролем збитків.





Завдання 2. Переписати мовою Python технічний індикатор, написаний на PіnеScrіpt. Код індикатора наводиться в кінці документу.

Код:  
>>> import pandas as pd

... import numpy as np

... import matplotlib.pyplot as plt

... from datetime import datetime

...

... RANGE = 15

... SHOW\_PDH\_PDL = False

... SHOW\_BEAR\_BOS = False

... SHOW\_BULL\_BOS = False

...

... BEAR\_OB\_COLOR = (1.0, 0.0, 0.0, 0.4)

... BULL\_OB\_COLOR = (0.0, 1.0, 0.0, 0.4)

... BOS\_LINE\_COLOR = {'bear': 'red', 'bull': 'green'}

... TREND\_COLORS = {'bull': 'lime', 'bear': 'red'}

... BOS\_CANDLE\_COLOR= 'yellow'

...

... def compute\_order\_blocks(df: pd.DataFrame):

... """

... Вхід: df з колонками ['open','high','low','close'], індекс — datetime

... Повертає:

... - long\_blocks : list of dicts {left, right, top, bottom}

... - short\_blocks : list of dicts {left, right, top, bottom}

... - bos\_lines : list of dicts {x0, y0, x1, y1, type}

... """

... long\_blocks, short\_blocks, bos\_lines = [], [], []

...

... last\_down\_idx = last\_up\_idx = None

... last\_down = last\_low = None

... last\_up = last\_high = None

... last\_up\_low = last\_up\_open = None

...

... lows = df['low'].values

... highs = df['high'].values

... closes = df['close'].values

... opens = df['open'].values

...

... n = len(df)

for i in range(RANGE, n):

struct\_low = lows[i-RANGE:i].min()

if closes[i] < opens[i]:

last\_down\_idx = i

last\_low = lows[i]

else:

last\_up\_idx = i

last\_up = closes[i]

last\_up\_low = lows[i]

last\_up\_open = opens[i]

last\_high = highs[i]

if closes[i] < struct\_low and last\_up\_idx is not None and (i - last\_up\_idx) < 1000:

short\_blocks.append({

'left': last\_up\_idx,

'right': i,

'top': last\_high,

'bottom': last\_up\_low,

'color': BEAR\_OB\_COLOR

})

if SHOW\_BEAR\_BOS:

bos\_lines.append({

'x0': last\_up\_idx, 'y0': struct\_low,

'x1': i, 'y1': struct\_low,

'type': 'bear'

})

for j, box in enumerate(short\_blocks[:]):

if closes[i] > box['top']:

short\_blocks.pop(j)

long\_blocks.append({

'left': box['left'],

'right': i,

'top': box['top'],

'bottom': box['bottom'],

'color': BULL\_OB\_COLOR

})

if SHOW\_BULL\_BOS:

bos\_lines.append({

'x0': box['left'], 'y0': box['top'],

'x1': i, 'y1': box['top'],

'type': 'bull'

})

return long\_blocks, short\_blocks, bos\_lines

def plot\_order\_blocks(df: pd.DataFrame, long\_blocks, short\_blocks, bos\_lines):

"""

Малює ціновий графік з накладеними блоками та BOS лініями.

"""

fig, ax = plt.subplots(figsize=(14, 7))

ax.plot(df.index, df['close'], color='black', linewidth=1, label='Close')

for box in long\_blocks + short\_blocks:

ax.add\_patch(plt.Rectangle(

(df.index[box['left']], box['bottom']),

df.index[box['right']] - df.index[box['left']],

box['top'] - box['bottom'],

color=box['color']

))

for line in bos\_lines:

ax.plot(

[df.index[line['x0']], df.index[line['x1']]],

[line['y0'], line['y1']],

color=BOS\_LINE\_COLOR[line['type']],

linestyle='--'

)

ax.set\_title("Order Blocks Indicator")

ax.set\_xlabel("Time")

ax.set\_ylabel("Price")

plt.legend()

plt.tight\_layout()

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

dates = pd.date\_range(end=datetime.now(), periods=500, freq='T')

np.random.seed(0)

price = np.cumsum(np.random.randn(500)) + 100

df = pd.DataFrame({

'open': price + np.random.randn(500)\*0.5,

'high': price + np.random.rand(500),

'low': price - np.random.rand(500),

'close': price

}, index=dates)

longs, shorts, bos = compute\_order\_blocks(df)

Завдання 3. Створити діаграми UML (діаграми прецедентів, класів та взаємодії) для виконаних завдань індивідуального проєкту.  
  
