

راهنمای کاربر برای pytse-filter

این راهنمای کاربر به شما کمک میکند تا از پروژه pytse-filter استفاده کنید، کتابخانه ی پایتون که قادر است داده های نمادهای بورس تهران را با استفاده از فیلتر متنی تعریف شده توسط کاربر پردازش کرده و نتایج را برگرداند.

پیش نیازها

برای استفاده از کتابخانه pytse-filter ، مطمئن شوید که:

- پایتون ۳ بر روی سیستم شما نصب شده است.
- بسته های requests, pandas, pandas_ta, tqdm با استفاده از pip نصب شده اند.
- بسته pytse-filter از طریق GitHub یا PyPI نصب شده است.

نصب:

شما می توانید pytse-filter را با استفاده از روش های زیر نصب کنید:

```
pip install pytse_filter
```

یا برای نسخه آخر مستقیماً از مخزن: github:

```
pip install git+"https://github.com/farhad-mohammadi/pytse-filter.git"
```

یا مخزن را کلون کرده و از دایرکتوری محلی نصب کنید:

```
git clone https://github.com/farhad-mohammadi/pytse-filter
```

```
cd pytse_filter
```

```
pip install .
```

توجه: کتابخانه های وابسته به طور خودکار همراه با کتابخانه اصلی نصب می شوند.

چگونگی استفاده از کتابخانه pytse-filter

کتابخانه pytse-filter به دو بخش اصلی تقسیم می شود RealTime و History ، که هر دو برای مبتدیان و حرفه ای ها در بخش های مختلف مناسب هستند.

استفاده مبتدی از RealTime

کلاس RealTime را از کتابخانه وارد کرده و متن فیلتر را همانطور که در مثال های زیر نشان داده شده است، ارسال کنید:

```
from pytse_filter import RealTime
```

#مثال ۱: نمادهایی که در منفی قدرت خریدار بیشتر از فروشنده است

```
conditions = 'pl < py and power > 1'
```

```
df = RealTime().filter_by_text_condition(conditions)
```

```
print(df)
```

#اختیاری: نتیجه را در یک فایل اکسل ذخیره کنید

```
# df.to_excel('myfilter.xlsx')
```

#مثال ۲: نمادهایی با قدرت خریدار به فروشنده بالا و خرید سرانه بیشتر از ۳۰ میلیون تومان

```
conditions = 'power > 2 and buy_per_capita > 30'
```

```
df = RealTime().filter_by_text_condition(conditions)
```

```
print(df)
```

```
# df.to_excel('myfilter.xlsx')
```

#مثال ۳: نمادهایی که درصد خرید حقیقی بالا و نسبت فروش حقوقی زیاد است

```
conditions = 'ind_buy_ratio > 80 and cor_sell_ratio > 50'
```

```
df = RealTime().filter_by_text_condition(conditions)
print(df)
# df.to_excel('myfilter.xlsx')
```

استفاده پیشرفته از RealTime

برای کاربران پیشرفته، امکان پیگیری مداوم داده ها و نظارت بر تغییرات وضعیت خاص در نمادها وجود دارد:

```
from pytse_filter import RealTime
from time import sleep
import sys

market = RealTime()
sell_queue_symbols = market.filter_by_text_condition('pl == tmin')
if len(sell_queue_symbols) == 0 :
    print('صف فروشی یافت نشد')
    sys.exit()

sell_queue_symbols = sell_queue_symbols['symbol'].tolist()
print(sell_queue_symbols)

while True:
    sleep(10) # هر ۱۰ ثانیه چک کنید
    updated_sell_queue_symbols = market.filter_by_text_condition('pl == tmin')
    if len(updated_sell_queue_symbols) == 0 :
        continue
```

```

updated_sell_queue_symbols = updated_sell_queue_symbols['symbol'].tolist()
for sym in sell_queue_symbols:
    if sym not in updated_sell_queue_symbols:
        print(f'پاک شد {sym} صف فروش')
sell_queue_symbols = updated_sell_queue_symbols

```

همچنین امکان تعیین شرایط مختلف با استفاده از RealTimeCondition وجود دارد و می توانید با دریافت داده ها تنها یک بار و استفاده از آرگومان update_data=False، چندین شرط مختلف را به داده ها اعمال کنید.

```

from pytse_filter import RealTime, RealTimeCondition
import sys

```

```

buy_queue_condition = RealTimeCondition('pl == tmax')
sell_queue_condition = RealTimeCondition('pl == tmin')
positive_conditions = RealTimeCondition('pl < tmax and plp > 1')
negative_conditions = RealTimeCondition('pl > tmin and plp < -1')
zero_condition = RealTimeCondition('plp <= 1 and plp >= -1')

```

```

market_data = RealTime()
market_data.download()
if not market_data.download_status:
    print("can't download data")
    sys.exit()

```

```
market_data.condition = buy_queue_condition
```

```
buy_queue_symbols = market_data.filter_by_obj_condition(update_data= False)
```

```
market_data.condition = sell_queue_condition
```

```
sell_queue_symbols = market_data.filter_by_obj_condition(update_data= False)
```

```
market_data.condition = positive_conditions
```

```
positive_symbols = market_data.filter_by_obj_condition(update_data= False)
```

```
market_data.condition = negative_conditions
```

```
negative_symbols = market_data.filter_by_obj_condition(update_data= False)
```

```
market_data.condition = zero_condition
```

```
zero_symbols = market_data.filter_by_obj_condition(update_data= False)
```

```
symbols_total = len(buy_queue_symbols) + len(sell_queue_symbols) +  
len(positive_symbols) + len(negative_symbols) + len(zero_symbols)
```

```
msg = f"بازار در یک نگاه:"
```

از مجموع {symbols_total} نماد

{len(buy_queue_symbols)} نماد در صف خرید

{len(sell_queue_symbols)} نماد در صف فروش

{len(positive_symbols)} نماد در محدوده ی مثبت

{len(negative_symbols)} نماد در محدوده ی منفی

{len(zero_symbols)} نماد در محدوده ی صفر تابلو در حال معامله هستند."

print(msg) می‌توانید متن را به شبکه های اجتماعی خود ارسال کنید.

متدهای اضافی در کلاس RealTime

- get_all_stocks_data

این متد داده های جاری تمام نمادها را بازیابی میکند، محاسبات لازم را انجام میدهد و داده ها را به صورت یک DataFrame پاندااس برمی گرداند.

نحوه استفاده:

```
from pytse_filter import RealTime
```

#بازیابی و محاسبه دادهها برای تمام سهامها

```
df = RealTime().get_all_stocks_data(update_data=True)
```

پارامتر update_data تعیین میکند که آیا داده های جدید باید دانلود شوند (True) یا از داده های موجود استفاده شود.(False)

- get_stocks_data_as_list:

مشابه get_all_stocks_data، اما خروجی را به صورت لیستی از دیکشنری ها برمیگرداند که هر کدام دادههای یک نماد خاص را دربردارند.

نحوه استفاده:

```
from pytse_filter import RealTime
```

#دریافت دادهها برای تمام سهامها به صورت لیستی از دیکشنری ها

```
list_of_dict = RealTime().get_stocks_data_as_list(update_data=True)
```

استفاده از کلاس History

کلاس History در کتابخانه `pytse-filter` ابزاری قدرتمند برای تحلیل داده های تاریخی سهام است. این امکان را به شما می دهد تا اندیکاتورهای مختلفی را اعمال کرده و محاسباتی را روی خرید و فروشهای حقیقی و حقوقی انجام دهید.

دانلود داده ها

قبل از اعمال فیلترها، نیاز است تا دادهها برای تمام نمادها دانلود و پردازش شوند. این کار با استفاده از متد `download_summery` انجام می شود:

```
from pytse_filter import History
```

```
#این کار را بعد از ساعت ۸ شب اجرا کنید تا داده ها با آخرین اطلاعات بازار بروز شوند
```

```
History().download_summery()
```

بسته به سرعت اینترنت شما، این فرآیند می تواند بین ۲۰ دقیقه تا بیش از یک ساعت طول بکشد. با این حال، پس از بروزرسانی داده ها، آنها به صورت محلی ذخیره میشوند که این امکان را به شما میدهد تا به سرعت فیلترهای متعددی را اعمال کنید.

فیلتر کردن ابتدایی

برای مبتدیان، کلاس History راهی ساده برای فیلتر کردن نمادها بر اساس اندیکاتورها و الگوهای خرید/فروش ارائه می دهد:

```
from pytse_filter import History
```

```
#مثال: نمادهایی که RSI آنها کمتر از ۳۰ است و همگرایی میانگین متحرک مثبت دارند
```

```
conditions = 'rsi < 30 and macd > 0'
```

```
df = History().filter_by_text_condition(conditions)
```

```
print(df)
```

#اختیاری: نتیجه را در یک فایل اکسل ذخیره کنید

```
# df.to_excel('filtered_symbols.xlsx')
```

استفاده پیشرفته

کاربران پیشرفته می توانند اندیکاتورها و تنظیمات را در فایل `inds_setting.py` شخصی سازی کنند. همچنین می توانند از متد `download_history` برای بازیابی تاریخیچه هر نماد، اعمال شاخصها و محاسبات، و ذخیره نتایج استفاده کنند:

```
from pytse_filter import History
```

#فایل `inds_setting.py` را با شاخصها و تنظیمات دلخواه خود بروز کنید

#مثال: فیلتر کردن نمادهایی با شاخص جریان نقدی بالا (MFI) و عبور صعودی MACD از سیگنال

```
conditions = 'mfi > 80 and y_macd < y_signal and macd > signal'
```

```
df = History().filter_by_text_condition(conditions)
```

```
print(df)
```

#داده های فیلتر شده را برای تحلیلهای بیشتر ذخیره کنید

```
# df.to_excel('advanced_filter.xlsx')
```

ترکیب History و Realtime

برای تحلیل جامع، می توانید داده های زمان واقعی و تاریخی را ترکیب کنید. این به شما امکان می دهد تا نمادهایی که معیارهای تاریخی خاصی را برآورده می کنند را ردیابی کرده و عملکرد آنها را در زمان واقعی نظارت کنید:

```
from pytse_filter import History, RealTime
```

```
from time import sleep
```



```
# نمادها را بر اساس RSI تاریخی کمتر از ۴۰ پیش فیلتر کنید
historical_symbols = History().filter_by_text_condition('rsi < 40')
historical_symbols = list(historical_symbols.index)

# این نمادها را در زمان واقعی برای قدرت خریدار بیشتر از ۲ نظارت کنید
while True:
    sleep(10) # هر ۱۰ ثانیه چک کنید

    current_power_symbols = RealTime().filter_by_text_condition('power > 2')
    current_power_symbols = current_power_symbols['symbol'].tolist()
    for symbol in current_power_symbols:
        if symbol in historical_symbols:
            print(f'RSI {symbol} تاریخی > ۴۰ و قدرت خریدار فعلی < ۲ برای RSI')
```

این رویکرد به معامله گران امکان می دهد تا سیستمی را راه اندازی کنند که زمانی که نمادهای قوی تاریخی علاقه خرید قابل توجهی در بازار فعلی نشان می دهند، آنها را مطلع سازد و فرصتی بالقوه برای معامله فراهم کند.

متدهای اضافی در کلاس History

• download_history

این متد تاریخچه یک نماد را بازیابی می کند، محاسبات و تحلیلهای شاخصها را انجام میدهد و نتیجه را برمی گرداند. همچنین خروجی را به صورت یک فایل اکسل در پوشه history ذخیره می کند.

نحوه استفاده:

```
from pytse_filter import History
```

#دانلود و پردازش داده های تاریخی برای یک نماد

```
df = History().download_history(symbol='  
, length=-1, calc_inds=True, calc_client=True)
```

پارامترها:

- symbol: نام نماد
- length: تعداد نقاط داده ای که باید بازیابی شوند، -۱ نشاندهنده کل تاریخچه است
- calc_inds: (True یا False) آیا محاسبه اندیکاتورها انجام شود
- calc_client: (True یا False) آیا محاسبات حقیقی و حقوقی انجام شود

نوشتن شرایط

شرط یک رشته متنی است که معیارهای فیلتر کردن داده های سهام را تعریف می کند. از قوانین نحوی زیر استفاده کنید:

• از متغیرهایی که در فایل های RealTimeVariables.xlsx و HistoryVariable.xlsx لیست شده اند، استفاده کنید.

• از عملگرهای مقایسه ای استفاده کنید. >=, >, <=, <, !=, == :

• مقایسه ها را با استفاده از عملگرهای منطقی ترکیب کنید. and, or :

• برای تغییر اولویت عملگرها، مقایسه ها را با پرانتز گروه بندی کنید.

مجوز

پروژه pytse-filter تحت مجوز MIT است. برای جزئیات بیشتر به فایل LICENSE مراجعه کنید.

جدول مربوط به متغیرهای RealTime:

ردیف	متغیر	معنا
1	pf	اولین قیمت
2	pmin	کمترین قیمت
3	pmax	بیشترین قیمت
4	pl	آخرین قیمت
5	pc	قیمت پایانی
6	py	قیمت دیروز
7	tmin	آستانه ی پایین قیمت
8	tmax	آستانه ی بالای قیمت
9	pfp	درصد اولین قیمت
10	pminp	درصد کمترین قیمت
11	pmaxp	درصد بیشترین قیمت
12	plp	درصد آخرین قیمت
13	pcp	درصد قیمت پایانی
14	tminp	درصد آستانه ی پایین قیمت
15	tmaxp	درصد آستانه ی بالای قیمت
16	tno	تعداد معاملات
17	tvol	حجم معاملات
18	tval	ارزش معاملات
19	bvol	حجم مبنا
20	z	تعداد سهام
21	buy_i_count	تعداد خریداران حقیقی
22	buy_n_count	تعداد خریداران حقوقی
23	buy_i_volume	حجم خرید خریداران حقیقی
24	buy_n_volume	حجم خرید خریداران حقوقی
25	sell_i_count	تعداد فروشندگان حقیقی
26	sell_n_count	تعداد فروشندگان حقوقی
27	sell_i_volume	حجم فروش فروشندگان حقیقی
28	sell_n_volume	حجم فروش فروشندگان حقوقی
29	buy_i_value	ارزش خرید خریداران حقیقی
30	buy_n_value	ارزش خرید خریداران حقوقی
31	sell_i_value	ارزش فروش فروشندگان حقیقی
32	sell_n_value	ارزش فروش فروشندگان حقوقی
33	buy_per_capita	سرنه خرید به میلیون تومان

34	sell_per_capita	سرايه فروش به ميليون تومان
35	power	قدرت خريدار به فروشنده ي حقيقي
36	volume	حجم معاملات
37	ind_buy_ratio	درصد خريد خريداران حقيقي
38	ind_sell_ratio	درصد فروش فروشندگان حقيقي
39	cor_buy_ratio	درصد خريد خريداران حقوقی
40	cor_sell_ratio	درصد فروش فروشندگان حقوقی
41	money_flow	ورود يا خروج پول حقيقي
42	d1_value	ارزش صف اول خريد
43	o1_value	ارزش ثف اول فروش
44	zo1	تعداد فروشندگان صف اول
45	po1	قيمت فروش صف اول
46	qo1	حجم فروش صف اول
47	zd1	تعداد خريداران صف اول
48	pd1	قيمت خريد صف اول
49	qd1	حجم خريد صف اول
50	zo2	تعداد فروشندگان صف دوم
51	po2	قيمت فروش صف دوم
52	qo2	حجم فروش صف دوم
53	zd2	تعداد خريداران صف دوم
54	pd2	قيمت خريد صف دوم
55	qd2	حجم خريد صف دوم
56	zo3	تعداد فروشندگان صف سوم
57	po3	قيمت فروش صف سوم
58	qo3	حجم فروش صف سوم
59	zd3	تعداد خريداران صف سوم
60	pd3	قيمت خريد صف سوم
61	qd3	حجم خريد صف سوم
62	zo4	تعداد فروشندگان صف چهارم
63	po4	قيمت فروش صف چهارم
64	qo4	حجم فروش صف چهارم
65	zd4	تعداد خريداران صف چهارم
66	pd4	قيمت خريد صف چهارم
67	qd4	حجم خريد صف چهارم
68	zo5	تعداد فروشندگان صف پنجم
69	po5	قيمت فروش صف پنجم
70	qo5	حجم فروش صف پنجم
71	zd5	تعداد خريداران صف پنجم
72	pd5	قيمت خريد صف پنجم
73	qd5	حجم خريد صف پنجم

جدول مربوط به متغیرهای History:

معنا	متغیر	ردیف
اولین قیمت	open	1
کمترین قیمت	low	2
بیشترین قیمت	high	3
آخرین قیمت	close	4
قیمت پایانی	adj_close	5
حجم معاملات	volume	6
ارزش معاملات	value	7
تعداد معاملات	count	8
قیمت دیروز	yesterday_adj_close	9
حجم خرید حقیقی	buy_i_volume	10
حجم خرید حقوقی	buy_n_volume	11
ارزش خرید حقیقی	buy_i_value	12
ارزش خرید حقوقی	buy_n_value	13
تعداد خریدار حقوقی	buy_n_count	14
حجم فروش حقیقی	sell_i_volume	15
تعداد خریدار حقیقی	buy_i_count	16
حجم فروش حقوقی	sell_n_volume	17
ارزش فروش حقیقی	sell_i_value	18
ارزش فروش حقوقی	sell_n_value	19
تعداد فروشنده ی حقوقی	sell_n_count	20
تعداد فروشنده ی حقیقی	sell_i_count	21
مقدار RSI	rsi	22
مقدار MACD اندیکاتور MACD	macd	23
مقدار histogram اندیکاتور MACD	histogram	24
مقدار signal اندیکاتور MACD	signal	25
مقدار K اندیکاتور StochAstic	k	26
مقدار D اندیکاتور StochAstic	d	27
مقدار اندیکاتور MFI	mfi	28
مقدار اندیکاتور SMA 50 کندلی	sma50	29
مقدار اندیکاتور SMA 21 کندلی	sma21	30
مقدار اندیکاتور EMA 21 کندلی	ema21	31
مقدار K اندیکاتور StochRSI	rsi_k	32
مقدار D اندیکاتور StochRSI	rsi_d	33
خط پایین اندیکاتور BollingerBands	lower_band	34
خط میانی اندیکاتور BollingerBands	mid_band	35
خط بالای اندیکاتور BollingerBands	upper_band	36
عرض باند اندیکاتور BollingerBands	band_width	37
مقدار خط Span A اندیکاتور IchiMoku	spana	38
مقدار خط Span B اندیکاتور IchiMoku	spanb	39

40	tenkan	مقدار خط Tenkan اندیکاتور IchiMoku
41	kijun	مقدار خط Kijun اندیکاتور IchiMoku
42	future_spana	مقدار خط Span A آینده اندیکاتور IchiMoku
43	future_spanb	مقدار خط Span B آینده اندیکاتور IchiMoku
44	highest21	بیشترین قیمت در ۲۱ روز گذشته
45	highest63	بیشترین قیمت در ۶۳ روز گذشته
46	lowest21	کمترین قیمت در ۲۱ روز گذشته
47	lowest63	کمترین قیمت در ۶۳ روز گذشته
48	dis_from_lowest21	فاصله ی قیمت کنونی از کمترین قیمت ۲۱ کندل گذشته به درصد
49	dis_from_lowest63	فاصله ی قیمت کنونی از کمترین قیمت ۶۳ کندل گذشته به درصد
50	dis_from_highest21	فاصله ی قیمت کنونی از بیشترین قیمت ۲۱ کندل گذشته به درصد
51	dis_from_highest63	فاصله ی قیمت کنونی از بیشترین قیمت ۶۳ کندل گذشته به درصد
52	buy_per_capita	سرانه ی خرید حقیقی به میلیون تومان
53	sell_per_capita	سرانه فروش حقیقی به میلیون تومان
54	power	قدرت خریدار به فروشنده
55	mony_flow	ورود یا خروج پول حقیقی به میلیون تومان
56	sell_per_capita_avg10	میانگین ۱۰ روزه سرانه فروش حقیقی
57	buy_per_capita_avg10	میانگین ۱۰ روزه سرانه خرید حقیقی
58	power_avg10	میانگین ۱۰ روزه قدرت خریدار به فروشنده
59	mony_flow_total10	برآیند ورود و خروج پول حقیقی ۱۰ روزه
60	y_open	اولین قیمت روز قبل
61	y_low	کمترین قیمت روز قبل
62	y_high	بیشترین قیمت روز قبل
63	y_close	آخرین قیمت روز قبل
64	y_adj_close	قیمت پایانی روز قبل
65	y_volume	حجم معاملات روز قبل
66	y_value	ارزش معاملات روز قبل
67	y_count	تعداد معاملات روز قبل
68	y_yesterday_adj_close	قیمت دیروز روز قبل
69	y_buy_i_volume	حجم خرید حقیقی روز قبل
70	y_buy_n_volume	حجم خرید حقوقی روز قبل
71	y_buy_i_value	ارزش خرید حقیقی روز قبل
72	y_buy_n_value	ارزش خرید حقوقی روز قبل
73	y_buy_n_count	تعداد خریدار حقوقی روز قبل
74	y_sell_i_volume	حجم فروش حقیقی روز قبل
75	y_buy_i_count	تعداد خریدار حقیقی روز قبل
76	y_sell_n_volume	حجم فروش حقوقی روز قبل
77	y_sell_i_value	ارزش فروش حقیقی روز قبل
78	y_sell_n_value	ارزش فروش حقوقی روز قبل
79	y_sell_n_count	تعداد فروشنده ی حقوقی روز قبل
80	y_sell_i_count	تعداد فروشنده ی حقیقی روز قبل
81	y_rsi	مقدار RSI روز قبل

82	y_macd	مقدار MACD اندیکاتور MACD روز قبل
83	y_histogram	مقدار histogram اندیکاتور MACD روز قبل
84	y_signal	مقدار signal اندیکاتور MACD روز قبل
85	y_k	مقدار K اندیکاتور StochAstic روز قبل
86	y_d	مقدار D اندیکاتور StochAstic روز قبل
87	y_mfi	مقدار اندیکاتور MFI روز قبل
88	y_sma50	مقدار اندیکاتور SMA 50 کندلی روز قبل
89	y_sma21	مقدار اندیکاتور SMA 21 کندلی روز قبل
90	y_ema21	مقدار اندیکاتور EMA 21 کندلی روز قبل
91	y_rsi_k	مقدار K اندیکاتور StochRSI روز قبل
92	y_rsi_d	مقدار D اندیکاتور StochRSI روز قبل
93	y_lower_band	خط پایین اندیکاتور BollingerBands روز قبل
94	y_mid_band	خط میانی اندیکاتور BollingerBands روز قبل
95	y_upper_band	خط بالای اندیکاتور BollingerBands روز قبل
96	y_band_width	عرض باند اندیکاتور BollingerBands روز قبل
97	y_spana	مقدار خط Span A اندیکاتور IchiMoku روز قبل
98	y_spanb	مقدار خط Span B اندیکاتور IchiMoku روز قبل
99	y_tenkan	مقدار خط Tenkan اندیکاتور IchiMoku روز قبل
100	y_kijun	مقدار خط Kijun اندیکاتور IchiMoku روز قبل
101	y_future_spana	مقدار خط Span A آینده اندیکاتور IchiMoku روز قبل
102	y_future_spanb	مقدار خط Span B آینده اندیکاتور IchiMoku روز قبل
103	y_highest21	بیشترین قیمت در ۲۱ روز گذشته روز قبل
104	y_highest63	بیشترین قیمت در ۶۳ روز گذشته روز قبل
105	y_lowest21	کمترین قیمت در ۲۱ روز گذشته روز قبل
106	y_lowest63	کمترین قیمت در ۶۳ روز گذشته روز قبل
107	y_dis_from_lowest21	فاصله ی قیمت کنونی از کمترین قیمت ۲۱ کندل گذشته به درصد روز قبل
108	y_dis_from_lowest63	فاصله ی قیمت کنونی از کمترین قیمت ۶۳ کندل گذشته به درصد روز قبل
109	y_dis_from_highest21	فاصله ی قیمت کنونی از بیشترین قیمت ۲۱ کندل گذشته به درصد روز قبل
110	y_dis_from_highest63	فاصله ی قیمت کنونی از بیشترین قیمت ۶۳ کندل گذشته به درصد روز قبل
111	y_buy_per_capita	سرايه ی خريـد حقيقي به ميليون تومان روز قبل
112	y_sell_per_capita	سرايه فروش حقيقي به ميليون تومان روز قبل
113	y_power	قدرت خريدار به فروشنده روز قبل
114	y_mony_flow	ورود يا خروج پول حقيقي به ميليون تومان روز قبل
115	y_sell_per_capita_avg10	میانگین ۱۰ روزه سرايه فروش حقيقي روز قبل
116	y_buy_per_capita_avg10	میانگین ۱۰ روزه سرايه خريـد حقيقي روز قبل
117	y_power_avg10	میانگین ۱۰ روزه قدرت خريدار به فروشنده روز قبل
118	y_mony_flow_total10	برآيند ورود و خروج پول حقيقي ۱۰ روزه روز قبل