

## راهنمای کاربر برای pytse-filter

این راهنمای کاربر به شما کمک می کند تا از پروژه pytse-filter استفاده کنید، کتابخانه ای 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.

### پیشنیازها

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

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

### نصب:

شما میتوانید pytse-filter را با استفاده از روشهای 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 میشود **RealTime** و **History** ، که هر دو برای مبتدیان و حرفه ای ها در بخشهای مختلف مناسب هستند.

### استفاده مبتدی از **RealTime**

کلاس **RealTime** را از کتابخانه وارد کرده و متن فیلتر را همانطور که در مثالهای 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_of_demand > 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_of_demand > 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 not sell_queue_symbols:
    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 updated_sell_queue_symbols is None:
    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 داده‌ها تنها یک بار و استفاده از آرگومان update\_data=False ، چندین شرط مختلف را به داده‌ها 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 & plp > 1')
negative_conditions = RealTimeCondition('pl > tmin & plp < -1')
zero_condition = RealTimeCondition('plp <= 1 & 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.
```

## استفاده از کلاس History

کلاس History در کتابخانه pytse-filter ابزاری قدرتمند برای تحلیل داده های تاریخی سهام است. این امکان را به شما میدهد 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 داده ها برای تمام نمادها دانلود و پردازش شوند. این کار با استفاده از متد download\_summary 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_summary()
```

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

## فیلتر کردن ابتدایی

برای مبتدیان، کلاس History راهی ساده برای فیلتر کردن نمادها بر اساس شاخصهای فنی و الگوهای خرید/فروش ارائه میدهد:

```
from pytse_filter import History
```

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

```
conditions = 'rsi14 < 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 میتوانند اندیکاتورها و تنظیمات را در فایل inds\_setting.py شخصی سازی کنند. همچنین می توانید از متد download\_history برای بازیابی تاریخچه هر نماد، اعمال شاخصها و محاسبات، و ذخیره نتایج استفاده کنید:

```
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، میتوانید 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_of_demand > 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.

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:

متغیر	معنا
pf	اولین قیمت
pmin	کمترین قیمت
pmax	بیشترین قیمت
pl	آخرین قیمت
pc	قیمت پایانی
py	قیمت دیروز
tmin	آستانه ی پایین قیمت
tmax	آستانه ی بالای قیمت
pfp	درصد اولین قیمت
pminp	درصد کمترین قیمت
pmaxp	درصد بیشترین قیمت
plp	درصد آخرین قیمت
pcp	درصد قیمت پایانی
tminp	درصد آستانه ی پایین قیمت
tmaxp	درصد آستانه ی بالای قیمت
bvol	حجم مبنا
buy_i_count	تعداد خریداران حقیقی
buy_n_count	تعداد خریداران حقوقی
buy_i_volume	حجم خرید خریداران حقیقی
buy_n_volume	حجم خرید خریداران حقوقی
sell_i_count	تعداد فروشندگان حقیقی
sell_n_count	تعداد فروشندگان حقوقی
sell_i_volume	حجم فروش فروشندگان حقیقی
sell_n_volume	حجم فروش فروشندگان حقوقی
buy_i_value	ارزش خرید خریداران حقیقی
buy_n_value	ارزش خرید خریداران حقوقی
sell_i_value	ارزش فروش فروشندگان حقیقی
sell_n_value	ارزش فروش فروشندگان حقوقی
buy_per_capita	سرانه خرید به میلیون تومان
sell_per_capita	سرانه فروش به میلیون تومان
power_of_demand	قدرت خریدار به فروشنده ی حقیقی
volume	حجم معاملات
ind_buy_ratio	درصد خرید خریداران حقیقی
ind_sell_ratio	درصد فروش فروشندگان حقیقی
cor_buy_ratio	درصد خرید خریداران حقوقی
cor_sell_ratio	درصد فروش فروشندگان حقوقی
zo1	تعداد فروشندگان صف اول
po1	قیمت فروش صف اول

qo1	حجم فروش صف اول
zd1	تعداد خریداران صف اول
pd1	قیمت خرید صف اول
qd1	حجم خرید صف اول
zo2	تعداد فروشندگان صف دوم
po2	قیمت فروش صف دوم
qo2	حجم فروش صف دوم
zd2	تعداد خریداران صف دوم
pd2	قیمت خرید صف دوم
qd2	حجم خرید صف دوم
zo3	تعداد فروشندگان صف سوم
po3	قیمت فروش صف سوم
qo3	حجم فروش صف سوم
zd3	تعداد خریداران صف سوم
pd3	قیمت خرید صف سوم
qd3	حجم خرید صف سوم
zo4	تعداد فروشندگان صف چهارم
po4	قیمت فروش صف چهارم
qo4	حجم فروش صف چهارم
zd4	تعداد خریداران صف چهارم
pd4	قیمت خرید صف چهارم
qd4	حجم خرید صف چهارم
zo5	تعداد فروشندگان صف پنجم
po5	قیمت فروش صف پنجم
qo5	حجم فروش صف پنجم
zd5	تعداد خریداران صف پنجم
pd5	قیمت خرید صف پنجم
qd5	حجم خرید صف پنجم

## جدول مربوط به متغیرهای History

معنا	متغیر
اولین قیمت	open
کمترین قیمت	low
بیشترین قیمت	high
آخرین قیمت	close
قیمت پایانی	adj_close
حجم معاملات	volume
ارزش معاملات	value
تعداد معاملات	count
قیمت دیروز	yesterday_adj_close
حجم خرید حقیقی	buy_i_volume
حجم خرید حقوقی	buy_n_volume
ارزش خرید حقیقی	buy_i_value
ارزش خرید حقوقی	buy_n_value
تعداد خریدار حقوقی	buy_n_count
حجم فروش حقیقی	sell_i_volume
تعداد خریدار حقیقی	buy_i_count
حجم فروش حقوقی	sell_n_volume
ارزش فروش حقیقی	sell_i_value
ارزش فروش حقوقی	sell_n_value
تعداد فروشنده ی حقوقی	sell_n_count
تعداد فروشنده ی حقیقی	sell_i_count
مقدار RSI	rsi
مقدار MACD اندیکاتور MACD	macd
مقدار histogram اندیکاتور MACD	histogram
مقدار signal اندیکاتور MACD	signal
مقدار K اندیکاتور StochAstic	k
مقدار D اندیکاتور StochAstic	d
مقدار اندیکاتور MFI	mfi
مقدار اندیکاتور SMA 50 کندلی	sma50
مقدار اندیکاتور SMA 21 کندلی	sma21
مقدار اندیکاتور EMA 21 کندلی	ema21
مقدار K اندیکاتور StochRSI	rsi_k
مقدار D اندیکاتور StochRSI	rsi_d
خط پایین اندیکاتور BollingerBands	lower_band
خط میانی اندیکاتور BollingerBands	mid_band
خط بالای اندیکاتور BollingerBands	upper_band
عرض باند اندیکاتور BollingerBands	band_width
مقدار خط Span A اندیکاتور IchiMoku	spana

spanb	مقدار خط Span B اندیکاتور IchiMoku
tenkan	مقدار خط Tenkan اندیکاتور IchiMoku
kijun	مقدار خط Kijun اندیکاتور IchiMoku
future_spana	مقدار خط Span A آینده اندیکاتور IchiMoku
future_spanb	مقدار خط Span B آینده اندیکاتور IchiMoku
highest21	بیشترین قیمت در ۲۱ روز گذشته
highest63	بیشترین قیمت در ۶۳ روز گذشته
lowest21	کمترین قیمت در ۲۱ روز گذشته
lowest63	کمترین قیمت در ۶۳ روز گذشته
buy_per_capita	سرنانه ی خرید حقیقی به میلیون تومان
sell_per_capita	سرنانه فروش حقیقی به میلیون تومان
power_of_demand	قدرت خریدار به فروشنده
indivisual_mony_flow	ورود یا خروج پول حقیقی به میلیون تومان
sell_per_capita_avg10	میانگین ۱۰ روزه سرنانه فروش حقیقی
buy_per_capita_avg10	میانگین ۱۰ روزه سرنانه خرید حقیقی
power_of_demand_avg10	میانگین ۱۰ روزه قدرت خریدار به فروشنده
indivisual_mony_flow_total10	برآیند ورود و خروج پول حقیقی ۱۰ روزه
y_open	اولین قیمت روز قبل
y_low	کمترین قیمت روز قبل
y_high	بیشترین قیمت روز قبل
y_close	آخرین قیمت روز قبل
y_adj_close	قیمت پایانی روز قبل
y_volume	حجم معاملات روز قبل
y_value	ارزش معاملات روز قبل
y_count	تعداد معاملات روز قبل
y_yesterday_adj_close	قیمت دیروز روز قبل
y_buy_i_volume	حجم خرید حقیقی روز قبل
y_buy_n_volume	حجم خرید حقوقی روز قبل
y_buy_i_value	ارزش خرید حقیقی روز قبل
y_buy_n_value	ارزش خرید حقوقی روز قبل
y_buy_n_count	تعداد خریدار حقوقی روز قبل
y_sell_i_volume	حجم فروش حقیقی روز قبل
y_buy_i_count	تعداد خریدار حقیقی روز قبل
y_sell_n_volume	حجم فروش حقوقی روز قبل
y_sell_i_value	ارزش فروش حقیقی روز قبل
y_sell_n_value	ارزش فروش حقوقی روز قبل
y_sell_n_count	تعداد فروشنده ی حقوقی روز قبل
y_sell_i_count	تعداد فروشنده ی حقیقی روز قبل
y_rsi	مقدار RSI روز قبل
y_macd	مقدار MACD اندیکاتور MACD روز قبل
y_histogram	مقدار histogram اندیکاتور MACD روز قبل

y_signal	مقدار signal اندیکاتور MACD روز قبل
y_k	مقدار K اندیکاتور StochAstic روز قبل
y_d	مقدار D اندیکاتور StochAstic روز قبل
y_mfi	مقدار اندیکاتور MFI روز قبل
y_sma50	مقدار اندیکاتور SMA 50 کندلی روز قبل
y_sma21	مقدار اندیکاتور SMA 21 کندلی روز قبل
y_ema21	مقدار اندیکاتور EMA 21 کندلی روز قبل
y_rsi_k	مقدار K اندیکاتور StochRSI روز قبل
y_rsi_d	مقدار D اندیکاتور StochRSI روز قبل
y_lower_band	خط پایین اندیکاتور BollingerBands روز قبل
y_mid_band	خط میانی اندیکاتور BollingerBands روز قبل
y_upper_band	خط بالای اندیکاتور BollingerBands روز قبل
y_band_width	عرض باند اندیکاتور BollingerBands روز قبل
y_spana	مقدار خط Span A اندیکاتور IchiMoku روز قبل
y_spanb	مقدار خط Span B اندیکاتور IchiMoku روز قبل
y_tenkan	مقدار خط Tenkan اندیکاتور IchiMoku روز قبل
y_kijun	مقدار خط Kijun اندیکاتور IchiMoku روز قبل
y_future_spana	مقدار خط Span A آینده اندیکاتور IchiMoku روز قبل
y_future_spanb	مقدار خط Span B آینده اندیکاتور IchiMoku روز قبل
y_highest21	بیشترین قیمت در ۲۱ روز گذشته روز قبل
y_highest63	بیشترین قیمت در ۶۳ روز گذشته روز قبل
y_lowest21	کمترین قیمت در ۲۱ روز گذشته روز قبل
y_lowest63	کمترین قیمت در ۶۳ روز گذشته روز قبل
y_buy_per_capita	سرايه ی خرید حقیقی به میلیون تومان روز قبل
y_sell_per_capita	سرايه فروش حقیقی به میلیون تومان روز قبل
y_power_of_demand	قدرت خریدار به فروشنده روز قبل
y_indivisual_mony_flow	ورود یا خروج پول حقیقی به میلیون تومان روز قبل
y_sell_per_capita_avg10	میانگین ۱۰ روزه سرايه فروش حقیقی روز قبل
y_buy_per_capita_avg10	میانگین ۱۰ روزه سرايه خرید حقیقی روز قبل
y_power_of_demand_avg10	میانگین ۱۰ روزه قدرت خریدار به فروشنده روز قبل
y_indivisual_mony_flow_total10	برآیند ورود و خروج پول حقیقی ۱۰ روزه روز قبل