

Хабр



КАК СТАТЬ АВТОРОМ



Зарплаты IT-специалистов



Войти



empenoso

11 мар в 05:26

Анализ торговых стратегий для акций Мосбиржи на дневных интервалах с помощью Python

Средний

6 мин

5.8K

Open source*, Финансы в IT, Python*

Аналитика

Торговля акциями требует гибкости, особенно когда речь идет о тестировании стратегий технического анализа на прошлых данных. Я выбрал Python и библиотеки `backtesting.py` и `aiomex`, потому что они позволяют анализировать рынок без сложных платформ и ограничений. Python дает свободу автоматизации, `backtesting.py` обеспечивает удобный и быстрый механизм тестирования стратегий, а `aiomex` позволяет скачивать данные напрямую с Московской биржи без привязки к брокеру.

Важно, что `backtesting.py` получил обновление после четырех лет без обновлений, что делает его актуальным инструментом. И в отличие от MetaTrader, StockSharp, TSLab и Quik, которые работают с Московской биржей, но требуют Windows, если брокер имеет API, то можно запускать скрипт на любом сервере, включая облачные решения и Raspberry Pi.

В этой статье я протестирую самую свежую стратегию теханализа Джона Ф. Элерса (John Ehlers), направленную на устранение запаздывания скользящей средней. Разберемся, как её адаптировать к акциям Московской биржи и протестировать с помощью Python.

Новый индикатор Джона Элерса «устранение запаздывания скользящей средней»

Одна из главных проблем стандартных скользящих средних (SMA) — это запаздывание. Поскольку SMA рассчитывается как среднее за определенный период, её значение всегда отстает от реальной цены, что мешает своевременному входу в сделку.

Джон Элерс предложил решение — прогнозируемая скользящая средняя (PMA, Projected Moving Average). В отличие от обычных скользящих, PMA использует линейную регрессию

РЕКЛАМА



Получи грант за код

Конкурс open source проектов



Формула PMA:

$$PMA = SMA + Slope * Length / 2,$$

где Slope — наклон линии регрессии.

Дополнительно Элсэрс предложил прогнозировать саму PMA:

$$PredictPMA = PMA + 0.5 (Slope - Slope[2]) Length$$

и наклон:

$$PredictSlope = 1.5 Slope - 0.5 Slope[4].$$

Пересечения PredictPMA и PMA помогают находить точки входа и выхода, делая стратегию более адаптивной к изменениям рынка.



Получи грант за код

Конкурс open source проектов

Стратегия на основе индикатора РМА Джона Ф. Элерса

Вход в длинную позицию:

- Цена закрытия на недельном графике выше 50-недельной РМА.
- Цена закрытия на дневном графике выше 50-дневной РМА.
- 10-дневная РМА выше 50-дневной.

Риск-менеджмент:

- Первоначальный стоп-лосс устанавливается на 10% ниже цены входа.
- Выход из позиции осуществляется по скользящему стопу на основе ATR.

Реализация бэктестинга через `backtesting.py`. Определение топ-20 акций по объему

Весь код [представлен на GitHub](#).

Модуль `data_loader.py`

Для тестирования стратегии необходимо загружать актуальные данные о торгах.

В этом помогает библиотека `aiomex`, которая предоставляет API-доступ к Московской бирже. В модуле `data_loader.py` реализована функция `fetch_moex_data`,



Получи грант за код

Конкурс open source проектов

Функция запрашивает данные за последние 1825 дней (примерно 5 лет) и конвертирует их в формат Pandas DataFrame. Особенность реализации — использование асинхронного HTTP-клиента `aiohttp`, что ускоряет загрузку. Данные приводятся к удобному формату: преобразуются даты, устанавливается индекс, а названия колонок заменяются на стандартные для анализа.

Фильтрация ликвидных бумаг для тестирования. Модуль `scanner.py`

После загрузки данных важно отобрать ликвидные бумаги. Для этого в модуле `scanner.py` реализована функция `get_top_20_stocks`, которая анализирует объем торгов за последние 14 дней и выделяет 20 наиболее ликвидных акций.

Алгоритм работы следующий:

1. Получение списка всех торгуемых акций на основном рынке (TQBR) через API Московской биржи.
2. Асинхронная загрузка дневных данных по каждому инструменту с помощью `fetch_moex_data`.
3. Расчет суммарного объема торгов за 14 дней.
4. Формирование списка из 20 акций с наибольшим объемом



Получи грант за код

Конкурс open source проектов

Таким образом, отбираются бумаги с высоким оборотом, что повышает надежность тестирования стратегии и снижает риск торговли неликвидными активами.

Реализация бэктестинга через `backtesting.py`. Тестирование стратегии на исторических данных

Зачем всё разделил на модули?

Разделение кода на модули делает его более удобным для сопровождения, масштабирования и переиспользования. В нашем случае:

- `data_loader.py` отвечает за загрузку данных с Московской биржи.
- `scanner.py` фильтрует ликвидные бумаги.
- `backtester.py` выполняет бэктестинг.
- `strategy.py` содержит описание стратегии.
- `main.py` запускает сканирование и тестирование

Такой подход позволяет независимо модифицировать и тестировать каждый компонент системы.

Пример кода для бэктестинга с использованием `backtesting.py`

```
import asyncio
import pandas as pd
from backtesting import Backtest
from data_loader import fetch_moex_data
from strategy import LongOnlyPMAMultiTimeframeATRTrailingStop

async def run_backtest(ticker):
    print(f"\n{'='*50}")
    print(f"🚀 Запуск бэктеста для {ticker}")
    print(f"\n{'='*50}\n")

    # Получаем данные
    df_start_str_end_str = await fetch_moex_data(ticker) # Получаем start_str и end_s

    print(🚀 Запуск бэктеста... )
```



Получи грант за код

Конкурс open source проектов

```

strategy_class = LongOnlyPMAMultiTimeframeATRTrailingStop # Класс стратегии остается
strategy_name = f"{ticker}_{start_str}_{end_str}_LongOnlyPMAMultiTimeframeATRTrailingStop"
DynamicStrategyClass = type(strategy_name, (strategy_class,), {}) # Создаем динамический класс
bt = Backtest(df, DynamicStrategyClass, cash=100_000, commission=0.002) # Используем Backtest
stats = bt.run()

# Вывод результатов
print("\n📊 Результаты бэктеста:")
print(f"⚙️ Стратегия: {strategy_name}") # Выводим динамическое имя стратегии
print(f"📅 Период тестирования: с {stats['Start']} по {stats['End']}")
print(f"💰 Начальный капитал: 100,000 руб.")
print(f"📈 Конечный капитал: {stats['Equity Final ($)']:.2f} руб.")
print(f"📊 Общая доходность: {stats['Return (%)']:.2f}%")
print(f"📈 Годовая доходность: {stats['Return (Ann.) (%)']:.2f}%")
print(f"📊 Коэффициент Шарпа: {stats['Sharpe Ratio']:.2f}")
print(f"📉 Максимальная просадка: {stats['Max. Drawdown (%)']:.2f}%")
print(f"🔄 Количество сделок: {stats['# Trades']}")
print(f"✅ Процент выигрышных сделок: {stats['Win Rate (%)']:.2f}%")
print(f"👉 Лучшая сделка: +{stats['Best Trade (%)']:.2f}%")
print(f"👈 Худшая сделка: {stats['Worst Trade (%)']:.2f}%")
print(f"🕒 Средняя продолжительность сделки: {stats['Avg. Trade Duration']}")

# Построение графика
print("\n📊 Построение графика результатов...")
try:
    bt.plot()
    print("✅ График успешно построен!")
except ValueError as e:
    print(f"❌ Ошибка при построении графика: {e}")
print(f"\n{'='*50}")
print(f"🏁 Бэктест для {ticker} завершен")
print(f"\n{'='*50}\n")
return stats

```

Основные метрики оценки

Для оценки стратегии используются ключевые метрики:

- **Общая доходность (%)** — показывает, сколько стратегия заработала за весь период



Получи грант за код

Конкурс open source проектов

- **Коэффициент Шарпа** — измеряет соотношение доходности к риску.
- **Максимальная просадка (%)** — определяет максимальную потерю капитала.
- **Процент выигрышных сделок.**
- **Средняя продолжительность сделки.**

Эти показатели позволяют оценить эффективность стратегии и принять решение о её использовании в реальной торговле.

Результаты тестирования на акциях Московской биржи

Как положительные, так и отрицательные. Примеры [в html файлах на GitHub'e](#).

СПБ Биржа (тикер SPBE):

Результаты бэктеста:

Стратегия: SPBE_2020-03-03_2025-03-02_LongOnlyPMAMultiTimeframeATRTrailingStop

Период тестирования: с 2021-11-19 00:00:00 по 2025-03-01 00:00:00

Начальный капитал: 100,000 руб.

Конечный капитал: 349138.97 руб.

Общая доходность: 249.14%



Получи грант за код

Конкурс open source проектов

максимальная просадка: -27.41%

Количество сделок: 11
Процент выигрышных сделок: 54.55%
Лучшая сделка: +36.56%
Худшая сделка: -8.31%
Средняя продолжительность сделки: 25 days 00:00:00

Новатэк ао (тикер NVTK):

Результаты бэктеста:
Стратегия: NVTK_2020-03-03_2025-03-02_LongOnlyPMAMultiTimeframeATRTrailingStop
Период тестирования: с 2020-03-03 00:00:00 по 2025-03-01 00:00:00
Начальный капитал: 100,000 руб.
Конечный капитал: 94443.56 руб.
Общая доходность: -5.56%
Годовая доходность: -1.15%
Коэффициент Шарпа: -0.07
Максимальная просадка: -38.70%
Количество сделок: 22
Процент выигрышных сделок: 27.27%
Лучшая сделка: +18.55%
Худшая сделка: -10.78%



Получи грант за код

Конкурс open source проектов

Проблема учета смены лидеров по объему

Мой код отдельно тестирует каждую акцию из топ-20 на момент отбора (на сегодня). Однако он не учитывает смену лидеров по объему и не позволяет работать с единой корзиной акций, где позиции могут удерживаться даже после выпадения бумаги из топ-20. Это важно, потому что иначе стратегия теряет контекст уже открытых сделок.

Решение - создание скользящего портфеля, учитывающего смену лидеров - это стратегия, при которой состав портфеля регулярно пересматривается и обновляется на основе новых данных.

Следующие шаги

Фиксированный список топ-акций по объему устаревает. Использование динамического реестра позволит оперативно учитывать смену лидеров, корректируя состав активных позиций в стратегии.

Библиотека `ta-lib` мне не очень понравилась из-за сложностей с установкой - проще переписать индикатор вручную в будущем.

Получится ли реализовать это через `backtesting.py` ? Скорее всего вряд ли.

Скорее всего придётся вернуться к `Backtrader` .

Заключение

Тестирование стратегии на акциях Мосбиржи показало её стабильную эффективность при использовании индикатора РМА на дневных свечах.

Python доказал свою ценность в алгоритмической торговле, обеспечивая гибкость и автоматизацию. Однако `backtesting.py` имеет ограничения.

Автор: Михаил Шардин

 [Моя онлайн-визитка](#)

 [Telegram «Умный Дом Инвестора»](#)

11 марта 2025 г.



Получи грант за код

Конкурс open source проектов

Хэбы: Open source, Финансы в IT, Python

Редакторский дайджест



Присылаем лучшие статьи раз в месяц



Оставляя свою почту, я принимаю [Политику конфиденциальности](#) и даю согласие на получение рассылок

**212****63.3**

Карма

Рейтинг

Михаил Шардин @empenoso

Автоматизация / Данные / Финансы / Умные дома

[Подписаться](#)[Сайт](#) [Сайт](#) [GitHub](#)**Получи грант за код**

Конкурс open source проектов

 Комментарии 16

Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ

**Tirarex**

14 часов назад

Как я делал сеть на 2,5 гигабита с минимальным бюджетом — апгрейд, доступный каждому



Простой



9 мин



11K

Тutorial



+41



83



34

**Erwinmal**

18 часов назад

Сэндвич, сэр? История британских бутербродов от аристократических салонов до вокзальных буфетов



Простой



13 мин



3K

Ретроспектива



+41



20



2

**oneastok**

19 часов назад

Умное зеркало на Raspberry Pi: пошаговое руководство



Простой



4 мин



4.5K

**Получи грант за код**

Конкурс open source проектов

**iLushkersky**

14 часов назад

Жизнь на Марсе? (снова)

**Простой**

3 мин



2.3K

**+17**

5



10

**TrexSelectel**

16 часов назад

Nintendo Virtual Boy: неожиданное возрождение виртуальной реальности из 90-х



5 мин



1.1K

**+14**

3



3

**mio_anni**

19 часов назад

От мини-ЭВМ и перфокарт к IDE и фреймворкам. Как поменялось программирование за 50 лет — взгляд изнутри



12 мин



1.9K

**+12**

15



35

**RED_OS_M**

18 часов назад

Станислав Петров: «Ключевые отличия РЕД ОС М от Android – вовсе не в интерфейсе»

**Средний**

8 мин



6.5K

Интервью

**+10**

10



43

**Albert Wesker****Получи грант за код**

Конкурс open source проектов

 Средний  11 мин  1.5K

Обзор

Перевод

 +9

 16

 0



kilokanat

5 часов назад

Механическая клавиатура LARKeyboard

 Простой  5 мин  508

Тutorial

 +8

 4

 2



beeline_cloud

10 часов назад

Научный «дипфейк»? Как галлюцинации нейросетей — и другие проблемы — просачиваются в академические статьи

 Простой  8 мин  885

Аналитика

 +8

 11

 2

«Раньше играл в Танки, сейчас веду кибертурниры» — истории современных студентов

Турбо

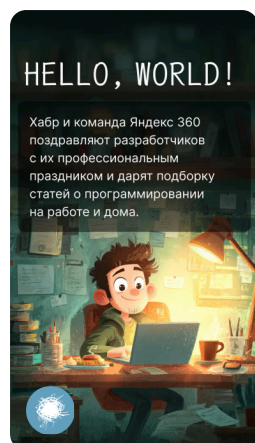
Показать еще

ИСТОРИИ

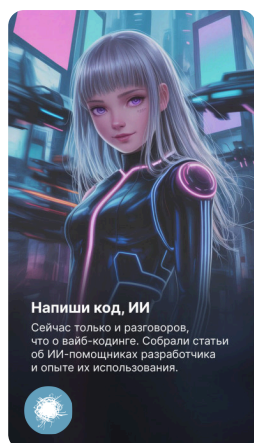


Получи грант за код

Конкурс open source проектов



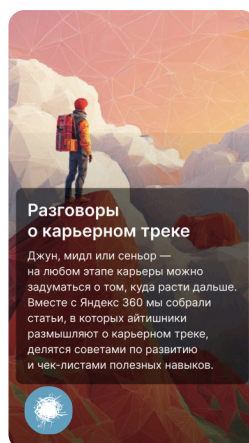
Чай, торт и код: с Днём программиста!



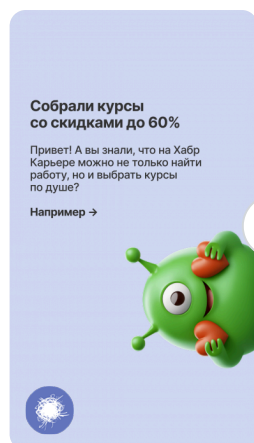
Made in AI



Чего хотят лиды в бигтехе?



Как расти в IT: советы, гайды и опыт сеньоров

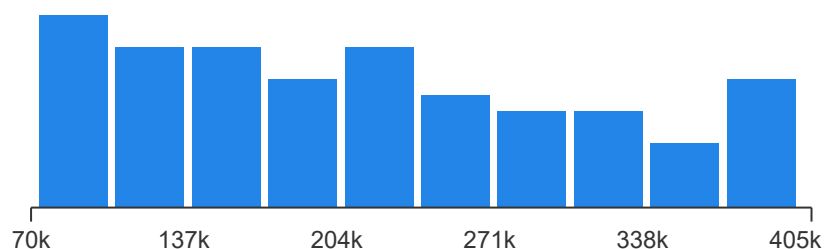


Курсы со скидками до 60%

СРЕДНЯЯ ЗАРПЛАТА В IT

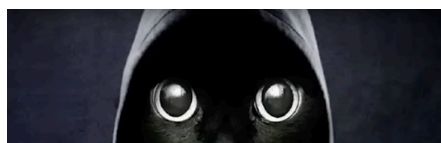
214 724 ₽/мес.

— средняя зарплата во всех IT-специализациях по данным из 27 443 анкет, за 2-ое пол. 2025 года. Проверьте «в рынке» ли ваша зарплата или нет!



[Проверить свою зарплату](#)

МИНУТОЧКУ ВНИМАНИЯ



Получи грант за код

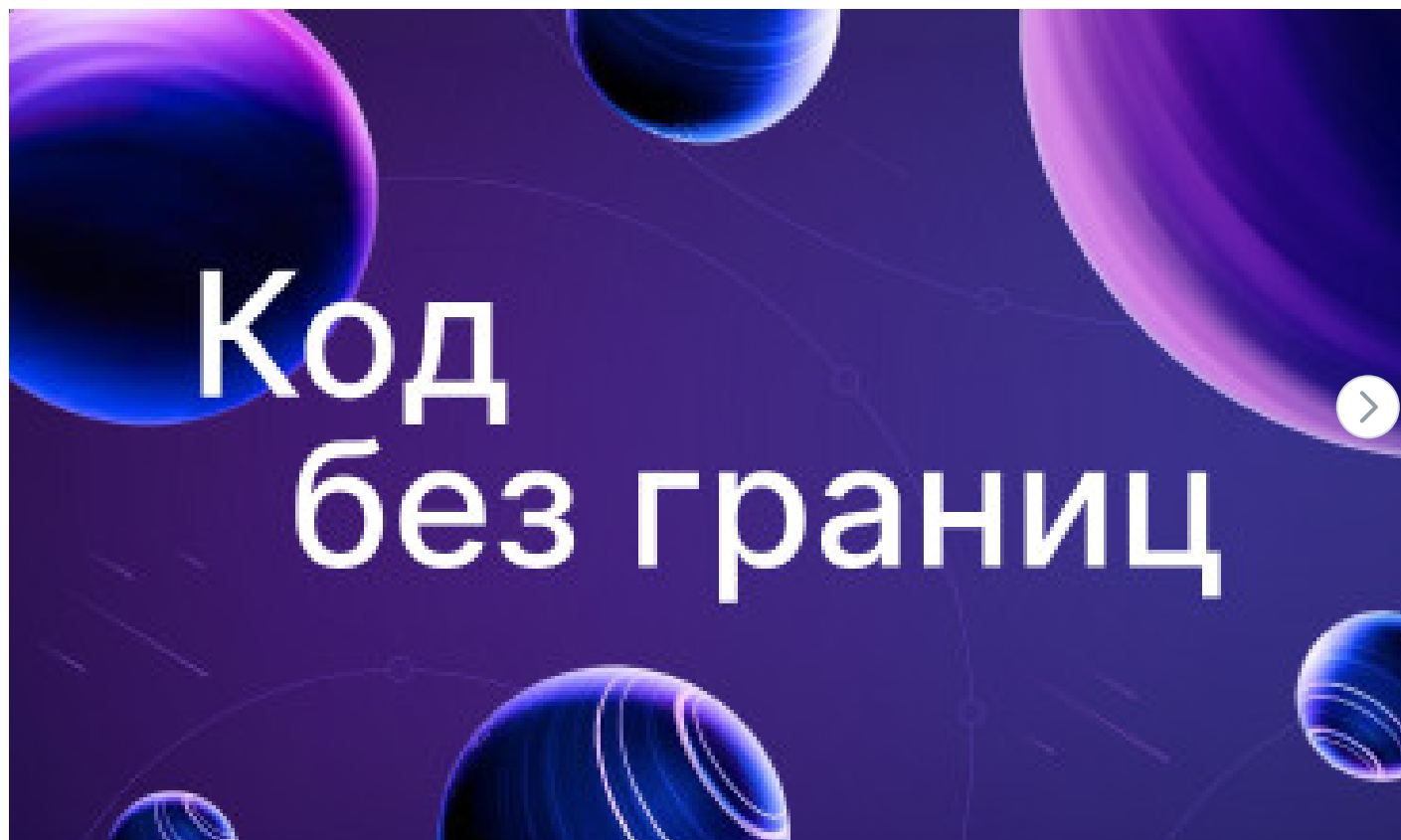
Конкурс open source проектов

Прямая трансляция IT Elements:
ключевые доклады

Где обучают вести
кибертурниры и разрабатывать
игры

Посмотри в Календарь, вдруг
сегодня есть мероприятие?

БЛИЖАЙШИЕ СОБЫТИЯ



3 сентября – 31 октября

Программа грантов для развития open source проектов «Код без грани

Онлайн

Разработка

Больше событий в календаре



Получи грант за код

Конкурс open source проектов



 [Настройка языка](#)

[Техническая поддержка](#)

© 2006–2025, Habr



Получи грант за код

Конкурс open source проектов