

Документация проекта

AI24: “Биржевая и внебиржевая торговля, оптимизация портфеля”

(Команда 31)

Техническая часть проекта представляет собой приложение, разработанное с использованием FastAPI и Streamlit для построения и управления моделями прогнозирования временных рядов.

Описание работы FastAPI:

Основной сервер реализует API для обучения и предсказания моделей временных рядов, обрабатывает запросы и отвечает в формате JSON, производит логирование: вся активность сервера логируется в файлы с помощью logging и RotatingFileHandler (логи хранятся в директории logs). Хранение и управление обученными моделями осуществляется через структуру MODELS Поддерживаемые технологии: ARIMA, SARIMA, VAR для анализа временных рядов, загрузка данных из Yahoo Finance с помощью библиотеки yfinance.

- Основные эндпоинты:

POST	/fit	Fit	▼
POST	/fit_yahoo	Fit Yahoo	▼
POST	/predict	Predict	▼
GET	/models	Get Models	▼
POST	/set_model	Set Model	▼
POST	/delete_all_models	Delete All Models	▼
POST	/upload_dataset	Upload Dataset	▼

	Эндпоинт	Описание и запрос	Ответ
POST	/fit – Обучение модели на пользовательских данных.	{ "data": [1.2, 3.4, 5.6], "model_type": "ARIMA", "parameters": {"order": [1, 1, 1]} }	ID модели, статус, и сводка модели.

POST	/fit_yahoo – Обучение моделей ARIMA, SARIMA или VAR на данных из Yahoo Finance.	{ "ticker": "AAPL", "period": "1y", "model_type": "ARIMA", "parameters": {"order": [p, d, q]} }	ID модели, статус, и сводка модели.
POST	/predict – Получение предсказаний для заданной модели.	{ "model_id": "model_1", "steps": 10 }	Список предсказанных значений.
GET	/models – Получение списка всех моделей.	–	ID моделей, тип и статус.
POST	/set_model – Установка активной модели.	{ "model_id": "model_1" }	Статус выполнения.
POST	/delete_all_models – Удаление всех сохраненных моделей.	–	Статус выполнения.
POST	/upload_dataset – Загрузка пользовательского датасета в формате CSV.	–	Первые 10 строк данных.

Для обучения моделей реализованы функции:

- train_arima(data, parameters) — обучает модель ARIMA.
- train_sarima(data, parameters) — обучает модель SARIMA.
- auto_arima_search(data, seasonal) — автоматический подбор гиперпараметров для SARIMA или ARIMA.
- train_var(df) — обучает модель VAR с использованием данных, загруженных с Yahoo Finance.

Есть несколько функций для загрузки данных:

- `load_data_from_yahoo_for_close(ticker, period)` — загружает данные о ценах закрытия для тикера с Yahoo Finance.
- `load_data_from_yahoo_for_var(ticker, period)` — загружает данные для модели VAR (включает дополнительные колонки, такие как "Volume", "High-Low").

Описание работы Streamlit:

В пользовательском интерфейсе есть кнопки для выбора популярных тикеров акций (например, AAPL, MSFT, TSLA). Также можно ввести собственный тикер в текстовое поле.

Выберите тикер актива



Или введите свой тикер:

Для выбранного тикера с помощью библиотеки `yfinance` загружается информация о компании (например, название, сектор, биржа, рыночная капитализация и текущая цена). График цен строится с использованием библиотеки `plotly`. Можно выбрать период и интервал данных для анализа (например, 1 месяц, 1 год и т.д.).

График цены для MSFT



Отображаются последние новости по выбранному тикеру, которые получаются через

NewsAPI.

Последние новости о NVDA:

[NVIDIA Stock Slips 2% Despite Analysts Remain Bullish on AI Giant's Future](#)

Yahoo Entertainment | Опубликовано: 2024-12-31T18:22:23Z

Описание: Low Volume Day for NVIDIA Stock as Analysts Boost Price Targets

[Jim Cramer's Take on NVIDIA \(NVDA\): The King of AI GPUs Under Pressure](#)

Yahoo Entertainment | Опубликовано: 2024-12-31T18:16:57Z

Описание: We recently published a list of Jim Cramer's Bold Predictions About These 9 Semiconductor Stocks. In this article, we are going to take a look at where...

[Nvidia Wraps Up 2024 as S&P 500's Top Performer, Eyes AI Expansion in 2025](#)

Yahoo Entertainment | Опубликовано: 2024-12-31T17:53:25Z

Описание: Nvidia was the third-best S&P 500 performance in 2024, after Palantir and Vistra Corp.

```
financial_domains = (  
    "bloomberg.com,cnbc.com,reuters.com,wsj.com,"  
    "marketwatch.com,ft.com,yahoo.com,forbes.com,investopedia.com"  
)
```

Также доступна визуализация, связанная с техническим анализом:

Отображаются графики с японскими свечами для выбранного тикера, строятся линии Боллинджера (верхняя и нижняя), при наличии данных строится график сезонности и остатков временного ряда с использованием seasonal_decompose.



Вторая вкладка посвящена прогнозированию цен акций с использованием функционала, который был реализован на этапе создания FastAPI.

На основе выбранного тикера строится график исторической цены. Отображаются графики автокорреляции (ACF) и частичной автокорреляции (PACF) для выбранного временного ряда с помощью matplotlib.



Результаты теста на стационарность (ADF)

p-value: 0.8102438721254295

Временной ряд нестационарен.

Выполняется тест на стационарность (ADF-тест), и выводится результат p-value. Если p-value меньше 0.05, временной ряд считается стационарным.

Пользователь может выбрать тип модели (ARIMA, SARIMA, VAR) и настроить параметры модели (например, порядок ARIMA, сезонность для SARIMA). При запуске модели происходит отправка запроса на сервер для обучения модели, а затем на сервер отправляется запрос для получения прогноза.

Тип модели

ARIMA

ARIMA

SARIMA

VAR

Прогнозируется будущее значение цен на основе выбранной модели и отображается график прогноза с использованием plotly. Пользователь может увидеть все

сохраненные модели, их summary, а также удалить все модели через интерфейс.

Все запросы на обучение моделей и прогнозирование делаются через API с использованием requests. Это позволяет интегрировать внешнюю модель для прогнозирования и получать результаты. Приложение динамично обновляет информацию и графики в зависимости от выбора пользователя.

Краткая инструкция по использованию:

Перед тем как запустить проект, убедитесь, что у вас установлены:

- Docker
- Docker Compose

Клонируйте репозиторий с помощью команды

```
git clone https://github.com/pAndrey200/portfolio-optimization.git
```

Затем перейдите в директорию проекта:

```
cd your-repo
```

Соберите и запустите контейнеры с помощью Docker Compose:

```
docker-compose up --build
```

После успешного запуска:

FastAPI будет доступен по адресу: <http://5.187.3.156:8000>

Streamlit будет доступен по адресу: <http://5.187.3.156:8501>

FastAPI:

Перейдите по адресу <http://5.187.3.156:8000/docs>, чтобы увидеть автоматически сгенерированную документацию Swagger.

Используйте доступные эндпоинты для обработки данных и обучения моделей.

Streamlit:

Перейдите по адресу <http://5.187.3.156:8501>, чтобы открыть интерфейс для визуализации данных и прогнозирования.

Чтобы остановить контейнеры, используйте команду:

```
docker-compose down
```

Для обновления образов после изменений в коде выполните:

```
docker-compose up --build
```


