StockAdvisor

Структура проекта ∂

- /app:
 - · Содержит frontend и backend компоненты:
 - backend.py: FastAPI endpoints для управления моделями и прогнозами.
 - frontend.py: Приложение Streamlit для взаимодействия с пользователем.
 - Вспомогательные модули для обработки данных, логирования и реализации моделей.
- · /checkpoints:
 - Содержит старые файлы и наработки из предыдущих этапов проекта
- . Idatacate
 - Директория для хранения исходных и обработанных данных.
- . Idocker
 - Конфигурация Docker для сборки и запуска контейнеров:
 - backend.Dockerfile, frontend.Dockerfile, dockercompose.vaml.
- /log
 - Файлы логов для мониторинга работы приложения (`StockAdvisor.log`).

Установка и запуск ⊘

Требования 🔗

• Убедитесь, что на вашем компьютере установлены Docker и Docker Compose.

Сборка и запуск контейнеров 🔗

- Клонируйте репозиторий:
- 1 git clone <URL_penoзитория> 2 cd StockAdvisor
- Выполните сборку и запуск контейнеров:
- 1 docker-compose up --build
- После запуска:
 - Backend доступен по адресу http://localhost:8080
 - Frontend доступен по адресу http://localhost:8501

Использование 🔗

Основные зависимости 🔗

Убедитесь, что установлены зависимости из файла requirements.txt:

1 pip install -r requirements.txt

Запуск локально 🔗

Backend

1 uvicorn backend:app --reload --host 0.0.0.0 --port 8080

rontend:

1 streamlit run frontend.py

Логи и данные 🔗

- Логи: ./log/StockAdvisor.log
- Данные: ./datasets

Инициализация приложения &

Общая информация 🔗

На текущий момент функционал приложения сосредоточен на использовании АRIMA-моделей для анализа временных рядов и прогнозирования.

- Moдели SimpleExpSmoothing и LinearRegression находятся в стадии подготовки к реализации и пока не включены в текущую версию.
- Причины:
- а. Во время планирования МVP произошли изменения в приоритетах и сроках разработки, что повлияло на выбор последовательности внедрения функционала.
- b. На этапе базового анализа (baseline) эти модели показали ограниченную ценность в сравнении с ARIMA, и их дальнейшее развитие было решено отложить.

Описание эндпоинтов (актуальное для текущего MVP) 🔗

1. /models (GET) 🔗

Описание: Возвращает список всех моделей (на текущий момент только ARIMA), которые были созданы или обучены в процессе работы приложения.

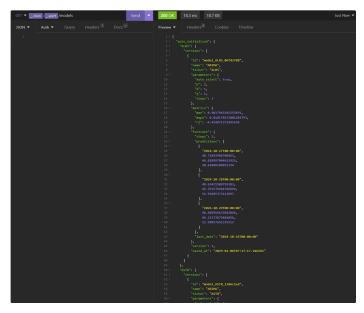
Пример запроса:

```
1 curl --request GET \
2 --url http://localhost:8080/models \
3 --header 'Content-Type: application/json'
```

Структура ответа (JSON):

```
// Название группы
      "group_name": {
 3
        "ticker_name": {
                                              // Название тикера
          "versions": [
                                             // Список всех версий модели для данного тикера
 5
             "id": "string",
                                             // Уникальный идентификатор модели
 6
 7
              "name": "string",
                                             // Название молели ("ARTMA")
 8
              "ticker": "string",
                                             // Название тикера
                                              // Параметры молели
 9
              "parameters": {
10
                "p": "integer",
                                              // Параметр р для модели
11
               "d": "integer",
                                             // Параметр d для модели
12
                                             // Параметр q для модели
13
                "steps": "integer"
                                             // Количество шагов прогноза
14
15
              "metrics": {
                                             // Метрики качества модели
                "mae": "float",
16
                                             // Средняя абсолютная ошибка
17
                "mape": "float",
                                             // Средняя абсолютная процентная ошибка
                "r2": "float"
18
                                             // Коэффициент детерминации (R2)
19
20
               "forecast": {
                                              // Информация о прогнозах
21
                "steps": "integer",
                                              // Количество шагов прогноза
22
                "predictions": [
                                              // Прогнозы по дням
                    "string",
24
                                              // Дата прогноза в формате YYYY-MM-DD
25
                    "float",
                                              // Предсказанное значение
                    "float",
26
                                              // Нижняя граница доверительного интервала
```

```
27
                  "float"
                                          // Верхняя граница доверительного интервала
28
                1
29
               1,
30
               "last_date": "string"
                                          // Последняя дата данных, использованных для обучения
31
32
             "version": "integer",
                                          // Номер версии модели
33
            "saved_at": "string"
                                          // Дата и время сохранения модели (ISO 8601)
34
35
36
      }
37
     }
38 }
39
```



2. /fit (POST) 🔗

Описание: Обучает временную модель для указанного тикера.

- Если указаны параметры р , d , q , обучается модель с заданными параметрами.
- Если включен флаг auto_select , подбираются оптимальные параметры для ARIMA.

Структура запроса (JSON):

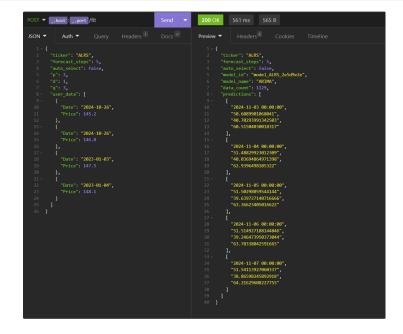
```
1 {
     "ticker": "string",
                                          // Название тикера
// Количество шагов прогноза
 2
 3
     "forecast_steps": "integer",
 4
     "auto_select": "boolean",
                                            // Флаг автоматического подбора параметров
 5 "p": "integer|null",
                                            // Параметр р, если auto_select=false
                                           // Параметр d, если auto_select=false
// Параметр q, если auto_select=false
 6
     "d": "integer|null",
     "q": "integer|null",
 8
      "user_data": [
                                            // Пользовательские данные
 9
      {
10
         "Date": "string",
                                           // Дата в формате YYYY-MM-DD
         "Price": "float"
11
                                             // Цена на указанную дату
      }
12
13
     1,
14
     "model_id": "string|null"
                                           // ID модели для переобучения
15 }
```

Пример запроса:

```
1 curl --request POST \
 2 --url http://localhost:8080/fit \
     --header 'Content-Type: application/json' \
    --data '{
 5
      "ticker": "ALRS",
      "forecast steps": 5,
 6
 7
      "auto_select": false,
      "p": 3,
 8
 9
      "d": 1,
10
      "q": 3,
11
      "user_data": [
12
13
             "Date": "2024-10-26",
14
              "Price": 145.2
15
        },
16
         {
             "Date": "2024-10-26",
17
              "Price": 146.0
18
19
```

Структура ответа (JSON):

```
2
     "ticker": "string",
                                          // Название тикера
 3
     "forecast_steps": "integer",
                                        // Количество шагов прогноза
     "auto_select": "boolean",
                                          // Флаг автоматического подбора параметров
 5
     "model_id": "string",
                                           // Уникальный идентификатор модели
 6
     "model name": "string",
                                          // Название модели ("ARIMA")
     "data_count": "integer",
 7
                                          // Количество строк данных, использованных для обучения
     "predictions": [
 8
                                          // Прогнозы, рассчитанные моделью
 9
10
         "string",
                                          // Дата прогноза в формате YYYY-MM-DD
11
        "float",
                                           // Предсказанное значени
12
        "float".
                                           // Нижняя граница доверительного интервала
13
        "float"
                                           // Верхняя граница доверительного интервала
     ]
15
     ]
16 }
```



3. /predict (GET) ⊘

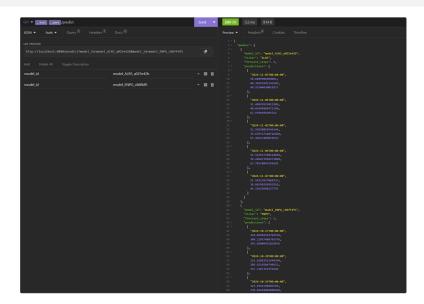
Описание: Получает прогнозы обученных моделей.

Пример запроса: (сначала необходимо получить список id моделей)

```
curl --request GET \
    --url 'http://localhost:8080/predict?model_id=model_ALRS_a025e42b&model_id=model_ENPG_c86ffdf1' \
    --header 'Content-Type: application/json'
```

Структура ответа (JSON):

```
1 {
 2
      "models": [
                                            // Список моделей с их прогнозами
          "model_id": "string",
                                            // Уникальный идентификатор модели
 5
         "ticker": "string",
                                            // Название тикера
          "forecast_steps": "integer",
 6
                                            // Количество шагов прогноза
 7
          "predictions": [
                                            // Список прогнозов (может отсутствовать, если есть ошибка)
 8
          [
            "string",
 9
                                            // Дата прогноза в формате YYYY-MM-DD
           "float",
"float",
10
                                            // Предсказанное значение
11
                                             // Нижняя граница доверительного интервала
12
            "float"
                                            // Верхняя граница доверительного интервала
13
14
15
          "error": "string|null"
                                            // Сообщение об ошибке (если применимо)
16
17
```



Описание интерфейса Streamlit &

Навигация 🔗

- Боковая панель Streamlit содержит переключатель для выбора разделов:
 - Главная
 - 。 /models: Просмотр списка моделей.
 - Ifit: Обучение моделей.
 - Ipredict: Получение прогнозов.
 - 。 /model_info: Детальная информация о моделях.
 - 。 **/eda**: Раздел для визуального анализа данных (**TBD**).

Paздел /models 🔗

Назначение: Просмотр списка доступных моделей, фильтрация по группам, тикерам и версиям.

Модели группируются по категориям:

- auto_initialized это модели, которые автоматически обучаются при старте приложения. Они используются для начальной оценки данных и создания базового набора прогнозов.
- $fit_endpoint$ это модели, которые были запущены и обучены пользователем через эндпоинт /fit.

Функционал:

- Фильтры:
 - Группы: Можно выбрать одну группу или "Все".
 - Тикеры: Фильтрация по тикерам.
- Отображение таблицы со следующими полями:
 - Группа.
 - ID модели.
 - Тип модели (например, ARIMA).
 - Тикер
 - ∘ Параметры модели (р, d, q).
 - Метрики качества (MAE, MAPE, R²).
 - Последняя дата данных.
 - Дата сохранения модели.



Раздел /fit ∂

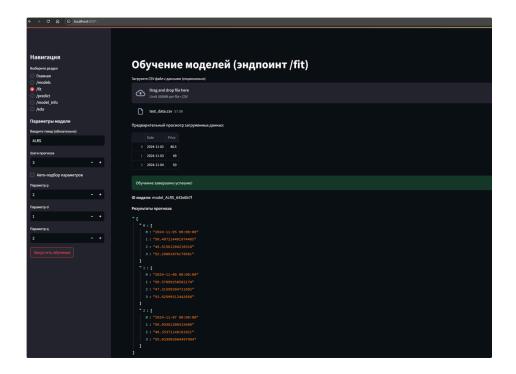
- Назначение: Обучение новых моделей на основе системных данных и/или пользовательских данных.
- Функционал:
 - Параметры модели:
 - Ввод тикера (обязательное поле).
 - Количество шагов прогноза.
 - Автоматический подбор параметров (auto_select).
 - Ручная настройка параметров (р , d , q) при отключенном авто-подборе.
 - Загрузка пользовательских данных:
 - Поддерживается загрузка CSV-файлов.
 - Отображение предварительного просмотра загруженных данных.
 - \circ Кнопка запуска обучения: Отправляет запрос к эндпоинту /fit .
 - **Результат обучения:** Отображение ID модели, прогноза и информации об успешности операции.

В репозитории предоставлен тестовый файл с данными, который можно использовать для загрузки через эндпоинт /fit . Файл находится по следующему пути:

datasets/test/test_data.csv

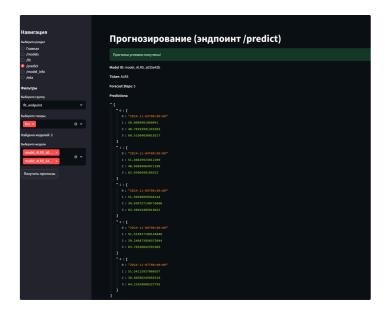
Содержимое файла:

- 1 Date,Price
- 2 2024-11-02,48.5
- 3 2024-11-03,49
- 4 2024-11-04,50



Раздел /predict 🔗

- Назначение: Получение прогнозов для ранее обученных моделей.
- Функционал:
 - Фильтры:
 - Выбор группы моделей.
 - Фильтрация по тикерам.
 - Выбор одной или нескольких моделей для прогнозирования.
 - ∘ Кнопка "Получить прогнозы":
 - Отправляет запрос к эндпоинту /predict.
 - Результаты прогноза:
 - Отображение данных по каждой выбранной модели:
 - ID модели.
 - Название тикера.
 - Количество шагов прогноза.
 - Список предсказаний.



Раздел /model_info ∂

• Назначение: Просмотр детальной информации о моделях и их прогнозах. Также предоставляет возможность визуализировать прогнозы нескольких моделей для сравнения.

Рекомендации по использованию: 🔗

- Для наглядности работы раздела рекомендуется:
 - а. Несколько раз запустить эндпоинт /predict на тестовых или других данных, используя один и тот же тикер, но с разными параметрами модели (например, с различными значениями р, d, q или с включенным/выключенным авто-подбором).
 - b. Полученные модели с разными настройками выбрать в разделе /model_info для сравнения их прогнозов.
- Основной функционал:
 - Фильтры для выбора моделей:
 - Группы:
 - auto_initialized (инициализированные при старте).
 - fit_endpoint (запущенные пользователем через /fit).
 - Тикеры:
 - Отображаются доступные тикеры для выбранной группы.
 - Модели:
 - Возможность выбрать одну или несколько моделей по ID.
 - Сравнение моделей:
 - Отображение ключевых параметров моделей:
 - Параметры ARIMA (p , d , q).
 - Количество шагов прогноза.
 - Метрики качества (МАЕ, МАРЕ, R²).
 - Визуализация прогнозов:
 - Интерактивные графики, показывающие предсказанные значения и доверительные интервалы для каждой выбранной модели.
 - Возможность наложить прогнозы нескольких моделей на один график.
 - Детальная информация:
 - Для каждой модели отображаются:
 - ID модели.
 - Название тикера.
 - Дата сохранения модели.
 - Метрики качества.
 - Данные о прогнозах (дата, предсказанное значение).

