Документация проекта

---

1. Описание структуры проекта

Проект состоит из следующих ключевых компонентов, каждый из которых выполняет уникальную роль в обеспечении обработки данных, построении моделей и предоставлении аналитики:

Каталоги

- `airflow/`
  - Управление рабочими процессами через DAGs и конфигурационные файлы Airflow.
  - **Ключевые файлы:**
    - `airflow.cfg` : Настройки для работы Airflow.
    - `Dockerfile` : Создание Docker-образа для Airflow.
    - `requirements.txt` : Зависимости Python.
- `api/`
  - Сервис FastAPI для управления моделями машинного обучения.
  - **Ключевые файлы:**
    - `api_router.py` : Определение маршрутов API.
    - `main.py` : Точка входа для API.
    - `Dockerfile` : Docker-образ для FastAPI.
    - `models/` : Сохранённые модели.
- `models/`
  - Jupyter Notebook и документация для анализа и построения baseline-моделей.
  - **Ключевые файлы:**
    - `base_line.ipynb` : Анализ данных и построение baseline.
    - `baseline.md` : Описание подхода baseline.
- `front/`
  - Интерфейс Streamlit для визуализации данных и взаимодействия с API.

- **Ключевые файлы:**
  - `front.py` : Основной файл Streamlit-приложения.
  - `logs/` : Логи с поддержкой ротации.

Другие ключевые файлы

- `docker-compose.yml` : Управление контейнерами для всех сервисов.
- `LICENSE` : Условия использования проекта.

---

2. Функционал API

Общая информация

- **Загрузка моделей:**
  - Все доступные модели автоматически загружаются при запуске сервера. Используются Lifecycle Events для устойчивости системы.

Конечные точки

1. `GET /models`
   - Возвращает список всех моделей с подробной информацией о них.
   - **Пример ответа:**

```
[
  {
    "id": "LinReg_GAZP",
    "name": "Model for GAZP (LinReg)",
    "type": "LinReg",
    "status": "loaded"
  }
]
```

2. `POST /set`
   - Устанавливает активную модель по `model_id`.
   - **Пример запроса:**

```
{
  "model_id": "LinReg_GAZP"
```

```
}
```

- **Пример ответа:**

```
{
  "id": "LinReg_GAZP",
  "name": "Model for GAZP (LinReg)",
  "type": "LinReg",
  "status": "active"
}
```

3. `POST /predict`

- Выполняет предсказание на основе активной модели.
- **Пример ответа:**

```
{
  "predictions": [123.45, 678.90]
}
```

4. `POST /fit`

- Запускает процесс обучения модели с заданными гиперпараметрами.
- **Пример ответа:**

```
{
  "status": "Запущено обучение."
}
```

5. **Дополнительные возможности**

- Динамическое добавление новых моделей.
- Использование Pydantic для строгой валидации данных.

---

3. Функционал Streamlit-приложения

Основные возможности

1. **Загрузка датасета**

- Поддержка загрузки данных в формате CSV.

- Табличное отображение загруженного датасета.

2. **Демонстрация аналитики (EDA)**
   - Отображение предварительно выполненного анализа через Jupyter Notebook.
   - Интеграция HTML-конверсии для визуализации данных.
3. **Создание моделей и настройка гиперпараметров**
   - Выбор типа модели (например, Linear Regression, ARIMA).
   - Настройка гиперпараметров через удобный интерфейс.
   - Запуск обучения и визуализация процесса.
4. **Просмотр информации о моделях и метриках**
   - Просмотр ключевых параметров текущей модели.
   - Загрузка и отображение кривых обучения.
5. **Инференс (предсказание)**
   - Загрузка данных для предсказания.
   - Отображение результатов в формате JSON.
6. **Логирование**
   - Полноценное логирование операций в папку `logs`.
   - Ротация логов для оптимального использования памяти.

Особенности

- Глубокая интеграция с FastAPI для выполнения всех операций.
- Удобный интерфейс для анализа данных и работы с моделями.
- Надёжная система логирования.

---

4. Минимальная инструкция для пользователя

Шаги для запуска

1. **Клонируйте репозиторий:**

```
git clone https://github.com/your-repo/project.git
cd project
```

2. **Настройте окружение:**
    - Создайте файл `.env` и заполните его необходимыми параметрами.
3. **Запустите контейнеры:**

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

4. **Доступ к сервисам:**
    - FastAPI: http://localhost:8001
    - Airflow: http://localhost:8080
    - Jupyter: http://localhost:8888
    - Streamlit: http://localhost:8501

Пример использования API

Для предсказаний:

```
curl -X POST "http://localhost:8001/predict" -F "file=@data.csv"
```
