Документация проекта

1. Описание структуры проекта

Проект состоит из двух основных компонентов:

1. **Серверная часть (FastAPI):**
   - **Файл:** `api_mvp.py`
   - Реализует HTTP API для управления моделями машинного обучения, выполнения их обучения, предсказания и предобработки данных.
   - Логирование осуществляется в файл `logs/app.log` с использованием ротации.
2. **Клиентская часть (Streamlit):**
   - **Файл:** `streamlit_app.py`
   - Предоставляет удобный веб-интерфейс для взаимодействия с серверной частью.
   - Пользователи могут загружать данные, анализировать их, обучать модели, просматривать кривые обучения и выполнять предсказания.

Структура папок

- `api_mvp.py`: Серверная часть на FastAPI.
- `streamlit_app.py`: Клиентская часть на Streamlit.
- `logs/`: Директория для хранения логов приложения.

---

2. Описание функционала API

Основные эндпоинты FastAPI

1. `GET /`
   - Проверка состояния сервера.
   - **Ответ:** `{ "message": "Мы работаем" }`
2. `POST /preprocess`
   - Выполняет предобработку загруженного датасета.
   - **Вход:** Файл в формате `.parquet`.
   - **Ответ:** Сообщение о статусе предобработки.
3. `POST /fit`
   - Обучает новую модель с заданной конфигурацией.
   - **Вход:** JSON с параметрами модели.

   Если config.id == "pretrained", считаем, что пользователь хочет использовать уже загруженную модель и не обучаем ничего Иначе – обучаем новую модель.

   - **Ответ:** Список ID обученных моделей.
4. `POST /predict`
   - Выполняет предсказание меток (хабов) и рейтингов на переданном датасете.
   - **Вход:** JSON с ID модели.

   Параметр config (JSON-строка) должен содержать 'id' (например, 'pretrained')

   - **Ответ:** JSON с предсказаниями.
5. `GET /models`
   - Возвращает список всех доступных моделей.
6. `POST /set`
   - Устанавливает текущую модель для предсказания.
   - **Вход:** ID модели.
   - **Ответ:** Подтверждение установки модели.

Логирование

- Логи записываются в файл `logs/app.log` с ротацией (максимум 1MB, 5 резервных копий).
- Формат логов: `[время] уровень – имя модуля – сообщение`.

---

3. Описание функционала Streamlit-приложения

Основные функции

1. **Загрузка данных:**
   - Поддерживается формат `.parquet`.
   - Загруженные данные отображаются в таблице.
2. **EDA:**
   - **Агрегированная информация:** Статистики (среднее, медиана, максимум, минимум, стандартное отклонение) по числовым колонкам.
   - **Распределение данных:** Построение гистограмм для числовых колонок.
   - **Топ-10 частотных слов:** Анализ наиболее популярных токенов в текстовых колонках.
   - **Облако слов:** Визуализация частотности токенов.
   - **Распределение частей речи:** График частоты частей речи из текста статей.

`To be continued`

4. Структура тестового датасета `data_sample_.parquet`, на котором тестируем сервис

Описание колонок

1. `author` (строка):
   - Имя автора статьи.
2. `publication_date` (дата/время):
   - Дата и время публикации статьи.
3. `hubs` (строка):
   - Тематические хабы (категории), к которым относится статья.
4. `comments` (числовое значение):
   - Количество комментариев к статье.
5. `views` (числовое значение):
   - Количество просмотров статьи.
6. `url` (строка):
   - Ссылка на статью.
7. `reading_time` (числовое значение):
   - Время, необходимое для чтения статьи, в минутах.
8. `individ/company` (строка):
   - Тип автора (индивидуальный или корпоративный).
9. `bookmarks_cnt` (числовое значение):
   - Количество добавлений статьи в закладки.
10. `text_length` (числовое значение):
    - Длина текста статьи в символах.
11. `tags_tokens` (список строк):
    - Токенизированные теги статьи.
12. `title_tokens` (список строк):
    - Токенизированное название статьи.
13. `rating_new` (числовое значение):
    - Рейтинг статьи.
14. `text_tokens` (список строк):
    - Токенизированное содержание текста статьи.
15. `text_pos_tags` (список строк):
    - Части речи для токенов текста (например, `NOUN`, `VERB`, `ADJ`).
16. `rating_level` (строка):
    - Уровень рейтинга статьи (например, `very positive`, `neutral`).

Пример строки

author	publication_date	hubs	comments	views	url	reading_time	individ/company	bookmarks_cnt
krig	2010-01-21 13:11:17+00:00	Мессенджеры	49	1000	https://habr.com/ru/articles/81478/	1.0	individual	7.0

5. Инструкция по использованию

Установка и запуск(это переделать, наверное)

1. **Установите зависимости:**

```
pip install -r requirements.txt
```

2. **Запустите сервер FastAPI:**

```
uvicorn api_mvp:app --reload
```

3. **Запустите клиент Streamlit:**

```
streamlit run streamlit_app.py
```

4. **Перейдите в браузер по адресу:**

Работа с приложением

1. **Загрузите данные** в формате `.parquet`.
2. **Выберите необходимый функционал через интерфейс:**
   - **EDA:** Изучите данные.
   - **Создание модели:** Настройте и обучите новую модель.
   - **Инференс:** Выполните предсказания на новых данных.
   - **Просмотр информации и кривые обучения:** Просмотрите доступные модели и кривые обучения.