**Построение вопросно-ответной системы с использованием**

**RAG (retrieval-augmented generation)**

1. **Структура проекта:**

app/

└── fastapi\_back/

├── .dockerignore

├── Dockerfile -- создание образа из наших файлов приложения

├── LICENSE

├── README.md

├── requirements.txt  - список зависимостей окружения для fastapi

└── src/

    ├── \_\_init\_\_.py

    ├── api/

    │   └── v1/

    │       ├── api\_route.py  -- Основные эндпоинты для взаимодействия с моделью

    │       ├── schemas.py  -- Схемы вопросов и ответов

    │       └── tfidf\_pretrained.joblib  -- Предобученная модель tf-idf

    ├── logger.py  -- Конфигурация инструмента для логирования

    ├── main.py -- Основной файл для запуска приложения

    ├── qdrant/

    │   └── load\_qdrant.py  -- Функции для интеграции с векторной БД qdrant

    └── tests/

        ├── conftest.py  -- Конфигурация тестов

        ├── full\_dataset.csv  -- Датасет для тестов

        └── test\_model.py  -- Тесты

└── grafana/provisioning/datasources/graf\_loki.yaml – файл настройки мониторинга приложения и логов

└── streamlit/

├── config.toml - конфигурация темы приложения

├── .dockerignore

├── Dockerfile -- создание образа из наших файлов приложения

├── eda.py - функции для отрисовки графиков, препроцессинг текстов

├── project\_logger.py - модуль логирования, установки и функция-обертка

├── st\_app.py - основной код приложений

├── validat\_df.py - модуль, отвечающий за валидацию в загружаемого приложение файла

├── requirements.txt  - список зависимостей окружения для streamlit

└── docker-compose.yml

└── promtail.yaml – файл настройки сборщика логов

└── loki.yaml – файл настройки сборщика логов

1. **Функционал API**

- Загрузка датасетов

- Обучение моделей (в настоящее время поддерживается TF-IDF)

- Интеграция с векторной базой данных Qdrant для эффективного поиска по схожести

- Загрузка и выгрузка моделей из оперативной памяти (TODO)

- Поиск контекста для заданных вопросов

- Тестирование качества и скорости модели

- Вывод списка загруженных наборов данных и обученных моделей

- Удаление моделей (по одной или всех сразу)

**API Endpoints**

- `POST /api/v1/models/load\_dataset`: Загрузить датасет

- `POST /api/v1/models/fit\_save`: Обучить модель и сохранить в Qdrant

- `POST /api/v1/models/load\_model`: Загрузить модель в оперативную память

- `POST /api/v1/models/unload\_model`: Выгрузить модели из оперативной памяти

- `POST /api/v1/models/find\_context`: Найти контекст для заданного вопроса

- `POST /api/v1/models/quality\_test`: Оценить точность и производительность модели

- `GET /api/v1/models/get\_datasets`: Получить список загруженных наборов данных

- `GET /api/v1/models/list\_models`: Получить список загруженных и обученных моделей

- `DELETE /api/v1/models/remove/{model\_id}`: Удалить конкретную модель

- `DELETE /api/v1/models/remove\_all`: Удалить все модели

1. **Функционал streamlit-приложения**

- Загрузка датасета и анализ данных

- Препроцессинг данных

- Построение графиков

- Конфигурирование и обучение модели

- Сравнение моделей

- Получение инференса

1. **Запуск приложения**

А) Приложение развернуто на VPS:

- <http://178.130.43.233:8501/>

- http://178.130.43.233:3000/ - Мониторинг и визуализацияя приложения и логов

Сервер очень слабый и полный функционал приложения показать не сможет, максимум допустимый датасет для использования на сервере должен содержать не более 100 записей.

Б) Запуск приложения на локальной машине:

Для запуска требуется

1. Установить по инструкции docker в зависимости от ОС https://docs.docker.com/compose/install/

2. В папке app прописать команду для сборки и запуска приложения:

docker-compose -p app\_rag up

3. Перейте по ссылке http://localhost:8501/ для входа в приложение

4. Для мониторинга приложения и сбора логов требуется:

- Перейте по ссылке http://localhost:3000/

- Подключить datasources loki, по адресу http://loki:3100/

- В разделе explore/loki - ведем мониторинг приложения и логов.

**Инструкция использования в файле https://drive.google.com/file/d/1B3RqPt2BVKPCuhqpltTlJFle6sKCiP\_t/view?usp=sharing**