

Проект состоит из основных частей: серверной(server.py) и клиентской части (app.py). Серверная часть реализована с помощью технологии FasAPI, а клиентская – с использованием Streamlit.

Серверная часть (server.py)

FstAPI: Используется для создания REST API, который обрабатывает запросы от клиентской части.

Модели данных: Расположены в директориях models/request_models.py и models/response_models.py. Они используются для валидации входных и выходных данных API.

Обучающее ядро: Расположено в директории trainer_core. В модуле upload_dataset.py реализованы функции upload_emotion_class, is_image и upload_dataset_inframe, которые отвечают за извлечение данных с датасета для предварительной обработки:

upload_emotion_class – извлекает классы из датасета. В корневом каталоге датасета должны находиться каталоги в количестве не менее двух, названия которых соответствуют названию классов.

is_image – проверка, является ли файл изображением

upload_dataset_inframe – извлечение данных из датасета и их преобразование в структурированный датафрейм

Модули dataset.py и extraction.py отвечают за обработку и извлечения признаков из изображений.

Модуль fit.py содержит функцию fit_train, которая отвечает за обучение модели.

Модуль eda.py содержит функцию eda_info, которая отвечает за аналитику данных.

Модуль predict содержит функции load_model_inference и predict_inference. load_model_inference отвечает за загрузку обученной для инференса, а predict_inference реализует сам инференс.

Хранение данных: Управляется через services/storage_service.py, который отвечает за загрузку, удаление и перечисление моделей и датасетов. Датасеты загружаются и хранятся в директории datasets, а модели в директории models_train.

Логирование: Настроено через core/logger.py

Эндпоинты (endpoints) расположены в service.py. В нем находятся:

/load_dataset - отвечает за загрузку датасета на сервер;

/eda – отвечает за аналитику датасета;

/fit – отвечает за обучение модели;

/load_model – отвечает за загрузку модели в инференс;

/list_models – отвечает за вывод списка моделей, которые хранятся в директории models_train;

/list_datasets – отвечает за вывод списка уже обученных моделей, которые хранятся в директории /datasets;

/remove_model – отвечает за удаление модели из директории models_train;
/remove_dataset – отвечает за удаление датасета из директории datasets;
/remove_all_datasets – отвечает за удаление всех датасетов из директории datasets;
/remove_all_models – отвечает за удаление всех моделей из директории models_train;
/predict – отвечает за инференс;

Клиентская часть (app.py)

Реализована с помощью фреймворка **Streamlit**: Используется для создания веб-интерфейса, который взаимодействует с серверной частью;

Функционал: Загрузка датасетов, обучение моделей, инференс моделей, удаление моделей и датасетов, а также проведение разведочного анализа данных ;

Загрузка датасета: Позволяет пользователю загрузить zip-файл с датасетом.

EDA: Показывает статистику по выбранному датасету;

Обучение модели: Позволяет выбрать датасет, указать параметры модели и обучить её;

Инференс: Позволяет загрузить zip-файл с датасетом для генерации предсказаний с помощью обученной модели;

Удаление моделей и датасетов: Позволяет удалить выбранные модели и датасеты;

Инструкция

Загрузка датасета

Загрузить датасет можно в формате zip-файла в разделе “Загрузка датасета”. Распакованной Zip-file должен быть единственной директорией. В этой директории должны содержаться директории, соответствующие различным классам. Названия директорий соответствует названиям классов. В каждой “классовой” директории должны содержаться файлы с картинками, соответствующими этому классу.

(Опционально) Проведение EDA.

Провести EDA можно в разделе “Exploratory Data Analysis”. Для этого надо выбрать один из уже загруженных датасетов и нажать на кнопку “Провести EDA”. В результате этого, на экран выводится статистика о размерах картинок и статистика о распределении по цветовым каналам.

Обучение модели.

В разделе “Обучение модели” надо выбрать один из уже загруженных датасетов, ввести название для обучаемой модели и задать гиперпараметры. Обучение представляет из себя обучения алгоритма SVM на векторных представления для картинок, полученных с помощью предобученной на ImageNet модели ResNet18. Надо выбрать следующие гиперпараметры:

- *Ядро регрессии.*

Возможные варианты: linear, poly или rbf. Ядро, используемое в SVM. Рекомендуемое значение: linear.

- *Коэффициент регуляризации.*

Коэффициент регуляризации, используемый в SVM. Рекомендуемое значение: 1.0

В результате обучения, модель с введенным названием будет сохранена на сервере и ее можно будет использовать для инференса.

Инференс модели.

Доступно в разделе “Инференс модели”. Пользователю следует выбрать модель, из сохраненных на сервере, которую он хочет использовать для инференса и нажать на кнопку “Загрузить модель”. После этого следует загрузить zip-file, который содержать единственную директорию, состоящую из изображений, подготовленных для инференса. Затем, после нажатия кнопки “Инференс модели”, на экране появятся изображения с подписанным названиями файлов и предсказанием модели.

(Опционально) Удаление датасетов и моделей.

В разделе “Удаление датасетов и моделей” пользователю предоставляется возможность удалить ненужные модели или датасеты с удаленного сервера. Удаление осуществляется посредством выбора датасета/модели в выпадающем микшере и последующем нажатии на кнопку “Удалить датасет/модель”.