Проект: Predicting the value of used cars

1. Структура проекта

```
frontend/
  Dockerfile              — сборка контейнера Streamlit
  requirements.txt        — зависимости фронтенда
  app.py                  — точка входа Streamlit
  api_client.py           — HTTP-клиент для FastAPI
  modules/                — страницы приложения:
        upload.py         — загрузка CSV
        eda.py            — аналитика и графики
        train.py          — интерфейс обучения
        predict.py        — интерфейс предсказания

backend/
  Dockerfile              — сборка контейнера FastAPI
  requirements.txt        — зависимости бэкенда
  app/                — модуль приложения:
        __init__.py
        helper.py               — утилиты и вспомогательные функции
        main.py                 — FastAPI-сервис, точка входа
        model_trainer.py        — класс Trainer для обучения
        schemas.py              — Pydantic-модели запросов/ответов
        train_process.py        — фоновый процесс обучения
  models/           — сохранённые модели:
        final_model.pkl         — предобученная модель

filebeat/
  filebeat.yml            — конфигурация Filebeat для сбора логов

docker-compose.yml    — оркестрация frontend, backend и filebeat
.venv/                — виртуальное окружение (игнорируется)
```

2. Функционал API (FastAPI)

GET /models — получить список моделей
POST /models/{id}/set — установить активную модель
POST /fit/json — обучение на JSON-данных
POST /fit/csv — обучение через CSV + query-параметры
POST /predict/json — предсказание по JSON
POST /predict/csv — пакетное предсказание по CSV

3. Функционал Streamlit-приложения

Upload — загрузка датасета (CSV) и просмотр
EDA — анализ данных: гистограммы, scatter, корреляция
Train — выбор гиперпараметров + обучение (JSON/CSV)
Predict — предсказание цены (форма или CSV)

4. Краткая инструкция

---

1. Клонировать репозиторий:
   git clone
https://github.com/chetaleks/Predicting-the-value-of-used-cars

2. Настроить backend:
   cd backend
   python -m venv venv
   source venv/bin/activate # PowerShell: .\venv\Scripts\Activate.ps1
   pip install -r requirements.txt

3. Запустить backend:
   uvicorn app.main:app --reload

4. Настроить frontend:
   cd frontend
   source venv/bin/activate # PowerShell: .\venv\Scripts\Activate.ps1
   pip install -r requirements.txt

5. Запустить frontend:
   streamlit run app.py

6. Открыть в браузере:
   http://localhost:8501
   и пользоваться приложением:
   - Upload: загрузить CSV
   - EDA: анализ данных
   - Train: обучить модель
   - Predict: предсказать цену

5. Запуск через Docker-Compose

---

1. В корне проекта есть файл docker-compose.yml

2. Собрать образы и запустите оба сервиса командой:
   docker-compose up --build -d

3. Проверить работу:
   Streamlit: http://localhost:8501

4. Просмотр логов:
   i. в папке logs\ в корне проекта
   ii. через Kibana: http://localhost:5601

5. Остановить и удалить контейнеры:
   docker-compose down