**Практика 7. Развёртывание модели через FastAPI и Docker**

**Цель**

Научиться:

* обучать модель и сохранять её через MLflow,
* запускать локальный REST-сервис для модели,
* упаковывать сервис и модель в Docker-контейнер,
* проверять работу контейнера через HTTP-запросы.

**Этап 1. Подготовка данных и обучение модели**

* Выберите **любой встроенный датасет** из sklearn.datasets.
* Обучите модель классификации или регрессии (файлы train\_cloud.py даются преподавателем).
* После запуска скрипта у вас должны появиться:
  + папка mlruns/ (лог экспериментов MLflow),
  + папка model/ (артефакт модели для сервиса).

**Этап 2. Локальный REST-сервис**

* Для запуска сервиса используется файл serve.py (он дан преподавателем).
* Запустите сервис:
* uvicorn serve:app --host 0.0.0.0 --port 8000
* Проверьте эндпоинт /predict, отправив POST-запрос с JSON, содержащим:
  + список колонок (columns),
  + список строк с признаками (data).

Пример PowerShell-запроса:

$body = @{

columns = @("feature1","feature2","feature3","feature4")

data = @(@(5.1,3.5,1.4,0.2))

} | ConvertTo-Json -Depth 5

Invoke-RestMethod -Uri http://127.0.0.1:8000/predict `

-Method POST -ContentType "application/json" -Body $body

**Этап 3. Установка Docker**

1. Перейдите на https://www.docker.com/products/docker-desktop/ и скачайте Docker Desktop.
2. Установите его и убедитесь, что демон работает:
3. docker --version
4. docker info

**Этап 4. Подготовка Dockerfile**

* Создайте файл Dockerfile в корне проекта:
* FROM python:3.10-slim
* WORKDIR /app
* ENV PYTHONDONTWRITEBYTECODE=1
* ENV PYTHONUNBUFFERED=1
* COPY requirements.txt /app/
* RUN pip install --no-cache-dir -r requirements.txt
* COPY serve.py /app/serve.py
* COPY model/ /app/model/
* EXPOSE 8000
* CMD ["python","-m","uvicorn","serve:app","--host","0.0.0.0","--port","8000"]
* Создайте .dockerignore:
* mlruns/
* \_\_pycache\_\_/
* .venv/
* \*.pyc
* \*.pyo

**Этап 5. Сборка и запуск контейнера**

1. Соберите образ:
2. docker build -t iris-model .
3. Запустите контейнер:
4. docker run --rm -p 1234:8000 iris-model
5. Проверьте работу сервиса:
6. Invoke-RestMethod -Uri http://127.0.0.1:1234/predict -Method POST `
7. -ContentType "application/json" -Body $body

**Что сдать**

1. Скриншоты:
   * MLflow UI с параметрами и метриками,
   * работающий сервис (Uvicorn),
   * успешный запрос к /predict локально,
   * успешный запрос к /predict в контейнере.
2. Репозиторий (или архив проекта) с файлами:
   * train\_cloud.py,
   * serve.py,
   * requirements.txt,
   * Dockerfile,
   * .dockerignore,
   * README.md (с инструкциями запуска).
3. Краткий отчёт: какой датасет выбран, какая задача решалась, какие метрики качества получены.