Приблизительная структура проекта детекции аномалий

anomaly-detection-project/

├── data/                           # Папка для данных (не коммитить большие файлы!)

│   ├── raw/                        # Исходные сырые данные

│   ├── processed/                  # Предобработанные данные

│   └── external/                   # Внешние наборы данных

│

├── notebooks/                      # Jupyter ноутбуки по уровням

│   ├── 0\_Basics/                   # Уровень 0: Основы

│   │   ├── 01\_Data\_Exploration.ipynb

│   │   ├── 02\_Statistical\_Methods.ipynb

│   │   └── datasets/               # Примеры маленьких датасетов для Level 0

│   │

│   ├── 1\_Classical\_ML/             # Уровень 1: Классические ML

│   │   ├── 01\_Isolation\_Forest.ipynb

│   │   ├── 02\_LOF.ipynb

│   │   └── ...

│   │

│   ├── 2\_RealWorld\_Challenges/     # Уровень 2: Реальные задачи

│   │   ├── 01\_Time\_Series\_Analysis.ipynb

│   │   ├── 02\_High\_Dimensionality.ipynb

│   │   └── ...

│   │

│   ├── 3\_Advanced\_Scaling/         # Уровень 3: Продвинутые методы

│   │   ├── 01\_Autoencoders.ipynb

│   │   ├── 02\_Dask\_Scaling.ipynb

│   │   └── ...

│   │

│   └── 4\_Final\_Experiments/        # Уровень 4: Финальные эксперименты

│       ├── 01\_Model\_Comparison.ipynb

│       └── ...

│

├── src/                            # Переиспользуемый код

│   ├── data\_processing/            # Скрипты обработки данных

│   │   ├── [preprocessors.py](http://preprocessors.py/)

│   │   └── time\_series.py

│   │

│   ├── models/                     # Реализации моделей

│   │   ├── isolation\_forest.py

│   │   ├── [autoencoders.py](http://autoencoders.py/)

│   │   └── ...

│   │

│   ├── evaluation/                 # Метрики и оценка

│   │   ├── [metrics.py](http://metrics.py/)

│   │   └── [visualization.py](http://visualization.py/)

│   │

│   ├── utils/                      # Вспомогательные утилиты

│   │   ├── logging\_config.py

│   │   └── [helpers.py](http://helpers.py/)

│   │

│   └── pipelines/                  # Готовые пайплайны

│       ├── training\_pipeline.py

│       └── inference\_pipeline.py

│

├── configs/                        # Конфигурационные файлы

│   ├── model\_configs.yaml

│   └── data\_config.yaml

│

├── saved\_models/                   # Сохраненные модели (в .gitignore)

│   ├── isolation\_forest\_model.joblib

│   └── autoencoder\_model.h5

│

├── tests/                          # Тесты для ключевых компонентов

│   ├── test\_data\_processing.py

│   └── test\_models.py

│

├── docs/                           # Документация

│   ├── project\_structure.md

│   ├── anomaly\_types.md

│   └── algorithm\_comparison.md

│

├── reports/                        # Генерация отчетов

│   ├── figures/                    # Сохраненные графики

│   └── final\_report.ipynb

│

├── .github/                        # CI/CD и шаблоны

│   ├── workflows/

│   │   └── tests.yml

│   └── ISSUE\_TEMPLATE.md

│

├── .gitignore                      # Игнорируемые файлы

#├── requirements.txt                # Основные зависимости

#├── environment.yml                 # Conda environment

├── pyproject.toml # Новый файл конфигурации Poetry

├── poetry.lock # Автоматически генерируемый файл

├── Dockerfile                      # Для контейнеризации

├── Makefile                        # Упрощение команд

└── [README.md](http://readme.md/)                       # Главное описание проекта

```