**Структура проекта для выполнения вне Google Colab.**

MLLibrariesComparison/

├── README.md

├── requirements.txt

├── data/

│ ├── raw/

│ └── processed/

│ ├── regression/

│ │ ├── X\_train.npy

│ │ ├── X\_val.npy

│ │ ├── X\_test.npy

│ │ ├── y\_train.npy

│ │ ├── y\_val.npy

│ │ └── y\_test.npy

│ └── classification/

│ ├── X\_train.npy

│ ├── X\_val.npy

│ ├── X\_test.npy

│ ├── y\_train.npy

│ ├── y\_val.npy

│ └── y\_test.npy

├── src/

│ ├── \_\_init\_\_.py

│ ├── data\_preparation.py

│ ├── constants.py

│ ├── utils.py

│ ├── models/

│ │ ├── \_\_init\_\_.py

│ │ ├── train\_sklearn\_regression.py

│ │ ├── train\_tensorflow\_regression.py

│ │ ├── train\_pytorch\_regression.py

│ │ ├── train\_sklearn\_classification.py

│ │ ├── train\_tensorflow\_classification.py

│ │ └── train\_pytorch\_classification.py

│ ├── evaluation/

│ │ ├── \_\_init\_\_.py

│ │ └── compare\_models.py

│ └── visualization/

│ ├── \_\_init\_\_.py

│ └── plots.py

├── models/

│ ├── regression/

│ │ ├── sklearn\_model.pkl

│ │ ├── tensorflow\_model.h5

│ │ └── pytorch\_model.pth

│ └── classification/

│ ├── sklearn\_model.pkl

│ ├── tensorflow\_model.h5

│ └── pytorch\_model.pth

├── results/

│ ├── regression/

│ │ ├── sklearn/

│ │ │ ├── metrics.json

│ │ │ ├── learning\_curve.png

│ │ │ └── other\_plots.png

│ │ ├── tensorflow/

│ │ │ ├── metrics.json

│ │ │ ├── learning\_curve.png

│ │ │ └── other\_plots.png

│ │ └── pytorch/

│ │ ├── metrics.json

│ │ ├── learning\_curve.png

│ │ └── other\_plots.png

│ ├── classification/

│ │ ├── sklearn/

│ │ │ ├── metrics.json

│ │ │ ├── learning\_curve.png

│ │ │ ├── confusion\_matrix.png

│ │ │ └── other\_plots.png

│ │ ├── tensorflow/

│ │ │ ├── metrics.json

│ │ │ ├── learning\_curve.png

│ │ │ ├── confusion\_matrix.png

│ │ │ └── other\_plots.png

│ │ └── pytorch/

│ │ ├── metrics.json

│ │ ├── learning\_curve.png

│ │ ├── confusion\_matrix.png

│ │ └── other\_plots.png

│ └── summary/

│ ├── comparison\_table.xlsx

│ ├── Результаты\_классификации.xlsx

│ ├── Результаты\_классификации.csv

│ ├── Результаты\_регрессии.xlsx

│ └── Результаты\_регрессии.csv

└── notebooks/

└── exploratory\_data\_analysis.ipynb

**1. README.md**

Содержит общее описание проекта, инструкции по установке и запуску, а также любую другую важную информацию.

**2. requirements.txt**

**3. data/**

Хранение данных проекта.

* **raw/**: сырые данные. Если вы используете исходные файлы данных, сохраните их здесь.
* **processed/**: предобработанные данные, готовые к использованию в моделях.
* **regression/**:
  + X\_train.npy, X\_val.npy, X\_test.npy: данные признаков для обучения, валидации и тестирования.
  + y\_train.npy, y\_val.npy, y\_test.npy: целевые значения для обучения, валидации и тестирования.
* **classification/**:
  + Аналогичные файлы для задачи классификации.

**4. src/**

Исходный код проекта.

* **\_\_init\_\_.py**: указывает Python, что это пакет.

**4.1. data\_preparation.py**

Скрипт для подготовки данных (загрузка, предобработка, разделение на обучающую, валидационную и тестовую выборки).

**4.2. constants.py**

Файл для хранения констант и настроек проекта (пути к файлам, гиперпараметры и т.д.).

**4.3. utils.py**

Вспомогательные функции и классы, используемые в проекте.

**4.4. models/**

Код для обучения моделей.

* **\_\_init\_\_.py**
* **train*sklearn*regression.py**: обучение модели регрессии с использованием scikit-learn.
* **train*tensorflow*regression.py**: обучение модели регрессии с использованием TensorFlow.
* **train*pytorch*regression.py**: обучение модели регрессии с использованием PyTorch.
* **train*sklearn*classification.py**: обучение модели классификации с использованием scikit-learn.
* **train*tensorflow*classification.py**: обучение модели классификации с использованием TensorFlow.
* **train*pytorch*classification.py**: обучение модели классификации с использованием PyTorch.

Каждый скрипт содержит:

* Загрузку предобработанных данных.
* Определение модели.
* Процесс обучения модели с возможностью ранней остановки.
* Оценку модели на тестовых данных.
* Сохранение модели и результатов (метрики, графики).

**4.5. evaluation/**

Код для оценки и сравнения моделей.

* **\_\_init\_\_.py**
* **compare\_models.py**: скрипт для создания сводных таблиц результатов по всем моделям.

**4.6. visualization/**

Код для визуализации данных и результатов.

* **\_\_init\_\_.py**
* **plots.py**: функции для построения графиков (кривые обучения, матрицы ошибок и т.д.).

**5. models/**

Сохраненные обученные модели.

* **regression/**:
* sklearn\_model.pkl
* tensorflow\_model.h5
* pytorch\_model.pth
* **classification/**:
* Аналогичные файлы для модели классификации.

**6. results/**

* **regression/**:
* **sklearn/**, **tensorflow/**, **pytorch/**:
  + metrics.json
  + learning\_curve.png
  + other\_plots.png (если есть)
* **classification/**:
* Аналогичная структура, в дополнение – confusion\_matrix.png.
* **summary/**:
* comparison\_table.xlsx: общая сводная таблица результатов.
* Результаты\_классификации.xlsx и .csv: сводные результаты классификации.
* Результаты\_регрессии.xlsx и .csv: сводные результаты регрессии.

**7. notebooks/**

Jupyter Notebooks для анализа данных и экспериментов.

* **exploratory*data*analysis.ipynb**