

Overview

???????????? MLumina

MLumina — библиотека Python (`src/mlumina`) с набором независимых друг от друга инструментов для ML-экспериментов на самостоятельно развёрнутом Ray-кластере с трекингом в MLflow. Это **не** единый AutoML-пайплайн с общим API — каждый подпакет решает свою задачу и используется отдельно.

“ Раздел `../make_docs/` содержит черновик Sphinx-документации, начатый ранее. Его вводный текст (`make_docs/source/introduction/what_is_mlumina.rst`) описывает MLumina как универсальный мультимодальный AutoML-фреймворк с эволюционным поиском пайплайнов — **это не соответствует текущему коду** и не должно использоваться как источник истины. Актуальное описание — в этой папке.

?? ????? ????????? ????????????

Пакет	Что делает	Документация
<code>mlumina.core</code>	Конфигурация (pydantic-settings, YAML+CLI) и логирование	<code>core.md</code>
<code>mlumina.xgboost</code>	Обучение и подбор гиперпараметров XGBoost (локально и через Ray)	xgboost.md
<code>mlumina.image_models</code>	Обёртка над <code>ultralytics.YOLO</code> : инференс, обучение, тюнинг	image-models.md
<code>mlumina.torch_models</code>	Обёртка над <code>torch.nn.Module</code> + примитивный перебор архитектур	torch-models.md
<code>mlumina.interpolation_tables</code>	2D-интерполяция табличных данных + сравнение sklearn-моделей	interpolation-tables.md
<code>mlumina.components_wear</code>	Симуляция износа и отказов графа взаимозависимых компонентов	components-wear.md
<code>mlumina.ray_wrapper</code>	Готовые пайплайны Ray Train/Tune (lin_reg, svm, fcnn, cnn, vit)	ray-wrapper.md

Пакет	Что делает	Документация
<code>mlumina.scheduler</code>	FastAPI-сервис приоритетной очереди задач поверх Ray Jobs API	scheduler.md

Также см.:

- [getting-started.md](#) — установка, окружение, запуск локального стека.
- [deployment.md](#) — Docker/Compose-развёртывание (Ray-кластер, MLflow, MinIO, MySQL, scheduler).
- [known-issues.md](#) — известные баги и расхождения между кодом и старой документацией (README/`make_docs`). Прочитать перед тем, как полагаться на `components_wear` или на примеры из корневого README.

? ???? ??????

1. [getting-started.md](#) — установка пакета и зависимостей.
2. Самый зрелый и рабочий модуль — `mlumina.xgboost`, см. [xgboost.md](#) и `examples/xgboost/example.py`.
3. Для распределённых экспериментов на Ray — [ray-wrapper.md](#) и [scheduler.md](#).
4. Перед использованием `components_wear` — обязательно [known-issues.md](#) (там баг, ломающий `update_wear`/`predict_fix`).