

# Руководство пользователя по запуску Boosting Testing Framework

## Структура проекта

В корне проекта находятся основные скрипты и каталоги:

- `main.py` — точка входа с CLI
- `configs/` — JSON-файлы с описанием экспериментов
- `datasets/` — папка с исходными и сгенерированными CSV
- `models/` — реализации алгоритмов бустинга
- `utils/` — утилиты для генерации данных, визуализации и парсинга конфигов
- `results/` — сюда будут сохраняться результаты учебного и тестового запусков

## Генерация синтетических датасетов

Запустите интерактивный генератор, чтобы создать двумерные выборки и сохранить их в `datasets/`. При нажатии на график добавляйте контрольные точки, настраивайте параметры шума и числа образцов, затем нажмите **Save**.

```
python main.py --mode generate
```

## Запуск обучения эксперимента

Выполните команду с указанием нужного JSON-конфига из папки `configs`. Если параметр `--cfg` не задан, будет взят файл `cfg.json`. На выходе в `results/` появится папка вида `<timestamp>-<cfg_name>`, содержащая копии `train/test` выборок, предсказания и файл `results.csv` со сводными метриками.

```
python main.py --mode train --cfg <имя_конфигурации>.json
```

## Визуализация результатов

Чтобы построить графики по уже имеющимся данным, запустите режим `plot`. Параметр `--plot_dirs` позволяет указать конкретные подпапки внутри `results`. В указанной директории появится папка `plots/` с PNG-файлами.

```
python main.py --mode plot --plot_dirs <папка1> <папка2> ...
```

## Полный цикл «обучение + визуализация»

Для автоматического последовательного запуска обучения и построения графиков используйте режим `trainplot`. Все визуализации будут сохранены в `results/<...>/plots/`.

```
python main.py --mode trainplot --cfg <имя_конфигурации>.json
```

## Настройка эксперимента

Откройте нужный JSON-файл в каталоге `configs/` и измените четыре раздела: список `algorithms` (имен моделей), блок `estimator` (тип и параметры базового классификатора), раздел `test` (параметры разбивки, число синтетических задач, многопоточность и список CSV) и раздел `model` (сетки гиперпараметров для каждого алгоритма). После сохранения изменений выполните команду запуска обучения или полного цикла.

## Советы по отладке и воспроизводимости

Всегда указывайте `random_state` в конфиге для идентичности выборок между запусками. Для контроля использования оперативной памяти можно вызывать функцию `utils/misc.get_memory_usage_mb()`. При необходимости запуска отдельных компонентов или отладки визуализации обращайтесь напрямую к функциям в `models/` и `utils/plotting.py`.