Домашка №3

- В данной домашке мы потренируемся в менеджменте ML экспериментов. Домашнее задание можно выполнять с **MLFlow** или **W&B**. Разницы между ними большой нет, единственное, MLFlow потребуется развернуть локально (обязательно в контейнере!);
- На данных, которые вы предобработали и положили в S3, необходимо провести эксперименты:
    - Выберите произвольную ML модель из scikit-learn, которая решает вашу задачу (классификация/регрессия/кластеризация/etc.);
    - Напишите питонячий скрипт или функцию, которая на вход принимает конфиг с гиперпараметрами ML модели и путь до датасета, обучает модель, вычисляет метрики качества и отправляет их в трекер экспериментов
    - Выберите сетку гиперпараметров (напр., перебираем в дереве решений max_depth и max_features), и для каждой комбинации гиперпараметров проведите обучение модели
    - Результат эксперимента, обученную модель, выложите на S3 в папку, у которой название совпадает с названием эксперимента
- Важно! Следуйте рекомендациям вашего шаблона проекта, за нарушение структуры проверяющий оставляет за собой право снизить баллы
- **Важно!** В домашней работе предусмотрите, чтобы запуск экспериментов проводился в **контейнере**. При этом хорошей практикой является для каждого эксперимента запускать отдельных контейнер. В домашней работе это необязательно, однако за реализацию запуска набора экспериментов, где отдельный запуск происходит в своем контейнере, можно получить дополнительно +5 баллов;
- **Важно 2!** Если вы решили использовать W&B, то при сдаче домашки приложите ссылки на ваши эксперименты
- **Важно 3! Постарайтесь сделать ваш код модульным** и изолировать различные шаги (например, отдельно скрипт скачивания и отправки данных в S3 и отдельно скрипт обработки);
- **Важно 4! Итоговый проект с экспериментами и всеми необходимыми скриптами из прошлых домашек должен проходить настроенный в проекте линтер и тайпчекер!**
- Для удобства сделайте bash скрипт, который будет создавать образ для обучения, поднимать s3 (при условии, что вы его используете локально), опционально поднимать MLFlow. Т.е. скрипт, который подготовит все для запуска экспериментов;
- Дополнительно сделайте bash или python скрипт, который будет принимать на вход конфиг с сеткой гиперпараметров для перебора и запускать эксперименты.

Алгоритм проверки домашки:

- Проверяющий клонирует репозиторий с проектом и разворачивает окружение по инструкции из README.md (или с помощью запуска bash/Makefile сценария). (возможно здесь стоит перейти на docker-compose для локального s3 и трекера экспериментов)
- Проверяющий убеждается, что все контейнеры поднялись и работают корректно;

- В случае MLFlow проверяющий запускает скрипты обучения и убеждается, что они корректно залогировались и модели в S3 сохранились;
- В случае W&B проверяющий может либо запустить код, но сохранить результаты экспериментов в своем пространстве (всм в своем аккакунте), либо провести код ревью и убедиться, что код выполняет заданные функции.

Важные уточнения:

- Итогом вашей работы всегда должен быть коммит в мастере через механизм PR, который прошел все проверки линтеров.
- Каждая лаба – логическое продолжение предыдущей, все изменения должны подливаться в осноной репозиторий в мастер.
- Код стоит поддерживать под unix (macos, linux), адаптировать под windows не нужно
- Шаблон, который собирается в первой лабе призван унифицировать работу с проектом. Следуйте тем правилам, которые в нем описаны.

Полезные советы:

- Постарайтесь аргументы, необходимые для запуска скриптов выносить в cli-аргументы (через argparse, click в питоне или через аргументы в bash); Для запуска скрипта **разработчик не должен лезть в код и хардкодить переменные**.
- Для создания конфигов с гиперпараметрами рекомендуется использовать **json**, либо **yaml**. Отдельно при использовании yaml-конфига, рекомендую посмотреть библиотеку OmegaConf;
- Рекомендую потренироваться и выполнить загрузку и выгрузку данных из S3 при помощи bash скриптов. (но это только рекомендация! На деле можно использовать питон или ваш любимый ЯП);