

Домашка №2

- В проекте ДЗ №1 **измените менеджер зависимостей с pip на ix (или poetry)**
- Выберите набор данных, с которым вы будете работать (например, с Kaggle). Это может быть датасет титаник, датасет с заболеванием диабетом, etc. Формат тоже произвольный csv, json, xml, etc. Идея домашки: потрогать инструменты
- В проекте из ДЗ №2 поддержите возможность работы с S3. Это может быть
 - Поднятый рядом контейнер с S3, например minio (рекомендуется)
 - S3 в облаке при желании
- Датасет из второго пункта положите в S3 as-is, т.е. в сыром виде
- Реализуйте в проекте скрипт со следующей логикой:
 - Из S3 скачивается датасет локально (вспомните, куда по шаблону кладутся сырые данные!)
 - Над данными производится некоторая обработка (произвольная, ее смысл сейчас не важен, хоть нормализация) и данные сохраняются в новый файл
 - Полученный файл выгружается обратно в S3
- Важно! Хотя скрипт, который выполняет логику пункта выше, должен запускаться одной «одной кнопкой», **постарайтесь сделать ваш код модульным** и изолировать различные шаги (например, отдельно скрипт скачивания и отправки данных в S3 и отдельно скрипт обработки)
- Важно! Следуйте рекомендациям вашего шаблона проекта, за нарушение структуры проверяющий оставляет за собой право снизить баллы

Алгоритм проверки домашки:

- Проверяющий клонирует репозиторий с проектом и разворачивает окружение по инструкции из README.md (или с помощью запуска bash/Makefile сценария)
- Проверяющий убеждается, что контейнер S3 поднялся и базовый датасет в нем загружен
- Проверяющий по инструкции запускает скрипт с необходимой логикой и проверяет результаты работы

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 основную 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:

- Для работы и взаимодействия с S3 в питоне можно взять библиотеку boto3, для работы из bash скриптов s3cmd. Для просмотра содержимого в S3 рекомендую использовать s3cmd или приложение с интерфейсом Cyberduck.
- 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)

- 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**.