

ТЗ ДЗ

Задача

Построить модель, согласно заветам MLOps. Это может быть как модель машинного обучения, так и нейросеть.

Подзадачи

Обучение модели

1. Найти интересующий вас датасет (если нет пожеланий, можете взять датасет из своего варианта для лабы №2)
2. Написать скрипт тренировки модели в .py файлах. Именно в .py файлах, а не в юпитер ноутбуках .ipynb. Юпитер ноутбуки можно использовать для прототипирования, тестирования гипотез, создания черновых вариантов скриптов. Best practice решением является именно составление пайплайна из .py файлов.
3. Согласно другим best practice, у проекта должна быть внятная и аккуратная архитектура папок. Для этого советую воспользоваться cookiecutter ds (<https://drivendata.github.io/cookiecutter-data-science/>)
4. Для написания пайплайнов советую пользоваться DVC (data version control, <https://dvc.org>). Фреймворк позволяет создавать воспроизводимые пайплайны для ML задач, версионировать данные и хранить их на множестве удаленных хранилищ (HDFS, S3, GDrive и пр.)
5. Тренировка модели должна осуществляться одной командой. `python src/train_pipeline.py` приемлемо, но я буду ждать оформление пайплайна через dvc. Пайплайн будет представлять собой последовательность стейджей, которые отслеживают изменения входных файлов, а так же файлов кода (<https://dvc.org/doc/start/data-management/data-pipelines>). Один стейдж - одна логически обособленная операция во всей последовательности тренировки модели. Условно так: "загрузка данных" -> "предобработка данных" -> "тренировка модели" -> "тестирование модели (сбор метрик)"

Инференс модели

На базе обученной модели необходимо будет организовать сервис, в который пользователь сможет отправлять запросы / фотографии / видео / etc (в зависимости от задачи).

Здесь крайне советую обернуть это дело в докер, так как это очередной best practice для инференса модели. Супер красивый сервис я ждать не буду, достаточно будет простого сервиса, чтобы отправить запрос.

Чек-лист успешного выполнения задания:

- ☐ Репозиторий в github/gitlab с кодом тренировки и инференса модели.
- ☐ Код соответствует требованиям PEP (написаны док-стринги, с отступами и пр.), в общем, красивый и читабельный код. Проверять качество кода буду через pylint, нижней границей качества будет 7 из 10 баллов.
- ☐ Скачав репозиторий и данные, я смогу вручную перезапустить весь пайплайн с помощью парочки команд в консоли (хоть у меня и есть в наличии GPU, лучше не добиваться 99.999% качества на терабайте данных, достаточно модели, которая сколько-нибудь сможет решать поставленную задачу).
- ☐ Имея на руках код, я могу локально поднять сервис, в который смогу отправить запрос и получить предсказание модели.