Отчет по лабораторной работе №2

Выполнил: Казаркин Тимофей, гр. 6232

**Apache Airflow, инференс модели**

Ход выполнения первого задания:

- Был сконфигурирован DAG-файл для выполнения пайплайна обработки данных и последующего инференса моделей;
- Для выполнения данного задания мною был выбран самый простой путь – воспользоваться Hugging Face API для инференса моделей;
- В качестве моделей были выбраны openai/whisper-small и Falsconsai/text_summarization, т.к. обладали возможностью инференса по API;
- Для получения результатов по API была выбрана библиотека requests и был использован Docker-образ nyurik/alpine-python3-requests;
- Для вывода результатов в PDF-файл была использована библиотека fpdf и Docker-образ bikc/report:1.1;
- Полученные результаты представлены в директории model_inference;

**Apache Airflow, обучение модели**

Ход выполнения второго задания:

- После множества попыток запустить Docker-in-Docker с доступом к GPU хоста, мне удалось разобраться с этим вопросом. В частности, была изменена конфигурация docker-compose, а именно один сервисов – docker proxy. Для успешной работы с GPU хоста был выбран образ, имеющий Nvidia CUDA и Docker-in-Docker конфигурацию – fvt34u/nvidia-dind:12.4, собранный мною образ с CUDA 12.4 на основе образа https://github.com/Extrality/nvidia-dind.
- Было реализовано 2 задания для DAG, одно по преобразованию одно типа датасета в другой, удовлетворяющий требования для обучения

YOLO-модели (образ fvt34u/python-pillow), и второе по обучению самой модели (образ ultralytics/ultralytics);

- Также был проведён инференс обученной модели, результаты которого можно увидеть в файле inference.ipynb;

Отдельно хотелось бы отметить пару моментов:

- Все конфигурационные файлы, __pycache__ и файлы виртуального окружения были исключены из репозитория путём добавления .gitignore-файла;
- Собранные мою образы можно найти на Docker Hub;
- Для запуска контейнеров с поддержкой GPU в DAG был добавлен параметр device_requests, а также параметр shm_size для увеличения количества разделяемой памяти.
- Наибольшие проблемы доставил тот факт, что хост-системой для запуска DockerOperator является именно docker proxy, это означает, что необходимые зависимости должны находится именно там (по крайней мере в моём случае наличие cuda на моём устройстве было недостаточно, было необходимо поставить её на docker proxy).