Лабораторная работа 2.

1. Инференс.

Был реализован пайплайн вида wait_for_new_file >> extract_audio >> transform_audio_to_text >> summarize_text >> save_to_pdf.

FileSensor(wait_for_new_file) сканирует папку и ожидает появления в ней нового файла.

DockerOperator(extract_audio) извлекает аудиодорожку из файла.

DockerOperator(transform_audio_to_text) использует API HuggingFace для перевода аудиофайла в текстовый. Тут я использовал готовый Docker образ nyurik/alpine-python3-requests

DockerOperator(summarize_text) делает конспект по сформированному текстовому файлу также с помощью API HuggingFace и Docker образа nyurik/alpine-python3-requests

DockerOperator(save_to_pdf) сохраняет конспекта в формат пдф с помощью библиотеки fpdf, для этого пришлось создать свой Docker образ и загрузить его на dock-hub.

Основной сложностью являлась работа с Docker образами, так как изначально я пытался создать свой собственный для всего задания, но у меня закончилось место на диске и сам этот процесс занимал очень много времени.

2. Обучение модели.

Был реализован пайплайн вида prepare_data >> train_model

Для выполнения задания я создал Docker образ на основе образа tensorflow/tensorflow: latest-py3 и использовал стандартный набор данных mnist.

DockerOperator(prepare_data) выгружал данные из mnist и сохранял их на компьютер.

DockerOperator(train_model) считывал данные и запускал обучение модели.

Основной сложностью, как и в первом задании, являлась работа с Docker образами.

UPD: также сложностью неожиданно оказалась загрузка данных на гитхаб, а именно данных из mnist, так как гитхаб не позволяет загружать файлы

размером более 50 MB, а использование Git Large File Storage запрещено для публичных форков. Так что для загрузки на гитхаб мне пришлось уменьшить размеры обучающей и тестовой выборок, но при выполнении лабораторной работы я использовал данные изначального размера.