БУЛДАКОВ ДМИТРИЙ

6233-010402D

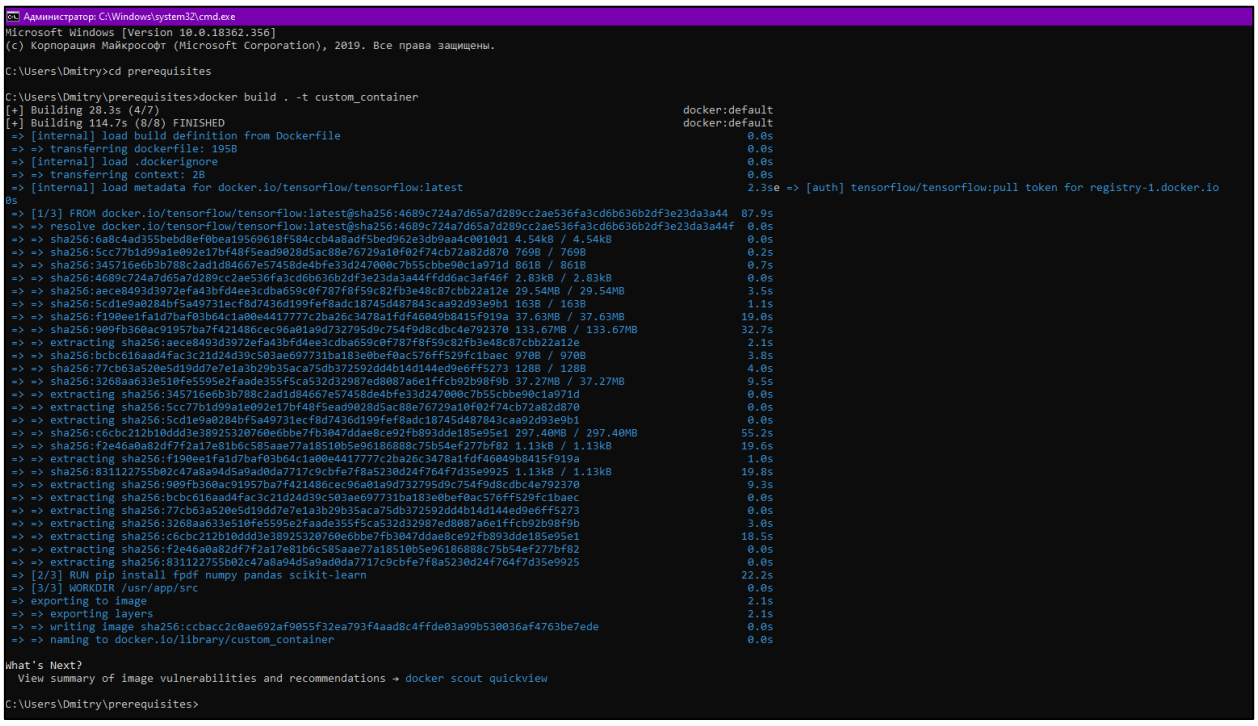
ХОД РАБОТЫ

Пайплайн для инференса данных

Перед началом работы требовалось создать собственный образ (или взять существующий) контейнера с необходимыми библиотеками для дальнейшей работы с первым и вторым заданием. Получился следующий докер-файл.

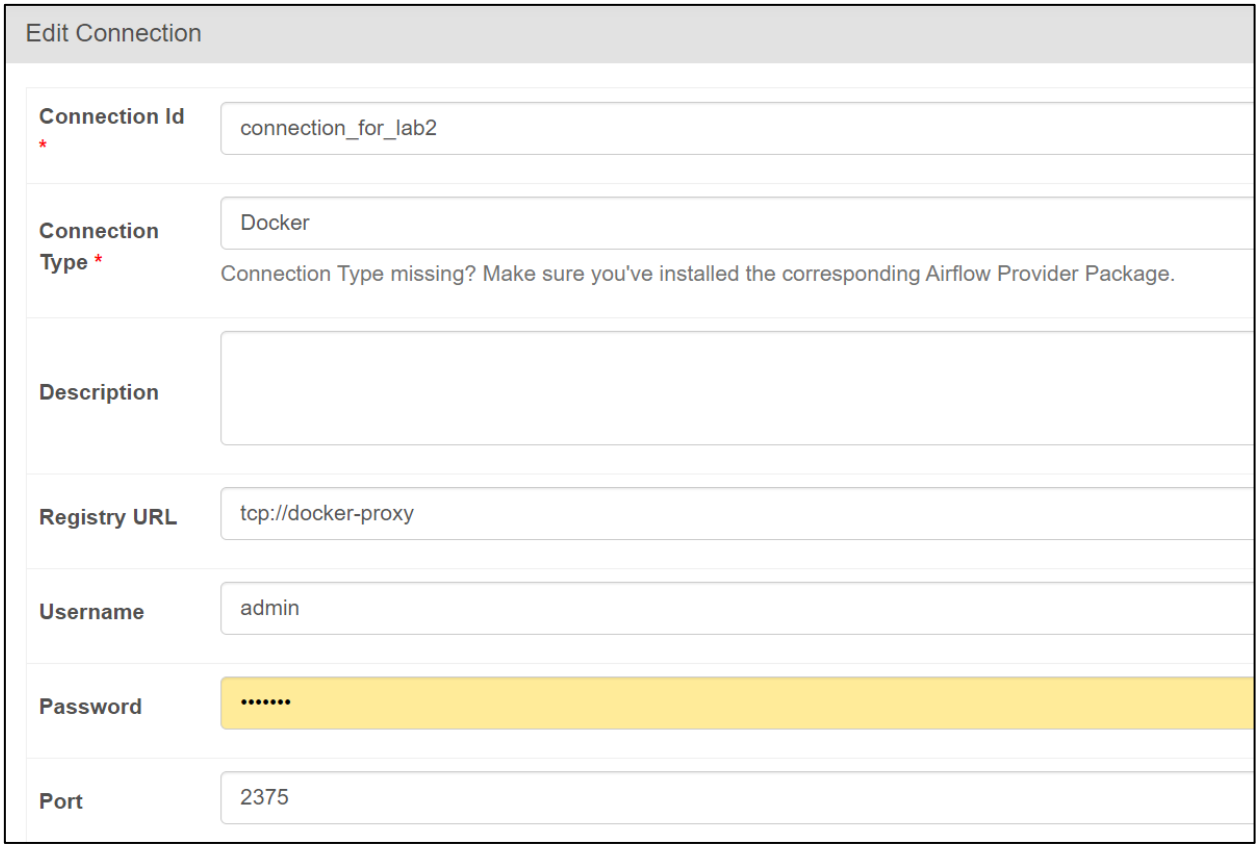
```
FROM tensorflow/tensorflow:latest

RUN pip install fpdf numpy pandas scikit-learn

LABEL Maintainer="vapaov.tensorflow_sklearn"

WORKDIR /usr/app/src
```

Затем образ собрали, присвоили тэг и отправили в докер-хаб.

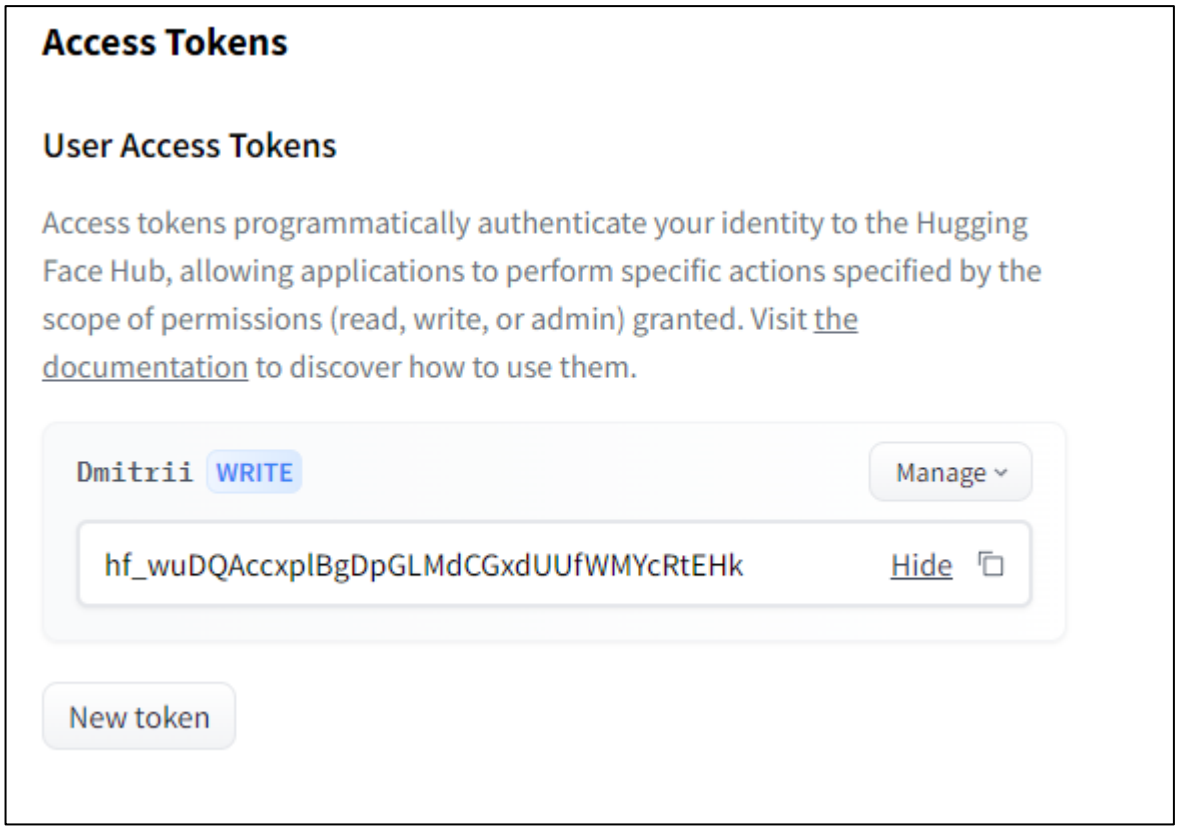


```
C:\Users\Dmitry\prerequisites>docker images
REPOSITORY                    TAG         IMAGE ID       CREATED              SIZE
custom_container              latest      ccbacc2c0ae6   About a minute ago   2.1GB
jenkins/jenkins               latest      190891a74cb9   13 days ago          476MB
mlflow-web                    latest      05851fea9392   2 weeks ago          765MB
airflow-airflow-triggerer     latest      87aba933ed10   2 weeks ago          1.82GB
airflow-airflow-worker        latest      2882a6206134   2 weeks ago          1.82GB
airflow-airflow-webserver     latest      d9219ed5d28b   2 weeks ago          1.82GB
airflow-airflow-init          latest      72a245a208c7   2 weeks ago          1.82GB
airflow-airflow-scheduler     latest      7c75bf149bef   2 weeks ago          1.82GB
minio/mc                      latest      a50d857f0878   3 weeks ago          76.9MB
minio/minio                   latest      603e753a418c   3 weeks ago          147MB
postgres                      15          30822e498539   3 weeks ago          419MB
postgres                      13          d28d5ae8775c   3 weeks ago          413MB
redis                         latest      7f27d60cb8e0   4 weeks ago          138MB
docker                        24-dind     5b912308540a   5 weeks ago          337MB
apache/nifi                   1.23.2      81455911cd05   3 months ago         1.94GB
dpage/pgadmin4                7.6         881febbc9e93   3 months ago         534MB
nshou/elasticsearch-kibana    kibana7     17f031ca3406   7 months ago         1.18GB
jrottenberg/ffmpeg            3-scratch   b76f13747dc1   8 months ago         63.5MB
mysql/mysql-server            5.7.28      c8c8ef4f3c81   4 years ago          310MB
jenkins                       2.60.3      cd14cecfdb3a   5 years ago          696MB

C:\Users\Dmitry\prerequisites>docker tag custom_container vapaov/custom_container:1.0

C:\Users\Dmitry\prerequisites>docker push vapaov/custom_container:1.0
[2023-12-04T16:08:27.185298900Z][docker-credential-desktop.system][W] Windows version might not be up-to-date: The system cannot find the file specified.
The push refers to repository [docker.io/vapaov/custom_container]
88997c67609c: Pushed
4574ed6e39d8: Pushed
75acb1242fe3: Mounted from tensorflow/tensorflow
1d7a2a211a6b: Mounted from tensorflow/tensorflow
2db699de670e: Mounted from tensorflow/tensorflow
0cf31f98a4b6: Mounted from tensorflow/tensorflow
f663f4c9c5b6: Mounted from tensorflow/tensorflow
104e4c35057a: Mounted from tensorflow/tensorflow
eb864c00a034: Mounted from tensorflow/tensorflow
94235a128255: Mounted from tensorflow/tensorflow
0ac81db158f3: Mounted from tensorflow/tensorflow
8e8c3d39273b: Mounted from tensorflow/tensorflow
f99aba8580cb: Mounted from tensorflow/tensorflow
256d88da4185: Mounted from tensorflow/tensorflow
1.0: digest: sha256:185d137138037255d0f0b20ec625ff60f6b90e8cacb89ff414cec93635a081d9 size: 3250
```

Далее необходимо создать новое подключение в Airflow. Заполненные поля соединения на скриншоте ниже.



В рамках первого задание будем использовать API Hugging Face. Для доступа к API требуется регистрация и получение специального токена.



DAG-файл получился длинным и адекватно в отчёт его не вставить. Можно посмотреть в репозитории: video_to_summary_pdf.py. Для пайплайна были написаны специальные функции.

1_audio_to_text.py – переводит звуковую дорожку файла в текст при помощи сервиса openai/whisper-small

```python
import requests
url = "https://api-inference.huggingface.co/models/openai/whisper-small"
headers = {"Authorization": "Bearer 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"}

with open('/data/audio.aac', "rb") as file:
    data = file.read()
    response = requests.post(url, headers=headers, data=data)
    result = response.json()
    text_from_audio = result['text']
    text_file = open("/data/text.txt", "w+")
    text_file.write(text_from_audio)
    text_file.close()
```

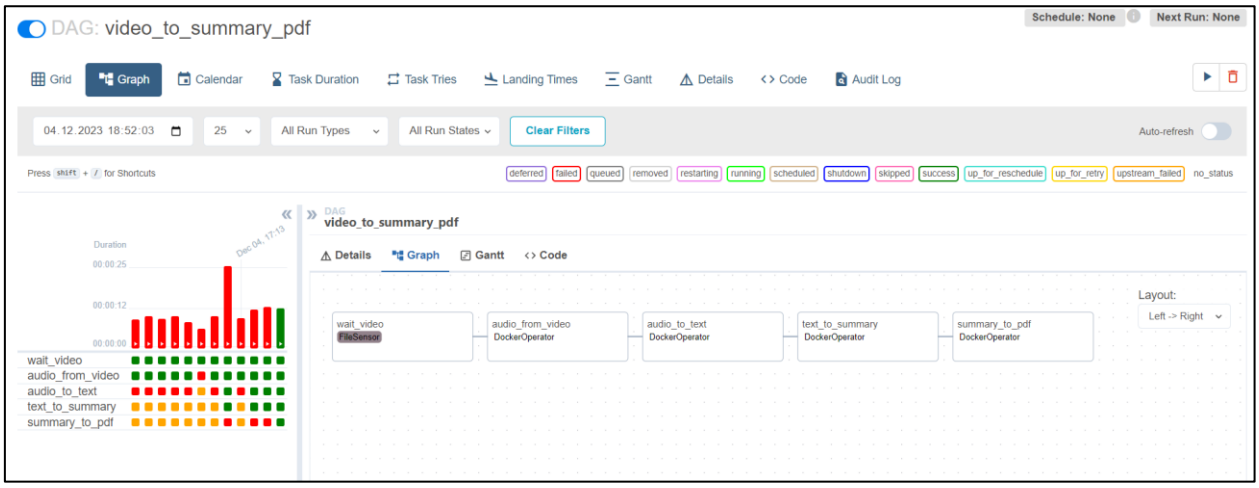
2_text_to_summary.py – кратко пересказывает написанное в текстовом файле. Используется сервис slauw87/bart_summarisation.

```python
import requests
url = "https://api-inference.huggingface.co/models/slauw87/bart_summarisation"
headers = {"Authorization": "Bearer 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"}

with open('/data/text.txt', "rb") as file:
    data = file.read()
    response = requests.post(url, headers=headers, json={'inputs': f"{data}",})
    result = response.json()
    summary = result[0]['summary_text']
    text_file = open("/data/summary.txt", "w+")
    text_file.write(summary)
    text_file.close()
```

3_summary_to_pdf.py – с помощью библиотеки FPDF формирует файл формата PDF.

```python
from fpdf import FPDF

file = open("/data/summary.txt","r")
pdf = FPDF()
pdf.add_page()
for text in file:
    pdf.set_font('Helvetica', '', size=14)
    pdf.multi_cell(w=100, h=10, border = 1, txt = text, align = 'J')
    pdf.output("/data/pdf_result.pdf")
```

После всех настроек и нескольких попыток, пайплайн успешно запустился.



В результате получился PDF файл с следующим содержанием:

Mark will buy a new shirt for Mark. Mark and Mark will have a coffee and then go shopping. Mark has a meeting at 12.30, so they have a lot of time to go shopping before the next meeting. Mark will be back in 10 minutes.

Пайплайн для обучения модели

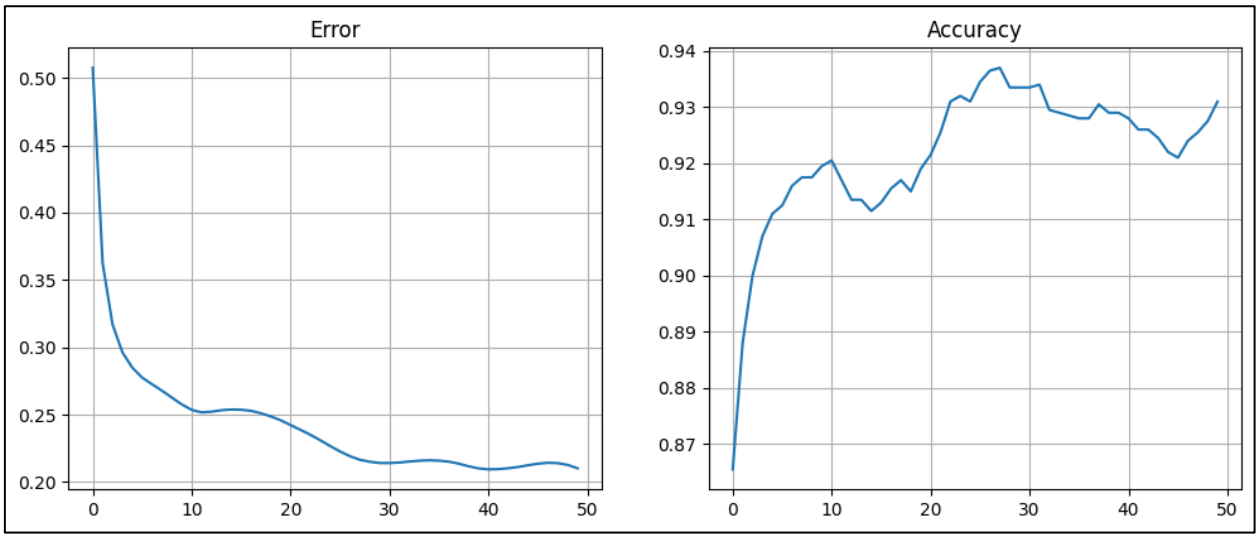
Для обучения модели написал небольшой DAG.

```python
from datetime import datetime
from airflow import DAG
from docker.types import Mount
from airflow.providers.docker.operators.docker import DockerOperator

default_args = {
    'owner': 'airflow',
    'start_date': datetime(2023, 12, 3),
}

dag = DAG(
    'train_mnist_model',
    default_args=default_args,
    schedule_interval=None,
)

train_model = DockerOperator(
    task_id='train_model',
    image='vapaov/custom_container:1.0',
    command='python /data/nn_mnist.py',
    mounts=[Mount(source='/data', target='/data', type='bind')],
    docker_url="tcp://docker-proxy:2375",
    dag=dag,
)

train_model
```

Я взял нейросеть, написанную мною в рамках курса «Нейронные сети и машинное обучение». Она имеет довольно простой алгоритм и невысокую точность, но, тем не менее, пайплайн я захотел протестировать именно на ней. Обучение происходит на MNIST-датасете. К сожалению гитхаб не позволяет загружать файлы больше 25 МБ.

Ссылка на датасет: https://www.kaggle.com/datasets/oddrationale/mnist-in-csv

На скриншоте ниже можно наблюдать графики функции потерь и точности.



В mnist_nn.py с помощью библиотеки logging реализовал логирование. После успешного запуска пайплайна логи можно посмотреть в файле log.log