

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»
(Самарский университет)

Институт информатики и кибернетики
Факультет информатики
Кафедра технической кибернетики

Отчет по лабораторной работе №2

Дисциплина: «Инженерия данных»

Тема: «Инференс и обучение НС»

Выполнил: Мелешенко И.С.

Группа: 6233-010402D

Самара 2023

Часть 1. Построение пайплайн для инференса данных.

Шаг 1. Разработка и реализация DAG-а

В рамках первого задания необходимо реализовать пайплайн, который реализует систему "Автоматического распознавания речи" для видеофайлов.

Построенный пайплайн будет выполнять следующие действия поочередно:

- Производить мониторинг целевой папки на предмет появления новых видеофайлов.
- Извлекать аудиодорожку из исходного видеофайла.
- Преобразовывать аудиодорожку в текст с помощью нейросетевой модели.
- Формировать конспект на основе полученного текста.
- Формировать выходной .pdf файл с конспектом.

Для реализации описанных действий мы будем использовать DockerOperator, а также FileSensor для получения необходимого видеофайла.

Для работы task-а по ожиданию получения нового видео необходимо создать новое подключение к airflow. Для создания подключения переходим в Airflow по адресу <http://localhost:8080/connection/list/> или мы можем в Airflow пройти по пути Admin>>Connections, как на рисунке ниже.

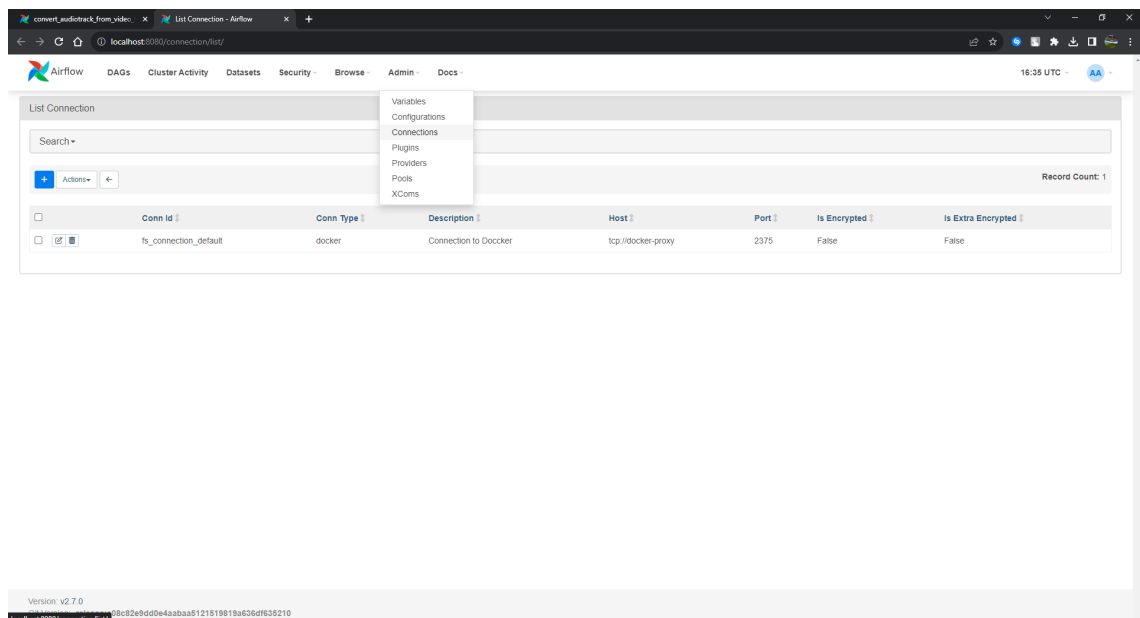


Рисунок 1 – Создание Connection

Connection Id *

Connection Type *
Connection Type missing? Make sure you've installed the corresponding Airflow Provider Package.

Description

Registry URL

Username

Password

Port

Extra

```
{  
  "reauth": false  
}
```

Рисунок 2 – Параметры Connection

Шаг 2. Регистрация на huggingface и получения токена API.

Далее для того, чтобы можно было преобразовать наш аудиофайл в текст, а после получить из него summary, необходимо зарегистрироваться на <https://huggingface.co/> и получить токен API с правами записи для возможности отправки и получения запросов к сайту.

Join Hugging Face
Join the community of machine learners!

Email Address

Hint: Use your organization email to easily find and join your company/team org.

Password

Next

Already have an account? [Log in](#)
SSO is available for companies

Рисунок 3 – Регистрация на huggingface

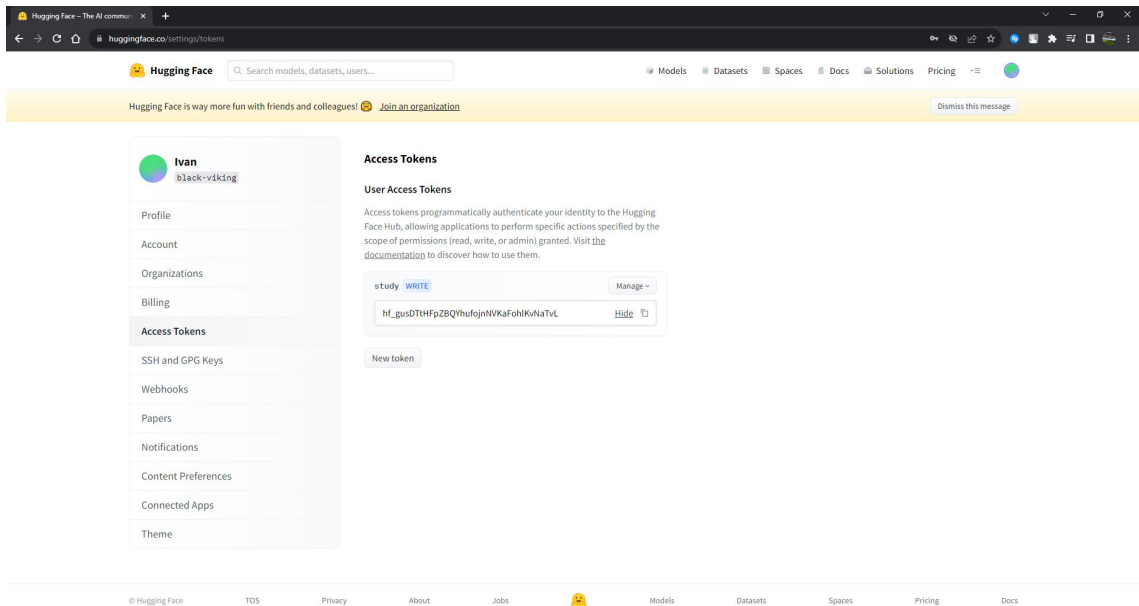


Рисунок 4 – Получение токена API

Теперь после всех необходимых настроек можем запустить наш DAG.

Переходим в airflow: <http://localhost:8080/home>

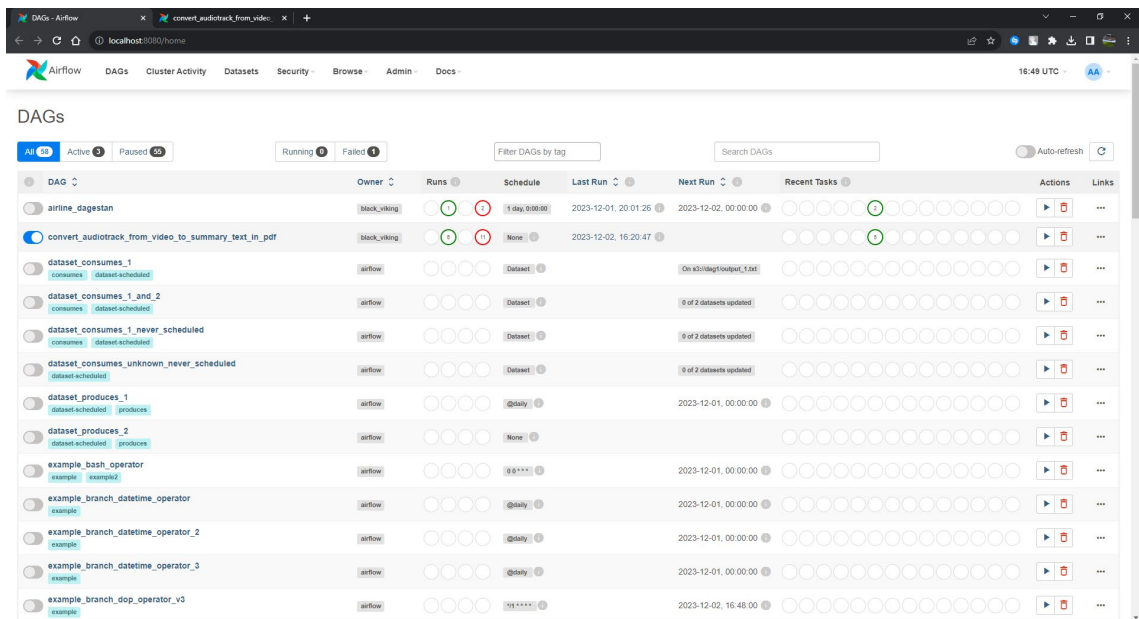


Рисунок 5 – Поиск DAG-а.

Далее запускаем наш DAG и наслаждаемся процессом.

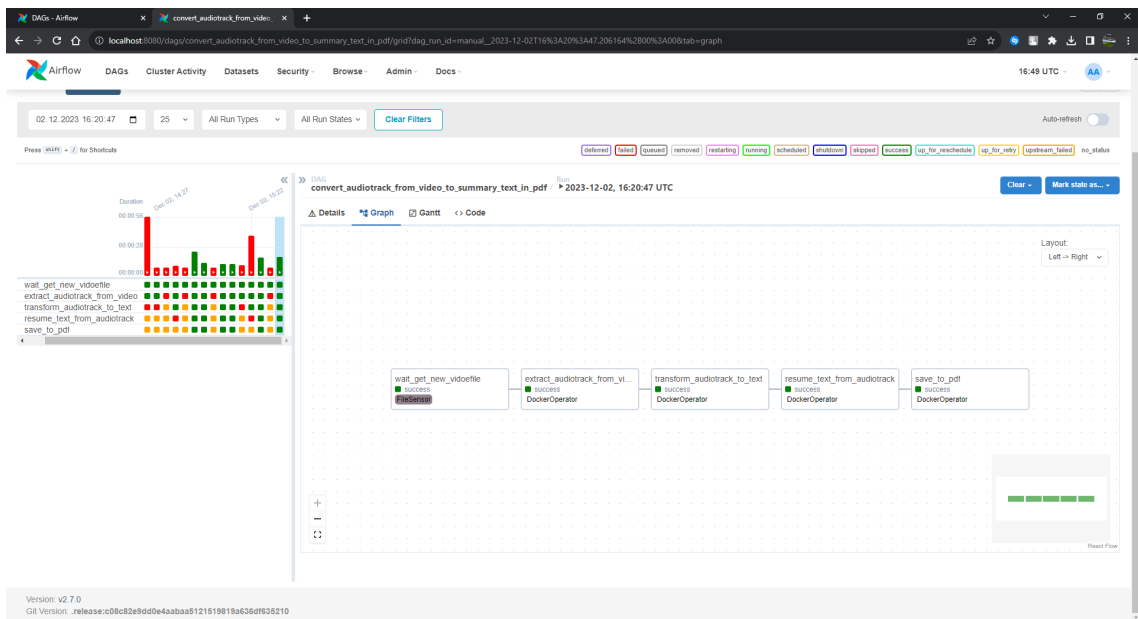


Рисунок 6 – Запуск DAG-а.

Для сохранения конспекта в PDF, необходимо было использовать библиотеку `fpdf`. Создадим необходимый для этого образ в Docker. Процесс представлен ниже.

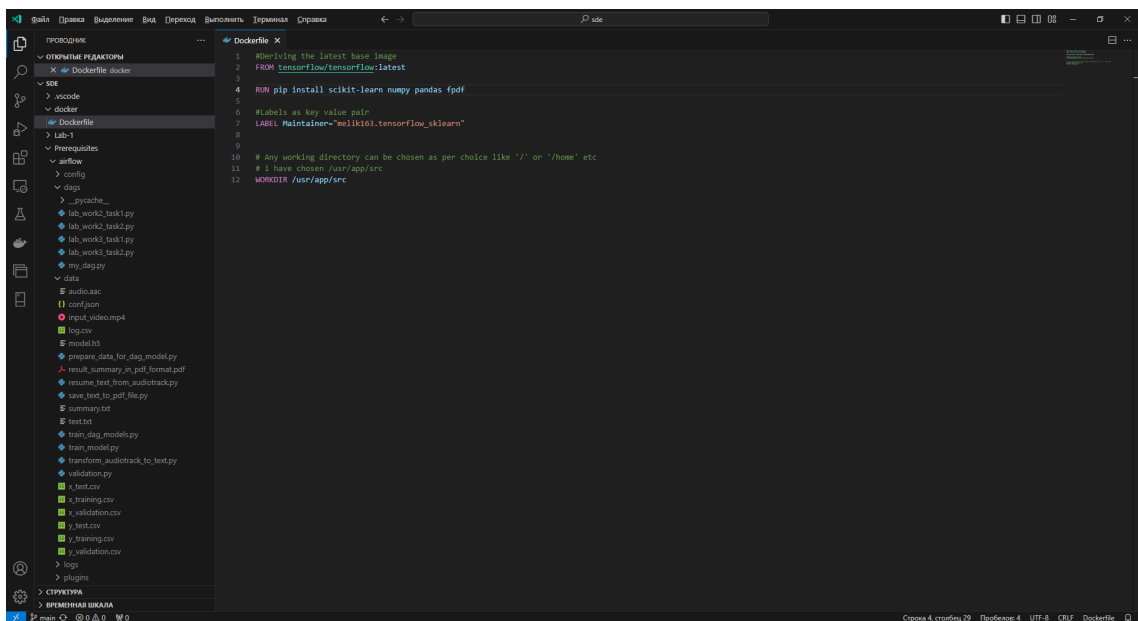


Рисунок 7 – Создание Dockerfile

```
Windows PowerShell
PS D:\sde\docker> docker build . -t our_tensorflow_container
[+] Building 22.0s (7/7) FINISHED
=> [internal] load build definition from Dockerfile                                docker:default 0.0s
=> => transferring dockerfile: 365B                                             0.0s
=> [internal] load .dockerignore                                                0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/tensorflow/tensorflow:latest         0.0s
=> CACHED [1/3] FROM docker.io/tensorflow/tensorflow:latest                  0.0s
=> [2/3] RUN pip install scikit-learn numpy pandas fpdf                       20.2s
=> [3/3] WORKDIR /usr/app/src                                                  0.0s
=> exporting to image                                                          1.7s
=> => exporting layers                                                         1.7s
=> writing image sha256:7685cbf0ad8744e3a02f15b6517b8fde0657a1f4fb9f398eba764c67b5c30a03 0.0s
=> => naming to docker.io/library/our_tensorflow_container                    0.0s

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS D:\sde\docker> |
```

Рисунок 8 – Сборка образа

```
Windows PowerShell
=> exporting to image                                                          1.7s
=> => exporting layers                                                         1.7s
=> writing image sha256:7685cbf0ad8744e3a02f15b6517b8fde0657a1f4fb9f398eba764c67b5c30a03 0.0s
=> naming to docker.io/library/our_tensorflow_container                    0.0s

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS D:\sde\docker> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
our_tensorflow_container  latest             7685cbf0ad87       About a minute ago  2.1GB
mlflow-web          latest             da035e44318c       4 days ago         765MB
airflow-airflow-triggerer latest             4df00f2bf0d5       4 days ago         1.82GB
airflow-airflow-worker latest             3d8f4b21596c       4 days ago         1.82GB
airflow-airflow-init latest             01d790f408e0       4 days ago         1.82GB
airflow-airflow-webserver latest             31c323886c61       4 days ago         1.82GB
airflow-airflow-scheduler latest             0a66fdf03f9a       4 days ago         1.82GB
tensorflow/tensorflow latest             6a8c4ad355be       11 days ago        1.78GB
minio/minio         latest             88c665b1183a       12 days ago        147MB
minio/mc             latest             eaa326464fd5       12 days ago        76.9MB
sasha151299/my_pdf  1.0               0a6eaffde1ba       3 weeks ago        1.12GB
redis               latest             961dda256baa       3 weeks ago        138MB
postgres            15                8cde386e2e85       3 weeks ago        419MB
postgres            13                19975f71ce75       3 weeks ago        413MB
docker              24-dind           daefcf9ccf3b       5 weeks ago        336MB
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
mysql/mysql-server  5.7.28            c8c8ef4f3c81       4 years ago        310MB
PS D:\sde\docker> docker tag our_tensorflow_container melik163/our_tensorflow_container:1.0
PS D:\sde\docker> |
```

Рисунок 9 – Присвоение тега образу

```
Windows PowerShell
tensorflow/tensorflow      latest      6a8c4ad355be 11 days ago 1.78GB
minio/minio                latest      88c665b1183a 12 days ago 147MB
minio/mc                   latest      eaa326464fd5 12 days ago 76.9MB
sasha151299/my_pdf        1.0        0a6eaffde1ba 3 weeks ago 1.12GB
redis                     latest      961dda256baa 3 weeks ago 138MB
postgres                  15         8cde386e2e85 3 weeks ago 419MB
postgres                  13         19975f71ce75 3 weeks ago 413MB
docker                    24-dind    daefcf9ccf3b 5 weeks ago 336MB
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
mysql/mysql-server        5.7.28     c8c8ef4f3c81 4 years ago 310MB
PS D:\sde\docker> docker push melik163/our_tensorflow_container:1.0
The push refers to repository [docker.io/melik163/our_tensorflow_container]
e1f360959148: Pushed
06e66be4628b: 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: sha256:2400063992556f7cc9612f9eb0feb9d0589f6656eb255a6ff179878d422e3737 size: 3250
PS D:\sde\docker> |
```

Рисунок 10 – Отправка образа в DockerHub

В качестве исходного видео использовался фрагмент из кинофильма «Крестный отец» длительностью 3 минуты 11 секунд. После чего мы получали аудиодорожку, которая использовалась в качестве основы для получения текстового файла.

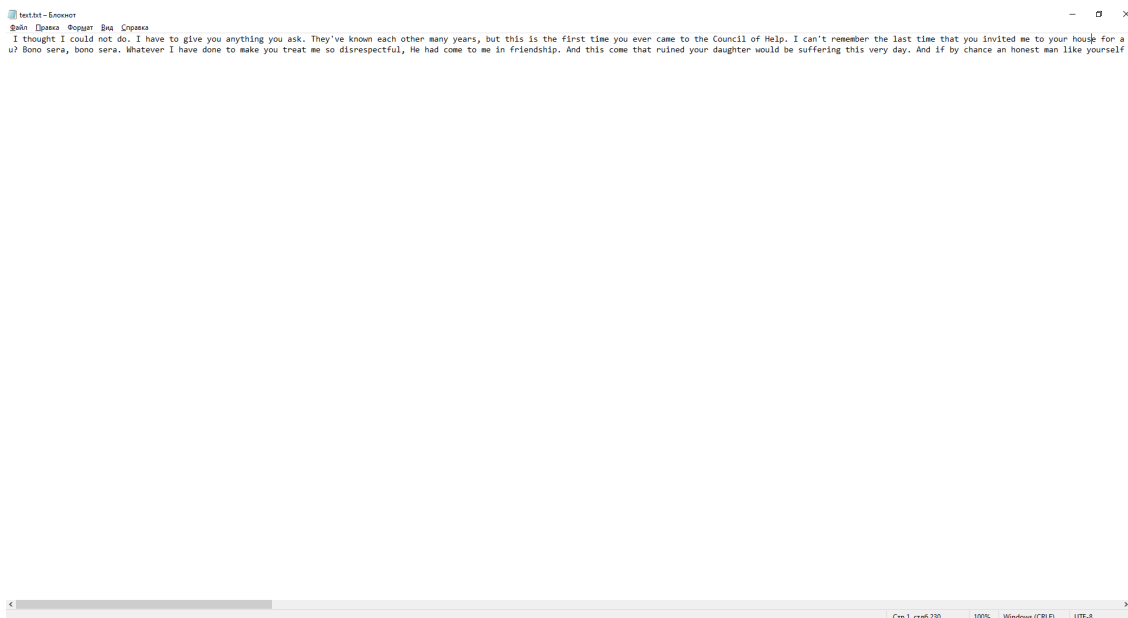


Рисунок 11 – Результат работы huggingface по преобразованию аудио в текст

После чего полученный результат мы еще раз передавали huggingface для получения уже конспекта по отправленному нами файлу. Полученный результат записывали pdf-файл.

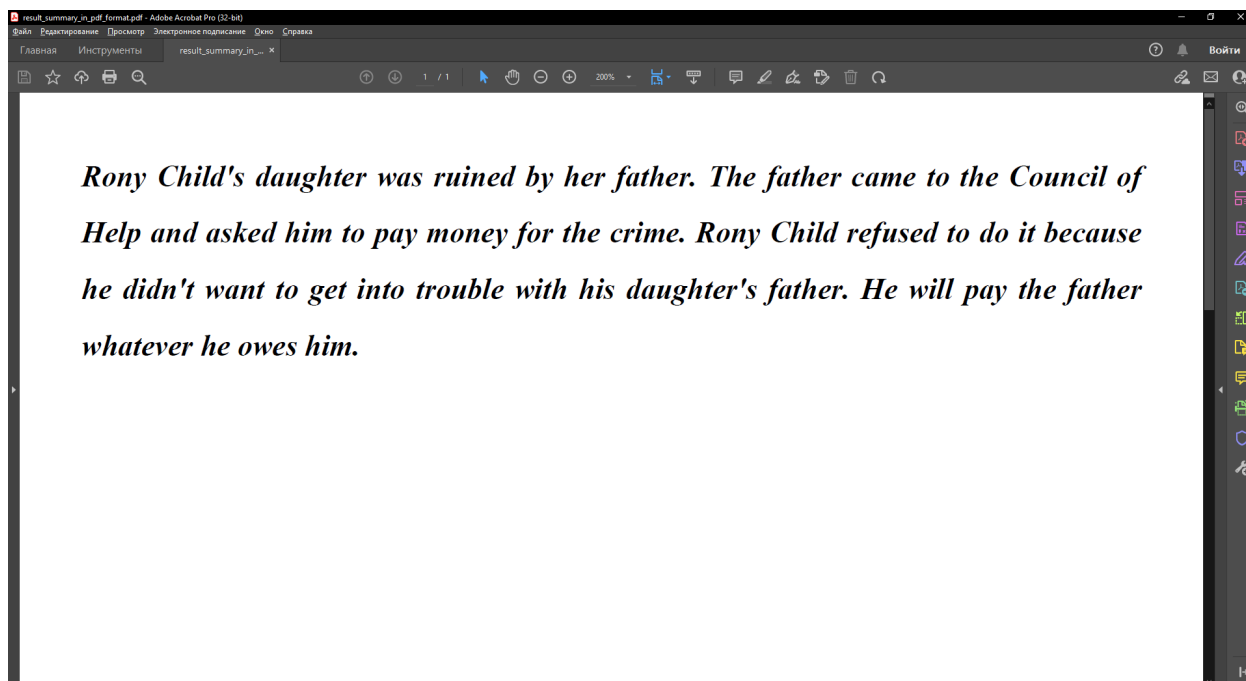


Рисунок 12 – Конспект текстового файла.

Получилось неплохо. Перейдем ко второй части.

Часть 2. Пайплайн, который реализует систему автоматического обучения/дообучения нейросетевой модели

В рамках второй части лабораторной работы нам необходимо было разработать пайплайн, который реализует систему автоматического обучения/дообучения нейросетевой модели.

Для этого мы разработали DAG

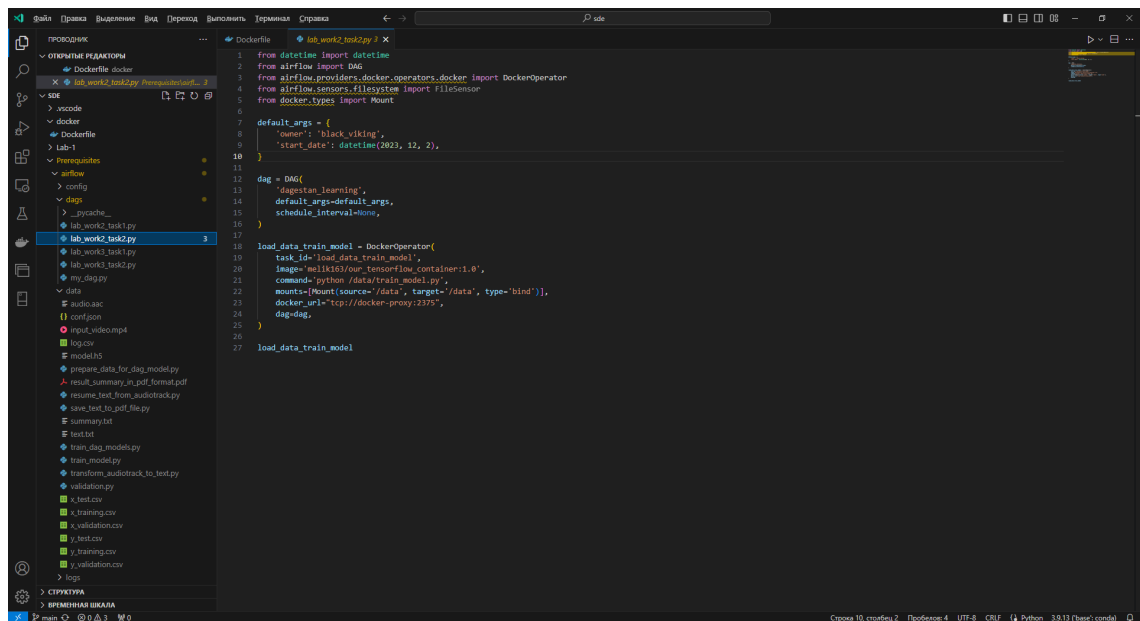
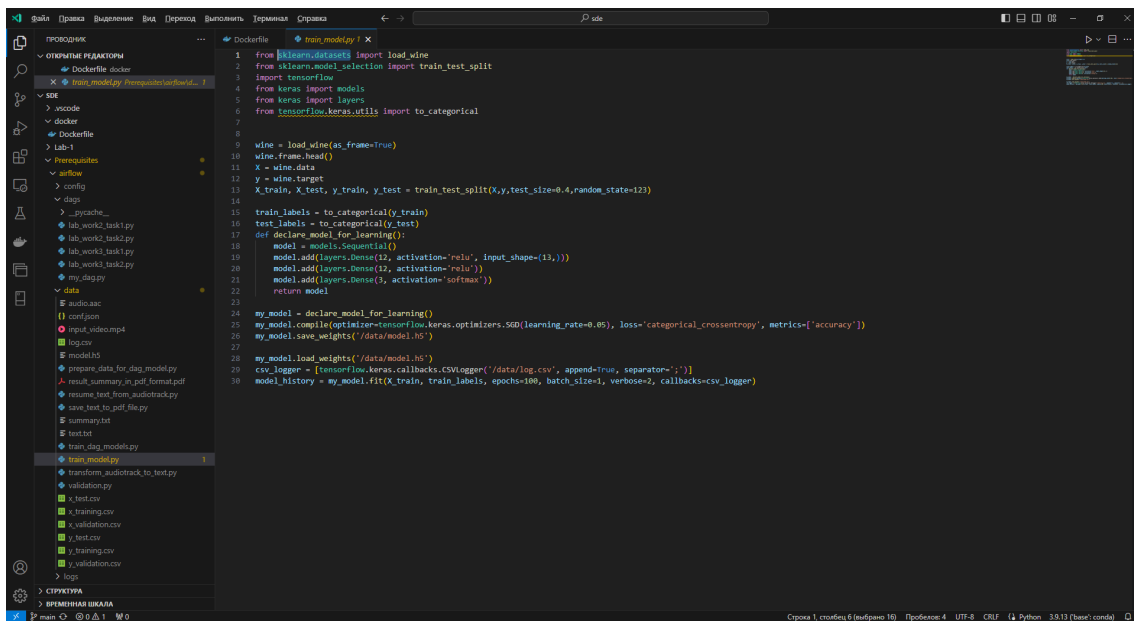


Рисунок 13 - Пайплайн

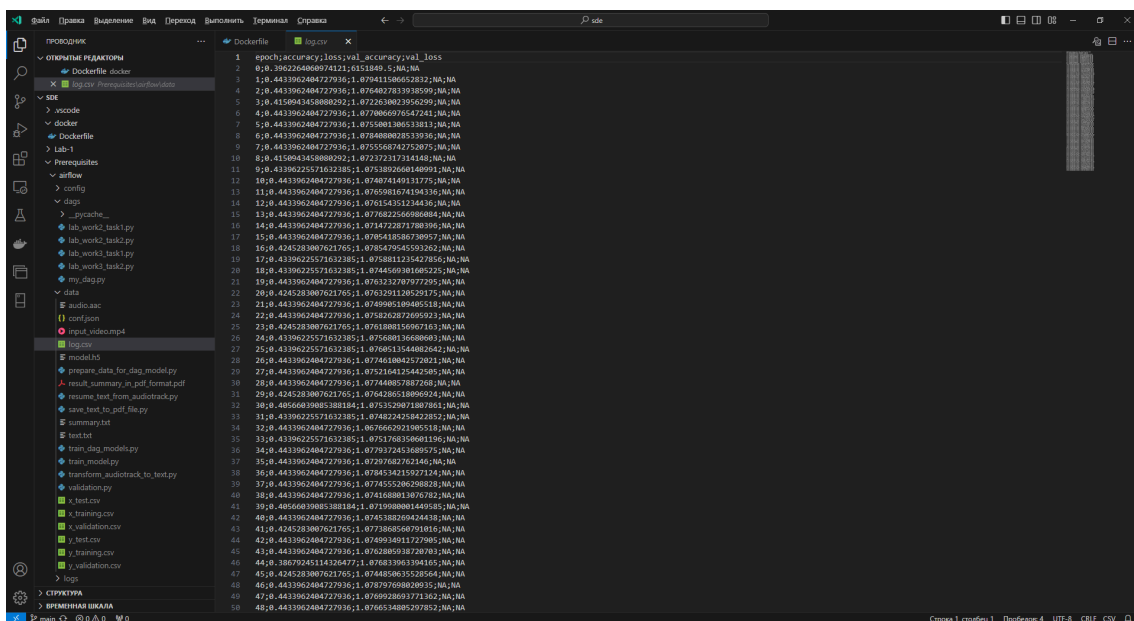
DAG запускал код, который получал датасет вин `load_wine` из `sklearn.datasets`, после чего мы проводили разбиение данных. Которые передаются в нейросеть, после чего модель проходит обучение. Процесс обучения логируется.



```
1 from sklearn.datasets import load_wine
2 from sklearn.model_selection import train_test_split
3 import tensorflow
4 from keras import models
5 from keras import layers
6 from tensorflow.keras.utils import to_categorical
7
8 wine = load_wine(as_frame=True)
9 wine_frame = wine.frame
10 wine_frame.head()
11 X = wine.data
12 y = wine.target
13 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=123)
14
15 train_labels = to_categorical(y_train)
16 test_labels = to_categorical(y_test)
17
18 def declare_model_for_learning():
19     model = models.Sequential()
20     model.add(layers.Dense(12, activation='relu', input_shape=(13,)))
21     model.add(layers.Dense(12, activation='relu'))
22     model.add(layers.Dense(3, activation='softmax'))
23     return model
24
25 my_model = declare_model_for_learning()
26 my_model.compile(optimizer='tensorflow.keras.optimizers.Adam(learning_rate=0.05)', loss='categorical_crossentropy', metrics=['accuracy'])
27 my_model.save_weights('data/model.h5')
28
29 my_model.load_weights('data/model.h5')
30
31 logger = [tensorflow.keras.callbacks.CSVLogger('data/log.csv', append=True, separator=';')]
32
33 model_history = my_model.fit(X_train, train_labels, epochs=100, batch_size=1, verbose=2, callbacks=csv_logger)
```

Рисунок 14 – Код обучения модели.

В итоге получили вот такой лог обучения.



```
1 epoch: accuracy: loss: val_accuracy: val_loss
2 0.0.3962264066974121; 0.51849; 5; NA; NA
3 1.0.4433962404727936; 1.078411586652832; NA; NA
4 2.0.4433962404727936; 1.074002783393899; NA; NA
5 3.0.4150943458808292; 1.0722630023956299; NA; NA
6 4.0.4433962404727936; 1.0770066976547241; NA; NA
7 5.0.4433962404727936; 1.075308130831819; NA; NA
8 6.0.4433962404727936; 1.0784008083533936; NA; NA
9 7.0.4433962404727936; 1.0755568742752075; NA; NA
10 8.0.4150943458808292; 1.07237171314148; NA; NA
11 9.0.43396225571632385; 1.07538266414091; NA; NA
12 10.0.4433962404727936; 1.074074149131775; NA; NA
13 11.0.4433962404727936; 1.075981674104336; NA; NA
14 12.0.4433962404727936; 1.075153123436; NA; NA
15 13.0.4433962404727936; 1.077822566880884; NA; NA
16 14.0.4433962404727936; 1.0714722871788396; NA; NA
17 15.0.4433962404727936; 1.0780138589739957; NA; NA
18 16.0.4245283807621765; 1.0785479545592525; NA; NA
19 17.0.43396225571632385; 1.0758811235427856; NA; NA
20 18.0.43396225571632385; 1.074450930160225; NA; NA
21 19.0.4433962404727936; 1.07632170777295; NA; NA
22 20.0.4245283807621765; 1.0763291128529175; NA; NA
23 21.0.4433962404727936; 1.0749985189485518; NA; NA
24 22.0.4433962404727936; 1.075828292369933; NA; NA
25 23.0.4245283807621765; 1.0761808156967163; NA; NA
26 24.0.43396225571632385; 1.07568813686869; NA; NA
27 25.0.43396225571632385; 1.076051354406262; NA; NA
28 26.0.4433962404727936; 1.0774610045372621; NA; NA
29 27.0.4433962404727936; 1.075216412544585; NA; NA
30 28.0.4433962404727936; 1.077440857887268; NA; NA
31 29.0.4245283807621765; 1.07452051808624; NA; NA
32 30.0.40566939085388184; 1.0751529871807981; NA; NA
33 31.0.43396225571632385; 1.074822425842852; NA; NA
34 32.0.4433962404727936; 1.076666292106518; NA; NA
35 33.0.43396225571632385; 1.0751768350601196; NA; NA
36 34.0.4433962404727936; 1.0779372453689575; NA; NA
37 35.0.4433962404727936; 1.0729702782184; NA; NA
38 36.0.4433962404727936; 1.074514211932124; NA; NA
39 37.0.4433962404727936; 1.077455526298828; NA; NA
40 38.0.4433962404727936; 1.074168813076782; NA; NA
41 39.0.40566939085388184; 1.07150880144505; NA; NA
42 40.0.4433962404727936; 1.074538826044438; NA; NA
43 41.0.4245283807621765; 1.0773886580793816; NA; NA
44 42.0.4433962404727936; 1.07409934918723965; NA; NA
45 43.0.4433962404727936; 1.076280938728793; NA; NA
46 44.0.38079245114326477; 1.076833963394165; NA; NA
47 45.0.4245283807621765; 1.0740489863528956; NA; NA
48 46.0.4433962404727936; 1.07579708020935; NA; NA
49 47.0.4433962404727936; 1.0769928693771362; NA; NA
50 48.0.4433962404727936; 1.076653485297852; NA; NA
```

Рисунок 15 – Лог процесса обучения нейросети.

В заключении хотелось бы отметить полезные навыки, полученные в результате выполнения лабораторной работы:

1. Работа с DAG в Airflow
2. Работе сетями на huggingface