Отчёт 2 лабораторной работы по дисциплине “Инженерия данных”

выполнил Доружинский Дмитрий из группы 6233-010402D.

1. Пайплайн для инференса данных

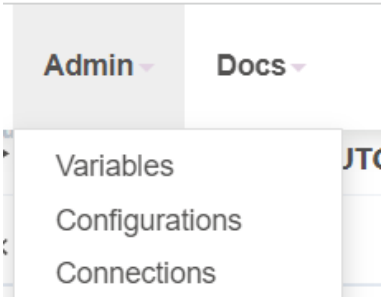
В 1 части лабораторной работы предлагается подстроить пайплайн со следующими этапами:

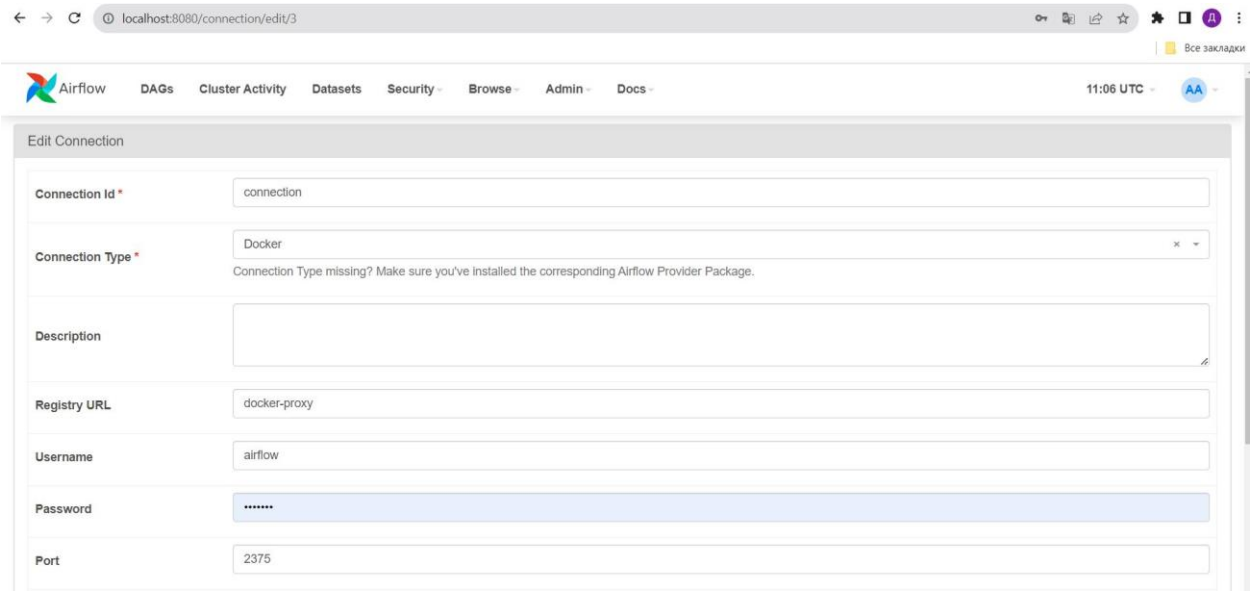
1. Производить мониторинг целевой папки на предмет появления новых видеофайлов.
2. Извлекать аудиодорожку из исходного видеофайла.
3. Преобразовывать аудиодорожку в текст с помощью нейросетевой модели.
4. Формировать конспект на основе полученного текста.
5. Формировать выходной .pdf файл с конспектом.

Пройдусь по каждому из них. В 1 пункте, как было предложено использовалась следующая конструкция.

```
wait_for_new_file = FileSensor(
    task_id='wait_for_new_file',
    poke_interval=10,  # Interval to check for new files (in seconds)
    filepath='/opt/airflow/data',  # Target folder to monitor
    fs_conn_id='connection',# Check FAQ for info
    dag=dag,
)
```

В параметре fs_conn_id необходимо указать имя, которое после необходимо указать при создание connections в выпадающем списке Admin графического интерфейса Airflow. Параметр poke_interval задает интервал времени в секундах обновления просмотра новых файлов





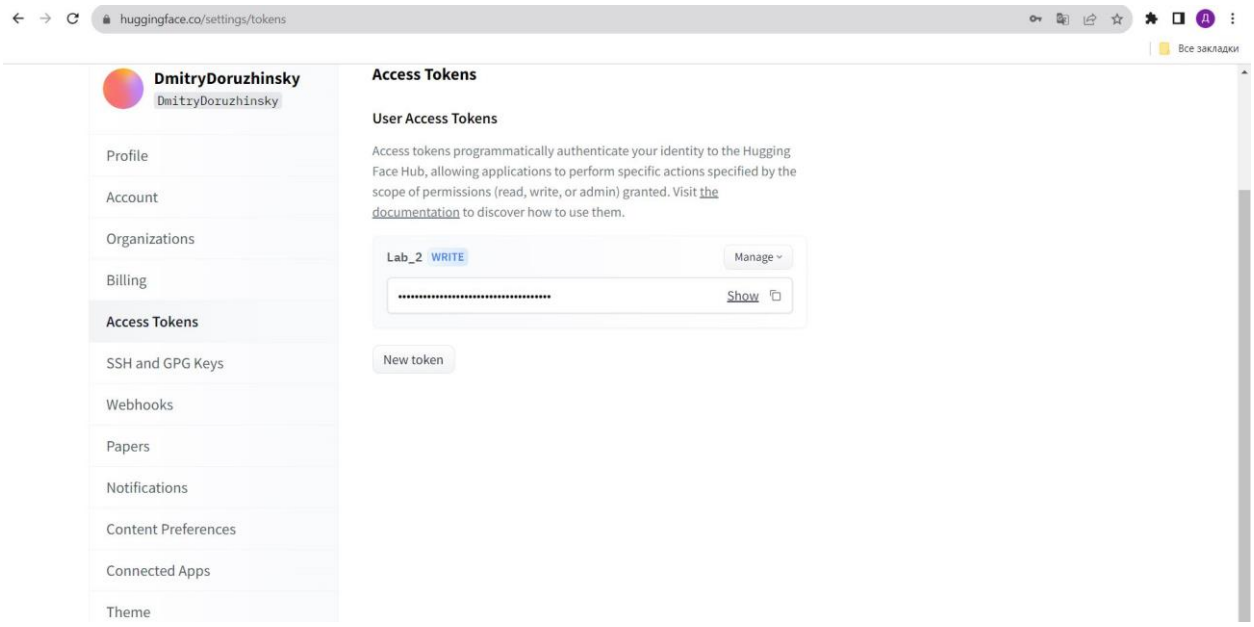
Извлекать аудио дорожку будет при помощи библиотеки ffmpeg и оператора DockerOperator. Хочу отметить, что это очень удобно, пользоваться DockerOperator, необходимо лишь найти нужный контейнер с библиотекой на DockerHub, либо создать его самостоятельно. В данном пункте используется DockerHub https://hub.docker.com/r/jrottenberg/ffmpeg

```
#Образ Docker с библиотекой FFmpeg для отделения аудио дорожки.
extract_audio = DockerOperator(
    task_id='extract_audio',
    image='jrottenberg/ffmpeg',
    command='-i /data/input_video.mp4 -vn -acodec copy /data/audio.aac',
    mounts=[Mount(source='/data', target='/data', type='bind')],
    docker_url="tcp://docker-proxy:2375",
    dag=dag,
)
```

Далее предлагается использовать нейронную сеть https://huggingface.co/openai/whisper-small для преобразования аудио в текст. Нужно отметить, что для этого необходимо подготовить следующий файл .py.

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

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

Нужно отметить, что для корректной работы необходимо получить личный токен на сайте www.huggingface.co после полной регистрации. И добавить его в переменную API_TOKEN



На этом шаге используем библиотеку request, для этого будет подкачивать следующий образ Docker, https://hub.docker.com/r/nyurik/alpine-python3-requests.

```
#Образ основан на frolvlad/alpine-python3,
#который сам по себе основан на образе Alpine Linux,
#который представляет собой образ размером всего 5 МБ и
#содержит Python 3.6 и библиотеку Requests .
#https://hub.docker.com/r/nyurik/alpine-python3-requests
transform_audio_to_text = DockerOperator(
    task_id='transform_audio_to_text',
    image='nyurik/alpine-python3-requests',
    command='python /data/transform_audio_to_text.py',
    mounts=[Mount(source='/data', target='/data', type='bind')],
    docker_url="tcp://docker-proxy:2375",
    dag=dag,
)
```

Далее необходимо при помощи следующей нейросети сделать краткую выжимку из получившегося текста (https://huggingface.co/slauw87/bart_summarisation). Этот этап похож на предыдущий.

```
summarize_text = DockerOperator(
    task_id='summarize_text',
    image='nyurik/alpine-python3-requests',
    command='python /data/summarize_text.py',
    mounts=[Mount(source='/data', target='/data', type='bind')],
    docker_url="tcp://docker-proxy:2375",
    dag=dag,
)
```

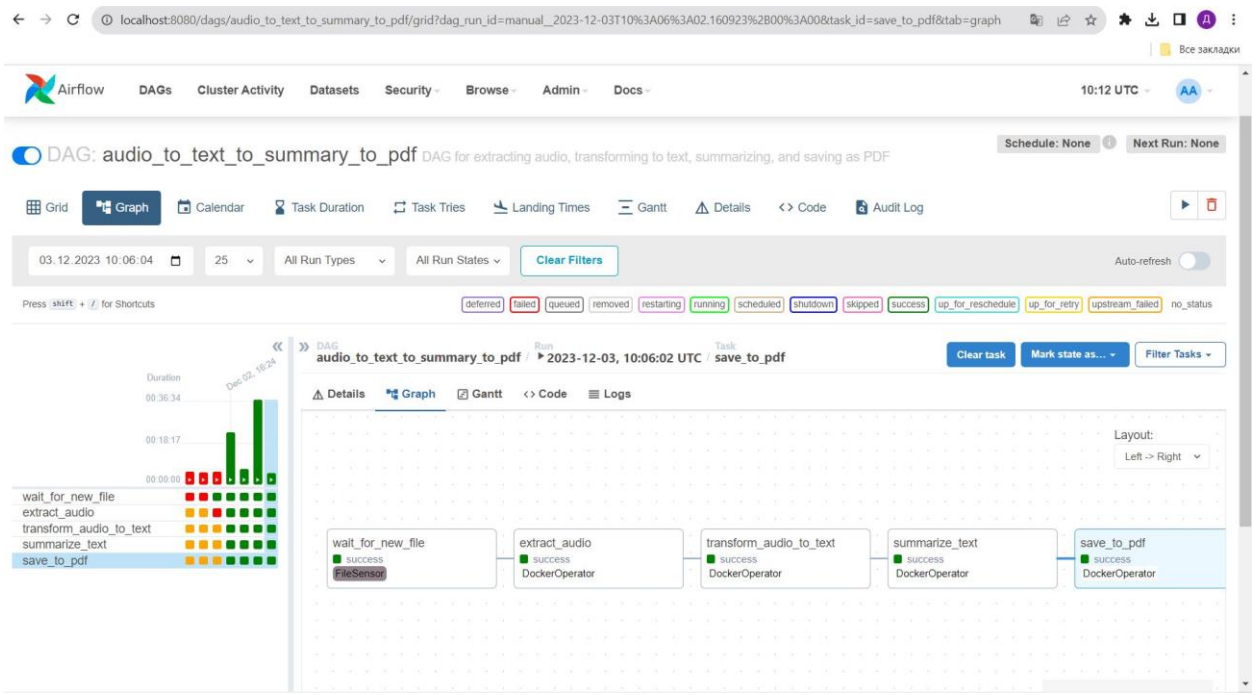
```
import requests
API_URL = "https://api-inference.huggingface.co/models/slauw87/bart_summarisation"
API_TOKEN = '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'
headers = {"Authorization": f"Bearer {API_TOKEN}"}

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

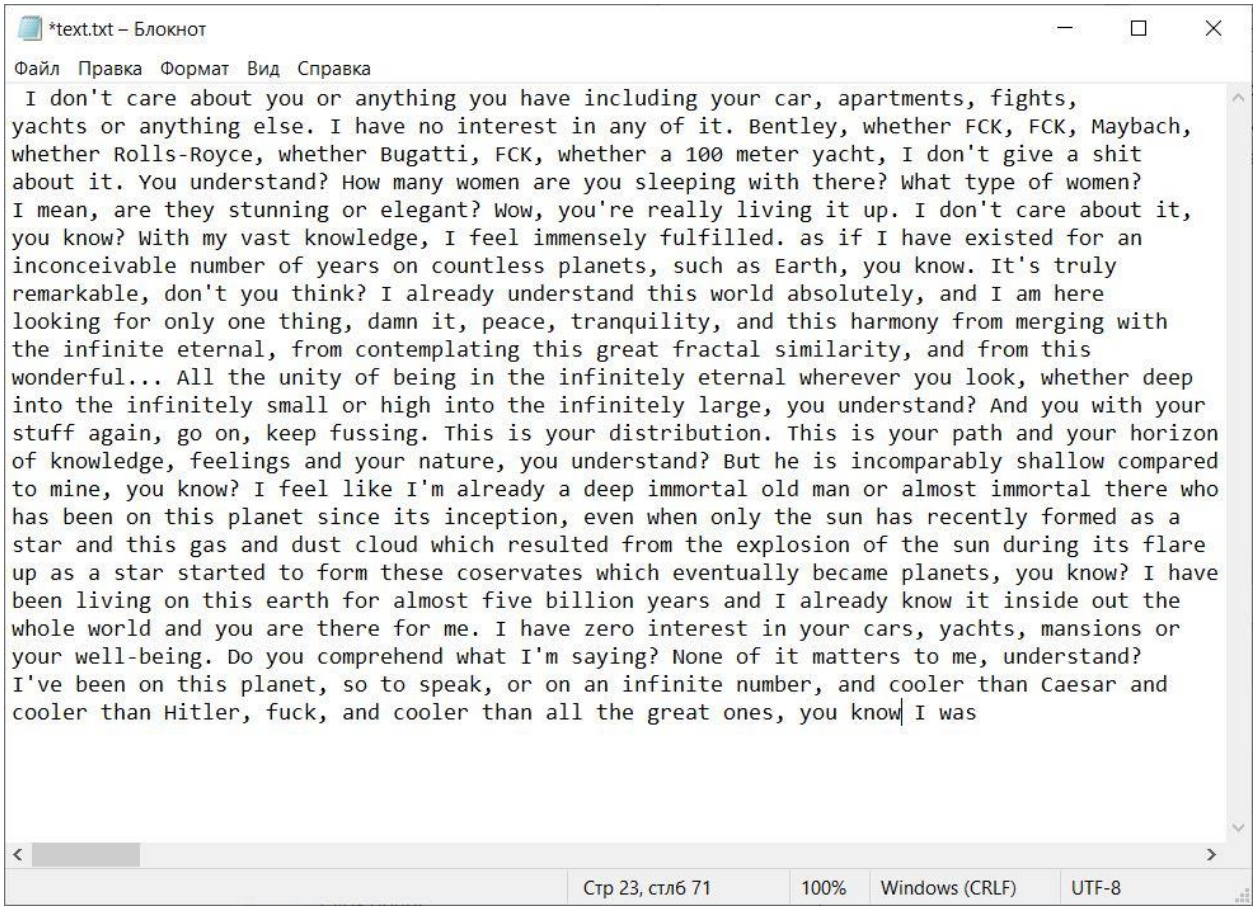
После необходимо сохранить получившийся текст сохранить в pdf для этого использовал бибилотеку fpdf, был найден на DockerHub образ для этого и подготовлен файл .py

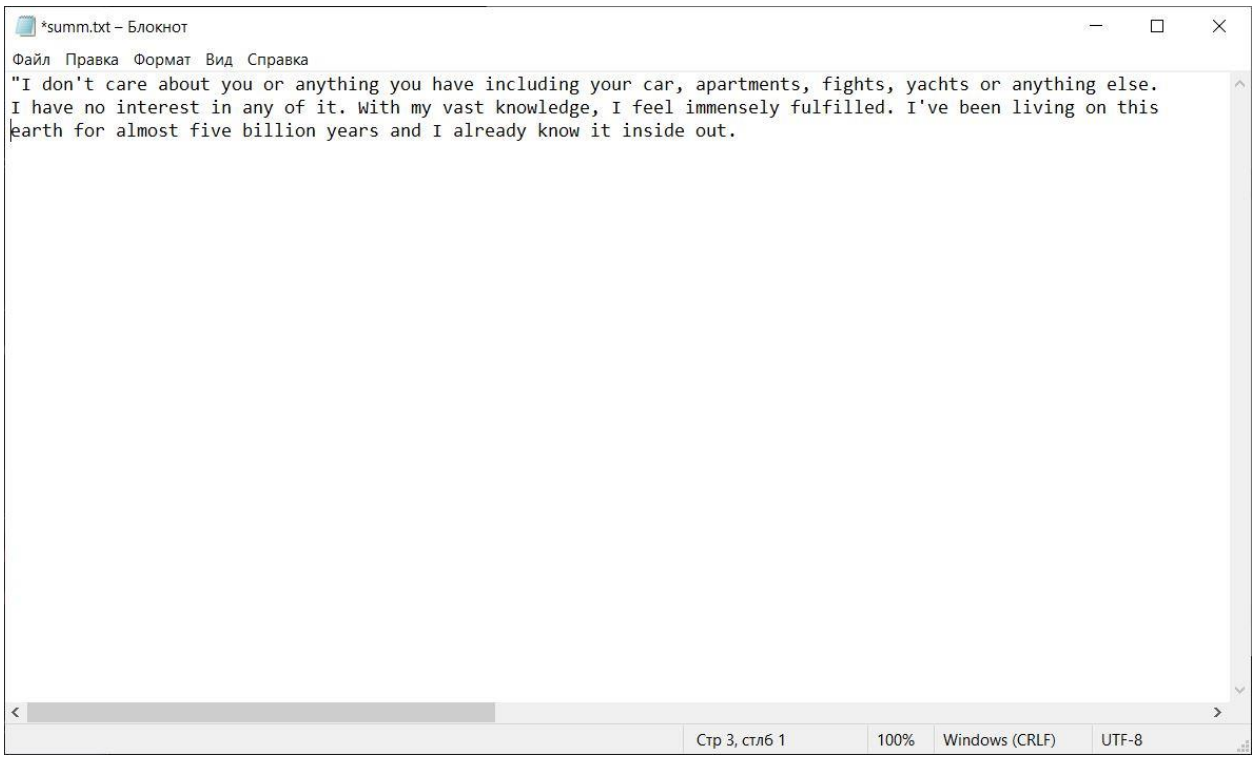
```
#Образ с библиотекой fpdf https://hub.docker.com/r/bikc/report/tags
save_to_pdf = DockerOperator(
    task_id='save_to_pdf',
    image='bikc/report:1.1',
    command='python /data/save_to_pdf.py',
    mounts=[Mount(source='/data', target='/data', type='bind')],
    docker_url="tcp://docker-proxy:2375",
    dag=dag,
)
```

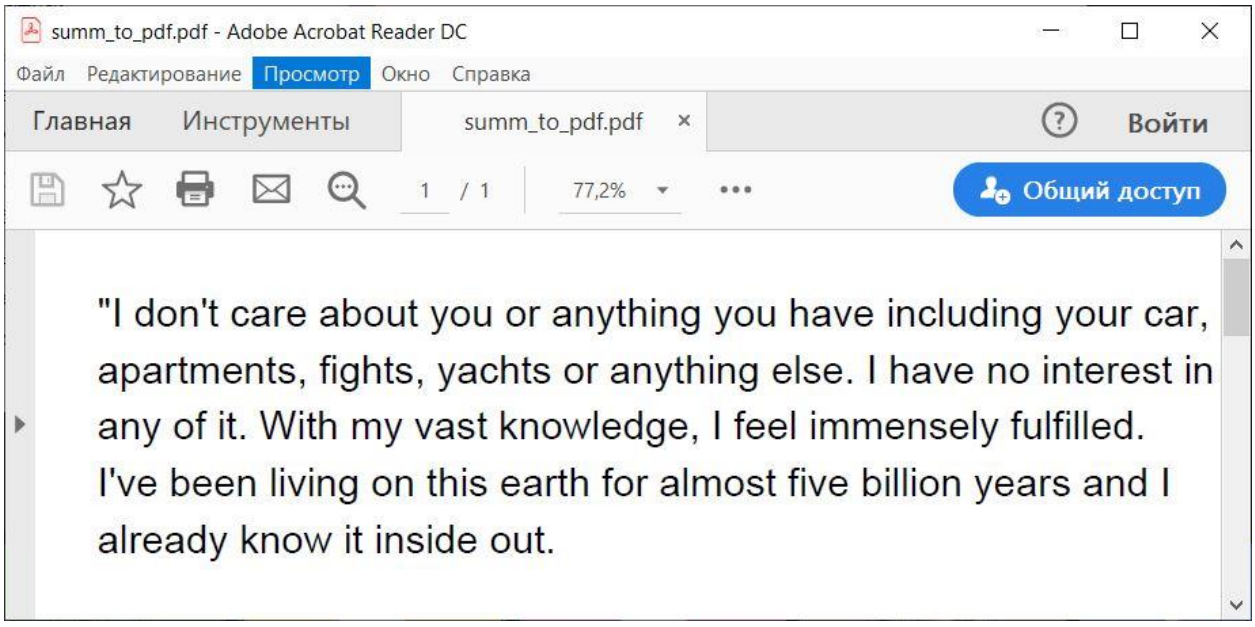
```
from fpdf import FPDF


file = open("/data/summ.txt")
pdf = FPDF()
pdf.add_page()
for text in file:
    pdf.set_font("Arial", size=20)
    pdf.multi_cell(w=200, h=10, txt=text, align="L",)
    pdf.output("/data/summ_to_pdf.pdf")
```

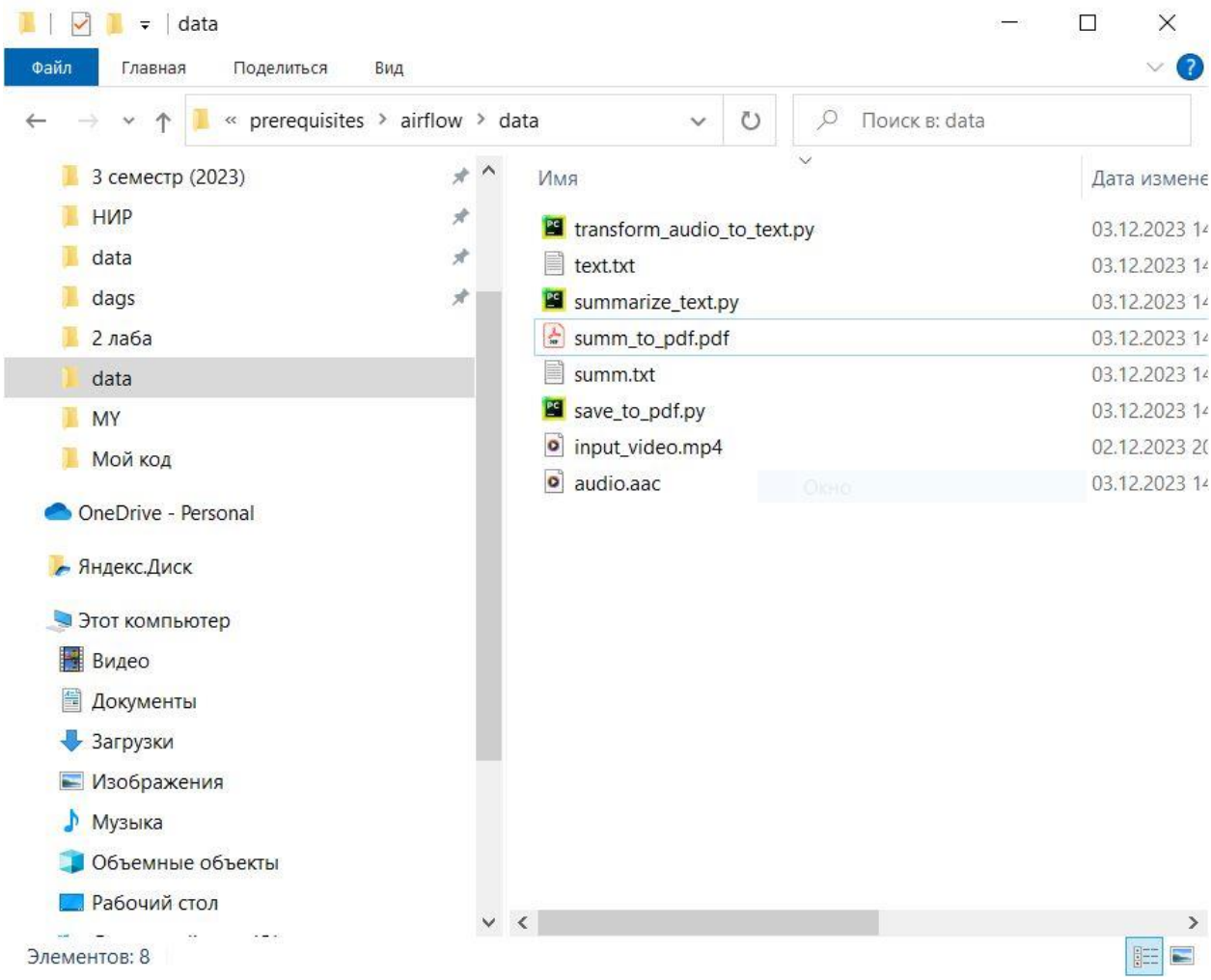
В итоге путём проб и ошибок получилось всё это заставить работать. В качестве входного видео было использован ролик “Идущий к реке” (https://www.youtube.com/watch?v=VHDwukiQPuE&ab_channel=%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D0%B0%D0%BD%D1%82%D0%B8%D0%BD%D0%9A%D0%BE%D0%BD%D0%BE%D0%B2%D0%B0%D0%BB%D0%BE%D0%B2) переведенный заранее на английский. Текст из аудио передан довольно точно и сформирован конспект. Результаты 1 части привожу ниже.


localhost:8080/dags/audio_to_text_to_summary_to_pdf/grid?dag_run_id=manual__2023-12-03T10%3A06%3A02.160923%2B00%3A00&task_id=save_to_pdf&tab=graph
Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
10:12 UTC
DAG: audio_to_text_to_summary_to_pdf
DAG for extracting audio, transforming to text, summarizing, and saving as PDF
Schedule: None
Next Run: None
Grid
Graph
Calendar
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Audit Log
03.12.2023 10:06:04
25
All Run Types
All Run States
Clear Filters
Auto-refresh
Press shift + / for Shortcuts
deferred
failed
queued
removed
restarting
running
scheduled
shutdown
skipped
success
up_for_reschedule
up_for_retry
upstream_failed
no_status
Duration
00:36:34
00:18:17
00:00:00
wait_for_new_file
extract_audio
transform_audio_to_text
summarize_text
save_to_pdf
DAG
audio_to_text_to_summary_to_pdf
Run
2023-12-03, 10:06:02 UTC
Task
save_to_pdf
Clear task
Mark state as...
Filter Tasks
Details
Graph
Gantt
Code
Logs
Layout:
Left -> Right
wait_for_new_file
success
FileSensor
extract_audio
success
DockerOperator
transform_audio_to_text
success
DockerOperator
summarize_text
success
DockerOperator
save_to_pdf
success
DockerOperator


*text.txt – Блокнот
Файл Правка Формат Вид Справка
I don't care about you or anything you have including your car, apartments, fights, yachts or anything else. I have no interest in any of it. Bentley, whether FCK, FCK, Maybach, whether Rolls-Royce, whether Bugatti, FCK, whether a 100 meter yacht, I don't give a shit about it. You understand? How many women are you sleeping with there? What type of women? I mean, are they stunning or elegant? Wow, you're really living it up. I don't care about it, you know? With my vast knowledge, I feel immensely fulfilled. as if I have existed for an inconceivable number of years on countless planets, such as Earth, you know. It's truly remarkable, don't you think? I already understand this world absolutely, and I am here looking for only one thing, damn it, peace, tranquility, and this harmony from merging with the infinite eternal, from contemplating this great fractal similarity, and from this wonderful... All the unity of being in the infinitely eternal wherever you look, whether deep into the infinitely small or high into the infinitely large, you understand? And you with your stuff again, go on, keep fussing. This is your distribution. This is your path and your horizon of knowledge, feelings and your nature, you understand? But he is incomparably shallow compared to mine, you know? I feel like I'm already a deep immortal old man or almost immortal there who has been on this planet since its inception, even when only the sun has recently formed as a star and this gas and dust cloud which resulted from the explosion of the sun during its flare up as a star started to form these coservates which eventually became planets, you know? I have been living on this earth for almost five billion years and I already know it inside out the whole world and you are there for me. I have zero interest in your cars, yachts, mansions or your well-being. Do you comprehend what I'm saying? None of it matters to me, understand? I've been on this planet, so to speak, or on an infinite number, and cooler than Caesar and cooler than Hitler, fuck, and cooler than all the great ones, you know I was
Стр 23, стлб 71
100%
Windows (CRLF)
UTF-8


*summ.txt – Блокнот
Файл Правка Формат Вид Справка
"I don't care about you or anything you have including your car, apartments, fights, yachts or anything else.
I have no interest in any of it. With my vast knowledge, I feel immensely fulfilled. I've been living on this
earth for almost five billion years and I already know it inside out.
Стр 3, стлб 1
100%
Windows (CRLF)
UTF-8


summ_to_pdf.pdf - Adobe Acrobat Reader DC
Файл Редактирование Просмотр Окно Справка
Главная Инструменты
summ_to_pdf.pdf
Войти
1 / 1
77,2%
Общий доступ
"I don't care about you or anything you have including your car, apartments, fights, yachts or anything else. I have no interest in any of it. With my vast knowledge, I feel immensely fulfilled. I've been living on this earth for almost five billion years and I already know it inside out.



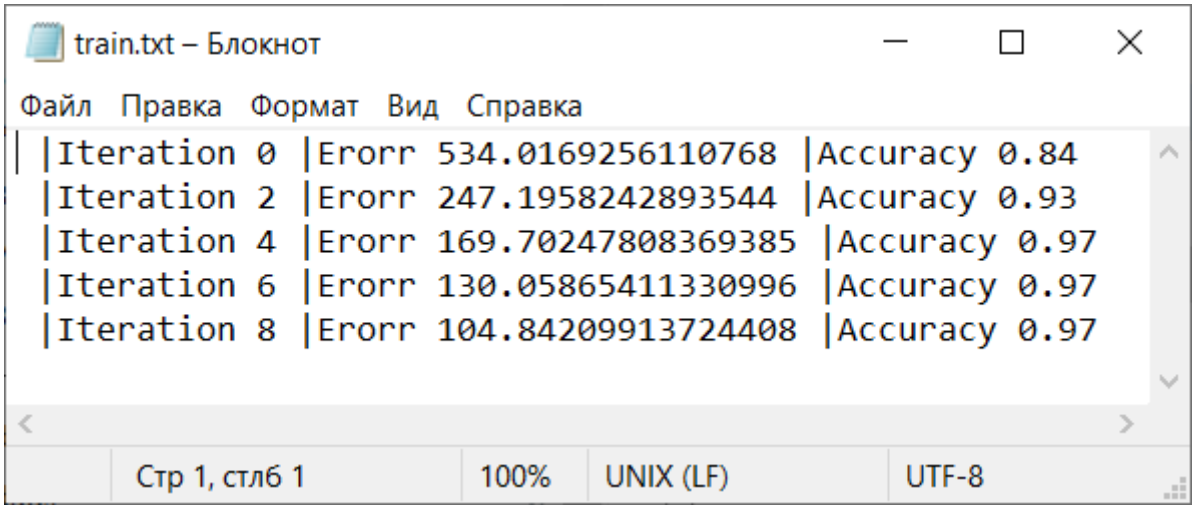
2. Пайплайн для обучения модели

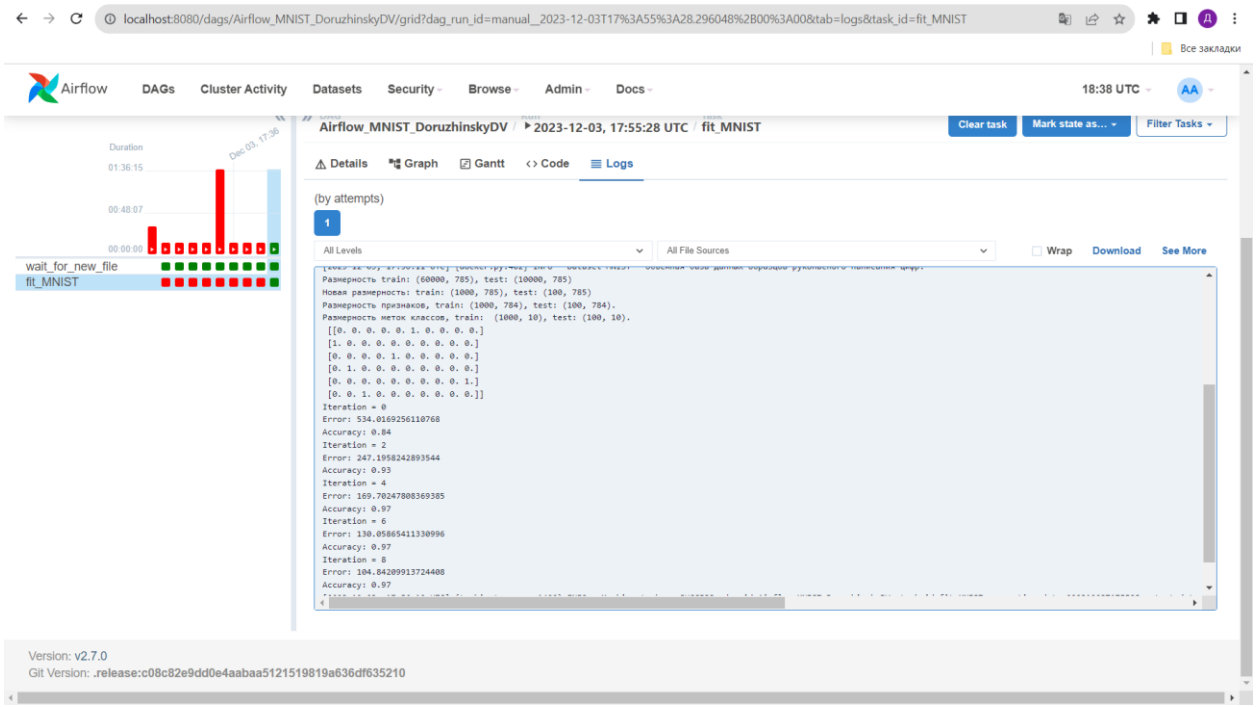
Во второй части было необходимо выполнить следующие шаги:

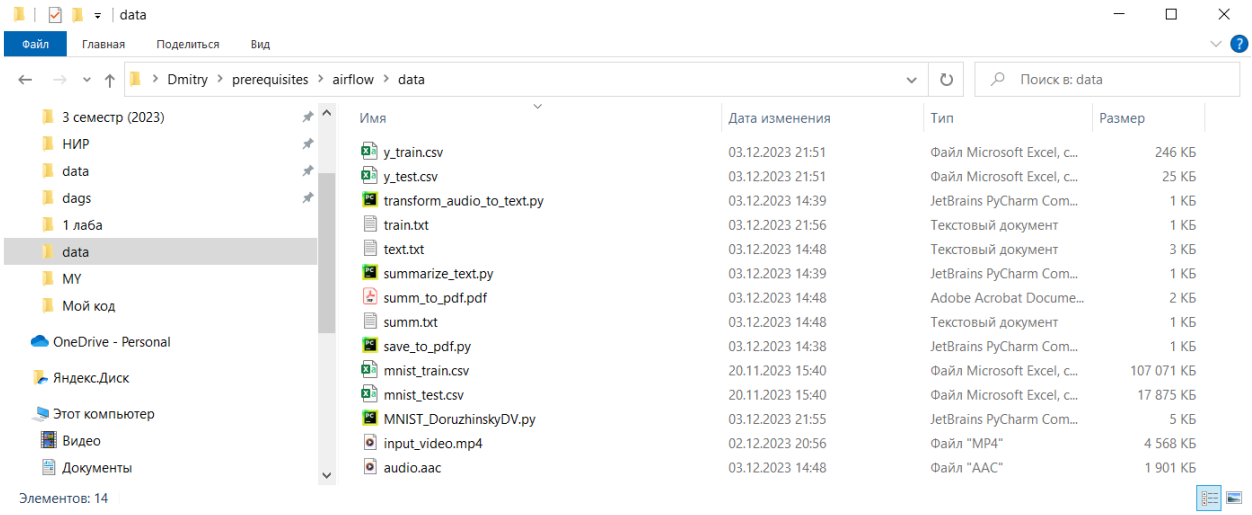
1. Читать набор файлов из определенного источника (файловой системы, сетевого интерфейса и т.д.).
2. Формировать пакет данных для обучения модели.
3. Обучать модель.
4. Сохранять данные результатов обученя (логи, значения функции ошибки) в текстовый файл

В качестве обучаемой модели была выбрана нейронная сеть с стохастическим градиентным спуском. Модель простая, код взят из лично выполненной лабораторной работы по курсу “Нейронные сети и глубокое обучение”. Набор данных MNIST загружается из папки airflow/data. В ходе выполнения столкнулся с несколькими проблемами. Первое долго не мог

найти подходящий образ Docker не большого размеры, но в итоге был найден huanjason/scikit-learn:latest, размером 1GB. В лабораторной работе использовался алгоритм OneHotEncoder для подготовки y_train и y_test, но в контейнер успешно запустить его не получилось, поэтому пришлось в файле .ipynb сохранить y_train и y_test в файл .csv и так же его считать уже в airflow. В итоге всё отработало успешно. Ограничил кол-во эпох и x_train из-за системных требований. Ниже привожу скриншоты выполнения, а также файла с сохраненными результатами.






data
Файл
Главная
Поделиться
Вид
Dmitry › prerequisites › airflow › data
Поиск в: data
3 семестр (2023)
НИР
data
dags
1 лаба
data
MY
Мой код
OneDrive - Personal
Яндекс.Диск
Этот компьютер
Видео
Документы
Имя
Дата изменения
Тип
Размер
y_train.csv 03.12.2023 21:51 Файл Microsoft Excel, c... 246 КБ
y_test.csv 03.12.2023 21:51 Файл Microsoft Excel, c... 25 КБ
transform_audio_to_text.py 03.12.2023 14:39 JetBrains PyCharm Com... 1 КБ
train.txt 03.12.2023 21:56 Текстовый документ 1 КБ
text.txt 03.12.2023 14:48 Текстовый документ 3 КБ
summarize_text.py 03.12.2023 14:39 JetBrains PyCharm Com... 1 КБ
summ_to_pdf.pdf 03.12.2023 14:48 Adobe Acrobat Docume... 2 КБ
summ.txt 03.12.2023 14:48 Текстовый документ 1 КБ
save_to_pdf.py 03.12.2023 14:38 JetBrains PyCharm Com... 1 КБ
mnist_train.csv 20.11.2023 15:40 Файл Microsoft Excel, c... 107 071 КБ
mnist_test.csv 20.11.2023 15:40 Файл Microsoft Excel, c... 17 875 КБ
MNIST_DoruzhinskyDV.py 03.12.2023 21:55 JetBrains PyCharm Com... 5 КБ
input_video.mp4 02.12.2023 20:56 Файл "MP4" 4 568 КБ
audio.aac 03.12.2023 14:48 Файл "AAC" 1 901 КБ
Элементов: 14