Conversational Emotional Diagnostics

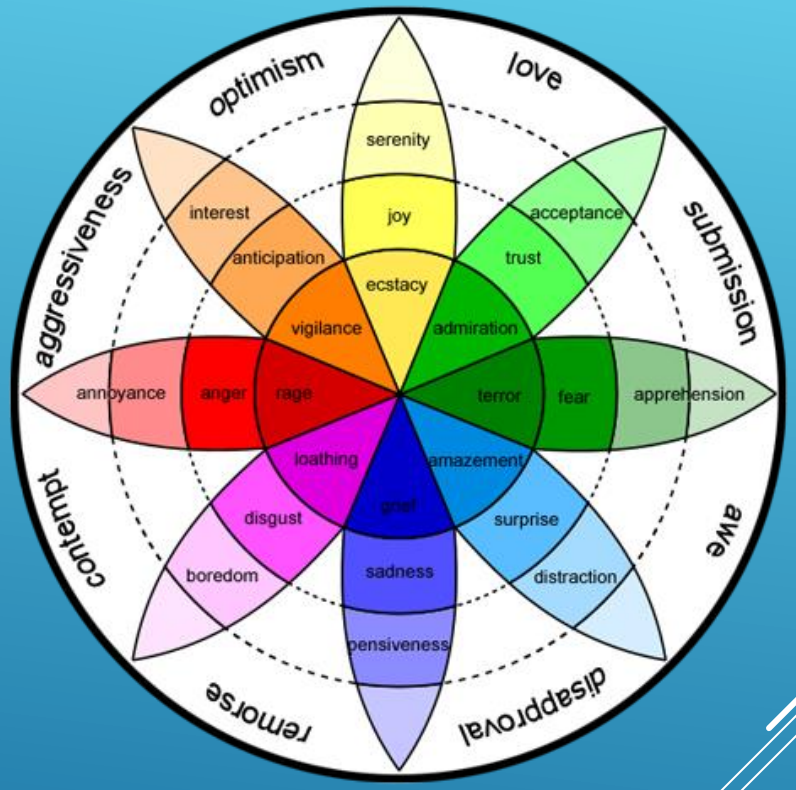
ДИАГНОСТИКА ЭМОЦИОНАЛЬНОГО СОСТОЯНИЯ РАЗГОВОРОВ

Выпускной проект

Слушателя курса «Data Science: машинное обучение и нейронные сети» Профессиональный уровень

Владыкиной Татьяны

АКТУАЛЬНОСТЬ ПРОЕКТА



ДИАЛОГ

ЭМОЦИИ СПИКЕРОВ

АКТУАЛЬНОСТЬ ПРОЕКТА

1. Клиентский сервис (улучшение качества обслуживания клиентов в колл-центрах за счет анализа эмоционального состояния операторов и клиентов)

2. Образование и тренинги (объективная оценка эмоционального состояния участников образовательных и тренинговых программ для корректировки подхода к обучению)

3. Научные исследования (ускорение и упрощение процесса анализа данных в психологии и поведенческих исследованиях)

4. Медицина и здравоохранение (помощь в диагностике и лечении психических расстройств через анализ эмоций пациентов)

5. Юридическая сфера (оценка эмоционального состояния свидетелей и участников судебных процессов)

6. Маркетинг и реклама (анализ эмоциональных реакций на рекламные кампании для повышения их эффективности)

7. Организация встреч (автоматическое составление протоколов встреч и постановка задач участникам на основе анализа голосов)

Minimum viable product

ВЕРХНЕУРОВНЕВАЯ СХЕМА ПРОЕКТА

По каждому из диалогов

Запись разговора .wav

Запись разговора .wav

Запись разговора .wav

Модель 1

Определение количества спикеров и запись речи каждого из спикеров в отдельный аудиофайл.

Использовалась готовая модель pyannote/speaker-diarization-3.1 из huggingface

По каждому спикеру в рамках диалога

Файл .wav с речью спикера

Файл .wav с речью спикера

Файл .wav с речью спикера

Модель 2

Определение эмоций в речи спикера.

Модель GRU (Gated Recurrent Unit).

Обучена на датасете RAVDESS Emotional speech audio для распознавания 8-ми эмоций.

По каждому спикеру в рамках диалога

Эмоция (emotion_label)

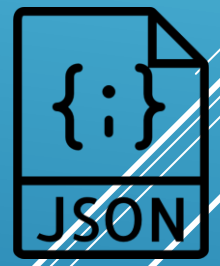
Эмоция (emotion_label)

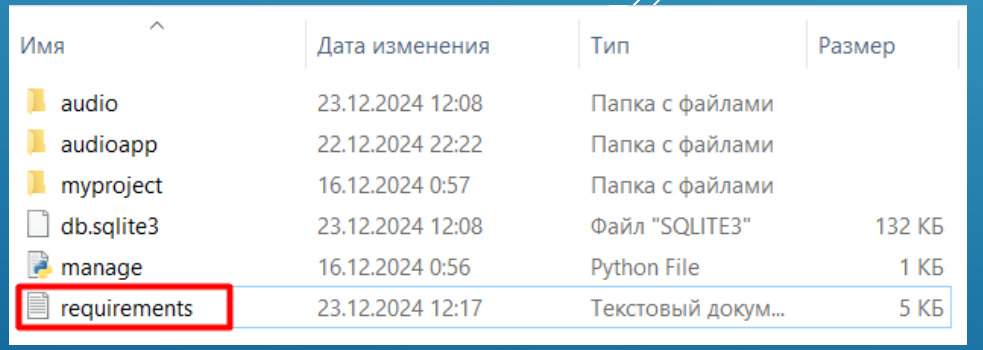
Эмоция (emotion_label)

Управленческое решение

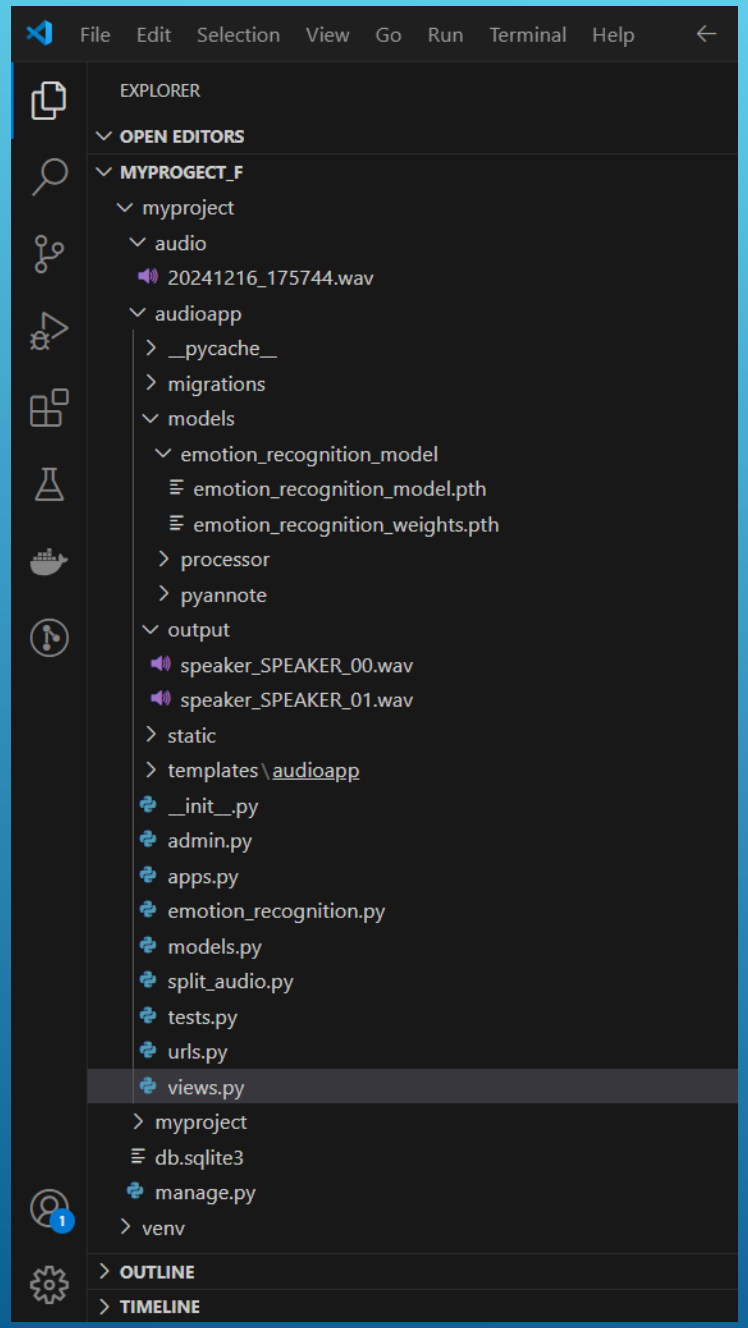
СТЕК ТЕХНОЛОГИЙ

- **Язык программирования:**
  - **Python**
- **Веб-фреймворк:**
  - **Django**
- **Модели:**
  - **pyannote/speaker-diarization-3.1** для разделения аудиофайлов на сегменты по голосам, доступная на платформе Hugging Face.
  - **GRU (Gated Recurrent Unit)** для распознавания эмоций в аудиофайлах.
- **Некоторые библиотеки:**
  - **PyTorch**
  - **Torchaudio**
  - **Scikit-learn**
  - **Matplotlib**
  - **JSON**
  - **Optuna**
- **Среда разработки:**
  - **Visual Studio Code**
  - **Google Colab** для обучение модели **GRU (Gated Recurrent Unit)** с целью последующего переноса результатов обучения на локальный диск ПК





Имя	Дата изменения	Тип	Размер
audio	23.12.2024 12:08	Папка с файлами	
audioapp	22.12.2024 22:22	Папка с файлами	
myproject	16.12.2024 0:57	Папка с файлами	
db.sqlite3	23.12.2024 12:08	Файл "SQLITE3"	132 КБ
manage	16.12.2024 0:56	Python File	1 КБ
requirements	23.12.2024 12:17	Текстовый докум...	5 КБ



ПАПКИ И ФАЙЛЫ ПРОЕКТА

Папки:

- **__pycache__** содержит скомпилированные байт-коды Python для ускорения загрузки модулей.
- **migrations** содержит файлы миграций базы данных для отслеживания изменений в моделях.
- **models** содержит файлы моделей данных, используемых в проекте.
- **static** содержит статические файлы, такие как CSS, JavaScript и изображения.
- **templates** содержит HTML-шаблоны для рендеринга страниц.

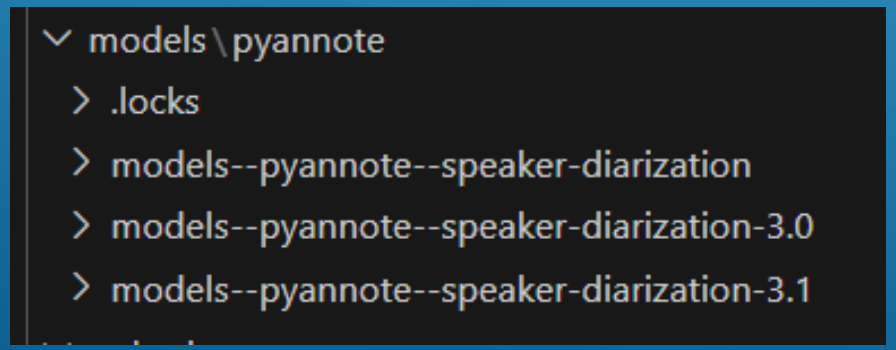
Файлы:

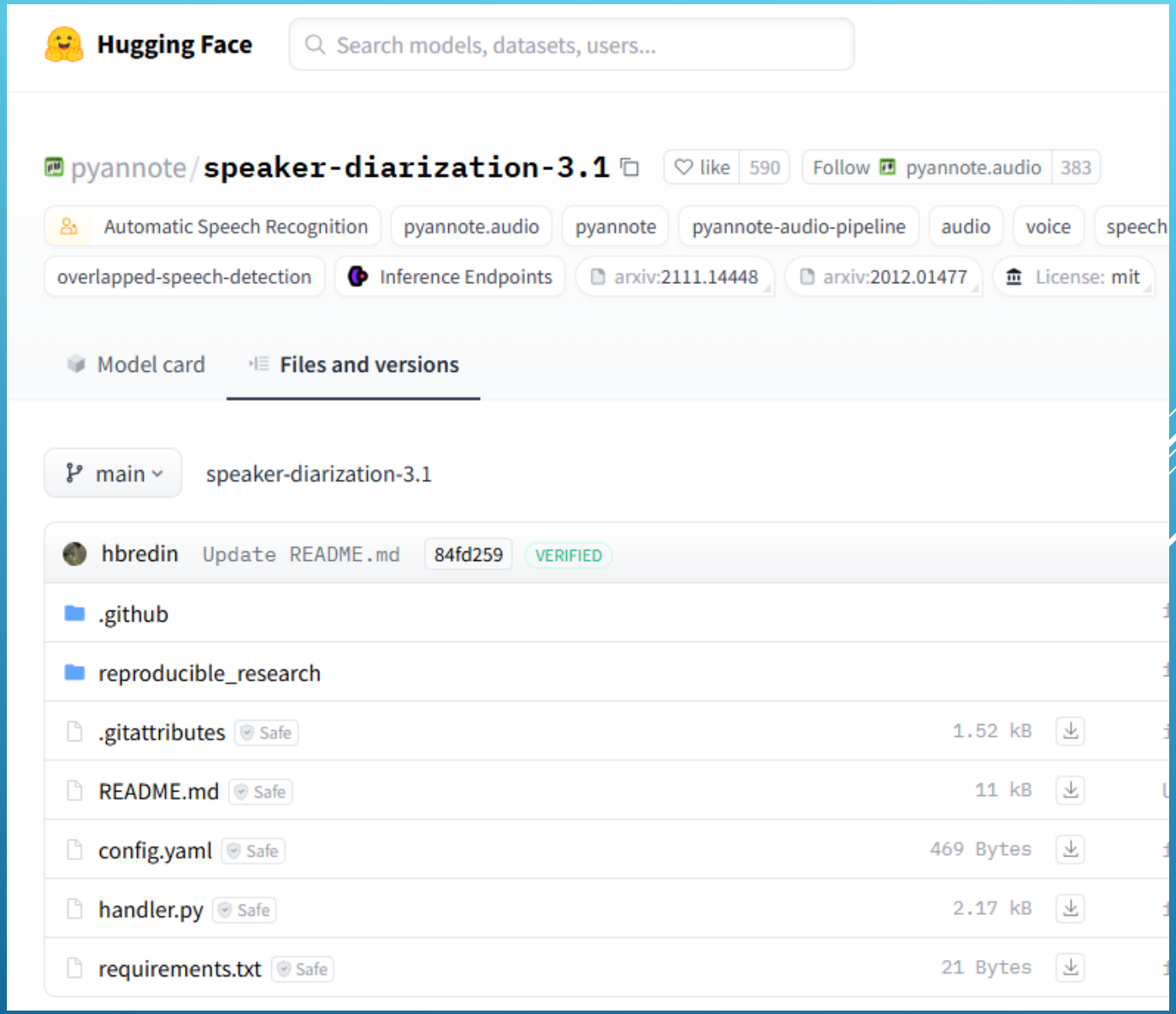
- **__init__.py** инициализирует пакет Python, позволяя импортировать модули из этой директории.
- **admin.py** конфигурация административного интерфейса Django для управления моделями.
- **apps.py** конфигурация приложения Django.
- **emotion_recognition.py** содержит функции для определения эмоций в аудиофайлах с использованием модели.
- **models.py** определяет модели данных для приложения.
- **split_audio.py** содержит функции для разделения аудиофайлов на отдельные файлы по голосам.
- **urls.py** определяет маршруты URL для приложения.
- **views.py** содержит представления (views) для обработки запросов и рендеринга ответов.

МОДЕЛЬ PYANNOTE/SPEAKER-DIARIZATION-3.1

Модель предназначена для автоматического распознавания и разделения речи говорящих в аудиозаписях. Она использует библиотеку pyannote.audio и может выполнять задачи, такие как:

- **Распознавание говорящих**: определение, кто говорит в каждый момент времени.
- **Обнаружение изменений говорящих**: определение моментов, когда один говорящий сменяется другим.
- **Обнаружение активности голоса**: определение, когда в аудиозаписи присутствует речь.
- **Обнаружение перекрывающейся речи**: определение моментов, когда несколько говорящих говорят одновременно.





МОДЕЛЬ GRU (GATED RECURRENT UNIT)

Это тип рекуррентной нейронной сети (RNN), разработанный для решения проблемы исчезающего градиента, которая часто возникает в традиционных RNN. GRU использует механизмы управления потоком информации через сеть, что позволяет лучше сохранять и передавать важную информацию на протяжении длинных последовательностей данных.

Основные компоненты модели GRU:

1. Загрузка датасета: датасеты загружаются из файлов и подготавливаются для обучения и валидации.

2. DataLoader: train_loader и val_loader используются для загрузки данных в батчах, что позволяет эффективно обрабатывать данные во время обучения.

3. Параметры модели:

- input_size: размер входных данных (количество признаков);
- hidden_size: размер скрытого слоя (32 нейронов в скрытом слое);
- num_layers: количество слоев GRU (3 слоя);
- num_classes: количество классов для классификации эмоций (8 классов).

{'num_epochs': 50, 'batch_size': 32, 'learning_rate': 0.0008578837882612859, 'hidden_size': 32, 'num_layers': 3}

4. Создание модели: Модель GRU создаётся с указанными параметрами.

5. Устройство (device): определяется устройство для выполнения вычислений (GPU или CPU). Модель переводится на это устройство для выполнения вычислений.

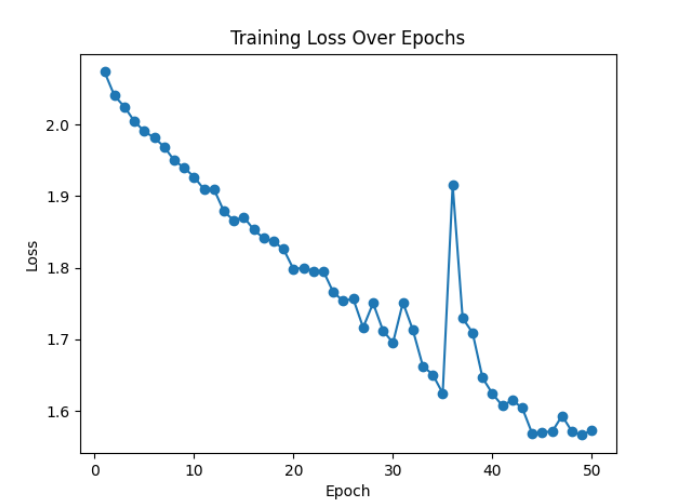
МОДЕЛЬ GRU (GATED RECURRENT UNIT)

6. Функция потерь и оптимизатор: определяются функция потерь (CrossEntropyLoss) и оптимизатор (Adam) для обучения модели.

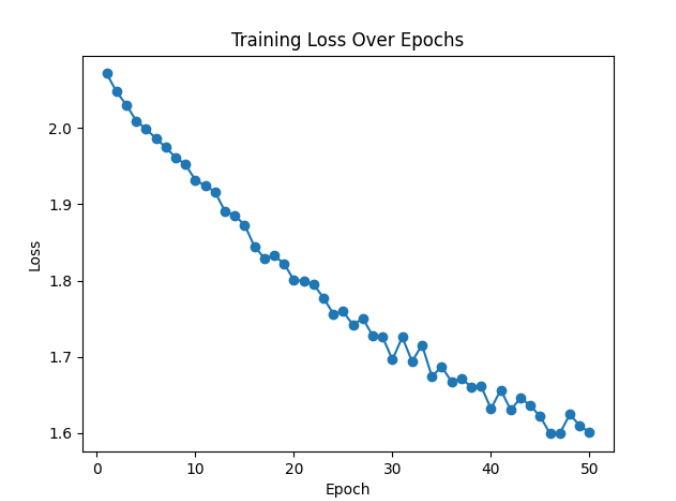
7. Цикл обучения: модель обучается в течение заданного количества эпох (num_epochs). В каждой эпохе данные проходят через модель, вычисляется функция потерь, и модель обновляется с помощью оптимизатора. Прогресс обучения и потери выводятся на экран.

8. Сохранение модели: обученная модель сохраняется в указанный файл.

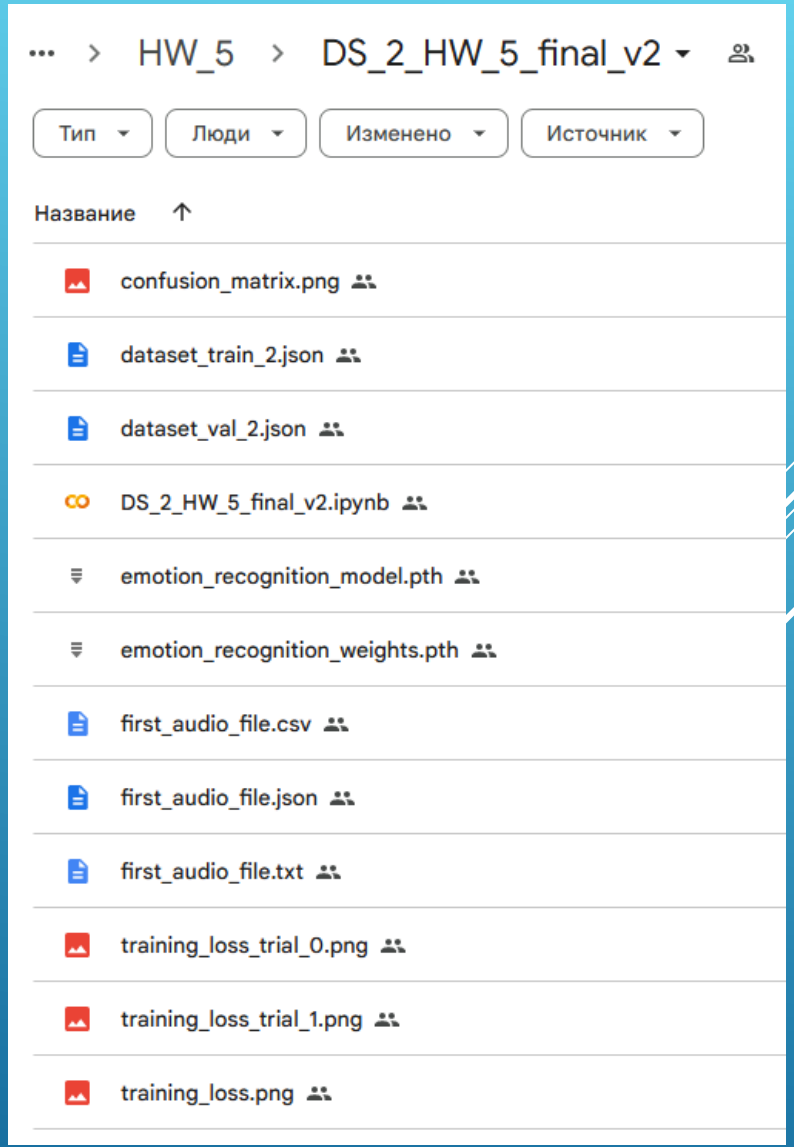
9. Визуализация потерь: потери на каждой эпохе сохраняются и отображаются на графике, который сохраняется в файл.



Optuna



GRU model



ДАТАСЕТ ДЛЯ МОДЕЛИ EmotionRecognitionModel

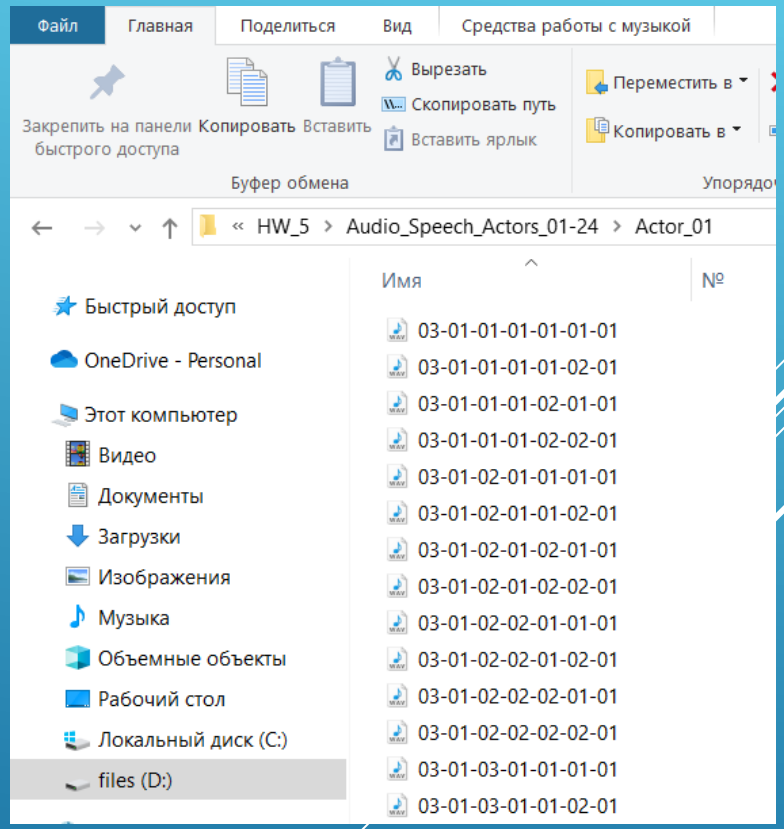
The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)
https://zenodo.org/records/1188976

Filename identifiers

- Modality (01 = full-AV, 02 = video-only, 03 = audio-only).
- Vocal channel (01 = speech, 02 = song).
- Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
- Emotional intensity (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion.
- Statement (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").
- Repetition (01 = 1st repetition, 02 = 2nd repetition).
- Actor (01 to 24. Odd numbered actors are male, even numbered actors are female).

Filename example: 02-01-06-01-02-01-12.mp4

1. Video-only (02)
2. Speech (01)
3. Fearful (06)
4. Normal intensity (01)
5. Statement "dogs" (02)
6. 1st Repetition (01)
7. 12th Actor (12)
8. Female, as the actor ID number is even.



Audio_Speech_Actors_01-24.zip
md5:bc696df654c87fed845eb13823edef8a

208.5 MB

Preview Download

ОПИСАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ПУТИ

- **Загрузка файла:**
  - Пользователь загружает аудиофайл на веб-странице приложения.

- **Обработка файла:**
  - Аудиофайл отправляется на сервер, где происходит его обработка.
  - Модель pyannote/speaker-diarization-3.1 определяет количество спикеров и разделяет аудиофайл на отдельные сегменты.
  - Модель GRU (emotion_recognition_model) анализирует каждый аудиофайл и определяет эмоции.

- **Отображение результатов:**
  - Обработанные данные возвращаются на веб-страницу.
  - Пользователь видит результаты анализа: количество спикеров и эмоции спикеров.

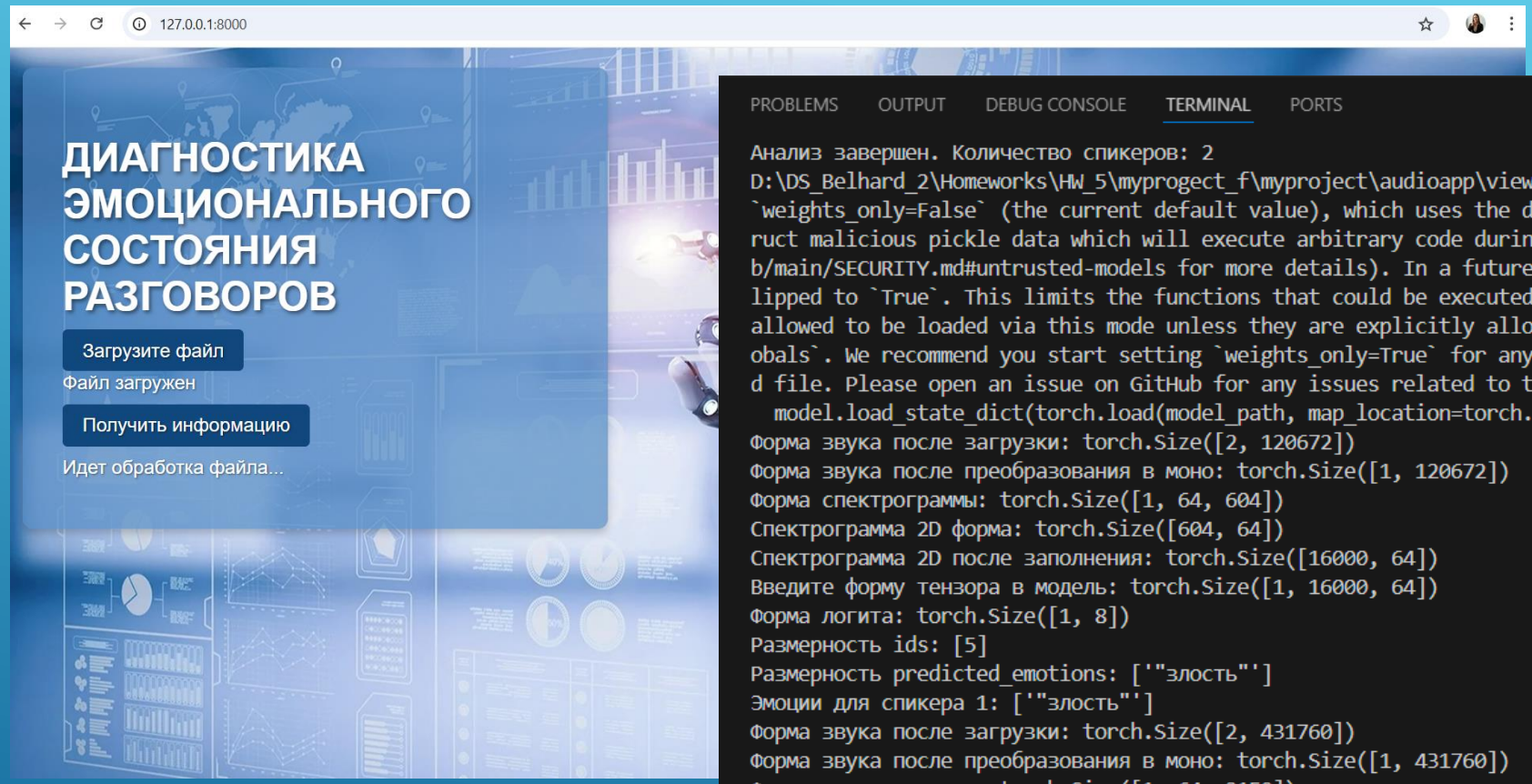
ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС


```
December 23, 2024 - 11:51:23
Django version 5.1.4, using settings 'myproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[23/Dec/2024 12:08:28] "GET / HTTP/1.1" 200 5932
Файл загружен, ID: 65
[23/Dec/2024 12:08:33] "POST /upload/ HTTP/1.1" 200 10
```

ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS        python - myproject

[23/Dec/2024 12:08:33] "POST /upload/ HTTP/1.1" 200 10
Начало анализа файла: D:\DS_Belhard_2\Homeworks\HW_5\myprogect_f\myproject\audio\20241216_175744.wav
Начало разделения аудио на файлы по голосам...
D:\DS_Belhard_2\Homeworks\HW_5\myprogect_f\venv\lib\site-packages\pyannote\audio\models\blocks\pooling.py:104: UserWarning
<= 0. Correction should be strictly less than the reduction factor (input numel divided by output numel). (Triggered inter
k\pytorch\pytorch\builder\windows\pytorch\aten\src\ATen\native\ReduceOps.cpp:1823.)
  std = sequences.std(dim=-1, correction=1)
Спикер: SPEAKER_01, Время начала: 638 мс, Время окончания: 3152 мс, Продолжительность: 2514 мс
Спикер: SPEAKER_00, Время начала: 4080 мс, Время окончания: 5414 мс, Продолжительность: 1334 мс
Спикер: SPEAKER_00, Время начала: 9902 мс, Время окончания: 14205 мс, Продолжительность: 4303 мс
Разделение аудио завершено.
Анализ завершен. Количество спикеров: 2
D:\DS_Belhard_2\Homeworks\HW_5\myprogect_f\myproject\audioapp\views.py:50: FutureWarning: You are using `torch.load` with
```

ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS python - myproject

```
Анализ завершен. Количество спикеров: 2
D:\DS_Belhard_2\Homeworks\HW_5\myprogect_f\myproject\audioapp\views.py:50: FutureWarning: You are using `torch.load` with
`weights_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to const
ruct malicious pickle data which will execute arbitrary code during unpickling (See https://github.com/pytorch/pytorch/blo
b/main/SECURITY.md#untrusted-models for more details). In a future release, the default value for `weights_only` will be f
lipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be
allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add_safe_gl
obals`. We recommend you start setting `weights_only=True` for any use case where you don't have full control of the loade
d file. Please open an issue on GitHub for any issues related to this experimental feature.
  model.load_state_dict(torch.load(model_path, map_location=torch.device('cpu')))
Форма звука после загрузки: torch.Size([2, 120672])
Форма звука после преобразования в моно: torch.Size([1, 120672])
Форма спектрограммы: torch.Size([1, 64, 604])
Спектрограмма 2D форма: torch.Size([604, 64])
Спектрограмма 2D после заполнения: torch.Size([16000, 64])
Введите форму тензора в модель: torch.Size([1, 16000, 64])
Форма логита: torch.Size([1, 8])
Размерность ids: [5]
Размерность predicted_emotions: ['"злость"']
Эмоции для спикера 1: ['"злость"']
Форма звука после загрузки: torch.Size([2, 431760])
Форма звука после преобразования в моно: torch.Size([1, 431760])
Форма спектрограммы: torch.Size([1, 64, 2159])
Спектрограмма 2D форма: torch.Size([2159, 64])
Спектрограмма 2D после заполнения: torch.Size([16000, 64])
Введите форму тензора в модель: torch.Size([1, 16000, 64])
Форма логита: torch.Size([1, 8])
Размерность ids: [4]
Размерность predicted_emotions: ['"грусть"']
Эмоции для спикера 2: ['"грусть"']
[23/Dec/2024 12:10:15] "POST /analyze/65/ HTTP/1.1" 200 147
```

ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

СЛОЖНОСТИ В ХОДЕ ВЫПОЛНЕНИЯ ПРОЕКТА

СЛОЖНОСТЬ ТЕМЫ

Работа с аудиофайлами и их обработка представляют собой сложную задачу по нескольким причинам:

1.Многоканальные аудиофайлы: Аудиофайлы могут быть многоканальными (например, стерео), что приводит к увеличению размерности тензоров при загрузке данных. Модель, ожидающая тензоры размерности 2 или 3, может столкнуться с ошибками при обработке тензоров размерности 4. Для решения этой проблемы необходимо преобразовать стерео звук в моно, усреднив каналы.

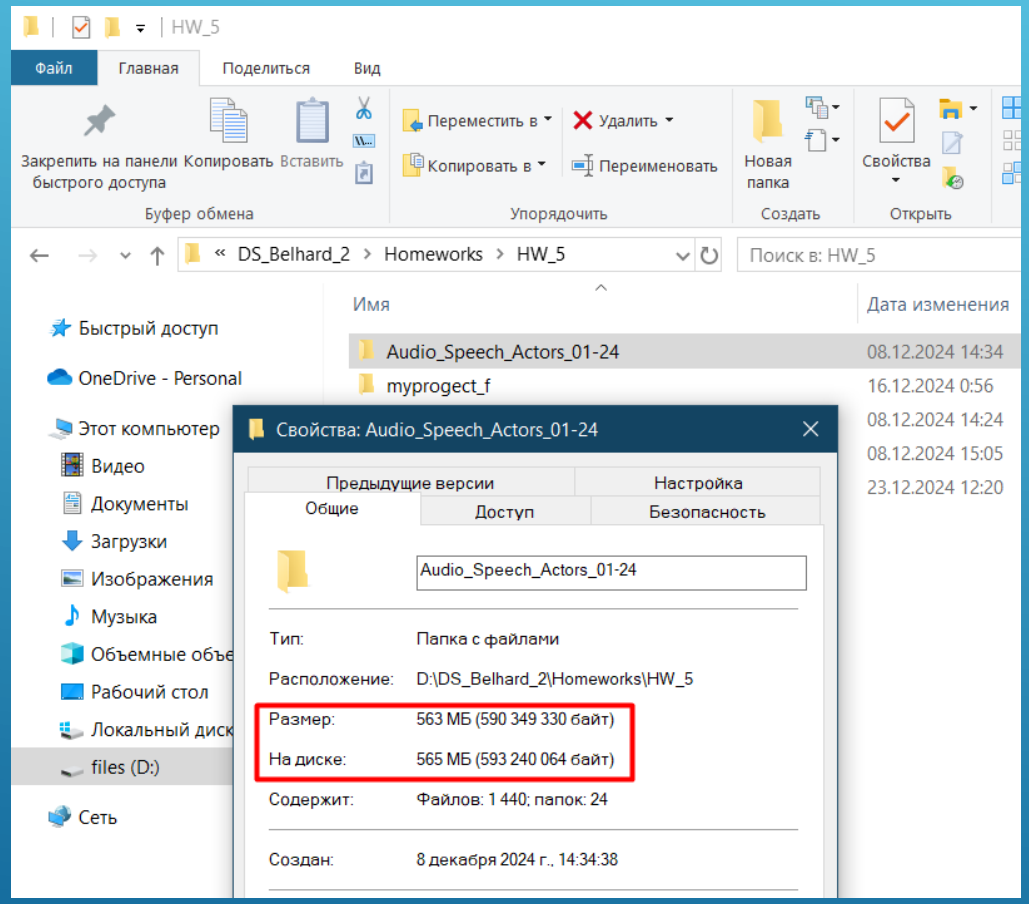
2.Преобразование аудио в спектрограмму: Преобразование аудиоданных в спектрограмму требует учета различных параметров, таких как количество мел-банов и размер окна преобразования. Неправильная настройка этих параметров может привести к некорректным результатам.

3.Выравнивание длины аудиофайлов: Аудиофайлы могут иметь разную длину, что затрудняет их обработку моделью. Для решения этой проблемы необходимо выравнивать длину аудиофайлов, используя данные из того же файла.

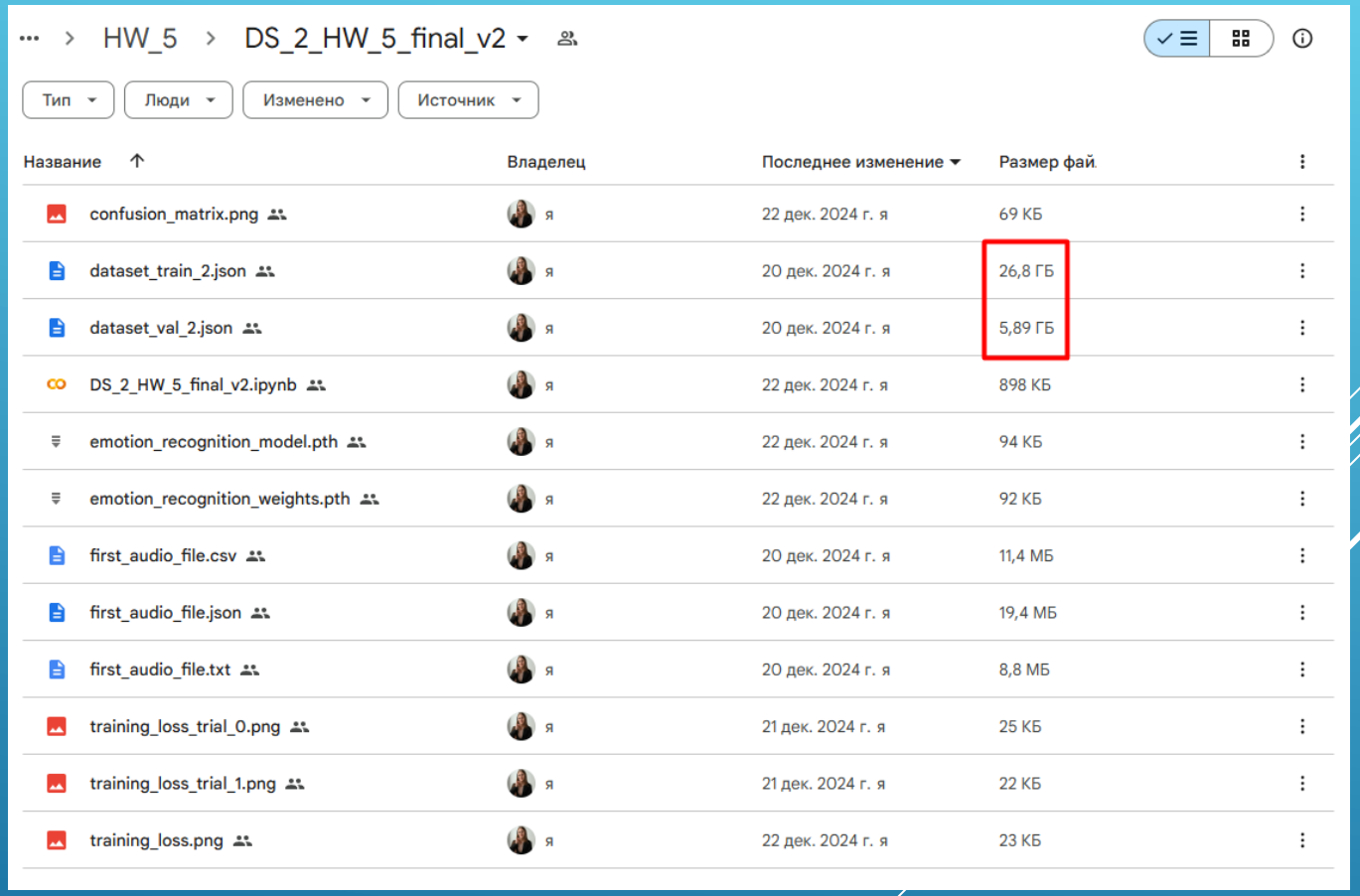
4.Проверка размерности тензоров: На каждом этапе преобразования аудиоданных необходимо проверять размерность тензоров, чтобы убедиться, что они имеют правильную форму для модели. Это помогает избежать ошибок и обеспечивает корректную обработку данных.

СЛОЖНОСТИ В ХОДЕ ВЫПОЛНЕНИЯ ПРОЕКТА

БОЛЬШОЙ ОБЪЕМ ДАННЫХ ДЛЯ ОБУЧЕНИЯ МОДЕЛИ



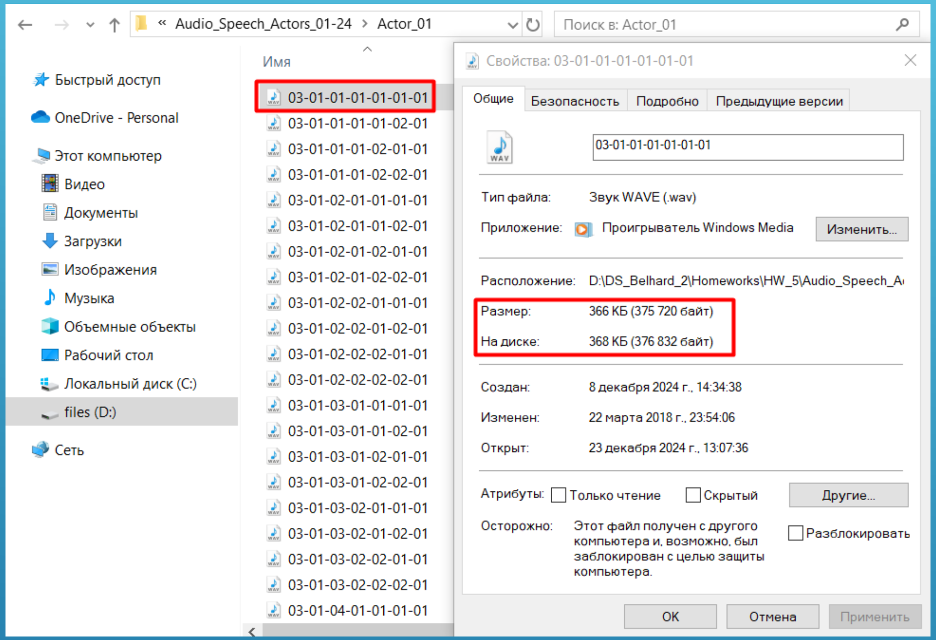
Исходный датасет

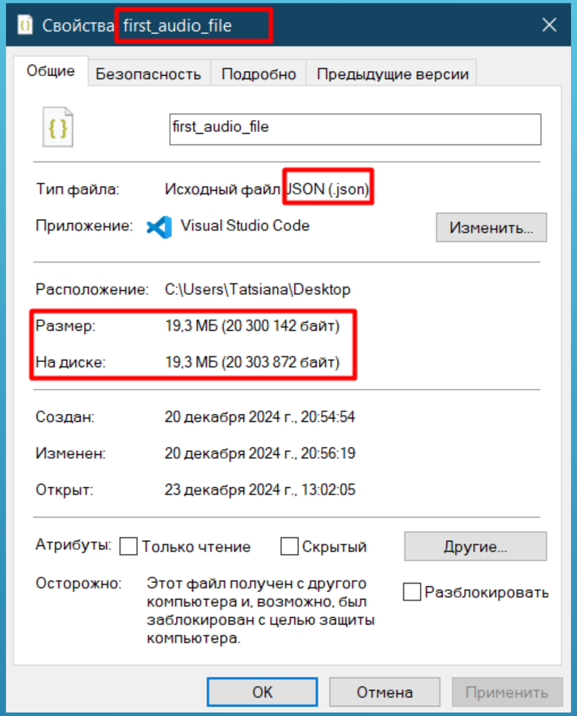


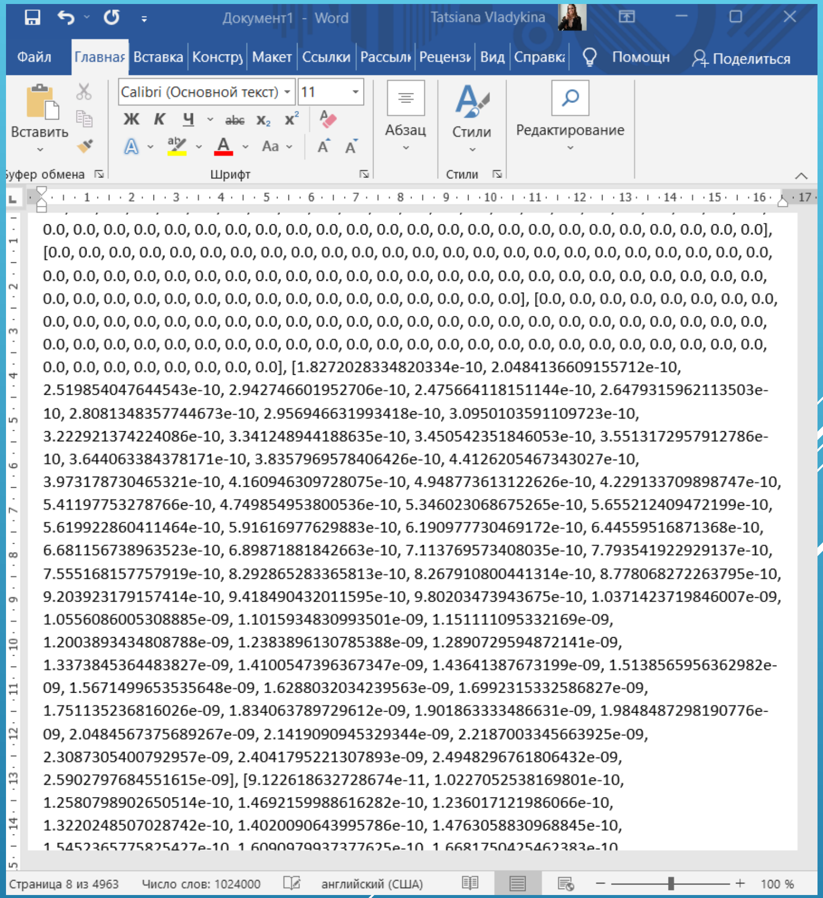
Данные, которые использовались для обучения модели

СЛОЖНОСТИ В ХОДЕ ВЫПОЛНЕНИЯ ПРОЕКТА

БОЛЬШОЙ ОБЪЕМ ДАННЫХ ДЛЯ ОБУЧЕНИЯ МОДЕЛИ



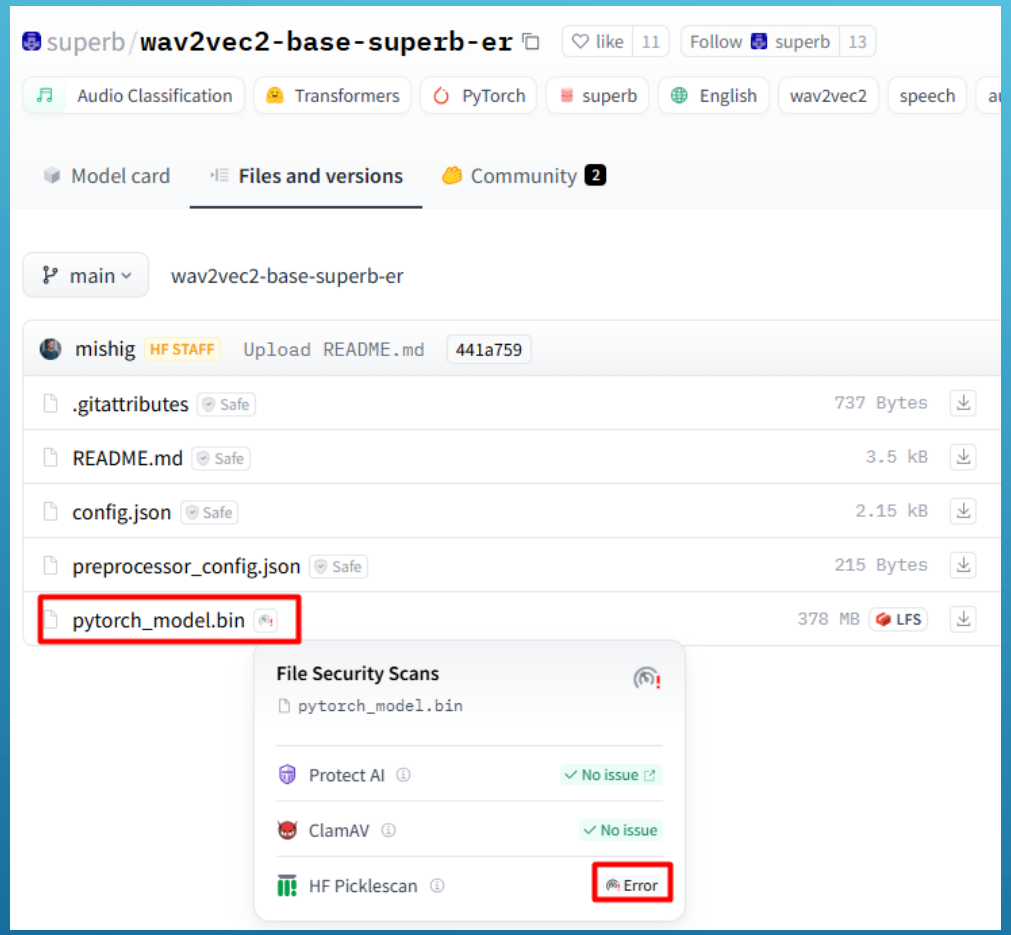


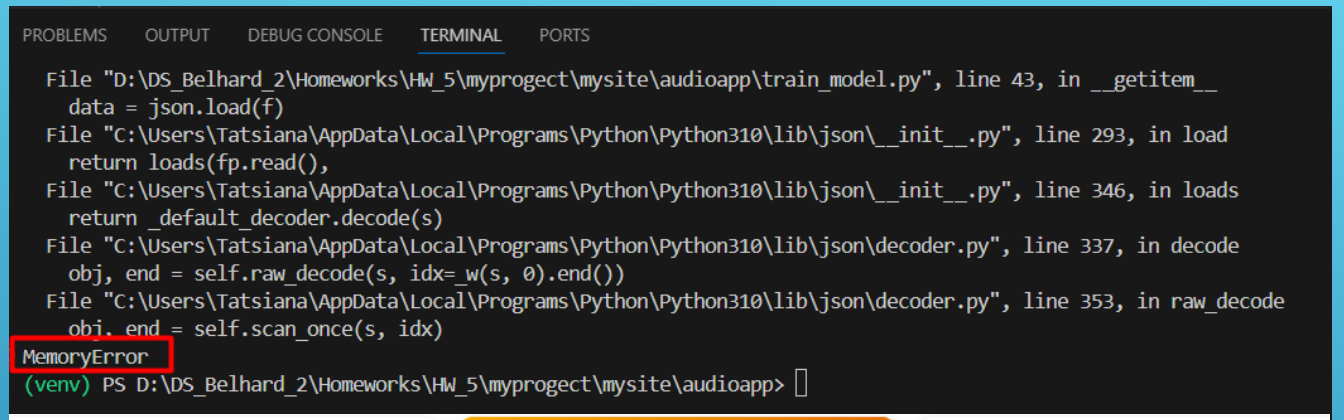


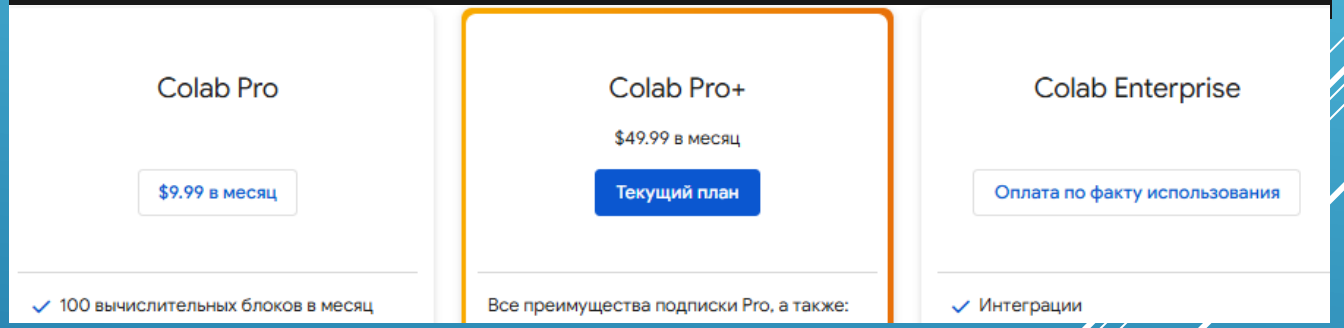
Для примера:
входные данные первого аудиофайла first_audio_file, извлеченные из json после всех необходимых преобразований и визуализированные в office word (4963 страницы, шрифт calibri 11)

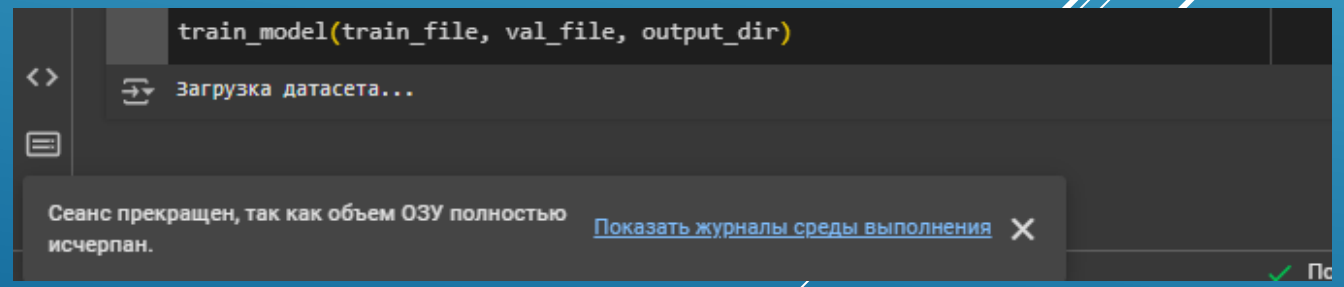
СЛОЖНОСТИ В ХОДЕ ВЫПОЛНЕНИЯ ПРОЕКТА

НЕКОРРЕКТНАЯ РАБОТА ГОТОВЫХ МОДЕЛЕЙ, ОТСУТСТВИЕ ВОЗМОЖНОСТИ ИХ ИСПОЛЬЗОВАНИЯ. ОТСУТСТВИЕ ДОСТАТОЧНОГО КОЛИЧЕСТВА ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ









*! Не всегда корректная обработка файлов первой моделью(в перспективе также требует переобучения)

Отсутствие русскоязычных датасетов и/или их большой объем

!!! ОБУЧЕНИЕ ПРОИЗВОДИЛОСЬ НА V2-8 TPU ПО ПРИЧИНЕ НАЛИЧИЯ БОЛЬШОГО ОБЪЕМА ОЗУ

ВОЗМОЖНОСТИ УЛУЧШЕНИЯ ПРОЕКТА

!!! ПРИ НАЛИЧИИ РЕСУРСОВ !!!

- **Переобучение модели pyannote/speaker-diarization-3.1**

В виде, представленном на Hagging Face, не всегда корректно отрабатывает. Встречается разделение файла на два файла по мужским и женским голосам, а не по количеству спикеров; а также разделение на весь диалог и отдельно последнюю реплику.

- **Расширение датасета для модели GRU**

Использование более разнообразного и большого датасета для обучения модели, чтобы улучшить точность распознавания эмоций.

- **Формирование русскоязычного датасета!**

- **Мультимодальные модели**

Интеграция моделей, которые анализируют не только аудио, но и текст и видео, для более точного определения эмоций.

- **Тонкая настройка**

Переобучение модели на специфических данных, чтобы улучшить её производительность в конкретных сценариях использования.

СПАСИБО ЗА ВНИМАНИЕ

Буду благодарна за обратную связь
+375 (44) 725 73 73
Telegram @TatsianaVladykina