

Министерство цифрового развития  
Федеральное государственное бюджетное образовательное учреждение высшего  
образования  
«Сибирский государственный университет телекоммуникаций и  
информатики»  
(СибГУТИ)  
Кафедра прикладной математики и кибернетики

## Отчёт

по лабораторной работе № 7  
«Распознавание голосовых сообщений с использованием  
готовой модели Whisper от OpenAI»

Выполнил:

студент группы      ИП-312  
Дорогин Н.С.

Работу проверил:    старший преподаватель  
кафедры                      ПМиК  
Дементьева К.И.

Новосибирск 2025 г.

## Введение (задание)

Цель: освоить на практике применение предобученных моделей для транскрибации аудио, оценки качества распознавания и визуализации результатов.

## Задание:

Выбираю для работы свой набор голосовых сообщений (практически все из Telegram).

## Задание 1. Подготовка данных:

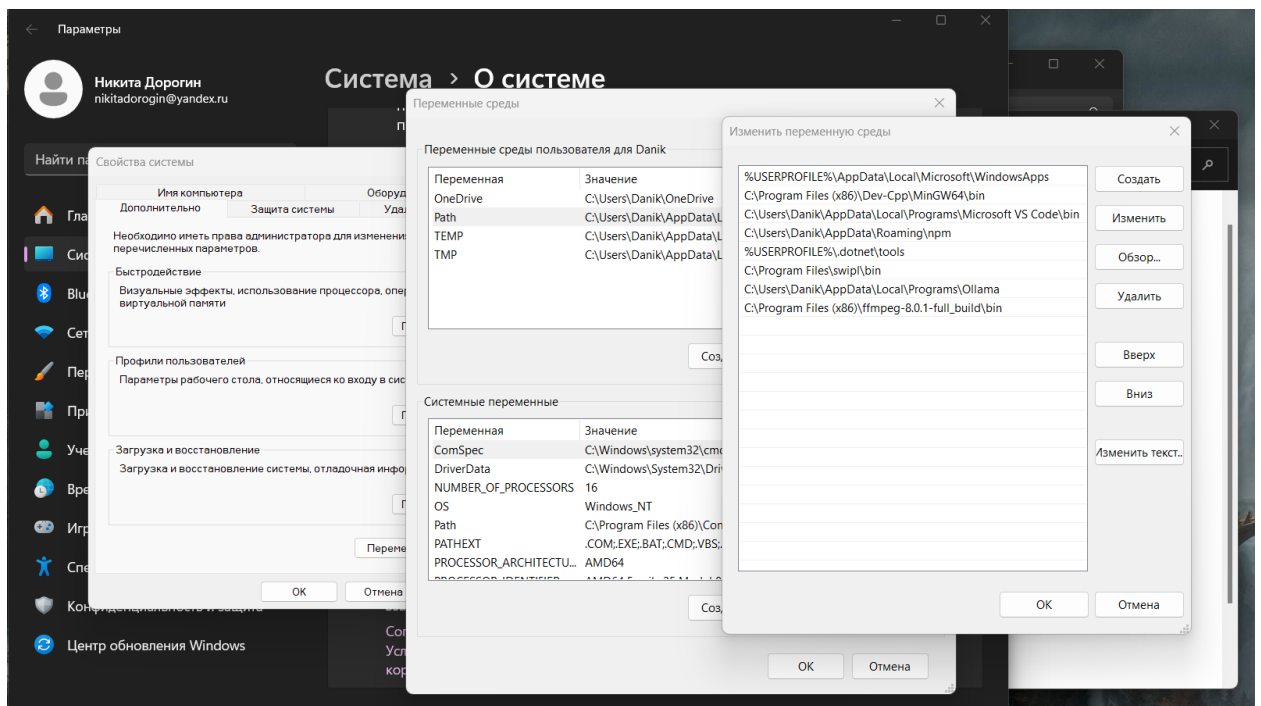
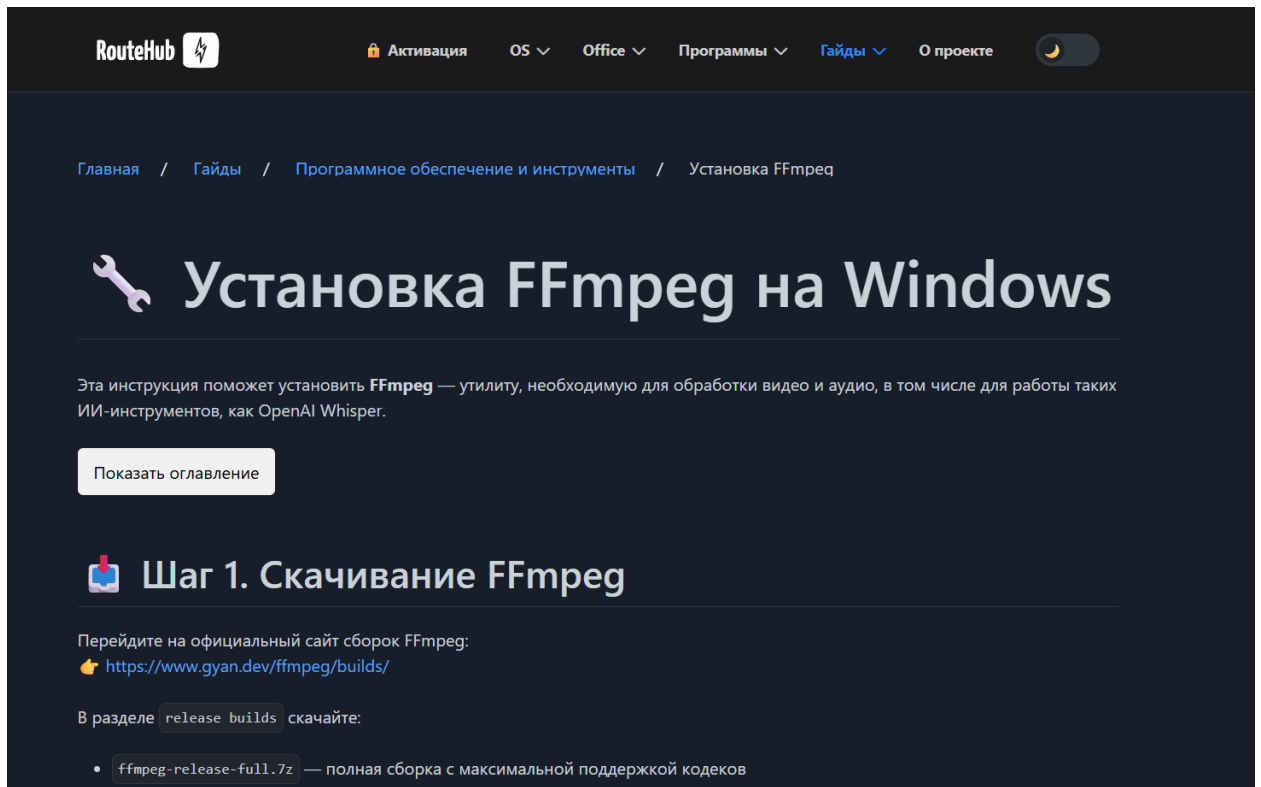
1. Устанавливаем необходимые библиотеки.

Гуглим как установить Whisper и что может помочь в анализе файлов (librosa).

```
PS C:\Users\Danik\Desktop\STUDY_SIBSUTIS_DOROGIN\ММО\лаба7> pip install librosa
Collecting librosa
  Downloading librosa-0.11.0-py3-none-any.whl.metadata (8.7 kB)
Collecting audioread>=2.1.9 (from librosa)
```

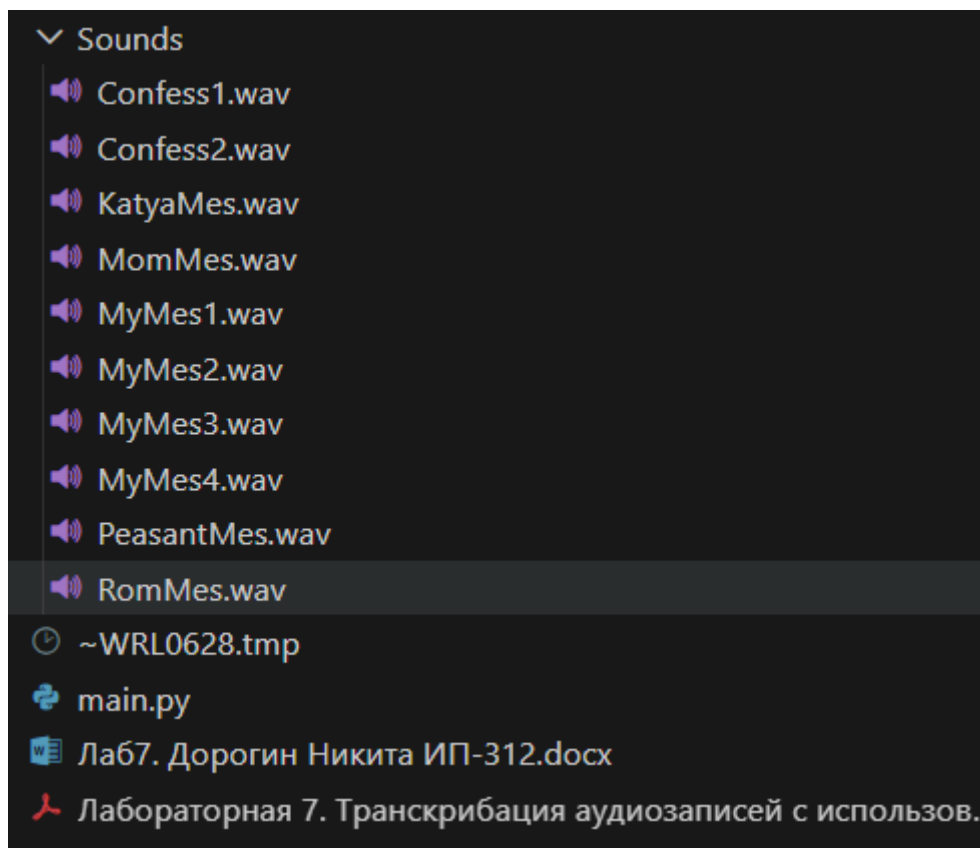
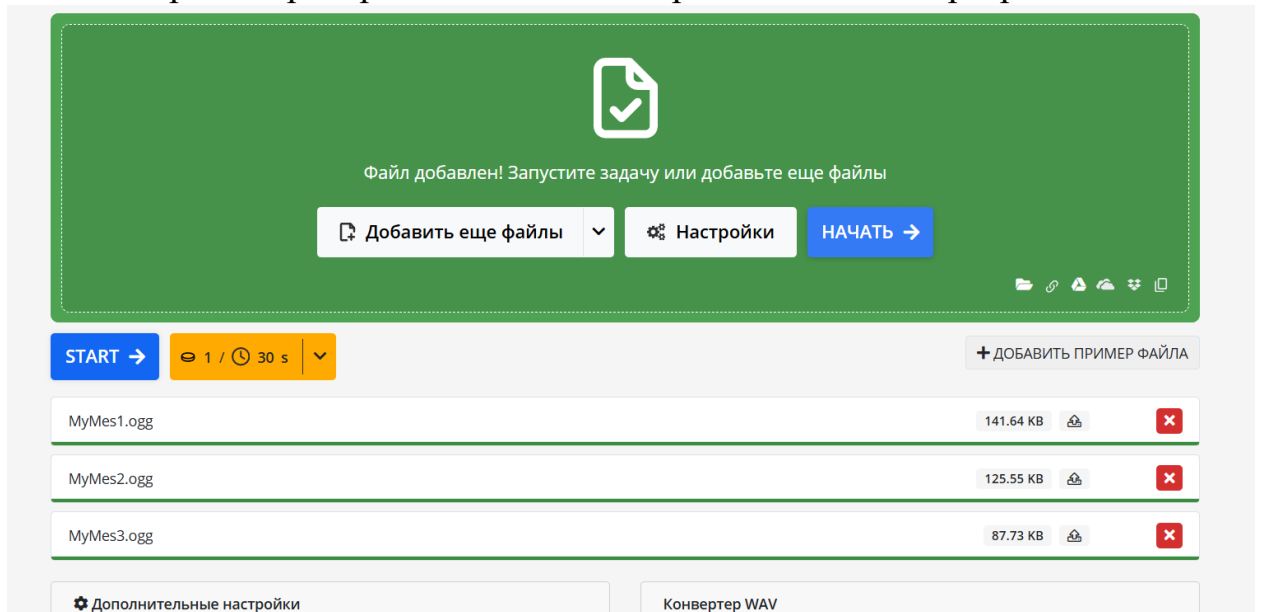
```
PS C:\Users\Danik\Desktop\STUDY_SIBSUTIS_DOROGIN\ММО\лаба7> pip install openai-whisper
Collecting openai-whisper
  Downloading openai_whisper-20250625.tar.gz (803 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 803.2/803.2 kB 5.0 MB/s 0:00:00
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
```

Так же для работы Whisper надо установить FFmpeg. Делаем по гайду с интернета.



2. Загружаем наши аудио.

Я заранее преобразовал их в wav через онлайн-конвертер:



## Задание 2. Транскрибация аудио с помощью модели Whisper:

Создаём дефолтную модель и прогоняем через неё наши аудио по одному и тому же сценарию:

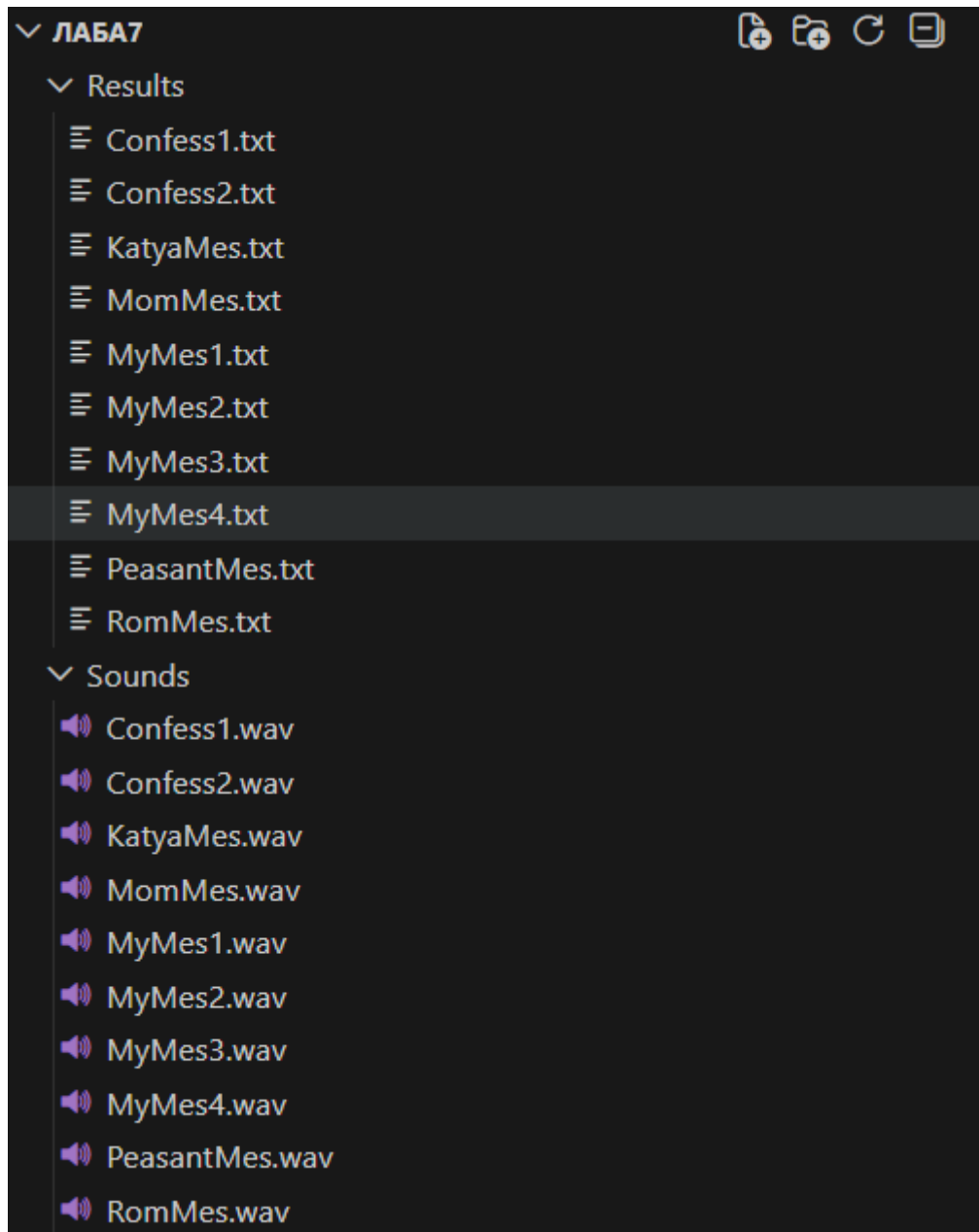
```
model = whisper.load_model("base")

MyMes1 = model.transcribe(".\Sounds\MyMes1.wav")
MyMes1_file = open(".\Results\MyMes1.txt", "w")
MyMes1_file.write(MyMes1["text"])
MyMes1_file.close()
```

При работе есть предупреждение, но оно не критичное:

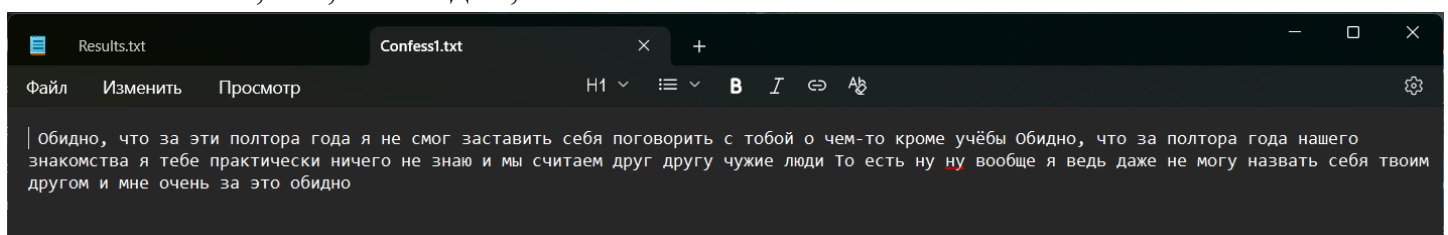
```
in.py"
C:\Users\Danik\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site
-packages\whisper\transcribe.py:132: UserWarning: FP16 is not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
□
```

Результат:



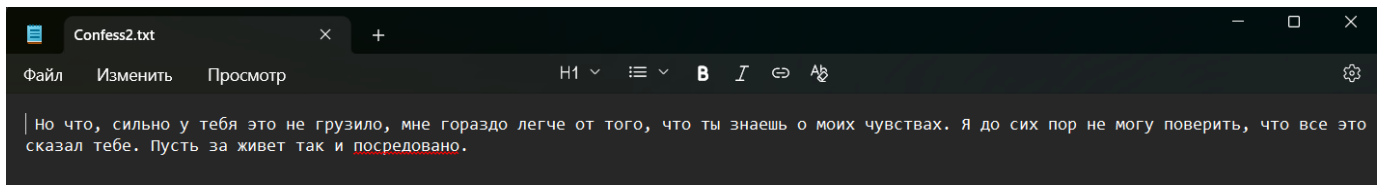
Упс, кажется, кто-то не знает русский язык. Пойдём гуглить как это исправить.

А, нет, не пойдём, я понял.



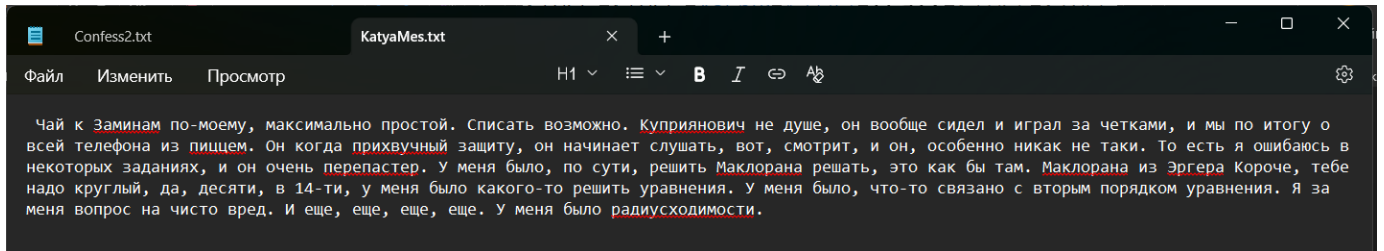
У меня VS Code не поддерживает русский.

Всё, тогда всё работает.

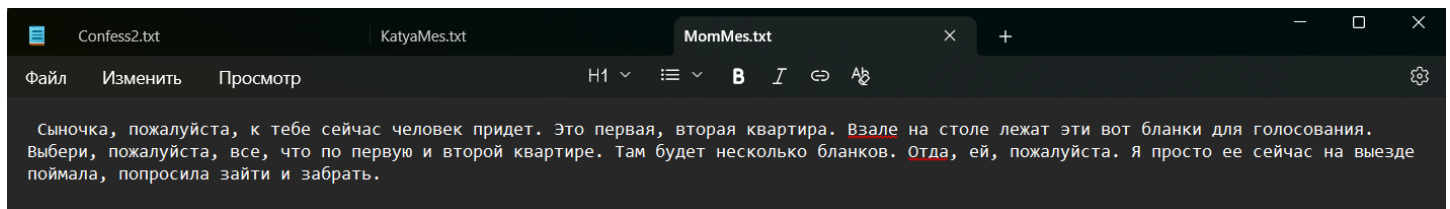


В двух сообщениях выше голос был громким, но его немного приглушал ветер – на улице была метель.

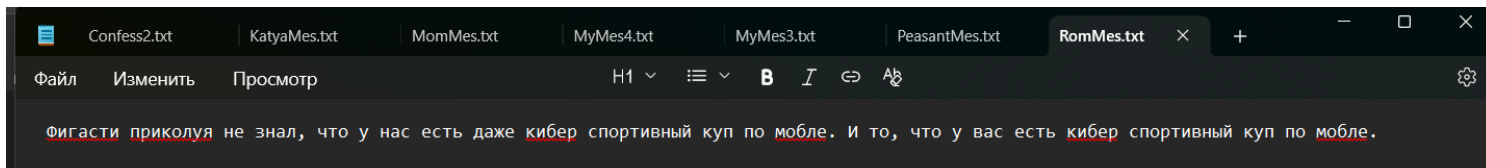
Это сообщение было очень тихим:



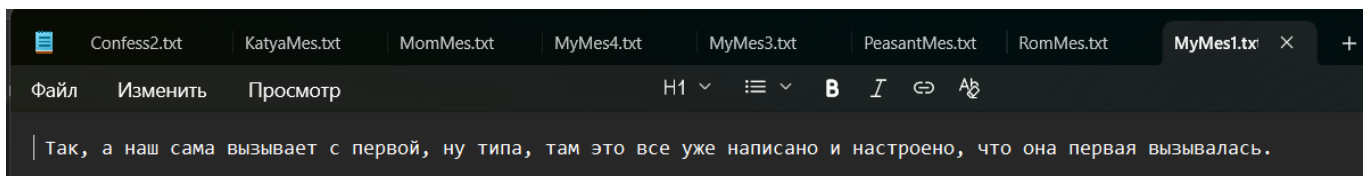
Это сообщение было записано с нормальной громкостью в помещении. Плюс оно единственное из WhatsApp:

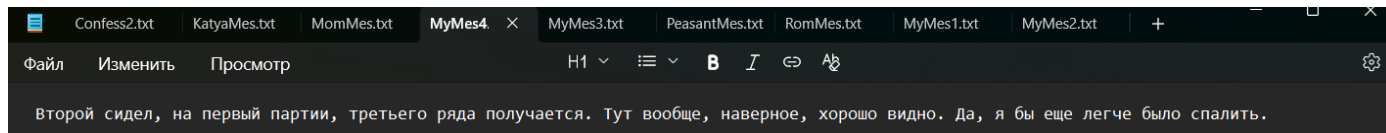
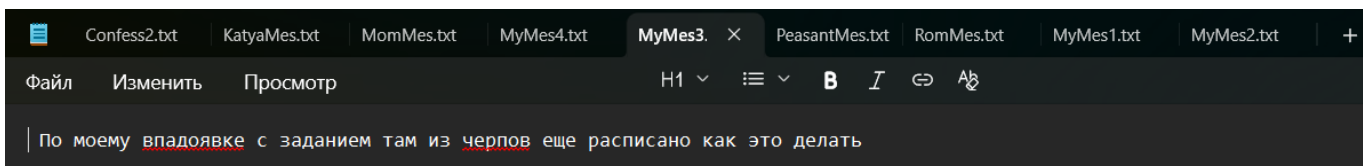
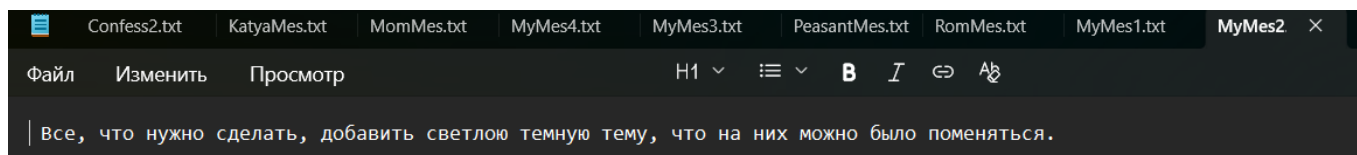


Это тоже без посторонних шумов:

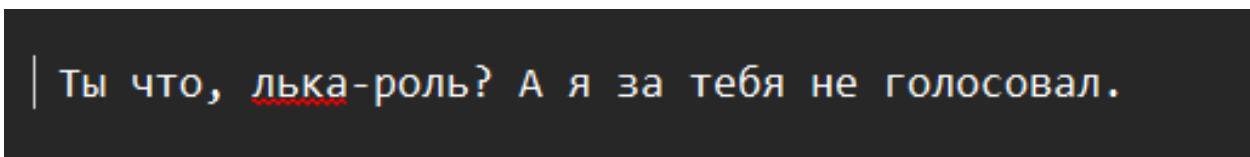


Следующие 4 сообщения были записаны на улице, но в спокойную погоду. На фоне мог быть разве что звук от проезжающих машин.





И последнее было записано довольно тихо. Я пытался звук с ноута записать на телефон и вышло паршивенько, но модель транскрибировала довольно неплохо.



Также я побалуюсь за кадром и протестирую распознавание нецензурной брани. Извините.

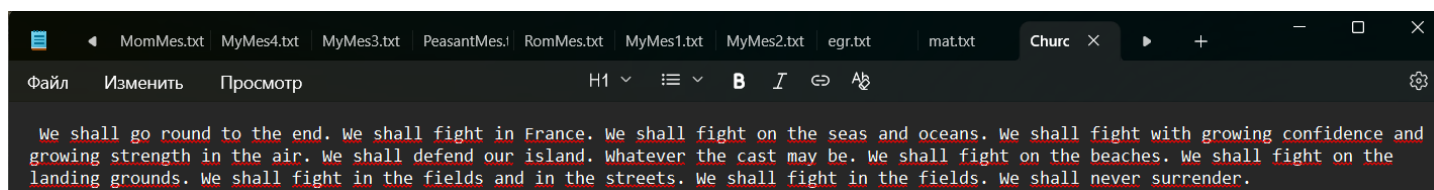
В первом файле он не распознал речь, потому что я слишком сильно кривил голосом и там не так просто распознать речь.

Второй он в принципе распознал правильно, ну, насколько правильно он ранее распознавал фразы.

Ну и напоследок пара экспериментов.

Попробую закинуть Черчилля в формате mp3.

```
chu = model.transcribe("Sounds\Churchill.mp3")
chu_file = open(".\Results\Churchill.txt", "w")
chu_file.write(chu["text"])
chu_file.close()
```



Ну и последний эксперимент – закинуть сообщение в том же формате, что и в мессенджере.



```
wake = model.transcribe("Sounds\Wakeup.ogg")
wake_file = open(".\Results\Wakeup.txt", "w")
wake_file.write(wake["text"])
wake_file.close()
```

Файл    Изменить    Просмотр

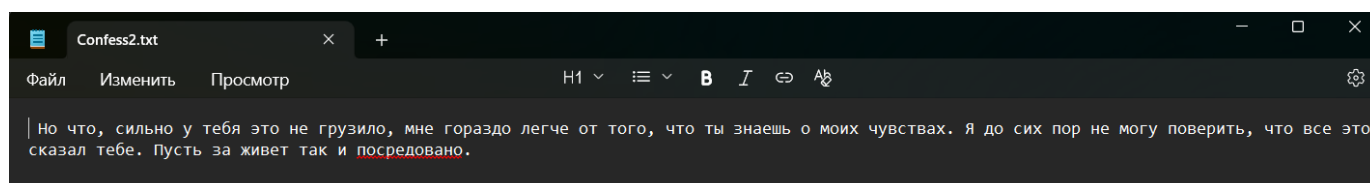
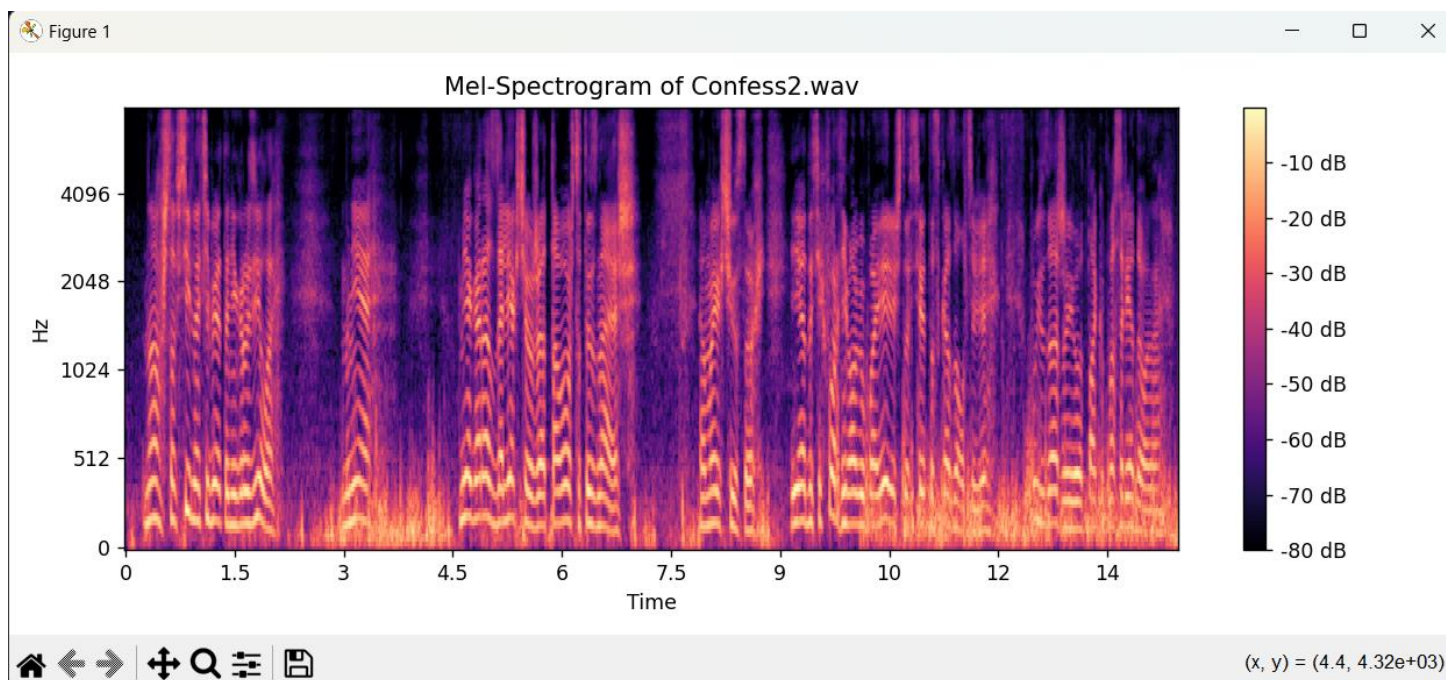
Паря, присыпаться!

В принципе сработало.

### Задание 3. Визуализация и анализ результатов:

```
76  ✓ import librosa
77      import librosa.display
78      import matplotlib.pyplot as plt
79      import numpy as np
80
81      audio_path = r".\Sounds\Confess2.wav"
82      y, sr = librosa.load(audio_path, sr=16000)
83
84      print("Sample rate:", sr)
85      print("Length (sec):", len(y) / sr)
86
87  ✓ s = librosa.feature.melspectrogram(
88      y=y,
89      sr=sr,
90      n_fft=1024,
91      hop_length=256,
92      n_mels=128,
93      fmin=20,
94      fmax=8000
95  )
96
97      s_db = librosa.power_to_db(s, ref=np.max)
98
```

```
plt.figure(figsize=(10, 4))
librosa.display.specshow(
    S_db,
    sr=sr,
    hop_length=256,
    x_axis='time',
    y_axis='mel'
)
plt.colorbar(format='%+2.0f dB')
plt.title("Mel-Spectrogram of Confess2.wav")
plt.tight_layout()
plt.show()
```



Сложно сказать, почему транскрипция на последнем предложении дала сбой.

“Пусть даже и вот...” (не “Пусть за живет...”) в аудиофайле звучит достаточно чётко и лишь на слове опосредованно имеет место дуновение

ветра на фоне. Возможно, на моменте “вот так опосредованно” имело место небольшое ускорение речи, сделавшее её менее распознаваемой.

## **Отчёт:**

GoogleCollab:

<https://colab.research.google.com/drive/1InYZSFYOxjGShodyJXLZvcIH-hd156SS#scrollTo=aF28ht0uim28>