МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**по курсу**

«Data Science»

**Тема:** «**Применение глубоких сверточных нейронных сетей в задаче классификации аудиоданных по гендерному признаку.»**

Слушатель Зотова Татьяна Алексеевна

# 

Москва, 2022

# **Содержание**

[Содержание 2](#_Toc106232838)

[Введение 3](#_Toc106232839)

[1. Аналитическая часть 5](#_Toc106232840)

[1.1. Постановка задачи 5](#_Toc106232841)

[1.2. Описание используемых методов 6](#_Toc106232842)

[1.3. Разведочный анализ данных 13](#_Toc106232843)

[2. Практическая часть 14](#_Toc106232844)

[2.1. Предобработка данных 14](#_Toc106232845)

[2.2. Разработка и обучение модели 30](#_Toc106232846)

[2.3. Тестирование модели 36](#_Toc106232847)

[2.4. Проверка работы модели на собственном голосе 38](#_Toc106232848)

[2.5. Разработка приложения 40](#_Toc106232849)

[2.6. Создание удалённого репозитория и загрузка проекта 44](#_Toc106232850)

[2.7. Заключение 45](#_Toc106232851)

[2.8. Список используемой литературы и веб ресурсы. 46](#_Toc106232852)

# **Введение**

Звук — это физическое явление, представляющее собой распространение в виде упругих волн механических колебаний в твёрдой, жидкой или газообразной среде.

Наша речь состоит из отдельных звуков, воспроизводимых голосовыми связками. Звуки невидимы, но слышимы человеком, поскольку идет воздействие на его барабанные перепонки.

Аудиосигналом называется колебание, соответствующее речи, музыке или другим звучаниям в диапазоне звуковых (слышимых) частот.

В настоящее время по всему миру активно проводятся исследования аудиосигналов. В повседневную жизнь пришли голосовые помощники, умеющие распознавать речь.

Аудиосигнал можно представить в виде массива данных, который возможно визуализировать в график колебаний, пересекающих нулевую границу.

В данной работе мы рассмотрим применение глубоких сверточных нейронных сетей в задаче классификации аудиоданных по гендерному признаку.

Эта тема представляет интерес для банковского сектора и компаний, имеющих собственные контактные центры. В настоящее время по звонку в контактный центр клиент имеет возможность проводить финансовые операции без физического предъявления документа, удостоверяющего личность. Участились случаи, когда под видом клиентов в контактный центр банка для подтверждения операции перевода денежных средств звонят мошенники. Часто мошенники знают персональные данные клиента и кодовые слова, а сам клиент раннее не давал согласие на использование голосовой биометрии. Если позвонивший корректно называет персональные данные, то оператор контактного центра банка обязан провести операцию даже если отчетливо слышит вместо голоса пожилой женщины хриплый мужской бас. Если бы проводилась проверка аудиосигнала и у оператора на экране появлялась информация, что голос позвонившего мужской \ женский, то сотрудник бы имел право не подтверждать операцию без дополнительных проверок. Это привело бы к сокращению случаев мошенничества.

Целью исследовательской работы является разработка модели, способной с высокой долей точности проводить классификацию аудиоданных по гендерному признаку. Для демонстрации работы этой модели мы создадим дружелюбное и доступное пользовательское веб - приложение с высоким юзабилити на фреймворке Flask.



Рисунок - скриншот того какой дизайн может быть у работающего приложения на фреймворке Flask

1. **Аналитическая часть**
   1. **Постановка задачи**

Для исследовательской работы мы будем использовать набор данных TIMIT, В нем содержатся аудиозаписи носителей восьми основных диалектов американского английского языка, для каждого из которых имеется по десять фонетически разнообразных фраз, часть из которых одинаковы для всех дикторов, часть отличается друг от друга. Всего в наборе имеются аудиозаписи 438 дикторов-мужчин и 192 дикторов-женщин. В набор данных TIMIT включено 4620 аудиозаписей в обучающем наборе данных и 1680 аудио в тестовом наборе данных.

Нам необходимо будет на примере аудиофайла показать способы предобработки аудиоданных. Далее мы проведем анализ набора данных TIMIT.

Затем обучим классификатор пола по голосу. Для этого зададим класс timit\_dataloader, который позволяет извлекать метки пола, возраста, идентификатор диктора, а также задает методы предобработки аудиосигнала.

Далее мы зададим архитектуру и обучим нейронную сеть. С помощью метрик оценим качество классификации и выберем модель, дающую лучший результат.

После этого для демонстрации работы модели разработаем приложение с графическим интерфейсом, с помощью которого можно записать голос и проверить корректно ли определяется пол человека по аудиосигналу. Также создадим репозиторий в GitHub и разместим в нем код исследования.

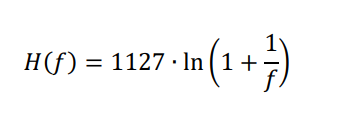
* 1. **Описание используемых методов.**

Речевой сигнал должен быть преобразован в определенный тип параметрического представления для дальнейшего анализа и обработки. Существует достаточно большое количество методов для того, чтобы с помощью вектора признаков представить речевой сигнал. Например, Linear Prediction Coding (LPC), Mel-Frequency Cepstral Coefficients (MFCC) и другие.

Среди них MFCC – метод мел-частотных кепстральных коэффициентов, является самым известным и популярным. Этот метод мы и будем использовать.

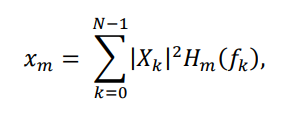
Метод основан на изменении человеческого голоса с критической пропускной способностью с использованием частотных треугольных фильтров. Они расположены с интервалами линейно низких частот и с логарифмическими высокими частотами для получения фонетически важных характеристик речи. Размеры окон рассчитываются при помощи мел-шкалы: первое окно очень узкое и показывает, сколько энергии содержится около нуля герц, по мере возрастания частоты размер окна становится шире, так как с ростом частоты звуки менее различимы человеческим ухом. MFCC обычно рассчитываются с использованием набора фильтров треугольной формы с фильтром центральной частоты, который расположен с линейными частотными интервалами менее 1000 Гц и логарифмически выше 1000 Гц. Пропускная способность каждого фильтра определяется центральными частотами двух соседних фильтров и зависит от частотного диапазона наборов фильтров и количества фильтров.

Для человеческой слуховой системы фильтры имеют собственную полосу пропускания, которая связана с центральной частотой фильтра. Психофизические исследования показали, что восприятие человеком содержания частоты звуков для речевых сигналов не зависит от линейных шкал. Для применения мел-фильтров представления сигнала переносится от частоты Гц к высоте мел звука, используется формула, которая описывает зависимость:

 (1)

где 𝑓 – частоты по обычной (линейной) шкале; 𝐻(𝑓) – частоты по мел-шкале.

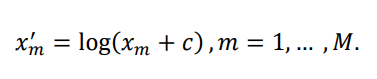
Энергия сигнала, которая попадает в каждое из окон анализа, получается перемножением векторов энергетического спектра сигнала и оконной функции:

 (2)

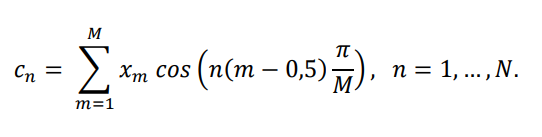
где 𝑥𝑚 – энергетический коэффициент; 𝑚 = 1, … , 𝑀 – количество фильтров;

𝑋𝑘 – амплитудные коэффициенты спектра сегмента; 𝐻𝑚(𝑓) – функция.

В результате вычисления получается набор коэффициентов 𝑥𝑚, содержащих спектральную информацию речевых сегментов. После вычисления энергии, выполняется логарифмирование коэффициентов. Человек может воспринимать громкость в нелинейной шкале – для удвоения воспринимаемой громкости необходимо увеличить энергию в 8 раз. За счет логарифмирования достигается эффективное сжатие пространства признаков. Но логарифм малых значений стремится к минусу бесконечности. Чтобы обойти этот эффект, можно применить метод маскировки, добавляя к значениям 𝑥𝑚 некоторую константу:

 (3)

Поскольку коэффициенты спектра mel являются действительными числами, мы можем преобразовать их во временной интервал, используя дискретное косинусное преобразование:

 (4)

Шкала Мел соотносит воспринимаемую частоту или высоту чистого тона (мел) с фактической измеренной частотой (Гц). Люди гораздо лучше различают небольшие изменения высоты звука на низких частотах, чем на высоких. Эта зависимость не совсем линейная и описывается следующей формулой:

**M(f) = 1127 ln(1 + f /700)** (5)

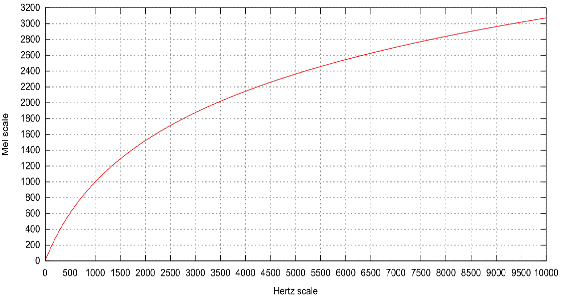


Рисунок 2 Шкала Мел

Вычисление мел-частотных кепстральных коэффициентов включает в себя следующие шаги:

1. Необходимо разделить исходный сигнал на фреймы. Размер фрейма обычно выбирается от 20 до 40 мс, так как считается, что речевой сигнал на этом промежутке не сильно меняется.
2. Речевой сигнал конечен и не является периодическим, поэтому из-за разрывов на его концах при применении преобразования Фурье проявляется эффект утечки. Для того, чтобы снизить его влияние на результат, каждый кадр умножается на оконную функцию Хемминга.
3. Вычисляем периодограмму для каждого фрейма (спектральную мощность)
4. Вычисляем блок мел-фильтров. Для этого треугольные фильтры (от 20 до 40) умножаются на периодограмму и суммируются. В результате мы получим энергии набора фильтров.
5. Полученные энергии логарифмируются. Это также мотивируется человеческим слухом: мы не слышим громкость в линейном масштабе. Обычно, чтобы удвоить воспринимаемую громкость звука, нам нужно затратить в 8 раз больше энергии. Это означает, что большие колебания энергии могут звучать не так уж и по-другому, если звук с самого начала громкий. Эта операция сжатия делает наши функции более близкими к тому, что на самом деле слышат люди.
6. Далее, используя дискретное косинусное преобразование, получим мел-кепстральные коэффициенты.

Визуально процесс представлен ниже на рисунках.

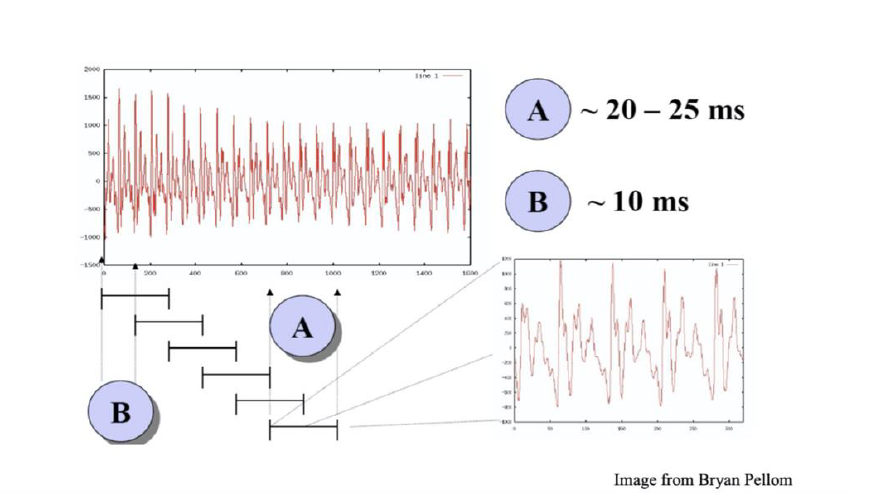


Рисунок 3 Разбиение на фреймы

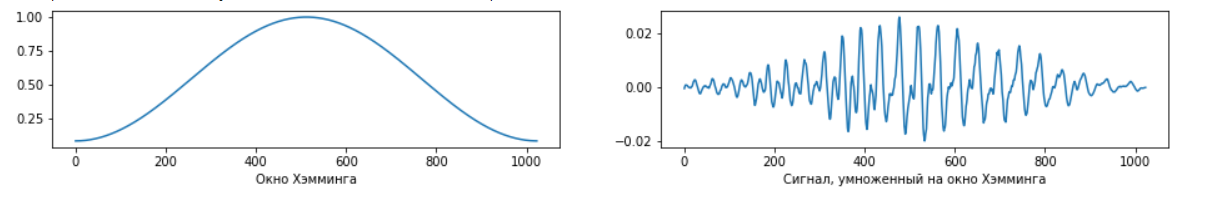


Рисунок 4 Сигнал, умноженный на окно Хэмминга

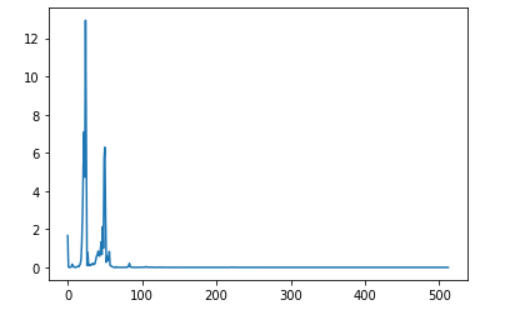


Рисунок 5 Периодограмма

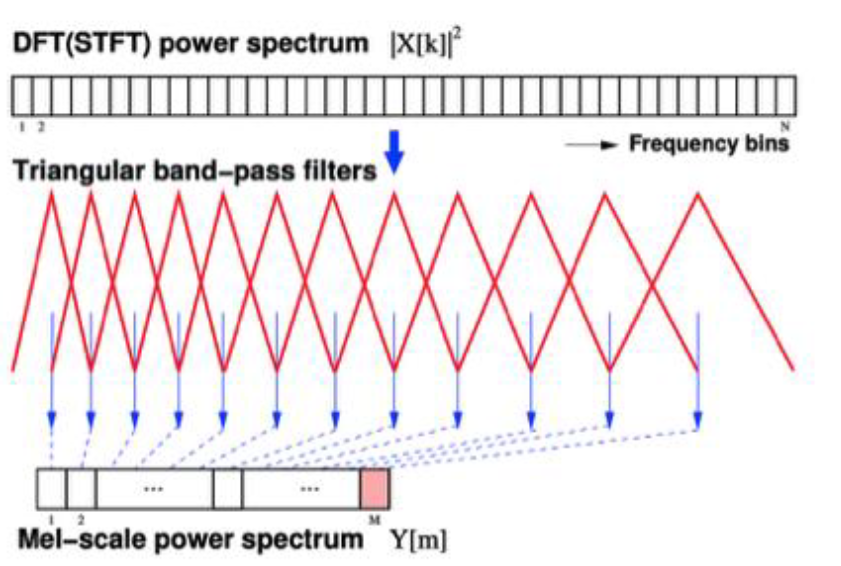


Рисунок 6 Блок мел-фильтров

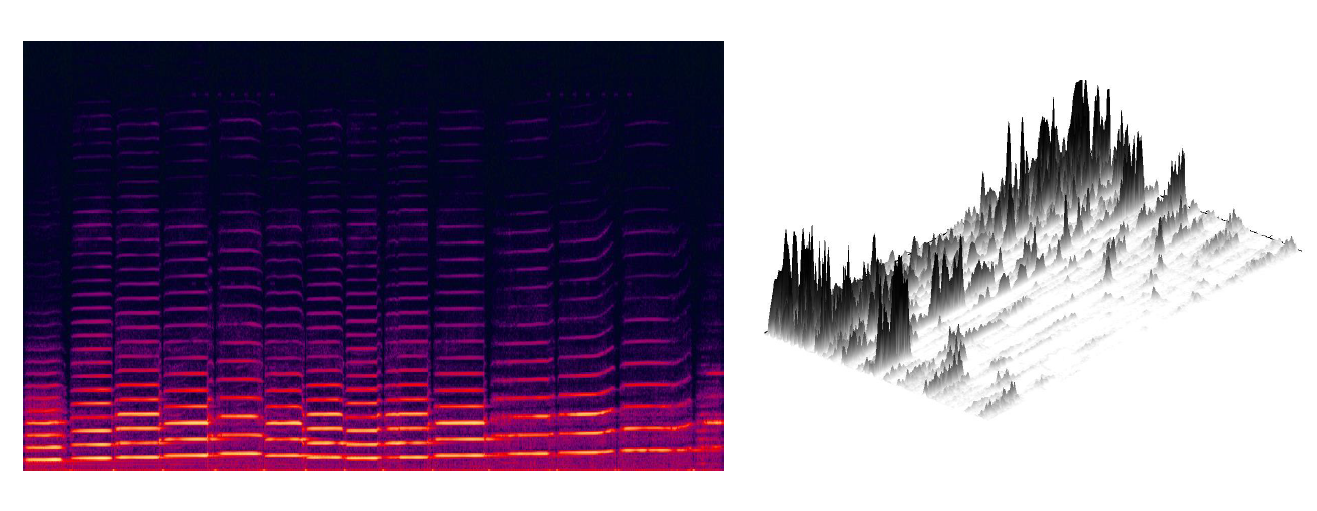


Рисунок 7 Мел- кепстральные коэффициенты

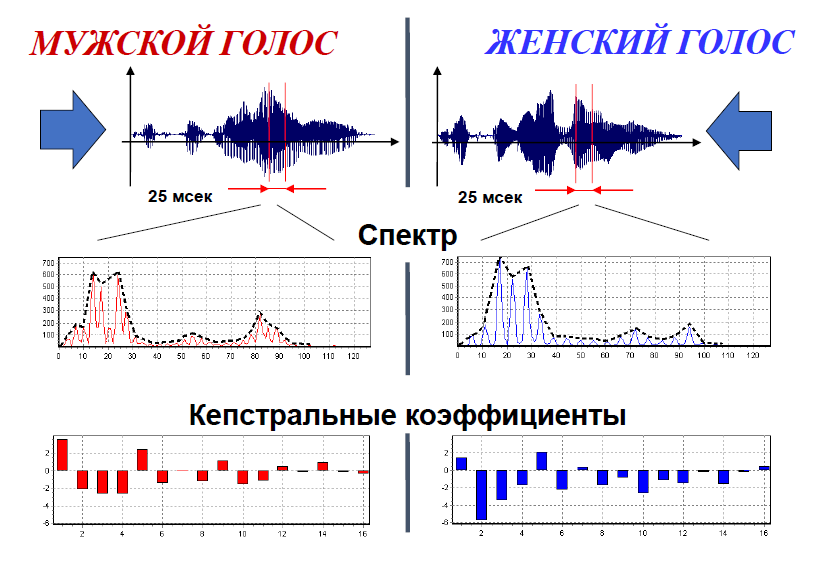


Рисунок 8 Сравнение мужского и женского голоса

* 1. **Разведочный анализ данных**

Прежде чем передать данные в работу моделей машинного обучения, необходимо обработать и очистить их. Очевидно, что «грязные» и необработанные данные могут содержать искажения и пропущенные значения – это ненадёжно, поскольку способно привести к крайне неверным результатам по итогам моделирования. Но безосновательно удалять что-либо тоже неправильно. Именно поэтому сначала набор данных надо изучить.

Цель разведочного анализа - получение первоначальных представлений о характерах распределений переменных исходного набора данных, формирование оценки качества исходных данных (наличие пропусков, выбросов), выявление характера взаимосвязи между переменными с целью последующего выдвижения гипотез о наиболее подходящих для решения задачи моделях машинного обучения.

Для аудиоданных разверточный анализ будет достаточно сильно отличаться от типового набора с табличными данными.

Набор аудиофайлов TIMIT не содержит шумов, темп голоса спокойный, речь записана профессиональными дикторами

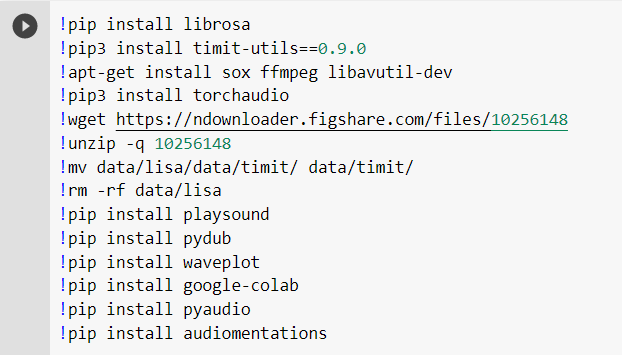
Чтобы продемонстрировать основные методы предобработки аудиоданных мы намеренно провели аугментацию данных. Чтобы исказить речь на аудиофайлах мы

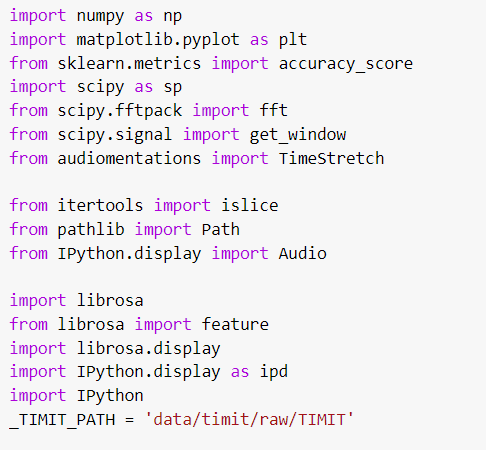
Использовали ускорение\замедление.

Чтобы аугментированные данные очистить мы провели мел-кепстральный анализ, очистку данных и на основании этого построили модель нейронной сети.

Анализировались как отдельные аудиозаписи так и весь набор данных TIMIT.

1. **Практическая часть**
   1. **Предобработка данных**
2. Произведем установку библиотек, необходимых нам для исследовательской работы.





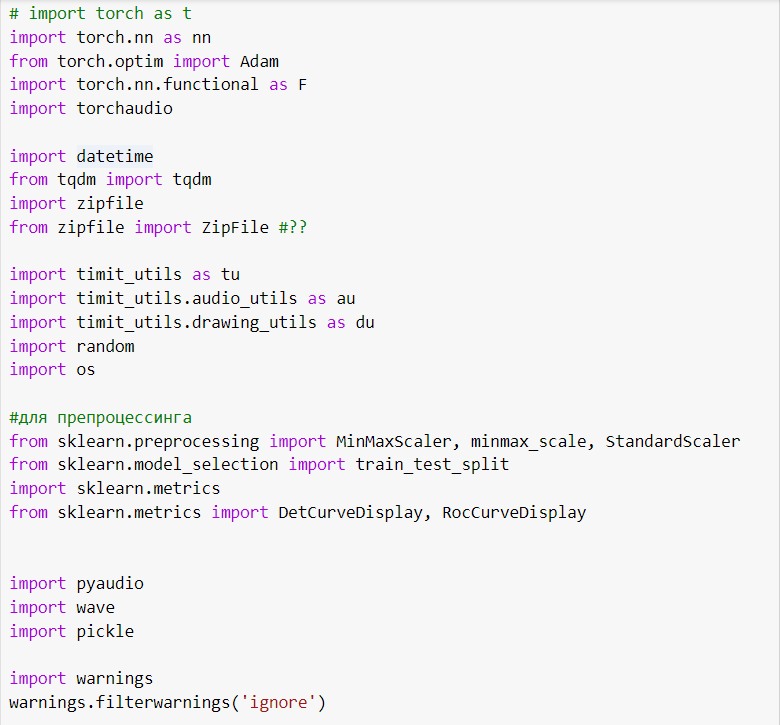
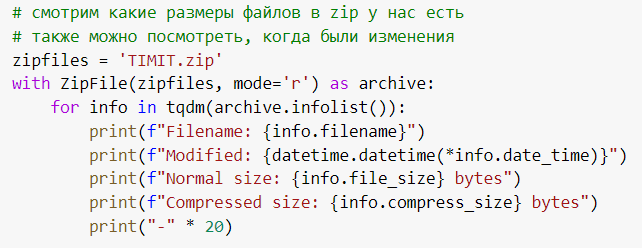
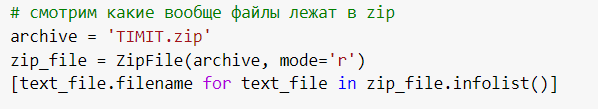
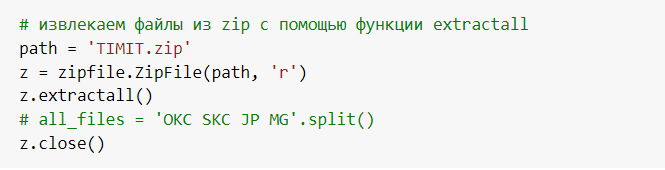


Рисунок 9 Пример кода

1. Загрузим файлы и проверим какие файлы находятся в zip архиве.







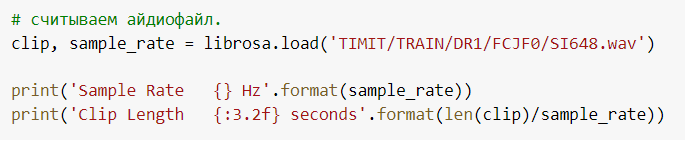
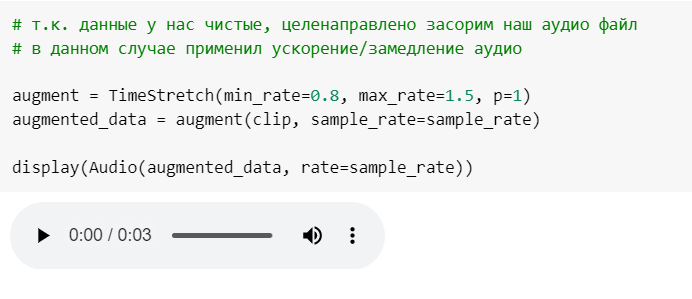
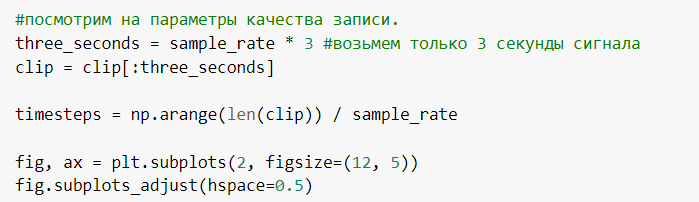
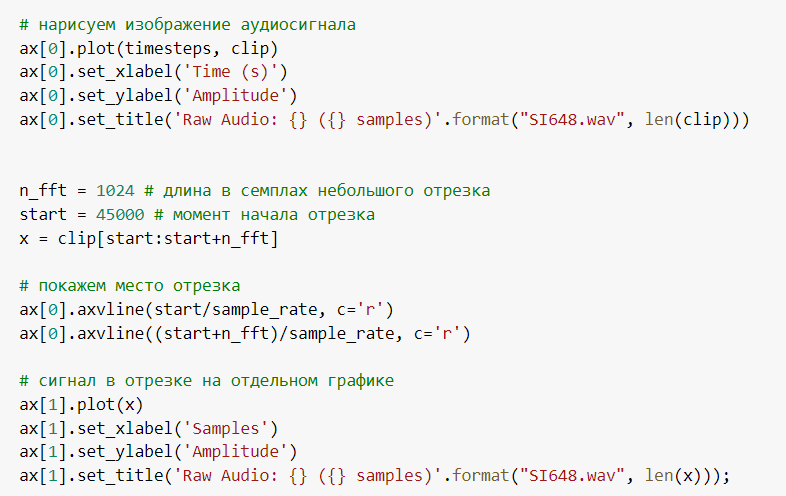


Рисунок 10 Пример кода

1. Предобработка данных на примере аудиофайла.







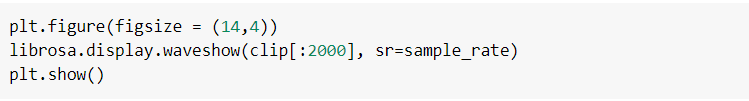
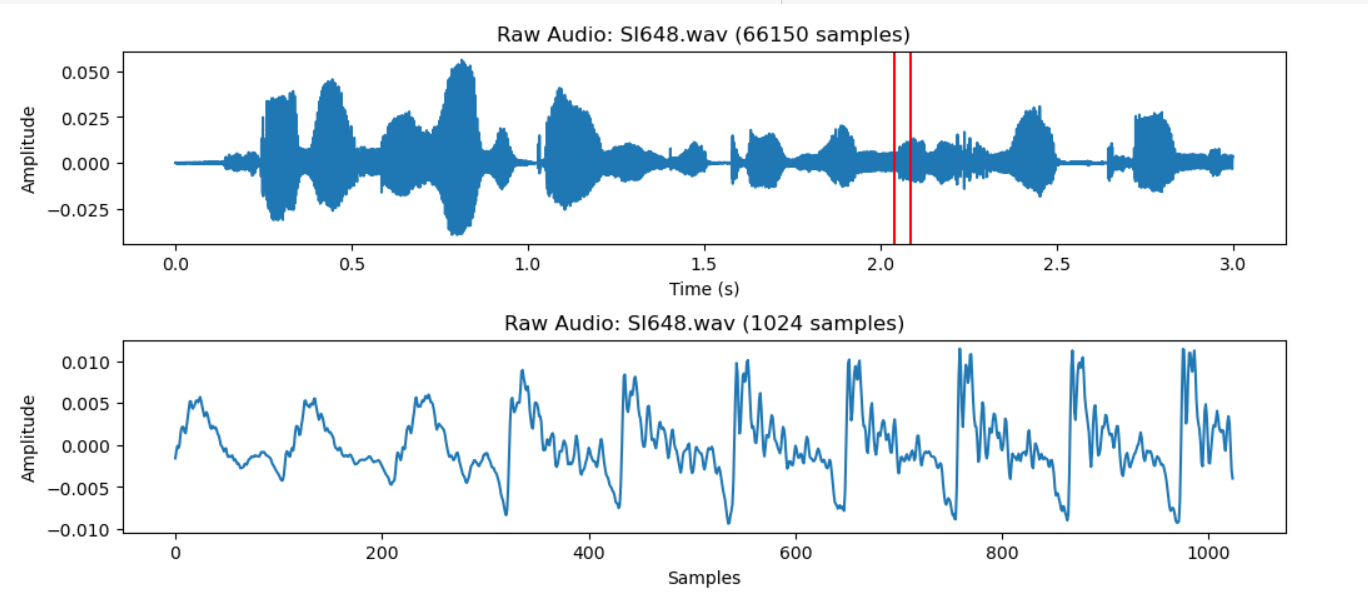


Рисунок 11 Пример кода

Результатом работы кода будут следующие графики:



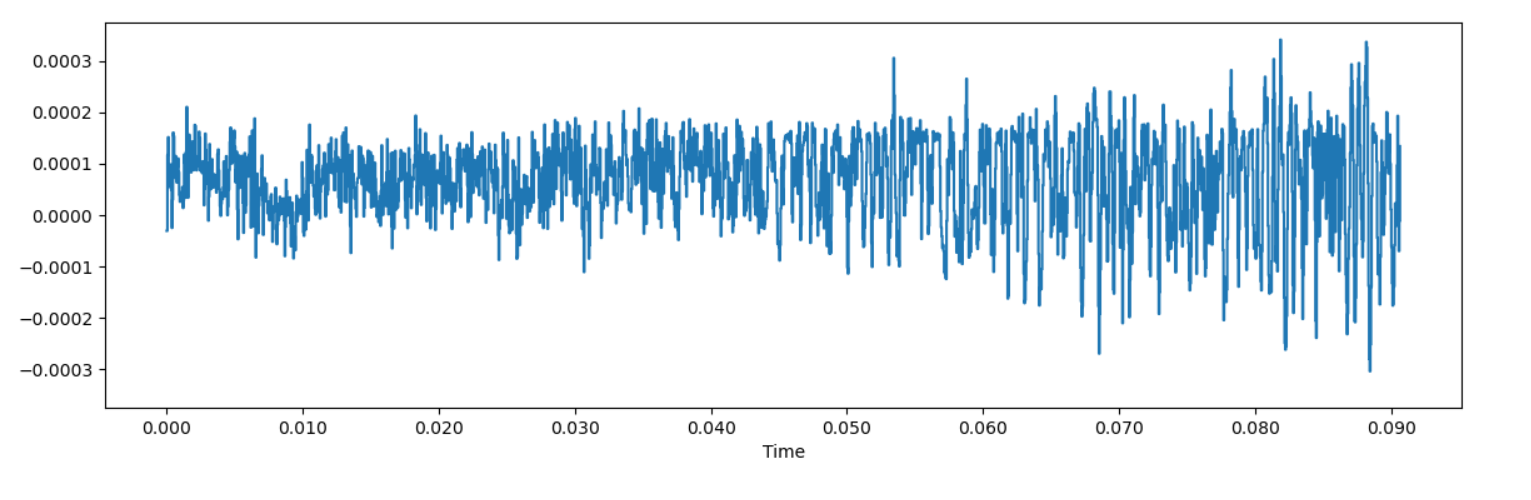


Рисунок 12 Графики

Проведем параметризацию.

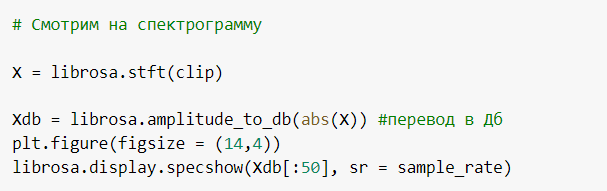


Рисунок 13 Пример кода

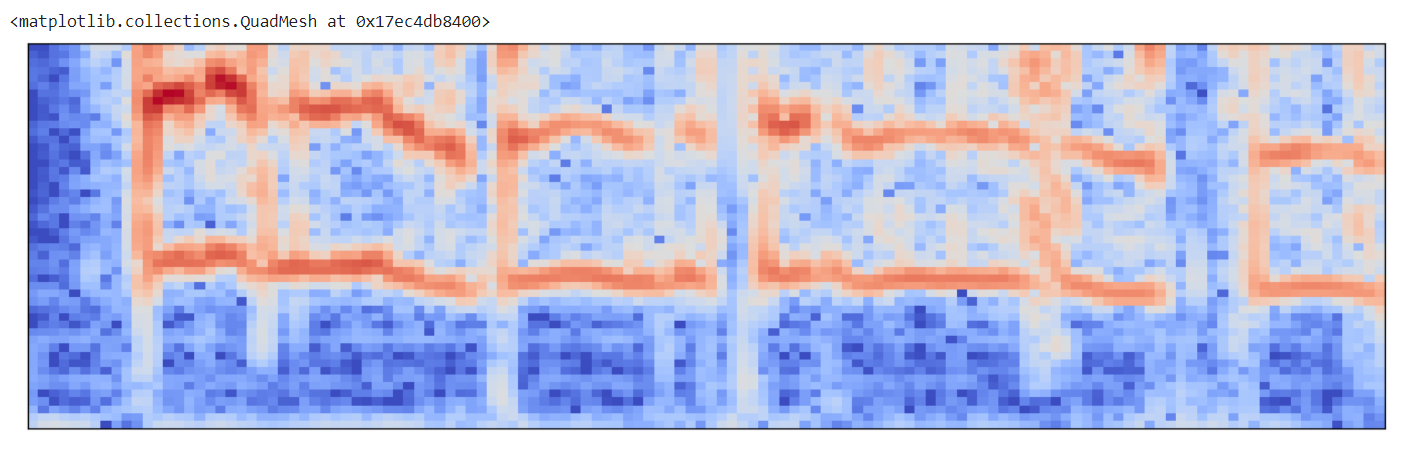


Рисунок 14 Спектрограмма



Рисунок 15 Пример кода

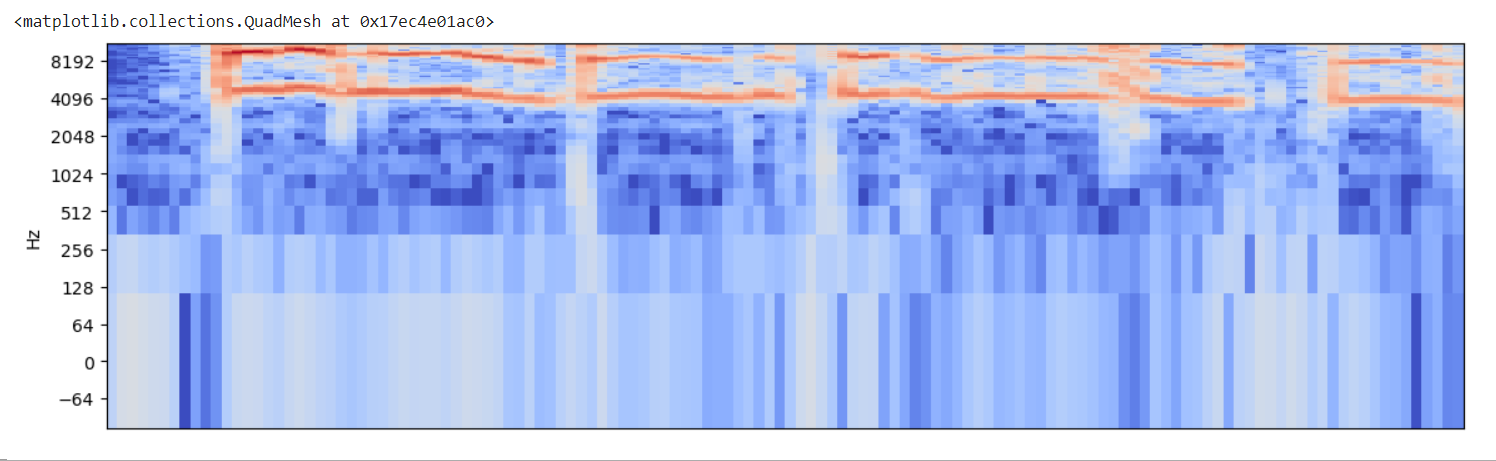


Рисунок 16 Спектрограмма логарифмированная

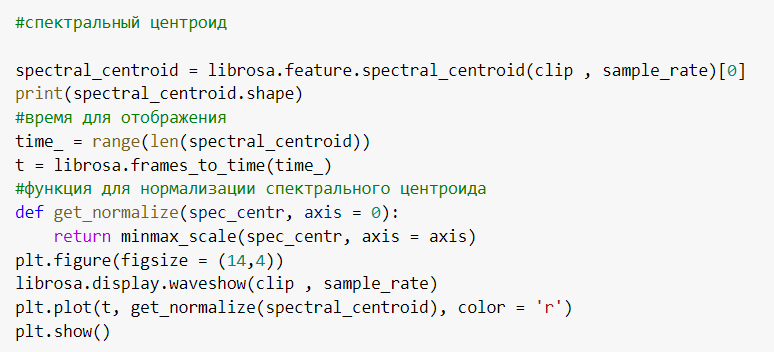


Рисунок 17 Пример кода

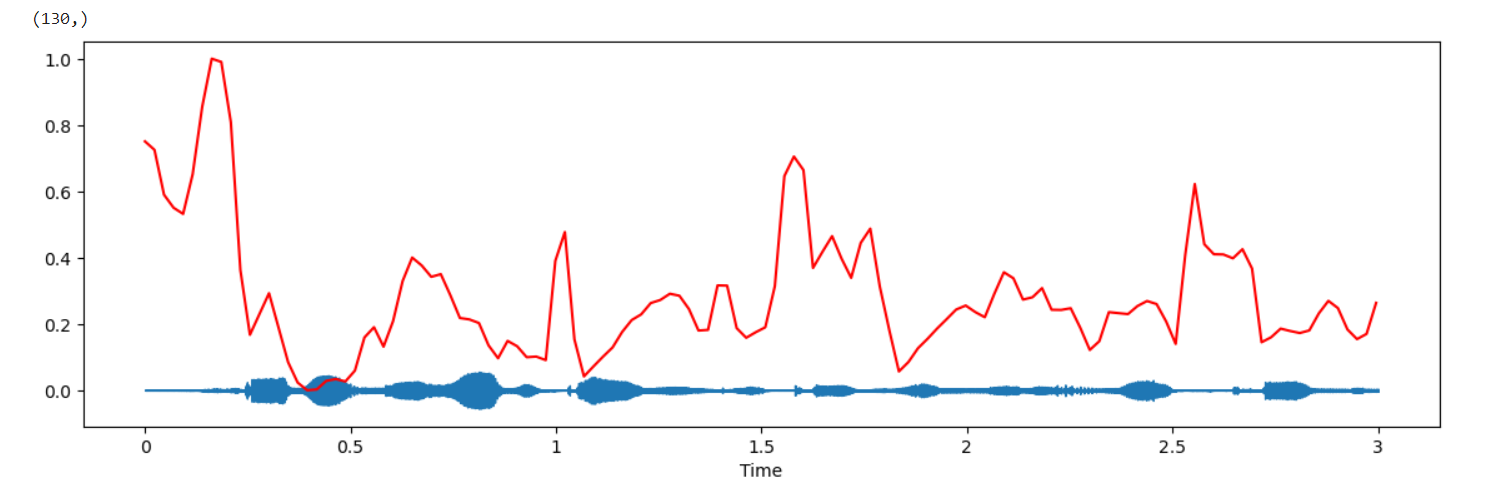


Рисунок 18 Спектральный центроид.

Чтобы получить признаки из аудиосигнала, которые можно подавать на классификатор, вычислим значения мел-кепстральных коэффициентов для данного сигнала.

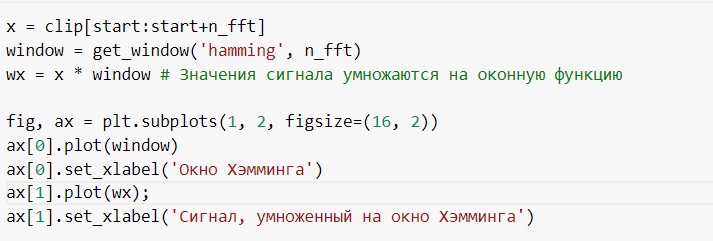


Рисунок 19 Пример кода

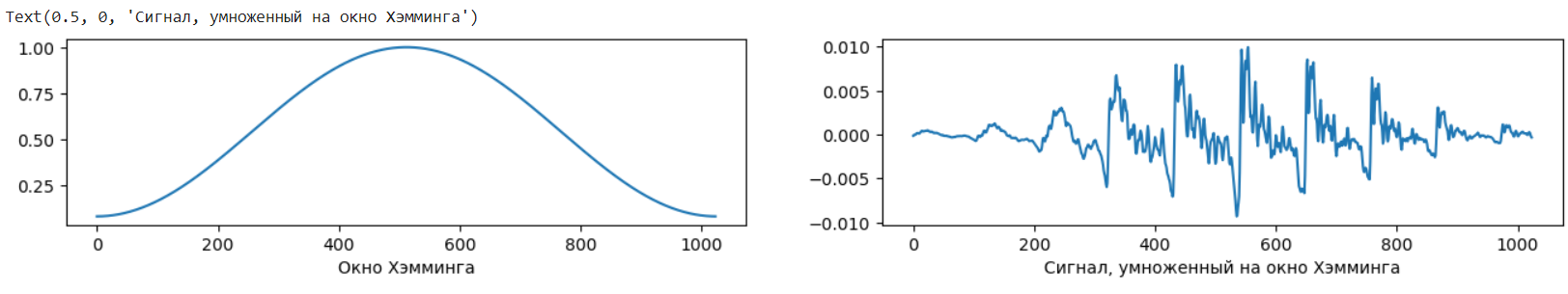


Рисунок 20 Графики

Используем быстрое преобразование Фурье в выбранном окне.

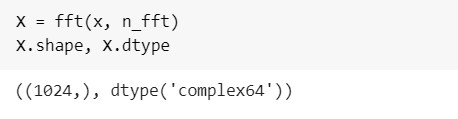


Рисунок 20 Пример кода с результатом его работы.

Вычисляем периодограмму

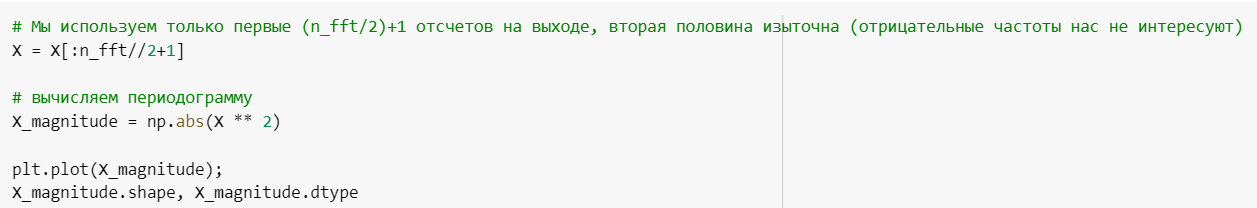


Рисунок 21 Пример кода

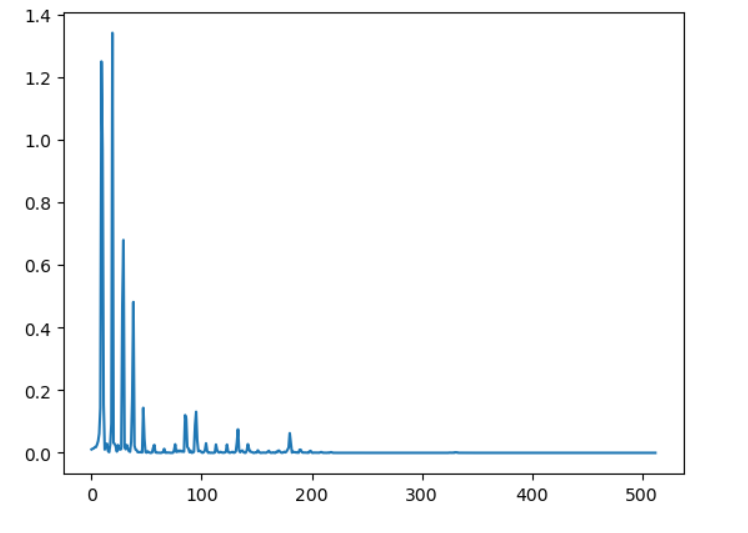


Рисунок 22 График

Вычисляем блок мел-фильтров. Для этого треугольные фильтры (от 20 до 40) умножаются на периодограмму и суммируются. В результате мы получим энергии набора фильтров. Фильтры собираются в области низких частот, обеспечивая более высокое "разрешение" там, где это необходимо для распознавания.

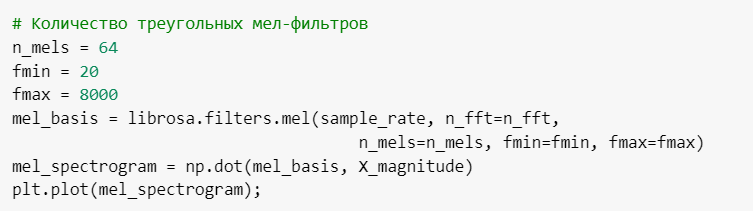


Рисунок 23 Пример кода

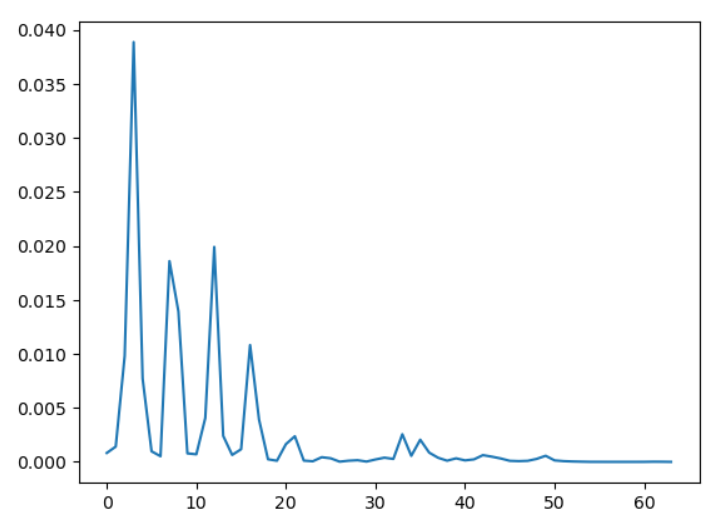


Рисунок 24 График

Логарифмируем полученные значения.

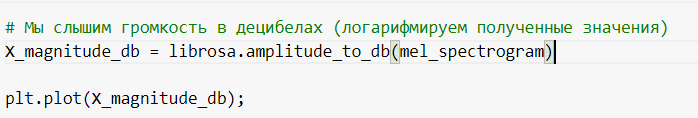


Рисунок 25 Пример кода

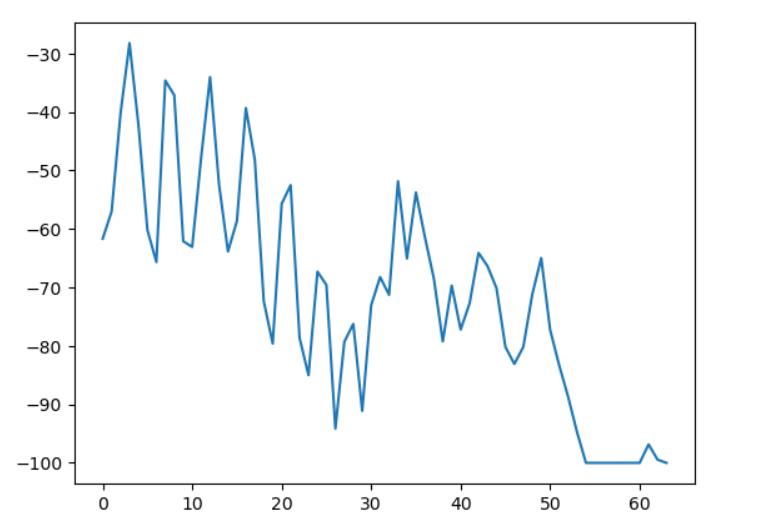


Рисунок 26 График

Применить ко всему сигналу, а не к одному окну и получим следующую картину:



Рисунок 27 Пример кода

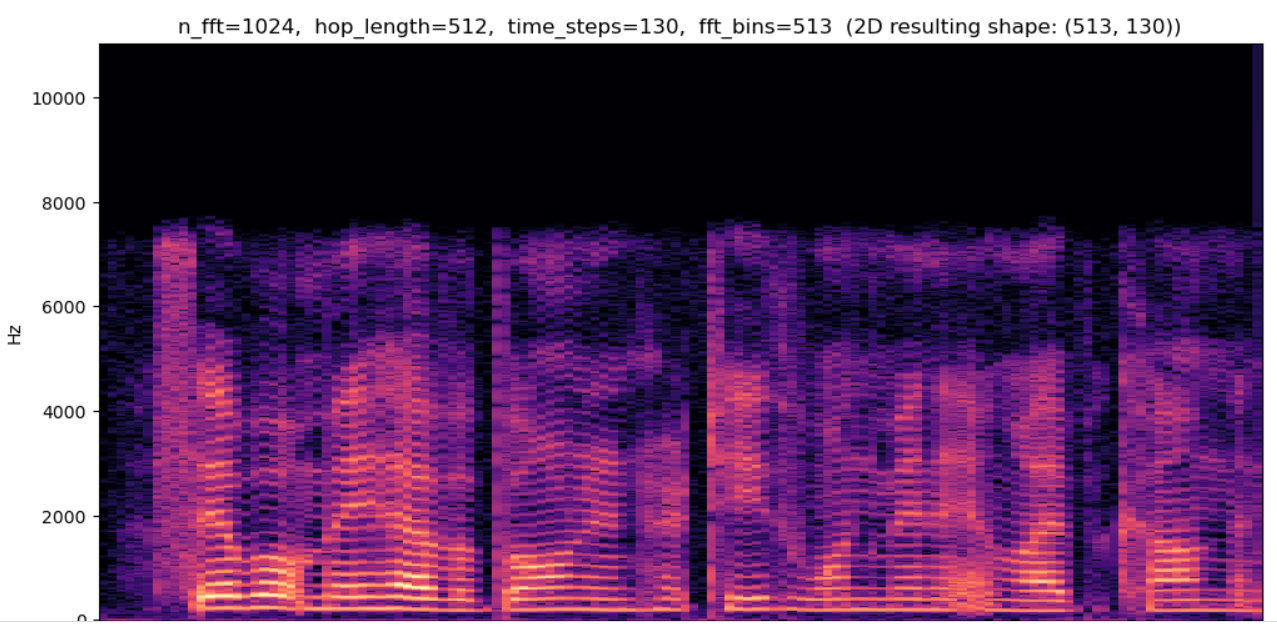


Рисунок 28 График

Применим дискретное косинусное преобразование

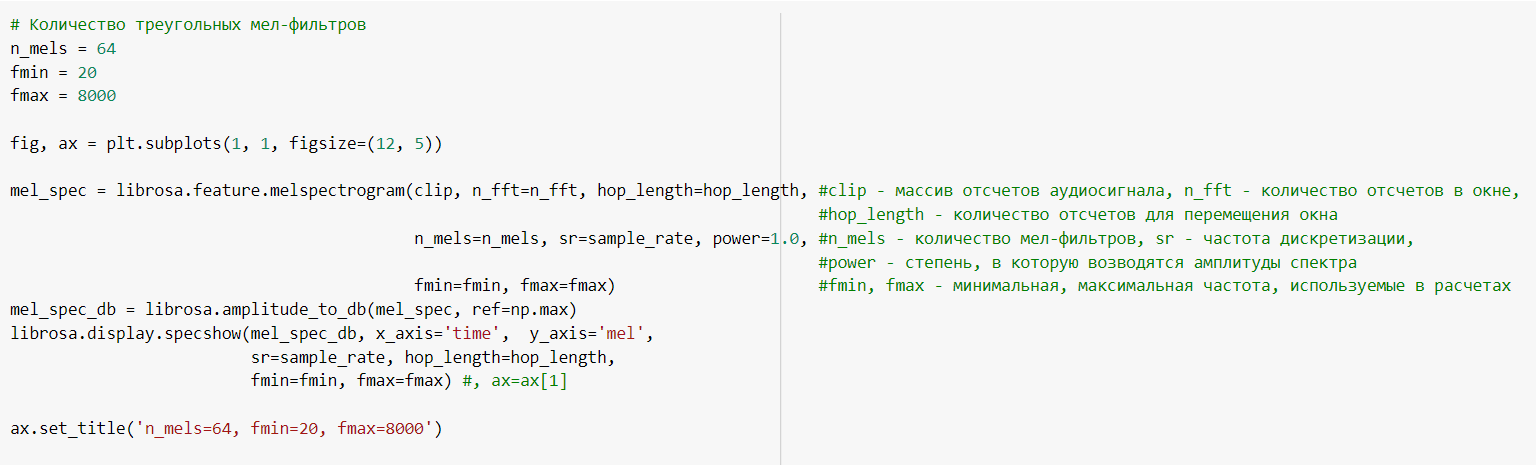


Рисунок 29 Пример кода

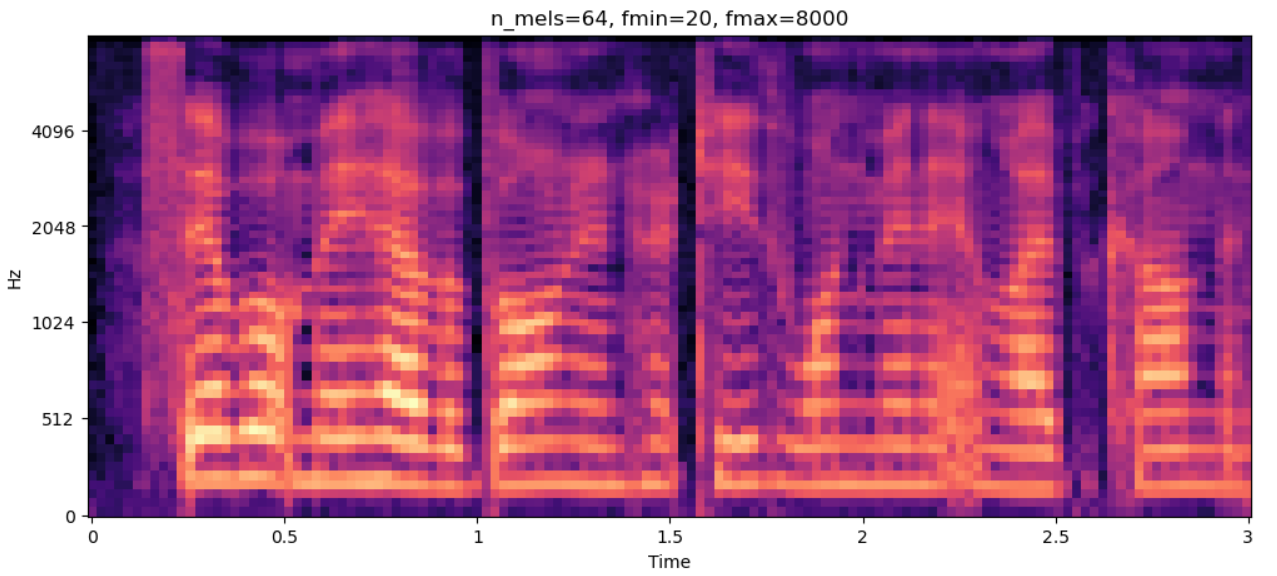


Рисунок 30 График.

Загрузим наши данные, содержащиеся в наборе TIMIT в корпус. Он содержит два подкорпуса, обучающую и тестовую выборку.

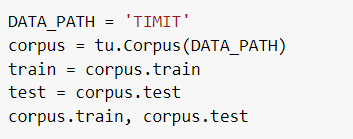


Рисунок 31 Пример кода

Отобразим аудио предложения, слова и транскрипций фонем



Рисунок 32 Пример кода с результатом выполнения.

Создадим Датафрейм, содержащий количество слов, произнесенных дикторами. Речь в наборе данных на английском языке. Артикль «a» характерен для английского языка, по его наличию можно проверить были ли пропуски. Убедимся, что в нашем случае пропуски отсутствуют.

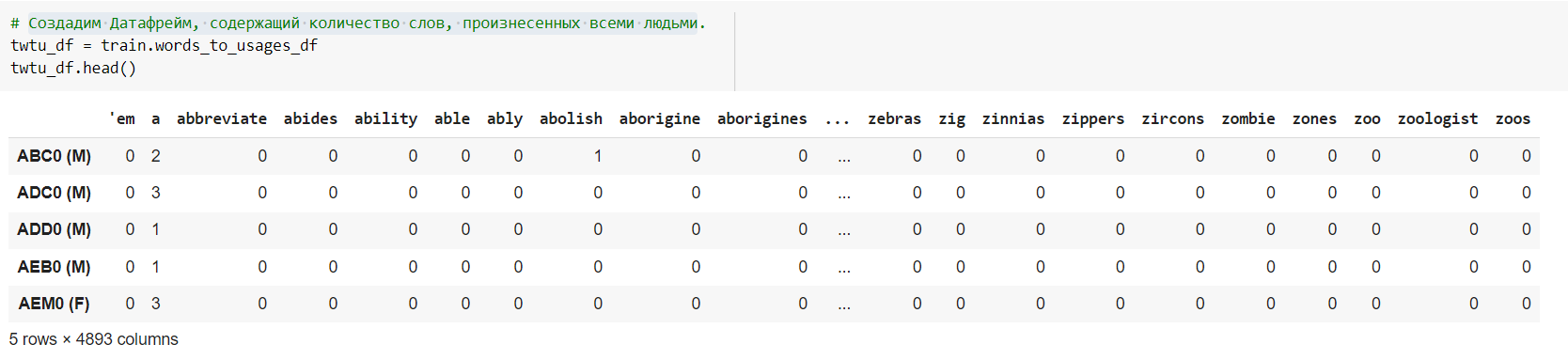
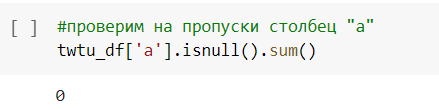


Рисунок 33 Пример кода с результатом выполнения.

Нас интересует столбец а из рисунка 33.



Пропуски отсутствуют. Создадим Датафрейм, содержащий количество фонем, произнесенных всеми людьми.

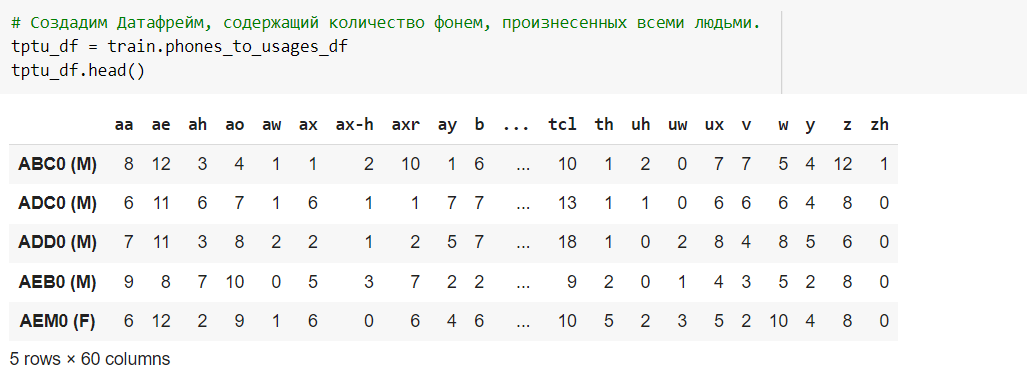


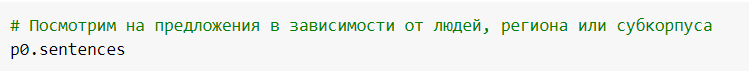
Рисунок 34 Пример кода с результатом выполнения.

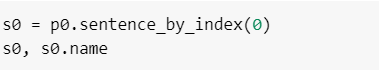


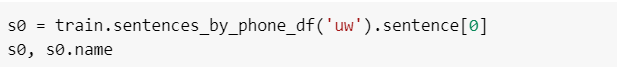
Рисунок 35 Пример кода с результатом выполнения.

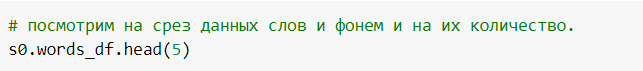
Изучим региональную составляющую по имеющемуся набору данных.



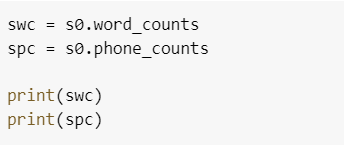












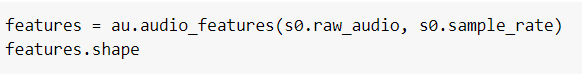
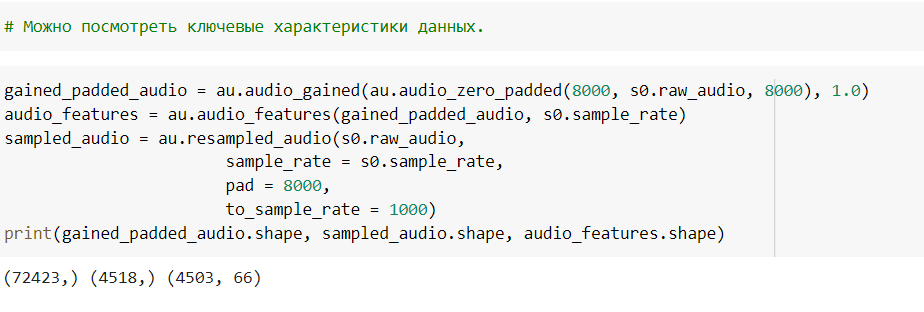


Рисунок 36 Примеры кода

Посмотрим ключевые характеристики данных.



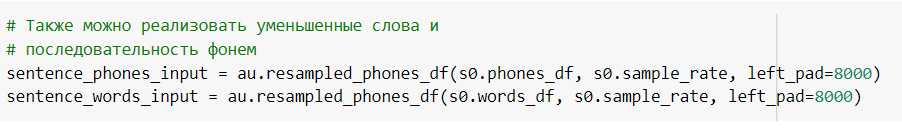


Рисунок 37 Примеры кода с результатом выполнения.

Визуализируем полученные данные с помощью графика.

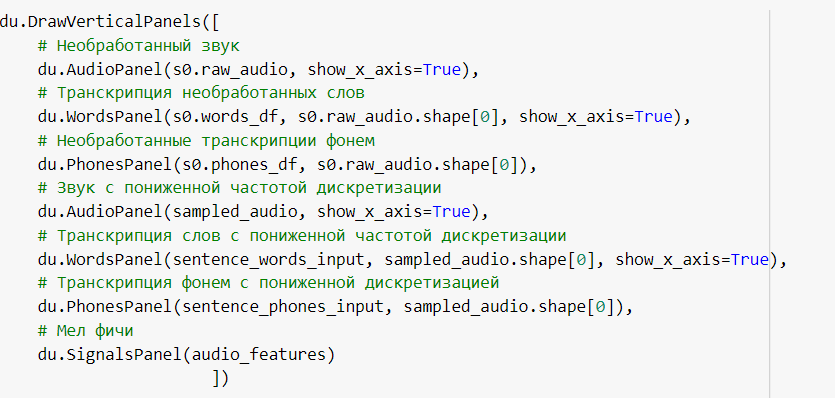


Рисунок 38 Пример кода



Рисунок 39 Пример кода

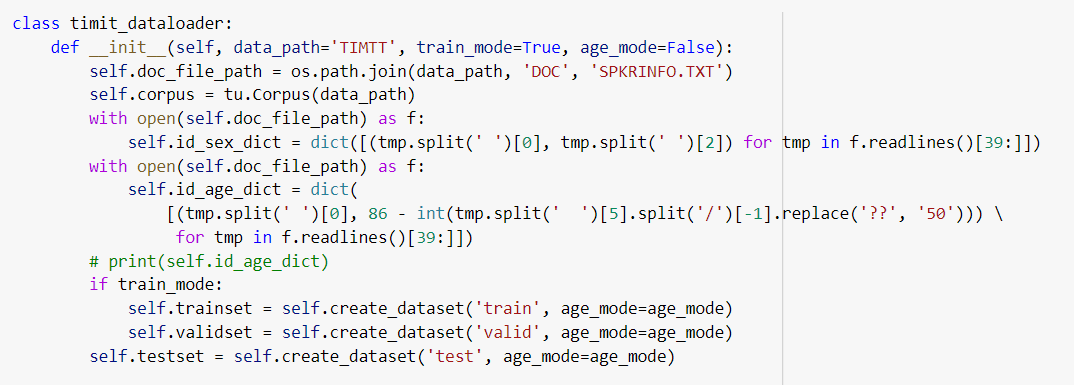
* 1. **Разработка и обучение модели**

Проведенный ранее анализ позволяет увидеть частоты слов и фонем, которые встречаются у людей. Эта информация нам пригодится для построения нейронной сети, которая будет определять кому принадлежит голос женщине или мужчине.

Обучим классификатор пола по голосу. Для этого зададим класс timit\_dataloader, который позволяет извлекать метки пола, возраста, идентификатор диктора, а также задает методы предобработки аудиосигнала.



Рисунок 40 Пример кода с результатом выполнения.





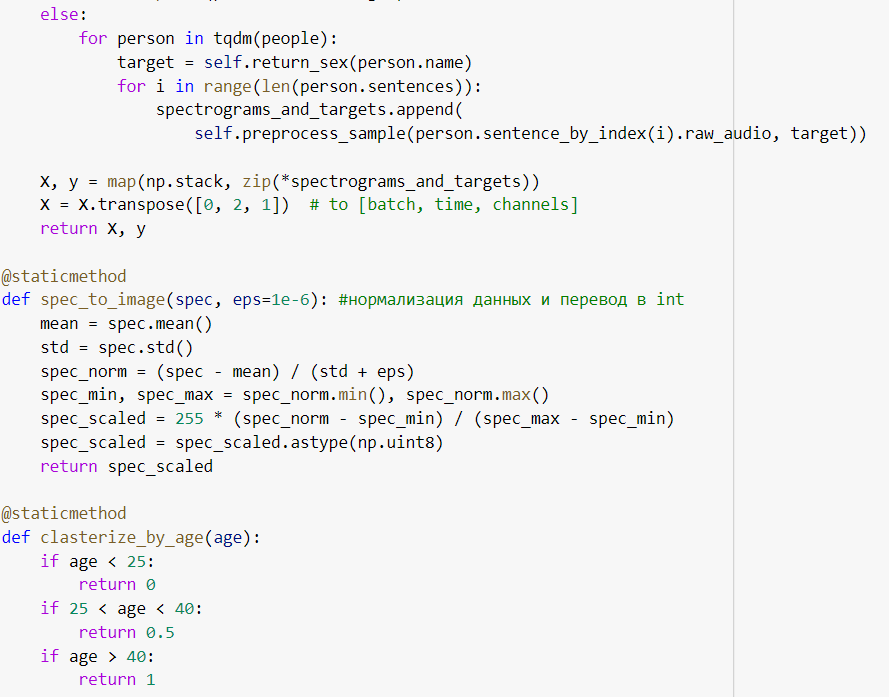


Рисунок 41 Пример кода.

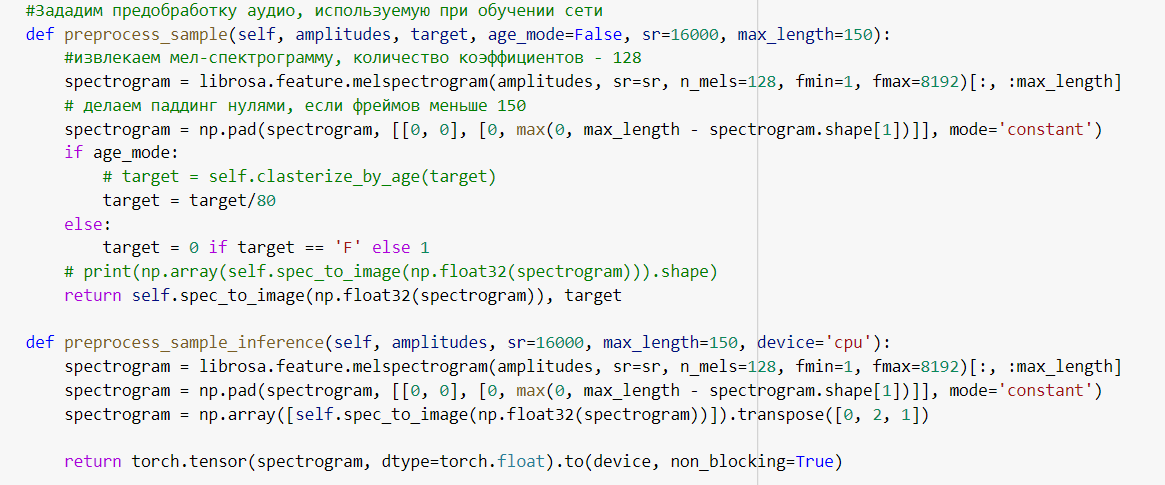


Рисунок 42 Пример кода.

Определим класс dataloader, который позволяет формировать батчи данных, подаваемые во время обучения нейросети.

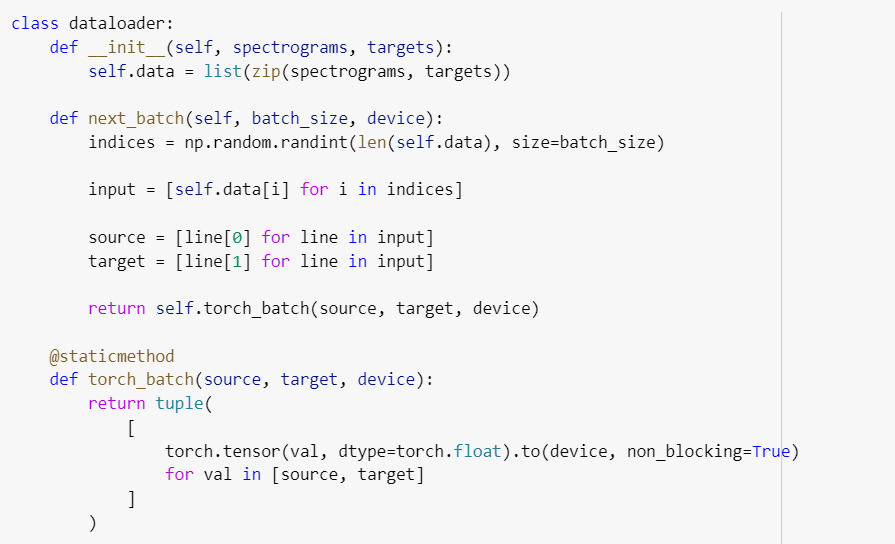


Рисунок 43 Пример кода.

Зададим архитектуру обучаемой сети.

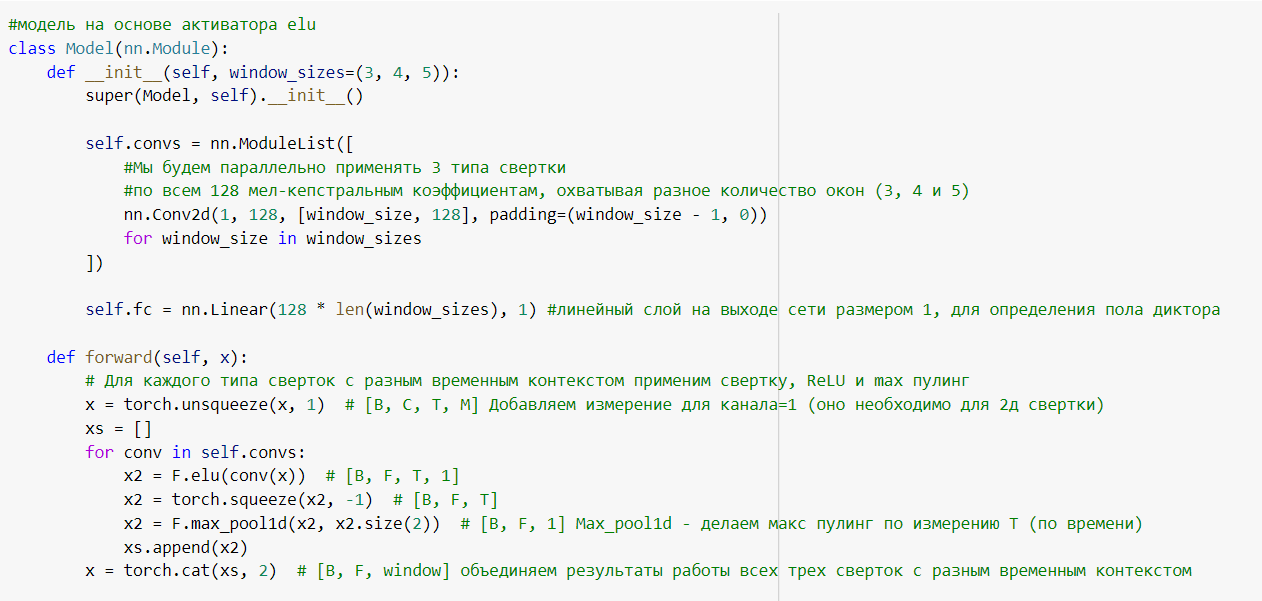




Рисунок 44 Пример кода. Модель на основе активатора elu.

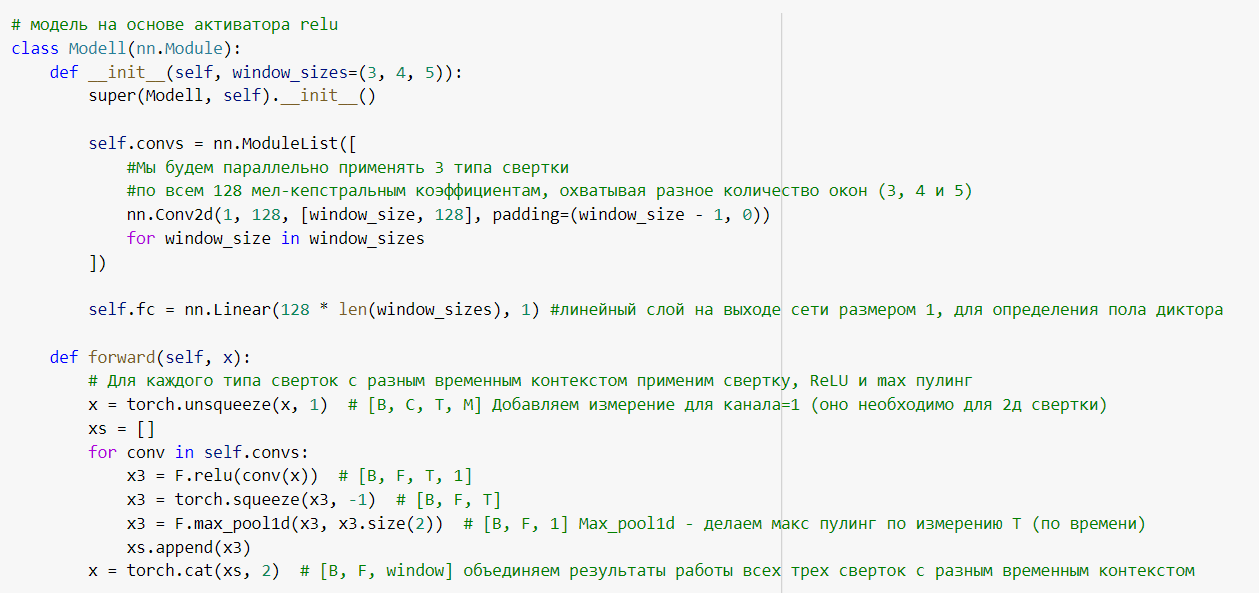




Рисунок 44 Пример кода. Модель на основе активатора relu.

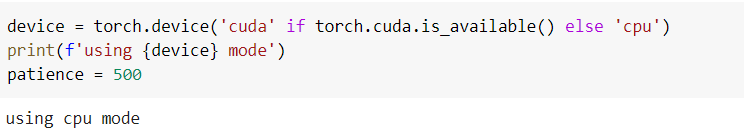
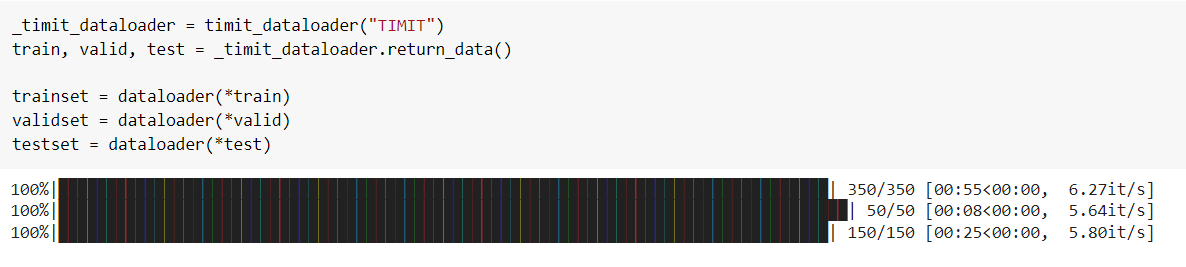
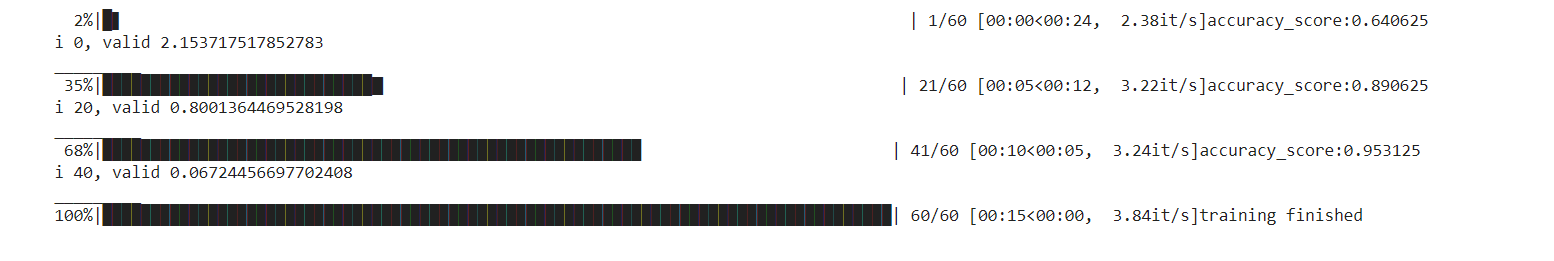


Рисунок 45 Пример кода.

Извлекаем мел-спектрограммы из всех аудиосигналов в наборе данных, подготовив обучающий, валидационный и тестовый наборы данных.



Обучаем нейронную сеть на основе активатора elu



Обучаем нейронную сеть на основе активатора relu

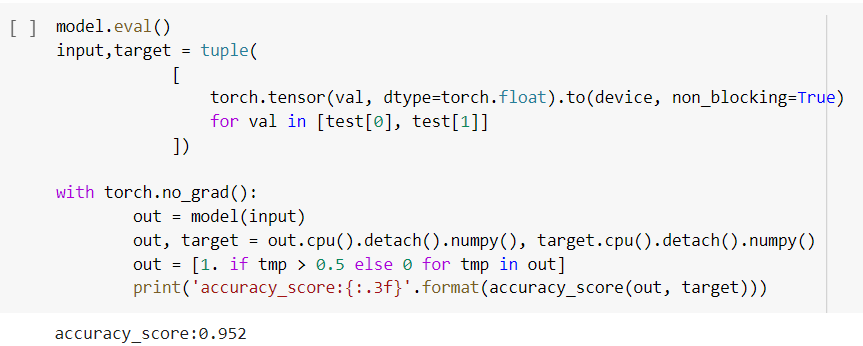


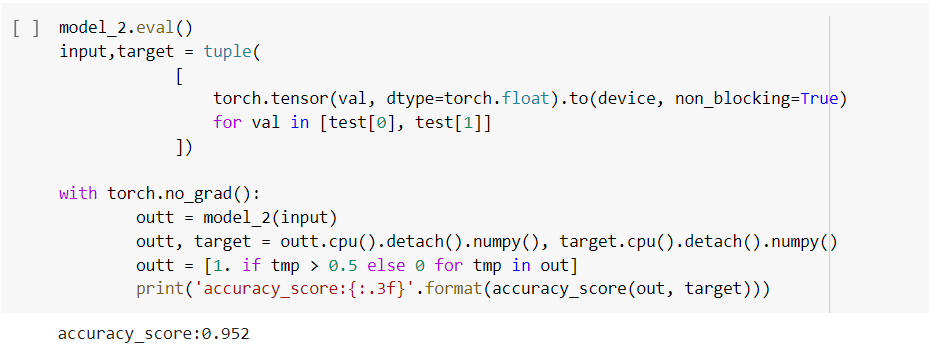
Рисунок 46 Пример кода с результатом выполнения.

* 1. **Тестирование модели**

После обучения моделей была проведем оценку точности этих моделей на обучающей и тестовых выборках.

Посчитаем метрики для обученных моделей. Для этого получим значения предсказанных вероятностей на тестовом наборе данных.



 Рисунок 47 Пример кода с результатом выполнения.

Зададим функцию, позволяющую рассчитать равную ошибку 1-го и 2-го рода (equal error rate, EER)

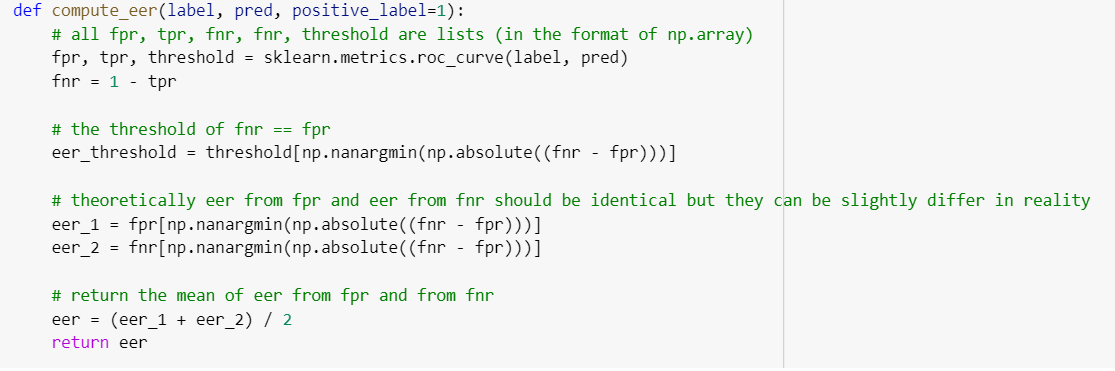


Рисунок 48 Пример кода.

Построим ROC-кривую и DET-кривую.

ROC-кривая или кривая ошибок – графическая характеристика качества бинарного классификатора, зависимость доли верных положительных классификаций от доли ложных положительных классификаций при варьировании порога решающего правила.

DET-кривая или кривая компромиссного определения ошибки показывает зависимость ошибки 1-го и 2-го рода бинарного классификатора от порога решающего правила

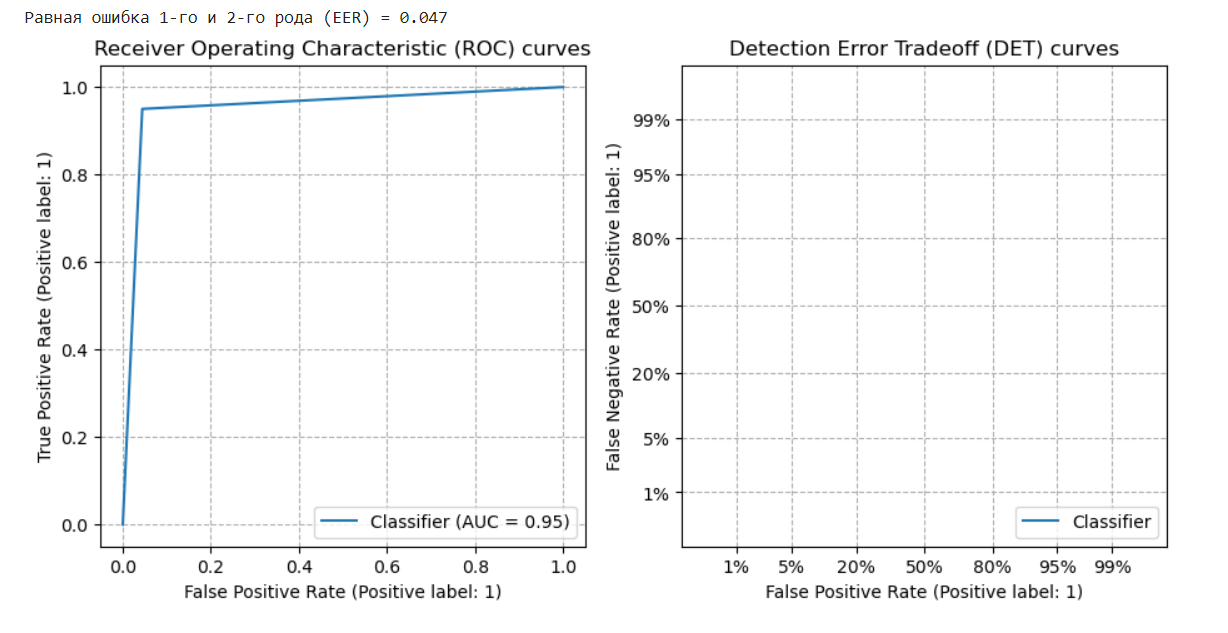
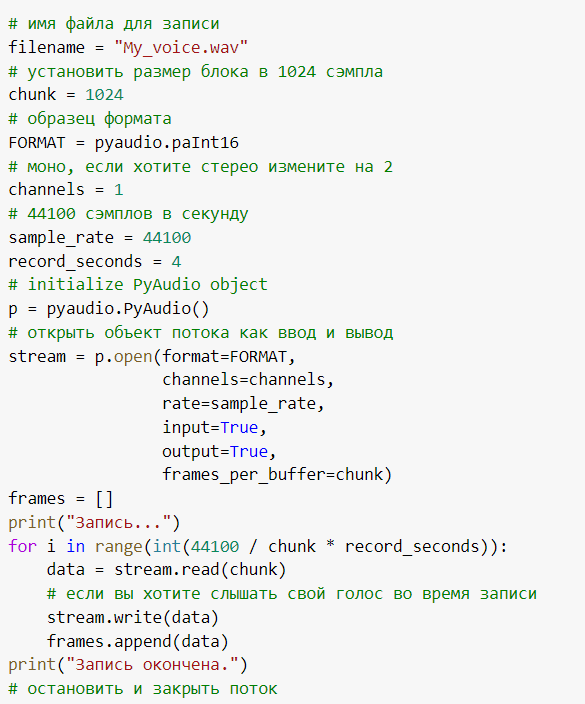


Рисунок 49 График

* 1. **Проверка работы модели на собственном голосе.**

Попробуем записать свой голос через микрофон и проверить, насколько хорошо работает модель с реальным голосом.



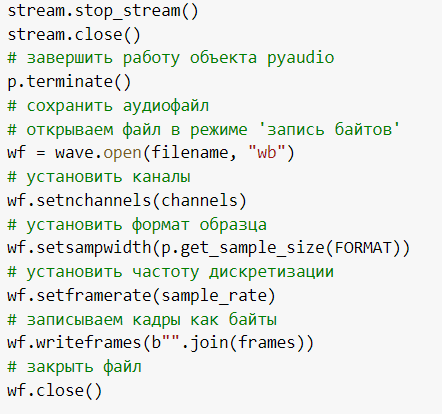
****

Рисунок 50 Пример кода



Рисунок 51 Пример кода

* 1. **Разработка приложения.**

Завершающим пунктом в нашей исследовательской работе была реализации приложения по модели, которую мы обучили в процессе работы. Приложение разрабатывалось на Flask, для этого мы установили соответствующую библиотеку Flask, а дальше задали route с методами POST и GET.

Затем на вход мы подали нашу модель обученной глубокой сверточной нейронной сети в задаче классификации аудиоданных и реализовали проверку при помощи подачи аудиофайла, который записывается пользователем. Этот файл прогоняется через нашу модель в результате мы получаем ответ кто говорит мужчина или женщина.

К сожалению, при разработке приложения мы столкнулись с рядом трудностей, а именно, было весьма сложно в условиях отключения российских каналов от всемирных серверов реализовать и испытать данное приложение, в связи с чем, в каких-то местах кода могут встречаться несовместимые для работы ошибки. Проверить правильность кода в связи с выше указанным не представлялось возможным.

Немного углубившись в начальные азы DataEngeneer мы пришли к выводу, что разработка приложения делится на Fronted и backend, реализовать backend получилось хорошо, а вот с реализацией fronted возникли сложности, в связи с тем, что в рамках курса мы очень кратко коснулись этой темы.

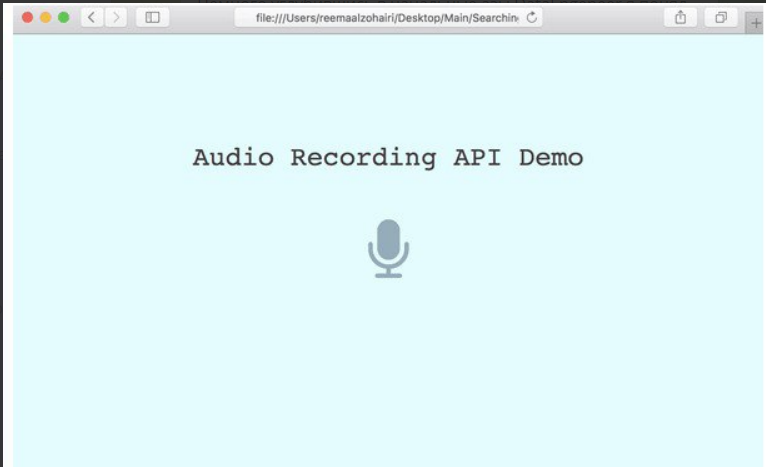


Рисунок 52 Скриншоты нашей работы над приложение



Рисунок 53 Скриншоты нашей работы над приложение

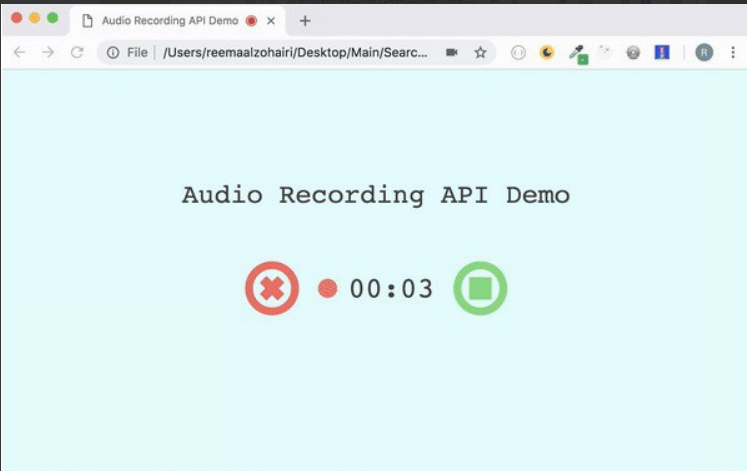


Рисунок 54 Скриншоты нашей работы над приложение

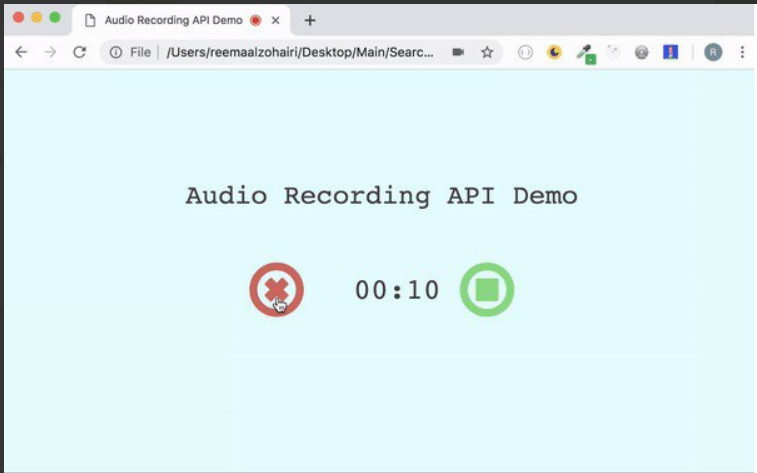


Рисунок 55 Скриншоты нашей работы над приложение

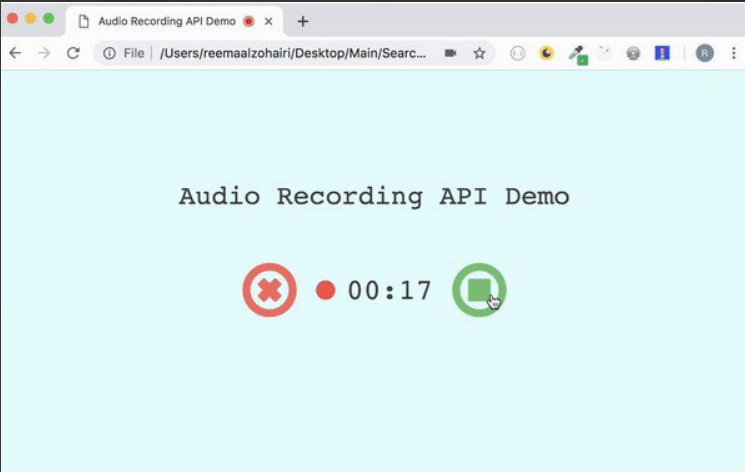


Рисунок 56 Скриншоты нашей работы над приложение



Рисунок 57 Пример кода

**2.6 Создание удалённого репозитория и загрузка проекта**

Репозиторий был создан на github.com по адресу: <https://github.com/Oleg-Evdokimov/KOMPOSIT>

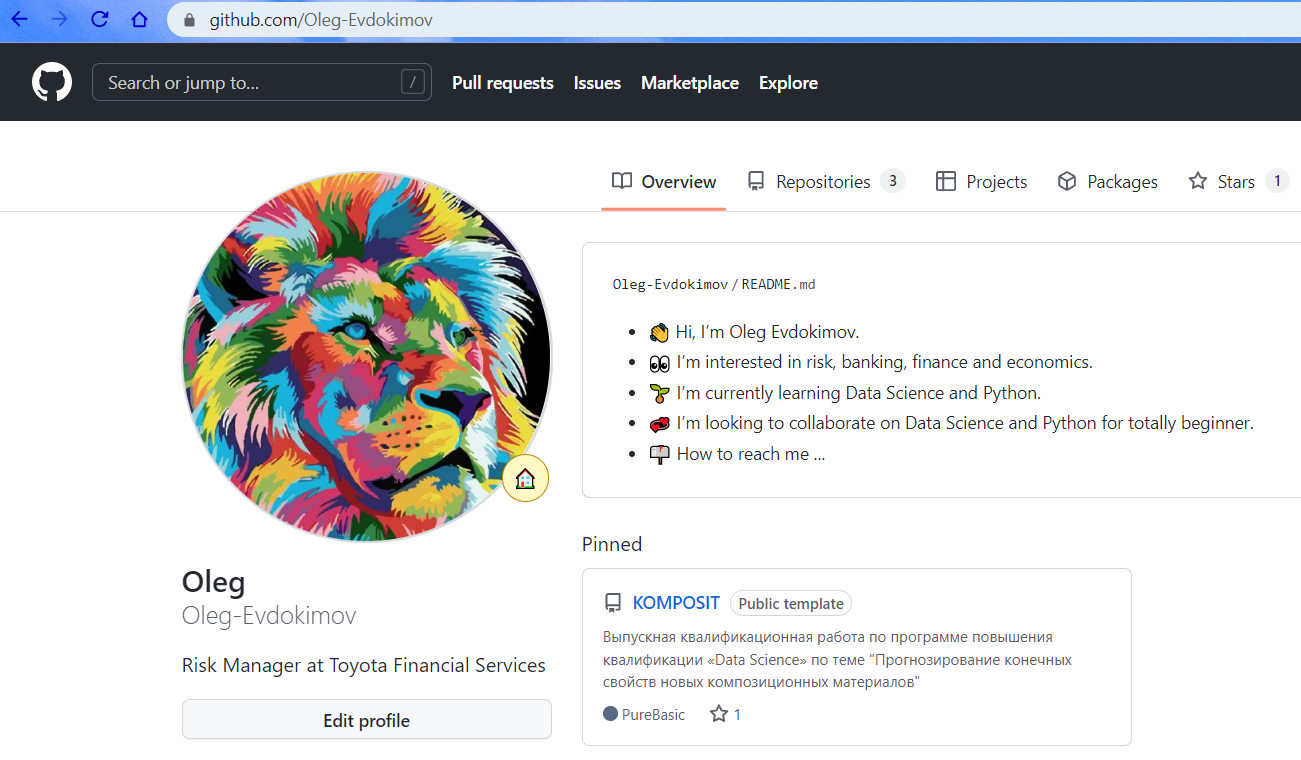


Рисунок 44 - часть страницы на github.com

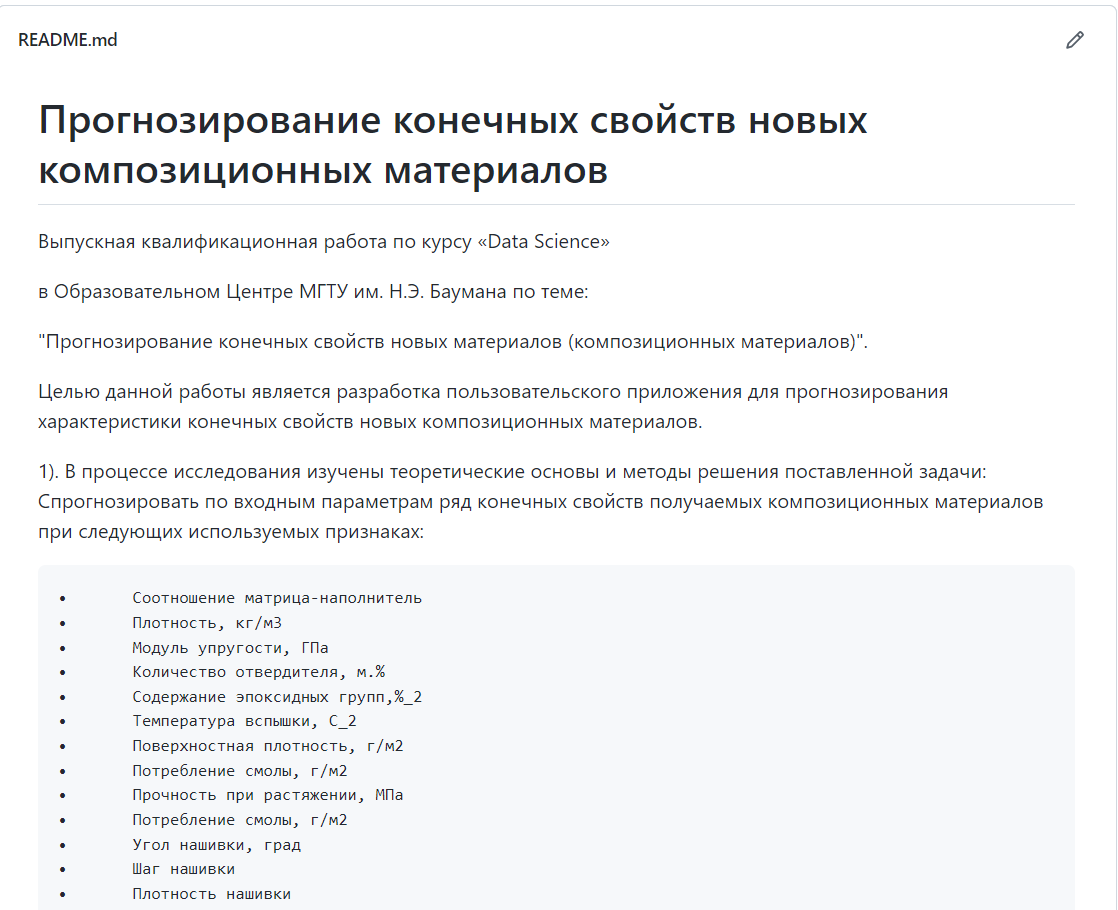


Рисунок 45 - часть созданного файла README

Ноутбук с решением и приложением так же можно найти по адресу: <https://colab.research.google.com/drive/1Pa9Y6gQq-vfE3cyAeAO7EhlQSHq2SO1P?authuser=2>

* 1. **Заключение**

Целью исследовательской работы являлась разработка модели, способной с высокой долей точности проводить классификацию аудиоданных по гендерному признаку

Данная цель была успешно достигнута. Мы успешно применили глубокую сверточную нейронную сеть и решили задачу бинарной классификации аудиоданных. Наша модель определяет гендер по голосу с точностью 95%.

Надеемся, что модель будет внедрена и будет применяться банками для снижения случаев мошенничества при подтверждении финансовых операций через контактный центр.

* 1. **Список используемой литературы и веб ресурсы**.

1. Alex Maszański. Метод k-ближайших соседей (k-nearest neighbour): – Режим доступа: <https://proglib.io/p/metod-k-blizhayshih-sosedey-k-nearest-neighbour-2021-07-19>. (дата обращения: 01.11.2022)
2. Andre Ye. 5 алгоритмов регрессии в машинном обучении, о которых вам следует знать: – Режим доступа: <https://habr.com/ru/company/vk/blog/513842/> (дата обращения: 01.11.2022).
3. Белов, Е. Б. Основы информационной безопасности : учебное пособие / Е. Б. Белов, В. П. Лось – М. : Горячая линия – Телеком, 2006. – 544 с.
4. Бизли Д. Python. Подробный справочник: учебное пособие. – Пер. с англ. – СПб.: Символ-Плюс, 2010. – 864 с., ил.
5. Гафаров, Ф.М., Галимянов А.Ф. Искусственные нейронные сети и приложения: учеб. пособие /Ф.М. Гафаров, Галимянов А.Ф. – Казань: Издательство Казанского университета, 2018. – 121 с.
6. Грас Д. Data Science. Наука о данных с нуля: Пер. с англ. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербурr, 2021. - 416 с.: ил.
7. Головко В.А. От многослойных персептронов к нейронным сетям глубокого доверия: парадигмы обучения и применения // Лекции по нейроинформатике. – М., 2017. – С. 47-84.
8. Devpractice Team. Python. Визуализация данных. Matplotlib. Seaborn. Mayavi. - devpractice.ru. 2020. - 412 с.: ил.
9. Джулли, Пал: Библиотека Keras - инструмент глубокого обучения / пер. с англ. А. А. Слинкин.- ДМК Пресс, 2017. – 249 с.
10. Документация по библиотеке keras: – Режим доступа: <https://keras.io/api/>. (дата обращения: 01.11.2022).
11. Документация по библиотеке librosa: – Режим доступа:  [https://librosa.org/](%20https://librosa.org/) . (дата обращения: 01.11.2022).
12. Документация по библиотеке matplotlib: – Режим доступа: <https://matplotlib.org/stable/users/index.html>. (дата обращения: 01.11.2022)
13. Документация по библиотеке numpy: – Режим доступа: <https://numpy.org/doc/1.22/user/index.html#user>. (дата обращения:01.11.2022).
14. Документация по библиотеке pandas: – Режим доступа: <https://pandas.pydata.org/docs/user_guide/index.html#user-guide>. (дата обращения: 01.11.2022).
15. Документация по библиотеке scikit-learn: – Режим доступа: <https://scikit-learn.org/stable/user_guide.html>. (дата обращения: 01.11.2022).
16. Документация по библиотеке seaborn: – Режим доступа: <https://seaborn.pydata.org/tutorial.html>. (дата обращения: 01.11.2022).
17. Документация по библиотеке PyTorch: – Режим доступа: https://pytorch.org/docs/stable/index.html (дата обращения: 01.11.2022).
18. Документация по языку программирования python: – Режим доступа:  <https://docs.python.org/3.8/index.html>. (дата обращения: 01.11.2022).
19. Иванов Д.А., Ситников А.И., Шляпин С.Д – Композиционные материалы: учебное пособие для вузов, 2019. 13 с.
20. Краткий обзор алгоритма машинного обучения Метод Опорных Векторов (SVM) – Режим доступа: <https://habr.com/ru/post/428503/> (дата обращения: 01.11.2022)
21. Курс «Нейронные сети и компьютерное зрение» – Режим доступа: <https://stepik.org/course/50352/> (дата обращения: 01.11.2022)
22. Миронов А.А. Машинное обучение часть I ст.9 – Режим доступа: <http://is.ifmo.ru/verification/machine-learning-mironov.pdf>. (дата обращения: 01.11.2022)
23. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение.Погружение в мир нейронных сетей. – СПб.: Питер, 2019. – 480 с.
24. Плас Дж. Вандер, Python для сложных задач: наука о данных и машинное обучение. Санкт-Петербург: Питер, 2018, 576 с.
25. Роббинс, Дженнифер. HTML5: карманный справочник, 5-е издание.: Пер. с англ. - М.: ООО «И.Д. Вильямс»: 2015. - 192 с.: ил.
26. Руководство по быстрому старту в flask: – Режим доступа: <https://flask-russian-docs.readthedocs.io/ru/latest/quickstart.html>. (дата обращения: 01.11.2022)
27. Силен Дэви, Мейсман Арно, Али Мохамед. Основы Data Science и Big Data. Python и наука о данных. – СПб.: Питер, 2017. – 336 с.: ил.
28. Скиена, Стивен С. С42 Наука о данных: учебный курс.: Пер. с англ. - СПб.: ООО "Диалектика", 2020. - 544 с.
29. Траск Э. Грокаем глубокое обучение. – СПб.: Питер, 2019. – 352с.
30. Тропченко А.Ю., Тропченко А.А. Методы сжатия изображений, аудиосигналов и видео. Учебное пособие по дисциплине «Теоретическая информатика» - Санкт-Петербург: , 2009. - 108 с.