Московский государственный университет имени М.В. Ломоносова

**Факультет вычислительной математики и кибернетики**



Программа профессиональной переподготовки

«**Разработчик компьютерных технологий**»

Итоговая аттестационная работа

**«Решение задачи диаризации методами машинного обучения»**

Работу выполнил

**Синёв Олег Сергеевич**

Научный руководитель:

Смирнов Илья Николаевич

**Москва**

**2023**

**Содержание**

[1. Введение 3](#_Toc138602522)

[1.1. Постановка задачи 3](#_Toc138602523)

[1.2. Актуальность задачи 5](#_Toc138602524)

[1.3. Обзор существующих решений 6](#_Toc138602525)

[1.3.1. VAD 6](#_Toc138602526)

[1.3.2. Классификация алгоритмов диаризации по промежуточным признакам 7](#_Toc138602527)

[1.3.3. Синусоидальная модель 10](#_Toc138602528)

[1.3.4. Кластеризация методом KMeans 11](#_Toc138602529)

[2. Исследование задачи 12](#_Toc138602530)

[3. Программная реализация 14](#_Toc138602531)

[1.1. Случай одного диктора 14](#_Toc138602532)

[3.1. Случай двух дикторов 17](#_Toc138602533)

[3.2. Случай трех дикторов 18](#_Toc138602534)

[3.3. Случай четырех дикторов 20](#_Toc138602535)

[4. Выводы 22](#_Toc138602536)

[5. Заключение 24](#_Toc138602537)

[6. Список литературы 25](#_Toc138602538)

# Введение

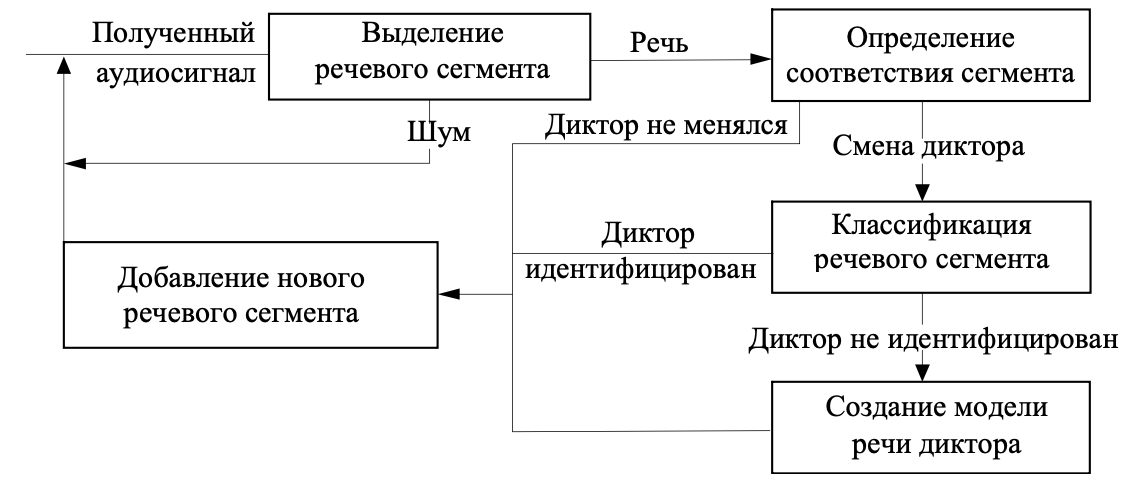
## Постановка задачи

Цель работы – разработать алгоритм, автоматически распознающий количество собеседников в аудиозаписи. Алгоритм должен осуществлять диаризацию (Speaker Diarization — SD, которая известна в англоязычной литературе как Who Spoke When). На вход программы подаётся аудиофайл, содержащий диалог двух или более человек, на выход выдается количество собеседников и их время, в которое они говорят.

Для достижения цели были поставлены следующие задачи:

* Удалить из аудиофайла тишину и посторонние шумы.
* Разбить аудиозапись на фреймы длительностью 1 секунда.
* Выделить из полученных фреймов признаки человеческой речи.
* Кластеризовать полученные признаки с помощью модели KMeans.
* Протестировать алгоритм на реальных данных.

Диаризация – это процесс разделения входящего аудиопотока на однородные сегменты в соответствии с принадлежностью аудиопотока тому или иному говорящему. Диаризация повышает качество текстов при автоматическом транскрибировании, а также может использоваться совместно с системой [распознавания речи](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%80%D0%B5%D1%87%D0%B8), значительно её улучшая. Диаризация используется для ответа на вопрос: «Кто сейчас говорит?». Схематично алгоритм выглядит следующим образом:



*Рис. 1. Структура типовой системы диаризации*

Первым делом программа очищает аудиозапись от тишины и посторонних шумов, не участвующих в определении признаков речи. Затем аудиозапись делится на фреймы для определения участков с активной речью. Данный сегмент будет использоваться для сравнения с соответствующей моделью речи собеседника. В случае если соответствие установлено, то выделенный фрейм добавляется к аудиофайлу данного диктора. В противном случае либо сравнение будет произведено до нахождения соответствующего говорящего, либо создастся модель нового собеседника и сегмент будет помечен, как принадлежащий последнему.

Часто для улучшения работы алгоритма программисты считаются с такими параметрами, как пол говорящего, его эмоции и качество сигнала.

Предварительное выделение участков, содержащих разного рода помехи, позволяет существенно уменьшить погрешность в работе систем диаризации. Даже различные методы определения речевой активности VAD, которые базируются на оценивании уровня сигнала его спектра, все же не решают всех проблем, которые возникают в задачах автоматического определения количества собеседников в аудиозаписи.



*Рис. 2. VAD алгоритм*

Именно поэтому пользуются популярностью методы, отвечающие за нормализацию многоканального сигнала, ведь существуют скрытые марковские модели, которые содержат не два состояния (речь/тишина), как обычно в VAD-методах, а 2K состояний, где K — количество дикторов. Главной особенностью данных моделей является обучаемость.

Большая часть передовых систем для построения алгоритмов определения количества собеседников пользуется моделями гауссовых смесей (Gaussian Mixture Models — GMMs). Они позволяют добиться высокой точности, хоть и требуют предварительного обучения моделей, что в то же время препятствует применимости метода в режиме реального времени.

Также применяется и метод параметрического представления звукового сигнала с использованием признаков MFCC (Mel Frequency Cepstral Coefficients).

## Актуальность задачи

Несмотря на уникальность голоса человека и существенное развитие информационных технологий, данная задача все еще остается актуальной и нет пока алгоритма, работающего на все 100%. Основными источниками ошибок могут являться следующие факторы:

* Внешняя среда (шум, посторонние звуки, частоты и т.д.);
* Особенности речи и дикции (дефекты речи, паузы, тембр и т.д.);
* Устройство (искажения микрофона, погрешности кодирования аудио и т.д.).

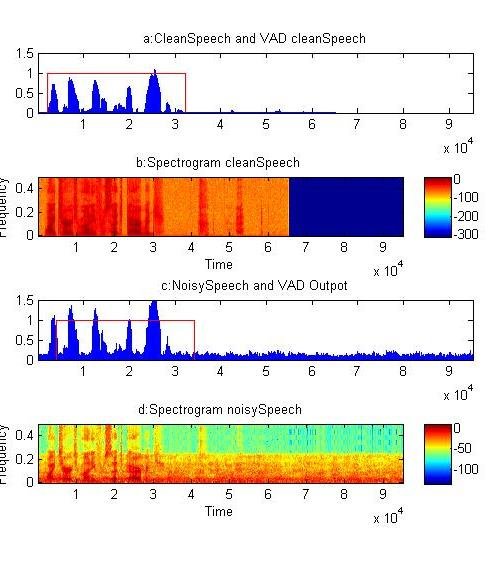
*Рис. 3. Управление умным домом Рис. 4. Голосовая идентификация*

Результаты данной работы востребованы в биометрическом поиске, голосовой идентификации, разграничении прав доступа к информации, создании субтитров к видеозаписям, системах умного дома, криминалистике, в качестве помощи к системам распознавании речи и т.д.

## Обзор существующих решений

### VAD

Задача VAD-метода заключается в определении участков с активной речью и выделении их в отдельный сегмент. Для такого разделения на фреймы применяется извлечение признаков, таких как Мел-Кепстральные коэффициенты, а также их дальнейшая кластеризация. Для нее могут использоваться такие модели, как метод KMeans, Глубинная нейронная сеть (DNN – deep neural network), Гауссова смесь распределений (GMM), Скрытые Марковские модели (CMM).



*Рис. 5. Пример работы VAD алгоритма*

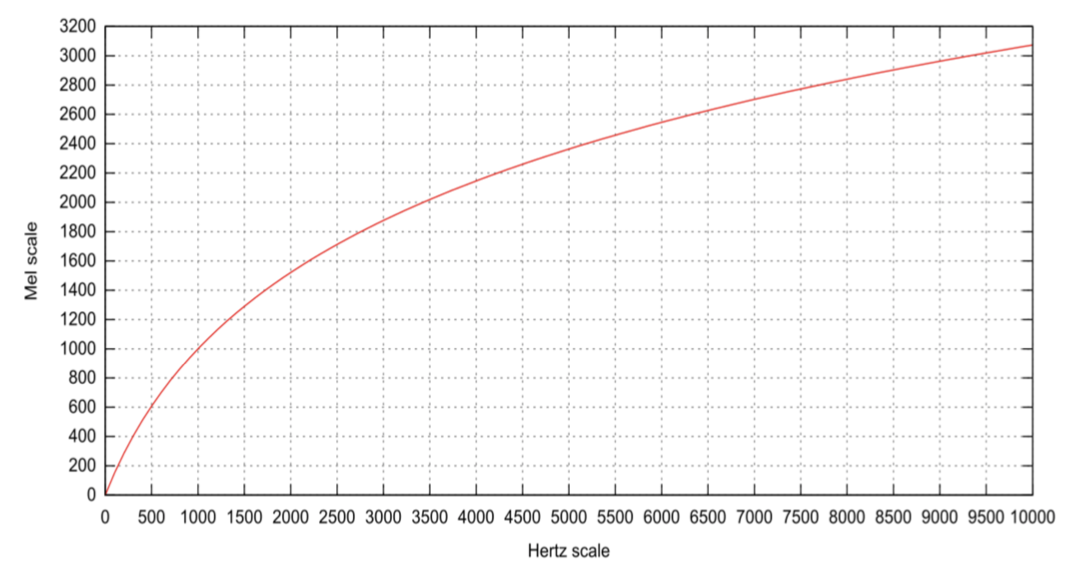
### Классификация алгоритмов диаризации по промежуточным признакам

Векторные характеристики представляют собой уникальный отпечаток голоса диктора, извлечённый с помощью GMM или DNN набором фильтров.

* *I­vector:* ихполучают через извлечение Мел­Кепстральных коэффициентов. Вычисление косинусного расстояния для двух i-векторов различных сегментов речи позволяет принимать решение об их принадлежности одному диктору. Как правило, при извлечении векторов аудиофайл делят на сегменты длиной от 20мс до 40мс (на фреймах такой длины изменения голоса несущественны), зачастую пересекая соседние сэмплы друг с другом от 50% до 75% своей длины;

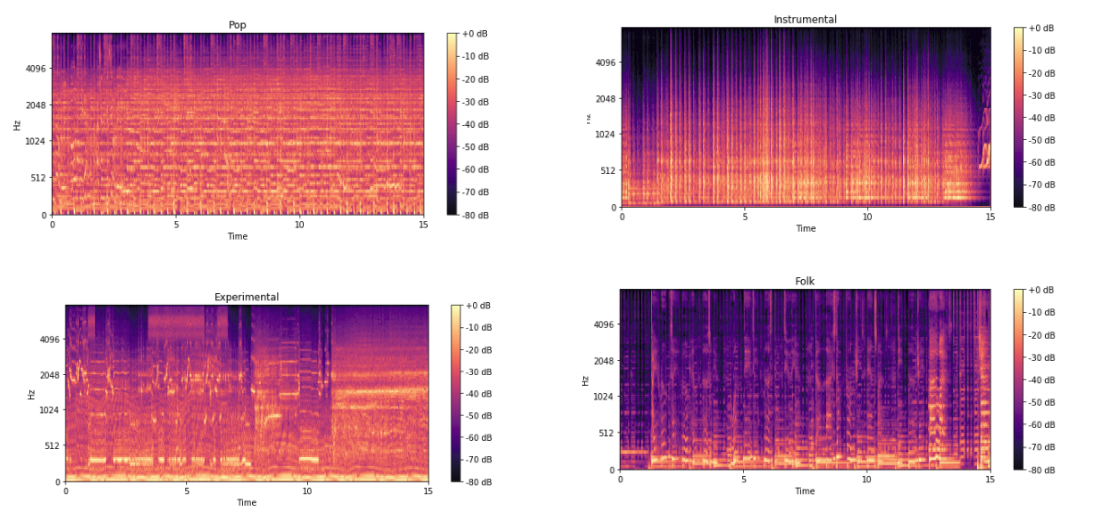


*Формула, описывающая отношение высоты чистого тона (мел) с фактической измеренной частотой (Гц).*



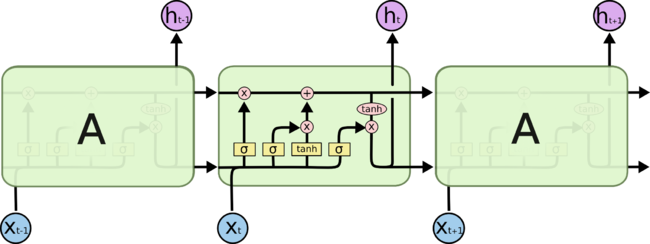
*Рис. 6. Шкала Мел*

* *X­vector:* такие векторы извлекают спектрограммы из аудиофайла и возвращают с помощью моделей глубокого обучения представление о сегменте. Несомненное превосходство метода в том, что нейронная сеть обучается специально для извлечения признаков, принадлежащих говорящему, что помогает улучшить результаты алгоритма диаризации;



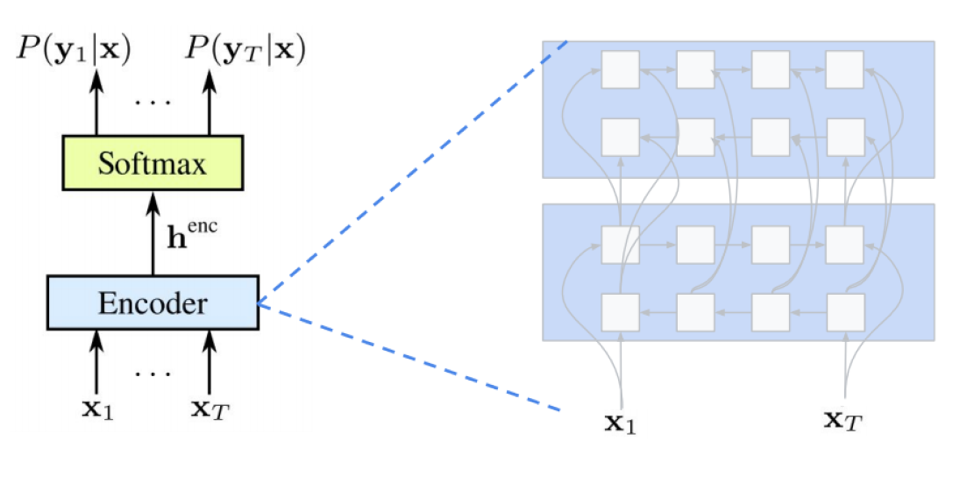
*Рис. 7. Примеры спектрограмм на музыкальных жанрах*

* *D-vector:* Данная характеристика извлекается с помощью обучения LSTM. В современном мире алгоритмы, использующие d-векторы для описания дикторов, показывают наилучшие результаты. Даже при небольшом количестве доступных примеров речи конкретного диктора эту характеристику можно успешно применять для его описания;



*Рис. 8. Пример работы LSTM*

* *end­to­end системы:* они не используют промежуточных представлений звука, лишь принимают на входе спектрограмму, а на выходе возвращают метку диктора.

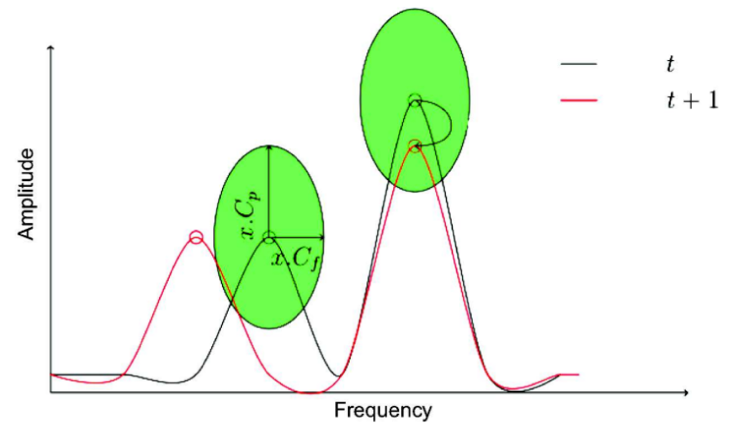


*Рис. 9. Пример работы end­to­end системы*

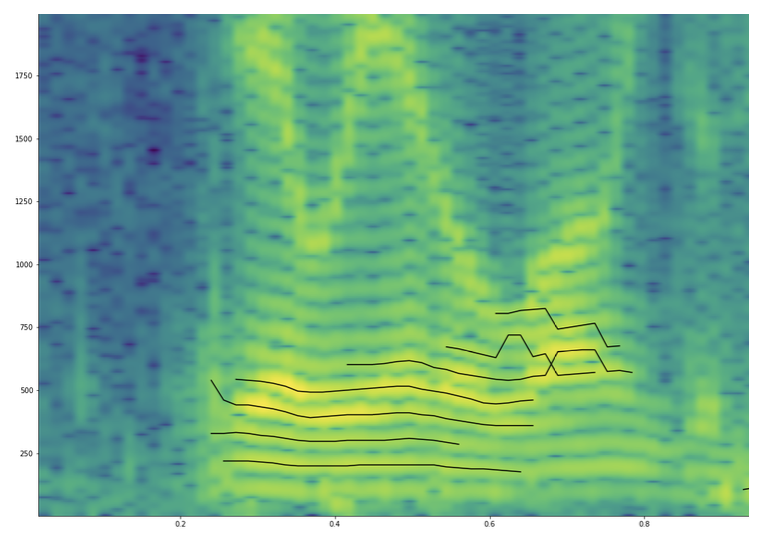
### Синусоидальная модель

Идея модели заключается в использовании алгоритма отслеживания треков в амплитудно­частотной области между фреймами.

* В каждом окне делается преобразование Фурье, на базе полученной амплитудно­частотной характеристики (АЧХ) берется n точек с наивысшей амплитудой;
* Полученные точки сопоставляются с точками из предыдущих и следующих окон.



*Рис. 10. Сопоставление пиков*

****

*Рис. 11. Синусоидальная модель*

### Кластеризация методом KMeans

Метод K-средних – это метод кластерного анализа, целью которого является разделение m наблюдений (из пространства ) на k кластеров, при этом каждое наблюдение относится к тому кластеру, к центру (центроиду) которого оно ближе всего.

В качестве меры близости используется Евклидово расстояние:

, где 

Метод k-средних используется для [кластеризации](https://wiki.loginom.ru/articles/clustering.html) данных на основе алгоритма разбиения векторного пространства на заранее определенное число [кластеров](https://wiki.loginom.ru/articles/cluster.html) k. Алгоритм представляет собой итерационную процедуру, в которой выполняются следующие шаги:

Выбирается число кластеров k.

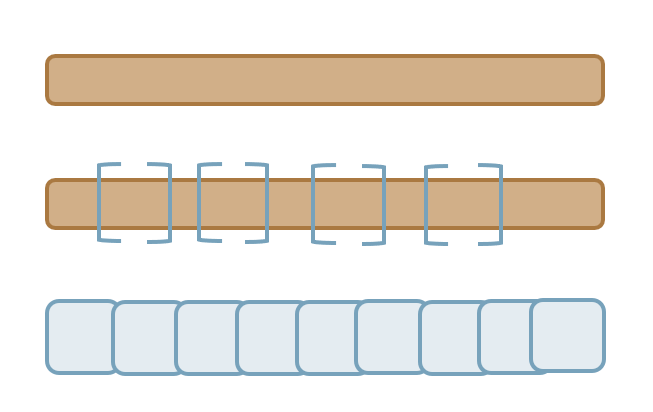
* Из исходного множества данных случайным образом выбираются k наблюдений, которые будут служить начальными центрами кластеров.
* Для каждого наблюдения исходного множества определяется ближайший к нему центр кластера (расстояния измеряются в метрике [Евклида](https://wiki.loginom.ru/articles/euclid-distance.html)). При этом записи, «притянутые» определенным центром, образуют начальные кластеры.
* Вычисляются центроиды — центры тяжести кластеров. Каждый центроид — это вектор, элементы которого представляют собой средние значения соответствующих признаков, вычисленные по всем записям кластера.
* Центр кластера смещается в его центроид, после чего центроид становится центром нового кластера.
* 3-й и 4-й шаги итеративно повторяются. Очевидно, что на каждой итерации происходит изменение границ кластеров и смещение их центров. В результате минимизируется расстояние между элементами внутри кластеров и увеличиваются междукластерные расстояния.

Остановка алгоритма производится тогда, когда границы кластеров и расположения центроидов не перестанут изменяться от итерации к итерации, т.е. на каждой итерации в каждом кластере будет оставаться один и тот же набор наблюдений. На практике алгоритм обычно находит набор стабильных кластеров за несколько десятков итераций.

При кластеризации признаков аудиофайла в качестве m наблюдений выступают Мел-Кепстральные коэффициенты фреймов. Каждый кластер, полученный методом KMeans, является отдельным говорящим.

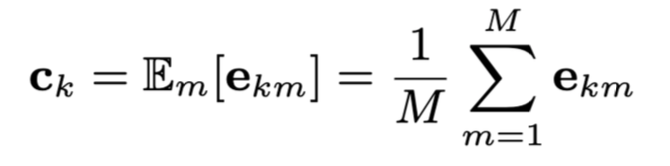
# Исследование задачи

Первый этап – извлечение характеристик, то есть данных, на основании которых можно получить информацию о голосе того или иного человека, идентифицировать его и использовать в алгоритме для решения поставленной задачи диаризации. Суть ее: разделить голоса отдельных дикторов, для этого в первую очередь необходимо заняться вопросом обнаружения участков активной речи. Предварительно аудиофайл разделяется на фреймы фиксированной длины с небольшим наложением;

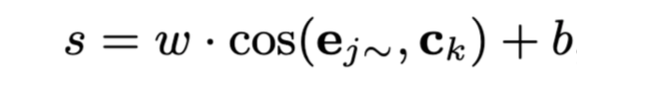


*Рис. 12. Деление аудиофайла на фреймы*

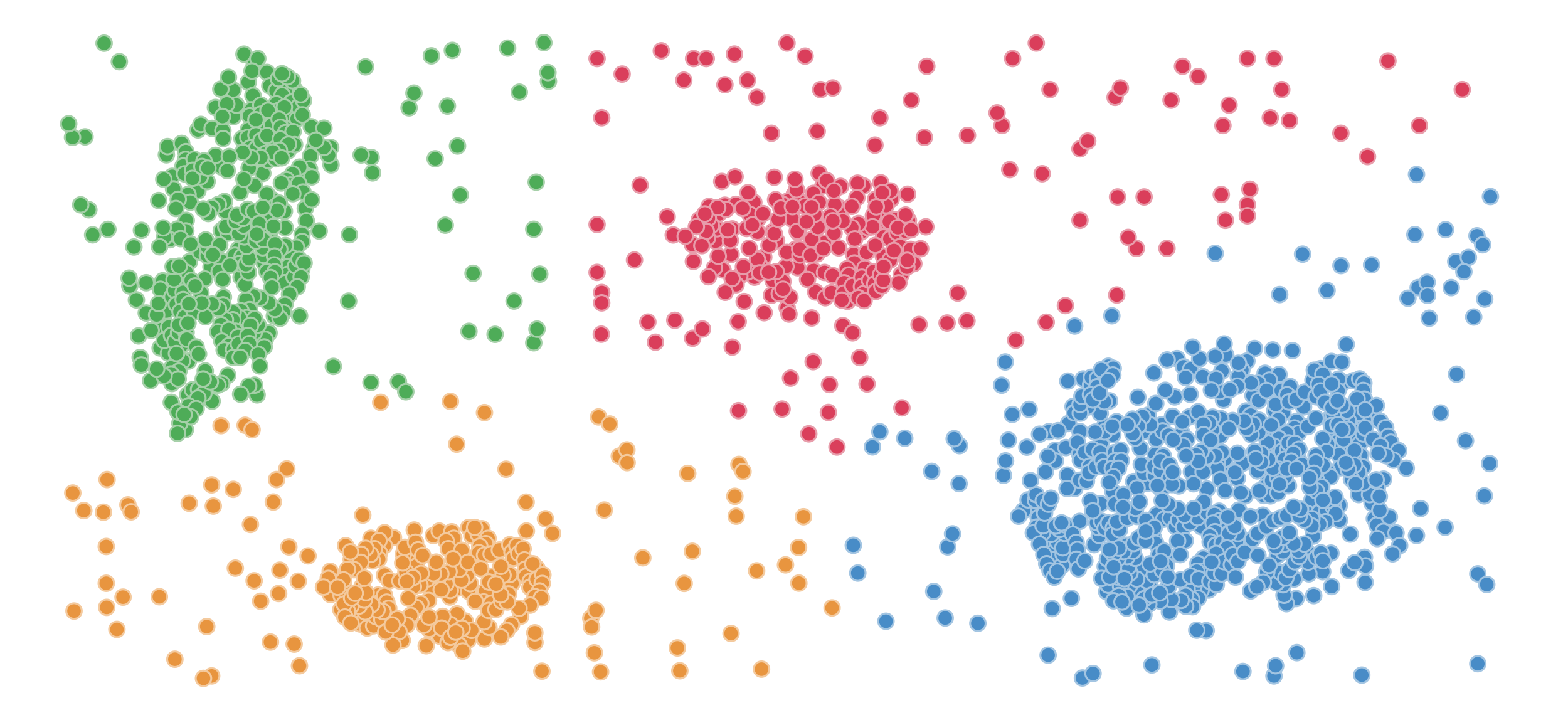
Данная характеристика извлекается с помощью обучения LSTM на фреймах, полученных из исходной аудиозаписи. К полученным значениям последнего слоя нейросети применяется L2 нормализация. Центроида полученного кортежа (где e – встраивание соответствующего фрейма, представляет собой уникальный голосовой отпечаток для M встраиваний;



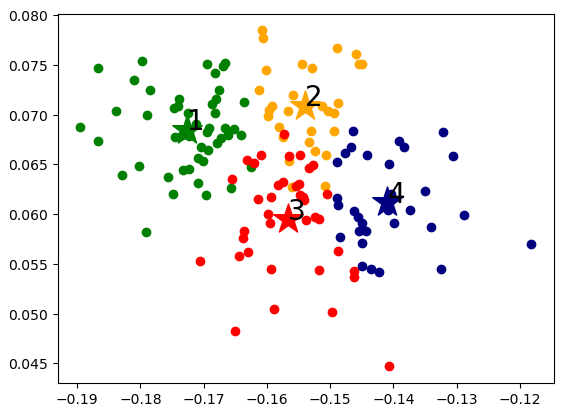
Для встраиваний вычисляется косинусное расстояние и производится сравнение с некоторым фиксированным пороговым значением;



Делается вывод о принадлежности речи тому или иному диктору и дальнейшая кластеризация фрагментов.

****

*Рис. 13. Кластеризация*



*Рис. 14. Визуализация кластеров*

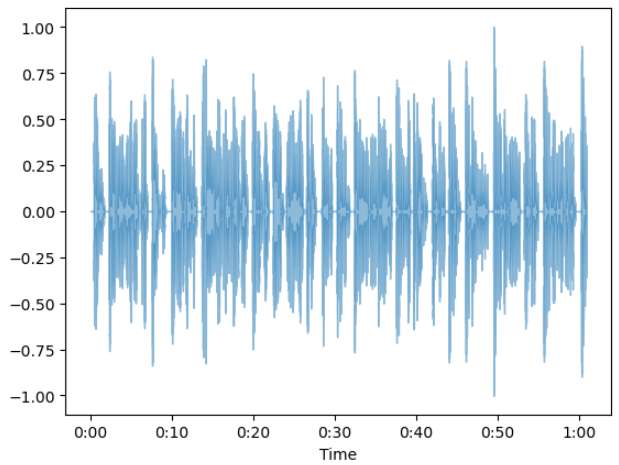
# Программная реализация

Программа реализована на языке Python. Для работы с аудиофайлами используется библиотека Librosa.

Librosa – библиотека Python для загрузки и анализа аудиофайлов. Она предоставляет строительные блоки для создания структур, которые помогают получать информацию о характеристиках аудиозаписи. С помощью метода librisa.feuters.mfcc можно получить признаки речи в виде массива данных. Этот массив используется для вычислений и анализа количества говорящих.

## Случай одного диктора

В аудиофайле содержится речь одного диктора. Длительность файла составляет одну минуту, в записи содержится тишина и есть посторонние шумы.

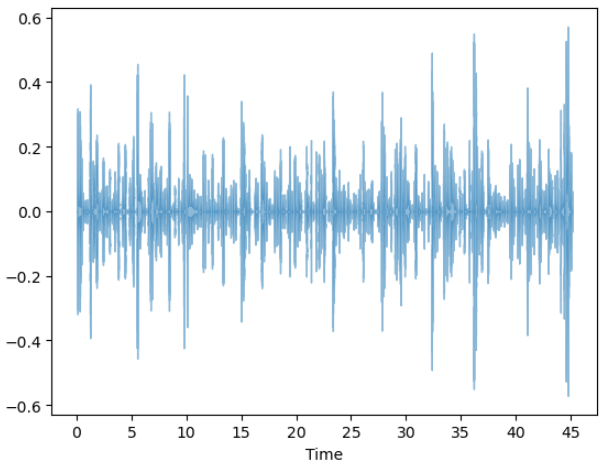
 *Рис. 15. Исходная аудиозапись*

С помощью метода librosa.effects.split производим первичную обработку аудиофайла с пороговым значением 20db, тем самым удаляем из файла всю тишину.



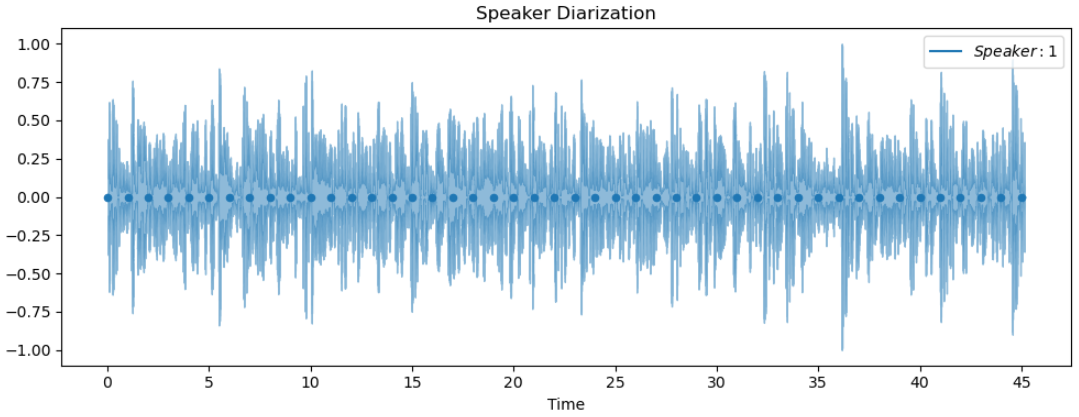
*Рис. 16. Аудиозапись без тишины*

Для снижения погрешности диаризации, с помощью библиотеки noisereduce уменьшаем амплитуду сигнала шумов на аудиозаписи.



*Рис. 17. Аудиозапись без шумов*

Далее алгоритм делит аудиозапись на фреймы длительностью 1 секунда с 12% наложением, threshold составляет 0.0099. Следующий этап – встраивание дикторов с помощью d-векторов для каждого сегмента и последующее вычисление косинусного расстояния, за которым следует кластеризация и определение количества собеседников.

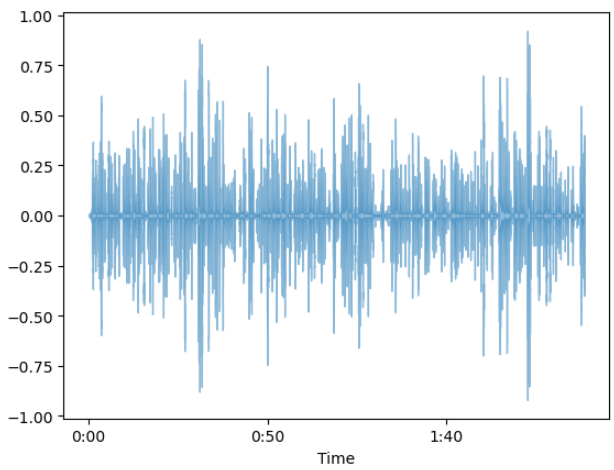


*Рис. 18. Результат диаризации*

На рисунке 18 мы видим вывод программы и его визуализацию. Светло-синим оттенком изображена исходная дорожка (оригинальная аудиозапись), синим показаны кусочки речи.

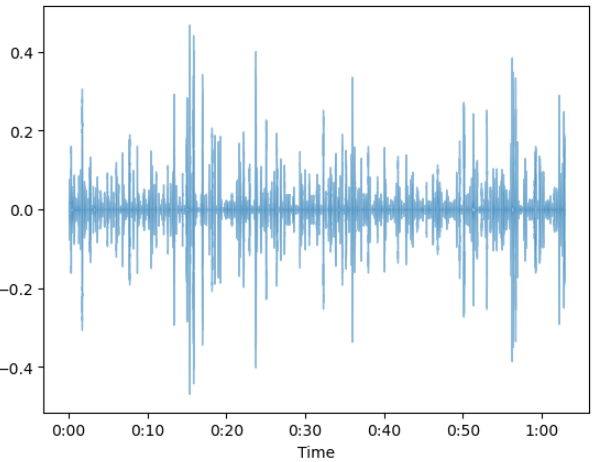
## Случай двух дикторов

а) Аудиофайл содержит диалог двух собеседников, качество записи среднее. По аудиозаписи понятно, что у первого диктора голос записан громче, а у второго есть небольшие дефекты в речи. Присутствуют несущественные шумы, помехи. Длительность составляет 2 минуты.



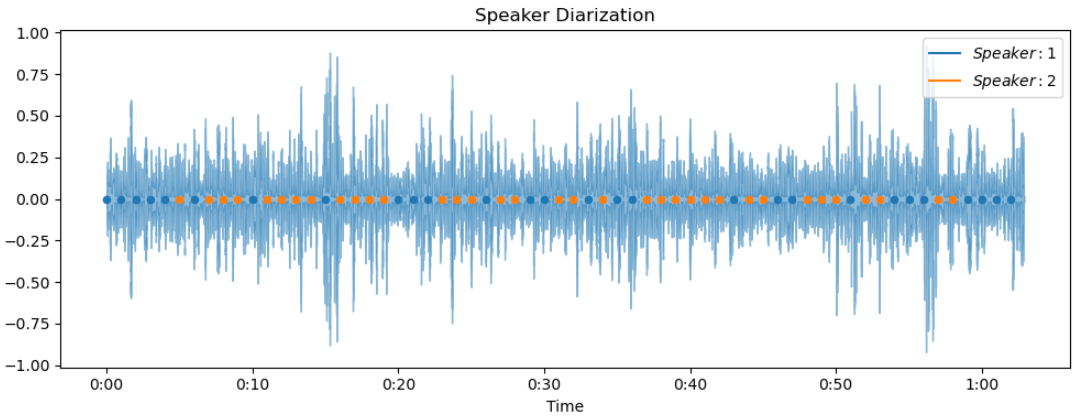
*Рис. 19. Исходная аудиозапись*

Для начала удаляем тишину и посторонние шумы.



*Рис. 20. Аудиозапись без тишины и шумов*

Для корректной обработки некоторые параметры были изменены: деление на фреймы длительностью 1 секунда, наложение 15%, threshold 0. 0009.

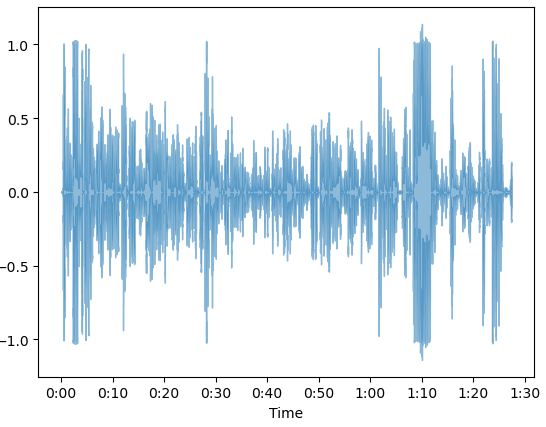


*Рис. 21. Результат диаризации*

На основании рисунка 21 можно сделать вывод, что алгоритм сработал верно. Количество собеседников двое, первый диктор говорит громче, и собеседники говорят по очереди.

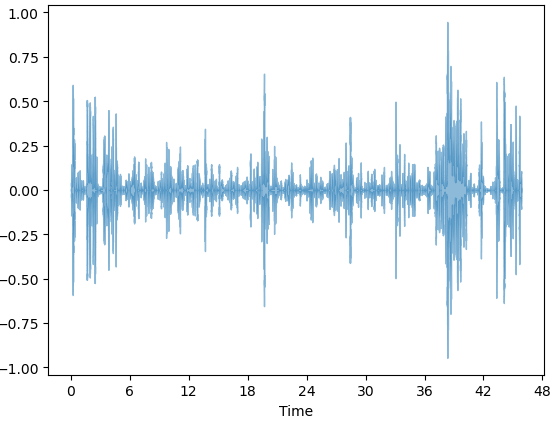
## Случай трех дикторов

б) Рассмотрим следующую аудиозапись, содержащую беседу трех человек. Шумов меньше, чем в предыдущем случае, однако собеседники на записи в некоторые моменты говорят одновременно. Длительность аудиозаписи около 1,5 минуты.



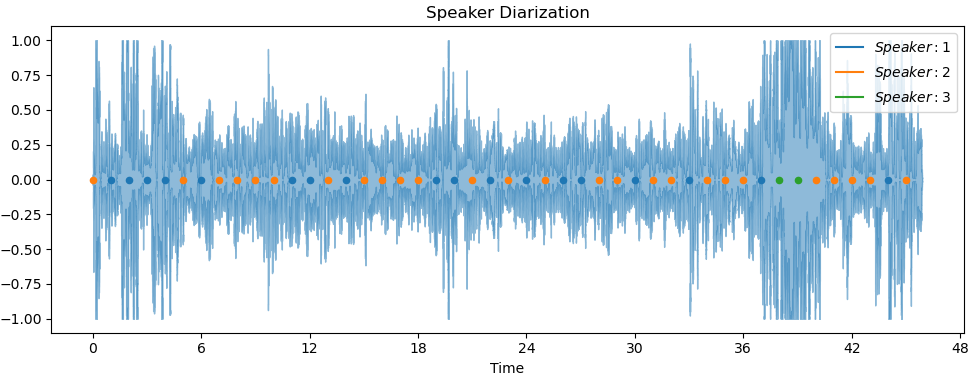
*Рис. 22. Исходная аудиозапись*

Удаляем тишину и посторонние шумы для корректного получения признаков.



*Рис. 23. Аудиозапись без тишины и шумов*

На рисунке 24 параметры сохранены, threshold остался прежним 0. 0009.

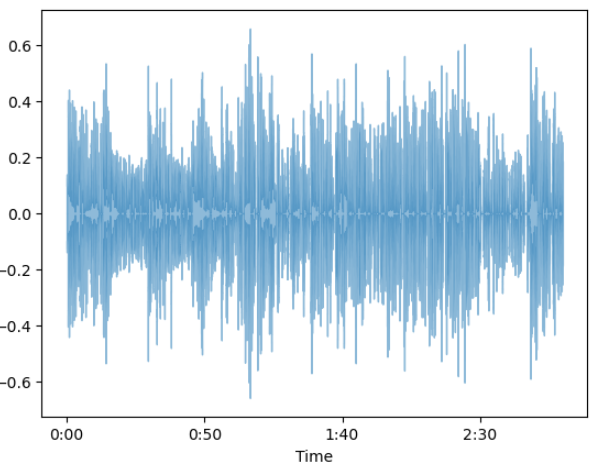


*Рис. 24. Результат диаризации*

Вывод: в силу сильной схожести голосов, показатель threshold напрямую влияет на результат работы программы, однако не дает верного ответа. Для корректного результата требуется использование больших характеристик и более глубокое изучение задачи диаризации.

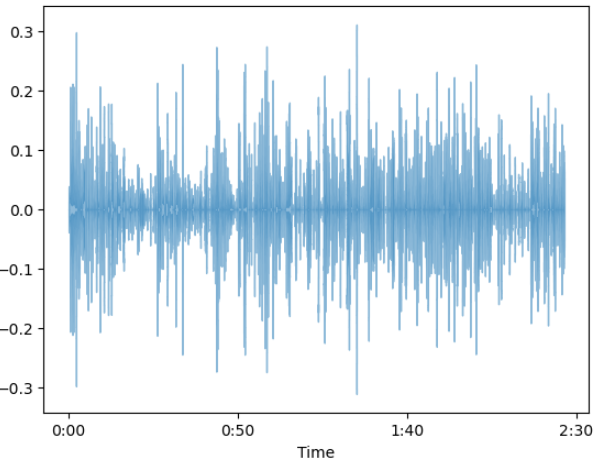
## Случай четырех дикторов

в) Аудиофайл содержит диалог четырёх собеседников. Речь практически беспрерывна, но есть паузы и посторонние шумы. Аудиозапись длится 3 минуты.



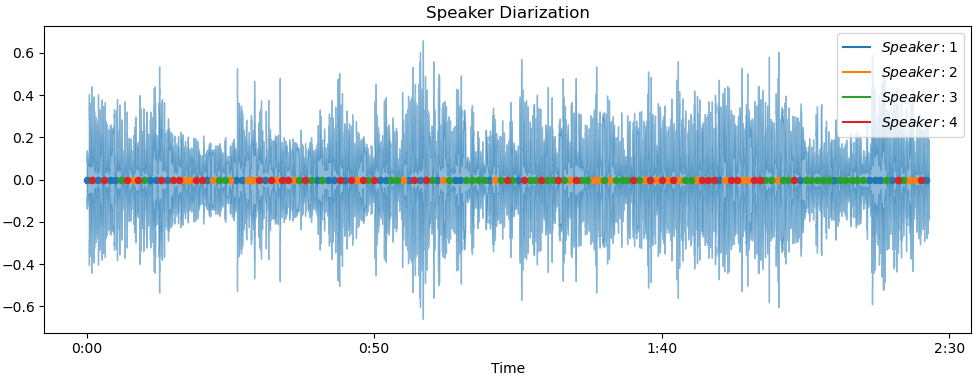
*Рис. 25. Исходная аудиозапись*

Удаляем тишину и посторонние шумы.



*Рис. 26. Аудиозапись без тишины и шумов*

Параметры: деление на фреймы длительностью 1 секунда, наложение составляет 15%, threshold 0.0099.



*Рис. 27. Результат диаризации*

Ответ получен верный, на записи зафиксированы все четыре голоса.

# Выводы

Таким образом, алгоритм определения количества собеседников действительно актуален, требует серьезных знаний не только в сфере программирования, но и в сфере музыки, а именно знаний о голосе, тембре, тонах и многом другом.

Проведенное мною исследование показывает, что задача разрешима даже в рамках курсовой работы, однако содержит в себе определенную погрешность, преодолеть которую возможно с увеличением характеристик голоса, более глубокого анализа и обучения нейронных систем.

Благодаря тому, что программа не привязана к тексту разговора, она может работать для любого иностранного языка, что является ее несомненным преимуществом. Однако алгоритм сильно зависим от устанавливаемых в коде параметров, таких как: процент наложения одного фрейма на другой, уровень threshold-а, длительность сегментов, на которые делится аудиофайл.

На примерах мы можем видеть, что программа зачастую игнорирует участки с речью, а шумы, напротив, включает в сегменты. Это дает серьезную погрешность и приводит к дальнейшим ошибкам в определении количества собеседников в аудиозаписи. Тем не менее данная проблема может быть разрешена посредством совершенствования алгоритма деления фреймов. Например, можно использовать не фиксированную величину разрезания на сегменты, а варьирующуюся в зависимости от частот или фигурирования в записи речи/шумов, а всяческие помехи незамедлительно удалять. Также программа дает менее качественные результаты для аудиозаписей с большей длительностью, а монолог одного диктора распознается лучше того, речь которого наименее представлена в аудиозаписи. Несмотря на это, алгоритм дает показатели, которые в целом соответствуют условиям поставленной задачи.

В дальнейшем развитии данной работы необходимо добавить характеристики, которые решали бы проблему идентификации дикторов, чьи голоса очень похожи и, возможно, принадлежат родственникам. Также можно усовершенствовать алгоритм шумоподавления и найти оптимальные значения параметров, которые давали бы удовлетворительные результаты для большинства тестируемых данных.

# Заключение

В рамках данной работы исследована задача диаризации, раскрыта ее актуальность, а также представлены различные подходы к ее решению и описаны значимые характеристики и методы, необходимые для эффективной реализации алгоритма автоматического определения количества собеседников в аудиозаписи.

Реализация программы на языке Python упрощается использованием готовых библиотек с открытым исходным кодом. Это сокращает время реализации программы.

Так как признаки говорящих основаны на значениях амплитуды аудиосигнала, программа может производить анализ говорящих на различных языках.

В ходе работы выявлено, что для улучшения качества диаризации необходимо минимизировать размер фрейма и увеличивать количество получаемых Мел-Кепстральных коэффициентов для одного фрейма.

Программная реализация раскрыта на тестах реальных данных, на основании анализа которых даны выводы о качестве работы кода и предложены пути улучшения и доработки алгоритма.

Конечный код программы выложен в общий доступ в сети Интернет:

<https://github.com/Minzdravv/FinalWork1>

# Список литературы

1. Б.М. Нутфуллин, Е.А. Ильюшин. Применение синусоидального моделирования речи к задаче диаризации звука. // International Journal of Open Information Technologies ISSN: 2307­8162 vol. 9, no.7, 2021
2. А.В. Судьенкова. Обзор методов извлечения акустических признаков речи в задаче распознавания диктора. // СБОРНИК НАУЧНЫХ ТРУДОВ НГТУ. – 2019. – No 3–4 (96). DOI: 10.17212/2307-6879-2019-3-4-139-164
3. А.Л. Ронжин, В.Ю. Будков. Анализ современных методов и систем диаризации дикторов. // ИЗВ. ВУЗОВ. ПРИБОРОСТРОЕНИЕ. 2012. Т. 55, No 11. УДК 004.896
4. П.А. Меньшаков, И.А. Мурашко. Методика голосовой идентификации на основе нейронных сетей. // ДОКЛАДЫ БГУИР 2017, No 4 (106). УДК 004.822:514
5. О.Ж. Мамырбаев, А.С. Кыдырбекова, Б.Ж. Жумажанов, Д.О. Оралбекова. Распознавание голоса с использованием Х-векторов. // ВЕСТНИК АЛМАТИНСКОГО УНИВЕРСИТЕТА ЭНЕРГЕТИКИ И СВЯЗИ, ISSN 1999-9801
6. Диаризация на основе модели GMM-UBM и алгоритма MAP adaptation. 14 сен 2018. https://habr.com/ru/articles/420515/
7. С.С. Астапов, Е.В. Шуранов, А.В. Лаврентьев, В.И. Кабаров. Автоматическое распознавание речи в условиях шума музыки на многоканальных записях с удаленного микрофона 2019. УДК 004.934
8. Ryan P. Adams, K-Means Clustering and Related Algorithms, COS 324 – Elements of Machine Learning Princeton University