ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ БЮДЖЕТНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**(ФИНАНСОВЫЙ УНИВЕРСИТЕТ)**

Департамент анализа данных

и машинного обучения

***Дисциплина: «Технологии анализа данных и машинного обучения»***

*Направление подготовки: «Прикладная математика и информатика»*

*Профиль: «Анализ данных и принятие решений в экономике и финансах»*

*Факультет информационных технологий и анализа больших данных*

*Форма обучения очная*

*Учебный 2022/2023 год, 6 семестр*

**Курсовая работа на тему:**

«Применение машинного обучения

в задаче жанровой классификации музыкальных композиций»

*Выполнил:*студент группы ПМ20-4  
Беленький М. А.

*Научный руководитель:*к.э.н., доцент Гринева Н.В.

**Москва 2023**

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc136476325)

[1. Рекомендательные системы и жанровая классификация 4](#_Toc136476326)

[2. Теория обработки аудиосигналов 6](#_Toc136476327)

[2.1. Дискретное преобразование Фурье 7](#_Toc136476328)

[2.2. Спектрограмма и оконное преобразование Фурье 8](#_Toc136476329)

[2.3. Мел-спектрограмма 11](#_Toc136476330)

[3. Извлечение признаков 12](#_Toc136476331)

[**3.1.** Спектральный центроид (Spectral Centroid) 13](#_Toc136476332)

[**3.2.** Спектральная полоса пропускания (Spectral Bandwidth) 14](#_Toc136476333)

[**3.3.** Спектральный спад (Spectral Rolloff) 15](#_Toc136476334)

[**3.4.** Спектральный поток (Spectral Flux) 16](#_Toc136476335)

[**3.5.** Частота пересечения нуля (Zero-Crossing rate) 16](#_Toc136476336)

[**3.6.** Низкоэнергетический уровень (Low Energy Feature) 17](#_Toc136476337)

[**3.7.** Мел-кепстральные коэффициенты (MFCC) 18](#_Toc136476338)

[**3.8.** Темп(tempo) 20](#_Toc136476339)

[4. Формирование наборов характеристик 21](#_Toc136476340)

[**4.1.** Используемый датасет 21](#_Toc136476341)

[**4.2.** Наборы признаков 22](#_Toc136476342)

[5. Техническая часть 25](#_Toc136476343)

[**5.1.** Теоретический обзор 25](#_Toc136476344)

[**5.2.** Перечень и реализация основных задач, результаты 29](#_Toc136476345)

[**5.3.** Система музыкальных рекомендаций 37](#_Toc136476346)

[Заключение 40](#_Toc136476347)

[Список использованных источников 41](#_Toc136476348)

[Приложение 42](#_Toc136476349)

**Введение**

Музыка – это самый популярный вид искусства, который исполняют и слушают миллиарды людей каждый день. Существует множество музыкальных жанров, таких как метал, поп, джаз, классика, регги, блюз, диско и хип-хоп.

В последние годы музыкальная индустрия все больше и больше смещается в сторону цифровой дистрибуции через стриминговые (потоковые) музыкальные сервисы и онлайн-магазины. Такая модель дистрибуции позволяет потребителям в любое время иметь возможность прослушивания музыки из большой музыкальной библиотеки, а музыкальным исполнителям легче монетизировать свои произведения. Примером мировых стриминговых сервисов могут быть Spotify, Apple Music, Amazon Music, YouTube Music и др., популярными российскими площадками являются Яндекс.Музыка, СберЗвук, VK Музыка, МТС Music и другие.

Потоковые сервисы привели к экспоненциальному увеличению объема мультимедийного контента, поэтому с этим возросла и необходимость структурирования музыки, эффективных инструментов для её организации и автоматической рекомендации. Ранее классификация жанров музыки, доступной в цифровом формате, выполнялась вручную. Такой способ классификации являлся сильно субъективным, он занимал много времени и труда, что становилось проблемой для больших объемов данных. От человека-классификатора требовалось знание различных жанров и стилей, что предельно трудоёмко при миллионах песен, существующих в современном мире.

В данной работе мы рассмотрим то, как звуковой сигнал обрабатывается в цифровой среде, выявим связь между преобразованием Фурье и спектрограммой, рассмотрим характеристики, которыми обладают аудиофайлы, также сравним несколько архитектур нейронных сетей, которые могли бы предсказывать жанр конкретной композиции, определим оптимальный для работы нейронной сети набор характеристик и на его основе построим базовую систему рекомендаций музыки.

Цель работы – получить классификатор, который с высокой точностью мог бы определять вероятность отношения композиции к одному из десяти жанров. Будут сравниваться четыре различных набора данных, с которыми предстоит работать каждой из трёх сравниваемых архитектур нейронной сети: сверточной нейронной сети, рекуррентной нейронной сети и многослойному перцептрону. Дополнительной задачей, не входящий в основную часть работы, является базовая система рекомендаций, которая на основе выбранного набора будет предлагать наиболее похожие композиции для конкретной песни. В работе представлены оригинальные иллюстрации.

1. **Рекомендательные системы и жанровая классификация**

Автоматическая рекомендация музыки:

* позволяет каждому пользователю получить рекомендации, учитывающие его личные предпочтения и историю прослушивания (например, на многих стриминговых платформах для пользователей формируются «плейлисты дня»);
* дает возможность пользователю открыть для себя новые жанры и новых исполнителей, соответствующих его вкусам;
* позволяет экономить время на поиске музыки, предлагая пользователю уже готовый список рекомендаций;
* способствует наиболее точной маркировке и упорядочиванию контента контент-провайдерами.

Рекомендательные системы подвергались глубокому изучению, однако проблема рекомендаций музыки осложняется огромным социальными и географическими факторами, влияющими на предпочтения слушателя, а также огромным разнообразием различных музыкальных стилей и жанров.

Музыкальные жанры – это один из способов организации и классификации музыкального контента, это категориальные ярлыки, созданные людьми для идентификации или характеристики стиля музыки. Музыкальный жанр является важным дескриптором, который широко используется для организации и управления большими цифровыми музыкальными базами данных и распространения музыки. Музыкальные жанры возникали в результате сложного взаимодействия между общественностью, маркетингом, историческими и культурными факторами. Определение жанра музыкального произведения является достаточно нетривиальной задачей из-за субъективности человеческого восприятия, из-за отсутствия стандартов классификации, а также из-за того, что музыкальные жанры трудно поддаются систематическому описанию.

Жанровая классификация музыки имеет ряд преимуществ и полезных аспектов, включая следующие:

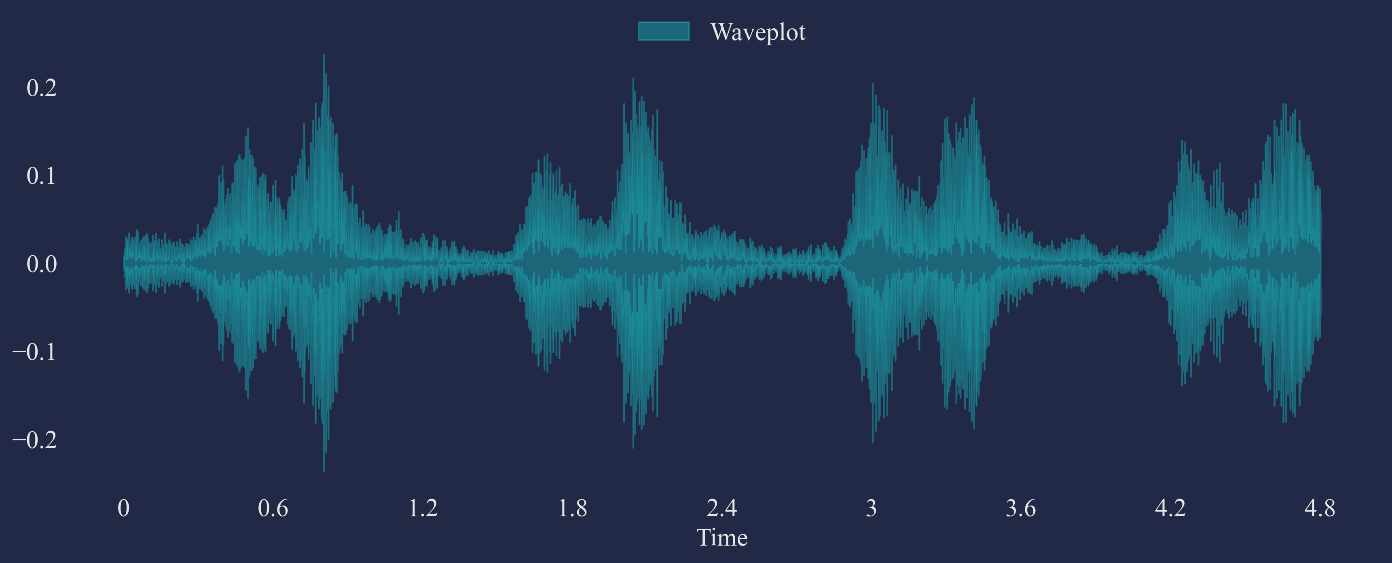
1. **Организация и систематизация**: Жанровая классификация музыки помогает систематизировать и организовать множество различных музыкальных произведений. Она позволяет классифицировать музыку по стилю, характеру и другим параметрам, что упрощает поиск и категоризацию музыкальных произведений.
2. **Легкость восприятия**: Жанровая классификация музыки помогает слушателям быстрее и легче понимать и оценивать музыку. Она позволяет слушателям лучше понимать музыкальные жанры, их стили и характеристики, что улучшает восприятие и оценку музыки.
3. **Социальный аспект**: Жанровая классификация музыки может использоваться для культурной и социальной классификации. Музыкальные жанры часто связаны с определенными культурными или социальными группами, что позволяет использовать их в контексте социальных и культурных исследований.
4. **Продвижение музыки**: Жанровая классификация музыки может использоваться в музыкальной индустрии для продвижения и продажи музыкальных произведений. Жанровая классификация позволяет определить целевую аудиторию и направить музыкальные произведения в нужное русло, что может помочь в продвижении музыкальных работ.
5. **Академический аспект**: Жанровая классификация музыки может использоваться в музыкальном образовании и исследованиях. Она позволяет студентам и исследователям понимать музыку, ее историю и развитие, и проводить анализ музыкальных произведений на основе жанровых характеристик.

Существуют критерии восприятия, связанные с текстурой, инструментовкой и ритмической структурой музыки, которые могут быть использованы для характеристики конкретного жанра.

1. **Теория обработки аудиосигналов**

Сигнал – это изменение определенной физической величины с течением времени. Аудиосигналом называется сложный сигнал, который состоит из нескольких одночастотных звуковых волн, совместно распространяемых в среде; для аудиосигнала изменяемая величина – это давление воздуха. В природе все сигналы являются аналоговыми, то есть непрерывными сигналами, которые могут принимать любое значение в определенном диапазоне и передаваться в виде непрерывной волны. Для работы сигналов (в частности аудиосигналов) в цифровых системах связи необходимо преобразовать аналоговый сигнал в цифровой. Сначала аналоговый сигнал подвергается дискретизации – разбиению на отдельные отрезки времени. Затем каждый отрезок амплитуды сигнала преобразуется в цифровое значение, которое записывается в память компьютера или другого цифрового устройства.

Осциллограмма – оцифрованная версия сигнала, которая отражает давление звуковой волны на мембрану микрофона, которая представляет собой огибающую амплитуды сигнала в зависимости от времени. На рисунке 1 представлена график осциллограммы, он может быть полезен для наглядной визуализации сигнала, его формы и частоты колебаний.

Рис. 1. Осциллограмма.

* 1. **Дискретное преобразование Фурье**

Преобразование Фурье – это математическая формула, которая позволяет нам разложить сигнал на составляющие его частоты и отображает амплитуду каждой частоты, присутствующей в сигнале. Другими словами, суть преобразования состоит в переходе от амплитудно-временного представления сигнала к амплитудам и фазам, зависящим от частоты. На рисунке 2 схематично изображен смысл преобразования Фурье, а ниже представлена формула:

где – непрерывная функция, – преобразование Фурье, – частота.

Изображение выглядит как диаграмма, линия, Шрифт, График

Автоматически созданное описаниеРис. 2. Преобразование Фурье.

Поскольку в данной работе мы имеем дело с ограниченными по времени, дискретными сигналами, применяется дискретное преобразование Фурье (ДПФ), математический метод, который преобразует последовательность дискретных данных из временной в частотную область. Формула дискретного преобразования Фурье выглядит, как:

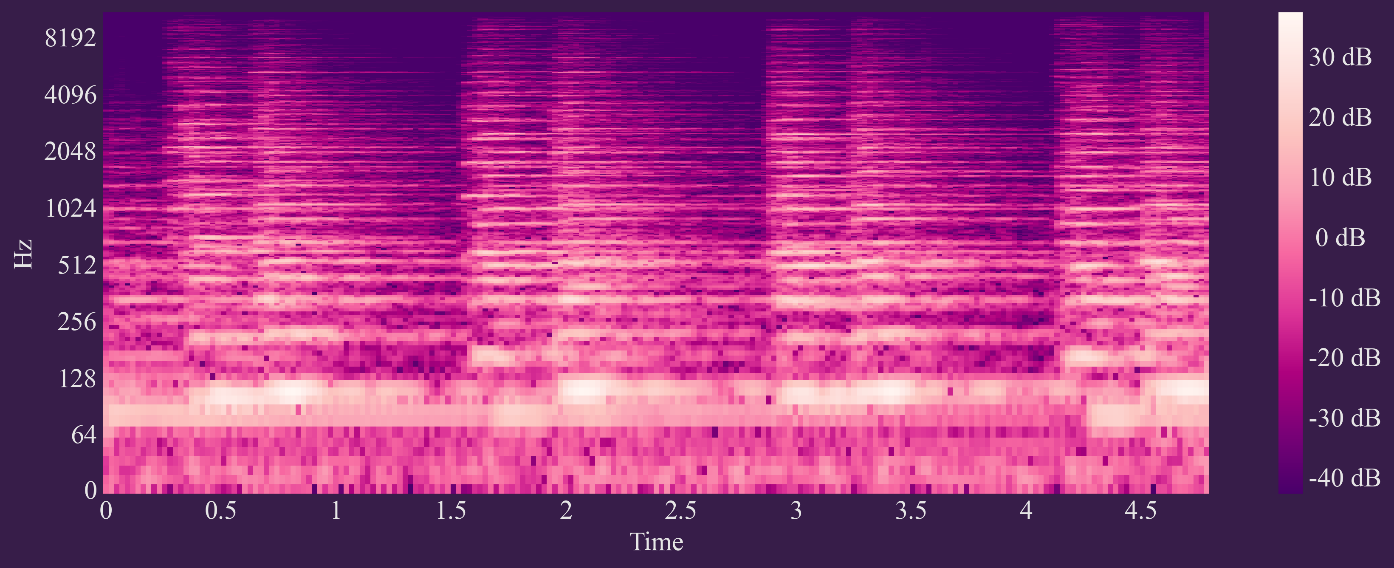
где – дискретный сигнал, – его преобразование Фурье, – количество отсчетов в сигнале, – номер частотной компоненты.

Спектр дискретного сигнала является результатом дискретного преобразования Фурье; это набор частот и их амплитуд, которые объединяются вместе для получения сигнала. Так как спектр изменяется со временем, изменяются и составляющие его частоты. Он может быть представлен суммой конечного числа гармоник.

* 1. **Спектрограмма и оконное преобразование Фурье**

Визуальный способ представления спектра различных частот, присутствующих в форме волны, называется *спектрограммой*. По оси *x* отображается время, по оси *y* – частота в герцах (Гц). Так как громкость сама по себе является не линейной, а логарифмической функцией, ось *y* обычно преобразуется из частот в децибелы (дБ). На рисунке 3 представлена спектрограмма, где на вертикальной оси показаны частоты (от 0 до 213 кГц), а на горизонтальной – время в секундах.

Спектрограмма изображается в виде тепловой карты, где разные цвета используются для обозначения амплитуды или силы каждой частоты. Чем ярче цвет, тем выше энергия сигнала. Каждый вертикальный «срез» спектрограммы представляет собой спектр сигнала в конкретный момент времени и показывает, как распределяется мощность сигнала на каждой частоте, присутствующей в сигнале.

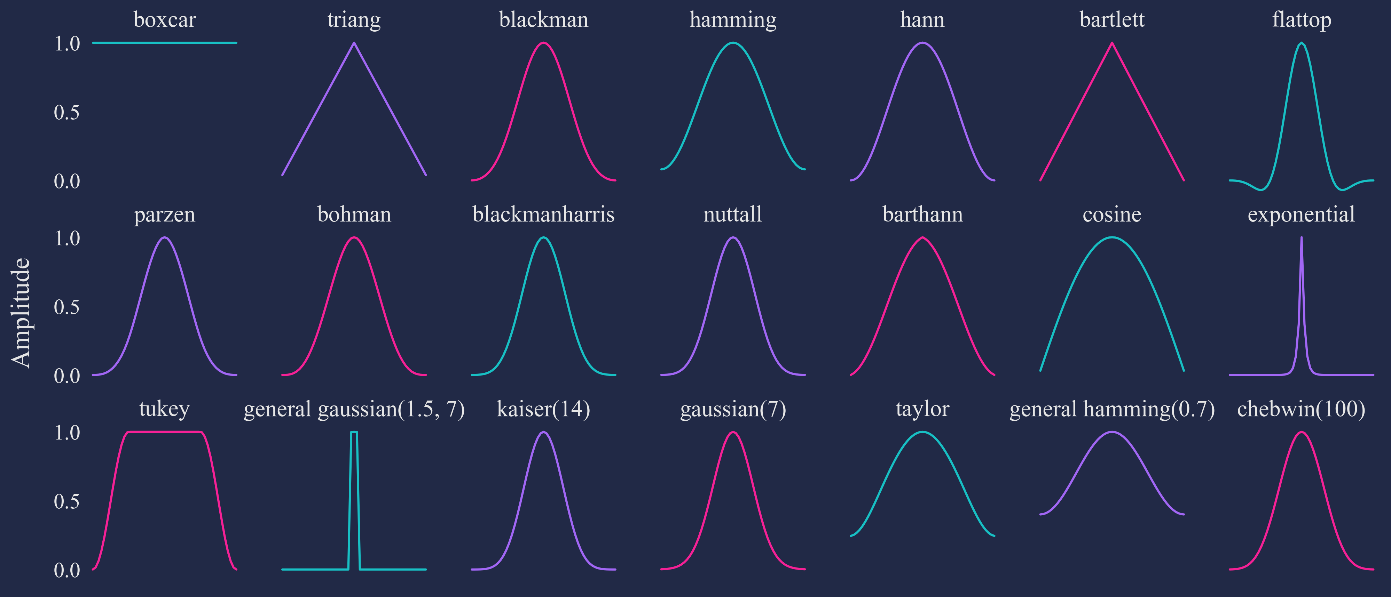
Рис. 3. Спектрограмма.

Чтобы создать спектрограмму, мы не можем использовать FFT[[1]](#footnote-1) для всей звуковой дорожки сразу. Для разложения сигнала на составляющие его частоты используется следующий алгоритм: спектрограмма разбивает сигнал на кратковременные сегменты, а затем вместо FFT для всего сигнала применяется STFT[[2]](#footnote-2) к каждому сегменту для того, чтобы определить частоты, содержащиеся в этом сегменте. Оконное преобразование Фурье для дискретного времени выглядит, как:

где – входной сигнал в момент времени , – некоторая оконная функция длины , – дискретно-временное преобразование Фурье оконных данных, центрированных по времени , – размер перехода в выборках между последовательными преобразованиями Фурье.

У спектрограммы есть некоторое количество характеристик, основными из них можно считать следующие:

* *Частота дискретизации* (sampling rate) – количество отсчетов в секунду. Наиболее распространенная частота семплирования составляет 44,1 кГц, она используется в рамках стандарта Audio CD. В данной работе мы будем использовать частоту по умолчанию в 2 раза меньше, 22050 Гц, чтобы уменьшить объём данных. Произведение частоты дискретизации и длины файла даёт нам общее количество выборок.
* *Размер окна* – длина участка сигнала, который анализируется с помощью оконной функции. Чем больше размер окна, тем более подробно анализируется каждый спектр каждого участка сигнала, однако высок риск потерять информацию о быстро меняющихся частотах. Окна должны быть достаточно малы, чтобы частотные характеристики спектра были относительно стабильными (то есть, чтобы сигнал в течение короткого временного промежутка являлся стационарным). В данной работе размер окна составляет 2048 сэмплов.
* *Тип окна* или *оконная функция* – функция, которая используется для ограничения сигнала до определенного участка или окна. Оконная функция умножается на сигнал для большей гладкости по краям при применении преобразования Фурье. Обычно в качестве оконной функции используется Гауссово окно, окно Хемминга, окно Ханна или окно Кайзера. На рисунке 4 изображено большинство разновидностей оконных функций. В данной работе по умолчанию используется окно Ханна.
* *Размер шага* определяет, насколько сильно перекрываются окна, используемые для анализа сигнала. Чем меньше размер шага, тем более подробно анализируется спектр каждого участка сигнала, но при этом увеличивается количество вычислений, необходимых для построения спектрограммы. В данной работе размер шага равен 512 сэмплам.

Рис. 4. Оконные функции для STFT.

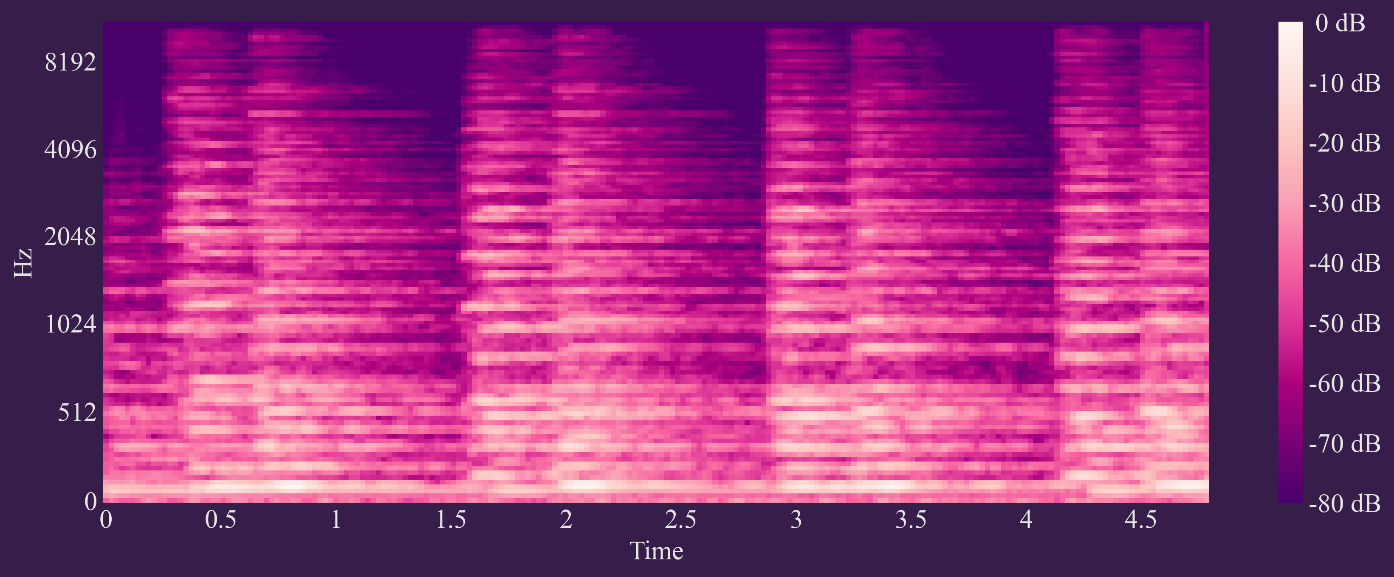
* 1. **Мел-спектрограмма**

Эксперименты ученых показали, что люди воспринимают частоты в нелинейном масштабе, гораздо лучше различая небольшие изменения высоты тона на низких частотах, чем на высоких. В результате был введён *мел* – психофизическая единица звука, близкая к тому, как звук воспринимается людьми. Мел-шкала соотносит воспринимаемую частоту чистого тона с его фактической измеренной частотой, оставляю полезную информацию с точки зрения восприятия слуха. Формула для преобразования частоты в шкалу Mel выглядит следующим образом:

Спектрограмма Mel вносит два важных изменения по сравнению с обычной спектрограммой, которая отображает зависимость частоты от времени:

* использует мел-шкалу вместо шкалы частот по оси *y*;
* использует шкалу децибел вместо амплитуды для обозначения цветов.

Мел-фильтры – треугольные функции, которые равномерно распределены на мел-шкале, переход к ней осуществляется с помощью матричного умножения мел-фильтров и спектрограммы. На рисунке 5 изображен пример мел-спектрограммы.

Рис. 5. Мел-Спектрограмма.

1. **Извлечение признаков**

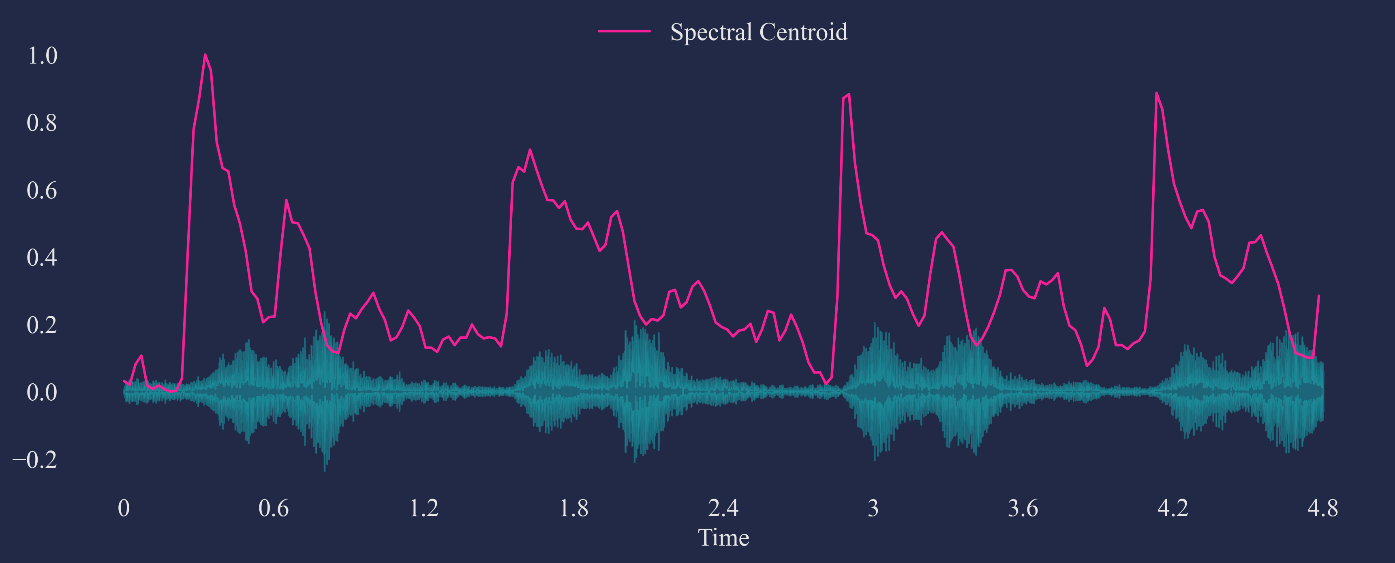
Осциллограмма (см. выше) хоть и является интуитивно понятной формой визуализации, однако аудиосигнал обычно не используется в таком виде в задачах машинного обучения. Для того, чтобы сделать сигнал полезным, необходимо извлечь менее очевидные особенности.

Чтобы получить адекватное представление музыкального сигнала, позволяющее осуществлять его дальнейшую обработку, применяется процедура выделения акустических характеристик для получения компактного представления сигнала в виде вектора признаков. Поэтому одной из важных подзадач внутри задачи классификации является задача идентификации и выбора надлежащего компактного подмножества исходного набора признаков, который бы обеспечил высокую точность определения жанра, в то же время не снижая показатели производительности и временные затраты.

Извлекаемые характеристики можно разделить по предмету зависимости анализируемых объектов: на временные (объекты зависят от времени) и на спектральные (объекты зависят от частоты). Из них формируется набор признаков, представленный 23-мерным вектором. Для данной задачи были выбраны как временные, так и спектральные характеристики, многие из них использовались в определённой части научных исследований. Рассмотрим каждый параметр подробно:

* 1. **Спектральный центроид** (Spectral Centroid)

Спектральный центроид – это точка равновесия спектра. Центроид является мерой спектральной формы, ее часто ассоциируют с понятием меры спектральной яркости, в данном случае – яркости звука. Большие значения центроида отражают смещение в сторону «более ярких», более высоких частот, тогда как меньшие значения отражают смещение в сторону более низких частот.

Рис. 6. Спектральный центроид.

Спектральный центроид может быть вычисляется, как:

где – величина преобразования Фурье в кадре , – частотный диапазон.

* 1. **Спектральная полоса пропускания** (Spectral Bandwidth)

Спектральная ширина в кадре t представляет собой среднеквадратичную разницу между частотным спектром и его центроидом. Если мы рассматриваем спектральный центроид как ожидаемое значение спектрального распределения кадра, то спектральную полосу пропускания можно рассматривать как вариант дисперсии спектрального распределения кадра. Большие значения спектральной полосы пропускания приравниваются к спектрально широкому кадру, малые значения – к узкому. Спектральная полоса пропускания может быть рассчитана, как:

где – спектральная величина в частотном диапазоне , – частота в диапазоне , а – спектральный центроид.

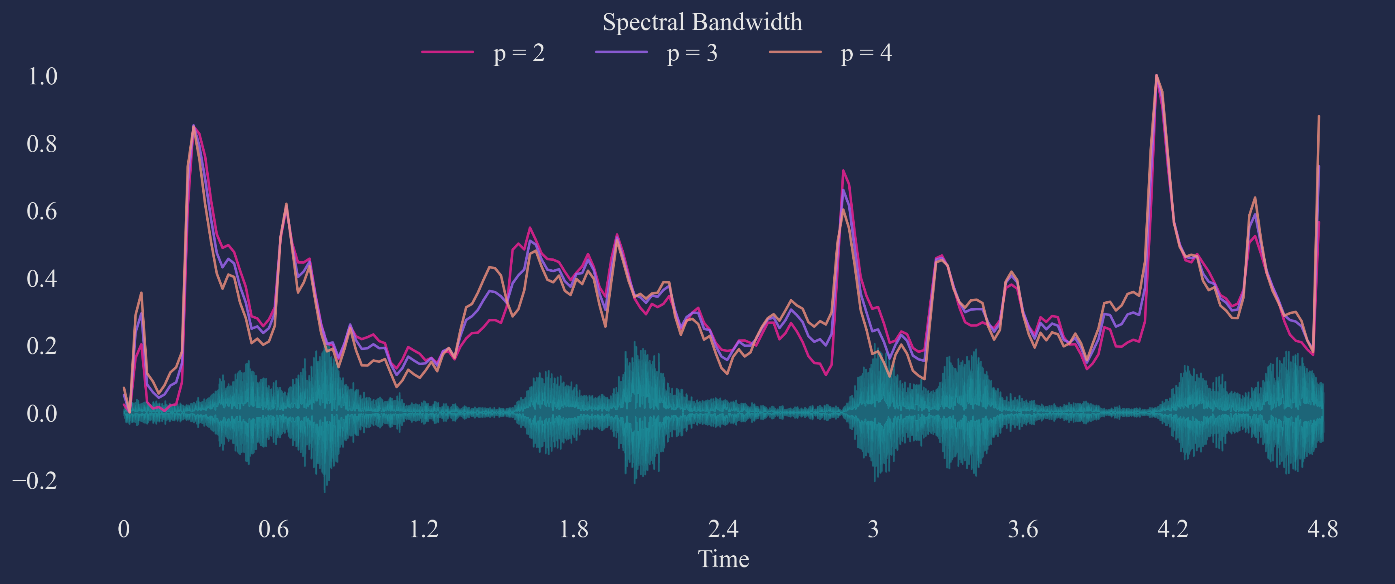
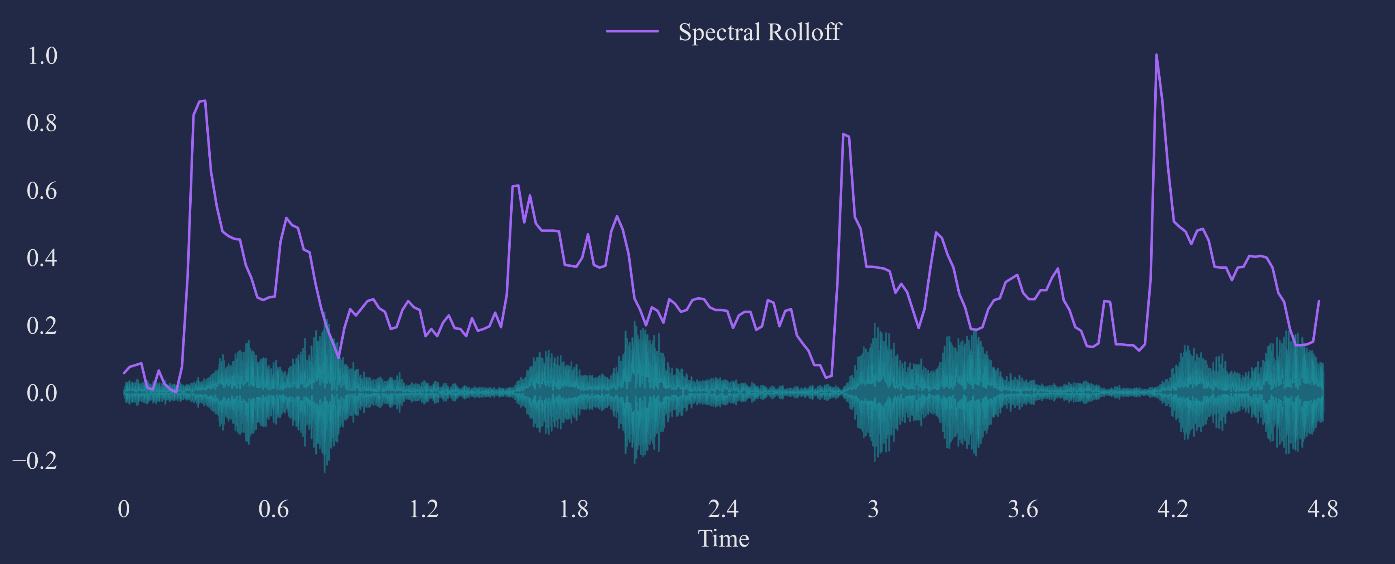


Рис. 7. Спектральная полоса пропускания для p=2, 3 и 4.

* 1. **Спектральный спад** (Spectral Rolloff)

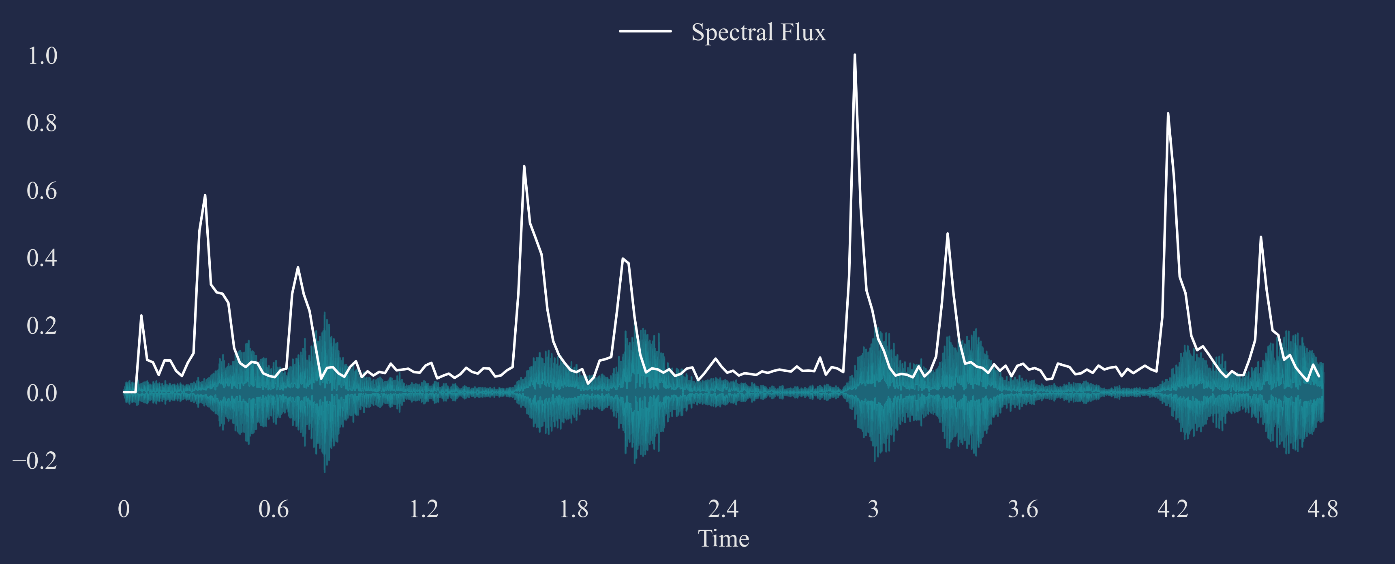
Спектральный спад определяется как частота , ниже которой находится определенный процент распределения амплитуд спектра, зачастую используются значения 85% или 95%, в данной работе пороговым значением принимается среднее значение по всему спектру. Спектральный спад так же, как и спектральный центроид, считается мерой спектральной формы. Он вычисляется как:

Рис. 8. Спектральный спад.

* 1. **Спектральный поток** (Spectral Flux)

Спектральный поток представляет собой меру величины спектрального изменения, которое происходит между последовательными кадрами аудиосигнала. Спектральный поток измеряет, насколько быстро изменяется спектральное содержание аудиосигнала с течением времени. Иными словами данная характеристика является показателем ритмической или текстурной вариативности композиции. Формула спектрального потока:

где - нормированная величина преобразования Фурье в окне .

Рис. 9. Спектральный поток.

* 1. **Частота пересечения нуля** (Zero-Crossing rate)

Частота пересечения нуля – мера шумности сигнала. Иначе говоря, это доля времени, в которую значение сигнала во временной области пересекает нулевую ось. Пересечение нуля происходит, когда последовательные отсчеты в цифровом сигнале имеют разные знаки. Периодические звуки, как правило, имеют небольшое значение этого параметра, в то время как шумные звуки, наоборот, имеют высокие показатели.

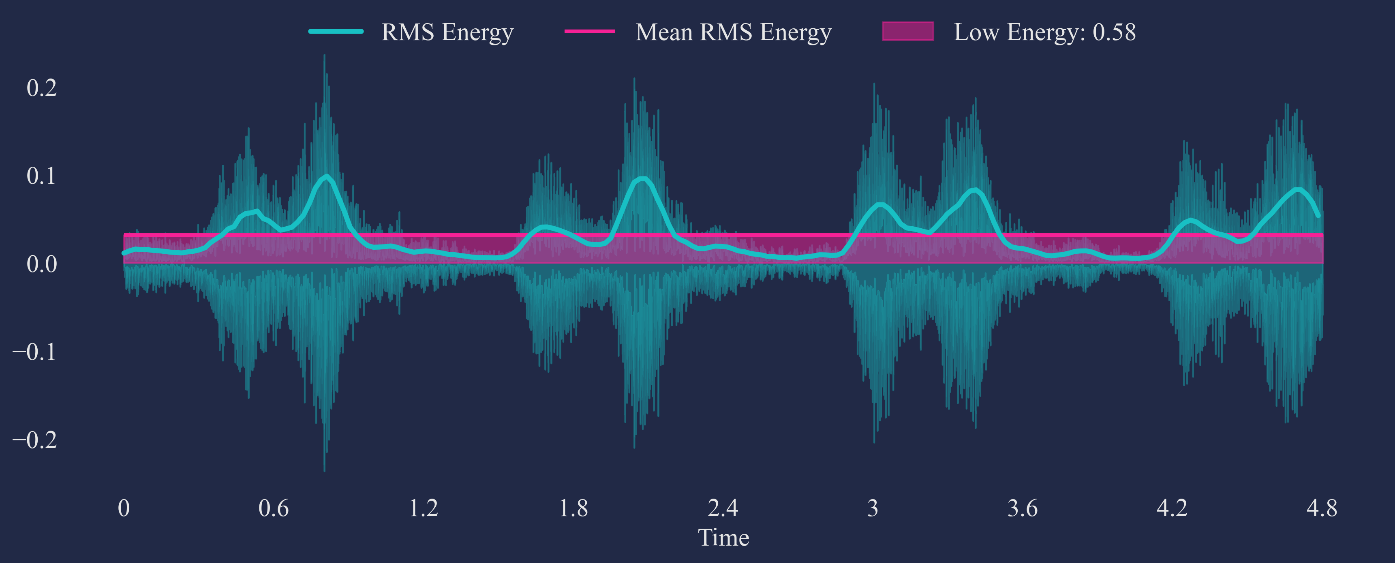
Рис. 10. Частота пересечения нуля.

Частота пересечения нуля вычисляется в каждом временном интервале сигнала по формуле:

где - сигнал во временной области, а функция имеет значение 1 или 0 для положительных и отрицательных аргументов соответственно.

* 1. **Низкоэнергетический уровень** (Low Energy Feature)

Функция Low Energy (низкоэнергетического уровня) – доля кадров, среднеквадратичное значение (RMS) которых ниже среднеквадратичного значения (RMS) песни в целом. Данная функция измеряет, насколько сконцентрирована энергия песни по отношению ко времени. Музыка, содержащая тихие части, будет иметь бо́льшую низкую энергетическую ценность, чем непрерывные звуки. В музыке разные жанры могут иметь разные характеристики громкости. Например, классическая музыка, как правило, имеет более широкий динамический диапазон, чем поп-музыка, которая обычно более сжата с точки зрения громкости.

Рис. 11. Low Energy.

* 1. **Мел-кепстральные коэффициенты** (MFCC)

Мел-кепстральные коэффициенты (Mel-frequency cepstral coefficients) – это перцептивно мотивированные коэффициенты кепстрального преобразования Фурье, которые используются для анализа звуковых сигналов. Другими словами, это набор спектральных коэффициентов, представляющих аудиосигнал с учетом особенностей человеческого восприятия звука.

После применения преобразования Фурье и получения спектрального представления аудиосигнала, спектр преобразуется с помощью логарифмической шкалы частот. Далее, полученный сигнал преобразуется обратно с помощью IFFT[[3]](#footnote-3), мы получаем кепстральный спектр, состоящий из последовательности коэффициентов. Кепстральные коэффициенты проходят через мел-фильтры, выполняется дискретное косинусное преобразование (DCT), результат которого и будет набором MFCC-коэффициентов. Алгоритм получения коэффициентов схематично изображен на рисунке 12.

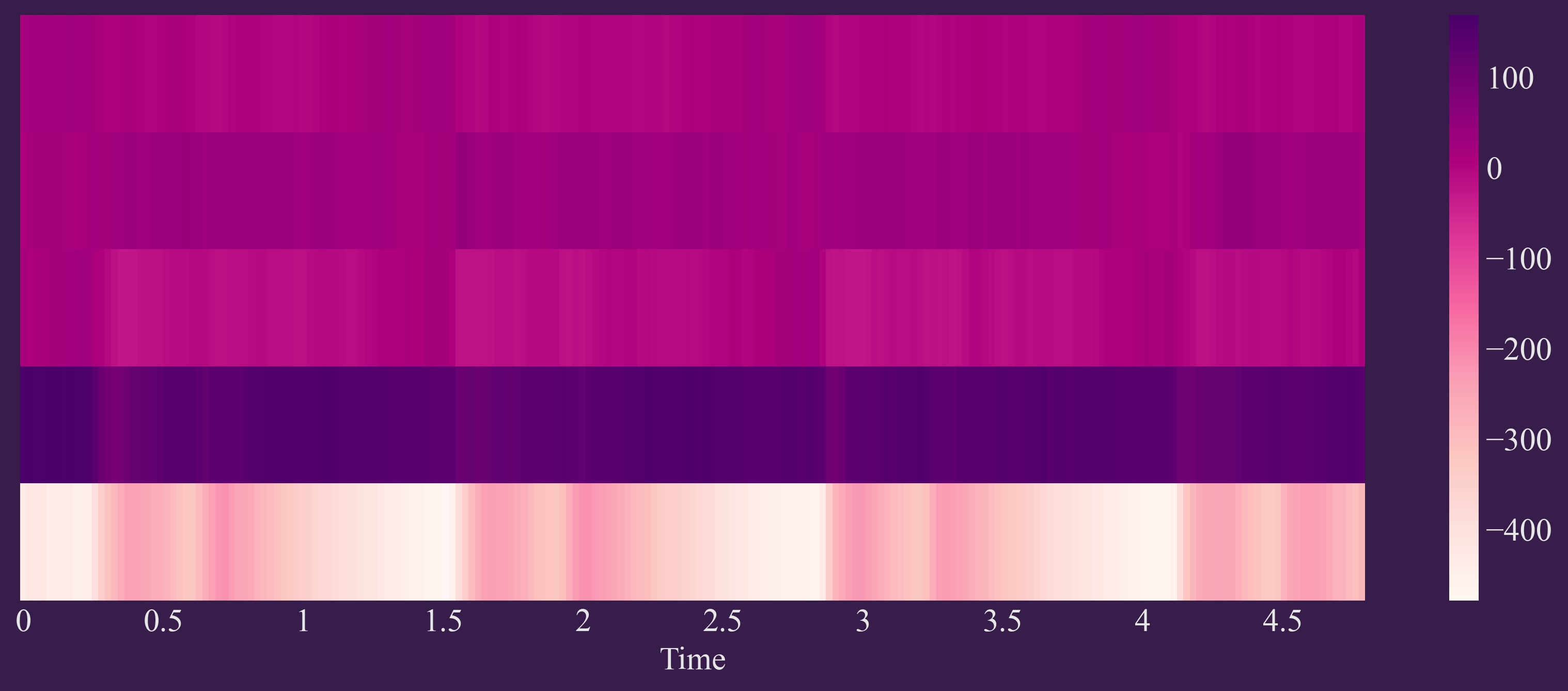
Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рис. 12. Алгоритм извлечения MFCC-коэффициентов.

После STFT диапазон частот может быть не похож на реальное восприятие звуков человеком. MFCC-коэффициенты улавливают спектральное восприятие лучше, они позволяет устранить лишние детали в сигнале, которые не несут важной информации (шумы), что улучшает качество распознавания звуковых сигналов.

Согласно результатам, полученным в ряде исследований, было определено, что только первые 5 коэффициентов MFCC обеспечивают наилучший уровень классификации музыкальных жанров. На рисунке 13 продемонстрировано изменение во времени первых 5 коэффициентов.

Рис. 13. MFCC-коэффициенты (1–5).

* 1. **Темп** (tempo)

Темп (или скорость) в музыке – наиболее ясная для понимания характеристика из ранее перечисленных. Темп определяется, как скорость звучания музыкального произведения. Темп обычно измеряется в BPM (Beats Per Minute) и указывает количество ударов, которые происходят в течение одной минуты. Таким образом, темп зависит от длительности каждого такта в музыке. Формула для вычисления темпа может быть представлена в виде следующего отношения:

где – количество тактов (ударов) в минуту, – длина такта в секундах.

Вычисление темпа может быть полезным в задачах жанровой классификации, так как музыка различных жанров может иметь различные темпы и ритмы.

Обычно ритмические характеристики не ограничиваются темпом: в некоторых исследованиях рассчитываются относительные амплитуды и периоды первого и второго пиков гистограмм биений, общая сумма этой гистограммы и т. д. Также, немало и спектральных характеристик, которые можно получить во время анализа композиции: это спектральный разброс (spread), спектральный скос (skew), спектральный эксцесс (kurtosis), спектральная плоскость (flatness), спектральный гребень (crest) и другие. К другим характеристикам можно отнести коэффициенты спектрального контраста (мера разницы в энергии между пиками и впадинами спектра), цветность (chroma, гармоническое содержание аудиосигнала) и другие. Все они могут быть полезны для различных задач обработки аудиосигналов. Но мы ограничимся вышеперечисленным набором характеристик для дальнейшего их использования в самой задаче жанровой классификации.

Таблица 1. Изначальный набор признаков.

|  |  |
| --- | --- |
| **№** | **Название характеристики** |
| 1 | Спектральный центроид |
| 2 | Спектральная полоса пропускания |
| 3 | Спектральный спад |
| 4 | Спектральный поток |
| 5 | Частота пересечения нуля |
| 6 | Низкоэнергетический уровень |
| 7 | MFCC-коэффициенты |
| 8 | Темп |

1. **Формирование наборов характеристик**

Для начала рассмотрим набор данных, который мы будем использовать для прикладной части данной работы.

* 1. **Используемый датасет**

Существует немало датасетов для задачи жанровой классификации композиций. Наиболее известные из них:

* Million Song Dataset (MSD): коллекция аудиофайлов и метаданных одного миллиона популярных музыкальных треков
* Free Music Archive (FMA): крупномасштабный набор данных, содержащих 106 тысяч аудиозаписей и 161 жанр
* GTZAN Dataset: набор данных, содержащий 1000 аудиофайлов в формате WAV (Waveform Audio Format) длительностью в 30 секунд каждый и 10 жанров, для каждого из которых есть по 100 примеров.

Последний датасет выбран для текущей задачи, его преимуществом является относительно небольшой объем данных (около 1 ГБ), обеспечивающий экономию вычислительной мощности. Наборы MSD и FMA являются более разнообразными, однако их объём составляет несколько сотен ГБ, их обработка может требовать предельно большой вычислительной мощности.

Датасет был собран в начале 2000-х годов как раз-таки с целью построения системы жанровой классификации музыки в автоматическом режиме. GTZAN содержит следующие музыкальные жанры:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| блюз | кантри | хип-хоп | метал | регги |
| классика | диско | джаз | поп-музыка | рок |

Набор хоть и ограничен набором жанров, однако он остается пригодным вариантом для обучения базовой модели жанровой классификации.

* 1. **Наборы признаков**

Для того, чтобы ответить на вопрос, какие признаки можно использовать для того, чтобы наиболее точно распознать жанр композиции, перед нами стоит задача подбора оптимального вектора характеристик, который мы будем использовать уже в задаче машинного обучения. Под оптимальностью в данном случае подразумевается:

1. Высокое качество: качество является очевидным требуемым результатом работы машинного обучения, оно может быть оценено с помощью различных метрик, например, с помощью accuracy, recall, precision и других метрик, используемых для оценки моделей машинного обучения.
2. Приемлемый размер: размер набора данных является важным критерием, ввиду технических и вычислительных ограничений в данном случае оптимальным набором будет считаться наиболее компактный (среди наборов, обеспечивающих одинаковое качество модели обучения, лучше тот, что содержит меньший объем данных).
3. Приемлемое время обучения: оптимальным набором будет считаться тот, который за наименьшее время обеспечит высокий уровень качества модели.

Было принято сформировать четыре различных набора данных, а затем сравнить их по указанным критериям. Опишем содержание каждого набора данных:

* Набор №1

В наборе используется расширенный перечень признаков, описанный в таблице 2. Для спектральных величин, первых пяти MFCC-коэффициентов и для частоты пересечения нули рассчитываются (среднее значение) и (стандартное отклонение), тем самым мы значительно уменьшаем объём получаемого набора. В наборе 1000 изначальных аудиодорожек (по 30 секунд), каждая аудиодорожка преобразуется в 22-мерный массив признаков. Общий объём данных составляет 22 тысячи элементов.

* Набор №2

В наборе аналогично используется расширенный перечень признаков, описанный в таблице 2. Предусмотрено увеличение данных путём разделения каждого аудиофайла на 3 отрывка одинаковой длины. В итоге в наборе 3000 изначальных аудиодорожек (по ~10 секунд), каждая аудиодорожка преобразуется в 22-мерный массив признаков. Общий объём данных составляет 66 тысячи элементов.

* Набор №3

В наборе аналогично используется расширенный перечень признаков, описанный в таблице 2. Предусмотрено увеличение данных путём разделения каждого аудиофайла на 5 отрывков одинаковой длины. В итоге в наборе 5000 изначальных аудиодорожек (по ~6 секунд), каждая аудиодорожка преобразуется в 22-мерный массив признаков. Общий объём данных составляет 110 тысяч элементов.

* Набор №4

Набор такого типа используется в схожем исследовании. Предусмотрено увеличение данных путём разделения каждого аудиофайла на 10 отрывков одинаковой длины. В итоге в наборе 10 тысяч изначальных аудиодорожек, каждая аудиодорожка содержит массив из 125 векторов, каждом из которых содержит первые пять MFCC-коэффициентов. Общий объём данных составляет чуть более 6 миллионов элементов.

Каждый из четырёх наборов в ходе его применения для задачи жанровой классификации будет проверен на оптимальность. Тот набор характеристик, который бы обеспечил наилучшую точность распознавания жанра за оптимальное время, в дальнейшем будет использоваться для:

* определения жанра отдельных музыкальных композиций (вероятности принадлежности к одному из жанров);
* составления базовых музыкальных рекомендаций на основе косинусного расстояния между характеристиками композиций.

Таблица 2. Расширенный набор признаков.

|  |  |
| --- | --- |
| **№** | **Название характеристики** |
| 1 – 2 | Спектральный центроид: среднее значение, СКО |
| 3 – 4 | Спектральная полоса пропускания: среднее значение, СКО |
| 5 – 6 | Спектральный спад: среднее значение, СКО |
| 7 – 8 | Спектральный поток: среднее значение, СКО |
| 9 – 10 | Частота пересечения нуля: среднее значение, СКО |
| 11 | Низкоэнергетический уровень |
| 12 – 13 | 1 коэффициент MFCC: среднее значение, СКО |
| 14 – 15 | 2 коэффициент MFCC: среднее значение, СКО |
| 16 – 17 | 3 коэффициент MFCC: среднее значение, СКО |
| 18 – 19 | 4 коэффициент MFCC: среднее значение, СКО |
| 20 – 21 | 5 коэффициент MFCC: среднее значение, СКО |
| 22 | Темп |

1. **Техническая часть**

Перейдём к технической части работы, в которой мы рассмотрим применение средств языка программирования Python в реализации поставленных целей. Но для начала определим некоторые термины, используемые в текущей части работы:

* 1. **Теоретический обзор**
* Тренировочная выборка (Training set) – часть исходного набора данных, которая используется для обучения модели: в этой выборке содержатся метки (целевые значения) для каждого примера, модель «тренируется» на основе сопоставления входных значений с соответствующими выходными значениями;
* Валидационная выборка (Validation set) – часть исходного набора данных, являющаяся независимой: эту выборку модель «не видит» в процессе обучения и не обновляет свои параметры, поэтому валидационная выборка позволяет оценить действительную производительность модели;
* Тестовая выборка (Test set) – отдельная часть исходного набора данных, независимая от тренировочной и валидационной выборок: используется тестовая выборка после обучения модели для окончательной оценки производительности;
* Функция потерь (Loss function) – мера оценки расхождения между предсказаниями модели и истинными значениями целевой переменной, цель состоит в минимизации значений. Примерами могут быть: среднеквадратичная ошибка (MSE), абсолютная ошибка (MAE), бинарная кросс-энтропия, категориальная кросс-энтропия, логарифмическая функция потерь (Log Loss) и другие;
* Оптимизатор – алгоритм оптимизации параметров модели в процессе обучения. Он использует градиентные методы для поиска оптимальных значений, минимизирующих функцию потерь. Примерами могут быть: стохастический градиентный спуск (SGD), Adam (Adaptive Moment Estimation), RMSprop (Root Mean Square Propagation), Adagrad (Adaptive Gradient) и другие;
* Батч – набор обучающих примеров, которые передаются модели одновременно для вычисления градиента и обновления параметров;
* Эпоха – полный проход всех обучающих примеров в наборе данных. Количество эпох настраивается в балансном количестве между ситуацией недообучения[[4]](#footnote-4) и переобучения[[5]](#footnote-5) модели и определяет количество проходов всего набора данных в процессе обучения.
* Функция активации – параметр нейронной сети, который определяет выходное значение каждого нейрона в слое и добавляет нелинейность в модель, позволяя сети обрабатывать сложные нелинейные зависимости в данных. Примерами могут быть сигмоида, гиперболический тангенс, ReLU, ELU, softmax и другие функции. Функцию ReLU, используемую в задаче, можно представить, как:

Также используемую функцию softmax, которая преобразует вектор входных значений в вероятностное распределение, можем представить, как:

Подробнее остановимся на архитектуре нейронной сети и видах архитектур, используемых а данной задаче

* Архитектура нейронной сети – структура и организация слоёв и нейронов в сети. Архитектура включается в себя различные типы слоёв (свёрточные, рекуррентные, полносвзяные), их количество, размер и их взаимосвязь;
* CNN, Convolutional Neural Network – сверточная нейронная сеть, тип нейронной сети, включающий в себя специализированные слои свертки и пулинга. Сверточные слои – слои, применяющие фильтры (ядра свертки) к входным данным. Свёртку можно представить, как:

где – входной N-мерный вектор характеристик, – ядро, и – индексы столбцов и строк матрицы-результата. Слой пулинга – следующий за слоем свертки слой, уменьшающий размерность выходных данных и улучшающий инвариантность к масштабированию и небольшим сдвигам. Одной из наиболее распространённых операций пуллинга является операция максимального подвыбора (Max Pooling), когда область входного представления делится на подобласти, в каждой из которых выбирается максимальное значение. Формально слой пуллинга можно описать, как:

где – выход слоя , – функция активации, и – коэффициенты сдвига слоя . В конце сети обычно следуют полносвязные слои, которые соединяют все признаки, выделенные предыдущими слоями, и используют их для классификации. Также для улучшения обобщающей способности и предотвращения переобучения используются методы регуляризации, такие как Dropout (случайное «отключение» некоторых нейронов во время обучения) и Batch Normalization (нормализация данных для стабилизации и ускорения обучения);

* RNN, Recurrent Neural Network – рекуррентная нейронная сеть, тип нейронной сети. Основным строительным блоком RNN является рекуррентный слой, он позволяет передавать информацию между последовательными элементами входных данных. В архитектуре RNN могут использоваться Dropout-слои, а в конце может использоваться один или несколько полносвязных слоев. В данной задаче используется LSTM (Long Short-Term Memory) вариант нейронной сети, предназначенный для преодоления проблемы затухающего градиента и обеспечения создания в последовательных данных долгосрочных зависимостей;
* MLP, Multilayer Perceptron – многослойный персептрон, архитектура нейронной сети, являющаяся одной из классических. MLP содержит входной и выходной слой, а также слои между ними, которые называются скрытыми. В каждом скрытом слое содержатся нейроны, которые связываются с каждым нейроном смежных слоёв (слои полносвязные). В процессе обучения персептрона используется метод обратного распространения ошибки, когда ошибки предсказаний сети обратно распространяются через сеть, и веса нейронов обновляются. MLP, в отличие от CNN и RNN, не учитывает пространственные или последовательные структуры в данных;
* Метрики оценки модели – показатели производительности и эффективности нейронной сети, они позволяют оценить качество модели на этапе обучения и качество предсказаний модели на этапе тестирования. Основными метриками являются: *accuracy* (метрика достоверности предсказания, насколько часто модель даёт правильные прогнозы – наиболее распространённая метрика), *precision* (метрика точности, доля истинно положительных образцов относительно всех положительно классифицированных примеров), *recall* (метрика полноты, доля истинно положительных образцов, правильно классифицированных моделью относительно всех истинно положительных примеров), *F1-score* (F-мера, комбинация precision и recall, сбалансированный показатель между точностью и полнотой). В задачах многоклассовой классификации есть особенности вычисления некоторых метрик[[6]](#footnote-6).

Теперь мы можем приступить к определению технических задач.

* 1. **Перечень и реализация основных задач, результаты**

Перед основной задачей было необходимо составить наборы данных, описанные в п.3. Для наборов 1-3 был использован подход, при котором параллельно выполняются несколько процессов, в программировании это называется мультипроцессингом. Его использование повысило производительность в 2.2, сэкономив более 15 минут ожидания. Быстрее всего собирались данные для набора №4 (77 секунд), медленнее – для набора №2 (303 секунды). Рассмотрим алгоритм выполнения задачи для каждого из четырёх наборов данных:

1. Обработка созданного в п.3 массива данных с характеристиками аудиофайлов, разделение датасета на выборки X и Y;
2. Разделение выборок X и Y на обучающую, валидационную и тестовую выборки;
3. Построение трёх различных архитектур нейронных сетей: сверточной нейронной сети (Convolutional Neural Network, CNN), рекуррентной нейронной сети (Recurrent Neural Network, RNN) и многослойного персептрона (Multilayer Perceptron, MLP);
4. Компиляция и обучение модели: выбор оптимизатора, функции потерь, количества эпох и размера батча. Вывод графика динамики метрики accuracy и функции потерь;
5. Проверка модели на тестовой выборке: построение матрицы ошибок и вывод четырёх метрик: accuracy, precision, recall и f1-score;
6. Сравнение результатов трёх архитектур и выбор лучшей по скорости обучения модели и её качеству.

Полный код и описание пунктов алгоритма представлено в виде программного кода, доступного в приложении №2 в следующих подпунктах: «Code 1. Набор 1», «Code 2 Набор 2.1.», «Code 2. Набор 2.2.», и «Code 3. Набор 3». Далее мы опишем основные итоги, полученные в ходе выполнения задачи.

В рамках задачи было построено 12 нейронных сетей по 3 типа для 4 наборов. Было принято исключить результат MLP для набора №4 по причине низкой работоспособности. В таблице 3 представлены показатели четырёх метрик для каждого из случаев. Было решено определить 3 лучшие модели, основываясь на метрике accuracy. Общая точность сверточной модели нейронной сети для наборов 2 и 3 показала значения 79,6% и 79,9% соответственно. CNN модель для набора №3 показывала максимальную точность 82,4% . Но наилучшее качество по метрике accuracy показала модель многослойного персептрона также для набора №3. Средняя достоверность предсказания MLP составляла 81,4%, а наилучшая – 84,9%.

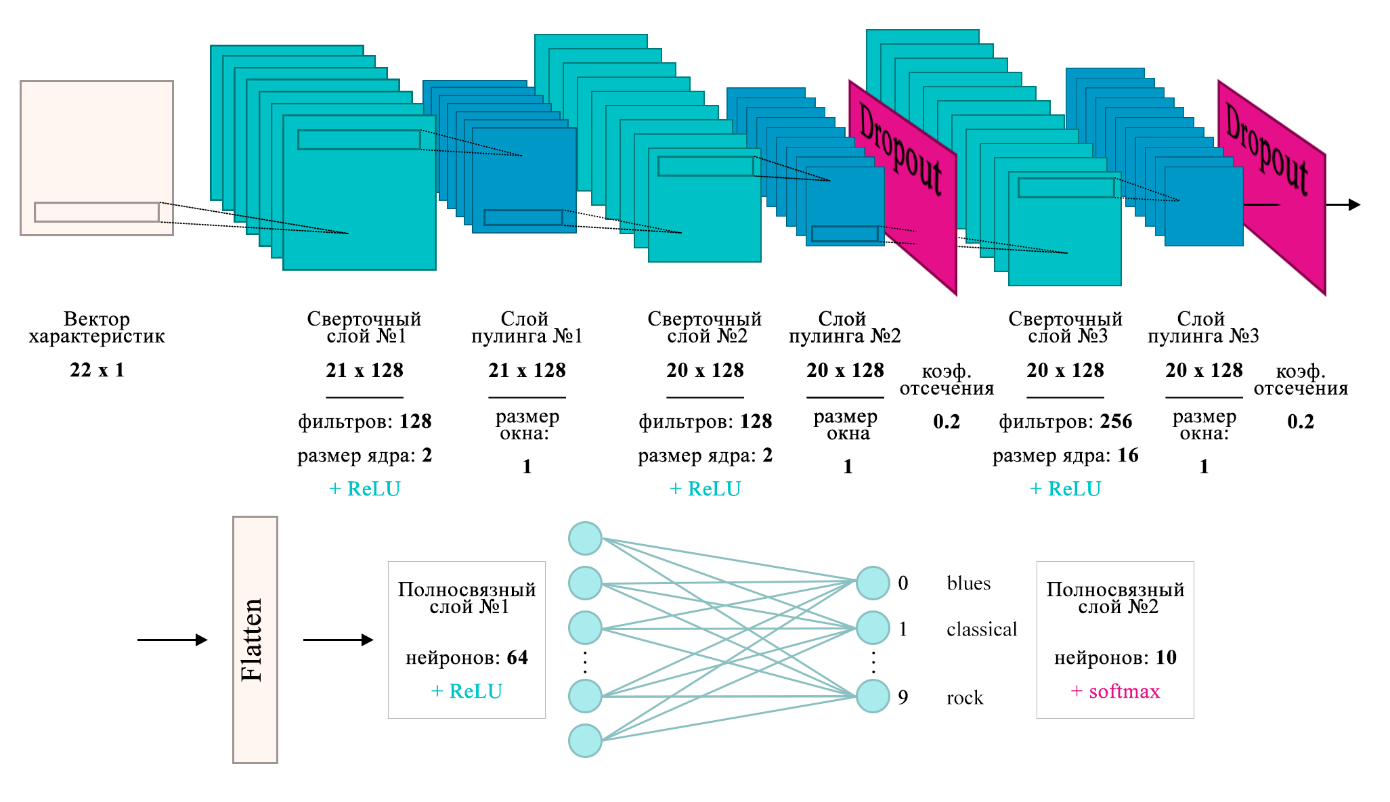
Таблица 3. Результаты работы .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Модель:** | **MLP** | **CNN** | **RNN** | **MLP** | **CNN** | **RNN** |
| **Набор:** | **Набор №1** | | | **Набор №2** | | |
| accuracy | 0,637 | 0,629 | 0,583 | 0,779 | **0,796** | 0,765 |
| recall | 0,635 | 0,629 | 0,577 | 0,779 | 0,795 | 0,766 |
| precision | 0,637 | 0,653 | 0,583 | 0,779 | 0,796 | 0,765 |
| f1 | 0,649 | 0,624 | 0,594 | 0,786 | 0,801 | 0,771 |
| **Набор:** | **Набор №3** | | | **Набор №4** | | |
| accuracy | **0,814** | **0,799** | 0,781 | – | 0,718 | 0,633 |
| recall | 0,813 | 0,801 | 0,780 | – | 0,715 | 0,627 |
| precision | 0,814 | 0,799 | 0,781 | – | 0,718 | 0,633 |
| f1 | 0,816 | 0,814 | 0,783 | – | 0,738 | 0,633 |

Опишем архитектуры моделей CNN и MLP, которые используются для набора №3. Архитектура CNN включает в себя следующие слои:

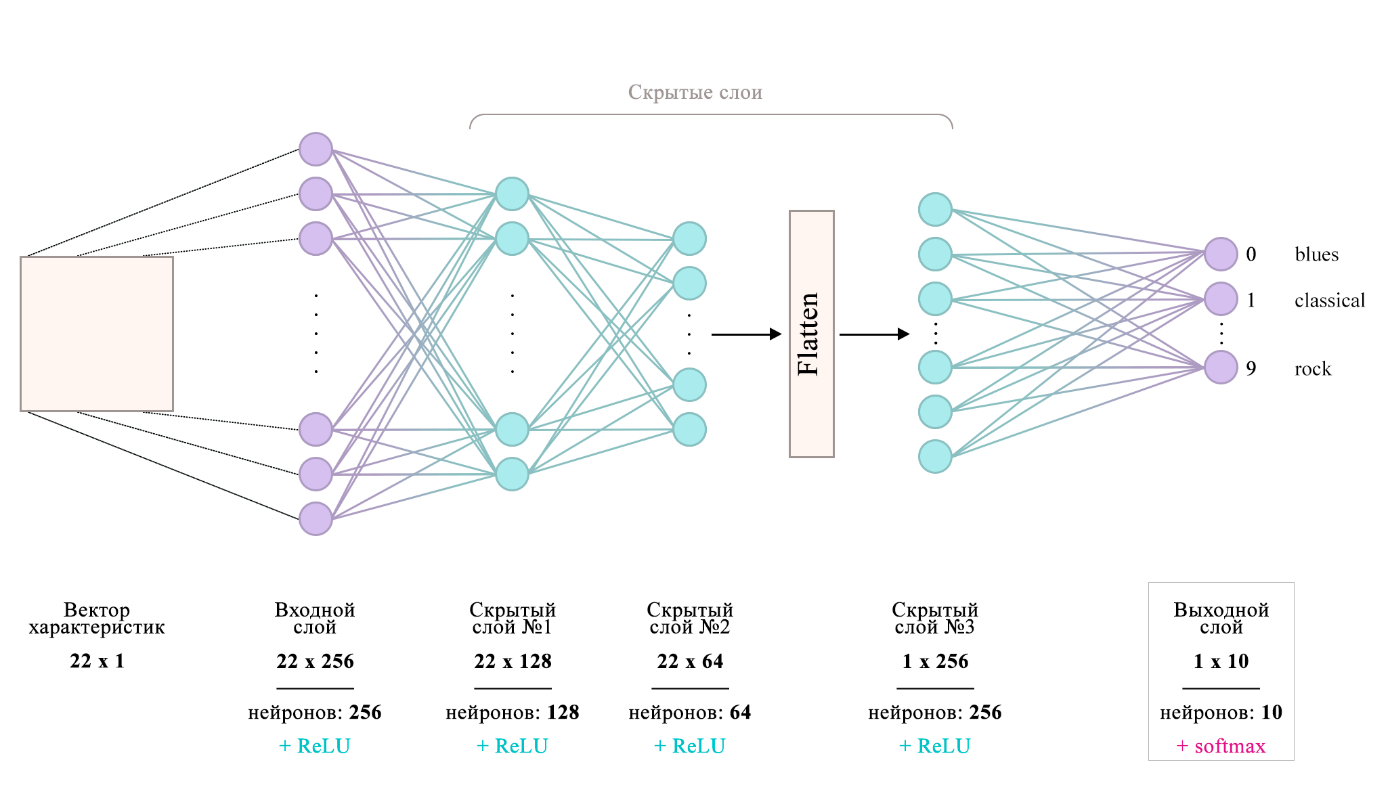
1. Conv1D №1: сверточный слой (1D – для одномерных данных) с 128 фильтрами, размером ядра = 2 и функцией активации ReLU.
2. MaxPooling1D №1: слой пулинга, с размером окна = 1.
3. Conv1D №2: сверточный слой с 128 фильтрами, размером ядра = 2 и функцией активации ReLU.
4. MaxPooling1D №2: слой пулинга, с размером окна = 1.
5. Dropout №1: слой регуляризации, с коэффициентом отсечения = 0.2.
6. Conv1D №2: сверточный слой с 256 фильтрами, размером ядра = 16 и функцией активации ReLU.
7. MaxPooling1D №3: слой пулинга, с размером окна = 1.
8. Dropout №2: слой регуляризации, с коэффициентом отсечения = 0.2.
9. Flatten: выравнивающий слой, который преобразует многомерные данные в одномерный вектор.
10. Dense №1: полносвязный слой с 64 нейронами и функцией активации ReLU.
11. Dense №2: полносвязный слой с 10 нейронами и функцией активации softmax. В этом слое используется L1 регуляризация[[7]](#footnote-7) с коэффициентом 0.15.

На рисунке 14 схематично изображена архитектура CNN для набора №3:

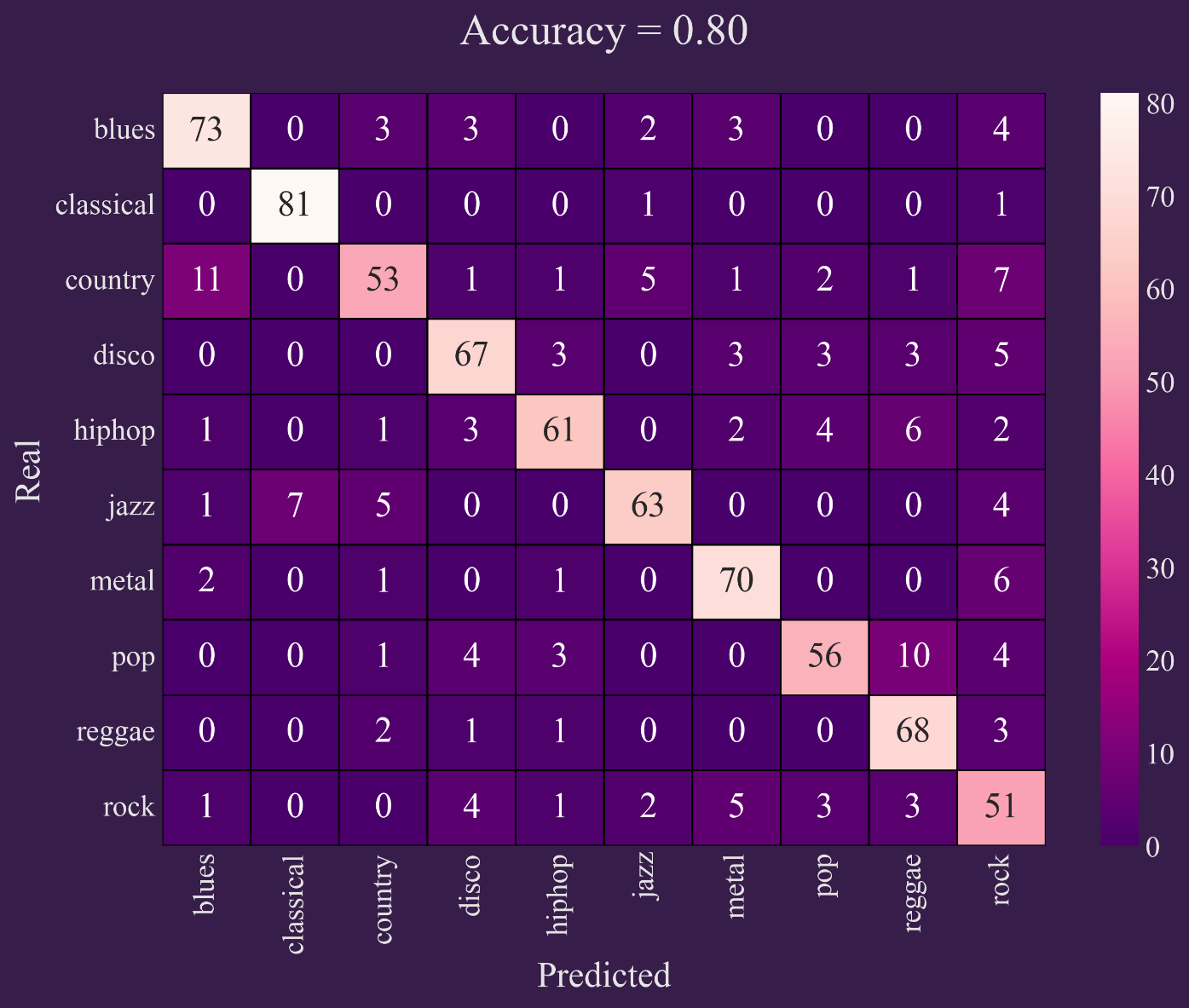
Рис. 14. Архитектура Сверточной Нейронной сети (CNN).

Архитектура MLP также схематично изображена на рисунке 15 и представляет следующий вид:

1. Входной слой: полносвязный слой с 256 нейронами и функцией активации ReLU.
2. Скрытый слой №1: полносвязный слой с 128 нейронами и функцией активации ReLU.
3. Скрытый слой №2: полносвязный слой с 64 нейронами и функцией активации ReLU.
4. Flatten: выравнивающий слой, который преобразует многомерные данные в одномерный вектор.
5. Скрытый слой №2: полносвязный слой с 256 нейронами и функцией активации ReLU.
6. Выходной слой: полносвязный слой с 10 нейронами и функцией активации softmax.

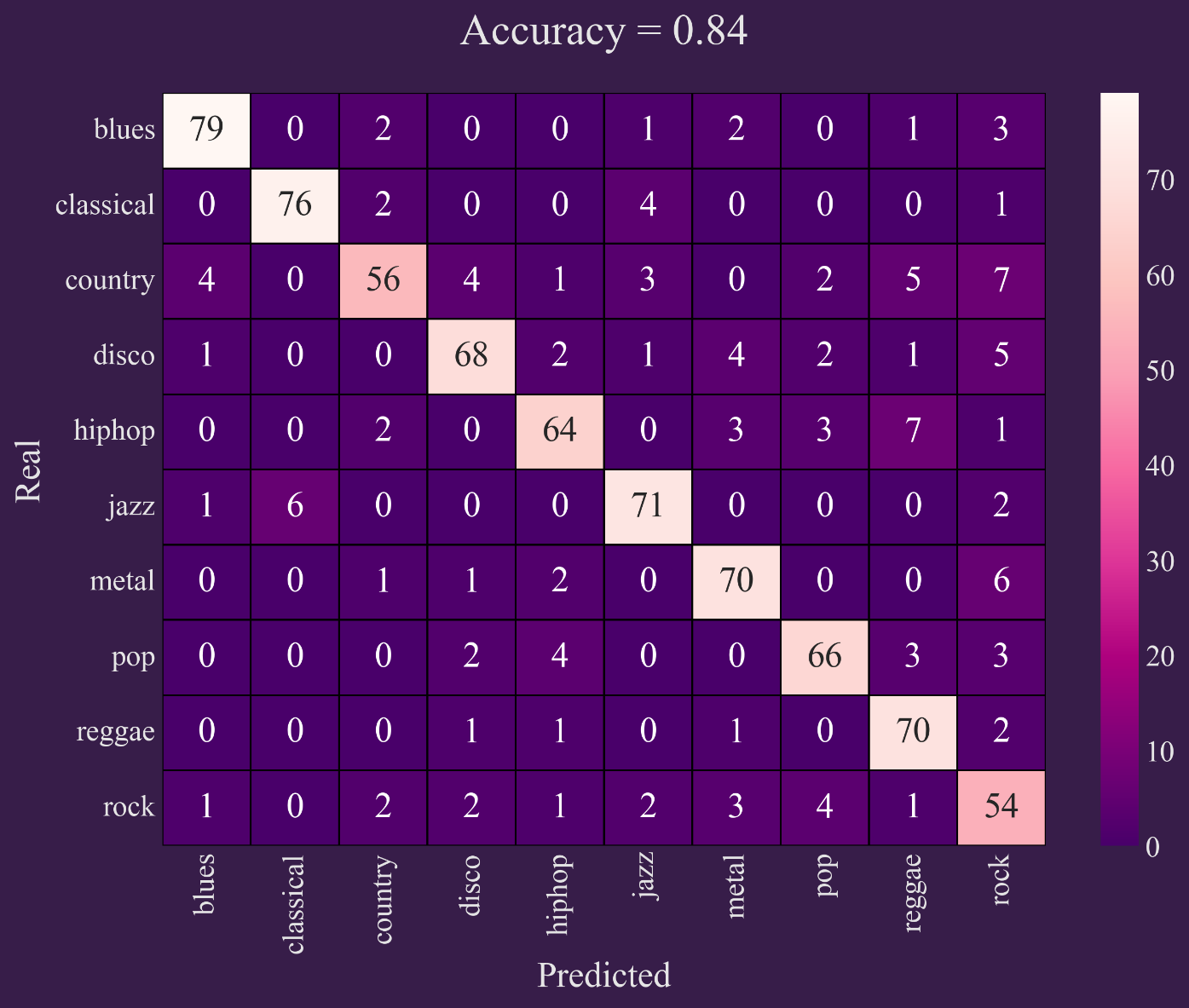
Рис. 15. Архитектура многослойного персептрона (MLP).

Рассмотрим матрицы ошибок для обеих моделей. Confusion Matrix является инструментом для оценки производительности моделей классификации. Она позволяет визуализировать результаты классификации в виде тепловой карты, на которой сравниваются фактические и предсказанные классы. На рисунках 16 и 17 можем увидеть результаты классификации на наборе №3 для моделей CNN и MLP соответственно.

Рис. 16. Матрица неточностей для модели CNN.

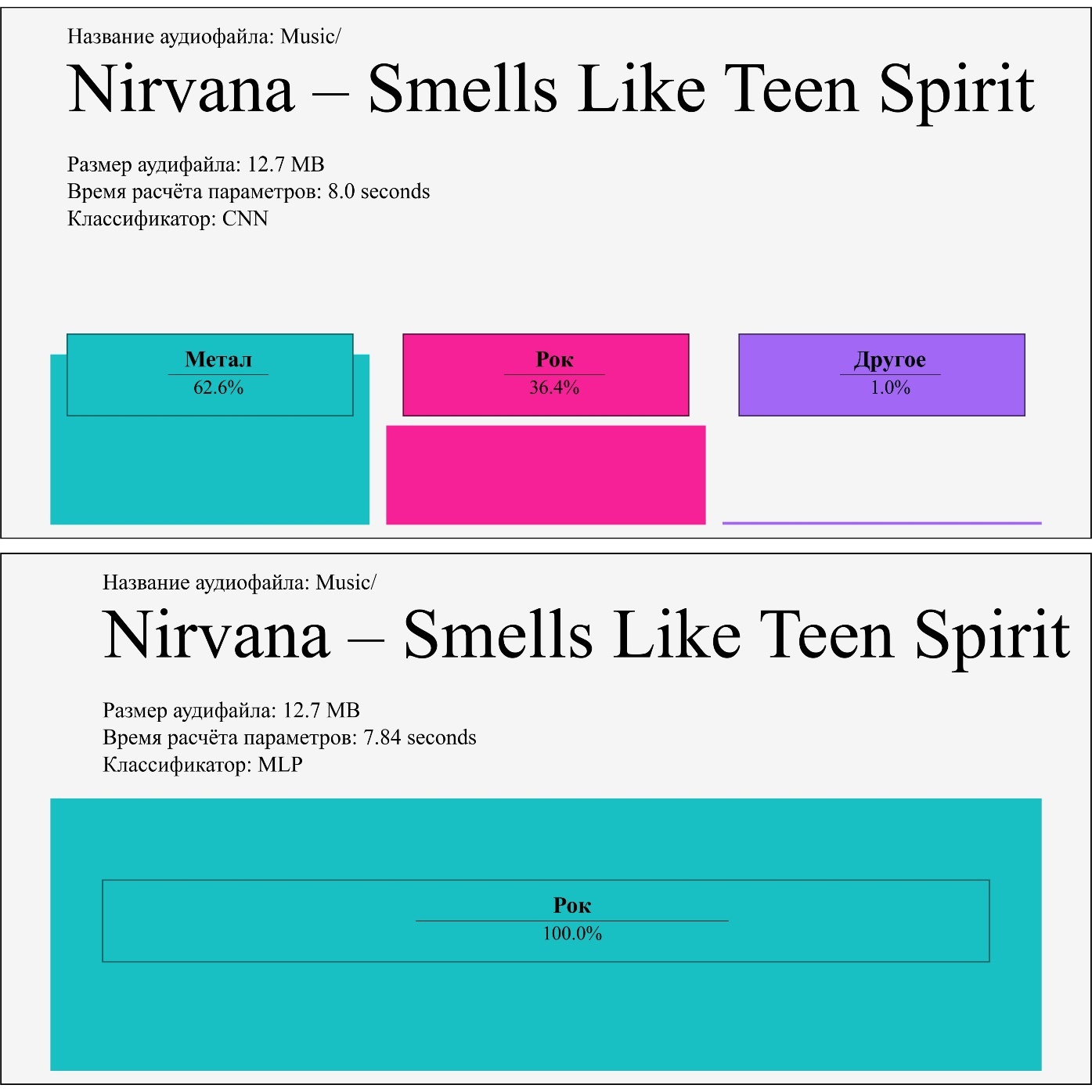
Видим, что в данном случае модель лучше всего справляется с классификацией классических произведений (98%), регги (91%) и метала (88%). Чаще всего модель ошибается при распознавании музыки кантри (65%), часто путая этот жанр с блюзом, поп-музыки (72%), ложно определяя регги, и рока (73%). Также видно, что некоторое количество раз модель ложно определила джаз, как классику, кантри и метал, как рок, однако общая точность модели всё равно считается достаточно хорошей.

Взглянем на матрицу ошибок модели персептрона:

Рис. 17. Матрица неточностей для модели MLP.

Метрика accuracy достигает 84%, из всех жанров в текущем примере лучше всего распознается регги (94%), классика (92%), блюз (90%), джаз (89%) и метал (88%), хуже всего – кантри (68%). Среди ошибок отметим наиболее частые: неверное распознавание джаза, как классической музыки; хип-хопа и кантри, как регги; кантри, диско и метала, как рок-музыки. Анализируя матрицу ошибок, существенная часть ошибок в распознавании жанра зачастую похожи на те, которые мог бы совершить обычный слушатель.

После того, как выбраны наиболее качественные модели, можем сохранить их веса и использовать для определения жанра отдельных композиций. Для этого в программном коде «Code 4. Code 4. Предсказание жанра» мы создаём функцию, которая определяет процентную вероятность принадлежности композиции к одному из десяти жанров. Она принимает на вход путь к файлу, а также один из двух типов модели классификации. Для наглядности создаётся «карточка» композиции, в которой указано её название, размер аудиофайла этой композиции, время, требуемое на сбор вектора из 22 характеристик, а также столбчатая диаграмма с наиболее вероятным жанром. Пример «карточки» изображён на рисунках 18-19.

Рис. 18. Сравнение результата классификации жанра песни Nirvana.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниеРис. 19. Сравнение результатов классификации   
фрагмента из балета «Щелкунчик».

* 1. **Система музыкальных рекомендаций**

В качестве дополнительной задачи реализуем базовую систему рекомендаций, основанную на измерении схожести между конкретной композицией и другими композициями из базы аудиофайлов GTZAN, дополненной собственным набором из ста композиций различных жанров.

Системы рекомендаций музыки в популярных стриминговых платформах обычно основываются на разных критериях, таких как история прослушивания, персональные предпочтения, тренды в регионе или стране, текстовое содержание. Успешно реализуется принцип коллаборативной фильтрации, который основывается на рекомендации треков, часто встречающихся у других пользователей с похожими предпочтениями. Однако в рамках данной задачи за наибольшую музыкальную схожесть будут отвечать метрики расстояния, являющиеся мерой сходства или различия между векторами ранее рассмотренных характеристик. Заранее были выбраны некоторые метрики расстояния , где u, перечислим их:

1. Евклидово расстояние (Euclidean Distance), наиболее распространенная метрика расстояния. Евклидово расстояние измеряет длину прямой линии, соединяющей две точки в пространстве. Вычисляется по формуле:
2. Косинусное расстояние (Cosine Distance) измеряет угол между двумя векторами в многомерном пространстве. Вычисляется, как:
3. Расстояние городских кварталов, также известное как манхэттенское расстояние (Manhattan Distance). Оно описывает ограниченное движение, будто мы перемещаемся по перпендикулярным улицам Манхэттена. Формула этого расстояния:
4. Корреляция по расстоянию (Correlation Distance). Эта метрика оценивает степень линейной зависимости между векторами и определяется по формуле:
5. Расстояние Чебышёва (Chebyshev Distance). Эта метрика измеряет наибольшее различие между двумя векторами в каждом измерении. Формула данного расстояния:
6. Метрика Канберры (Canberra Distance). Она широко используется для измерения различий между двумя распределениями или наборами данных. Метрика рассчитывается, как:

Наиболее похожие композиции будут иметь наименьшие значения расстояний, то есть при минимальных значениях расстояний композиции будут наиболее похожи друг на друга. Приведем пример реализации данной системы рекомендаций с помощью программного кода «Code 5. Музыкальные рекомендации», на рисунке 20 отобразим наиболее похожие композиции для произведения Фридерика Шопена «Фантазия-экспромт»:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниеРис. 20. Список рекомендаций для аудиофайла   
«Chopin – Fantasie Impromptu Op.66».

Из 10 композиций 6 – классические произведения и по 2 – в стиле джаз и блюз, выбраны они на основе общего ранга, являющегося суммой рангов по шести метрикам расстояний. Список рекомендаций из 50 песен формируется примерно за 100 миллисекунд, сам список достаточно разнообразный, несмотря на то, что «база» треков предельно небольшая (на некоторых стриминговых платформах количество треков превышает 100 миллионов).

Можно утверждать, что не существует лучшей метрики, ведь сами по себе рекомендации сугубо субъективны, предложенные композиции могут понравиться одной части аудитории и не заинтересовать другую. Однако в ходе наблюдения наибольшее разнообразие в подборе рекомендаций показала метрика Канберры: как фактор сортировки метрика показывала весьма интересные и неожиданные предложения.

# **Заключение**

В данной работе мы погрузились в теорию обработки сигналов, в спектральные и временные характеристики, которыми обладают музыкальные композиции. Мы проанализировали 4 набора данных, а также построили 12 различных нейронный сетей, в результате которых мы можем считать, что лучшими для работы с жанровой классификацией оказались MLP и CNN нейронные сети, обученные на наборе из 22 характеристик, вычисляемых на пятикратно разделенных аудиофайлах: лучший показатель метрики accuracy составил 85%. Результат не является пределом, точность может быть увеличена с увеличением наборов для обучения, с улучшением вычислительной техники и модификацией архитектуры нейронных сетей.

Классификация музыки, в том числе жанровая, построение рекомендательных систем и анализ музыки в целом – все эти области являются сложными и активно изучаемыми в наши дни. Развитие и углубление в каждую из них требует дальнейших исследований, использования новых алгоритмов и моделей нейронных сетей, а также внимания к уникальным особенностям музыкальных произведений и предпочтений слушателей.

# **Список использованных источников**

1. C. N. Silla, A. L. Koerich and C. A. A. Kaestner, «A feature selection approach for automatic music genre classiﬁcation» in International Journal of Semantic Computing, 03, 183–208, 2009.
2. P. Geoffroy, «A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project», 2004.
3. C. Silla, A. Koerich, and C. Kaestner, «A machine learning approach to automatic music genre classification,» in Journal of the Brazilian Computer Society, vol. 14, no. 3, 2008.
4. G. Tzanetakis and P. Cook, «MARSYAS: a framework for audio analysis» in Organised Sound, 4(3), 169–175, 2000.
5. G. Tzanetakis, G. Essl, and P. Cook, «Automatic musical genre classification of audio signals,» in Proc. Int. Symp. Music Information Retrieval (ISMIR), Oct. 2001.
6. G. Tzanetakis and P. Cook, «Musical genre classification of audio signals,» IEEE Transactions on speech and audio processing, vol. 10, no. 5, pp. 293–302, 2002.
7. K. Choi, G. Fazekas and M. Sandler, «Automatic tagging using deep convolutional neural networks,» in Intl. Society for Music Information Retrieval Conf. (ISMIR), 2016.
8. K. Choi, G. Fazekas, M. Sandler, and K. Cho, «Convolutional recurrent neural networks for music classification,», 2016.
9. Q.G. Rafi, M. Noman, S.Z. Prodhan, S.S. Alam and D. Nandi, «Comparative Analysis of Three Improved Deep Learning Architectures for Music Genre Classification» in International Journal of Information Technology and Computer Science, 13, 1-14, 2021.
10. A. Van den Oord, S. Dieleman, and B. Schrauwen, «Deep content-based music recommendation,» in Advances in Neural Information Processing Systems, pp. 2643–2651, 2013.
11. A. Van den Oord, S. Dieleman, and B. Schrauwen, «Transfer learning by supervised pre-training for audio-based music classification,» in Conference of the International Society for Music Information Retrieval, (ISMIR 2014), 2014.
12. M. Vaibhavi, P. R. Krishna, «Music Genre Classification using Neural Networks with Data Augmentation», 2021.
13. M. Creme, C. Burlin, R. Lenain, “Music Genre Classification ”, Stanford University, December 15, 2016.

# **Приложение**

Список программ, в которых реализована техническая часть работы:

* Code 0. Получение данных
* Code 1. Набор 1
* Code 2. Набор 2.1
* Code 2. Набор 2.2
* Code 3. Набор 3
* Code 4. Предсказание жанра
* Code 5. Музыкальные рекомендации
* Code 6. Дополнительная визуализация

Программные коды и другие файлы, используемые в работе, доступны в публичном репозитории по [данной ссылке](https://github.com/m1157/FinU/tree/82e443de1594f7344435689e0325f100f28c3662/%D0%A2%D0%90%D0%94%D0%B8%D0%9C%D0%9E_%D0%9A%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F).

1. **FFT** – Быстрое преобразование Фурье (БПФ), алгоритм ускоренного вычисления дискретного преобразования Фурье (ДПФ). [↑](#footnote-ref-1)
2. **STFT** – Оконное преобразование Фурье, разновидность преобразования Фурье, метод анализа коротких участков сигнала определенной длины, называемых окнами. [↑](#footnote-ref-2)
3. **IFT** – Обратное преобразование Фурье (ОПФ), метод, который позволяет восстановить сигнал из его частотного представления [↑](#footnote-ref-3)
4. **Недообучение** – ситуация, когда модель недостаточно точно изучила тренировочные данные. [↑](#footnote-ref-4)
5. **Переобучение** – ситуация, когда модель слишком точно запомнила тренировочные данные и не обобщает их на ранее не используемые данные. [↑](#footnote-ref-5)
6. Для многоклассовой классификации используется микроусреднение, макроусреднение, взвешенное усреднение. Последнее используется в данной работе для метрик precision, recall и f1-score. Взвешенные метрики учитывают пропорции классов и используют средневзвешенные значения. [↑](#footnote-ref-6)
7. **L1-регуляризация** – это метод добавления штрафа на модуль весов в целевую функцию при обучении модели, чтобы стимулировать разреженность весов и выполнение отбора признаков, приводящего к уменьшению числа значимых признаков и повышению интерпретируемости модели. [↑](#footnote-ref-7)