Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (ТГУ)
Институт прикладной математики и компьютерных наук

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ОПОП
д-р техн. наук, профессор
_____ А.В. Замятин
«03» июня 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРА
(МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

МУЗЫКАЛЬНАЯ РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА

по направлению подготовки 01.04.02 Прикладная математика и информатика,
направленность (профиль) «Интеллектуальный анализ больших данных»

Березовский Константин Андреевич

Руководитель ВКР
д-р техн. наук, профессор
_____ А.В. Замятин
«28» мая 2023 г.

Автор работы
студент группы № 932128
_____ К. А. Березовский
«28» мая 2023 г.

Томск – 2023

Министерство науки и высшего образования Российской Федерации.
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ
Руководитель ОПОП
д-р техн. наук, профессор
_____ А.В. Замятин
подпись
« 10 » ноября 20 22 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы магистра обучающемуся

Березовскому Константину Андреевичу

Фамилия Имя Отчество обучающегося

по направлению подготовки 01.04.02 Прикладная математика и информатика, направленность (профиль) «Интеллектуальный анализ больших данных»

1 Тема выпускной квалификационной работы

Музыкальная рекомендательная система

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:

а) в учебный офис / деканат – 28.05.2023 б) в ГЭК – 07.06.2023

3 Исходные данные к работе:

Объект исследования	–	Музыкальная рекомендательная система, которая использует анализ контента (музыкальные характеристики) для предоставления персонализированных рекомендаций пользователям
Предмет исследования	–	Полезные признаки для МРС
	–	Методы машинного обучения для МРС
Цель исследования	–	Создание функциональной и эффективной системы музыкальных рекомендаций

Задачи:

1. Исследовать различные подходы к рекомендациям.
2. Изучить доступные наборы данных.
3. Выявить методы машинного обучения для системы рекомендаций.
4. Определить полезные признаки.

Методы исследования:

Поиск и анализ необходимой литературы

Создание моделей машинного обучения

Тестирование МРС

Организация или отрасль, по тематике которой выполняется работа, –

Национальный исследовательский Томский государственный университет

4 Краткое содержание работы

1. Выявление типов рекомендательных систем.
2. Выявление признаков для классификации музыки
3. Изучение моделей машинного обучения для создания МРС
4. Выбор наборов данных для обучения
5. Проведение экспериментов с выявлением лучшей модели и оценка результатов
6. Тестирование МРС

Научный руководитель выпускной квалификационной работы
д-р техн. наук, директор ИПМКН НИ ТГУ — *должность, место работы*
_____ / А.В. Замятин
подпись — *И.О. Фамилия*

Задание принял к исполнению
студент группы № 932128 — *должность, место работы*
_____ / К. А. Березовский
подпись — *И.О. Фамилия*

АННОТАЦИЯ

С развитием потоковых платформ в последние годы и увеличением потребления музыки встал вопрос о музыкальных рекомендациях. Музыкальные приложения стремятся улучшить свои системы рекомендаций, чтобы предложить своим пользователям наилучший опыт прослушивания и удержать их на своей платформе. Для этой цели появились две основные модели: коллаборативная фильтрация и модель на основе контента. В первом случае рекомендации основаны на вычислении сходства между пользователями и их музыкальными предпочтениями. Основная проблема этого метода называется "холодный старт" и означает, что система не будет хорошо работать с новыми элементами, будь то музыка или пользователи. Во втором случае рекомендации основаны на анализе самих музыкальных данных для рекомендации похожих треков. В данной диссертации был реализован второй метод.

Целью работы является создание функциональной и эффективной системы музыкальных рекомендаций. В долгосрочной перспективе целью является не только рекомендация существующих песен, но также создание песен, адаптированных к музыкальным предпочтениям пользователя.

Для достижения данной цели были выдвинуты следующие задачи:

1. Исследовать различные подходы к рекомендациям.

2. Изучить доступные наборы данных.

3. Выявить методы машинного обучения с целью построения подходящей системы рекомендаций.

4. Определить полезные признаки.

Анализ современных методов, основанных на контенте, показывает, что существует множество признаков, которые могут быть извлечены. Это включает низкоуровневые признаки, такие как временные (частота пересечения нуля), спектральные (спектральное убывание) или восприятие (громкость), требующие знаний в физике и обработке сигналов. Существуют также среднеуровневые признаки, понятные музыкальными экспертам (ритм, высота звука и т.д.). Наконец, есть высокоуровневые признаки, понятные всем (настроение, танцевальность и т.д.). Следует отметить, что также было идентифицировано множество моделей во время чтения научных статей.

В работе используя два набора данных GTZAN и FMA, мы собираемся сначала найти лучшую модель, сосредоточившись только на обучаемых моделях, а также на их гиперпараметрах, чтобы достичь релевантной рекомендации. С другой стороны, необходимо также определить наилучший набор признаков для характеристики музыки, избегая избыточной и навязчивой информации.

Работа будет структурирована следующим образом: вначале будет подробно описана техническая основа, необходимая для данного проекта. Будут представлены различные подходы, которые могут быть использованы для реализации систем рекомендаций, и будут описаны методы машинного обучения, которые будут исследованы в рамках этой диссертации. Также будут представлены способы оценки наших результатов.

В разделе “Реализация рекомендательной системы" будет описана проведенная работа. Здесь будет представлен выбранный набор данных и объяснено причины его выбора. Затем произойдет углубление в детали проведенных экспериментов. После этого будут подробно объяснены различные проведенные эксперименты.

В заключительном разделе "Результаты" будут выделены и проиллюстрированы основные полученные результаты. Количественные и качественные интерпретации этих результатов позволят нам получить единственную окончательную модель.

В разделе "Заключение" будут обобщены выводы по проделанной работе, описаны количественные и качественные результаты.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 5
1 Изучение предметной области 7
1.1 Обзор рекомендательных систем 7
1.1.1 Определение рекомендаций 7
1.1.2 Музыка отличается 7
1.1.3 Что такое хорошая рекомендация? 8
1.1.4 Извлечение информации о музыке: признаки и контекст 9
1.2 Типы рекомендательных систем 10
1.2.1 Коллаборативный подход 10
1.2.2 Подход, основанный на контенте 12
1.2.3 Подход, основанный на контексте 12
1.2.4 Гибридный подход 13
1.3 Модели машинного обучения для рекомендации на основе контента 13
1.3.1 Логистическая регрессия 14
1.3.2 Деревья решений 16
1.3.3 Случайный лес (Random Forest) 17
1.3.4 Бустинг: Adaboost 18
1.3.5 k-ближайших соседей 19
1.3.6 Метод опорных векторов 21
1.3.7 Наивный Байес 22
1.3.8 Нейронные сети 22
1.4 Признаки для контентной рекомендации 25
1.4.1 Низкоуровневые признаки 25
1.4.2 Признаки среднего уровня 34
1.4.3 Высокоуровневые признаки 35
1.5 Алгоритмы выбора признаков 36
1.5.1 Модель фильтрации 37
1.5.2 Модель обертки 38
2 Реализация рекомендательной системы 40

2.1 Выбранный подход 40
2.2 Наборы данных 40
2.2.1 Используемые наборы данных 41
2.2.2 Аугментация данных 44
2.3 Извлечение признаков 45
2.3.1 Предварительная обработка 45
2.3.2 Выбранные признаки 45
2.3.3 Модель-обертка для выбора признаков 46
2.4 Модели 46
2.4.1 Настройка гиперпараметров 46
2.5 Оценка 48
2.5.1 Оценка классификации с использованием меток 48
2.5.2 Оценка предсказаний с использованием матрицы путаниц 49
2.6 Алгоритм рекомендательной системы 50
3 Результаты 51
3.1 Предварительные результаты 51
3.1.1 Тесты на FMA 51
3.1.2 Тесты на GTZAN 55
3.2 Создание набора данных 58
3.3 Настройка гиперпараметров 60
3.3.1 Оптимизация логистической регрессии 60
3.3.2 Оптимизация дерева решений и случайного леса 61
3.3.3 Оптимизация Adaboost 62
3.3.4 Оптимизация метода ближайших соседей (k-nearest-neighbours) 63
3.3.5 Оптимизация метода опорных векторов (Support Vector Machine) 63
3.3.6 Нейронная сеть прямого распространения 64
3.3.7 Общие результаты после настройки гиперпараметров 64
3.4 Выбор признаков 65
3.4.1 Самые важные признаки 70
3.5 Аугментация данных 71

3.6 Финальные примеры рекомендаций 72
ЗАКЛЮЧЕНИЕ..... 75
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ..... 77

ВВЕДЕНИЕ

В 1979 году появилась первая система рекомендаций. Элейн Рич описала свою библиотечную систему Grundy [1]: она используется для рекомендации книг пользователям после небольшого интервью, в ходе которого пользователю сначала предлагается заполнить свое имя и фамилию, а затем, чтобы определить предпочтения пользователя и классифицировать их "стереотипами", Grundy просит пользователя описать себя несколькими ключевыми словами. После записи информации Grundy делает первое предложение, отображая краткое описание книги. Если предложение не нравится пользователю, Grundy задает вопросы, чтобы понять, в каком аспекте книги он ошибся, и предлагает новую. Однако ее использование ограничено, и Рич сталкивается с проблемами обобщения.

Системы рекомендаций, которые действительно начали развиваться в 1990-х годах, сильно продвинулись в последние годы, особенно с появлением машинного обучения и сетей. С одной стороны, все большее использование современной цифровой среды, характеризующейся избытком информации, позволило нам получить большие базы данных пользователей. С другой стороны, увеличение вычислительной мощности позволило обрабатывать эти данные, особенно благодаря машинному обучению, когда возможности человека перестали быть достаточными для исчерпывающего анализа такого объема информации.

В отличие от поисковых систем, которые получают запросы от пользователя с точными сведениями о том, что они ищут, система рекомендаций не получает прямого запроса от пользователя, а должна предложить новые возможности, изучая их предпочтения на основе их предыдущего поведения.

Торговые сайты, которые стремятся продать максимум товаров или услуг (путешествия, книги и т. д.) клиентам, должны быстро рекомендовать подходящие товары. Что касается сайтов, предлагающих потоковую передачу музыки и фильмов, их целью является удержание пользователей на своей платформе как можно дольше. Общим моментом является необходимость делать адекватные рекомендации. Последние достижения в этой области являются значительными, и такие рекомендации одинаково полезны как для компаний, максимизирующих свою прибыль, так и для клиентов, которые больше не ощущают перенасыщение количеством возможностей. Принятие решений становится проще, и хорошая рекомендация является значительным экономителем времени.

В 2006 году Netflix, который был онлайн-сервисом по аренде DVD, запустил челлендж Netflix с призовым фондом в 1 миллион долларов. Цель конкурса заключалась в создании алгоритма рекомендаций, способного превзойти существующий на 10% в тестах. Конкурс вызвал большой интерес как в научном сообществе, так и среди любителей кино. Приз был выигран спустя 3 года и подчеркнул несколько методов и направлений исследований для решения таких проблем.

Система рекомендаций будет определена в соответствии с определением Берка [2]: это система, способная предоставлять персонализированные рекомендации или направлять пользователя к интересным или полезным ресурсам (называемым элементами) в большом пространстве данных.

1 Изучение предметной области

1.1 Обзор рекомендательных систем

1.1.1 Определение рекомендаций

В данной диссертации основное внимание будет уделено рекомендательным системам. Рекомендательная система - это набор техник и сервисов, целью которых является предложение пользователям продуктов, которые им могут быть интересны... В настоящее время они реализованы на платформах распространения мультимедийного контента (Netflix, Yandex.Music, Spotify, ...), интернет-площадках продаж (Amazon, Ebay, ...), социальных сетях (VK, Facebook, ...), и т.д... Рекомендательные системы особенно полезны, когда количество пользователей и статей становится очень большим. Это связано с тем, что пользователи маловероятно знают всю полноту каталога, предлагаемого сервисом, и можно сказать, что практически невозможно сделать персонализированный рецепт для всех пользователей сервиса. Целью рекомендательной системы является проведение пользователей через огромное количество доступных данных, особенно на платформах электронной коммерции, фильтруя эти данные, чтобы автоматически предложить каждому потребителю товары, которые могут быть для него интересны.

1.1.2 Музыка отличается

Рекомендательные системы все больше используются в различных областях: отелей, путешествий, товаров. Но музыкальная сфера имеет некоторые особенности, которые следует учесть. [3] Первый фактор, который следует учесть, это продолжительность музыкальной композиции. Поскольку трек короткий, намного менее критично дать неправильную рекомендацию, чем, например, для фильма или книги. Пользователь также может быстро прослушать музыку, чтобы определить, подходит ли она его вкусу или нет. Вторая специфика заключается в количестве доступных треков, действительно выбор очень широк, оценивается, что как минимум десятки миллионов песен доступны в Интернете. Часто повторяющиеся рекомендации одной и той же музыки могут быть оценены положительно. В то время как в случае путешествий или фильмов пользователь ищет разнообразие, пользователю может понравиться слушать одну и ту же музыку снова и снова. Более того, возможно, что при первом прослушивании пользователь не был внимателен, так как прослушивание музыки часто сопровождается параллельной деятельностью (спорт, работа и

т. д.). Внимательное прослушивание требует качественного аудиооборудования, правильного настроения и времени, посвященного исключительно прослушиванию. Более того, довольно легко извлечь набор признаков из музыкальной композиции. Информация может быть извлечена с помощью обработки сигналов, благодаря музыкальным знаниям, благодаря текстам песен или просто с использованием отзывов пользователей. Старая музыка так же актуальна, как и новая: недавняя музыка, а также музыка нескольких десятилетий назад или классическая музыка могут быть так же приятными. Здесь важно правильно понять вкусы пользователя. Также следует учесть, что прослушивание музыки часто является пассивным: слушатель не обязательно внимательно ее слушает: в магазинах, барах, во время работы и т.д. Последний аспект, который отличает музыку от других объектов, которые могут быть рекомендованы, заключается в том, что музыка часто воспроизводится последовательно. Фактически, поскольку треки короткие, их часто объединяют в форме плейлиста.

1.1.3 Что такое хорошая рекомендация?

Учитывая эти особенности, теперь необходимо сделать адекватную рекомендацию. Естественно, основная цель - достичь высокого уровня точности, что означает предсказание музыки, которую пользователь полюбит и будет слушать. Чем больше пользователь доверяет рекомендательной системе и знает, как она работает, тем более эффективной она будет.

Успешная рекомендация предполагает баланс между эксплуатацией и исследованием: [4] С одной стороны, эксплуатация состоит в воспроизведении безопасной музыки (рекомендательная система знает, что эта музыка нравится пользователю). Это называется "lean-back" опытом и приносит краткосрочные результаты. С другой стороны, исследование заключается в воспроизведении новой музыки, делая новые открытия. Это называется "lean-in" опытом и приносит долгосрочные результаты. Если система правильно настроена, немного случайности может понравиться [5]. Это означает, что нам нужно найти подходящий баланс между новизной и знакомым, разнообразием и сходством, а также популярностью и персонализацией.

Наконец, прозрачность перед пользователями является ключевым моментом. Было доказано, что объяснение того, как работает алгоритм пользователю, улучшает их доверие и, следовательно, время, которое они проведут на платформе, во-первых, чтобы усовершенствовать свой профиль, и во-вторых, потому что они получат лучшие рекомендации. [4]

1.1.4 Извлечение информации о музыке: признаки и контекст

Основная цель извлечения информации о музыке заключается в получении наиболее значимой информации из различных представлений музыки (аудио, тексты песен, веб, метаданные, ...)

Извлеченные признаки могут быть разделены на четыре категории. Первая категория - это контент музыки [6], она объединяет три типа признаков. Техники обработки сигналов предоставляют нам доступ к низкоуровневым признакам, то есть признакам, понятным машине. Они могут быть временными (частота пересечения нуля) или спектральными (спектральный поток, спектральное убывание и т. д.) признаками. Музыкальные знания необходимы для извлечения признаков среднего уровня, таких как ритм, тональность, высота и т. д. Наконец, если предыдущие признаки понятны только машине или экспертам, то признаки высокого уровня доступны каждому, такие как танцевальность или живость, например.

Вторая категория - это контекст музыки. Целью является получение максимального количества информации (страны, связанные артисты, жанры) на основе метаданных благодаря веб-страницам, блогам, текстам песен, тегам и т. д. [6]

Используемые данные не связаны только с самой музыкой, но также могут быть связаны с пользователем. Прежде всего, есть свойства слушателя. У каждого есть вкусы и предпочтения, которые могут быть неявно получены (воспроизведения, плейлисты) или явно (оценки, отметки). [6]

Наконец, последняя категория - это контекст слушателя. Данные могут быть полезны, потому что в зависимости от настроения слушателя, его деятельности (спорт, работа и т. д.) желаемая музыка сильно варьируется.[6] Существуют разные методы получения информации о контексте пользователя: [4] Информацию можно получить явно, то есть напрямую у пользователя (с помощью форм, опросов, оценок и т. д.). Некоторую информацию также можно неявно вывести из сенсоров наших устройств (частота сердечных сокращений, интенсивность света, акселерометр, положение, погода и т. д.). Другой способ - выводить информацию на основе методов машинного обучения и статистики. Эти методы могут использоваться для получения выводов, например, на основе положения и скорости движения можно сделать вывод об активности.

1.2 Типы рекомендательных систем

Существуют три основных типа рекомендательных систем, которые позволяют создавать плейлисты с музыкой, адаптированной к пользователю: коллаборативная фильтрация, методы информационного поиска на основе контента и контекстная рекомендация. Также возможно сочетание предыдущих методов, которое называется гибридной системой. [7]

1.2.1 Коллаборативный подход

Этот метод рекомендаций основан на анализе поведения слушателей и поведения всех остальных пользователей платформы. Основное предположение здесь заключается в том, что мнения других пользователей могут быть использованы для предсказания предпочтений конкретного пользователя по отношению к элементу, который он еще не оценил: пользователю предлагаются рекомендации на основе пользователей, с которыми он имеет схожие вкусы. На протяжении многих лет мы просили наших друзей, семью и коллег рекомендовать нам музыку, рестораны, фильмы и т.д. И именно этот механизм пытаются воспроизвести здесь. Netflix был пионером этого метода (на основе звезд, присуждаемых другими пользователями), но он теперь широко используется, включая Spotify's , Yandex и др. [8]

Первая группа методов коллаборативной фильтрации называется подходом на основе памяти. Принцип заключается в хранении всех данных в матрице Пользователи/Песни. Это можно сделать благодаря неявной или явной обратной связи. В первом случае, если песня была прослушана хотя бы один раз, значение равно 1, в противном случае - 0. Во втором случае значение представляет собой количество звезд, если они доступны, иначе - 0.

Мы получаем большую матрицу. Чтобы ее уменьшить, Spotify пытается приблизить эту матрицу внутренним произведением двух других, более маленьких матриц. [9]

Благодаря факторизации матрицы, у нас теперь есть два типа векторов: вектор пользовательских предпочтений X и вектор признаков песни Y для каждого слушателя.

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} \vdots \\ X \\ \vdots \end{pmatrix} \times (\cdots \quad Y \quad \cdots) \tag{1}$$

Последний шаг - найти сходство между векторами, чтобы иметь возможность рекомендовать музыку слушателям. Для этого существуют два метода: [10]

- Сходство между пользователями : Сравнение вектора слушателя с векторами других пользователей для поиска тех, у кого схожие вкусы.
- Сходство между треками : Сравнение векторов треков для определения того, который наиболее близок к текущей прослушиваемой музыке.

Существует второй подход, называемый модельным: его целью является предсказание рейтинга пользователя для отсутствующих элементов с использованием моделей машинного обучения.

Главное преимущество коллаборативного подхода заключается в том, что нам не нужно анализировать и извлекать признаки из исходных файлов, поэтому нет необходимости иметь аудиофайлы или глубокие познания в музыке или физике. Более того, он вносит эффект неожиданности, который пользователь может получить, получив релевантную рекомендацию, которую они бы не нашли самостоятельно.

Существуют три основных недостатка. Первый называется проблемой холодного старта и включает две проблемы: проблему нового пользователя и проблему нового элемента. [11] Первая отражает отсутствие данных пользователя для составления релевантной рекомендации, а вторая отражает факт, что мы не знаем, кому рекомендовать новые элементы. Следующая проблема - масштабируемость, поскольку большое количество пользователей и элементов требует высоких вычислительных ресурсов. Последний недостаток - разреженность, поскольку количество элементов большое, каждый пользователь может оценить только небольшой поднабор из них. [11]

1.2.2 Подход, основанный на контенте

Метод рекомендаций, основанный на контенте, состоит в анализе содержания кандидатов на рекомендацию. Этот подход направлен на вывод предпочтений пользователя с целью рекомендовать ему композиции, содержание которых схоже с уже понравившимися ему композиций. Для этого метода не требуется обратная связь от слушателя, он основан только на звуковой схожести, которая выясняется из признаков, извлеченных из ранее прослушанных песен. [8] Этот метод основан на сходстве между различными треками. Для оценки сходства необходимо извлечь признаки, наилучшим образом описывающие музыку. Затем алгоритмы машинного обучения рекомендуют наиболее близкий трек к тем, которые уже понравились пользователю.

Таким образом, необходимо создать профили композиций на основе извлеченных признаков. Кроме того, этот метод требует профилей пользователей, основанных как на их предпочтениях, так и на их истории на платформе. Эти профили будут иметь следующую форму: список весов (которые показывают важность), соответствующих каждому выбранному признаку.

Основным преимуществом этого подхода является то, что неизвестная музыка имеет такую же вероятность быть рекомендованной, как и в настоящее время популярная или давно известная. Это позволяет также представить новых артистов с небольшим количеством просмотров. Кроме того, этот подход избегает проблемы "холодного старта", особенно связанной с новыми элементами: когда новые элементы вводятся в систему, их можно непосредственно рекомендовать, не требуя времени на интеграцию, как это происходит в рекомендательных системах, основанных на коллаборативной фильтрации.

Отрицательным аспектом является то, что этот метод ограничивает разнообразие в рекомендациях, склонен к излишней специализации. Кроме того, интеграция нового пользователя не может быть мгновенной, ему необходимо прослушать и оценить определенное количество песен, прежде чем он сможет получить рекомендации. Это называется проблемой "холодного старта" пользователя.

1.2.3 Подход, основанный на контексте

Исследования [12] показали, что настроение, активность или даже местоположение человека влияют на музыку, которую они хотят слушать. Мы слушаем музыку в определенный момент, в определенном эмоциональном состоянии и установленных

обстоятельствах (вечеринка, работа и т. д.). И эти предрасположенности будут играть решающую роль в том, как мы воспринимаем музыку. Хотя существует множество приложений [12] такого типа рекомендаций, например, туристические приложения с адаптивными фоновыми песнями, на эту тему все еще мало конкретных приложений.

Многие преграды все еще мешают исследованиям в этой области. Действительно, характер данных, которые следует учитывать, сильно варьирует и зависит от окружающей среды (время, место, погода, культура и т. д.) или самих пользователей (скорость движения, эмоции, пульс и т. д.). Еще более значительной проблемой является недостаток доступных данных для исследований. В реальном мире их также не всегда легко получить, так как пользователи не всегда хотят передавать много информации с датчиков своего мобильного телефона.

1.2.4 Гибридный подход

Также возможно объединить предыдущие дополняющие методы для создания рекомендательной системы. Гибридный подход также может основываться на всех остальных менее известных методах, таких как рекомендации на основе местоположения. Этот метод может смягчить проблемы холодного старта и разреженности. Можно создать несколько реализаций: во-первых, системы рекомендаций могут быть объединены в одну. Также можно сохранить несколько систем отдельно и назначить им веса или возможность переключения между ними по желанию. Наконец, можно извлечь результаты из одной системы и использовать их в качестве входных данных для следующей.

1.3 Модели машинного обучения для рекомендации на основе контента

Во время этапа изучения существующих исследований было выяснено, что для рекомендации можно использовать различные модели. Эти модели будут протестированы в данной диссертации и, следовательно, представлены в данном разделе.

Выбранные модели являются моделями машинного обучения с учителем. Машинное обучение - это тип искусственного интеллекта, где алгоритм автоматически изменяет свое поведение с целью улучшить свою производительность на задаче на основе набора данных. Этот процесс называется обучением, поскольку алгоритм оптимизируется на основе наблюдений и пытается извлечь статистические закономерности из них. В обучении с учителем целью алгоритма является предсказание явного и известного значения цели на основе обучающих данных. Два наиболее распространенных типа целей - это либо

непрерывные значения, где $t \in R$(задача регрессии), либо дискретные классы, где $t \in 1, ..., N_C$ для задачи с N_C классами (задача классификации).

1.3.1 Логистическая регрессия

Логистическая регрессия зарекомендовала себя в области классификации музыки, хотя она не является самым эффективным методом [13], она обладает преимуществом быстроты. Логистическая регрессия часто используется для многоклассовой классификации. Здесь речь идет о нахождении оптимальной границы решений для разделения различных классов. [14] Самый простой случай - это когда есть только два класса (0 и 1), в этом случае, так как логистическая регрессия является линейной моделью, функция оценки может быть записана следующим образом:

$$(X^{(i)}) = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_n x_n \quad (2)$$

Где $(X^{(i)})$ - наблюдение (из обучающего или тестового набора) в виде вектора $x_1, x_2, \ldots, x_n$;

x_i - одна из значимых особенностей предиктивной модели;

θ_0 - постоянная, называемая смещением (bias);

θ_i - веса (связанные с особенностями), которые должны быть вычислены.

Это можно более компактно записать, обозначив Θ вектором, содержащим компоненты $\theta_0, \theta_1, \ldots, \theta_n$ и X вектором, содержащим $x_1, x_2, \ldots, x_n$:

$$S(X) = \Theta X \quad (3)$$

Затем цель состоит в поиске коэффициентов $\theta_0, \theta_1, \ldots, \theta_n$ таких, что:

• $S(X^{(i)}) > 0$, если образец принадлежит положительному классу (метка 1)

• $S(X^{(i)}) < 0$, если образец принадлежит отрицательному классу (метка 0)

Затем к функции оценки применяется сигмоидная функция $\mathrm{sigmoid}(x) = e^{\frac{1}{1+e^X}}$ (рисунок 1), которая позволяет получить значения между 0 и 1. Таким образом, общая гипотезная функция для логистической регрессии имеет вид:

$$H(X) = \mathrm{Sigmoid}(S(X)) = \frac{1}{1+e^{\Theta X}} \quad (4)$$

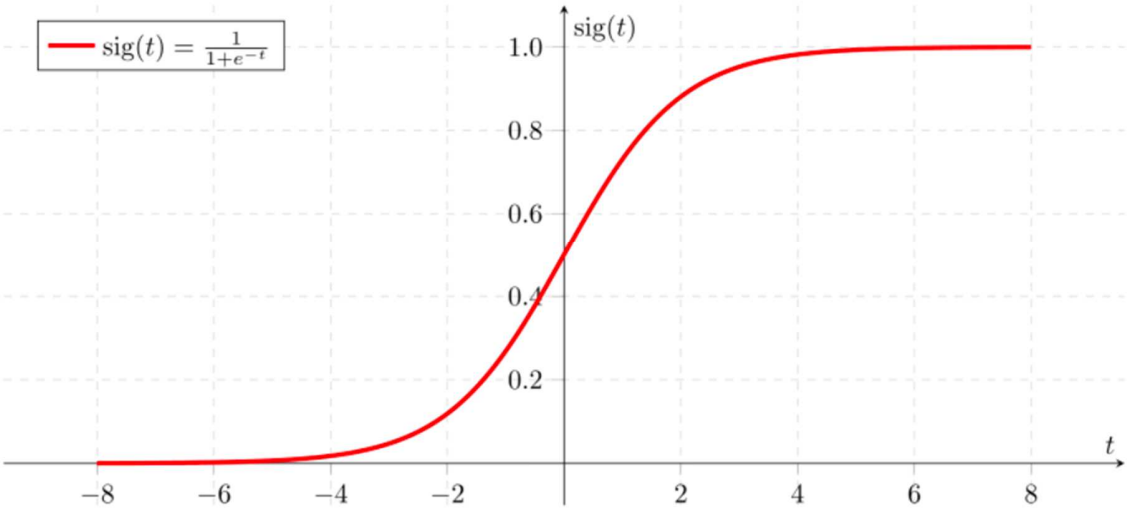


Рисунок 1- Сигмоидальная функция

Классификация музыки по жанрам является случаем многоклассовой классификации (рисунок 2), и алгоритм, обычно используемый этой моделью, - это "Один против всех" ("One-Versus-All"): он заключается в разделении проблемы на несколько подзадач бинарной классификации. Сначала класс 1 отделяется от всех остальных, затем класс 2 от всех остальных и так далее.

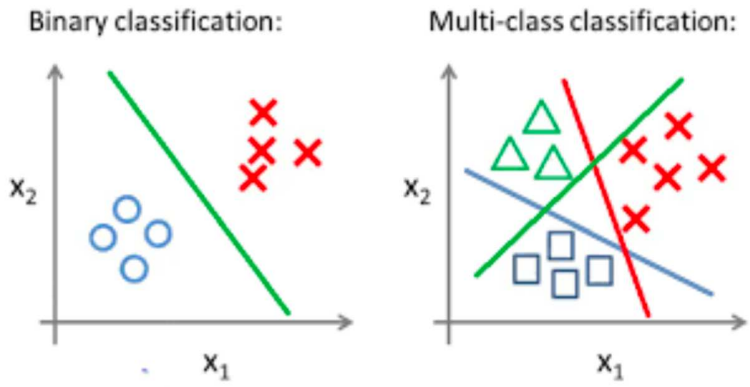


Рисунок 2 - Логистическая регрессия для многоклассовой классификации [16]

Для предотвращения переобучения можно использовать методы регуляризации l1 и l2 для настройки значений весов w_i:

• Lasso (Least Absolute Shrinkage and Selection Operator) Regression (l1): он состоит в добавлении регуляризационного члена к функции потерь:

$$L(x, y) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \sum_{i=1}^{n} |w_i| \quad (5)$$

Lasso имеет несколько решений и склонен уменьшать веса менее значимых признаков до нуля, поэтому он особенно эффективен, когда у вас большое количество признаков, и вы хотите выбрать наиболее важные.

• Гребневая регрессия (Ridge Regression) (l2):

$$L(x, y) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \sum_{i=1}^{n} {w_i}^2 \quad (6)$$

Гребневая регрессия имеет только одно решение, она не уменьшает количество признаков, а скорее уменьшает влияние каждого из них.

1.3.2 Деревья решений

Эта статья [17] показывает, что в классификации латинской музыки решающие деревья могут быть эффективными. В дереве решений [18] можно классифицировать элементы, последовательно разделяя их по параметрам. Дерево решений состоит из трех основных элементов: узлы - это проверки, выполняемые на атрибутах, ребра - это результаты проверок, они соединяют узлы одного уровня с следующими, и листья - это последние узлы дерева, они представляют конечные классы. Существуют два типа деревьев решений: деревья регрессии и деревья классификации. Первые могут принимать непрерывные значения целей, например, для прогнозирования цены дома, а вторые состоят из вопросов "Да/Нет", и их цели являются дискретными/категориальными. Это итерационный процесс, состоящий в разделении данных на части и их распределении по каждой из ветвей.

Алгоритм, используемый для обучения деревьев решений в классификации, называется "Разделяй и властвуй". Он направлен на разбиение набора данных на подмножества на каждом узле. Принцип состоит в выборе теста для первого узла, называемого корневым узлом, который позволяет разделить набор на две подчасти, максимизируя информационный выигрыш (Gain of Information) или критерий Джини (Gini

Impurity). Затем это действие повторяется рекурсивно до тех пор, пока не будет ветвь, в которой все экземпляры относятся к одному классу. Чтобы избежать переобучения, можно установить ограничение на глубину дерева.

Для определения информационного выигрыша и критерия Джини необходимо определить понятие энтропии.

Энтропия - это мера неоднородности, беспорядка или неопределенности в наборе примеров. Для набора данных с C классами и с pi - пропорцией элементов класса i в наборе данных:

$$E = -\sum_{i}^{C} p_i \log_2 p_i \tag{7}$$

Информационный выигрыш (Information Gain, IG) измеряет, насколько "информативной" является характеристика относительно класса. Его можно вычислить следующим образом:

$$IG = \text{Entropy(parent)} - [\text{weightedaverage}] * \text{Entropy[children]} \tag{8}$$

Критерий Джини - это вероятность неправильной классификации случайно выбранного элемента в наборе данных, если бы он был случайным образом помечен в соответствии с распределением классов в наборе данных:

$$G = \sum_{i}^{C} p_i (1 - p_i) \tag{9}$$

1.3.3 Случайный лес (Random Forest)

Случайный лес - это метод ансамблевого обучения. Благодаря своей разнообразности, независимости, децентрализации и агрегации комбинация нескольких классификаторов дает лучший результат. Для достижения оптимальных результатов классификаторы с высокой дисперсией и низким смещением должны быть сгруппированы вместе, чтобы снизить общую дисперсию, сохраняя при этом низкое смещение. В случае деревьев решений высокая

дисперсия означает границы, которые сильно зависят от обучающего набора данных, низкое смещение означает границы, которые в среднем близки к истинной границе.

Случайный лес является особой разновидностью метода “Бэггинг” (бутстрэп-агрегирование). Цель таких методов состоит в снижении дисперсии, внесенной отдельным деревом, и, таким образом, сокращении ошибки прогнозирования. Для предсказания результата алгоритм Случайного леса усредняет прогнозы нескольких независимых моделей (в контексте классификации предсказывает наиболее частую категорию). Для создания этих моделей создаются несколько повторных выборок обучающего набора путем выбора с возвращением, таким образом, все наши модели могут быть обучены параллельно. Этот алгоритм не только использует метод "Бэггинга", но также случайным образом выбирает признаки на каждом узле.

1.3.4 Бустинг: Adaboost

Adaboost является одним из наиболее используемых алгоритмов в бустинге, который также оптимизирует деревья решений. Берутся несколько бинарных классификаторов, каждый из которых основан на одном признаке, что дает нам набор слабых классификаторов. Принцип Adaboost основан на предположении, что набор слабых классификаторов может дать сильный классификатор (рисунок 3). Принцип заключается в циклическом обучении классификаторов с взвешенными образцами, и когда образец неправильно классифицируется, его вес увеличивается. Конкретные шаги следующие:

1. Инициализация весов: в начале они равномерны.

2. Обучение одного дерева решений.

3. Вычисление взвешенной ошибки: вычисление количества элементов (с учетом их веса), которые были неправильно классифицированы.

4. Вычисление веса дерева решений в зависимости от его ошибки.

$$W\_tree = learning\_rate * \log(\frac{1-e}{e}) \quad (10)$$

5. Обновление весов неправильно классифицированных элементов.

$$W\_item\_new = W\_item\_old * \exp(W\_tree) \quad (11)$$

6. Повторение шагов с 2 по 5 для каждого дерева.

7. Окончательное решение.

$$Pred\_Final = \sum_{t \in trees} W\_t * Pred(t) \quad (12)$$

Это означает, что каждая модель обучается последовательно и учится на ошибках, совершенных предыдущими моделями. В то время как Случайный лес стремится уменьшить дисперсию, а не смещение, Adaboost стремится уменьшить смещение, но не дисперсию.

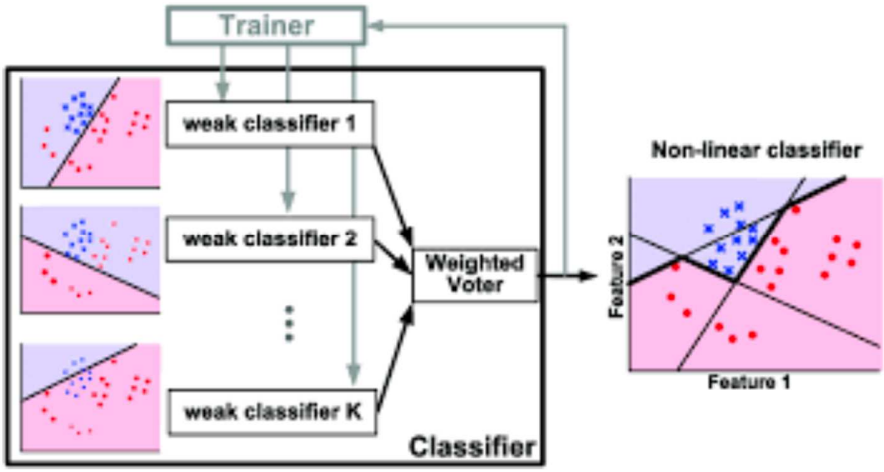


Рисунок 3 - Классификатор Адабуст [19]

1.3.5 k-ближайших соседей

Метод k-ближайших соседей (k-Nearest Neighbours, k-NN) делает прогноз на основе всего набора данных. Когда мы хотим предсказать новое значение, алгоритм будет искать K наиболее близких экземпляров из набора данных. Затем, он будет использовать выходные значения ближайших K соседей для вычисления значения переменной, которую необходимо предсказать. [20]

Параметр K должен быть определен, требуется достаточно высокое значение, чтобы избежать недообучения. Однако, если значение слишком высокое, возникает риск переобучения. Необходимо найти компромисс. Для каждого нового элемента i первым шагом является вычисление его расстояния d до всех остальных значений набора данных и выбор K элементов, у которых расстояние минимально. [21] Затем оптимальное значение K

используется для предсказания: в случае регрессии следующим шагом является вычисление среднего (или медианного) значения выходных значений выбранных K соседей. В случае классификации это означает определение наиболее представленного класса среди K соседей.

Для определения расстояния доступны различные формулы, при усло-вии, что они удовлетворяют критериям неотрицательности, идентичности, симметрии и неравенства треугольника. Наиболее распространенные из них:

• Расстояние Хэмминга: для двух строк одинаковой длины, это количество позиций, в которых символы отличаются.

• Манхэттенское расстояние:

$$d_m(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{13}$$

• Евклидово расстояние:

$$d_e(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{14}$$

• Расстояние Минковского:

$$d_n(x,y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}, p \geqslant 1 \tag{15}$$

• Расстояние Чебышева:

$$d_t(x,y) = \max_{i=1}^{n} |x_i - y_i| \tag{16}$$

Этот метод имеет преимущество в том, что он прост, понятен и при этом дает надежные результаты, но он чувствителен к избыточным или бесполезным признакам. [20] k-NN, похоже, дает хорошие результаты для классификации музыки при использовании функций MFCC (определенных в подразделе 2.4). [22]

1.3.6 Метод опорных векторов

Метод опорных векторов (Support Vector Machine, SVM) широко используется для классификации музыки и дает хорошие результаты, но используемые признаки часто ограничиваются MFCC и некоторыми ритмическими признаками. Этот метод использует несколько последовательных SVM для улучшения результатов. [23] Основной принцип заключается в разделении двух групп данных с максимизацией зазора вокруг границы (расстояния между двумя классами). Он основан на идее, что практически все становится линейно разделимым, когда представлено в пространствах высокой размерности. Поэтому два шага: преобразование ввода в подходящее пространство высокой размерности, а затем нахождение гиперплоскости, разделяющей данные, при этом максимизируя зазоры. На практике для получения выгоды от пространства высокой размерности без фактического представления данных используются ядерные функции. Единственной операцией, выполняемой в пространстве высокой размерности, является вычисление скалярных произведений между парами элементов. Обычно используются следующие типы ядер:

• Линейное:

$$K(\vec{x}, \vec{y}) = \vec{x} \cdot \vec{y} \tag{17}$$

• Полиномиальное:

$$K(\vec{x}, \vec{y}) = (\vec{x}^{t}\vec{y} + 1)^{p} \tag{18}$$

• Радиально-базисное:

$$K(\vec{x}, \vec{y}) = e^{-\frac{1}{2\rho^2}\|\vec{x}-\vec{y}\|^2} \tag{19}$$

В некоторых случаях можно принять некоторое количество выбросов, которые будут находиться в пределах зазора, чтобы разделить данные. Хотя SVM были созданы для решения бинарных задач, существуют два способа адаптировать их к многоклассовым задачам:

• One-versus-all: заключается в преобразовании задачи классификации C классов с использованием единственного разделителя в C задач бинарной классификации. Рейтинг определяется классификатором, наилучшим образом соответствующим данным.

• One-versus-one: на этот раз берутся $\frac{k(k-1)}{2}$ бинарных классификаторов, идея состоит в том, чтобы для каждого класса C_i сравнить его со всеми остальными классами $C_j \neq C_i$. Окончательный рейтинг определяется голосованием большинства.

1.3.7 Наивный Байес

Наивный байесовский (Naive Bayes) классификатор основан на вероятностном подходе с использованием теоремы Байеса. Он дает вероятность того, что элемент принадлежит классу C_i, при условии, что элемент имеет набор признаков $x = (x_1, \ldots, x_F)$.

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{P(x)} = \frac{P(x|C_i)P(C_i)}{\sum_j P(x|C_j)P(C_j)} \quad (20)$$

Где $P(C_i)$ - называется априорной вероятностью ;

$P(C_i|x)$ - называется апостериорной вероятностью,;

$P(x|C_i)$ - называется правдоподобием;

$P(x)$ -называется доказательством.

Часто вычисление результата основано на нескольких переменных. Поскольку вычисление сложно, одним из типов классификаторов, которые часто используются, является наивный байесовский классификатор [24]: предполагается, что эти переменные являются независимыми. Это сильное предположение, поэтому используется слово “наивный".

1.3.8 Нейронные сети

Нейронные сети всех типов также широко применяются, включая прямопроходные нейронные сети. Результаты, основанные только на высокоуровневых признаках, дают общую точность 85% [27]. Нейронная сеть (рисунок 4) - это система, архитектура которой вдохновлена функционированием биологических нейронов, однако сейчас она все больше и больше приближается к математическим и статистическим методам.

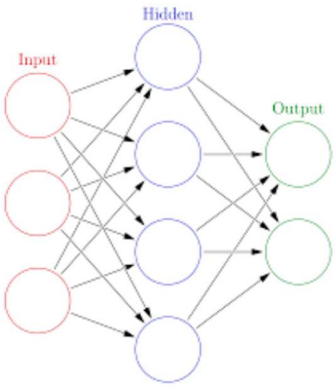


Рисунок 4 - Иллюстрация искусственной нейронной сети [28]

Формальный нейрон является элементарной единицей искусственной нейронной сети. Получая сигналы от других нейронов в сети, формальный нейрон отвечает, производя выходной сигнал, который передается другим нейронам в сети. Полученный сигнал является взвешенной суммой сигналов от разных нейронов. Окончательный выходной сигнал является функцией этой взвешенной суммы:

$$y_i = f(\sum_{i=1}^{N} w_{i,j} x_i) \tag{21}$$

Где y_j - выход формального нейрона j ;

x_i для $i \in \{1, \ldots, N\}$ - сигналы, полученные нейроном j от нейронов i;

$w_{i,j}$ - веса между связями нейронов i и j;

f-функция активации, определяет выходное значение. Обычно используются функции идентичности, сигмоиды или гиперболического тангенса.

Для многоклассовой классификации самой простой нейронной сетью является многослойный перцептрон (Multi-Layer Perceptron, MLP) [29]. Это сеть, содержащая несколько слоев (и на каждом из них несколько единиц), полностью связанных между собой. Метод обучения называется градиентным распространением ошибки и используется для нахождения значений весов для каждого нейрона, наиболее важных для последующей классификации. Существуют три типа нейронов: входные ячейки, связанные с данными (по одной для каждого входного признака), выходные нейроны, каждый из которых связан с определенным классом, и скрытые нейроны, находящиеся в промежуточных слоях.

При использовании слишком глубоких нейронных сетей возникают различные проблемы. Первая проблема связана с временем и вычислительной мощностью, которые очень быстро становятся огромными. Алгоритм обучения также имеет проблемы с правильной работой, поскольку часто возникают проблемы с взрывным или исчезающим градиентом.

Существуют различные способы, называемые методами регуляризации, для борьбы с переобучением:

• Dropout: заключается в деактивации определенного процента единиц для конкретного слоя во время обучения (рисунок 5). Более конкретно, на каждом этапе обучения нейроны либо сохраняются с вероятностью p, либо исключаются с вероятностью 1-p. Это улучшает обобщение, поскольку заставляет слой учить одинаковое понятие с использованием разных нейронов. Этот метод часто применяется на полностью связанных слоях.

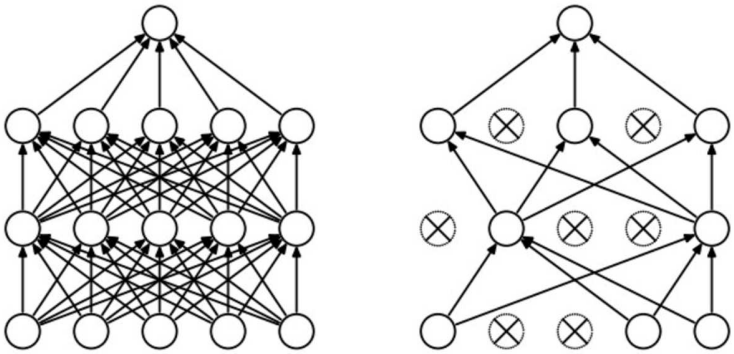


Рисунок 5 - Нейронная сеть БЕЗ / С Dropout [30]

• Преждевременная остановка: идея заключается в остановке обучения, когда система начинает переобучаться, то есть когда точность на тестовом наборе начинает уменьшаться (рисунок 6). Для этого необходимо создать валидационный набор данных, который позволяет тестировать модель на каждой эпохе и, следовательно, останавливать обучение, как только точность на валидационном наборе данных начинает уменьшаться и появляется переобучение.

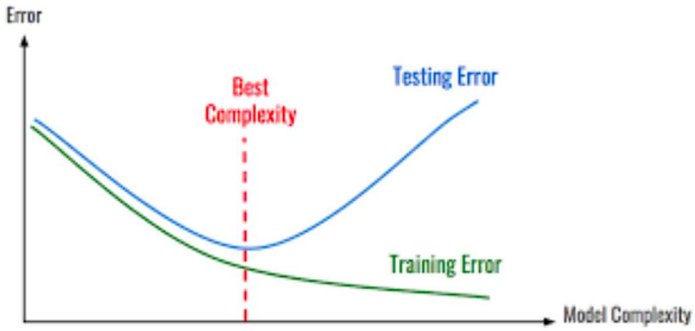


Рисунок 6 – Переобучение [31]

1.4 Признаки для контентной рекомендации

Признаки, упомянутые в статьях, прочитанных для выполнения данной диссертации, и которые могут быть извлечены из музыки, являются многочисленными и могут быть классифицированы в соответствии с их уровнем (низкий, средний, высокий). Существует несколько представлений музыки. С физической точки зрения звук является волной, то есть колебанием давления, которое обычно передается через окружающий воздух. Звук представляет собой суперпозицию звуковых волн различных частот с различными характеристиками, такими как амплитуда и фаза. Главным образом именно из этого представления можно извлечь признаки. В то время как некоторые признаки используют сигнал во временной области, другие сосредоточены на его частотной форме. Фактически, дискретное преобразование Фурье (ДПФ) может быть использовано для разложения цифрового временного сигнала на его синусоидальные компоненты и перевода его в частотную область.

1.4.1 Низкоуровневые признаки

Низкоуровневые признаки - это те признаки, которые могут быть вычислены непосредственно из исходного аудиофайла с использованием статистических, сигнально-обрабатывающих и математических методов. Низкоуровневые признаки могут быть сгруппированы по своей природе: временные, спектральные, энергетические или восприятийные. Аудиосигнал постоянно меняется, поэтому первый шаг заключается в разделении сигнала на короткие фреймы. Это позволяет сделать предположение, что сигнал статистически стационарен внутри каждого фрейма. Обычно сигнал разделяется на фрагменты продолжительностью 20-40 мс. Если фрагменты слишком короткие, мы не

получим достаточно длинный сегмент для надежного результата, а если слишком длинные, сигнал слишком сильно меняется.

В начале внимание уделяется временным признакам.

• Частота пересечения нуля (Zero-crossing rate). Это соответствует количеству пересечений сигнала с нулевой осью за заданный временной фрейм (который представляет собой короткий промежуток времени) в сигнале. Высокое значение характерно для шумного звука, а низкое значение указывает на периодический сигнал. [32] Он используется в основном в информационном поиске в музыке для выявления шума и ударных звуков. [7] Поскольку это значение имеет тенденцию быть выше для ударных звуков, оно помогает, например, отличить рок от металла. Может быть вычислено для фрейма t, где K - размер фрейма, следующим образом: [7]

$$\mathrm{CR_t} = \frac{1}{2} \sum_{\mathrm{k=t\cdot K}}^{\mathrm{(t+1)\cdot K-1}} |\mathrm{sign(s(k))} - \mathrm{sign(s(k+1))}| \tag{22}$$

Где

$$\mathrm{sign(s(k))} = \begin{cases} 1 & \text{при} \mathrm{s(k)} \geqslant 0 \\ -1 & \text{при} \mathrm{s(k)} < 0 \end{cases}$$

• Окружающая амплитуда (Amplitude envelope): Вычисляется как максимальная амплитуда среди всех сэмплов для фрейма t: [7]

$$\mathrm{AE_t} = \max_{\mathrm{k=t\cdot K}}^{\mathrm{(t+1)\cdot K-1}} \mathrm{s(k)} \tag{23}$$

• Среднеквадратическая энергия (Root-mean-square energy). Этот признак коррелирует с восприятием интенсивности звука, поэтому его можно использовать для оценки громкости. Низкая энергия особенно характерна для классической музыки. [7] Для одного фрейма t:

$$\mathrm{RMS_t} = \sqrt{\frac{1}{\mathrm{K}} \cdot \sum_{\mathrm{k=t\cdot K}}^{\mathrm{(t+1)\cdot K-1}} \mathrm{s(k)^2}} \tag{24}$$

Спектральные признаки определяются следующим образом:

• Спектральный центроид (Spectral centroid). Он указывает на расположение центра масс/барицентра спектра и представляет диапазон, где находится большая часть энергии. [33] Вычисляется как взвешенное среднее частот в звуке. Низкое значение спектрального центроида обычно соответствует классической музыке, особенно той, в которой присутствует только фортепиано. У других видов музыки спектральные центроиды часто меняются. [7] Для его вычисления спектр рассматривается как распределение: значения - это частоты, а вероятности их наблюдения нормализуются по амплитуде. [32]

$$\mu_1 = \frac{\sum_{k=b_1}^{b_2} f_k s_k}{\sum_{k=b_1}^{b_2} f_k} \tag{25}$$

Где f_k - частота в Гц, соответствующем интервалу k;

s_k - спектральное значение в интервале k;

b_1, b_2 - ребра полосы в интервалах, по которым можно вычислить спектральный центроид.

• Спектральное распространение (Spectral spread). Может быть определено как дисперсия распределения и показывает, насколько распределение спектра распределено вокруг его среднего значения. [32]

$$\mu_2 = \sqrt{\frac{\sum_{k=b_1}^{b_2} (f_k - \mu_1)^2 s_k}{\sum_{k=b_1}^{b_2} s_k}} \tag{26}$$

Где f_k - частота в Гц, соответствующем интервалу k;

s_k - спектральное значение в интервале k;

b_1, b_2 - ребра полосы в интервалах, по которым можно вычислить спектральное распространение;

μ_1 - спектральный центроид.

Спектральное распространение представляет “мгновенную полосу пропускания” спектра. Это используется в качестве индикации относительно преобладания тона.

• Спектральная скошенность (Spectral skewness). Показывает, насколько асимметрично распределение вокруг его среднего значения. [32] Значение 0 характеризует симметричное распределение, более высокое значение указывает на концентрацию энергии слева, а более низкое значение - на более высокую концентрацию энергии справа. Вычисляется с третьего момента порядка.

$$\mu_3 = \frac{\sum_{k=b_1}^{b_2}(f_k-\mu_1)^3 s_k}{(\mu_2)^3\sum_{k=b_1}^{b_2} s_k} \quad (27)$$

Где f_k - частота в Гц, соответствующем интервалу k;

s_k - спектральное значение в интервале k;

b_1, b_2 - ребра полосы в интервалах, по которым можно вычислить спектральную скошенность;

μ_1 - спектральный центроид;

μ_2 - спектральное распространение.

• Спектральный эксцесс (Spectral kurtosis). Он показывает, насколько плоским является распределение вокруг его среднего значения. [32] Для нормального распределения значение равно 3, более высокое значение означает более острый пик, более низкое значение - более плоское распределение. Вычисляется с четвертого момента порядка.

$$\mu_3 = \frac{\sum_{k=b_1}^{b_2}(f_k-\mu_1)^4 s_k}{(\mu_2)^4\sum_{k=b_1}^{b_2} s_k} \quad (28)$$

Где f_k - частота в Гц, соответствующем интервалу k;

s_k - спектральное значение в интервале k;

b_1, b_2 - ребра полосы в интервалах, по которым можно вычислить спектральный эксцесс;

μ_1 - спектральный центроид;

μ_2 - спектральное распространение.

• Частота спада спектра (Spectral roll-off frequency). Она соответствует значению частоты f_c, при котором процент (например, 95%) энергии сигнала содержится ниже этого значения. [32] sr / 2 - частота Найквиста:

$$\sum_{0}^{f_c} a^2(f) = 0.95 \sum_{0}^{sr/2} a^2(f) \tag{29}$$

Где a - амплитуды спектральных коэффициентов.

Амплитуды вычисляются путем взятия модуля комплексного значения каждого спектрального коэффициента. Это делается для того, чтобы извлечь информацию о вкладе каждой частоты в составляющих звукового сигнала.

• Отношение энергии в полосе частот (Band energy Ratio). Этот показатель измеряет степень доминирования низких частот над высокими. Рассчитывается путем выбора предельного значения, называемого разделительной частотой F. [7]

$$BER_t = \frac{\sum_{n=1}^{F-1} a^2(f)}{\sum_{n=F}^{N} a^2(f)} \tag{30}$$

• Спектральный поток (Spectral Flux). Он отображает изменение мощности между двумя последовательными фреймами.

$$F_t = \sum_{n=1}^{N} (s_k(t) - s_k(t-1))^2 \tag{31}$$

• Спектральный наклон (Spectral Slope) - это показатель объема уменьшения спектра: [32]

$$slope = \frac{\sum_{k=b_1}^{b_2} (f_k - \mu_f)(s_k - \mu_s)}{\sum_{k=b_1}^{b_2} (f_k - \mu_f)^2} \tag{32}$$

Где f_k - частота в Гц, соответствующем интервалу k;

μ_f - средняя частота;

s_k - спектральное значение в интервале k;

b_1, b_2 - ребра полосы в интервалах, по которым можно вычислить спектральный наклон;

μ_s - среднее спектральное значение.

• Спектральное убывание (Spectral Decrease). Представляет объем уменьшение спектра, при_наклонов более низких частот: [32]

$$\text{decrease} = \frac{\sum_{k=b_1+1}^{b_2} \frac{s_k - s_{b_1}}{k-1}}{\sum_{k=b_1+1}^{b_2} s_k} \tag{33}$$

Где s_k - спектральное значение в интервале k;

b_1, b_2 - ребра полосы в интервалах, по которым можно вычислить спектральное уменьшение.

• Спектральная плоскость (Spectral Flatness). Плоскость показывает, насколько близким является звук к белому шуму. Плоский спектр мощности (высокое значение плоскости) соответствует белому шуму. Выражается как отношение геометрического среднего к арифметическому среднему спектра мощности: [32]

$$\text{flatness} = \frac{(\prod_{k=b_1}^{b_2} s_k)^{\frac{1}{b_2-b_1}}}{\frac{1}{b_2-b_1}\sum_{k=b_1}^{b_2} s_k} \tag{34}$$

Где s_k - спектральное значение в интервале k;

b_1, b_2 - ребра полосы в интервалах, по которым можно вычислить спектральную плоскость.

• Мел частотные кепстральные коэффициенты, МЧКК (Mel Frequency Cepstral Coefficient, MFCC). Они были введены для распознавания речи и дикторов и оказались эффективными для извлечения спектра мощности аудиосигнала. Искусственные звуки фильтруются формой голосового аппарата (рот, язык, зубы и т. д.). Поэтому вопрос заключается в точном определении формы звука, что должно дать нам точное представление о производимом явлении и, следовательно, о способе его восприятия. Различные исследования показали, что Мел-частотные кепстры также могут быть полезны в области сходства музыки. [34] Алгоритм нахождения МЧКК представлен на рисунке 7.

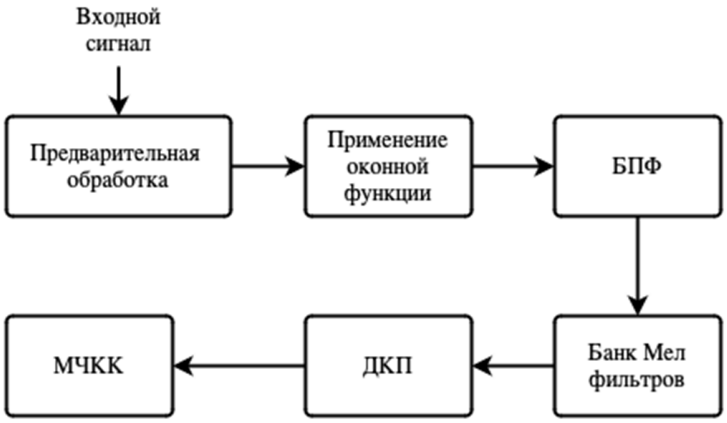


Рисунок 7 - Вычисление МЧКК [35]

1. Сначала проводится предварительная обработка сигнала.

2. Затем применяется оконная функция Хэмминга к каждому фрейму, чтобы уменьшить эффекты края.

3. Следующий шаг - просто преобразовать сигнал в частотную область. Для этого мы используем метод быстрого преобразования Фурье (БПФ, FFT), чтобы получить желаемую спектрограмму.

4. Использование банка Мел фильтров обусловлено работой вестибулярного аппарата человека, который имеет свойство вибрировать в соответствии с частотой воспринимаемого звука. Более точно, в зависимости от точного расположения вибрирующего вестибулярного аппарата (обнаруживаемого маленькими волосками), нервы передают в мозг информацию о присутствующих частотах. Поскольку вестибулярный аппарат не может различать разницу между близкими частотами, сегменты суммируются, чтобы определить количество энергии в каждом частотном диапазоне. Для этого мы используем банк Мел фильтров, где первый фильтр очень узкий и указывает на концентрированную энергию около 0 Гц. С увеличением частот фильтры становятся шире, поскольку вариации имеют меньшее значение.

5. Заключительным шагом является вычисление дискретного косинусного преобразования (ДКП) фильтров. Фильтры перекрываются, поэтому цель этого шага - декоррелировать энергии друг от друга.

6. Наконец, так как люди могут более легко различать малые изменения в высоте звука в низких частотах, чем в высоких частотах, масштаб Мел более подходящий. Обычно для каждого фрейма сохраняются 13 коэффициентов.

Частоты в Мел: [7]

$$M(f) = 1125 \cdot \ln(1 + \frac{f}{700}) \quad (35)$$

Учитывается также энергия сигнала:

• Глобальная энергия (Global energy): Показывает оценку сигнальной мощности в заданный момент времени. [32]

• Гармоническая энергия (Harmonic energy): Дает оценку мощности гармонической составляющей в заданный момент времени. [32]

• Энергия шума (Noise energy): Дает оценку мощности шумовой составляющей в заданный момент времени. [32]

Эта последняя часть определяет психоакустические признаки (восприятийные). Цель этих характеристик заключается в характеризации и моделировании слуховой системы. Они позволяют оценить значения, воспринимаемые людьми, и предсказать дискомфорт или раздражение, которые могут вызвать определенные звуки.

• Громкость (Loudness) - это дескриптор качества звука, который является воспринимаемыми и субъективными метриками, часто используемыми для оценки шумных помех, вызванных продуктами/рабочими местами. [37] Его значение является нелинейным и представляет собой громкость звука, воспринимаемую человеческим ухом, это ощущение интенсивности. [33]

Обычно человеческое ухо фокусируется на частотах от 2000 до 5000 Гц. Именно поэтому даже песни, у которых одинаковое звуковое давление, физически измеренное в децибелах (дБ), но с частотами, не входящими в этот диапазон, воспринимаются как более тихие для человеческого уха. [6]

Вычисление громкости N достигается благодаря модели Цвикера и Стивенса: [37]

$$N = \int_{0Bark}^{24Bark} N'(z)dz \quad (36)$$

Где c - это коэффициент коррекции;

$g_s(z)$ - это весовая функция для резкости;

Где N′ - это специфическая громкость, это плотность громкости согласно частоте критической полосы (измеряется в сонах). Таким образом, N′(z) - это громкость в z-й полосе Барк.

N - громкость, представляет собой значение в сонах и соответствует звуковой громкости. Один сон представляет восприятие звуковой громкости, эквивалентной чистому звуку 1 кГц при уровне давления 40 дБ. Таким образом, два сона соответствуют звуку вдвое более интенсивному, чем один сон для среднего слушателя.

Барк - это шкала, используемая для описания частотного восприятия звука человеком. Она основана на работе нидерландского исследователя Эрнста Терквика (Ernst Terkivka) и представляет собой психоакустическую шкалу, где интервалы между значениями барка отражают восприятие изменений в частоте звука человеком. Барк-шкала позволяет более точно описать, как человеческое ухо воспринимает различные частоты звуков.

Сон - это единица измерения громкости звука, которая отражает восприятие громкости человеком. Сон представляет собой психоакустическую шкалу, основанную на модели Цвикера и Стивенса, где значения в сонах отражают восприятие относительной громкости звука. Один сон соответствует восприятию звука с определенной громкостью, например, звука частотой 1 кГц при уровне давления 40 дБ. Значение в сонах указывает на относительную интенсивность или громкость звука, воспринимаемую человеком.

Таким образом, барк и соны измеряют разные аспекты аудиальных характеристик звука: барк - частотное восприятие, а соны - громкость восприятия.

• Воспринимаемая резкость (Perceptual sharpness) - это показатель восприятия шума как высокочастотного. [33] Является эквивалентом спектрального центроида на воспринимаемом уровне, поэтому он вычисляется на основе громкости. [32] Низкая острота соответствует "тусклым звукам", а высокая острота соответствует "визгливым звукам". В целом слушатели предпочитают глухие звуки, но чрезмерно низкое значение также может быть раздражающим. Одна из возможных моделей называется Aures и вычисляется следующим образом: [37]

$$S = c \cdot \int_{0\text{Bark}}^{24\text{Bark}} \frac{N'(z) g_S(z)}{\ln \frac{N+20}{20}} dz \quad (37)$$

Где c - это коэффициент коррекции;

$g_s(z)$ - это весовая функция для резкости;

S - измеряется в акумах (acum).

• Воспринимаемая распространенность (Perceptual spread) - это мера расстояния между наибольшей специфической громкостью и общей громкостью: [32]

$$Sd = \left(\frac{N - \max_z N'(z)}{N}\right)^2 \quad (38)$$

• Воспринимаемая шероховатость. Этот критерий оценивает восприятие модуляций временной огибающей для частот от 20 до 150 Гц, максимум на 70 Гц (вариации низких и средних частот). Он позволяет количественно оценить быстрые изменения, которые могут восприниматься как диссонанс для слушателя. [33] Как и в случае с громкостью: [38]

$$R = \int_{0Bark}^{24Bark} R'(z)dz \quad (39)$$

Где R' - это специфическая шероховатость.

1.4.2 Признаки среднего уровня

Признаки среднего уровня сосредоточены на аспектах, которые имеют музыкальную значимость и понятны музыкальному эксперту. Первые из них сосредоточены на гармонии и мелодии музыки. Гармония определяется как совместное использование различных значений высоты тона и аккордов в музыке, она называется вертикальной частью музыки. Мелодия является горизонтальной частью, описывающей последовательность звуков, воспринимаемых как целое. [39] Различные характеристики позволяют извлекать информацию о гармонии и мелодии:

• Высота тона (Pitch tuning). Эта характеристика связана с фундаментальной частотой, то есть частотой, которая наилучшим образом соответствует спектральному содержанию сигнала. [40] Она используется для определения звуков как "высоких" или "низких" в смысле, связанном с музыкальными мелодиями. Для оценки высоты тона мы часто оцениваем так называемую систему настройки. Она определяет тоны (выбор числа и расстояния между значениями частоты), используемые в музыке.

• Тональность. Она очерчивает отношения между текущими и последовательными тональностями. [40] Она указывает, является ли режим трека мажорным или минорным.

• Следующие признаки сосредоточены на временных и ритмических свойствах музыки:

• Продолжительность композиции - это простой элемент, который можно извлечь и который может помочь нам классифицировать музыку.

• Моменты начала звуковых событий (Onset event). Обнаружение начала связано с определением временной позиции всех звуковых событий в музыкальном произведении.

• Метрические уровни. Эта характеристика соответствуют различным уровням вложенных импульсов, присутствующих в музыкальном произведении, причем более высокие метрические уровни являются кратными более низким. [40] Наименьший уровень, называемый татум, соответствует самым коротким длительностям. Тот, который слушатель описал бы как "самый важный", называется такт, он соответствует обиванию ногой сильных долей или тому, что обычно называют ритмическим пульсом. Такт позволяет нам определить темп, который является скоростью тактового пульса. [41]

• Удар - это фундаментальная единица времени. Обычно он составляет от 40 до 200 ударов в минуту. [40]

• Ритм также описывает повторение паттерна во времени на более длительных периодах, чем удары. [40]

1.4.3 Высокоуровневые признаки

Высокоуровневые признаки - это те, которые может понять любой слушатель, они описывают музыку так, как она воспринимается людьми. Поскольку они требуют интерпретации, они иногда кажутся интуитивными, но их надежное извлечение является сложным. С этими признаками необходимо быть осторожным, поскольку они не всегда являются актуальными.

• Танцевальность (Danceability) - это параметр оценивания способности музыки заставить людей танцевать. Обычно он принимает значения от 0 до 3, чем выше значение, тем более танцевальная музыка. [42] Один из способов его вычисления может быть основан на скорости v для каждого момента времени t и темпе музыки: [43]

$$D = \sum_{t} tempo * v(t) \quad (40)$$

• Живое выступление. Эта характеристика заключается в определении наличия или отсутствия аудитории во время записи.

• Речевость - это преобладание голосов в музыке позволяет, например, различить стихи/рэп с очень высокими значениями от джаза/классики, где значения будут очень низкими. [43]

• Инструментальность. Это характеристика противопоставляется предыдущей, высокое значение инструментальности соответствует сильному преобладанию инструментов. [43]

• Инструменты и певец. Знание присутствующих инструментов и наличие певца, а также информация о том, является ли он мужчиной или женщиной, могут помочь рекомендовать лучшую музыку.

• Оценка настроения. Характеризует настроение музыки, высокое значение соответствует радостной, живой музыке, в то время как низкое значение скорее всего означает грустную, низкую энергию или депрессивную музыку...

• Тексты. Настроение музыки также можно определить по текстам. Для извлечения информации из текстов используется обработка естественного языка (Natural Language Processing, NLP). Этот метод машинного обучения используется для анализа текстов и извлечения соответствующей информации.

1.5 Алгоритмы выбора признаков

Из аудиофайлов можно извлечь множество признаков. Задача состоит в исключении нерелевантных или менее значимых характеристик, которые могут увеличить сложность модели и время вычислений, при этом делая прогнозы менее надежными. Выбор признаков обычно определяется как процесс исследования с целью нахождения "релевантного" подмножества признаков. Алгоритмы выбора признаков, используемые для оценки подмножества признаков, могут быть классифицированы на три основные категории: фильтр, обертка и встроенные алгоритмы.

1.5.1 Модель фильтрации

Цель заключается в оценке важности признаков на основе мер, которые базируются на свойствах обучающих данных. Это этап предварительной обработки, который фильтрует признаки перед выполнением фактической классификации.

Пусть $X = \{x_k | x_k = (x_{k,1}, x_{k,2}, \ldots, x_{k,n}), k = 1,2,\ldots,m\}$ - набор из m обучающих значений. Пусть $Y = \{y_k, k = 1,2,\ldots,m\}$ - метки обучающих значений. Для определения важности признака существует несколько критериев оценки.

• Критерии корреляции: они используются в случае бинарной классификации, μ_i и μ_y соответственно представляют средние значения признака i и его меток: [44]

$$C(i) = \frac{\sum_{k=1}^{m}(x_{k,i}-\mu_i)(y_k-\mu_y)}{\sqrt{\sum_{k=1}^{m}(x_{k,i}-\mu_i)^2\sum_{k=1}^{m}(y_k-\mu_y)^2}} \quad (41)$$

• Критерий Фишера: измеряет степень отделимости классов с использованием заданного признака. [45]

$$F(i) = \frac{\sum_{C=1}^{C} n_c(\mu_c^i-\mu^i)^2}{\sum_{C=1}^{C} n_c(\sigma_c^i)^2} \quad (42)$$

Где n_c -количество образцов;

μ_c^i - среднее значение;

σ_c^i - стандартное отклонение для i-го признака внутри класса C;

μ^i - это общее среднее значение для первого признака.

• Коэффициент сигнал-шум (Signal-to-Noise Ratio): аналогично критерию Фишера, это показатель, который измеряет различительную способность характеристики между двумя классами:

$$SNR(i) = \frac{2 \cdot |\mu_{Ci1} - \mu_{Ci2}|}{\sigma(\mu_{Ci1} - \mu_{Ci2})} \quad (43)$$

Этот метод фильтрации эффективен и устойчив к переобучению. Однако, поскольку он не учитывает взаимодействие между признаками, он склонен выбирать признак с избыточной информацией, а не дополняющую информацию. Кроме того, этот метод не учитывает производительность методов классификации после выполнения выбора.

1.5.2 Модель обертки

Метод обертки был предложен Кохави и Джоном [46]. В данном случае оценка выполняется с использованием классификатора, который определяет значимость заданного подмножества признаков. Поэтому подмножество признаков, выбранных этим методом, соответствует используемому алгоритму классификации, но не обязательно будет действительным при изменении классификатора.

Наиболее распространенные реализации метода обертки включают:

• Прямой отбор (Forward selection): начинается без признаков и на каждом шаге добавляется наиболее значимый:

1. Выбрать уровень значимости (например, 0.05).

2. Выбрать признак, который лучше всего соответствует модели с более низким p-vaule.

3. Если p-vaule < α, добавить признак в набор признаков и вернуться к шагу 2, в противном случае завершить процесс. Здесь α - уровень значимости.

4. • Обратное исключение (Backward elimination): начинается со всех признаков и на каждом шаге удаляется незначимый:

5. Выбрать уровень значимости (например, 0,05).

6. Построить модель с использованием всех признаков из набора признаков.

7. Рассмотреть признак с наибольшим p-vaule.

8. Если p-vaule > α, удалить признак из набора признаков и вернуться к шагу 2, в противном случае завершить процесс.

• Двунаправленное исключение (Bidirectional elimination): Похоже на прямой отбор, при каждой итерации добавляется признак, но также проверяется значимость уже добавленных признаков и при необходимости может выполняться обратное исключение.

9. Выбрать уровень значимости (например, 0,05).

10. Выполнить шаги 2 и 3 прямого отбора.

11. Выполнить шаги 2, 3 и 4 обратного исключения.

12. Повторять шаги 2 и 3 до нахождения оптимального набора признаков.

Этот метод считается лучшим. Он выбирает небольшие подмножества признаков, которые эффективны с использованием выбранного классификатора, однако существует два основных недостатка, которые ограничивают эти методы: во-первых, время вычислений значительно дольше, чем у предыдущего метода, а использование кросс-валидации для уменьшения риска переобучения увеличивает эту проблему. Второй вызов заключается в применении этого механизма выбора для каждого классификатора, который нужно протестировать.

2 Реализация рекомендательной системы

2.1 Выбранный подход

Так как найти достаточное количество открытых данных о пользователях оказалось сложно, было решено разработать систему рекомендаций на основе контента. В литературе описано множество работ, выполненных с использованием различных моделей и признаков, но нет единого мнения о том, какая из них является "лучшей". Фактически, несколько статей описывают свой опыт с использованием одной модели и иногда очень ограниченного набора признаков, что приводит к отсутствию стандартизации и сравнению. Цель - определить как модель (и ее оптимальные параметры), так и набор признаков, чтобы получить наилучшие рекомендации.

2.2 Наборы данных

Хороший набор данных должен включать в себя некоторые характеристики. Во-первых, требуется большое количество данных. Нам нужно много музыки, но также нам нужно разнообразие жанров и исполнителей. Это позволит нам минимизировать риск переобучения модели. Второе основное требование заключается в том, чтобы аудиофайлы были доступны, чтобы мы могли извлечь как можно больше признаков. Это позволит не ограничиваться только музыкой из набора данных, а также добавлять музыку и вычислять признаки для новой музыки таким же образом, как это было сделано с набором данных.

Набор данных LFM-1b был принят на конференции ICMR 2016. Он состоит из миллиарда событий прослушивания музыки, сгенерированных около 120 тысячами пользователями с Last.fm для 32 миллионов треков. Он предоставляет доступ как к метаданным пользователей (страна, возраст, количество прослушиваний музыки и т. д.), так и к метаданным объектов, поэтому его основным преимуществом является возможность использования коллаборативных подходов к фильтрации. Однако этот набор данных не предоставляет доступ к исходным файлам, и поэтому невозможно извлекать признаки.

Набор данных MAESTRO (MIDI and Audio Edited for Synchronous TRacks and Organization-Dataset) был создан для международного конкурса по фортепианной музыке. Он состоит из 1200 треков виртуозного фортепиано, и аудиофайлы для этих треков можно скачать. Доступны некоторые метаданные, а также файлы MIDI. Это может быть полезно, в частности, для генерации музыки с помощью искусственного интеллекта. Несмотря на то, что доступны исходные аудиофайлы, набор данных остается очень небольшим, а музыка

принадлежит одному и тому же жанру, поэтому будет сложно предложить разнообразный контент, который может понравиться всем вкусам.

2.2.1 Используемые наборы данных

GTZAN - это минималистичный набор данных, содержащий одну тысячу музыкальных сэмплов продолжительностью 30 секунд из 10 разных жанров. Предоставляются исходные аудиофайлы с однократной маркировкой. Музыка, выбранная для каждого жанра, является типичной, и их сходство очень высоко. Этот набор данных широко используется в области рекомендации музыки. Он содержит музыку, которая типично представляет свой жанр и обычно дает хорошие результаты. Высокая производительность предыдущих работ также объясняется тем, что этот набор данных содержит много повторяющихся элементов. Некоторые из этих повторений являются точными, то есть отпечатки музыки идентичны, но также есть различные версии (студийные, живые) одной и той же музыки. Наконец, исполнители часто представлены несколькими композициями. В этом наборе данных также можно наблюдать ошибки маркировки и искажения.

FMA (Free Music Archive) - это масштабный набор данных, созданный для анализа музыки, который состоит из более чем 100 000 треков из 161 жанра. Он предоставляет различные метаданные на уровне трека (включая низкоуровневые признаки), артиста и альбома в файлах CSV. Основным преимуществом является то, что набор данных содержит исходные аудиофайлы высокого качества и полной длительности. Музыка маркирована по основному жанру (из 16 жанров), но также имеет несколько поджанров. Более того, можно загрузить меньшие поднаборы данных. Набор данных FMA кажется наиболее подходящим, поскольку он довольно большой и предоставляет аудиофайлы, позволяющие проанализировать их и извлечь нужные признаки.

Недостатком этого набора данных является то, что он содержит исключительно данные о музыке (а также об исполнителе и альбоме). Поскольку у нас нет никаких данных о пользователе, мы будем полагаться только на контентные рекомендации.

Доступны следующие поднаборы данных:

• Малый: 8 000 треков - 8 жанров - 30 секунд, 1000 композиций в каждом жанре, все сбалансированы.

Таблица 1 - Количество образцов на класс в небольшом наборе данных

Жанры	**Количество треков**
Electronic	8000
Experimental	8000
Folk	8000
Hip-Hop	8000
Instrumental	8000
International	8000
Pop	8000
Rock	8000

• Средний: 25 000 треков - 16 жанров - 30 секунд.

Таблица 2 - Количество образцов на класс в среднем наборе данных

Жанры	**Количество треков**
Blues	74
Classical	619
Country	178
Easy Listening	21
Electronic	6311
Experimental	2249
Folk	1516
Hip-Hop	2197
Instrumental	1349
International	1018

Окончание таблицы 2

Жанры	**Количество треков**
Jazz	384
Old-Time / Historic	510
Pop	1186
Rock	7097
Soul-RnB	154
Spoken	118

- Большой: 103 000 треков - 161 жанр - 30 секунд.

Таблица 3 - Количество образцов на класс в большом наборе данных

Жанры	**Количество треков**
Blues	110
Classical	1230
Country	194
Easy Listening	24
Electronic	9372
Experimental	10608
Folk	2803
Hip-Hop	3552
Instrumental	2079
International	1389
Jazz	571
Old-Time / Historic	554

Окончание таблицы 3

Жанры	Количество треков
Pop	2332
Rock	14182
Soul-RnB	175
Spoken	423
NaN	56976

• Полный: 106 000 треков - 161 жанр - полная длительность.

В среднем и большом наборе данных количество примеров в каждом классе неодинаково, поэтому придется иметь дело с проблемами дисбаланса классов.

2.2.2 Аугментация данных

Одним из недостатков набора данных FMA, который можно увидеть по предыдущим таблицам, является проблема дисбаланса классов: некоторые жанры преобладают в сравнении с другими. Чтобы справиться с этой проблемой, можно использовать методы недо- и пересэмплинга. Первый метод заключается в уменьшении числа примеров из преобладающих классов, а второй метод заключается в увеличении данных из недостаточно представленных классов для получения большего количества данных. Чтобы предотвратить переобучение на новом наборе данных и повысить устойчивость системы, эти методы могут быть комбинированы с аугментацией данных. Кроме того, это может быть полезно для наборов данных, таких как GTZAN, содержащих очень мало музыки. Дополнение данных широко используется для изображений, но также существуют некоторые техники для аудиофайлов. Это позволяет добавлять немного отличающиеся данные, применяя небольшие вариации. Мною выбраны для реализации процедуры, введенные в [55]. Один из возможных методов - добавление шума (случайно генерируемого) к сигналу. Интенсивность шума может быть изменена, но необходимо обеспечить сохранение мелодии. Также можно "сдвигать музыку во времени", то есть дополненная музыка начинается не с первой секунды, а оригинальное начало музыки прикрепляется в конце. Еще один метод - изменение скорости музыки, что позволяет растягивать ее или делать короче. Наконец, можно случайным образом изменять высоту звука, чтобы больше акцентировать внимание на басе или высоких

звуках. Эффективно использовать смесь нескольких таких вариаций. Исследовательская статья [55] указывает, что, хотя результаты обнадеживающие, они являются предварительными и необходимо обращать внимание на используемые методы в соответствии с целью проекта. Набор признаков будет немного адаптирован в соответствии с используемыми методами. Действительно, при использовании метода изменения высоты звука признаки, зависящие от высоты звука, не будут учитываться, и то же самое относится к темпу при изменении скорости.

2.3 Извлечение признаков

2.3.1 Предварительная обработка

Музыка в наборе данных FMA представлена в формате mp3, а в наборе данных GTZAN - в формате wav. Первый шаг - извлечение спектральных огибающих. Затем наборы данных разделены на обучающий набор, содержащий 90% данных, и тестовый набор, содержащий 10% данных. Для последующей оптимизации гиперпараметров будет введен третий набор данных, называемый набором данных для валидации.

Из спектральных огибающих каждой музыки предпринимается попытка извлечь как можно больше полезных признаков, чтобы наилучшим образом описать все характеристики музыки. В зависимости от признаков существуют три разных режима извлечения: либо на заданном временном отрезке t, на всей музыке, либо (в большинстве случаев) по кадру.

2.3.2 Выбранные признаки

После завершения предварительной обработки и извлечения спектральных огибающих, были вычислены признаки. Для этой цели были использованы три библиотеки. Основная - это librosa (библиотека для распознавания и организации речи и аудио), [56] это открытая библиотека на языке Python, используемая для обработки аудио и сигналов музыки. Essentia [57] использовалась для извлечения танцевальности. Эта библиотека на языке C++, также с открытым исходным кодом, предназначена для проектов извлечения информации о музыке на основе аудио. Наконец, для извлечения восприятий признаков была использована библиотека Yaafe [58] (Yet Another Audio Feature Extractor).

Извлеченные признаки для проведения тестов включают: громкость, воспринимаемая резкость, шероховатость и распространенность, танцевальность, спектральное убывание, частота спада спектра, спектральный поток, спектральный центроид, спектральная

плоскость, спектральный наклон, система настройки**,** среднеквадратическая энергия, количество пересечений нуля, моменты начала звуковых событий и 40 коэффициентов МЧКК.

Среди извлеченных признаков есть два типа: глобальные и мгновенные. Глобальный тип имеет уникальное значение для каждой музыки, которое вычисляется на всем сигнале. Признаки типа мгновенный вычисляются на коротком отрезке времени, называемом фреймом (примерно 20 мс). В этом случае были изучены два метода ввода входных значений в наши модели: первый - сохранить только статистические характеристики (среднее, дисперсия, асимметрия, эксцесс и т. д.) значений, полученных на разных фреймах. Второй метод - сохранить его временную размерность и использовать адаптированную нейронную сеть.

2.3.3 Модель-обертка для выбора признаков

Для избежания избыточности информации и ограничения посторонних признаков, модель-обертка, определенная в разделе “Изучение предметной области” в подразделе “Модель обертки”, будет применяться для сохранения только подмножества значимых признаков.

2.4 Модели

Все модели, определенные в обзоре существующих методов, были протестированы в экспериментах: логистическая регрессия, дерево решений, случайный лес, адаптивный бустинг, метод ближайших соседей, метод опорных векторов, наивный байесовский классификатор и нейронные сети прямого распространения. Параметры этих моделей были изменены с целью получить оптимальные настройки для каждой модели. Эти настройки называются гиперпараметрами - параметрами, которые устанавливаются заранее и не обучаются на данных. Для объективной оценки модели лучшие гиперпараметры будут выбраны путем расчета ошибки на валидационном наборе данных, а обобщающая способность модели будет определена путем расчета ошибки на тестовом наборе данных.

2.4.1 Настройка гиперпараметров

Для конкретной модели проводится различие между параметрами и гиперпараметрами. Параметры являются внутренними для модели, они будут оцениваться во время обучения и будут использоваться позже для предсказаний. Гиперпараметры являются

внешними параметрами, которые определяются перед обучением модели. Для определения оптимальных гиперпараметров использовался поиск по сетке (grid search). Принцип заключается в указании диапазона значений для каждого гиперпараметра, который нужно оптимизировать. Чтобы избежать переобучения насколько это возможно, используется стратегия перекрестной проверки. Для получения релевантных результатов при разумном времени вычислений была использована перекрестная проверка с пятью фолдами.

Для логистической регрессии основным параметром для изменения является solver (метод решения). Первый метод, адаптированный для многоклассовой классификации, - это метод нелинейной сопряженной градиентной оптимизации (newton-cg): основанный на гессиане, этот метод склонен быть медленным для больших наборов данных, так как требуется вычислить вторую частную производную. Второй метод - метод ограниченной памяти Бройдена-Флетчера-Голдфарба-Шанно (lbfgs): он также требует вычисления второй производной, что приводит к замедлению, однако использование памяти более оптимизировано, поскольку он сохраняет только некоторые обновления. Метод стохастического среднего градиента (sag и saga) быстрее для больших наборов данных, но затратен по памяти, это связано с оценкой градиента путем вычисления его на случайно выбранном подмножестве. Параметр регуляризации также может быть изменен между l1, l2 и elasticnet в зависимости от выбранного метода решения.

Для нахождения оптимальных параметров дерева решений необходимо определить как критерий принятия решения (gini, энтропия), так и максимальную глубину дерева, чтобы достичь хорошей производительности, избегая переобучения. Аналогично, для случайного леса варьируются как критерий, так и максимальная глубина дерева, а также количество деревьев (называемое количеством оценщиков). Для бустинга также можно настраивать количество оценщиков.

Веса точек для метода k-ближайших соседей могут вычисляться равномерно (то есть все точки имеют одинаковый вес) или с использованием весов, обратно пропорциональных расстоянию (то есть ближайшие соседи имеют большее влияние, чем самые удаленные соседи).

Метод опорных векторов (Support Vector Machine) поддерживает использование различных ядер (линейное, полиномиальное, сигмоидное или rbf). Регуляризационный параметр, называемый C, используется для изменения штрафа.

При использовании линейного дискриминантного анализа (Linear Discriminant Analysis) два основных параметра могут влиять на результаты: метод решения (solver) и интенсивность сжатия (shrinking intensity). Основные методы решения включают наименьшие квадраты (lsqr), собственное разложение (eigen) и сингулярное разложение (svd). Интенсивность сжатия может варьироваться от 0 до 1, и для автоматизации этого процесса можно использовать лемму Ледуа-Вольфа (Ledoit-Wolf lemma).

Гиперпараметров нейронных сетей много, поэтому поиск оптимальных параметров может занимать много времени. В первую очередь необходимо определить количество эпох (количество образцов, проходящих через модель во время прямого и обратного прохода) и размер пакета (batch size), используемого для каждой эпохи. Затем выбирается оптимизатор, который дает наилучшие результаты. Также можно использовать известный алгоритм стохастического градиентного спуска при оптимизации скорости обучения (learning rate) и момента (momentum), чтобы контролировать, насколько обновляются веса. Целью будет определить, какой из двух предыдущих подходов кажется наилучшим. Способ инициализации весов перед первым прямым проходом также влияет на результаты. Основные подходы включают инициализацию всех весов одним значением (обычно 0 или 1) или их случайную инициализацию (часто из равномерного распределения). Чтобы избежать проблем с взрывным или исчезающим градиентом, используются эвристики (формула которых зависит от количества слоев) для нахождения весов. Для обеспечения сходимости сети и контроля скорости обучения существуют различные функции активации. Наконец, для предотвращения переобучения применяется метод исключения (dropout): определяется процент нейронов, которые будут случайным образом отключены (временно выключены) на каждой эпохе.

2.5 Оценка

Оценка рекомендательной системы проходит в два этапа: первый этап - это количественная оценка на основе математических и статистических методов, а на втором этапе несколько человек прослушивают рекомендации.

2.5.1 Оценка классификации с использованием меток

Каждая музыка имеет определенный жанр, который считается истинным, и модель будет предсказывать жанр для каждой музыки из тестового набора данных. Для оценки классификации песен используются несколько методов [7]:

Первый метод - это точность (precision), который показывает количество извлеченных элементов, которые действительно являются релевантными. [7] Пусть C - количество классов, $\mathrm{Rel_c}$ - множество релевантных элементов для класса c, $\mathrm{Ret_c}$ - множество извлеченных элементов для класса C. Точность класса c и средняя точность определяются следующим образом:

$$\mathrm{P_c} = \frac{|\mathrm{Rel_c} \cap \mathrm{Ret_c}|}{\mathrm{Ret_c}} \qquad \mathrm{P} = \frac{1}{|\mathrm{C}|} \sum_{c \in C} \mathrm{P_c} \tag{44}$$

В рекомендательных системах целью является обеспечение того, чтобы большинство предсказанных элементов были релевантными. Действительно, желательно, чтобы платформы удерживали своих клиентов, поэтому желательна высокая точность.

Второй метод - это полнота (recall), который показывает, сколько релевантных элементов было извлечено. [7] Полнота класса c и средняя полнота определяются следующим образом:

$$\mathrm{R_c} = \frac{|\mathrm{Rel_c} \cap \mathrm{Ret_c}|}{\mathrm{Rel_c}} \qquad \mathrm{R} = \frac{1}{|\mathrm{C}|} \sum_{c \in C} \mathrm{R_c} \tag{45}$$

Наконец, в качестве меры точности теста также можно использовать F-меру. Средняя F-мера определяется следующим образом:

$$\mathrm{F_\beta} = \frac{(1+\beta) \cdot \mathrm{P} \cdot \mathrm{R}}{\beta \cdot \mathrm{P} + \mathrm{R}} \tag{46}$$

F1-мера, называемая гармоническим средним, часто используется, но можно использовать более высокое значение бета, чтобы уделять больше внимания полноте по сравнению с точностью.

2.5.2 Оценка предсказаний с использованием матрицы путаниц

Наконец, очень полезным инструментом является матрица путаниц (confusion matrix). При многоклассовой классификации использование матрицы ошибок является важным. Она показывает, какие жанры музыки лучше всего распознаются и, прежде всего, как ошибки классифицируются. Действительно, поскольку некоторые жанры близки друг к другу,

влияние некоторых ошибок невелико. Идеально было бы увидеть на матрице диагональ с высокими процентами точности и остальные значения как можно ниже. Однако визуализация ошибок также будет очень важной, поскольку границы между жанрами не всегда очень хорошо определены.

2.6 Алгоритм рекомендательной системы

Алгоритм рекомендательной системы представлен на рисунке 8.

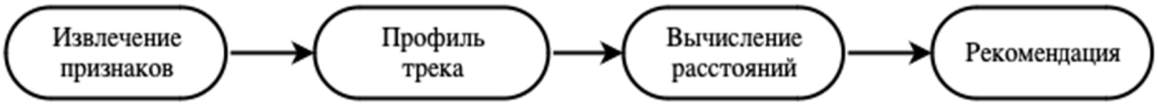


Рисунок 8 - Алгоритм рекомендательной системы.

1. Извлечение признаков: Первым шагом будет извлечение признаков из данного трека. Это включает выше описанные низкоуровневие, среднеуровневые и выскокоуровневые признаки.

2. Создание профиля трека: На основе извлеченных признаков создается профиль трека. Профиль представлен числовым вектором, содержащим значения признаков трека.

3. Вычисление расстояния до треков: Далее происходит вычисление расстояния между профилем данного трека и профилями других треков. Для определения сходства используется метрика евклидова расстояния.

4. Рекомендация треков: Используя вычисленные расстояния, выбираются треки, которые наиболее близки к данному треку.

3 Результаты

3.1 Предварительные результаты

Были проведены некоторые предварительные тесты на каждом из наборов данных (FMA и GTZAN), чтобы определить, какой из них будет использоваться.

3.1.1 Тесты на FMA

Для каждой музыки есть уникальное значение "genre top", которое относится к одному из 16 основных жанров (блюз, классика, кантри, легкая музыка, электроника, экспериментальная, фолк, хип-хоп, инструментальная, международная, джаз, старинная, поп, рок, соул-ар-энд-би и спокен). Это значение отсутствует для половины обширного набора данных, но всегда указывается для остальных. Другой полезной меткой является "genres all", она содержит набор жанров (среди 161 поджанров) для каждой музыки.

Для достижения контролируемой классификации первым шагом предварительной обработки является отбрасывание музыки из наборов данных, у которой отсутствует метка (ни "genre top", ни "genres all"). Жанр "Spoken" был исключен, так как нас интересует только музыка. Были также отброшены нераспознаваемые аудиофайлы. Получив уменьшенный набор данных, из каждой музыки извлекается спектральная огибающая и частота дискретизации, чтобы получить признаки.

Первые результаты, представленные в таблице 4 и полученные с использованием малого набора данных FMA, оказались не такими хорошими, как ожидалось. Максимальная точность, достигнутая на тестовом наборе, составляет 55% при использовании нейронной сети. Просматривая матрицы путаницы (рисунок 9), можно сказать, что достигнута далеко не ожидаемая диагональ. После прослушивания некоторых образцов и изучения этого набора данных выявляются несколько неточностей. Прежде всего, музыка, классифицированная как "экспериментальная", скорее соответствует повседневным шумам (звуки разрушения объектов), чем настоящей музыке, поэтому было решено удалить их из конечного набора данных. Кроме того, рассматривая жанр "international", он фактически включает африканскую, латинскую, индийскую, французскую музыку и т.д. Поэтому понятно, почему нашим моделям трудно распознать этот жанр и найти сходства между разными музыками.

Таблица 4 - Точность обучения и тестирования на малом наборе данных FMA

	Тренировочный набор	**Тестовый набор**
Логистическая регрессия	58%	42%
Дерево решений	100%	30%
Случайный лес	100%	32%
Adaboost	32%	28%
K-NN	68%	53%
Метод опорних векторов (SVM)	70%	51%
Наивный Баейс	37%	27%
Нейронная сеть прямого распространения (FFNN)	100%	55%

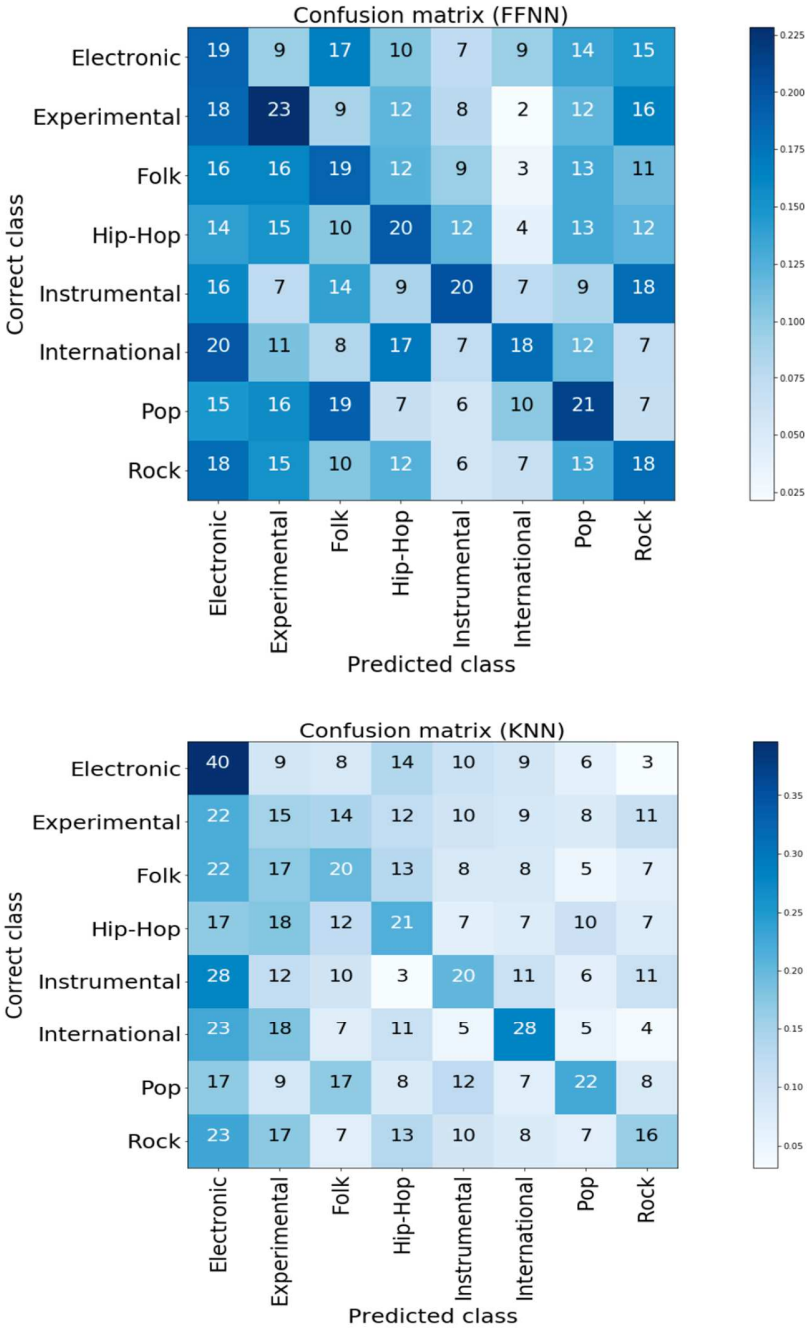


Рисунок 9 - Матрицы путаницы для малого набора данных FMA (в процентах)

Таблица 5 - Точность обучения и тестирования на большом наборе данных FMA

	Тренировочный набор	**Тестовый набор**
Логистическая регрессия	55%	28%

Окончание таблицы 5

	Тренировочный набор	**Тестовый набор**
Дерево решений	100%	28%
Случайный лес	100%	29%
Adaboost	24%	23%
K-NN	42%	29%
Метод опорних векторов (SVM)	43%	29%
Наивный Баейс	31%	19%
Нейронная сеть прямого распространения (FFNN)	100%	21%

В то время как жанры сбалансированы в малом наборе данных FMA, это не так в случае более крупных наборов данных, поэтому начальные результаты оказываются плохими (таблица 5). Поэтому, используя набор данных в его текущем состоянии, классификаторы склонны классифицировать всю музыку в наиболее представленные жанры, в данном случае рок, экспериментальная и электронная музыка. Этот эффект особенно заметен на матрицах путаницы (рисунок 10).

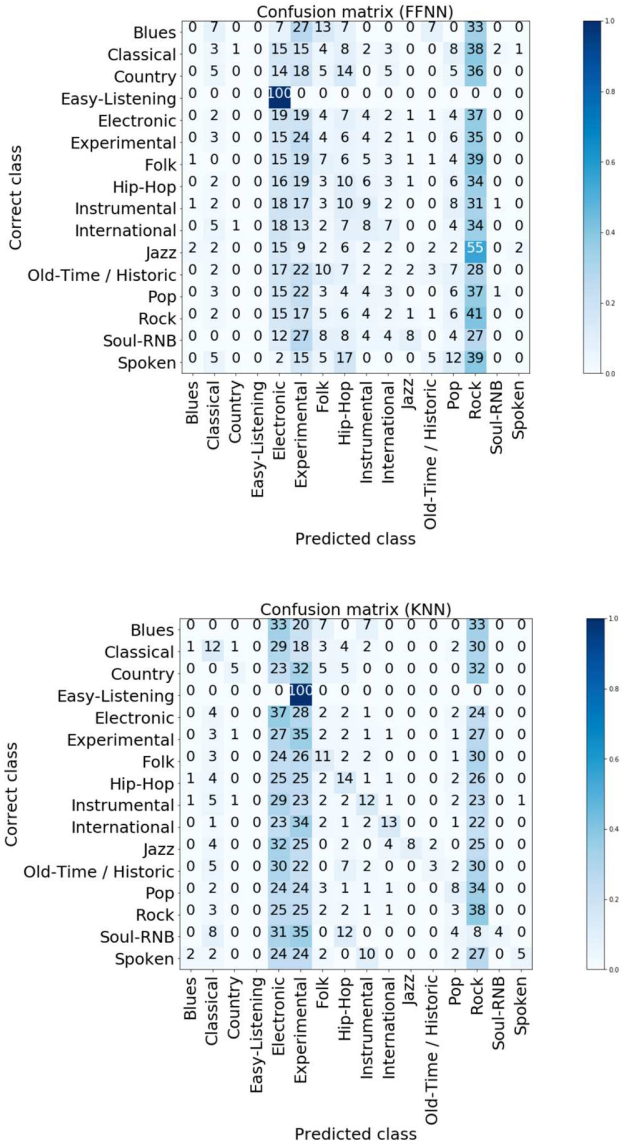


Рисунок 10 - Матрицы путаницы для большого набора данных FMA (в процентах)

3.1.2 Тесты на GTZAN

Что касается этого набора данных, первые результаты, как числовые (таблица 6), так и при прослушивании, гораздо более обнадеживающие.

Таблица 6 - Точность обучения и тестирования на наборе данных GTZAN

	Тренировочный набор	**Тестовый набор**
Логистическая регрессия	87%	65%

Окончание таблицы 6

	Тренировочный набор	**Тестовый набор**
Дерево решений	100%	45%
Случайный лес	100%	58%
Adaboost	32%	29%
K-NN	80%	64%
Метод опорних векторов (SVM)	86%	65%
Наивный Баейс	55%	8%
Нейронная сеть прямого распространения (FFNN)	100%	63%

Судя по таблице 6, одна вещь, которая выделяется, заключается в том, что результаты, полученные с помощью модели наивного Байеса, не могут быть использованы. Как уже упоминалось ранее, поскольку признаки сильно коррелируют между собой, этот алгоритм дает плохие результаты и классифицирует каждую музыку как классическую. Более интересным фактом является то, что желаемая диагональ (рисунок 11) присутствует гораздо больше, чем в случае с набором данных FMA, и это верно для всех других моделей, особенно при использовании метода опорных векторов. Также можно заметить, что большинство методов склонны к переобучению и достигают 100% точности на тренировочном наборе данных, в то время как точность на тестовом наборе гораздо ниже. Изменение параметров и добавление штрафа к этим моделям может помочь уменьшить важность этой проблемы. Наконец, случайный лес оказывается значительным улучшением по сравнению с классическим деревом решений, тогда как метод Adaboost здесь не дает достаточно хороших результатов.

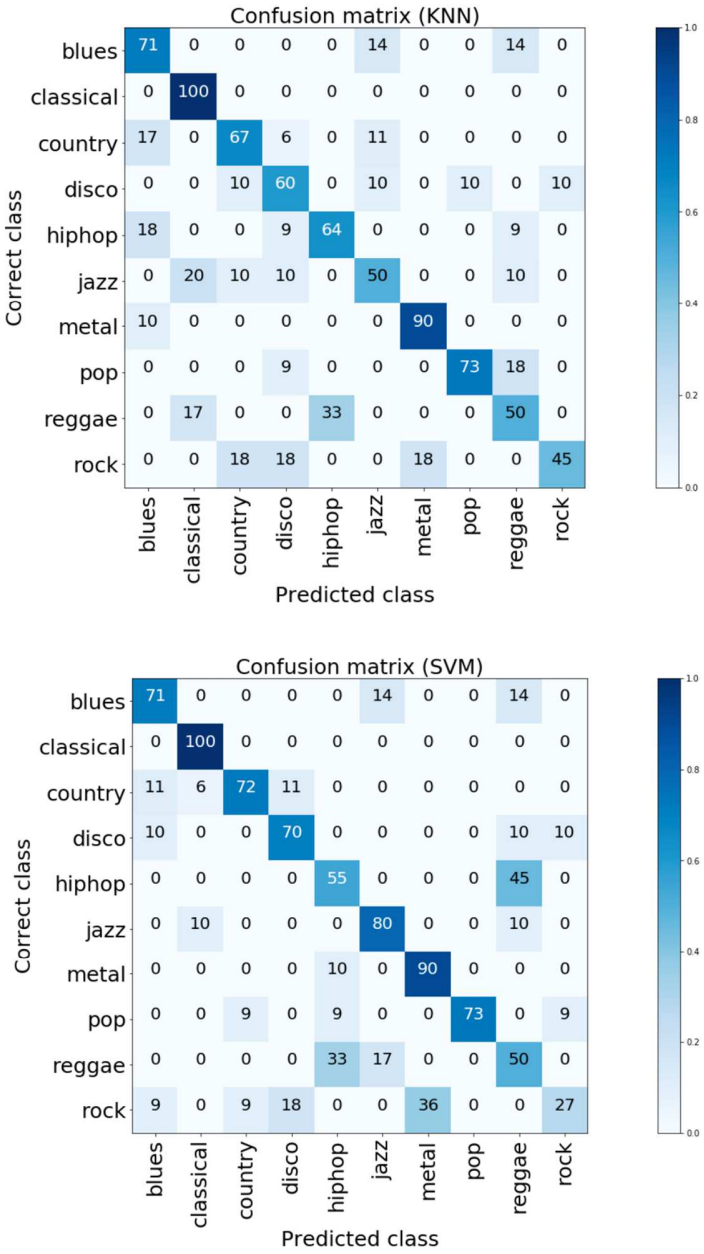


Рисунок 11 - Матрицы путаницы для набора данных GTZAN (в процентах)

Однако, как объяснено выше, если эти предварительные результаты намного обнадеживающие, это связано с тем, что выбранные композиции (их характеристики) очень похожи друг на друга и специально выбраны, чтобы быть репрезентативными для своих жанров. Также артисты не очень разнообразны. Это незначительная, но существующая проблема, поскольку мы стремимся разработать систему рекомендаций, которую можно обобщить на музыку, не представленную в нашем наборе данных.

3.2 Создание набора данных

Из-за ранее упомянутых проблем было принято решение создать новый набор данных для этого проекта. Этот новый набор данных, используемый в данной магистерской диссертации, будет комбинацией наборов данных GTZAN и FMA. Сам набор данных FMA имеет слишком много недостатков, чтобы использовать его отдельно. Множество повторений в наборе данных GTZAN и его ограниченность затрудняют обобщение нашей модели на новую музыку.

Новый набор данных будет состоять из 100 песен из 10 жанров, представленных в наборе данных GTZAN, к которым мы добавили по 50 песен из набора данных FMA для каждого жанра. В итоге каждый из 10 жанров будет представлен 150 песнями, достаточно разнообразными, чтобы ограничить переобучение модели на наборе данных. Эти результаты более исчерпывающие (таблица 7, рисунок 12) и более представительные для нашего набора данных, они представляют каждый жанр, избегая проблемы дисбаланса классов.

Таблица 7 - Точность обучения и тестирования на наборе данных GTZAN / FMA

	Тренировочный набор	**Тестовый набор**
Логистическая регрессия	87%	71%
Дерево решений	100%	48%
Случайный лес	100%	60%
Adaboost	33%	31%
K-NN	81%	65%
Метод опорних векторов (SVM)	90%	67%
Наивный Баейс	58%	48%
Нейронная сеть прямого распространения (FFNN)	100%	69%

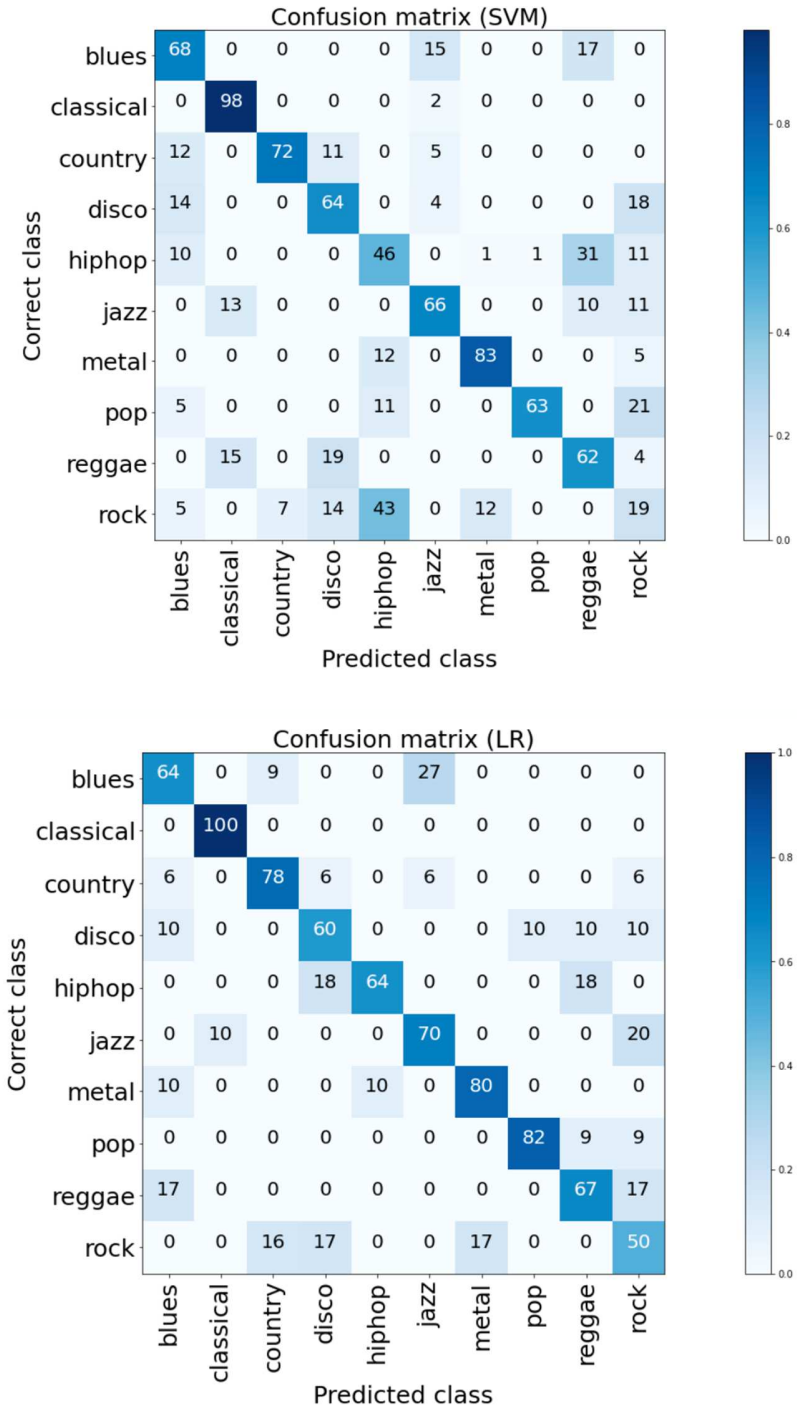


Рисунок 12 - Матрицы путаницы для нового набора данных (в процентах)

3.3 Настройка гиперпараметров

Целью этой части является настройка гиперпараметров моделей для достижения наилучших возможных результатов. Чтобы ограничить риски переобучения и получить лучшую обобщенность на внешних данных, была использована кросс-валидация с k-фолдами (k находится между 5 и 10). Выбор гиперпараметров будет основан на полученных результатах, как средней точности, так и дисперсии, а также на времени, необходимом для выполнения вычислений.

3.3.1 Оптимизация логистической регрессии

На нашем наборе данных все решатели дают схожие результаты (рисунок 13). Максимальное количество итераций будет установлено на 100. Чтобы избежать переобучения, значение параметра C будет установлено относительно низким: 1. Далее будет использован решатель liblinear с типом штрафа l1, так как он хорошо подходит для небольших наборов данных, таких как GTZAN.

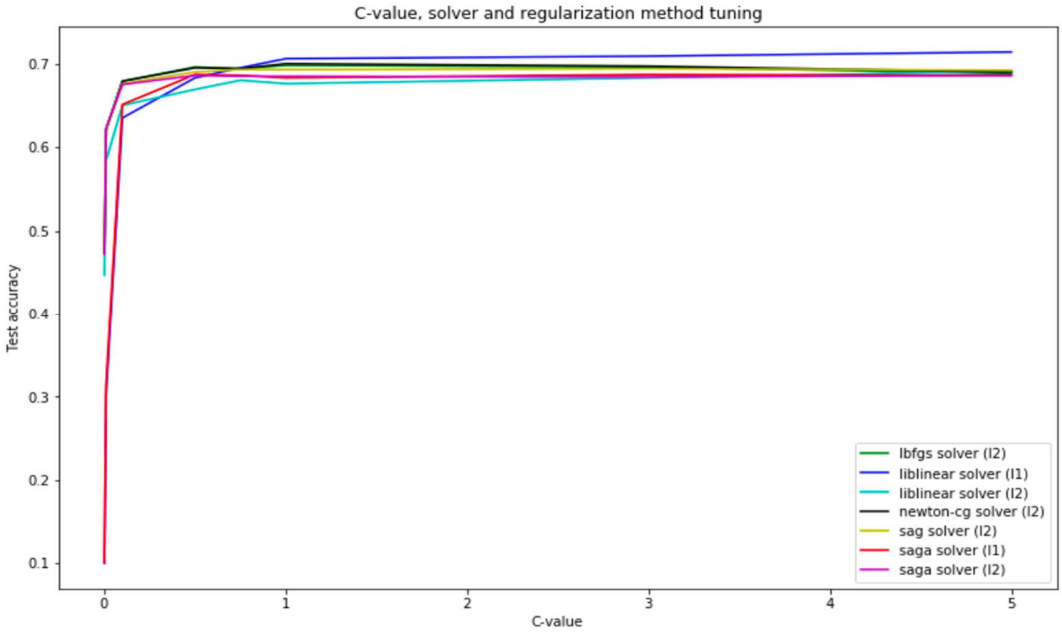


Рисунок 13 - Настройка гиперпараметров логистической регрессии

3.3.2 Оптимизация дерева решений и случайного леса

Для простых деревьев решений (рисунок 14) критерий принятия решений (который измеряет качество разбиения) на основе энтропии и прироста информации дает лучшие результаты: точность 57% по сравнению с 54%, используя критерий Gini.

С другой стороны, для случайных лесов обе функции дают очень схожие результаты (рисунок 15). Наблюдается явное улучшение точности до 73% при использовании случайного леса вместо дерева решений, что подтверждает мощность этих методов в целом. В следующей части проекта будет выбрана функция энтропии и максимальная глубина 24 для максимизации средней точности и минимизации дисперсии.

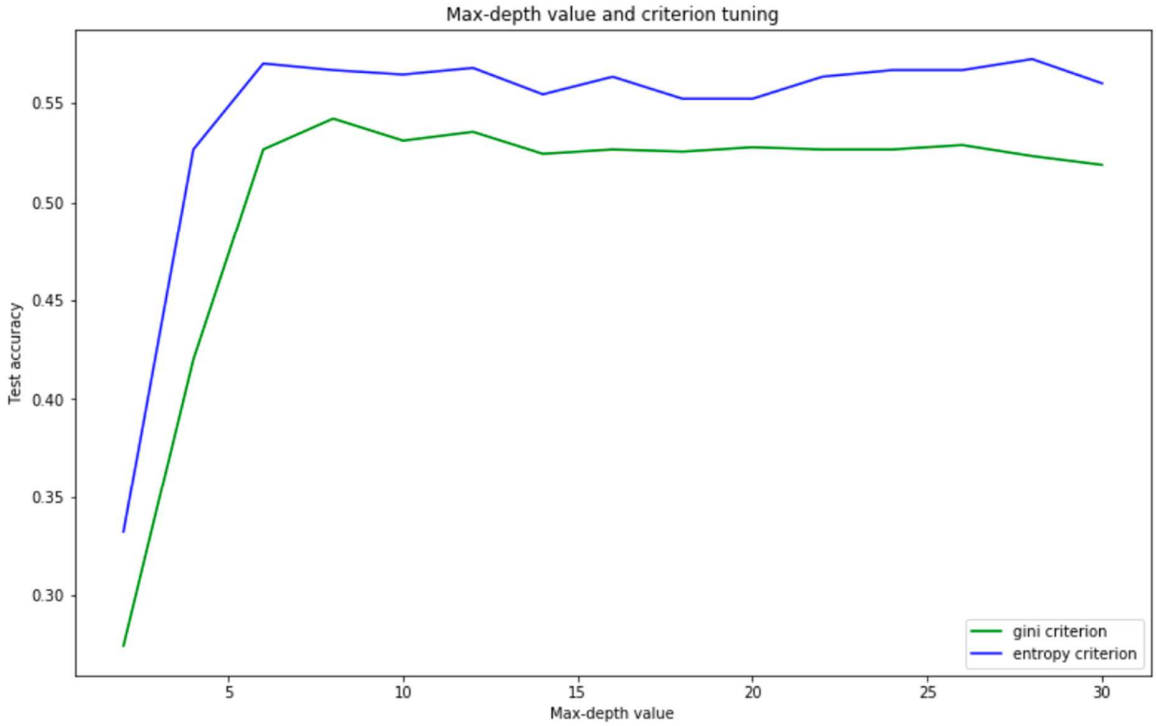


Рисунок 14 - Настройка гиперпараметров дерева решений

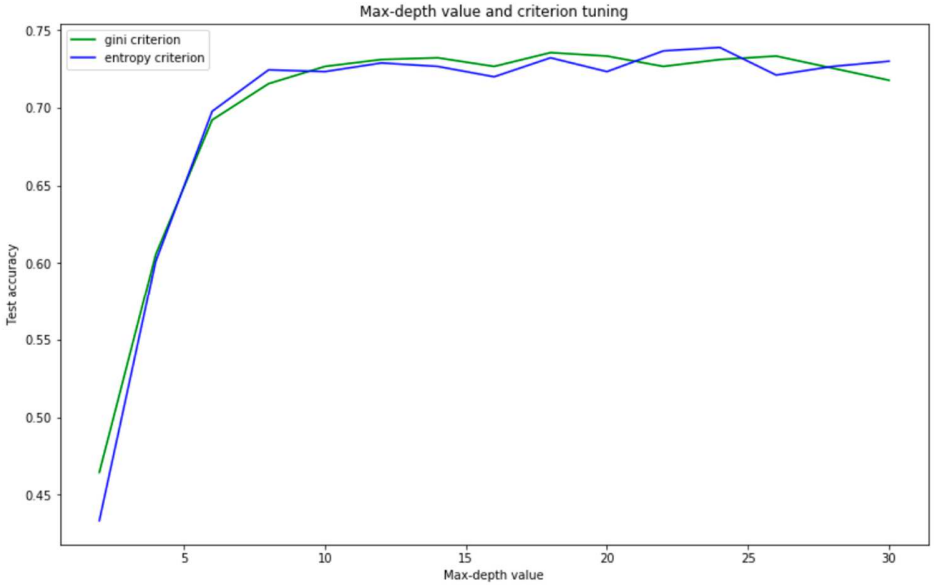


Рисунок 15 - Настройка гиперпараметров случайного леса

3.3.3 Оптимизация Adaboost

Оптимальные параметры для алгоритма Adaboost - 0,01 для скорости обучения и 60 оценщиков. Однако достигнутая точность остается ограниченной с максимальным значением 41%.

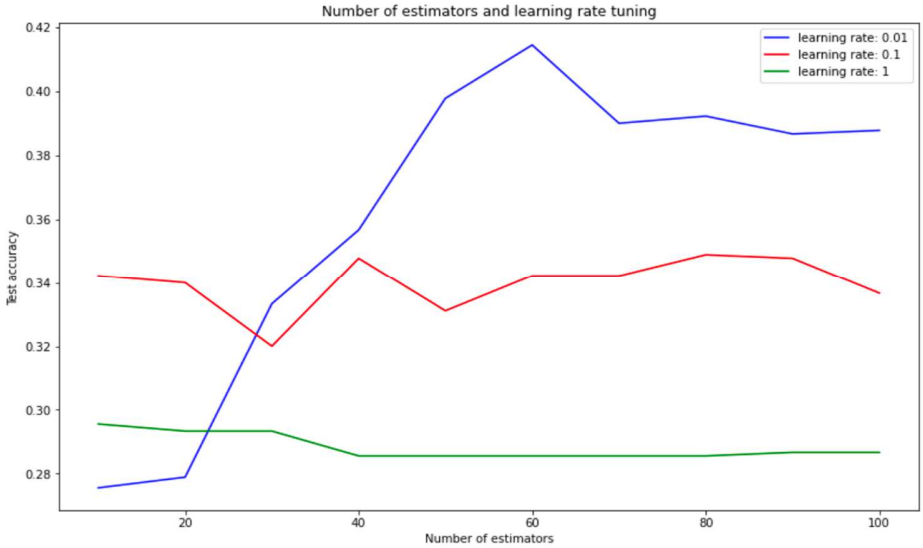


Рисунок 16 - Настройка гиперпараметров Adaboost

3.3.4 Оптимизация метода ближайших соседей (k-nearest-neighbours)

В целом, согласно рисунку 17, результаты лучше при использовании евклидова и манхэттенского расстояний. Поскольку полученные дисперсии при использовании евклидова расстояния ниже, в дальнейших экспериментах будет предпочтительно использовать именно это расстояние. Кроме того, если веса точек будут обратно пропорциональны их расстоянию до изучаемых точек, достигается более высокая точность, чем если веса были бы равномерными. Количество соседей K будет установлено на 5 для достижения точности 72%.

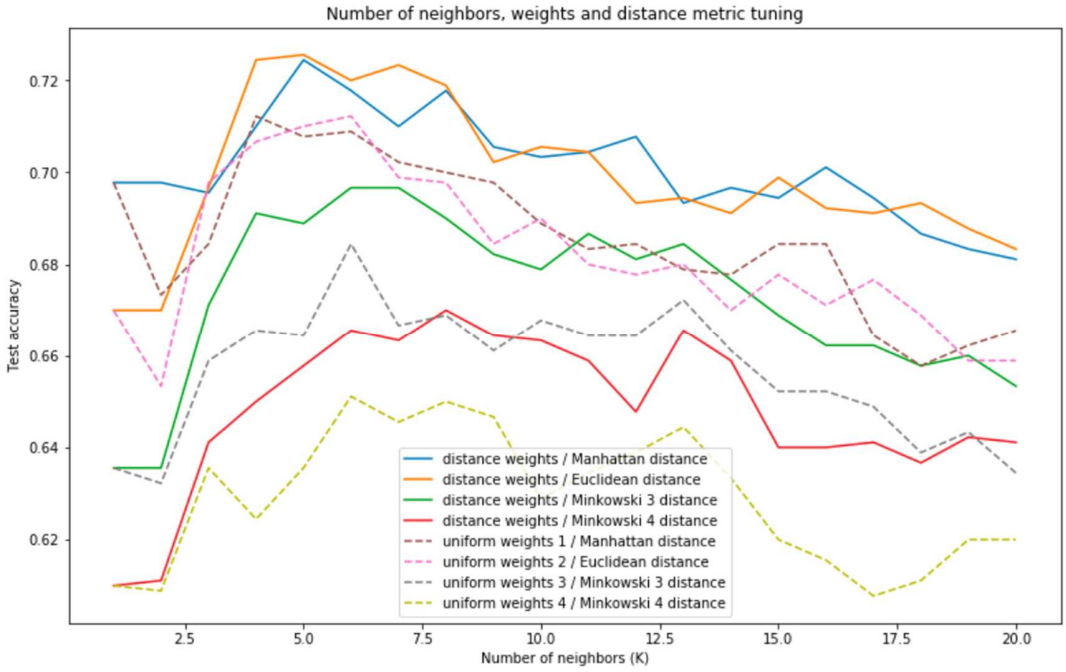


Рисунок 17 - Настройка гиперпараметров K-NN

3.3.5 Оптимизация метода опорных векторов (Support Vector Machine)

Наиболее подходящим типом ядра, достигающим точности 73%, выглядит ядро rbf (рисунок 18). Чтобы минимизировать переобучение, выбирается относительно низкое значение параметра штрафа C, здесь 2.

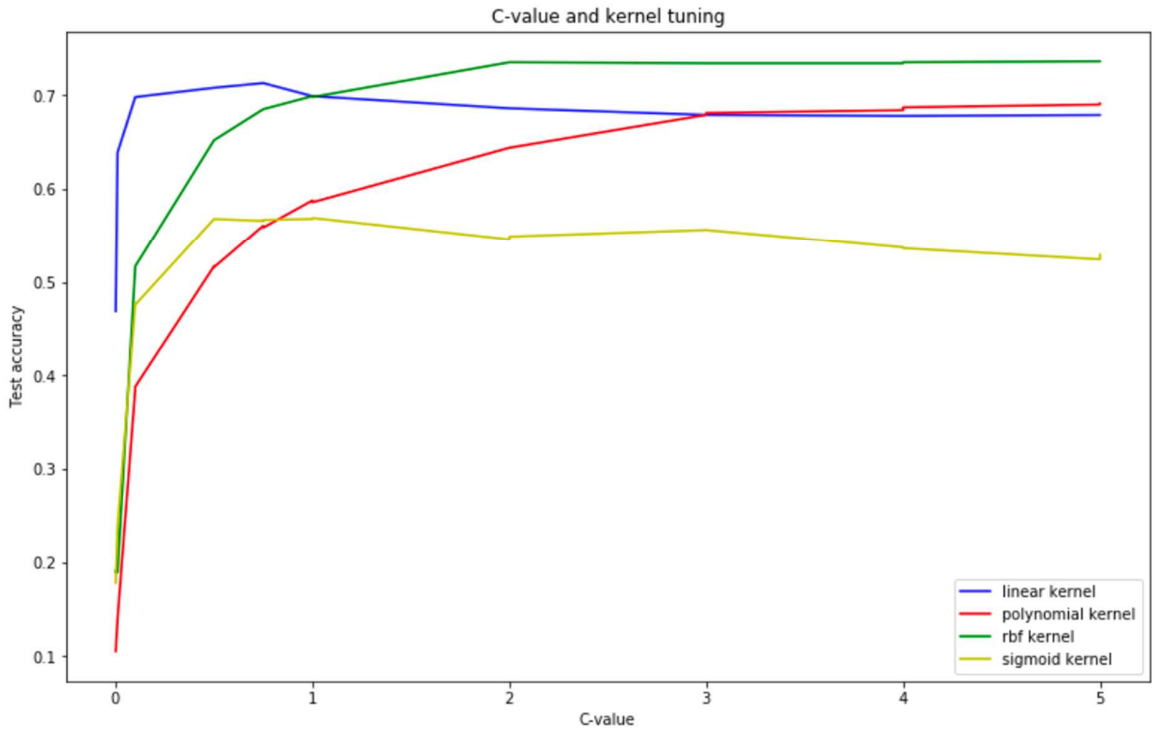


Рисунок 18 - Настройка гиперпараметров SVM

3.3.6 Нейронная сеть прямого распространения

Различные гиперпараметры нейронных сетей были протестированы параллельно. Оптимизированная модель будет состоять из трех скрытых слоев (с размерами 256, 128 и 64 соответственно). Кроме того, обучение будет проводиться на пакете размером 64 и с числом итераций 100 эпох. Для избежания переобучения будут использованы методы отсева (dropout) и ранней остановки (early stopping). Для других основных параметров будет использоваться гиперболический тангенс (tanh) в качестве функции активации, постоянная скорость обучения установлена на 0.001, импульс равен 0.9, и наконец, используется решатель adam.

3.3.7 Общие результаты после настройки гиперпараметров

По результатам настройки гиперпараметров (Таблица 8) получены следующие результаты:

Таблица 8 - Точность тестирования на наборе данных GTZAN / FMA до и после настройки гиперпараметров.

	До настроки	**После настройки**
Логистическая регрессия	68%	70%
Дерево решений	48%	54%
Случайный лес	60%	73%
Adaboost	31%	41%
K-NN	66%	72%
Метод опорних векторов (SVM)	68%	75%
Наивный Баейс	8%	8%
Нейронная сеть прямого распространения (FFNN)	70%	74%

3.4 Выбор признаков

Для определения подмножества признаков, наиболее подходящих для каждой модели, были использованы три метода типа обертки. Оценка производительности основана на точности. Тесты проводились с использованием кросс-валидации при k = 5. Как показывают полученные результаты в Таблице 9, в большинстве моделей наиболее точным подмножеством признаков является двунаправленный метод, который требует больше времени и вычислительных ресурсов.

Таблица 9 - Точность тестирования на наборе данных GTZAN / FMA

	Точность прямого отбора (Количество признаков)	**Точность обратного исключения (Количество признаков)**	**Точность двунаправленного исключения (Количество признаков)**
Логистическая регрессия	71% (45)	72% (40)	73% (36)
Дерево решений	58% (50)	60% (27)	61% (28)
Случайный лес	75% (24)	74% (25)	74% (28)
Adaboost	40% (9)	39% (9)	41% (17)
K-NN	74% (36)	75% (36)	76% (27)
Метод опорних векторов (SVM)	78% (40)	79% (40)	80% (35)
Наивный Баейс	62% (25)	62% (24)	63% (27)
Нейронная сеть прямого распространения (FFNN)	75% (41)	75% (42)	75% (38)

Первое интересное наблюдение состоит в том, что точность метода наивного байеса может значительно повыситься при уменьшении количества признаков. Фактически, можно достичь точности в 63%, используя менее половины признаков. Выбор всех признаков увеличивает вероятность зависимости между каждым из них и, следовательно, снижает производительность этого алгоритма. Из рисунка 19 видно, что точность уменьшается с увеличением числа признаков. Поскольку использовалась кросс-валидация, кривая на графиках соответствует средней точности, а синяя полоса вокруг нее соответствует дисперсии. Тем не менее, точность этого метода все равно слишком низкая по сравнению с другими моделями.

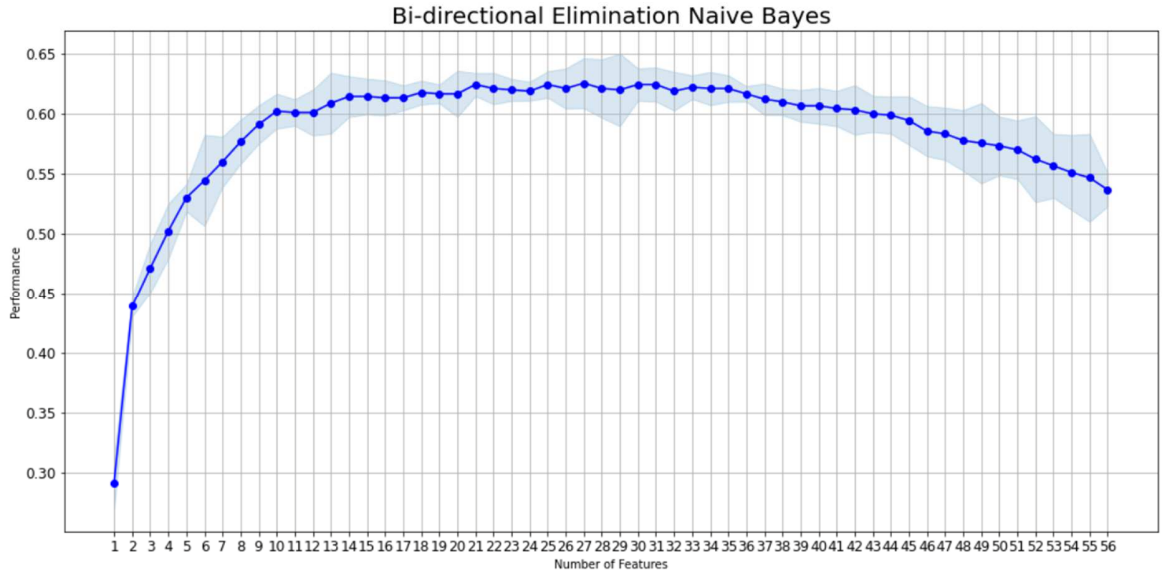


Рисунок 19 - Выбор признаков - наивный байесовский подход

Наиболее перспективными моделями сейчас являются метод опорных векторов (Support Vector Machine, SVM) (рисунок 20) и метод ближайших соседей (k-Nearest Neighbors, k-NN) (рисунок 21). Однако результаты, полученные с использованием метода случайного леса (Random Forest) (рисунок 22) и логистической регрессии (Logistic Regression) (рисунок 23), также являются значимыми. Число признаков тщательно выбирается для максимизации точности при прогнозировании жанров и одновременно ограничения разброса результатов.

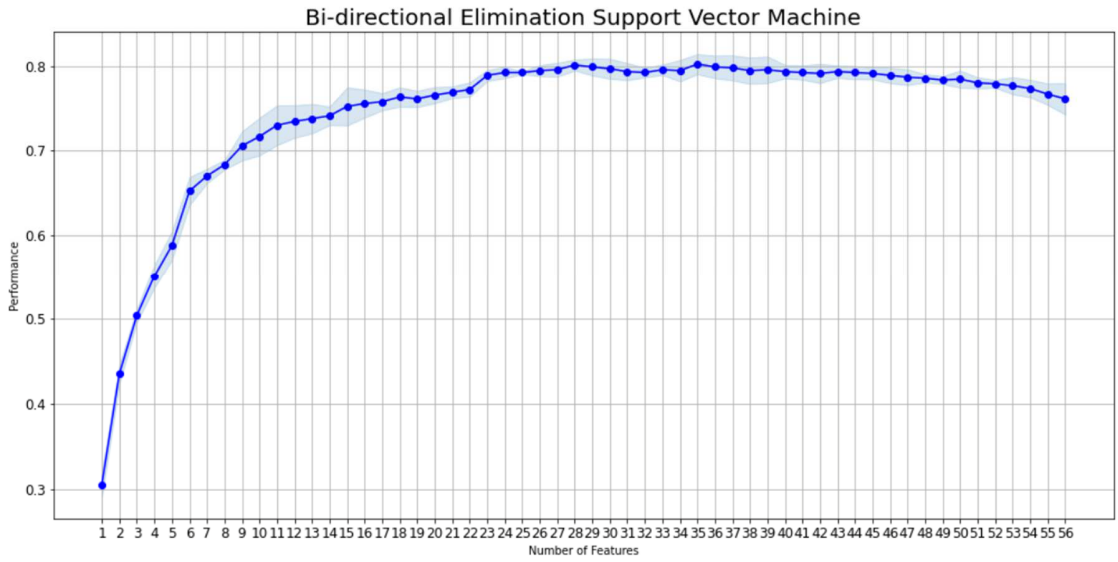


Рисунок 20 - Выбор признаков - SVM

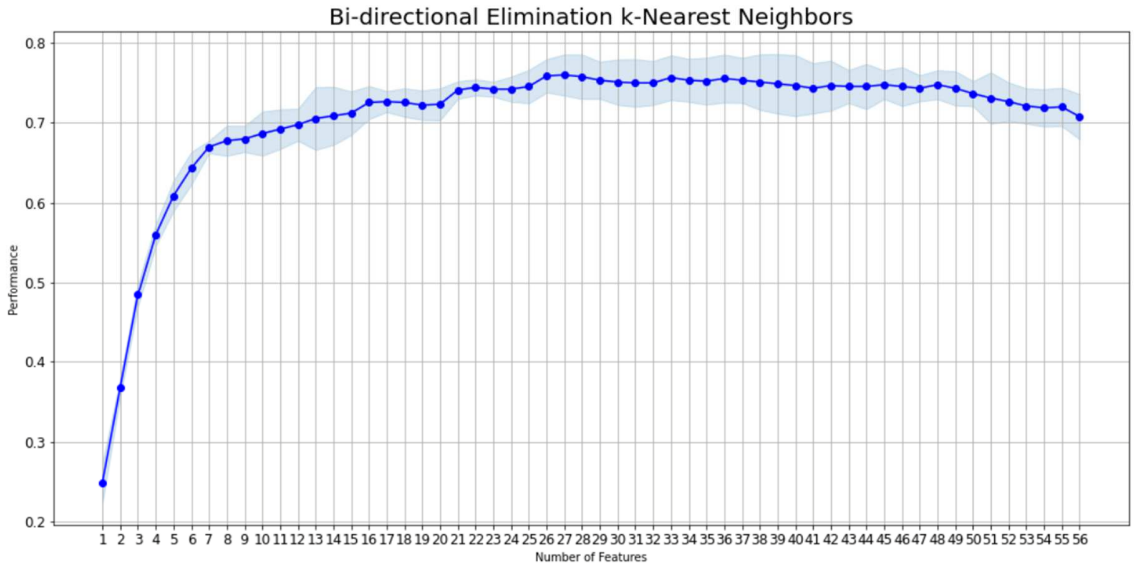


Рисунок 21 - Выбор признаков - k-NN

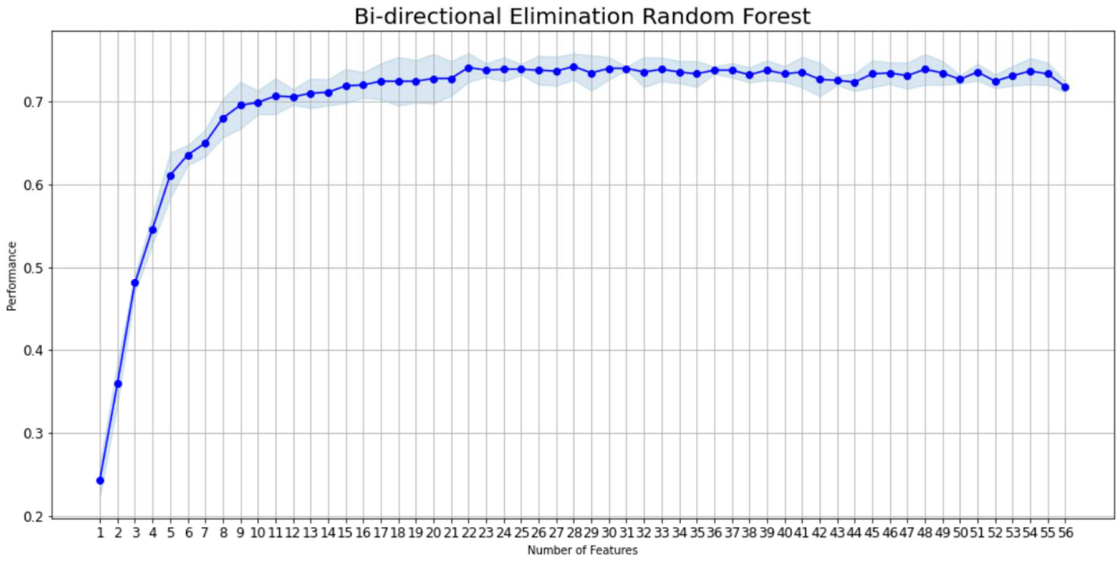


Рисунок 22 - Выбор признаков - Случайный лес

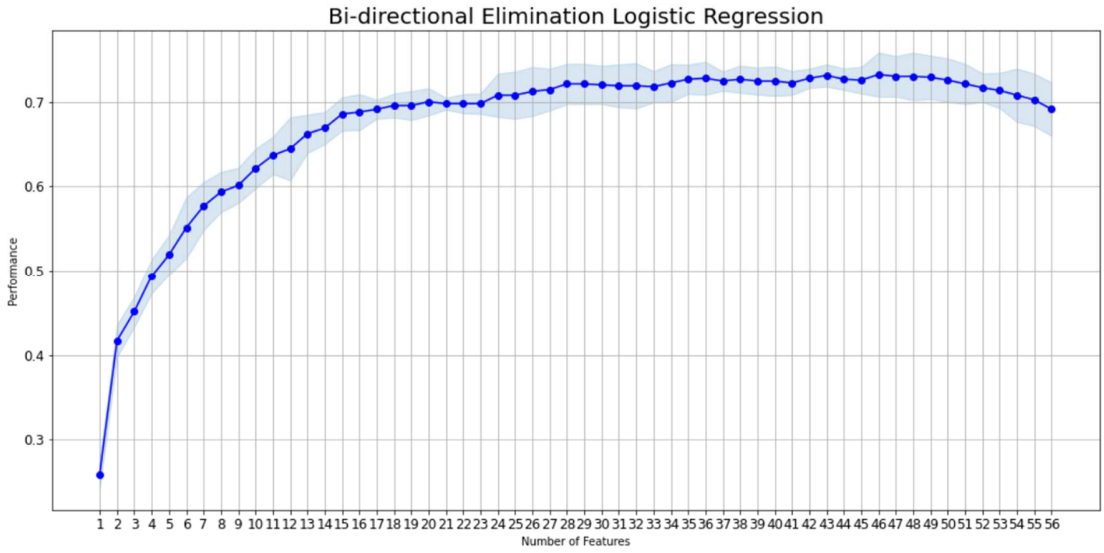


Рисунок 23 - Выбор признаков - Логистическая регрессия

Для удержания только наиболее релевантных моделей, решающее дерево (Decision Tree), Adaboost и метод наивного байеса больше не будут участвовать в конкурсе, проводимом в рамках данной диссертации.

3.4.1 Самые важные признаки

Зная оптимальное подмножество признаков для каждой модели, можно увидеть, какие признаки в среднем наиболее полезны. Результаты представлены на рисунке 24. Прежде всего, как указано в статьях, представляющих воспринимаемые признаки (громкость, воспринимаемая резкость, воспринимаемая ширина [37] и коэффициенты MFCC [34]), эти признаки особенно мощны и эффективны для рекомендации музыки. Относительно коэффициентов MFCC можно отметить, что первые коэффициенты являются наиболее важными, а наивысшие коэффициенты используются реже в конечных подмножествах. Кроме того, события начала (характеризующие ритм) и танцевальность также являются очень сильными признаками, которые всегда используются независимо от модели. Следует отметить, что хотя использование высокоуровневых признаков вызывает много споров, использование танцевальности здесь представляется релевантным выбором. Спектральные признаки, а также скорость пересечения нуля и высота звука, используются менее часто.

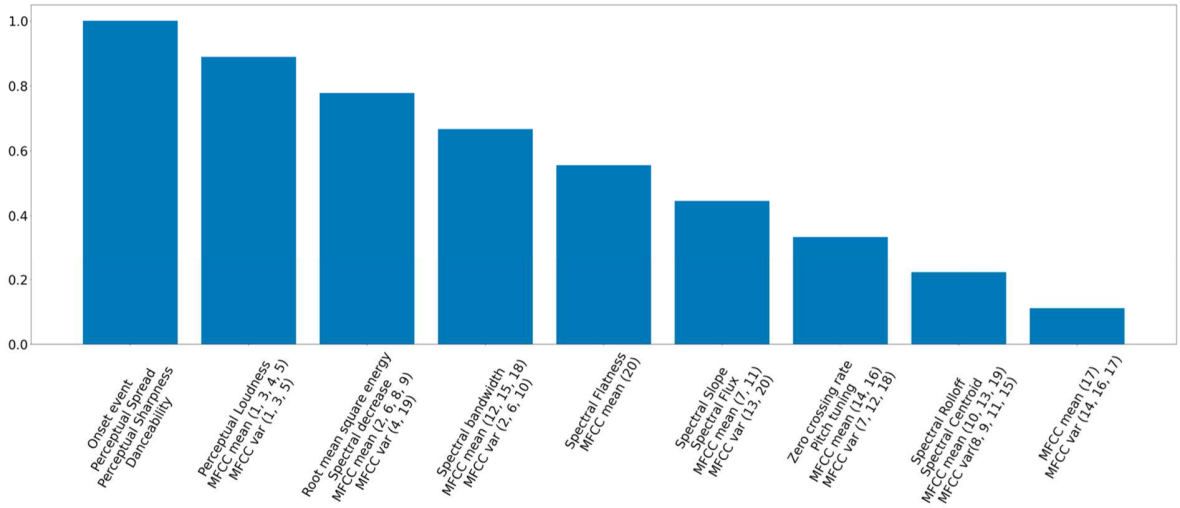


Рисунок 24 - Частота использования признаков

Статистическое сравнение частоты появления признаков в конечных подмножествах с средней точностью, полученной при использовании самих признаков (представленных в таблице 10), является важным. Если рассмотреть только коэффициенты MFCC (их среднее и дисперсию), то точность составляет 52%. Проводились тесты с моментами третьего и четвертого порядка, но это не принесло улучшений. Пересечение двух результатов показывает, что хотя танцевальность присутствует во всех подмножествах, при использовании ее в одиночку точность составляет всего 20%. Вывод заключается в том, что

этот высокоуровневый признак предоставляет информацию, отличную от других признаков более низкого уровня и не является избыточной. То же самое, хотя и в меньшей степени, относится к энергии среднеквадратического значения. Хотя спектральные признаки обеспечивают более высокую точность, они не всегда считаются необходимыми, так как они не предоставляют полезных деталей для классификации музыки.

Таблица 10 - Точность при использовании только одного признака

Признак	**Средняя точность**
MFCC среднее и дисперсия	52%
MFCC среднее	48%
MFCC дисперсия	35%
Спектральный наклон	33%
Воспринимаемая резкость	31%
Спектральное убывание	32%
Частота спада спектра	29%
Спектральный поток	28%
Моменты начала звуковых событий	28%
Воспринимаемая распространенность	27%
Спектральный центроид	26%
Система настройки	25%
Громкость	24%
Спектральная плоскость	23%
Среднеквадратическая энергия	21%
Танцевальность	20%
Частота пересечения нуля	18%

3.5 Аугментация данных

Был проведен ряд тестов с использованием различных методов увеличения данных (добавление шума, сдвиг музыки по времени, изменение скорости, высоты звука и т. д.) отдельно и в комбинации. Таблица 11 показывает результаты только для логистической регрессии и k-Nearest Neighbors, хотя результаты для других методов были схожими. Первым выводом, которым можно сделать, является то, что метод сдвига музыки по времени не меняет результаты на тестовой выборке. Это можно объяснить тем, что извлекаемые

признаки в основном усредняются по времени. Поэтому акцент делается на других трех методах: добавление шума, изменение скорости и высоты звука, сначала отдельно, а затем вместе. К сожалению, производительность не улучшается, наоборот, повышается переобучение - результаты на обучающей выборке очень хорошие, а на тестовой выборке они ухудшаются. Поэтому выбор окончательного метода будет сделан без использования увеличения данных.

Таблица 11 - Точности с использованием аугментации данных

	Шум		**Сдвиг по t**		**Скорость**		**Высота звука**		**Шум, высота звука и скорость**	
	train	**test**	**train**	**test**	**train**	**test**	**train**	**test**	**train**	**test**
Лог. Рег.	98%	72%	92%	73%	99%	68%	98%	65%	100%	65%
k-NN	91%	72%	80%	76%	93%	69%	92%	67%	93%	68%

3.6 Финальные примеры рекомендаций

После извлечения признаков, наиболее точно представляющих музыку, можно прослушать рекомендации. После всех этих шагов становится ясно, что наиболее эффективный и надежный метод, как по оценкам в матрицах путаниц, так и по результатам прослушивания человеком, - это метод опорных векторов. Именно этот метод будет использоваться в этой финальной части. Матрица ошибок представлена на рисунке 25. Процентные значения лучше, а ошибки кажутся вполне простительными. Теперь проведем прослушивание для проверки точности предсказаний.

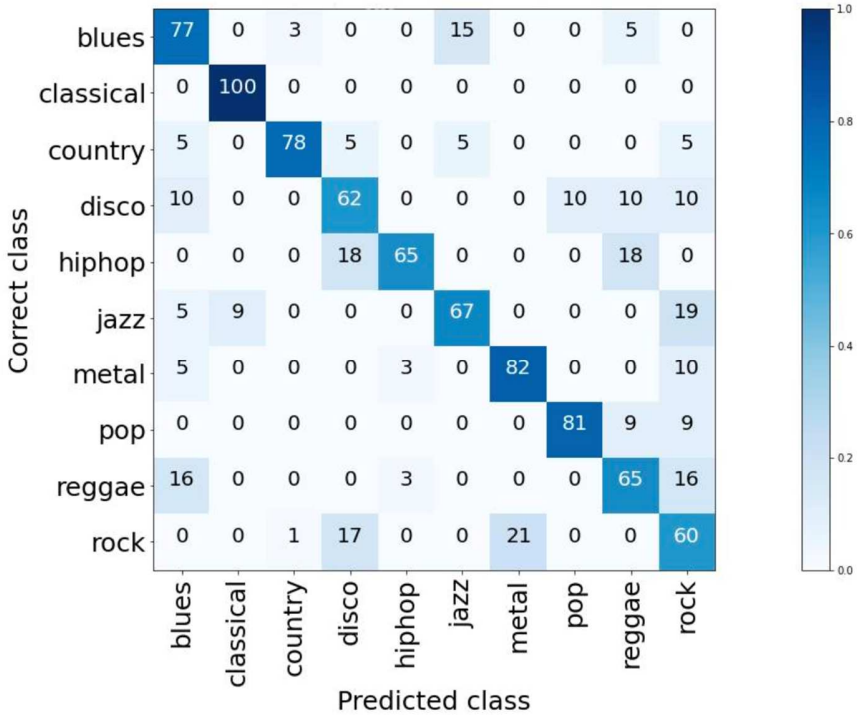


Рисунок 25 - Финальная матрица путаниц для SVM

Эта модель прошла через различные тесты, и вот результаты для стилей Classical, Blues и Rock.

В первом примере использовалась классическая музыка. Как и ожидалось согласно матрице ошибок, жанр музыки был распознан, и рекомендованные треки (таблица 12) являются достаточно похожими классическими композициями, в основном с использованием струнных инструментов.

Таблица 12 - Пример рекомендованных песен 1 (Classical)

	Название	**Артист**	**Жанр**
Выбранная композиция	Spring Allegro III	Vivaldi	Classical
Рекомендованные песни	Spring Allegro II Violin Concerto Allegro assai con spirito Sonata XIII	Vivaldi Karol Szymanowski F.J. Haydn Giovanni Gabrieli	Classical Classical Classical Classical

Во втором примере была взята музыка в стиле Blues. Из таблицы 13 видно, что первая рекомендованная музыка принадлежит тому же исполнителю, две другие - тому же жанру, и еще одна - Reggae. Несмотря на то, что они принадлежат к разным жанрам, эти рекомендации кажутся вполне верными.

Таблица 13 - Пример рекомендованных песен 2 (Blues)

	Название	**Артист**	**Жанр**
Выбранная композиция	One More Night	Hot Toddy	Blues
Рекомендованные песни	Rescue Me Hobo's Son Could You Be Loved I'm Bad Like Jesse James	Hot Toddy Kelly Joe Phelps Bob Marley John Lee Hooker	Blues Blues Raggae Blues

Наконец, взята музыка в стиле Rock, и результаты более разнообразны. Одна из песен - диско, а половина рекомендаций - металлическая музыка (показано в таблице 14). Диско-музыка довольно близка к исходной песне, а металл является поджанром рока. После прослушивания эти результаты кажутся также релевантными.

Таблица 14 - Пример рекомендации песен 3 (Rock)

	Название	**Артист**	**Жанр**
Выбранная композиция	Like Swimming	Morphine	Rock
Рекомендованные песни	Caught in the Middle I Know You Pt. III High Energy Freedom	DIO Morphine Evelyn Thomas Rage Against The Machine	Metal Rock Disco Metal

Большинство проведенных прослушиваний кажутся релевантными или по крайней мере не совсем непонятными. Однако для каждого жанра потребуется внешняя экспертиза для получения более точных оценок и предсказаний модели с человеческой оценкой.

ЗАКЛЮЧЕНИЕ

Подводя итог, в терминах количественных результатов, окончательная точность классификации по жанрам с использованием модели Support Vector Machine составляет 80% на тестовом наборе данных. Однако стоит отметить, что другие модели, в частности нейронная сеть и случайный лес, показывают многообещающие результаты. Это значение может быть достигнуто после оптимизации гиперпараметров модели, а также этапа выбора признаков. Эти два метода позволяют существенно повысить точность, ограничивая переобучение. Следует учитывать, что это значение является лишь показателем для части проекта, связанной с классификацией. Матрица путаницы (Рис. 25 в предыдущем разделе) ясно предоставляет нам больше информации о результатах. То, на что мы обращаем внимание в матрице путаницы, это диагональ темного цвета, характеризующая высокую степень восстановления исходного жанра, в то время как остальные случаи должны быть как можно более белыми. Окончательная матрица соответствует желаемому. Из матрицы следует, что некоторые жанры легче для нашей системы различить, чем другие. Действительно, классический жанр всегда определяется. В жанрах поп, металл, кантри и блюз система также довольно эффективна и находит жанр четыре раза из пяти. В отличие от этого, рок и диско сложнее определить. Кроме того, хорошая классификация - это не единственный показатель, на котором следует основываться для рекомендации. Действительно, эти категории не являются взаимоисключающими, и некоторые являются поджанрами других, например металл является поджанром рока. Самые полезные признаки для количественной оценки сходства музыки - это восприятие (громкость, резкость, ширина и MFCC), ритм и танцевальность. Действительно, только восприятие вносит более 60% точности с использованием метода SVM. Следует также отметить, что танцевальность, высокоуровневый признак, предоставляет информацию, существенно отличающуюся от более классических признаков, как физических, так и музыкальных, что позволяет явно улучшить производительность (примерно на 10%).

Результаты прослушивания обнадеживающие. Следует отметить, что эти прослушивания были оценены только одним человеком (мной, автором), и оценка по группе лиц позволила бы лучшую оценку. Однако прогресс между прослушиваниями в начале и после оптимизации явный. В то время как система ранее ошибочно рекомендовала блюзовые песни для совершенно другой поп-музыки, рекомендации теперь более логичные и

последовательные. Система в основном рекомендует музыку в рамках того же жанра (иногда от того же исполнителя), и реже - более различные композиции. Во всех случаях есть сходство в общем настроении или тональности выбранной музыки, ее ритме, мелодии или даже инструментации.

В заключении стоит отметить, что для того чтобы учесть вкусы пользователя, возможны несколько подходов, большинство из которых уже объяснены в разделе "Введение". В данной диссертации был выбран и реализован метод контентной рекомендации. Полученные результаты показывают, что можно довольно эффективно изучать вкусы пользователя на основе предыдущей прослушанной им музыки для рекомендации новых композиций. Принцип заключается в том, чтобы извлечь как можно больше информации из аудиозаписи в формате mp3 или wav. Музыкальная композиция рассматривается как свидетельство музыкальных предпочтений пользователя. Извлеченная информация может быть музыкальной (высота звука, ритм), физической, полученной с помощью обработки сигналов (спектральное уменьшение), или признаками более высокого уровня (танцевальность). Важно убедиться, что информация является актуальной и нераздутой с помощью алгоритмов выбора признаков. Из лучшего набора признаков алгоритм машинного обучения (в данном случае, метод опорных векторов) классифицирует музыку по жанрам. После этапа классификации следует этап рекомендации. Он заключается в поиске четырех других композиций, которые будут рекомендованы пользователю. Эти композиции будут теми, которые наиболее близки по признакам. Одной из самых сложных задач в этом проекте является оценка. Довольно сложно измерить производительность нашей системы. Исходные результаты основывались на различных классификаторах, изученных в проекте. В то же время проводились прослушивания для оценки рекомендаций и моделей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Rich E. User modeling via stereotypes* // Cognitive Science. – 1979. – Vol. 3, № 4. – P. 329–354.

2. Burke R., Ramezani M. Matching Recommendation Technologies and Domains. – 2011. – P. 367–386.

3. Schedl M. Deep learning in music recommendation systems // Frontiers in Applied Mathematics and Statistics. – 2019. – Vol. 5. – P. 44.

4. Schedl M., Knees P., Gouyon F. New Paths in Music Recommender Systems Research. – 2017.

5. McNee S. M., Riedl J., Konstan J. A. Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems // CHI EA '06. – 2006. – P. 1097–1101.

6. Knees P., Schedl M. Music retrieval and recommendation – a tutorial overview // Proceedings of the 38th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR). – 2015.

7. Knees P., Schedl M. Music Similarity and Retrieval: An Introduction to Audio- and Web-based Strategies. – 2016.

8. P´erez-Marcos J., Batista V. Recommender system based on collaborative filtering for Spotify's users. – 2018. – P. 214–220.

9. Johnson C. From idea to execution: Spotify's discover weekly. – November 2015.

10. Schafer B., Frankowski D., Herlocker J., Sen S. Collaborative Filtering Recommender Systems. – 2007.

11. Elahi M., Ricci F., Rubens N. A survey of active learning in collaborative filtering recommender systems // Computer Science Review. – 2016.

12. Kaminskas M., Ricci F. Contextual music information retrieval and recommendation: State of the art and challenges // Computer Science Review. – 2012. – Vol. 6, № 2. – P. 89–119.

13. Gu Z., Guo L., Liu T. Music genre classification via machine learning.

14. Park H-A. An introduction to logistic regression: From basic concepts to interpretation with particular attention to nursing domain // Journal of Korean Academy of Nursing. – 2013. – Vol. 43. – P. 154–164.

15. Sigmoid-function-2. – URL: https://commons.wikimedia.org/wiki/File:Sigmoid- function-2.svg (access date: 11.04.2023).

16. Terry-Jack M. Stips and tricks for multi-class classification. – URL: https://medium.com/@b.terryjack/tips-and-tricks-for-multi-classclassification-c184ae1c8ffc (access date: 11.04.2023).
17. de Azevedo B. C. F., Bressan G. M., Lizzi E. Ap. S. A decision tree approach for the musical genres classification. – 2017.
18. Quinlan J. R. Induction of decision trees // Machine Learning. – 1986. – Vol. 1, № 1. – P. 81–106.
19. Wang Z., Zhang J., Verma N. Realizing low-energy classification systems by implementing matrix multiplication directly within an adc // IEEE Transactions on Biomedical Circuits and Systems. – 2015. – Vol. 9. – P. 1–1.
20. Cunningham P., Delany S. k-nearest neighbour classifiers // Mult Classif Syst. – 2007.
21. Indyk P., Motwani R. Approximate nearest neighbors: Towards removing the curse of dimensionality // Conference Proceedings of the Annual ACM Symposium on Theory of Computing. – 2000. – P. 604-613.
22. Thiruvengatanadhan R. Speech/music classification using mfcc and knn // International Journal of Computational Intelligence Research. – 2017. – Vol. 13, № 10. – P. 2449–2452.
23. Xu C., Maddage N., Shao X., Cao F., Tian Q. Musical genre classification using support vector machines // V – 429. – 2003. – Vol. 5.
24. Zisopoulos H., Karagiannidis S., Demirtsoglou G., Antaris S. Content-based recommendation systems. – 2008.
25. Fisher R. A. The use of multiple measurements in taxonomic problems // Annals of Eugenics. – 1936. – Vol. 7, № 2. – P. 179–188.
26. Li T., Ogihara M., Li Q. A Comparative Study on Content- Based Music Genre Classification. – 2003. – P. 282–289.
27. Masood S. Genre classification of songs using neural network. – 2014.
28. Artificial neural network. – URL: https://en.wikipedia.org/wiki/Artificial_neural_network (access date: 11.04.2023).
29. Mehlig B. Artificial neural networks. – 2019.
30. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting // Journal of Machine Learning Research. – 2014. – Vol. 15, № 56. – P. 1929–1958.

31. Early stopping with pytorch to restrain your model from overfitting. – URL: https://mc.ai/early-stopping-with-pytorch-to-restrain-your-model-from-overfitting/ (access date: 11.04.2023).
32. Peeters G. A large set of audio features for sound description (similarity and classification) in the cuidado project. – 2004.
33. McKinney M., Breebaart J., Prof (wy. Features for audio and music classification. – 2003.
34. Jensen J., Christensen M., Murthi M., Jensen S. Evaluation of mfcc estimation techniques for music similarity. – 2006.
35. Zhu L., Chen L., Zhao D., Zhou J., Zhang W. Emotion recognition from Chinese speech for smart affective services using a combination of svm and dbn // Sensors. – 2017. – Vol. 17, № 7.
36. Luo X., Liu X., Tao R., Shi Y. Content-based retrieval of music using mel frequency cepstral coefficient (mfcc). – 2015.
37. Shin S.-H. Comparative study of the commercial software for sound quality analysis // Acoustical Science and Technology. – 2008. – Vol. 29. – P. 221–228.
38. Moravec O., Stepanek J. Possibility of application of objective psychoacoustic metrics on musical signals. – 2006.
39. Scaringella N., Zoia G., Mlynek D. Automatic genre classification of music content: a survey // IEEE Signal Processing Magazine. – 2006. – Vol. 23, № 2. – P. 133–141.
40. Vatolkin I., Theimer W. Introduction to methods for music classification based on audio data. – 2020.
41. Klapuri A. P., Eronen A. J., Astola J. T. Analysis of the Meter of Acoustic Musical Signals. – 2004. – P. 342–355.
42. Essentia: an audio analysis library for music information retrieval // International Society for Music Information Retrieval Conference (ISMIR'13). – 2013. – P. 493–498.
43. Karamchandani S., Matodkar P., Iyer S., Gori N. Score Formulation and Parametric Synthesis of Musical Track as a Platform for Big Data in Hit Prediction. – 2018. – P. 363–374.
44. Guyon I., Elisseeff A. An introduction to variable and feature selection // J. Mach. Learn. Res. – 2003. – Vol. 3. – P. 1157–1182.
45. Duda R. O., Hart P. E., Stork D. G. Pattern Classification. – Wiley, New York, 2 edition, 2001.
46. Kohavi R., John G. H. Wrappers for feature subset selection // Artificial Intelligence. – 1997. – Vol. 97, № 1. – P. 273 – 324.

47. Bertin-Mahieux T., Ellis D., Whitman B., Lamere P. The million song dataset // Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011). – 2011. – P. 591–596.
48. Hauger D., Kosir A., Tkalcic M., Schedl M. The million musical tweets dataset: What can we learn from microblogs. – 2013.
49. Aggarwal A. MSD Getting the dataset. – URL: http://millionsongdataset.com/pages/getting-dataset/ (access date: 11.04.2023).
50. Schedl M. The lfm-1b dataset for music retrieval and recommendation. – 2016. – P. 103–110.
51. Hawthorne C., Stasyuk A., Roberts A., Simon I., Huang C.-Z. A., Dieleman S., Elsen E., Engel J., Eck D. Enabling factorized piano music modeling and generation with the MAESTRO dataset. – 2019.
52. Sturm B. The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use. – 2013.
53. Defferrard M., Benzi K., Vandergheynst P., Bresson X. Fma: A dataset for music analysis. – 2016.
54. Vandergheynst P., Bresson X., Defferrard M., Benzi K. Fma A Dataset For Music Analysis. – URL: https://github.com/mdeff/fma (access date: 11.04.2023).
55. McFee B., Humphrey E. J., Bello J. P. A software framework for musical data augmentation. – 2015.
56. McFee B., Raffel C., Liang D., Ellis D. P. W., McVicar M., Battenberg E., Nieto O. librosa: Audio and music signal analysis in python. – 2015.
57. Bogdanov D., Wack N., Gómez E., Gulati S., Herrera P., Mayor O., Roma G., Salamon J., Zapata J., Serra X. Essentia: An open-source library for sound and music analysis // Proceedings - 21st ACM International Conference on Multimedia. – 2013.
58. Mathieu B., Essid S., Fillon T., Prado J., Richard G. YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software. – 2010.
59. Song Y., Dixon S., Pearce M. A survey of music recommendation systems and future perspectives. – 2012.
60. Sturm B. L. Classification accuracy is not enough // J. Intell. Inf. Syst. – 2013. – Vol. 41, № 3. – P. 371–406.

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет о проверке на заимствования №1

Автор: Березовский Константин Андреевич
Проверяющий: Романович Ольга Владимировна
Организация: Томский Государственный Университет

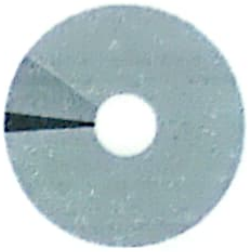
Отчет предоставлен сервисом «Антиплагиат» - http://tsu.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 35
Начало загрузки: 07.06.2023 05:52:15
Длительность загрузки: 00:00:27
Имя исходного файла: Диплом Березовский (2).docx
Название документа: Диплом Березовский (2)
Размер текста: 118 кБ
Символов в тексте: 120942
Слов в тексте: 14282
Число предложений: 1060

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 07.06.2023 05:52:43
Длительность проверки: 00:01:49
Комментарии: не указано
Поиск с учетом редактирования: да
Проверенные разделы: титульный лист с. 1, основная часть с. 2-5,9-79, содержание с. 6-8, библиография с. 80-83
Модули поиска: ИПС Адилет, Библиография, Сводная коллекция ЭБС, Интернет Плюс*, Сводная коллекция РГБ, Цитирование, Переводные заимствования (RuEn), Переводные заимствования по eLIBRARY.RU (EnRu), Переводные заимствования по коллекции Гарант: аналитика, Переводные заимствования по коллекции Интернет в английском сегменте, Переводные заимствования по Интернету (EnRu), Переводные заимствования по коллекции Интернет в русском сегменте, Переводные заимствования издательства Wiley , eLIBRARY.RU, СПС ГАРАНТ: аналитика, СПС ГАРАНТ: нормативно-правовая документация, Медицина, Диссертации НББ, Коллекция НБУ, Перефразирования по eLIBRARY.RU, Перефразирования по СПС ГАРАНТ: аналитика, Перефразирования по Интернету, Перефразирования по Интернету (EN), Перефразированные заимствования по коллекции Интернет в английском сегменте, Перефразированные заимствования по коллекции Интернет в русском сегменте, Перефразирования по коллекции издательства Wiley , Патенты СССР, РФ, СНГ, СМИ России и СНГ, Шаблонные фразы, Модуль поиска "tsu", Кольцо вузов, Издательство Wiley, Переводные заимствования



СОВПАДЕНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
3,06%	0%	7,61%	89,33%

Совпадения — фрагменты проверяемого текста, полностью или частично сходные с найденными источниками, за исключением фрагментов, которые система отнесла к цитированию или самоцитированию. Показатель «Совпадения» – это доля фрагментов проверяемого текста, отнесенных к совпадениям, в общем объеме текста.

Самоцитирования — фрагменты проверяемого текста, совпадающие или почти совпадающие с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа. Показатель «Самоцитирования» – это доля фрагментов текста, отнесенных к самоцитированию, в общем объеме текста.

Цитирования — фрагменты проверяемого текста, которые не являются авторскими, но которые система отнесла к корректно оформленным. К цитированиям относятся также шаблонные фразы; библиография; фрагменты текста, найденные модулем поиска «СПС Гарант: нормативно-правовая документация». Показатель «Цитирования» – это доля фрагментов проверяемого текста, отнесенных к цитированию, в общем объеме текста.

Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.

Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.

Оригинальный текст — фрагменты проверяемого текста, не обнаруженные ни в одном источнике и не отмеченные ни одним из модулей поиска. Показатель «Оригинальность» – это доля фрагментов проверяемого текста, отнесенных к оригинальному тексту, в общем объеме текста.

«Совпадения», «Цитирования», «Самоцитирования», «Оригинальность» являются отдельными показателями, отображаются в процентах и в сумме дают 100%, что соответствует полному тексту проверяемого документа.

Обращаем Ваше внимание, что система находит текстовые совпадения проверяемого документа с проиндексированными в системе источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности совпадений или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в тексте	Доля в отчете	Источник	Актуален на	Модуль поиска	Комментарии
[01]	6,85%	6,85%	не указано	29 Сен 2022	Библиография	
[02]	1,65%	0,03%	https://www.ismir2020.net/assets/img/proceedings/20... https://ismir2020.net	25 Янв 2023	Интернет Плюс*	
[03]	1,12%	1,12%	Спектральные дескрипторы https://docs.exponenta.ru	29 Мая 2023	Интернет Плюс*	
[04]	1,09%	0,79%	диплом.pdf	02 Июн 2022	Модуль поиска "tsu"	
[05]	0,96%	0,23%	СТРУКТУРА ЛАНДШАФТОВ И КАЧЕСТВЕННАЯ ОЦЕНК...	09 Июн 2022	Модуль поиска "tsu"	
[06]	0,95%	0,21%	VKR_Subach-2.docx	16 Янв 2022	Модуль поиска "tsu"	
[07]	0,88%	0%	ВКР Субач-2.docx	16 Янв 2022	Модуль поиска "tsu"	
[08]	0,84%	0%	Силур-девонские кораллы Горного Алтая из коллек...	05 Июн 2022	Модуль поиска "tsu"	
[09]	0,76%	0,76%	не указано	29 Сен 2022	Шаблонные фразы	

Ознакомлен(а) с результатом ознакомления Заметин А.В