

МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ «СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ШКОЛА №1(ПОДРАЗДЕЛЕНИЕ Николая Громова)»

Московская область, г.Сергиев Посад, ул.Н.Громова, д.1, 141300

тел.8-496-540-20-91, 8-496-540-59-09, электр.почта: scool22sp@mail.ru

Проект

Природа звука, его представления и характеристики.
Алгоритмический анализ звука.

Векторное представление музыки и построение
рекомендательной системы.

Подготовили ученики 10Д класса
Руководитель темы

Наумов Н.А; Полбин А. И.
Белова Е. В.

г. Сергиев Посад 2026

Содержание

1. Введение.....	4
1.1 Цели и задачи.....	4
1.2 Актуальность.....	4
2. Физические основы звука.....	5
2.1 Механические колебания и волны.....	5
2.1.1 Основные характеристики звука.....	5
2.2 Восприятие звука ухом человека.....	6
2.2.1 Строение слухового аппарата.....	6
2.2.2 Психоакустические особенности.....	6
2.3 От волны к цифровому сигналу.....	7
2.3.1 Аналого-цифровое-преобразование (АЦП).....	7
3. Методология исследования.....	9
3.1 Теоретические основы рекомендательных систем.....	9
3.1.1 Векторные представления (эмбединги).....	9
3.1.2 Метрики сходства.....	10
3.1.3 Типы рекомендательных систем.....	11
3.1.3.1 Контентная фильтрация (Content-Based Filtering).....	12
3.1.3.2 Коллаборативная фильтрация (Collaborative Filtering).....	12
3.1.3.3 Гибридная система.....	12
3.2 Сравнение подходов и выбор типа рекомендательной системы для проекта.....	13
3.2.1 Контентная фильтрация: преимущества и ограничения.....	13
Преимущества:.....	13
3.2.2 Коллаборативная фильтрация: преимущества и ограничения.....	14
3.2.3 Обоснование выбора контентной фильтрации в рамках проекта.....	15
3.3 Выбор набора музыки (датасета).....	16
3.3.1 Критерии выбора.....	17
3.3.2 Сравнение доступных датасетов.....	17
3.3.3 Обоснование выбора датасета.....	18
3.3.4 Состав метаданных в FMA Medium.....	19
3.3.5 Предобработка данных.....	19
3.4 Методология извлечения признаков.....	19
3.4.1 Акустические признаки музыкальных треков.....	19
3.4.2 Процесс создания эмбедингов.....	21

4. Технологический стек и архитектура системы.....	22
4.1. Используемые технологии.....	22
4.2 Архитектура системы.....	22
5. Практическая реализация рекомендательной системы.....	24
5.1 Общая структура системы.....	25
5.2 Обработка аудио и формирование эмбединга.....	25
5.3 Алгоритм поиска похожих музыкальных треков.....	27
5.4 Хранение данных и импорт датасета.....	28
5.5 Пользовательский интерфейс и API системы.....	29
5.6 Результаты и ограничения.....	31
Источники.....	33

1. Введение

Проект исследует физическую природу звука, его характеристики и способы измерения. Рассматриваются устройства захвата и воспроизведения звука (микрофоны и динамики), а также методы визуализации сигналов. Практическая часть включает построение простой модели анализа звука, перевод аудиосигнала в векторное представление и реализацию мини-рекомендательной системы для музыки. Проект объединяет знания из физики, информатики и программирования, демонстрируя принципы обработки звука.

1.1 Цели и задачи

Цель: Изучить природу звука и применить полученные знания.

Задачи:

- Изучить физическую природу звука и его основные характеристики.
- Рассмотреть способы измерения и визуализации звуковых сигналов.
- Ознакомиться с устройствами захвата и воспроизведения звука (микрофоны и динамики).
- Исследовать цифровые методы представления звука (векторизация, спектрограммы и др.).
- Реализовать простую модель анализа аудиосигнала.
- Создать мини-рекомендательную систему для музыкальных треков на основе анализа звука.
- Продемонстрировать взаимосвязь физики, информатики и программирования в обработке звуковых данных.

1.2 Актуальность

- Звук – один из основных способов передачи информации и восприятия мира человеком.
- Современные технологии широко используют анализ и обработку звука (музыкальные сервисы, голосовые помощники, диагностика).
- Цифровое и векторное представление звука связывает физику и информатику, позволяя применять методы машинного обучения.
- Междисциплинарный подход способствует развитию современных ИИ-систем и технологий обработки сигналов.
- Тема актуальна в контексте цифровизации, автоматизации и развития интеллектуальных аудиосистем.

2. Физические основы звука

Звук является очень важной частью нашей жизни, ведь он окружает нас повсюду.

В этом разделе будет рассмотрен звук с точки зрения физики, и рассмотрены его основные свойства и характеристики.

2.1 Механические колебания и волны

Звук представляет собой механические колебания частиц упругой среды (воздуха, воды, твёрдых тел), распространяющиеся в виде продольных волн давления. Когда источник звука (например, гитарная струна) приходит в движение, он создаёт области сжатия и разрежения в окружающем воздухе, которые передаются от молекулы к молекуле.

2.1.1 Основные характеристики звука

1. Амплитуда (A) - максимальное отклонение частиц от положения равновесия.

Определяет громкость звука. Измеряется в децибелах (дБ). Человеческое ухо воспринимает звуки в диапазоне от 0 дБ (порог слышимости) до 120-130 дБ (болевой порог)

2. Частота (f) - количество полных колебаний в секунду. Определяет высоту тона. Измеряется в герцах (Гц). Диапазон слышимости человека: 20 Гц - 20 000 Гц.

3. Длина волны (λ) - расстояние между двумя соседними точками волны, находящимися в одинаковой фазе колебаний. Связана с частотой и скоростью звука формулой:

$$\lambda = v * T$$

где “v - скорость звука в среде (≈ 343 м/с в воздухе при 20°C).

4. Период “T” - время одного полного колебания: ($T = 1/f$).

5. Скорость звука “v” - зависит от среды: воздух (343 м/с), вода (1482 м/с), сталь (5960 м/с).

2.2 Восприятие звука ухом человека

2.2.1 Строение слухового аппарата

Человеческое ухо состоит из трёх основных отделов:

- Наружное ухо:
 - Ушная раковина - собирает и направляет звуковые волны
 - Слуховой проход - резонатор, усиливает частоты 2000-5000 Гц
 - Барабанная перепонка - преобразует звуковые волны в механические колебания
- Среднее ухо:
 - Система слуховых косточек (молоточек, наковальня, стремечко)
 - Усиливает колебания в 20-25 раз
 - Защищает от слишком громких звуков (акустический рефлекс)
- Внутреннее ухо (улитка):
 - Спиральная структура длиной примерно 35 мм
 - Заполнена жидкостью (перилимфа)
 - Содержит около 3500 волосковых клеток
 - Разные участки резонируют на разных частотах (бас - верхушка, высокие - основание)
 -

2.2.2 Психоакустические особенности

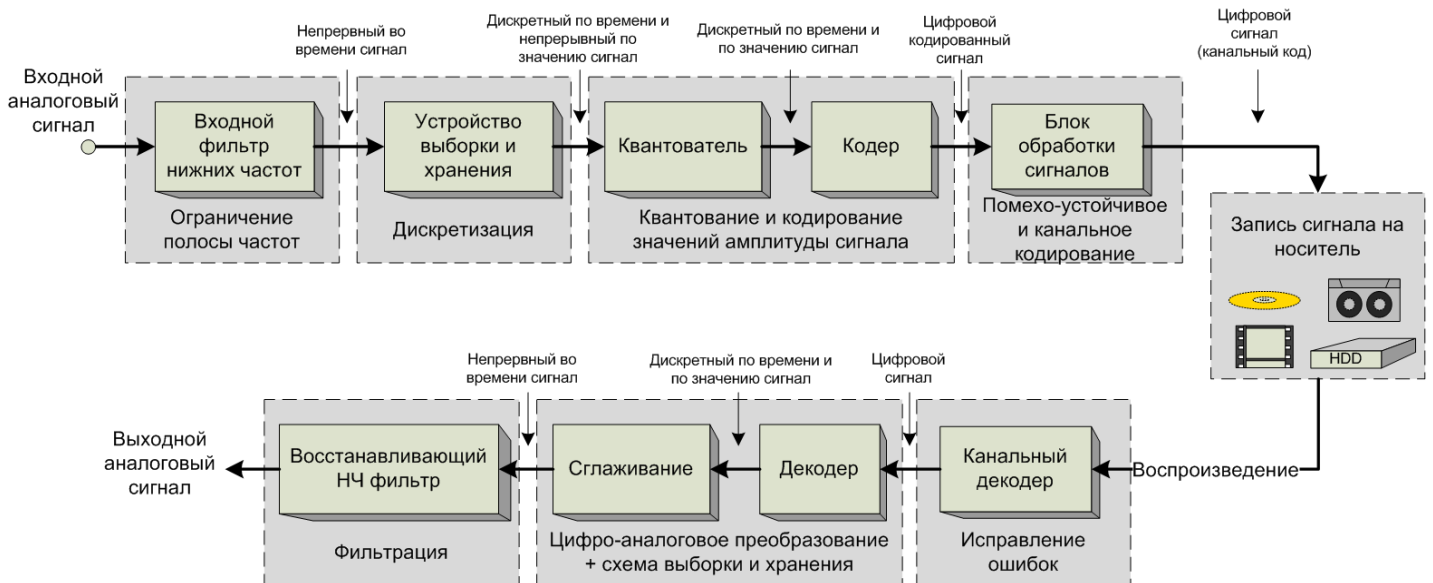
1. Неравномерность восприятия - ухо более чувствительно к средним частотам (1000-4000 Гц), чем к крайним.

2. Логарифмическая шкала громкости - увеличение громкости на 10 дБ воспринимается человеческим ухом как удвоение громкости.

3. Эффект маскировки - громкие звуки маскируют тихие на близких частотах.

4. Критические полосы - диапазон порядка $1/3$ октавы, в пределах которого звуки взаимодействуют.

2.3 От волны к цифровому сигналу



Источник изображения (<https://dic.academic.ru/dic.nsf/ruwiki/1524417>)

2.3.1 Аналого-цифровое-преобразование (АЦП)

Процесс преобразования непрерывного аналогового сигнала в дискретный(цифровой) включает три этапа, и называется импульсно кодовой модуляцией (ИКМ / PCM):

1. Дискретизация (сэмплирование)

Аналоговый сигнал измеряется через равные промежутки времени Δt .

Частота дискретизации:

$$f_d = \frac{1}{T_d}, \text{ где } T_d - \text{ время между отсчетами}$$

Теорема Котельникова-Найквиста: Для точного восстановления сигнала частота дискретизации должна быть как минимум в два раза выше максимальной частоты в сигнале.

Стандартные частоты дискретизации:

- Телефонная связь: 8 кГц
- Аудио CD: 44.1 кГц (захватывает до 22.05 кГц)
- Профессиональная запись: 48 кГц, 96 кГц, 192 кГц

2. Квантование

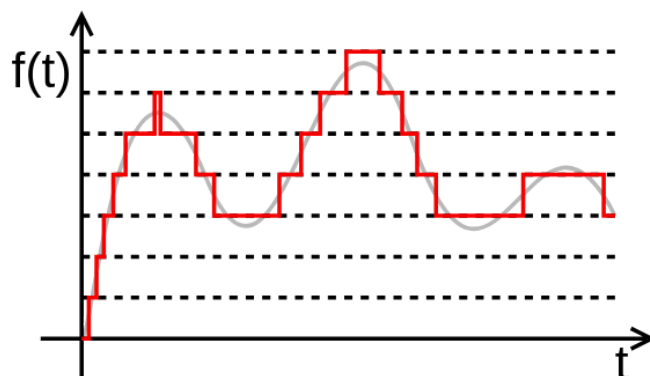
Каждое измеренное значение амплитуды округляется до ближайшего уровня из конечного набора. Глубина квантования (разрядность) определяет количество возможных уровней.

Максимальное количество уровней квантования определяется выбранной разрядностью (по сути степенью двойки).

Например:

8 бит -> 256 уровней

16 бит -> 65 536 уровней, стандарт для CD дисков. Примерный диапазон 96 дБ



3. Кодирование - это процесс при котором полученные квантованные уровни, кодируются в виде чисел в двоичной системе.

Повышая частоту дискретизации и битность квантования, мы улучшаем качество получаемого звука.

Формула для вычисления объёма аудиозаписи словесно будет выглядеть так:

Размер(в битах) = Частота дискретизации(Гц) * Уровень квантования(в битах) *
Продолжительность аудио * Количество каналов(моно/стерео)

3. Методология исследования

В предыдущем разделе были рассмотрены физические основы звука, механизмы человеческого восприятия и принципы цифрового представления аудиосигналов. Эти знания формируют фундамент для создания музыкальной рекомендательной системы. Этот раздел посвящен методологии исследования - переходу от теоретических концепций к практической реализации. Основная задача заключается в разработке алгоритмического подхода, который позволит преобразовать физические характеристики звука в формализованные представления, пригодные для машинного анализа и сравнения между собой.

Здесь мы рассмотрим основные понятия, основы рекомендательных систем, и ещё их типы. Также рассмотрим источники данных, чтобы определиться с источником музыки для анализа и рекомендаций.

Этот раздел послужит связующим звеном между практической и теоретической частями проекта.

3.1 Теоретические основы рекомендательных систем

3.1.1 Векторные представления (эмбеддинги)

В машинном обучении массово используются векторные представления (aka Эмбеддинги). При помощи них описывается положение какой либо сущности(в нашем случае музыкального трека) в абстрактном многомерном пространстве, где близкие семантически(в нашем случае по звучанию) сущности расположены близко друг с другом. Эмбеддинг по сути является вектором, что для компьютера равно фиксированному по длине набору чисел.

Числа, которые являются по сути координатами в нашем абстрактном пространстве, могут как иметь, так и не иметь, смысла для нас. Главное, чтобы эти числа отражали близость наших сущностей.

Для музыкальных треков эмбединг обычно представляет собой вектор акустических признаков:

- MFCC коэффициенты (13-20 значений)
- Спектральные характеристики (центроид, полоса пропускания)
- Ритмические параметры (темп, regularity)
- Хроматические признаки (тональность, гармонии)

Пример: Два джазовых трека будут иметь близкие эмбединги, в то время как джазовый и хард-метал треки будут значительно различаться.

3.1.2 Метрики сходства

После того как музыка была преобразована в

Метрики сходства отражают насколько похожи два вектора в числовом дискретном значении.

Выделим следующие основные метрики сходства:

Евклидово расстояние. Измеряет расстояние между 2-мя точками в пространстве, и чем меньше расстояние между точками тем более похожими они считаются.

Формула достаточно простая:

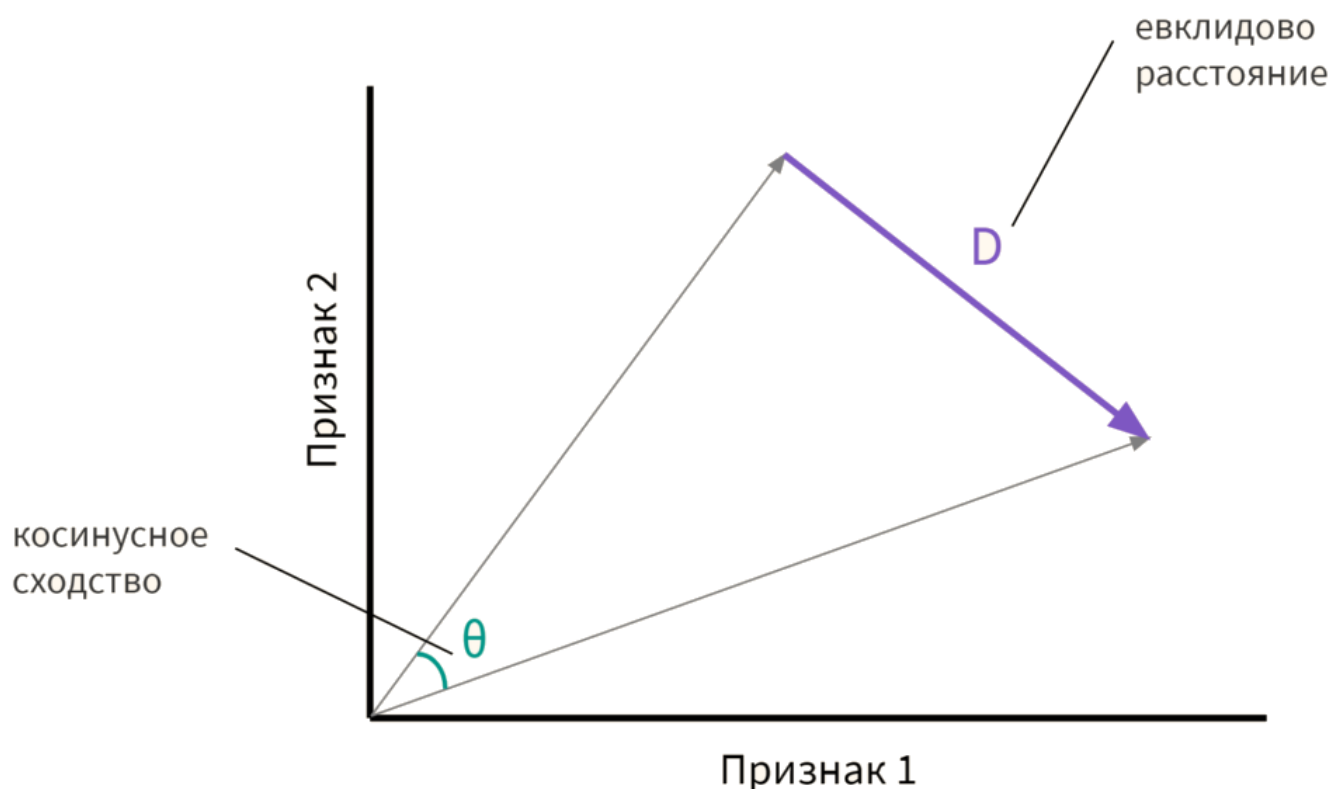
$$d(p, q) = \sqrt{\sum_{k=1}^n (p_k - q_k)^2}$$

Косинусное сходство. Это мера похожести двух векторов, равная косинусу угла между ними. Оно показывает, насколько совпадают направления векторов независимо от их длины. Вот формула:

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Косинусное расстояние. Это мера различия двух векторов, основанная на косинусном сходстве. Обычно определяется как единица минус косинусное сходство, поэтому

одинаковые по направлению векторы имеют расстояние 0, а максимально разные - близкое к 1



Источник изображения (<https://www.dmitrymakarov.ru/intro/clustering/>)

Манхэттенское расстояние. Редко используется для музыки, но всё же достойно упоминания. Эта метрика является суммой абсолютных разностей координат двух точек, она определяет кратчайший путь между ними при движении только по горизонтали и вертикали. Выглядит это так:

$$d = |x_1 - x_2| + |y_1 - y_2|$$

3.1.3 Типы рекомендательных систем

Тут мы разделим их на 3 типа, по тому на каких данных они основываются.

3.1.3.1 Контентная фильтрация (Content-Based Filtering)

При таком типе, система анализирует содержание объектов и рекомендует похожие по характеристикам.

Принцип работы:

1. Извлекаются признаки из объектов (аудио-признаки для музыки)
2. Создаётся профиль объекта (эмбединг)
3. Для целевого объекта находятся самые близкие соседи, при помощи алгоритма KNN (K-ближайших соседей)
4. Эти соседи рекомендуются пользователю

Пример: Если пользователю понравился джазовый трек с саксофоном, система порекомендует другие джазовые треки с саксофоном.

3.1.3.2 Коллаборативная фильтрация (Collaborative Filtering)

Система использует историю взаимодействий пользователей с объектами.

Выделяют 2 подхода:

- User-based: "Пользователи, похожие на вас, также слушали..."
- Item-based: "Тем, кто слушал этот трек, также понравилось..."

Проблема холодного старта: Невозможно давать рекомендации новым пользователям или новым трекам.

3.1.3.3 Гибридная система

Комбинация контентной и коллаборативной фильтрации для преодоления ограничений каждого метода. Её мы рассматривать не будем из за чрезмерной сложности подхода.

3.2 Сравнение подходов и выбор типа рекомендательной системы для проекта

3.2.1 Контентная фильтрация: преимущества и ограничения.

Преимущества:

1. Отсутствие проблемы "холодного старта" для новых объектов - система может рекомендовать новые треки сразу после их добавления, так как для этого достаточно проанализировать их акустические характеристики.
2. Не требует данных о пользовательских предпочтениях - система работает исключительно на основе анализа содержания аудиозаписей, что упрощает разработку и развертывание.
3. Прозрачность рекомендаций - можно объяснить, почему был рекомендован конкретный трек ("этот трек имеет похожие спектральные характеристики").
4. Специализация на нишевом контенте - эффективно работает с музыкой малых жанров или редких исполнителей, где недостаточно данных для коллаборативной фильтрации.
5. Устойчивость к изменениям популярности - рекомендации основаны на объективных характеристиках звука, а не на временных трендах.

Ограничения:

1. Ограниченная серендипность (serendipity) - система рекомендует только то, что похоже на уже известное, что может ограничивать музыкальные открытия.
2. Зависимость от качества извлечения признаков - эффективность системы напрямую зависит от того, насколько хорошо выбранные акустические признаки отражают музыкальные предпочтения.
3. Сложность учёта контекстуальных факторов - не учитывает время суток, настроение пользователя, социальный контекст прослушивания.

4. Требуется экспертных знаний - для выбора подходящих акустических признаков необходимо понимание как музыкальной теории, так и обработки сигналов.

3.2.2 Коллаборативная фильтрация: преимущества и ограничения

Преимущества:

1. Учёт коллективной мудрости - использует паттерны прослушивания большого количества пользователей, что может выявить неочевидные связи между треками.
2. Высокая серендипность - способна рекомендовать музыкальные открытия, основанные на поведении похожих пользователей.
3. Не требует анализа содержания - работает исключительно с метаданными о взаимодействиях, что упрощает архитектуру.
4. Эффективность для популярного контента - оптимальна для мейнстрим-музыки с большим количеством пользовательских взаимодействий.

Ограничения:

1. Проблема "холодного старта":
 - Для новых пользователей: Невозможно давать персонализированные рекомендации до накопления истории прослушиваний.
 - Для новых треков: Трек не будет рекомендован, пока не наберет достаточное количество прослушиваний.
2. Требуется масштабных данных о пользователях - необходимо собирать и хранить историю взаимодействий тысяч пользователей, что сложно для учебного проекта.
3. Проблема разреженности данных - матрица пользователь-трек обычно очень разрежена (большинство пользователей слушали лишь малую часть треков).
4. Восприимчивость к манипуляциям - рекомендации могут быть искажены искусственным накручиванием прослушиваний.
5. Сложность объяснения рекомендаций - трудно понятно объяснить пользователю, почему был рекомендован конкретный трек ("потому что другие похожие пользователи его слушали").

3.2.3 Обоснование выбора контентной фильтрации в рамках проекта

Для реализации музыкальной рекомендательной системы в рамках школьного проекта был выбран подход контентной фильтрации, что обусловлено следующими факторами:

Технические и практические соображения:

1. Отсутствие требований к пользовательским данным - как отмечалось в преимуществах контентной фильтрации, данный подход не требует сбора и обработки персональных данных пользователей, что соответствует этическим нормам и упрощает архитектуру системы.
2. Доступность необходимых данных - для работы системы достаточно только аудиозаписей, которые можно получить из открытых датасетов (FMA к примеру), в отличие от коллаборативной фильтрации, требующей масштабных данных о пользовательских взаимодействиях.
3. Соответствие масштабу проекта - контентная фильтрация может быть эффективно реализована на относительно небольшом датасете (~ 25,000 треков), тогда как коллаборативная фильтрация требует значительно больших объемов данных для достижения приемлемого качества.
4. Возможность демонстрации связи теории и практики - подход позволяет наглядно показать, как теоретические знания о физической природе звука применяются для извлечения акустических признаков и создания рекомендаций.
5. Прозрачность и объяснимость - контентная фильтрация предоставляет возможность понятно объяснить механизм работы системы.
6. Фокус на музыкальных характеристиках - система будет анализировать акустические свойства музыки, что соответствует характеру проекта.
7. Модульность системы - контентная фильтрация допускает поэтапную разработку:
 - Этап 1: Извлечение акустических признаков
 - Этап 2: Создание эмбедингов и вычисление сходства
 - Этап 3: Реализация интерфейса рекомендаций

8. Масштабируемость - система может быть легко расширена за счёт добавления новых типов признаков или улучшения алгоритмов сравнения.

Сравнительная таблица подходов в рамках проекта

Критерий	Контентная фильтрация	Коллаборативная фильтрация	Вывод о применимости в проекте
Требуемые данные	Только музыка	Данные о пользовательских взаимодействиях с музыкой	Выигрывает контентная фильтрация из за доступности данных
Сложность реализации	Средняя (Основная сложность в анализе звука)	Высокая, т.к требует хранения и обработки гигантского количества данных.	В рамках этого проекта стоит выбирать более доступный вариант
Объяснимость	Высокая. Основывается только на объекте рекомендательной системы	Низкая. (Основывается на статистике)	Контентная т.к более приближена к теме проекта.
Холодный старт	Нет проблем	Имеются проблемы т.к полностью основывается на интересах и вкусах пользователя	Контентная из за простоты и старта “с нулевой”

3.3 Выбор набора музыки (датасета)

После обоснования подхода построения рекомендательной системы (контентная фильтрация) следующим критическим шагом является выбор подходящего музыкального датасета. Качественный датасет должен соответствовать нескольким ключевым критериям: достаточный объем данных, разнообразие музыки, доступность для исследовательских целей.

3.3.1 Критерии выбора

При выборе датасета для музыкальной рекомендательной системы были сформулированы следующие требования:

1. Объем данных:

- Не менее 5 000 треков для обеспечения статистической значимости
- Достаточное количество треков в каждом жанре для обучения
- Оптимальный размер для обработки на доступном оборудовании

2. Разнообразие жанров:

- Широкий спектр музыкальных направлений (классика, рок, джаз, поп, электроника и др.)
- Четкая жанровая маркировка для валидации результатов
- Сбалансированное распределение по жанрам

3. Качество аудио:

- Стандартные форматы (MP3, WAV, FLAC)
- Приемлемое качество записи (не ниже 128 kbps для MP3)
- Единообразие параметров (частота дискретизации, битовая глубина)

4. Метаданные:

- Наличие базовой информации о треках (исполнитель, альбом, год)

3.3.2 Сравнение доступных датасетов

Были проанализированы три популярных музыкальных датасета:

[GTZAN Genre Collection](#)

Предоставляет порядка 1000 треков в формате WAV, фрагментами по 30 секунд. Данный датасет имеет готовое разбиение по жанрам и хорошо по ним сбалансирован. Имеет простую структуру и де-факто стандартный бенчмарк для различных моделей классификаций жанров музыки.

[Million Song Dataset \(MSD\)](#)

Предоставляет порядка 1 000 000 треков по 30 секунд. Также представляет метаданные и другую доп. информацию.

Можно выделить следующие преимущества датасета:

- Огромный объем данных
- Богатые метаданные
- Широкий опыт применения сообществом

И вот такие недостатки:

- Юр. ограничения на распространение
- Гигантский размер, что усложняет задачу
- Избыточность для проекта

[Free Music Archive \(FMA\)](#)

Этот вариант предоставляет несколько разных по объему датасетов:

- FMA small - содержит 8 000 треков по 30 секунд. Размер: 7.2 ГБ
- FMA medium - содержит 25 000 треков по 30 секунд. Размер: 22 ГБ
- FMA large 106 574 трека по 30 секунд. Размер 93 ГБ
- FMA full 106 574 полных трека. Размер: 879 ГБ

Все треки в формате MP3 и распространяются под Creative Common лицензией.

3.3.3 Обоснование выбора датасета

Для данного проекта выбран датасет FMA Medium по следующим причинам:

- Оптимальный объем: 25 000 треков (22 ГБ данных), этого достаточно построения рекомендательной системы, но и не превышает наших возможностей обработки.
- Хорошие метаданные.
- Лицензионная чистота. Всё распространяется под лицензиями Creative Common что позволяет использовать данные в исследовательских и образовательных целях.
- Широкий спектр жанров от классики до экспериментальной электроники.

3.3.4 Состав метаданных в FMA Medium

Каждый трек имеет поля в метаданных, и вот то что нам будет полезно:

Продолжительность, Год записи, Альбом, Автор, Доп.Текст(указание лицензии и источника), Название, Жанр, Битрейт.

Наличие этих полей было подтверждено при помощи утилиты exiftool

3.3.5 Предобработка данных

Она будет включать в себя следующие этапы:

1. Проверка целостности при помощи прикрепленных хэшей
2. Сохранение метаданных в базу данных

3.4 Методология извлечения признаков

3.4.1 Акустические признаки музыкальных треков

Для преобразования музыкальных треков в формализованные представления, пригодные для машинного анализа, необходимо извлечь набор акустических признаков, которые отражают ключевые характеристики звука. В рамках данного проекта рассматриваются следующие категории признаков:

Мел-частотные кепстральные коэффициенты (MFCC)

MFCC являются стандартом в анализе аудиосигналов и широко применяются в задачах распознавания речи и музыки. Они моделируют человеческое восприятие звука, фокусируясь на частотных диапазонах, наиболее значимых для слуховой системы.

Процесс вычисления MFCC:

1. Преобразование в мел-шкалу - нелинейное преобразование частотной оси, имитирующее восприятие человеческого уха
2. Динамическое сжатие - логарифмирование амплитуд спектра для учёта психоакустических особенностей
3. Дискретное косинусное преобразование - выделение независимых признаков

Для музыкального анализа рекомендуется использовать 13-20 MFCC коэффициентов, что позволяет сохранить баланс между информативностью и размерностью признакового пространства.

Спектральные характеристики

Спектральные признаки описывают распределение энергии по частотам и включают:

1. Спектральный центроид - средневзвешенная частота спектра, характеризующая "яркость" звучания.
2. Спектральная полоса пропускания - мера распространения спектра вокруг центроида
3. Спектральный спад - частота, ниже которой сосредоточена определенная доля спектральной энергии (обычно 85%)
4. Спектральный контраст - разница между пиками и долинами спектра, отражающая различие между гармониками и шумом

Эти признаки позволяют количественно описать тембральные характеристики музыкального звучания.

Ритмические параметры

Ритмические признаки описывают временную организацию музыки:

1. Темп (BPM) - количество ударов в минуту, определяющее скорость музыкального произведения
2. Регулярность ритма - мера предсказуемости временных интервалов между ударными событиями
3. Пульс (beat pulse) - основная метрическая структура произведения

Для извлечения ритмических признаков используются алгоритмы обнаружения ударных событий и анализа временных паттернов.

Хроматические признаки

Хроматические признаки описывают гармоническое содержание музыки:

1. Хромаграмма - 12-мерный вектор, представляющий энергию в каждом из 12 полутонов октавы
2. Тональный центр (key) - основная тональность произведения
3. Гармонические изменения - паттерны смены аккордов и гармоний

Эти признаки особенно важны для различения музыкальных жанров и стилей.

3.4.2 Процесс создания эмбеддингов

Создание эмбеддингов для музыкальных треков включает несколько последовательных этапов:

Этап 1: Загрузка и предобработка аудио

1. Чтение и загрузка MP3 файлов
2. Ресемплинг - приведение всех файлов к единой частоте дискретизации
3. Нормализация громкости - выравнивание амплитуд для устранения различий в громкости записи

Этап 2: Извлечение временных признаков

1. Разделение на части - разбиение аудиосигнала на перекрывающиеся окна длительностью 20-50 мс.
2. Вычисление признаков для каждой части- извлечение MFCC(мел частотных признаков), спектральных и других характеристик.
3. Агрегация по времени - вычисление статистических характеристик (среднее, стандартное отклонение, минимум, максимум) для всего трека.

Этап 3: Формирование единого вектора признаков

1. Конкатенация признаков - объединение признаков разных типов в единый вектор.
2. Векторизация - преобразование результата в числовой вектор фиксированной длины.
3. Документирование признаков - сохранение информации о структуре и семантике каждого компонента вектора.

4. Технологический стек и архитектура системы

В этом разделе будут описаны использованные при написании практической части инструменты и фреймворки, а также базовая архитектура системы.

4.1. Используемые технологии

Для реализации системы выбран стек основанный на экосистеме Python, включающий фреймворк FastAPI в качестве backend-фреймворка, PostgreSQL вместе с SQLAlchemy для хранения данных, библиотеку Librosa для обработки аудио и извлечения MFCC-признаков, а также библиотека Annoy для эффективной индексации и хранения эмбедингов.

4.2 Архитектура системы

В качестве архитектуры мы ориентировались на гексагональную архитектуру, в таком варианте мы имеем по сути 3 логические части:

Ядро (core) - модуль который реализует основную логику(извлечения акустических признаков, MFCC и алгоритм поиска ближайших соседей)

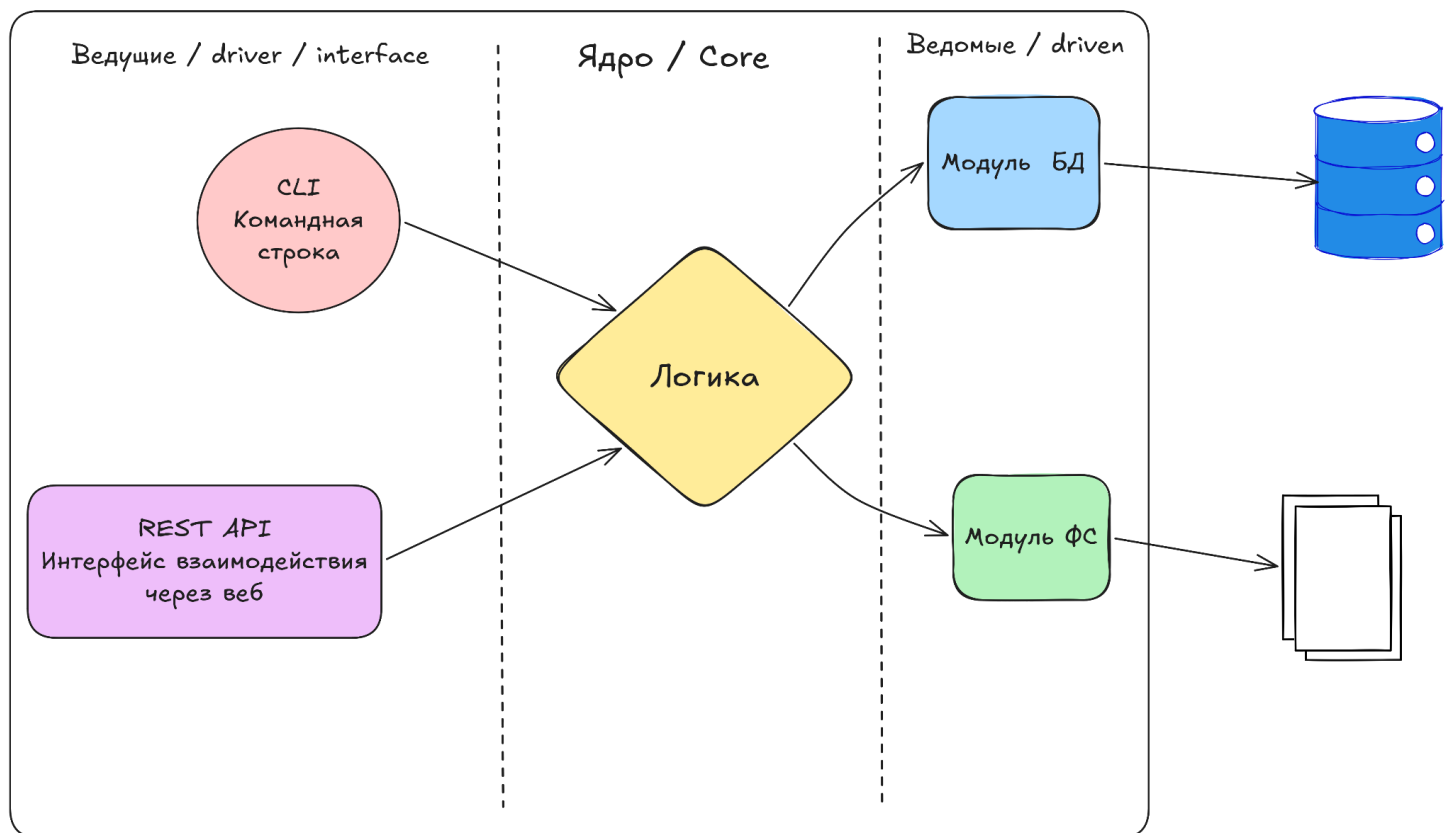
Ведомые (driven) - модуль реализующий слой работы с внешними частями от которых зависит наша система, например: базой данных, файловой системой и т.д

Ведущие (driver / interfaces) - модуль который реализует интерфейс взаимодействия внешнего пользователя с нашей системой, этот может быть как интерфейс командной строки, так и полноценный веб-сервис.

Такая архитектура позволит системе оставаться гибкой и расширяемой сохраняя относительную простоту.

Вот примерная схема такой архитектуры:

Приложение



5. Практическая реализация рекомендательной системы

В данном разделе рассматривается практическая реализация разработанной системы музыкальных рекомендаций. Основной задачей являлось создание программного решения, способного анализировать аудио сигнал, преобразовывать его в векторное представление и находить музыкальные треки со схожими акустическими характеристиками.

Практическая часть включает разработку системы для обработки и хранения аудиоданных, построение конвейера обработки извлечения признаков и формирования эмбедингов, а также создание пользовательского веб-интерфейса для взаимодействия с системой.

В рамках реализации система выполняет следующие функции:

- загрузка и хранение музыкальных треков
- извлечение акустических признаков из аудиосигнала
- формирование векторного представления треков
- поиск похожих композиций на основе сходства эмбедингов
- предоставление REST API для доступа к данным
- отображение результатов через веб-интерфейс

Таким образом, практическая часть демонстрирует полный цикл работы рекомендательной системы: от обработки аудиофайла до получения пользователем списка похожих музыкальных композиций. В следующих подразделах будет подробно рассмотрена архитектура системы, алгоритмы обработки аудиосигнала, механизм построения рекомендаций и пользовательский интерфейс.

Таким образом, каждый музыкальный трек преобразуется в набор чисел, отражающих его акустические свойства. Сравнивая такие векторы между собой, система может находить композиции со схожим звучанием.

5.1 Общая структура системы

Разработанная система музыкальных рекомендаций состоит из двух основных частей: backend и frontend. Такое разделение является стандартным для современных веб-приложений и позволяет разделить обработку данных и пользовательский интерфейс.

Backend - это серверная часть системы, которая отвечает за обработку аудиофайлов, хранение данных, построение эмбедингов и поиск похожих треков. Backend принимает запросы от пользователя, выполняет необходимые вычисления и возвращает результат. В рамках данного проекта backend реализует алгоритмы анализа аудиосигнала, формирование векторных представлений и работу рекомендательной системы.

Frontend - это пользовательский интерфейс системы, который отображается в веб-браузере. Он позволяет пользователю просматривать список треков, открывать страницу конкретной композиции, прослушивать аудио, выполнять поиск и загружать собственные файлы для поиска похожей музыки. Frontend отправляет запросы к backend и отображает полученные результаты в удобном виде.

Взаимодействие между frontend и backend осуществляется через REST API. Пользователь выполняет действие в интерфейсе, например открывает страницу трека или загружает аудиофайл. Frontend отправляет HTTP-запрос к backend, backend обрабатывает его, выполняет необходимые вычисления и возвращает данные в формате JSON. После этого frontend отображает полученный результат пользователю.

5.2 Обработка аудио и формирование эмбединга

Ключевой частью разработанной рекомендательной системы является преобразование аудиосигнала в числовое векторное представление (эмбединг). Такое представление позволяет сравнивать музыкальные треки между собой с помощью математических методов и находить похожие композиции.

На вход системы поступает аудиофайл в формате MP3. Далее система выполняет предварительную обработку, чтобы привести аудио к нужному виду. Вот как проходит предварительная обработка:

1. Загрузка аудиофайла
2. Преобразования сигнала в формат с моно каналом.
3. Приведение к единой частоте дискретизации
4. Нормализация громкости
5. Удаление тишины по краям

После предварительной обработки начинается этап извлечения акустических признаков, которые описывают различные характеристики звука и позволяют качественно представить музыку.

Система использует следующие группы признаков:

Мел-частотные кепстральные коэффициенты - описывают тембральные характеристики звука и моделируют особенности восприятия человеческим слухом.

Спектральные признаки - характеризуют распределение энергии по частотам. К ним относятся спектральный центроид, спектральная ширина, спектральный спад и спектральный контраст.

Хроматические признаки - отражают гармоническую структуру музыки и распределение энергии по нотам музыкальной шкалы.

Ритмические признаки - описывают временную структуру сигнала, включая интенсивность и изменение громкости во времени.

Изначально также планировалось использовать признак тональности (tonnetz), который позволяет описывать гармонические отношения между нотами и тональный центр композиции. Однако при развертывании системы на сервере возникла проблема: библиотека librosa вызывала ошибку при вычислении данного признака. В связи с этим извлечение тональности было отключено, чтобы обеспечить стабильную работу системы. Несмотря на исключение этого признака, оставшийся набор акустических характеристик оказался достаточным для построения рекомендательной системы и получения корректных результатов.

Каждый из перечисленных признаков вычисляется не для всего трека сразу, а для небольших временных окон аудиосигнала. После этого выполняется агрегация признаков по всем “окнам” при помощи статистических характеристик - среднего значения, стандартного отклонения, дельта-метод(mean delta).

После вычисления всех признаков они объединяются в один числовой вектор фиксированной длины. Этот вектор и является эмбедингом музыкального трека.

Также хочется отметить, что система подразумевает - вектор будет фиксированной длины для любого из треков, но сам по себе не зафиксирован на каком то конкретном размере, и зависит от реализации модуля обработки аудио. В текущей реализации вектор представляет размерность в 189 измерений.

Полученный эмбединг нормализуется, что позволяет корректно сравнивать различные треки между собой. Далее вектор сохраняется в базе данных и используется для поиска похожих композиций.

Таким образом, каждый музыкальный трек преобразуется в набор чисел, отражающих его акустические свойства. Сравнивая такие векторы между собой, система может находить композиции со схожим звучанием.

5.3 Алгоритм поиска похожих музыкальных треков

После преобразования музыкальных треков в эмбединги система может выполнять поиск похожих композиций. Основная идея заключается в сравнении эмбедингов между собой и поиске ближайших векторов в многомерном пространстве признаков.

Каждый трек в базе данных представлен числовым вектором фиксированной длины. Если два трека имеют схожее звучание, их векторы будут расположены близко друг к другу. Соответственно, задача рекомендательной системы сводится к поиску ближайших соседей для заданного вектора.

В рамках проекта используется алгоритм поиска ближайших соседей (Nearest Neighbors). Он позволяет находить несколько треков, наиболее близких к заданному по значению признаков. Однако при большом количестве данных точный перебор всех векторов

становится медленным. Поэтому используется приближённый поиск ближайших соседей (Approximate Nearest Neighbors, ANN).

Для реализации ANN-поиска применяется библиотека Annoy. Она строит специальный индекс по всем эмбедингам и позволяет быстро находить ближайшие векторы без полного перебора. Индекс создаётся один раз при запуске системы на основе всех сохранённых эмбедингов.

Процесс поиска похожих треков выполняется следующим образом:

1. Выбирается исходный трек или загруженный пользователем аудиофайл.
2. Для него формируется эмбединг (если он еще не существует).
3. Полученный вектор передается в индекс Annoy.
4. Алгоритм находит несколько ближайших векторов.
5. Найденные векторы сопоставляются с треками из базы данных.
6. Система возвращает список наиболее похожих композиций.

Чтобы оценить степени похожести используется косинусное сходство. Эта метрика измеряет угол между двумя векторами и показывает, насколько они направлены в одну сторону. Чем меньше угол между векторами, тем более похожими считаются треки.

В результате мы получаем список музыкальных композиций, отсортированных по степени сходства. Каждый элемент списка содержит информацию о треке и числовую оценку похожести.

Таким образом, рекомендательная система выполняет поиск музыки не по жанру или названию, а по реальному сходству аудиосигнала. Это позволяет находить композиции со схожим звучанием даже при отсутствии общих метаданных.

5.4 Хранение данных и импорт датасета

Для работы рекомендательной системы недостаточно только алгоритма обработки аудио. Необходимо также организовать хранение музыкальных файлов, метаданных и векторных представлений треков. Кроме того, для заполнения базы данных в проекте был реализован механизм массового импорта музыкального датасета.

В системе данные разделены на несколько частей. Аудиофайлы хранятся отдельно от метаданных и эмбеддингов. Такой подход удобен, потому что база данных не перегружается большими файлами, а вместо этого содержит только информацию о треке.

Для хранения аудио была создана специальная подсистема-хранения она полностью отвечает за хранение и получение треков. В текущей реализации эта подсистема хранит данные локально в файловой системе, создавая иерархию папок для эффективного хранения. Но также архитектура позволяет быстро и удобно реализовать подсистему для хранения аудио в облачных хранилищах (по типу s3) и заменить текущую подсистему, но в текущей реализации это избыточно.

Метаданные хранятся в базе данных, и включают в себя название трека, исполнителя, альбом, жанр, год выпуска и другую служебную информацию. Эти данные используются для отображения на сайте и для текстового поиска по каталогу. Векторные представления треков сохраняются отдельно, так как именно они применяются при поиске похожих композиций.

Для заполнения системы данными используется пакетный импорт музыкальных файлов. Во время импорта программа проходит по каталогу с аудиофайлами, определяет, какие треки ещё не были загружены, и обрабатывает только новые файлы. Чтобы избежать повторной загрузки одинаковых файлов, используется криптографическое хэширование (SHA 256). Если файл уже присутствует в системе, он пропускается.

Импорт выполняется автоматически и может использовать несколько процессов одновременно. Это ускоряет обработку большого количества аудиофайлов. После чтения файла система извлекает из него метаданные, строит эмбеддинг, сохраняет результат в базу данных и отдаёт на запись в подсистему-хранилище.

5.5 Пользовательский интерфейс и API системы

Чтобы пользователь мог взаимодействовать с системой был создан веб-интерфейс, а также программный интерфейс взаимодействия (API) через который фронтенд отправляет и получает данные с сервера. Такое разделение позволяет отделить логику обработки данных от отображения информации и делает систему более гибкой и расширяемой.

Веб-интерфейс(Frontend)

Интерфейс разработан с использованием React, TypeScript и библиотеки компонентов Mantine. Выбор данного технологического стека обусловлен предыдущим опытом разработки с React и TypeScript, что позволило ускорить создание интерфейса и сосредоточиться на логике взаимодействия с системой, а не на изучении новых инструментов. Использование TypeScript также повысило надёжность кода за счёт статической типизации, а Mantine упростил создание аккуратного и функционального пользовательского интерфейса.

Основные возможности интерфейса:

- просмотр списка музыкальных треков;
- поиск треков по названию, исполнителю, альбому или жанру;
- открытие страницы конкретного трека с подробной информацией;
- прослушивание аудио фрагмента;
- получение списка похожих треков;
- загрузка собственного аудио файла для поиска похожей музыки.

Каждый трек отображается в виде карточки, содержащей основную информацию и элементы управления. При переходе на страницу трека пользователь может увидеть более подробные данные и список рекомендаций, построенных на основе аудиоанализа.

REST API (backend)

Backend предоставляет REST API - набор HTTP-эндпоинтов, через которые frontend получает доступ к данным и функциональности системы. API возвращает данные в формате JSON и связь между пользовательским интерфейсом и серверной логикой.

Основные эндпоинты системы:

- получение списка треков с пагинацией
- получение информации о конкретном треке
- текстовый поиск по метаданным
- получение случайных треков
- получение списка похожих треков
- загрузка аудиофайла для поиска похожих композиций
- получение аудио файла для прослушивания

Роль API в системе

API является ключевым связующим звеном всей архитектуры. Оно обеспечивает:

- доступ к данным музыкальной коллекции
- выполнение операций поиска и рекомендаций
- загрузку и обработку пользовательских аудиофайлов
- интеграцию frontend и backend в единую систему

5.6 Результаты и ограничения

В результате реализации проекта была создана полноценная музыкальная рекомендательная система, которая выполняет анализ аудиосигнала, формирует его векторное представление и на основе этого находит похожие музыкальные композиции.

Практическая проверка системы показывает, что она корректно находит музыкальные треки со схожим звучанием. Например, при загрузке аудиофайла пользователь получает список композиций с близкими акустическими характеристиками, даже если жанры или названия треков отличаются. Это подтверждает, что система действительно опирается на анализ звука, а не на простое сравнение метаданных.

Также система поддерживает работу с достаточно большим количеством треков благодаря использованию индекса Annoy, который позволяет выполнять поиск похожих векторов значительно быстрее, чем полный перебор всех элементов.

Ограничения

Несмотря на успешную реализацию, система имеет ряд ограничений, характерных для учебного проекта и выбранного подхода:

1. Ограниченность набора признаков

Векторное представление основано только на наборе акустических признаков (MFCC, спектральные и хроматические характеристики). Это не позволяет учитывать более сложные музыкальные факторы, такие как структура композиции или эмоциональная окраска.

2. Потеря некоторых признаков из-за технических ограничений

Изначально планировалось использовать признак тональности, однако его вычисление оказалось нестабильным, поэтому он был исключен из финального набора признаков.

3. Отсутствие пользовательской персонализации

Система не учитывает предпочтения конкретного пользователя, так как не реализована коллаборативная фильтрация и нет истории взаимодействий. Все рекомендации основаны только на содержимом аудиофайлов.

4. Ограниченная масштабируемость индекса

Индекс Annoy пересобирается при запуске системы целиком. Это упрощает реализацию, но делает процесс обновления данных менее эффективным при частом добавлении новых треков.

5. Ограничения учебного формата проекта

Проект реализован в учебных целях, поэтому некоторые инженерные решения были упрощены в пользу наглядности и объяснимости.

Итог

Несмотря на указанные ограничения, система демонстрирует работоспособный подход к построению музыкальной рекомендательной системы на основе анализа аудиосигнала. Ознакомиться с реализацией системы можно по ссылке:

<https://ttt.delroms.ru>

Или по QR-коду:

Исходный код доступен на GitHub:

<https://github.com/AltairGeo/music-recomend-sys>

<https://github.com/AltairGeo/music-recomend-sys-frontend>



Источники

“Звук. Знание.Вики”

<https://znanierussia.ru/articles/Звук>

“Психоакустика. Википедия”

<https://ru.wikipedia.org/wiki/Психоакустика>

“Квантование. Википедия”

[https://ru.wikipedia.org/wiki/Квантование_\(обработка_сигналов\)](https://ru.wikipedia.org/wiki/Квантование_(обработка_сигналов))

“Римский-Корсаков А.В. Электро-акустика”

“Импульсно-кодовая модуляция. Википедия”

https://ru.wikipedia.org/wiki/Импульсно-кодовая_модуляция

“Кодирование звуковой информации. Академик”

<https://dic.academic.ru/dic.nsf/ruwiki/1524417>

“Что такое эмбединги и как с ними работать. Вводная для начинающих. Хабр”

<https://habr.com/ru/articles/947216/>

“Эмбединги для начинающих. Хабр”

<https://habr.com/ru/companies/otus/articles/787116/>

Мел-кепстральные коэффициенты (MFCC) и распознавание речи

<https://habr.com/ru/articles/140828/>

“Кластерный анализ. Дмитрий Макаров”

<https://www.dmitrymakarov.ru/intro/clustering/>

“Ищем по-соседски: методы приближенного поиска ближайших соседей для А/В-тестирования гипотез. Хабр”

<https://habr.com/ru/companies/sberbank/articles/726532/>

“Рекомендательные системы: идеи, подходы, задачи”

<https://habr.com/ru/companies/jetinfosystems/articles/453792/>

FMA Датасет

<https://github.com/mdeff/fma>

GTZAN датасет

<https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification>

Million Songs Dataset (MSD)

<http://millionsongdataset.com/>

“Анализ аудиоданных. Хабр”

<https://habr.com/ru/articles/668518/>

Снижение размерности

https://ru.wikipedia.org/wiki/Снижение_размерности

Спектральный анализ сигналов

<https://habr.com/ru/articles/253447/>

“Что такое гексагональная архитектура. Разделение бизнес-логики и инфраструктуры с помощью портов и адаптеров”

<https://habr.com/ru/companies/timeweb/articles/771338/>

“Классификация аудиофайлов с библиотекой Librosa. Habr”

<https://habr.com/ru/companies/otus/articles/741080/>