МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
(НИЯУ МИФИ)
Предуниверситарий НИЯУ МИФИ

Выпускная работа обучающегося IT-класса Приедуниверситария НИЯУ МИФИ

«Система музыкальных рекомендаций»

Обучающиеся — Полосин Павел Ильич, 11И класс
Маляренко Андрей Игоревич, 11И класс

Научный руководитель
Должность, степень, звание — Лукаш Владислав Валерьевич, data scientist

Дата защита:
Результат защиты:

Москва 2022

Реферат

Аннотация: Поиск новой музыки может заниматься много времени. Для того, чтобы помочь пользователю упростить задачу поиска похожих музыкальных композиций, решено построить систему музыкальных рекомендаций.

В работе рассматривается подход для решения данной задачи при помощи метода машинного обучения – ассоциативных правил и его реализация в виде конечного продукта для взаимодействия с пользователем.

Отчет содержит состоит из 20 страниц, снабжен тремя таблицами и тремя пояснительными риснуками.

Ключевые слова: Ассоциативные правила, Association rule learning, Apriori, Support, Confidence, Lift, система музыкальных рекомендаций, расстояние Дамерау-Левенштейна

Содержание

Оглавление

Введение

В современном мире рекомендательные системы используются повсеместно [1]. Они моделируют поведение пользователя путем обработки данных о поведении множества людей, позволяют пользователям принимать решения проще и быстрее. Для работы, рекомендательные системы используют отзывы о продуктах в качестве входных данных. Иными словами, отзывы – это обратная связь пользователя о каком-то конкретном продукте. Различают два типа обратной связи: явная и неявная. Явная предполагает оценку предлагаеомего контена пользователем, невявная, напротив, основывается на косвенных факторах, например, сколько раз прослушал песню пользователь, подписался ли на исполнителя и др.

Рекомендательные системы широко используются во многих сферах нашей жизни. Одним из примеров популярного использования - рекмоендательные системы для поиска новой музыки, ведь данный процесс может заниматься много времени – человеку потребуется прослушать большое количество песен, пока не найдется та, которая понравится. Выбор пользователя основывается на множестве факторов: настроение, местоположение и т. д. Более того, подобные системы позволяют расширять горизонт знаний о музыкальных исполнителях пользователя. Имеено поэтому существует ниша рекомендательных систем, целью которых является подбор музыкальных произведений на основе предпочтений пользователя.

Современные системы машинного обучения для персонализации (подбора персональных рекомендаций) могут быть построены на рзличных методах анализа информации. Выделяют несколько подходов построения рекомендательной системы [2]:

- Content based filtering: мы сопоставляем пользователей с тем контентом или товарами, которые им нравились или были ими куплены.

- Collaborative filtering: Те, кто одинаково оценивал какие-либо предметы в прошлом, склонны давать похожие оценки другим предметам и в будущем.
- Hybrid collaborative filtering: Техника обединяющая предыдущие две для получения лучшего результата.

Исследования показывают, что наиболее эффективными методами являются системы основанные на Hybrid collaborative filtering. Поэтому целью данной работы является изучение существующих гибридных музыкальных рекомендательных систем и разработка новой модели, основаной на гибридной технике указаной выше.

В этом исследовании мы не предсказываем явно поведение пользователя в данный момент, а скорее пытаемся предсказать, песни какого исполнителя могут понравиться следующими, основываясь на выборе в прошлом. Для анализа была выбарана модель на основе метода машинного обучения - association rule mining.

Связанные работы

В соврвеменно мире существуют различные рекомендательные системы, которые доступны обычному пользователю, такие как например «Яндекс. Музыка», spotify, pandora, youtube music, apple music. Однако, представленные системы не описыват то, как происходит процесс рекомендации и оставляют свои алгоритмы и модели в рамках корпоративной тайны. Поэтому в данной работе мы сфокусируемся на опубликованных исследования с описанием обучения и тестирования моделей машинного обучения. Далее крато разобраны основные методы и принципы решения из наиболее известных публикаций. Заинтересованный читатель может подробнее ознакомиться с решениями и деталями публикаций в приведенных ниже ссылках.

Авторы [3] разработали систему музыкальных рекомендаций, основанную на анализе личных предпочтений. Сначала они построили модель, используя Скрытые Модели Маркова (Hidden Markov Models, HMM) с Мел-кепстральными коэффициентами (Mel Frequency Cepstral Coefficients, MFCC), а также рассчитали HMM для каждой композиции, представляя одну модель для каждой песни. Затем ищутся похожие среди них, вычисляются и сохраняются векторы, имеющие сходства. Авторы исследования используют улучшенный алгоритм K-Means, чтобы ограничить радиус кластера. Для анализа предпочтений и присвоения весов кластерам они ставят более высокие веса тем, которые находятся дальше друг от друга, и меньший вес для кластеров, которые находятся рядом. Их система рекомендаций использует эти веса для моделирования сходства, чтобы рекомендовать музыку пользователям. Их алгоритм даёт очень точные рекомендации для некоторых жанров.

Авторы [4] использовали необычный способ сравнения музыки. Они построили тембровый дескриптор для музыкальных композиций и

вычислили его для измерения сходства песен. Тембр можно назвать “цветом тона”. Музыкальные инструменты создают разные частоты для одной и той же ноты. Тембр используется для того, чтобы выявить эти инструментальные различия. Они разделили песню на 2048 частей и рассчитали MFCC. Что касается огромного набора данных, то было бы невозможно сохранить все значения MFCC, поэтому они использовали смешанную модель Гаусса (GMM) для хранения среднего значения и ковариации. Они рассмотрели два способа измерения расстояния между моделями – правдоподобие и выборка. Правдоподобие, по-видимому, сложнее вычислить, поэтому они использовали выборку. Для этого было необходимо выбрать выборку из одного GMM и вычислить её правдоподобие с учётом другого GMM. Их исследование показало, что музыкальное сходство можно использовать для генерации плейлистов.

В исследовании [5] были представлены два алгоритма для рекомендации музыки. Они учитывают музыкальный контент, информацию о певце / жанре и популярности во время рекомендации. Один из их алгоритмов основан на энтропии: они определили соответствующий набор контента с учётом выбранной пользователем песни. Затем они сгруппировали эти функции как можно компактнее. Они использовали энтропию в качестве меры компактности. Для дальнейшего совершенствования они внедрили другой алгоритм обучения, который использовал информацию о музыкальном контенте, певце, жанре и популярности.

В исследовании [6] авторы предложили систему поиска музыки и рекомендаций, основанную на настроении музыкальной последовательности. Сначала они разделили песни на сегменты. Для этого разделения они использовали несколько тембровых особенностей. После этого сгруппировали сегменты с похожими характеристиками, используя алгоритм кластеризации K-medoids, и пометили каждый кластер уникальными

символами настроения. Чтобы оценить сходство музыки, они использовали алгоритм Смита-Уотермана. Создатели также использовали предложенный ими метод рекомендаций для классификации жанров, который достиг 70% точности.

Однако, стоить заметить, что все вышеперчисленные системы основаны на математических моделях которые получили много внимания в прошлые года и поэтому могут считаться устаревшими. Одной из ключевых проблем таких систем является то, что они предоставляют в качестве рекомендации только одно музыкальное произведение. В современном мире, когда каждый пользователь обладает большой музыкальной библиотекой, подобные предложения могут считаться очень слабыми, потмоу что пользователь хочет слушать больше чем один новый трек. Поэтому, в данной работе мы рассматриваем использование ассоциативных правил, как метод рекомендаций, который позволяет более мягкие рекомендации и тем самым расширяет диапазон ответов, которые система может порекомендовать.

Основная часть проекта

Ассоциативные правила

Одним из способов решения поставленной задачи является метод машинного обучения, подраздел искусвтенного интелекта и науки о данных, специализирующийся на использовании данных и алгоритмов для имитации процесса наработки опыта человеком с постепенным повышением точности, - обучение ассоциативным правилам или поиск ассоциативных правил — это метод обучения машин на базе правил обнаружения интересующих нас связей между переменными в большой базе данных. Ассоцитивное правило — это импликационное выражение формы X→Y, где X и Y — непересекающиеся наборы элементов(Tan, Steinbach, Kumar. Introduction to Data Mining. Pearson New International Edition. Harlow: Pearson Education Ltd., 2014.). Иллюстрацией ассоциативного правила можно считать следующее:

$$\{\text{Jay} - \text{z}\} - > \text{Nas}$$

Предпологающее, что пользователям, предпочитающим исполнителя Jay-Z нравится исполнитель Nas.

Поиск ассоциативных правил

Очевидно, что при большом количестве исполнителей выявлять такие ассоциативные правила интуитивно или перебором – трудно. Оценим сложность такой задачи. Пусть рассматриваются n исполнителей. Тогда число всевозможных подмножеств - 2^n .

Для сокращения количества просматриваемых множеств, те из них, которые встречаются очень редко можно исключить.

Для решения поставленной задачи будем использовать алгоритм Apriori [7]. Априори получал популярность в мире и используется во многих сферах. В основу этого алгоритма заложено использование априорных данных о частоте выбора пользователями тех или иных наборов продуктов. Этот

постулат звучит следующим образом: «Все подмножества множества продуктов, пользующихся большим спросом, также имеют большую частоту встречаемости».

Для работы алгоритм использует несколько метрик, введем их:

Основные метрики

Support – это относительная частота появления набора X среди всех транзакий, т.е. выборочная оценка для априорной вероятности появления набора X.

$$\text{Support} = \frac{\text{count(X)}}{N}$$

Confidence - мера мощности прогноза, которое даёт правило. Иными словами, если пользователь послушал X, то вероятность, что он послушает Y есть confidence.

$$\text{confidence}(X \to Y) = \frac{\text{support}(X \to Y)}{\text{support}(X)}$$

Вообще говоря, $\text{confidence}(X \to Y) \neq \text{confidence}(Y \to X)$.

$$lift(X \to Y) = \frac{confidence(X \to Y)}{\text{support}(Y)}$$

Если правило имеет лифт 1, это означает, что событие в левой части независимо от события в правой части. Если два события независимы, никакого правила нельзя вытащить из этих двух событий.

Если lift> 1, это позволяет нам знать степень, насколько события связаны друг с другом, и делает эти правила потенциально полезными для предсказания следствия в будущих наборах данных.

Если lift <1, это означает, что объекты заменяют друг друга. Это означает, что наличие одного объекта имеет отрицательный эффект на присутствие второго объекта, и наоборот.

Приведем пример вычисления основных метрик. Пусть имеется набор данных в формате, представленном в таблице. Будем называть его транзакциями, так как каждому пользователю соответствует набор композиций, которые он выбрал (условно «купил»). Введем основные понятия.

Таблица 1. Представление данных в формате транзакций. [8]

TID	Items
1	Jay-z, Rihann
2	Jay-z, Nas, Beyonce
3	Nas
4	Jay-z, Rihanna, Beyonce
5	Jay-z, Rihanna, Nas, Beyonce

$support(Jay - Z) = \frac{4}{5} = 0.8$, исполнитель Jay-Z встречается в строках 1, 2 и 5

$support(Jay - Z \rightarrow Rhianna) = 0.6$, Пара (Jay-Z, Rhianna) в 2, 4 и 5.

$confidence(Jay - Z \rightarrow Rhianna) = \frac{0.6}{0.8} = 0.75$

Будем называть правила *сильными*, если они будут иметь большие значения показателей support и confidence (например правило $(Jay - Z \rightarrow Rhianna)$ можно считать сильным).

Априорный алгоритм

Работа априорного алгоритма состоит из нескольких этапов:

1. Определение товарных наборов X, для которых величина support(X) превышает некоторое заранее заданное пороговое значение.

2. Выявление на этих товарных наборах X ассоциативных правил X –> Y, для которых величина confidence(X –> Y) превышает некоторое заранее заданное пороговое значение

1. $F_1 = \{$ Часто встречающиеся 1-элементные наборы$\}$;
2. для $(k = 2, F_{k-1} <> \varnothing, k{+}{+})$
3. $\{$
4. $C_k = \{$ Генерация кандидатов k на основе $Apriorigen(F_{k-1})\}$
5. для всех транзакций $t \in D$
6. $\{$
7. $C_t = subset(C_k, t)$ // Удаление избыточных правил
8. для всех кандидатов $c \in C_t$
9. $c.count{+}{+}$
10. $\}$
11. $F_k = [c \in C_k | c.count >= minsupp]$ // Отбор кандидатов
12. $\}$
13. Результат $\cup F_k$

Существуют и другие алгоритмы поиска ассоциативных правил, однако различия в их результате, времени работы практически нет.[1]

[1] Эксперимент проводился на следующем оборудовании:
Processor 1.6 GHz Intel Core i5
Memory 8 GB 2133 MHz LPDDR3
Startup Disk Macintosh HD
Graphics Intel UHD Graphics

Таблица 2. Сравнение скорости работы различных алгоритмов поиска ассоциативных правил.

Размер набора данных **(invoices / items)**	**Apriori**	**Fp-growth**	**FP-max**
(9531, 95)	182 ms	112 ms	114 ms

Создание модели

Для выявления правил и корректной работы алгоритма требуются данные. Мы собрали и обработали их: изначально было несколько наборов данных. Первый состоял из 200.000 строк и содержал такие поля, как ID пользователя, группа, исполнителя, которого он прослушал, страна, в который в данный момент находился пользователь и другие. Второй набор представлял несколько ином формате и, кроме того, содержал справочные сведения об исполнителе (жанр, популярные песни и др.). Основная задача обработки и исследования данных – это удаление дубликатов и пустых строк, приведения набора к формату транзакций, где каждому пользователю соответствует набор композиций, который он прослушал.

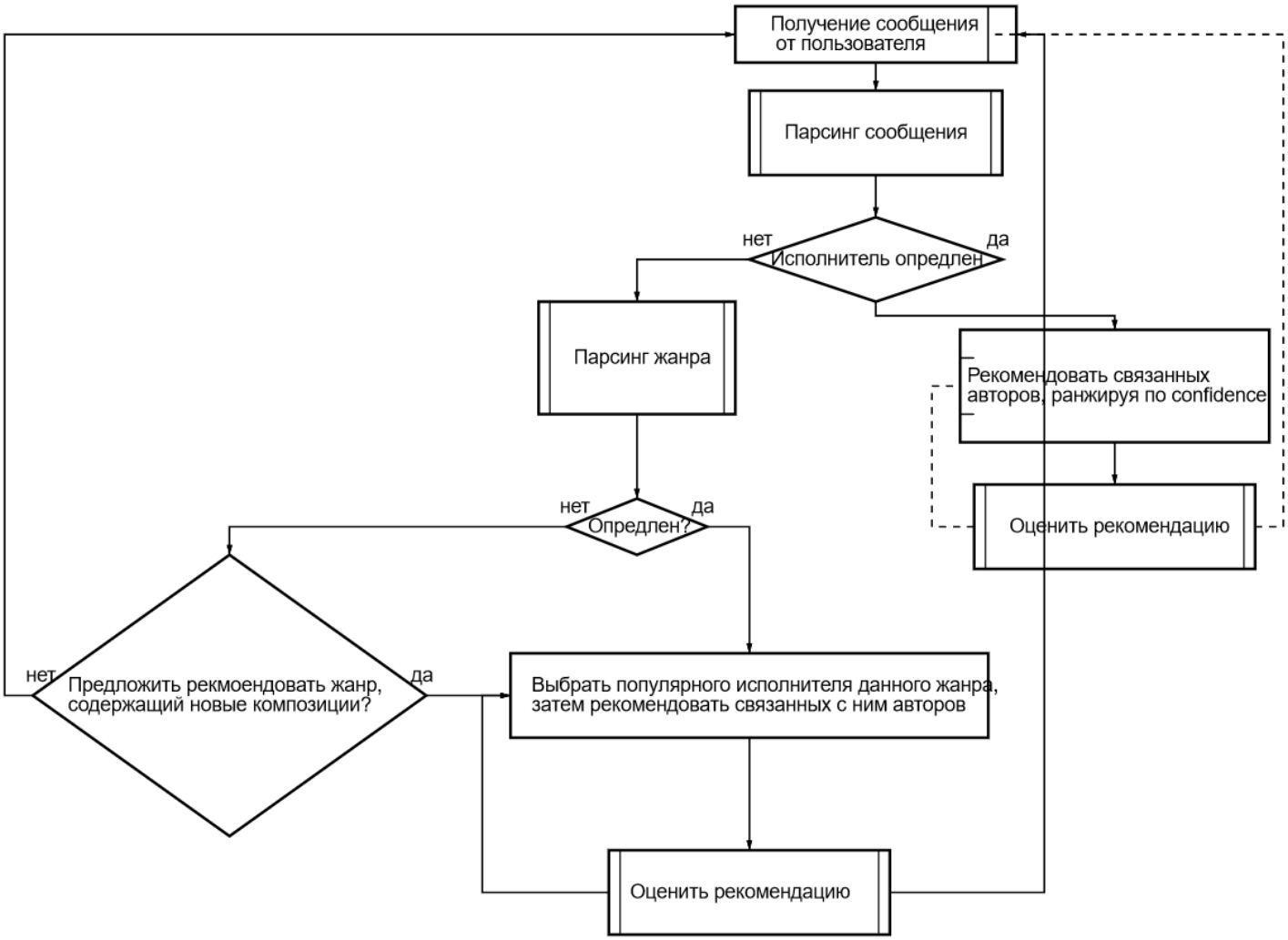
Используя алгоритм Apriori, эмпирическим путем были выявлены следующие оптимальные параметры коэффициентов confidence> 0.2, support> 0.01, при котором правила остаются достаточно сильными**.** На текущей момент выявлено около 2.000.000 связей, часть из которых представлена в таблице

Таблица 3. Пример ассоциативных правил, полученных в результате работы алгоритма.

X	Y	Support	Confidence	Recommendation
«jay-z»	«nas»	0.010600	0.349451	«jay-z»=>»nas»
«a perfect circle»	«tool»	0.0162	0.442831	«a perfect circle» =>«tool»

Взаимодействие с пользователем

Взаимодействие с пользователем реализовано через telegram-бота,



архитектура представлена на блок-схеме ниже:

Рисунок 1. Архитекутра ПО, для общения с пользователем.

Общение начинается с отправки сообщения с указанием исполнителя. Далее при помощи нечеткого поиска опредлеятся композитор, которого имел ввиду пользователь(он мог опечаться или ввести с ошибкой). Причем для реализации

нечеткого поиска используется алгоритма поиска **расстояния Дамерау-Левенштейна:** поиска минимального числа операций вставки, удаления, замены одного символа и транспозиции двух соседних символов, необходимых для перевода одной строки в другую, основанный на идеи динамического программирования по префиксу.

Дамерау показал, что 80% человеческих ошибок при наборе текстов составляют перестановки соседних символов, пропуск символа, добавление нового символа, и ошибка в символе. [9] Поэтому метрика Дамерау-Левенштейна часто используется в редакторских программах для проверки правописани.

Пусть S и T — строки, между которыми требуется найти расстояние Дамерау-Левенштейна; M и N — их длины соответственно. Будем хранить матрицу $D[0..M+1][0..N+1]$, где $D[i+1][j+1]$ — расстояние Дамерау-Левенштейна между префиксами строк S и T, длины префиксов — i и j соответственно. Для учёта транспозиции потребуется хранение следующей информации. Инвариант: lastPosition[x] — индекс последнего вхождения x в S.

last — на i-й итерации внешнего цикла индекс последнего символа T[last]=S[i]. Тогда если на очередной итерации внутреннего цикла положить: i'=lastPosition[T[j]], j'=last, то

$D(i,j)=\min(A,D(i',j')+(i-i'-1)\cdot deleteCost+transposeCost+(j-j'-1)\cdot insertCost)$ $(*)$,

где

$$A = \begin{cases} 0 & ;\ i = 0,\ j = 0 \\ i * deleteCost & ;\ j = 0,\ i > 0 \\ j * insertCost & ;\ i = 0,\ j > 0 \\ D(i-1, j-1) & ;\ S[i] = T[j] \\ \min (& \\ \quad D(i, j-1) + insertCost & \\ \quad D(i-1, j) + deleteCost & ;\ j > 0,\ i > 0,\ S[i] \neq T[j] \\ \quad D(i-1, j-1) + replaceCost & \\ ) & \end{cases}$$

Рисунок 2. Алгоритм поиска расстояния Дамерау-Левенштейна.

Этот алгоритм используется и для парсинга жанра, причем пользователь имеет возможность вводить запросы не только на русском языке, но и на английском.

Также реализована возможность оценки рекомендаций. При помощи интерфейса, пользователь может отмечать понравившиейся рекомендации формируя таким образом собственный плейлист. Однако, дополнительно, мы имеем возможность создавать собственный набор данных для того, чтобы через равные промежутки времени пересчитывать связи и формировать новые правила. Такая система позволяет гибко реагировать на изменения вкусов пользователей и масштабировать проект, добавляя новые композиции.

Оценка модели

В качестве системы оценки точности работы обученной модели была выбрана стандратная схема, при которой оригнальные данные разделаются в соотношении 80 к 20, где 80% данных используется для обучения модели, а 20% данных для тестирования модели.

Для выолпнения теста проводились следующие действия:

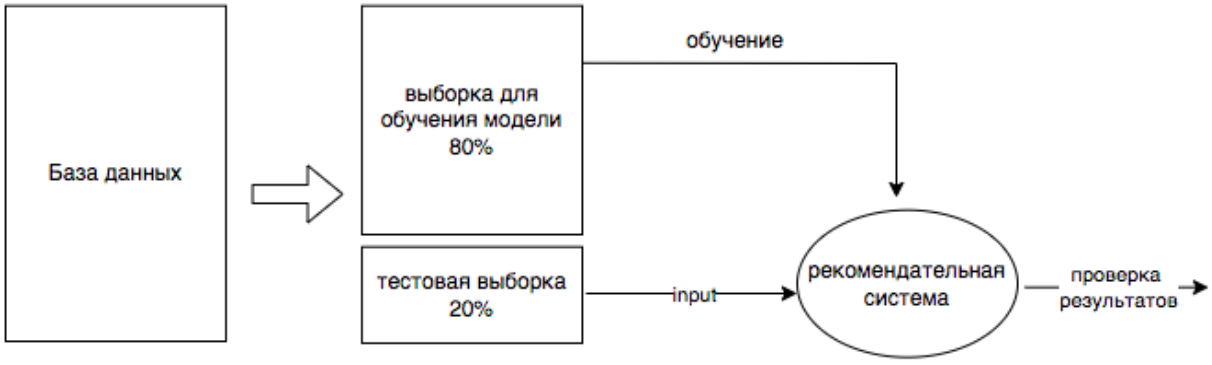
1) В каждой строчке из тестовой выборки была случайным образом взята какая-то запись. Все остальные были спрятаны.
2) Данная нескрытая запись была отправлена в обученную модель.
3) Полученный ответ модели, который содержит список различных рекомендаций, оценивался с учетом «спрятаных» ранее записей. Для вычисления точности используется следующая формула:

$$\text{точность} = \frac{\text{количество верно рекомендованых исполнителей}}{\text{общее количество "спрятанных" исполнителей}}$$

Если все спрятаные записи присутствуют среди рекомендаций – это говорит о полном совпадении рекомендаций.

Ниже ппривелена схема разъясняющая данную модель оценки и показывюшая пример.

Рисунок 3.1. Схема для оценки точности модели.



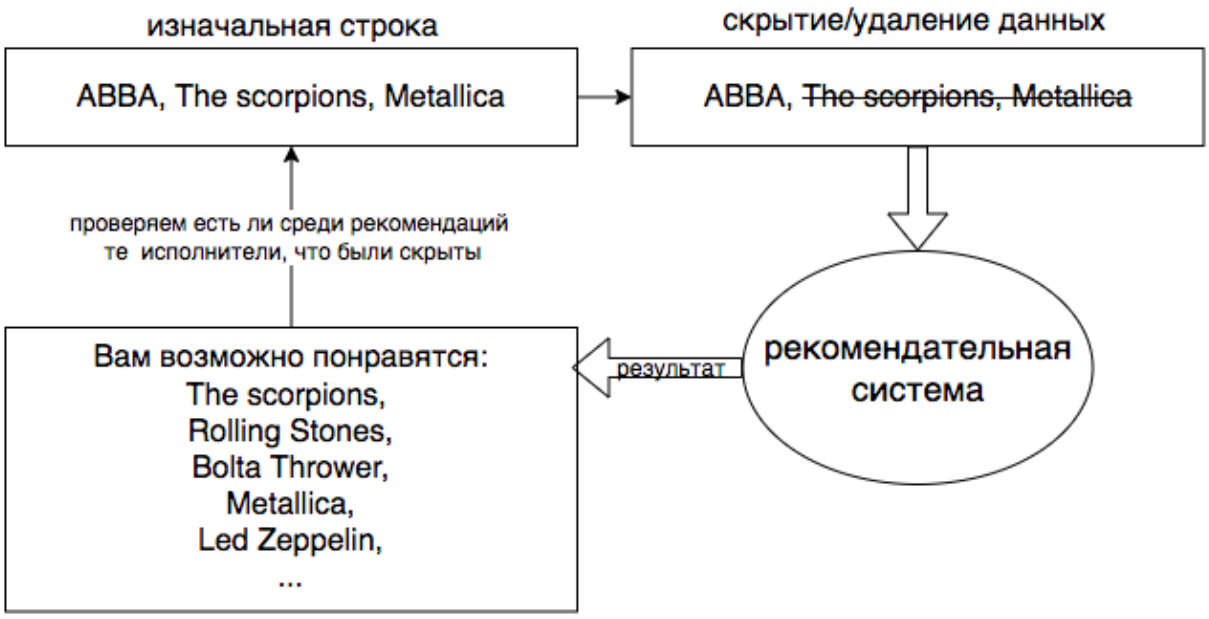


Рисунок 3.2. Схема для оценки точности модели.

Описаная система оценки показала, что точность разработоного метода составляет 79%. Данные значение можно считать достаточно высоким для подобных систем.

Заключение

Современные системы рекомендаций широко представлены на рынке и могут быть адаптированы под разные сферы. Одной из поулярных сфер применения являтся рекомендация музыкальных произведений или испольнителей. Среди популярных сервисов можно Spotify, Apple Music и Яндекс.Музыка.

В данной работе мы рассмотрели популярные публикации с описанием подобных систем персонализации, их методы и подходы к оценке результатов. На основе рассмотреных работ был сделан вывод, что существующие системы обычно предоставляют в качестве рекомендации один трек, что может расцениваться как недостаток в определнных случаях. Поэтому в данной

работе мы рассматривали задачу создания системы, которая предоставляет список рекомендаций в удобном для пользователя интерфейсе.

В ходе работы была создана система, которая состоит из рекомендательной модели основаной на ассоциативных правилах. Далее данная двухступенчатая система подбора исполнителей была встроена в интерфейс пользователя, который представляет собой бот-телеграмм. В качестве входных данных данный бот получает название трека или исполнителя, и затем выдает список соотвествующих треков (или исполнителей) которые могут понравиться пользователю. Причем пользователь может вводить сообщение с ошибками, ведь в данной работе предусмотерна функция нечеткого поиска. Кроме списка предоставляются ссылка на композиции данного автора. Помимо этого, система дополнена возможностью оставлять отзыв к рекомендациям, что позвояет улучшать и обновлять базу данных, формируя более точный результат с каждым новым отзывом.

Оценка созданной модели проходила с использованием стандартной схемы, когда начальные данные разделяются в соотношении 80 к 20 и большая часть используется для обучения модели, а меньшая для тестирования точности полученных рекомендаций. В ходе оценки точности получилось, что вероятность того что система угадывает любимых исполнителей составляет 79% процентов, что можно считать высоким значением и сравнимо с точностью работ разобранных в данном документе. В результате выполнения данной работы была выполнена цель по созданию рекомендательной системы с достаточно высокой точностью рекомендаций, гибкой системой обучения и удобным интерфейсом.

Список литературы

[1] M. A. G. A. K. Sobia Zahra, Novel centroid selection approaches for KMeans clustering based recommender systems, May 2015.

[2] M. H. K. Marwa Hussien Mohamed, Recommender Systems Challenges and Solutions Survey, 2019.

[3] K. L. D. Y. T. &. L. J. Kim, A music recommendation system based on personal preference, 2008.

[4] J. &. P. F. Aucouturier, «Music similarity measures: What's the use? in Proc. Int. Conf. Music,» 2002.

[5] Z. &. A. B. Cataltepe, *Music Recommendation by Modeling User's Preferred Perspectives of Content, Singer/Genre and Popularity,* 2009.

[6] S. R. S. &. H. E. Jun, *Music retrieval and recommendation scheme based on varying mood sequences.*, International Journal on Semantic Web and Information Systems, 2012.

[7] I. T. &. S. Agrawal R., *A. Mining Association Rules Between Sets of Items in Large Databases. In ACM SIGMOD Record.*

[8] G.-C. C. T. K. &. B. S. Tew C., *Behavior-based clustering and analysis of interestingness measures for association rule mining. Data Mining and Knowledge Discovery*, 2014.

[9] F. J. Damerau, "A technique for computer detection and correction of spelling errors", 1964.

[10] F. &. A. J. J. o. N. Pachet, «Improving timbre similarity: How high is the sky,» 2004.