Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Международных образовательных програм» (ФМОП)
«Информатика и Системы Управления» (ИУ)

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии» (ИУ7)

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ НА ТЕМУ:

Рекомендательная система для витаминов, на основе ранних действиях пользователя

Студент __ИУ7И-76Б_____ _____ А.С.Миневска
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель _____ __ П. В. Клорикьян __
(Подпись, дата) (И.О.Фамилия)

2020 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой __ИУ7____
(Индекс)
_____ _И. В. Рудаков_
(И.О.Фамилия)
« 23» ноября 2020 г.

З А Д А Н И Е
на выполнение научно-исследовательской работы

по теме *Рекомендательная система для витаминов, на основе ранних действий пользователя*

Студент группы: ИУ7И-76Б_Миневска Ани Стовянова
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)
Учебная

Источник тематики (кафедра, предприятие, НИР)
кафедра

График выполнения проекта: 25% к __ нед., 50% к __ нед., 75% к __ нед., 100% к _ нед.

Техническое задание *Разработать рекомендательная система для витаминов, на основе ранних действиях пользователя*

Оформление курсового проекта:

Расчетно-пояснительная записка на _10-20_ листах формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания «23» ноября 2020 г.

Руководитель НИР _____ __П.В.Клорикьян
(Подпись, дата) (И.О.Фамилия)

Студент _____ __А.С.Миневска___
(Подпись, дата) (И.О.Фамилия)

Оглавление

Введение

Витамины — это органические вещества, благодаря которым тело любого организма функционирует правильно и все химические реакции протекают без помех. Большинство витаминов не синтезируются самим организмом. Их можно получить через пищу. При этом доза витаминов, необходимая для нормальной работы организма, совсем небольшая — около 100 мг в день.

Биологически активные добавки — это «природные (идентичные природным) биологически активные вещества, предназначенные для употребления одновременно с пищей или введения в состав пищевых продуктов». Витамины принято использовать для укрепления здоровья.

По этой причине, люди, которые заинтересованы своим здоровый покупают витамины. В современном мире, с очень популярны онлайн магазины. Они увеличивают удобство, скорость покупки и облегчают выбор.

Но из-за этого людям доступно все больше информации во всех сферах жизни. Обыкновенному человеку, который не является врачом, почто не посильно одиночку разобраться со всей информации без стороной помощи. Чтобы справится с этой проблемой, можно использовать рекомендательные системы. Через нее быстрее и легче можно получить рекомендации в интернет-магазинах, в частности при покупке витаминов.

Существует два наиболее распространённых подхода к построению рекомендательных систем — коллаборативная фильтрация и рекомендации на основе содержимого (content based). Также существуют гибридные системы, которые являются комбинацией этих двух подходов. При рекомендациях на основе содержимого обо всех пользователях собирается информация, которая может говорить об их предпочтениях. Также для каждого объекта, который можно порекомендовать пользователям, выделяются признаки, по которым можно охарактеризовать этот объект.

На основе информации о пользователе ему подбираются объекты с нужными признаками. При коллаборативной фильтрации пользователю рекомендуются те объекты, которые понравились другим пользователям с похожими оценками.

1. Аналитический раздел

1.1 Обзор предметной области

Есть две разновидности интернет-магазинов, продажи витаминов, в зависимости от вида торговли:

- Магазины, которые продают товар со своего склада. Такой магазин – прекрасный вариант дополнительного сбыта товара;
- Магазины, которые продают товар других магазинов/людей.

Программный продукт, который будет разработан в данной работе, будет подходить для оба типов магазинов.

Предмет рекомендации: Покупка витаминов

Цель рекомендации: Облегчение выбора и покупки правильного продукта, в зависимости от того, что клиенту необходимо.

Контекст рекомендации: Система будет работать во время поиска витаминов

Источники рекомендации:

- схожие по интересам пользователи
- анализ истории действий пользователя

1.2 Существующие методы и алгоритмы

Несмотря на множество существующих алгоритмов, все они сводятся к нескольким базовым подходам, которые будут описаны далее. К наиболее классическим относятся алгоритмы Content-based (модели, основанные на описании товара), Collaborative Filtering (коллаборативная фильтрация) и некоторые другие.

1.2.1 Content-based рекомендации

Персональные рекомендации предполагают максимальное использование информации о самом пользователе, в первую очередь о его предыдущих покупках. Одним из первых появился подход content-based filtering. В рамках данного подхода описание товара (content) сопоставляется с интересами пользователя, полученными из его предыдущих оценок. Чем больше товар этим интересам соответствует, тем выше оценивается потенциальная заинтересованность пользователя. Очевидное требование здесь — у всех товаров в каталоге должно быть описание.

Историческим предметом Content-based рекомендаций чаще были товары с неструктурированным описанием: фильмы, книги, статьи. Такими признаками могут быть, например, текстовые описания, рецензии, состав актеров и прочее. Однако ничто не мешает использовать и обычные числовые или категориальные признаки.

Неструктурированные признаки описываются типичным для текста способом – векторами в пространстве слов (Vector-Space model). Каждый элемент такого вектора – признак, потенциально характеризующий интерес пользователя. Аналогично, продукт – вектор в том же пространстве.

По мере взаимодействия пользователя с системой (скажем, он покупает фильмы), векторные описания приобретенных им товаров объединяются (суммируются и нормализуются) в единый вектор и, таким образом, формируется вектор его интересов. Далее достаточно найти товар, описание которого наиболее близко к вектору интересов, т.е. решить задачу поиска n ближайших соседей.

Не все элементы одинаково значимы: например, союзные слова, очевидно, не несут никакой полезной нагрузки. Поэтому при определении числа совпадающих элементов в двух векторах все измерения нужно предварительно взвешивать по их значимости. Данную задачу решает хорошо известное в Text Mining преобразование TF-IDF, которое назначает больший вес более редким интересам. Совпадение таких интересов имеет большее значение при определении близости двух векторов, чем совпадение популярных.

$$W_{x,y} = tf_{x,y} * \log\left(\frac{N}{df_x}\right) \quad (1)$$

Где:

- $W_{x,y}$ – весь слова x в описании товара y
- $tf_{x,y}$ – частота слова x в описании товара y
- df_x – количество товаров, содержащих слова x
- N – общее количество товаров

Принцип TF-IDF здесь в той же мере применим и к обычным номинальным атрибутам, таким, как например, тип, производитель, государство. TF — мера значимости атрибута для пользователя, IDF — мера «редкости» атрибута.

Существует целое семейство похожих преобразований (например, BM25 и аналогичные), но содержательно все они повторяют ту же логику, что TF-IDF: редкие атрибуты должны иметь больший вес при сравнении товаров. Рисунок 1 ниже иллюстрирует, как именно зависит вес TF-IDF от показателей TF и IDF. Ближняя горизонтальная ось — это DF: частота атрибута среди всех товаров, дальняя горизонтальная ось — TF: логарифм частоты атрибута у пользователя.

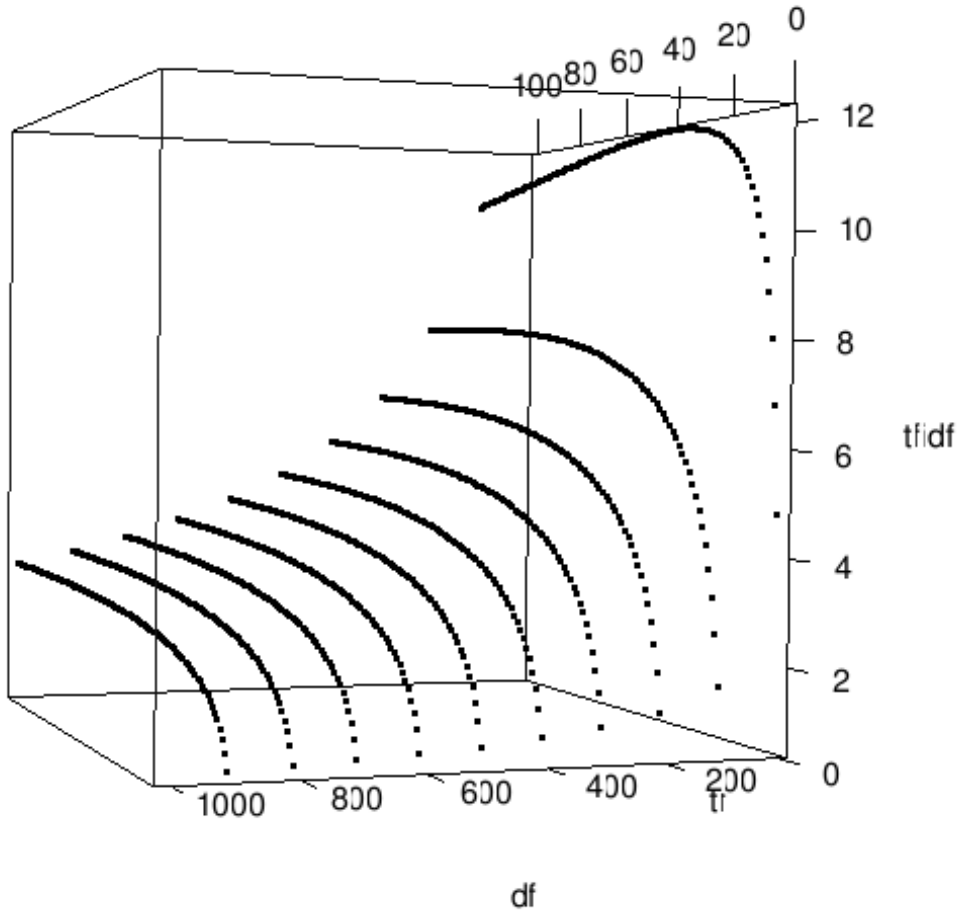

Рисунок 1 Картинка ниже иллюстрирует, как именно зависит вес TF-IDF от показателей TF и IDF

Некоторые моменты, которые можно учесть при реализации.

- При формировании vector-space представления товара вместо отдельных слов можно использовать n-граммы (последовательные пары слов, тройки и т.д.). Это сделает модель более детализированной, однако потребуется больше данных для обучения.
- В разных местах описания товара вес ключевых слов может отличаться (например, описание фильма может состоять из заголовка, краткого описания и детального описания).
- Описания товара от разных пользователей можно взвешивать по-разному. Например, можем давать больший вес активным пользователям, у которых много оценок.
- Аналогично можно взвешивать и по товару. Чем больше средний рейтинг объекта, тем больше его вес.
- Если описание товара допускает ссылки на внешние источники, то можно анализировать также всю связанную с товаром стороннюю информацию.

Видно, что content-based фильтрация почти полностью повторяет механизм query-document matching, используемый в поисковых системах типа Яндекс и Google. Отличие лишь в форме поискового запроса — здесь это вектор, описывающий интересы пользователя, а там — ключевые слова запрашиваемого документа. Когда поисковики стали добавлять персонализацию, различие стерлось еще больше.

При добавлении новой оценки вектор интересов обновляется инкрементально (только по тем элементам, которые изменились). При пересчете имеет смысл давать новым оценкам чуть больше веса, поскольку предпочтения могут меняться.

1.2.2 Коллаборативная фильтрация (User-based вариант)

Данный класс систем начал активно развиваться в 90-е годы. В рамках подхода рекомендации генерируются на основании интересов других похожих пользователей. Такие рекомендации являются результатом «коллаборации» множества пользователей. Отсюда и название метода.

Классическая реализация алгоритма основана на принципе k ближайших соседей. На пальцах – для каждого пользователя ищем k наиболее похожих на него (в терминах предпочтений) и дополняем информацию о пользователе известными данными по его соседям. Так, например, если известно, что ваши соседи по интересам в восторге от фильма «Кровь и бетон», а вы его по какой-то причине еще не смотрели, это отличный повод предложить вам данный фильм для субботнего просмотра.

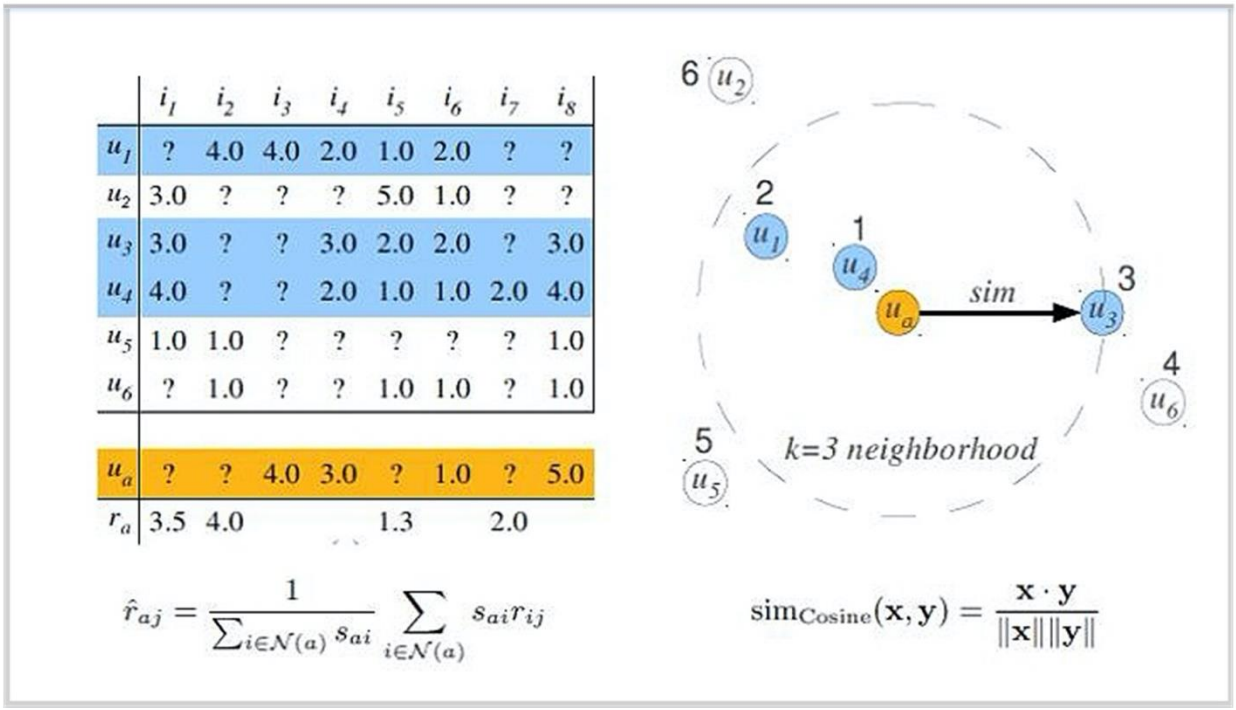

Рисунок 2 Принцип работы коллаборативной фильтрации

Рисунок 2 иллюстрирует принцип работы метода. В матрице предпочтений желтым цветом выделен пользователь, для которого мы хотим определить оценки по новым товарам (знаки вопроса). Синим цветом выделены три его ближайших соседа.

«Похожесть» – в данном случае синоним «корреляции» интересов и может считаться множеством способов (помимо корреляции Пирсона, есть еще косинусное расстояние, есть расстояние Жаккара, расстояние Хэмминга и пр.).

У классической реализации алгоритма есть один явный минус – он плохо применим на практике из-за квадратичной сложности. Действительно, как любой метод ближайшего соседа, он требует расчета всех попарных расстояний между пользователями (а пользователей могут быть миллионы). Нетрудно посчитать, что сложность расчета матрицы расстояний будет O(n2m), где n — число пользователей, а m — число товаров. При миллионе пользователей для хранения матрицы расстояний в сыром виде, потребуется минимум 4TB.

Данная проблема отчасти может быть решена покупкой высокопроизводительного железа. Но если подходить с умом, то лучше ввести корректировки в алгоритм:

- обновлять расстояния не при каждой покупке, а батчами (например, раз в день),
- не пересчитывать матрицу расстояний полностью, а обновлять ее инкрементально,
- сделать выбор в пользу итеративных и приближенных алгоритмов (например ALS).

Для того чтобы алгоритм был эффективен, важно чтобы выполнялось несколько допущений.

- Вкусы людей не меняются временем (или меняются, но для всех одинаково).
- Если вкусы людей совпадают, то они совпадают во всем.

Например, если два клиента предпочитают одни фильмы, то книги им тоже нравятся одинаковые. Так часто бывает, когда рекомендуемые товары однородны (например, только фильмы). Если это же не так, то у пары клиентов вполне могут совпадать предпочтения в еде, а политические взгляды быть прямо противоположными — здесь алгоритм будет менее эффективным.

Окрестность пользователя в пространстве предпочтений (его соседи), которую мы будем анализировать для генерации новых рекомендаций, можно выбирать по-разному. Мы можем работать вообще со всеми пользователями системы, можем задать некий порог близости, можем выбрать несколько соседей случайным образом или брать n наиболее похожих соседей (это наиболее популярный подход).

Важный этап подготовки данных — нормализация оценок.

Преимущества метода:

1) Является достаточно универсальным подходом, поэтому часто дает высокие результаты.
2) Для работы данного метода не нужна детальная информация о продуктах. В примере с книжным магазином - автор, жанр, описание книги. Вместо этого используется как история оценок самого пользователя, так и других пользователей.

Недостатки метода:

1) Работа системой, когда еще нет истории покупок (задача холодного старта).
2) Работа с новыми объектами, которые еще никто не оценил.
3) Пользователи, которые не оценивают товары
4) Ресурсоемкость вычислений, которая замедляет время работы системы.
5) Необходим большой объем данных для высокой точности предсказаний.

1.2.3 Стандартизация данных (scaling)

Поскольку все пользователи оценивают по-разному – кто-то всем подряд пятерки ставит, а от кого-то четверки редко дождешься – перед расчетом данные лучше нормализовать, т.е. привести к единой шкале, чтобы алгоритм мог корректно сравнивать их между собой.

Естественно, предсказанную оценку затем нужно будет перевести в исходную шкалу обратным преобразованием (и, если нужно, округлить до ближайшего целого числа).

Нормализовать можно несколькими способами:

- центрированием (mean-centering) — из оценок пользователя просто вычитаем его среднюю оценку,

  ** актуально только для небинарных матриц*
- стандартизацией (z-score) — в добавок к центрированию делим оценку ее на стандартное отклонение у пользователя,

  ** здесь после обратного преобразования рейтинг может выйти за пределы шкалы (т.е. например, 6 по пятибальной шкале), но такие*

ситуации довольно редки и решаются просто округлением в сторону ближайшей допустимой оценки.

- двойной стандартизацией — первый раз нормируем оценками пользователя, второй раз — оценками товара.

  Если у фильма «Самый лучший фильм» средняя оценка 2.5, а пользователь ей ставит 5, то это сильный фактор, говорящий о том, что такие фильмы ему явно по вкусу.

«Похожесть» или корреляцию предпочтений двух пользователей можно считать разными способами. По сути, нам надо просто сравнить два вектора. Перечислим несколько наиболее популярных.

1. Корреляция Пирсона — классический коэффициент, который вполне применим и при сравнении векторов.

$$\rho = \frac{\sum_i^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i^N (x_i - \bar{x})^2 \sum_i^N (y_i - \bar{y})^2}} \quad (2)$$

   Основной его минус — когда пересечение по оценкам низкое, корреляция может быть высокой просто случайно.

   Для борьбы со случайно завышенной корреляцией можно домножить на коэффициент 50 / min(50, Rating intersection) или любой другой damping factor, влияние которого уменьшается с ростом числа оценок.
2. Корреляция Спирмана

   Основное отличие — коэффициент ранговый, т.е. работает не с абсолютными значениями рейтингов, а с их порядковыми номерами. В целом дает результат очень близкий к корреляции Пирсона.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (3)$$

3. Косинусное расстояние

Еще один классический коэффициент. Если приглядеться, косинус угла между стандартизированными векторами — это и есть корреляция Пирсона, одна и та же формула.

$$similarity = \cos(\theta) = \frac{A * B}{||A||||B||} = \frac{\sum_{i=1}^{N} A_i B_i}{\sqrt{\sum_{i=1}^{N} A_i^2}\sqrt{\sum_{i=1}^{N} B_i^2}} \quad (4)$$

Почему косинусное — потому что, если два вектора сонаправлены (т.е. угол между ними нулевой), то косинус угла между ними равен единице. И наоборот, косинус угла между перпендикулярными векторами равен нулю.

Интересное развитие коллаборативного подхода — так называемые Trust-based recommendations, в которых учитывается не только близость людей по интересам, но также их «социальная» близость и степень доверия между ними. Если, например видим, что на фейсбуке девушка периодически заходит на страницу с аудиозаписями подруги, значит доверяет её музыкальному вкусу. Следовательно, в рекомендации девушке можно вполне подмешивать новые песни из плейлиста подруги.

Преимущества метода:

1) Не требует большой группы пользователей для достижения высокой точности рекомендаций.
2) Новые элементы можно рекомендовать сразу, как только у них появляются заполненные характеристики.

Недостатки метода:

1) Сильная зависимость от предметной области, полезность рекомендаций ограничена.
2) Профиль пользователей и элементов должен состоять из одинакового набора характеристик, чтобы их можно было сравнивать.
3) Работа с новым пользователем, для которого еще не данных

1.3Способы получения информации:

Для получения необходимой информации можно использовать следующие методы:

- Оценка каких-либо объектов пользователем/ями (рейтинг);
- Купленные товары
- Сравнение каких-либо объектов пользователем;
- Выбор лучшего объекта пользователем из группы объектов;
- Отслеживание истории просмотров пользователя;
- Отслеживание поведения пользователя в интернете.

1.4Обзор существующих решений

1.4.1 Apple Music

В Apple Music существует раздел «Для вас», в котором рекомендуются музыкальные композиции по предпочтениям пользователя. При этом учитываются такие параметры, как:

- лайки
- прослушивания
- медиатека пользователя
- ваши указания — в начале работы пользователя в Apple music система выясняет, какие жанры и исполнители больше нравятся. Позже это возможно изменить.

В процессе определения музыкальных композиций определѐнного жанра, которые следует порекомендовать пользователю, принимают участие **эксперты** из разных музыкальных журналов, что выгодно отличает данную рекомендательную систему от остальных.

1.4.2 Netflix

Одно из самых больших достижений Netflix – это рекомендация контента, который пользователи. Комбинируя машинное обучение (machine learning) и алгоритмы, компания изучает предубеждения потребителей и рекомендует им то, что они обычно не смотрят, но все же отвечает их предпочтениям.

Компания описывает эту систему как трехногий стул. Первая ножка стула – это пользователи и, в частности, каждый отдельный профиль (у каждого подписчика может быть до пяти профилей) и его привычки. Платформа учитывает как то, что пользователи смотрят в данный момент, так и ранее просмотренный контент, который просматривался год назад или недавно, а также в какое время суток он было просмотрен. Затем эти данные объединяются со второй ножкой стула – подробным тегом для каждого фильма или сериала. Теги создаются людьми, которые внимательно смотрят каждый эпизод или фильм, и содержат неожиданные подробности о них. Какое у них настроение, какие эмоции они вызывают, есть ли самостоятельный главный герой, коррумпированное правительство, происходит ли действие в космосе.

Затем данные об использовании сервиса и теги переходят к третьей ножке стула – алгоритмам. С ними Netflix понимает, что важно для аккаунта в данный момент. Что важнее? Вчерашний фильм или тот сериал, который вы смотрели год назад? Неужели 10 минут сериала сегодня имеют больший вес, чем просмотр двух сезонов за раз месяц назад? Все эти данные приводят к созданию «сообществ по вкусам», в которые входят профили со всего мира.
Ключевой момент в том, что зрители попадают в разные сообщества – фактор, который влияет на то, что отображается на главном экране сервиса. Важная деталь: у Netflix нет причин рекомендовать определенный контент больше, чем другой. Создателям контента выплачивается фиксированная плата, поэтому не имеет значения, смотрят ли фильм или сериал 1 или 100 миллионов человек – расходы для Netflix такие же. Вот почему компания ограничивается рекомендациями, которые просто удерживают пользователей в сервисе.

1.4.3 Spotify

В отличие от Apple Music, где рекомендации делают музыкальные редакторы, Spotify использует алгоритмы. Как и Netflix, Spotify создает вкусовой профиль для всех пользователей, беря за основу то, что они слушают. Благодаря этому профилю платформа дает рейтинг близости к исполнителям, который основан на алгоритме, определяющем важность исполнителя для слушателя. Алгоритм также учитывает жанры, которые пользователь слушает, чтобы определить, в каких из них вы, скорее всего, захотите услышать что-то новое.

Работы алгоритма заключается в поиске других пользователей, у которых есть плейлисты с пользовательскими любимыми музыкантами и песнями. Из этих плейлистов, он берет песни, которые пользователь не слушал, фильтрует их по его вкусовому профилю и добавляет в рекомендации. Его поведение с Discover Weekly влияет на его вкусовой профиль, и любая понравившаяся, пропущенная или добавленная песня в список воспроизведения является сигналом того, что нравится, а что нет.

1.4.4 Необходимость от людей

У Spotify есть системы анализа естественного языка, которые ищут в популярных блогах в поисках новой сенсационной музыки, которая становится частью плейлистов Fresh Finds. Система анализа звука определяет качество совершенно новой музыки, которой нет в чьем-либо блоге или плейлисте.

Примечательно, что эти ключевые алгоритмы сильно зависят от людей. И речь идет не об инженерах и программистах, разрабатывающих алгоритмы, а о миллионах пользователей Spotify, которые добавляют треки в плейлисты, и десятках сотрудников и фрилансеров, которые тегируют контент Netflix. Без этого алгоритмы были бы абсолютно неспособны принять правильное решение для отдельного пользователя.

1.5 Вывод

Существует два наиболее распространённых подхода к построению рекомендательных систем — **коллаборативная фильтрация** и **рекомендации на основе содержимого**. У оба метода есть большие преимущества и недостатки. Они описаны в подразделе выше. Большинство из них взаимоисключающие. Поэтому цель данной работы, создать гибридный алноритм, который минимизирует недостатки этих алгоритмов отдельностью.

В следующих разделах необходимо найти решение для следующих проблемах:

1. Как система может автоматически создать профиль пользователя и затем улучшать в процессе обновления данных?
2. Как определить какой элемент соответствует предпочтениям пользователя?
3. Как автоматически извлекать информацию о продукте, чтобы избежать ручного заполнения?
4. Как для данного пользователя, для которого мы хотим сделать рекомендацию, определить пользователей, которые имеют схожие предпочтения?
5. Как измерять схожесть между пользователями?
6. Что делать если имеется мало данных о рейтингах?
7. Какие подходы могут быть объединены, и какие условия должны выполняться, чтобы это могло быть сделано?
8. Должны ли разные техники использоваться в разных ситуациях, либо результат каждой должен браться с определенным весом?
9. Как результат разных методов должны быть взвешены, чтобы на выходе получить один результат?

Список использованных источников

1. Altarena. Интернет издание [Электронный ресурс]. – https://altarena.ru/spotify-i-netflix-kak-algoritmy-ponimayut-chto-my-hotim-smotret-i-slushat/ (дата обращения: 22.12.2020)
2. Habr. Сообщество программистов [Электронный ресурс]. – https://habr.com/ru/company/lanit/blog/420499/ (дата обращения: 25.12.2020)
3. CMS Magazine. Digital журнал [Электронный ресурс]. – https://cmsmagazine.ru/journal/research-7-algoritmov-tovarnyh-rekomendacij/ (дата обращения: 15.12.2020)
4. Mkechinov. Блог студия разработки [Электронный ресурс]. – https://mkechinov.ru/recommenders-libraries.html (дата обращения: 14.12.2020)