МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.04.03 Прикладная информатика
(код и наименование направления подготовки)

Информационные системы и технологии корпоративного управления
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему «Исследование алгоритмов построения рекомендательных систем на основе анализа интересов пользователей»

Студент К.Ю. Калугин
(И.О. Фамилия) (личная подпись)

Научный руководитель канд. пед. наук, доцент, О.М. Гущина
(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Оглавление

Введение 4
Глава 1 Исследование рекомендательных систем 8
1.1 Актуальность рекомендательных систем в области малой электронной коммерции 8
1.2 Обзор литературы, посвященной исследованию рекомендательных систем для малых предприятий розничной торговли 10
1.3 Подходы к построению рекомендательных систем 12
1.4 Выбор оптимального подхода к построению рекомендательных систем в рамках малой электронной коммерции 20
1.5 Постановка задачи прогнозирования пользовательского интереса 22
Глава 2 Описание принципа работы предлагаемой гибридной рекомендательной системы 24
2.1 Краткое описание принципа работы гибридной рекомендательной системы в области малой электронной коммерции 24
2.2 Неперсонализированная рекомендательная система 25
2.3 Фильтрация на основе содержания 27
Глава 3 Практическая реализация гибридной рекомендательной системы 41
3.1 Стек технологий, использованный при реализации предлагаемой рекомендательной системы 41
3.2 Классы «UserSupercalss» и «ProductSuperclass» 42
3.3 Поведенческий профиль пользователя 43
3.4 Анализатор содержания описаний 45
3.5 Анализатор содержания характеристик 47
3.6 Анализатор содержания изображений 47
3.7 Реализация меры подобия 49
3.8 Формирование пользовательских рекомендаций 50
3.9 Требования к пользовательской стороне рекомендательной системы 51
Глава 4 Тестирование разработанной рекомендательной системы 53

4.1 Тестирование качества прогнозирования интереса пользователя рекомендательной системы 53
4.2 Тестирование возможности интеграции рекомендательной системы в приложения малой электронной коммерции 58
4.3 Анализ результатов проведенных экспериментов 65
4.4 Развитие предлагаемой рекомендательной системы в дальнейшем 66
Заключение 68
Список используемой литературы 70
Приложение А Блок-схема алгоритма стемминга Портера 74
Приложение Б UML-диаграмма классов, описывающих действия пользователя с товарами в приложении 76
Приложение В UML диаграмма-классов, образующих поведенческий профиль пользователя 77
Приложение Г Блок-схема алгоритма преобразования пикселей изображения из RGB модели в HSB 78
Приложение Д Блок-схема алгоритма k-means 79

Введение

За последнее десятилетие электронная торговля в экономическом пространстве приобрела колоссальное значение. И действительно, согласно результатам исследования, проведенного центром исследований в области розничной торговли, в Европе доля оборота бизнеса, занимающегося электронной коммерцией, увеличилась с 9% (2004 год) до 16,7% (2016 год). Также наблюдается тенденция к увеличению доли электронной торговли в обороте и в США. В 2018 году процент продаж электронной коммерции, составил 42,5% от общего объема продаж, в то время как розничные продажи составили 6,4% [18]. На территории же Российской Федерации за 2018 год объем российского рынка онлайн-торговли вырос на 59% по сравнению с показателем предыдущего года и достиг 1,66 трлн руб., исходя из презентации исследования Ассоциации компаний Интернет-торговли (АКИТ) [3].

Из озвученных цифр можно сделать вывод, что во всём мире, несомненно, происходит увеличение доли оборота в сфере электронной торговли, поэтому в ситуации растущей конкуренции бизнесменам следует прилагать больше усилий в рассматриваемой области для привлечения новых клиентов и поддержания их удовлетворенности сервисом, что в свою очередь подразумевает необходимость в специальных инструментах [12].

Одним из таких инструментов, который был представлен в 1992 году, являются рекомендательные системы. Основная цель данного инструмента – это составление списка рекомендаций объектов, хранящихся в приложении, которые бы потенциально могли бы заинтересовать пользователя, исходя из его личных характеристик или журнала взаимодействия с объектами (оценивание/просмотр/добавление в корзину). Рекомендательные системы могут оказать существенное влияние на доход, путём грамотного продвижения товара и удовлетворения спроса пользователей.

Актуальность исследования заключается в том, что разработке рекомендательных систем для малой электронной коммерции уделяется

крайне мало внимания, в результате чего традиционные рекомендательные решения довольно проблематично применять в озвученной предметной области по причине незначительного объёма имеющихся данных и отсутствия точной информации вернется ли пользователь в дальнейшем [11].

Целью магистерской диссертации является исследование алгоритмов построения рекомендательных систем в области малой электронной коммерции, направленных на формирование индивидуального списка товаров для пользователя. В данном случае под алгоритмом подразумевается ряд шагов, выполнение которых приведет к построению наиболее эффективной рекомендательной системы в рассматриваемой предметной области.

Рекомендательную систему планируется реализовать в виде библиотеки, которую можно будет интегрировать в веб-приложения, реализованные на фреймворке Spring.

Для достижения указанной цели были поставлены следующие **исследовательские задачи**:

– проанализировать существующие подходы к построению рекомендательных систем;

– провести анализ проблем, свойственных области малой электронной коммерции, и на их основе выбрать подходящий подход к построению рекомендательной системы;

– описать механизм работы рекомендательной системы, адаптированной под область малой электронной коммерции, посредством использования различных методов и алгоритмов;

– практически реализовать рекомендательную систему на основе описанного механизма работы;

– протестировать качество прогнозирования интереса пользователя и возможность интеграции разработанной рекомендательной системы в сторонние приложения.

Объектом исследования стал процесс прогнозирования интересов пользователя в малой электронной коммерции. **Предметом исследования**

работы является разработка алгоритма для реализации рекомендательной системы, прогнозирующей интерес пользователя к товару в области малой электронной коммерции.

Главной гипотезой, положенной в основу магистерской диссертации, является то, что если осуществить реализацию рекомендательной системы, учитывающей проблемы, свойственные представителям малой электронной торговли, то её внедрение способно повысить эффективность маркетинга, тем самым поспособствовав увеличению объёма продаж товара и лояльности покупателей.

Методологической основой исследования являются работы отечественных и зарубежных специалистов, посвященные проблемам изучению и моделированию рекомендательных систем.

Научная новизна выполненных исследований заключается в разработке наиболее эффективного алгоритма построения рекомендательной системы в области малой электронной коммерции.

Теоретическая значимость работы состоит в уточнении и развитии теоретических основ разработки рекомендательных систем для построения персональных рекомендаций на основе личных предпочтений пользователей.

Практическая значимость определяется тем, что поскольку число рекомендательных алгоритмов для малых предприятий электронной розничной торговли невелико, то создание нового функционального алгоритма может предоставить прекрасную возможность для увеличения продаж малых предприятий электронной коммерции.

Публикации по теме исследования. Основные результаты теоретической части исследования изложены в статьях:

1. Калугин К. Ю. Выбор оптимального подхода к построению рекомендательной системы в сфере малой электронной коммерции. VI Международная научно-практическая конференция (школа-семинар) молодых ученых «Прикладная математика и информатика: современные исследования в области естественных и технических наук» – 2020 – С. 71-74.

2. Калугин К. Ю. Гибридная рекомендательная система в сфере малой электронной коммерции. VII Международная научно-практическая конференция (школа-семинар) молодых ученых «Прикладная математика и информатика: современные исследования в области естественных и технических наук» – 2021.

Положения, выносимые на защиту:

1. Результаты исследования области применения рекомендательных систем и принципов их построения.

2. Обоснование выбора подхода к построению рекомендательной системы в области малой электронной коммерции.

3. Описание принципа функционирования предложенной рекомендательной системы.

4. Практическая реализация рекомендательной системы для малой электронной коммерции и результаты её тестирования.

В ходе работы над диссертацией было написано четыре главы.

Первая глава является теоретической и описывает назначение с характеристиками основных подходов к построению рекомендательных систем. А также обосновывает выбор подхода к построению рекомендательной системы для малой электронной коммерции.

Вторая глава также является теоретической и подробно описывает выбор методов и алгоритмов для реализации выбранного подхода к построению рекомендательный системы.

Третья глава носит практический характер. В этой главе описываются все основные этапы разработки рекомендательной системы.

В четвертой главе осуществляется тестирование качества прогнозирования интереса пользователя посредством нахождения похожих товаров на тот, что его заинтересовал ранее, а также её возможность интеграции в сторонние приложения.

Работа изложена на 75 страницах и включает 24 рисунка, 3 таблицы, 33 источника и 5 приложений.

Глава 1 Исследование рекомендательных систем

1.1 Актуальность рекомендательных систем в области малой электронной коммерции

Первое полугодие 2020 года стало самым непредсказуемым за все время развития электронной коммерции в мире и переживает на данный момент одну из самых удивительных трансформаций. По меркам рынка в одно мгновение изменились все догмы индустрии – покупательское поведение, способы и место совершения покупки, средний чек и другие факторы.

Пандемия буквально вынудила многих покупателей приобрести новый опыт: более 40% приобретали онлайн то, что раньше покупали только в розничной точке продаж, 25% за период изоляции нашли новые магазины и также остались лояльны к ним после снятия ограничений.

Объем российского рынка электронной коммерции в 2020 году достиг 2,7 трлн рублей, увеличившись на 58% в сравнении с 2019-м. Об этом свидетельствуют данные Data Insight.

Возрастание темпа роста рынка в сравнении с прошлым годом аналитики объяснили распространением пандемии коронавируса COVID-19, по причине которой люди стали чаще проводить время дома и заказывать товары через интернет [6].

Мотивирующие цифры несут в себе и новые вызовы. Стоит понимать, что чем больше покупок совершается онлайн, тем выше становятся требования к продавцам. Более 40% пользователей отказываются от онлайн-магазина, если их что-то не устраивает в процессе совершения покупки. При выборе магазина решающими факторами становятся бесплатная и быстрая доставка, скидочные купоны и, разумеется, персонализированные пользовательские предложения.

В результате сложившейся ситуации рекомендательные системы стали очевидным выбором для компаний на рынке электронной коммерции [9].

Рекомендательные системы обычно принято описывать, как «интеллектуальное программное обеспечение, предоставляющее легкодоступные высококачественные рекомендации для онлайн-покупателей». Такие системы на сегодняшний день считаются «серьезным бизнес-инструментом».

Рекомендательные системы в электронной коммерции имеют множество преимуществ как для предпринимателя, так и для простых пользователей. И если для пользователя – это предоставление товаров, которые потенциально могут вызвать у него интерес, то для предпринимателя это целый ряд преимуществ:

1. Увеличение количества продаж. Наиболее веская причина, почему следует инвестировать в рекомендательные системы. При внедрении рекомендательных систем наблюдается повышение коэффициента конверсии пользователей.

2. Продажа разнообразного товара. Рекомендательные системы предлагают пользователям те товары, которые зачастую остаются незамеченными. В подобном продвижении непосредственно заинтересован предприниматель.

3. Увеличения количества перекрестных продаж продукции. Рекомендательные системы могут предлагать при покупке товара некие аксессуары, связанные с покупкой.

4. Повышение у клиентов симпатии к магазину. Клиенты в основном возвращается на те платформы, где присутствует удобный сервис, а их потребности были удовлетворены.

5. Понимание потребительского спроса, что в свою очередь, может служить для контроля запасов и закупки востребованной продукции.

Крупные компании, специализирующиеся на электронной коммерции, повсеместно внедряют рекомендательные системы в свои сервисы, к примеру такие гиганты, как Amazon, eBay и Netflix, не только активно эксплуатируют данный инструмент, но интенсивно развивают.

Однако, если речь заходит о малых электронных предприятиях розничной торговли, то рекомендательные системы в этом случае сталкиваются с рядом проблем:

- разреженность данных;
- небольшое число возвращающихся пользователей;
- ограниченные вычислительные возможности.

Данные ограничения вынуждают использовать иные подходы или модификации распространённых методов построения рекомендательных систем.

1.2 Обзор литературы, посвященной исследованию рекомендательных систем для малых предприятий розничной торговли

В области рекомендательных систем опубликован значительный объем исследований. Подавляющее большинство этих исследований было посвящено выявлению новых рекомендательных алгоритмов с более повышенной точностью или же пересмотру ранее использовавшихся методов с целью поиска оптимальных характеристик для лучшей эффективности. Однако лишь в немногих исследованиях изучался вопрос рекомендации товара для малых предприятий электронной розничной торговли.

В своём исследовательском труде «Product recommendation for small-scale retailers» Marius Kaminskas представил вместе с соавторами новый гибридный подход, построенный на сравнении товаров из истории просмотра с текстовым описанием товаров из базы данных. Тестирование проводилось на основе данных двух небольших интернет-магазинов. Также важно отметить, что для решения проблемы разреженности данных исследователи фокусируются только на факте просмотре товара, а не на его покупке [28]. Эти же авторы провели еще один эксперимент спустя год, используя схожий подход. Условия проведения эксперимента были аналогичны тем, что были и

в прошлый раз, данные использовались от всё тех же розничных предпринимателей. Однако, помимо нажатия на товар, новый метод учитывал факт добавления в корзину [29]. Важно отметить, что в обоих экспериментах исследователи проводили оффлайн и онлайн тестирование рекомендательной системы. Обе рекомендательные системы были внедрены в веб-сайты мелких предприятий розничной торговли, что позволило оценить их реальное влияние на продажи. Практическим путём было установлено, что разработанные решения привели к увеличению объема сделанных заказов, что принесло более высокие доходы в обоих случаях.

Ming Li вместе коллегами в научной работе «Grocery shopping recommendations based on basket-sensitive random walk» предложил смоделировать проблему рекомендаций в виде двудольного графа с пользователями и товарами в качестве узлов, а также ребрами, отображающих покупку товара пользователем [31]. Авторы вычислили сходство товаров, используя вероятности перехода между товарами в графе. Первый порядок вероятностей перехода позволял установить сходство только между товарами, которые были куплены вместе. Однако более высокие порядки сходства позволили установить сходство между товарами, которые не появлялись в одних и тех же корзинах, но были связаны через общих соседей, что позволило решить проблему разреженности данных.

В исследовании Junnan Chen и соавторов «Product recommendation system for small online retailers using association rules mining», посвященного рекомендательным систем для мелкомасштабных розничных магазинов, был предложен программный продукт на базе правил ассоциаций и общих характеристик [23]. Рекомендательная система в данном исследовании использует алгоритм Apriori, а также частотный анализ для учитывания демографических особенностей при построении модели правил ассоциаций. Результаты показали, что разработанное решение является масштабируемым и эффективным, так как время выполнения системы увеличивалось линейно с

размером тестового набора, а список рекомендаций формировался менее чем за 0,1 секунды.

1.3 Подходы к построению рекомендательных систем

На рисунке 1 представлены основные подходы, используемые при построении рекомендательных систем.

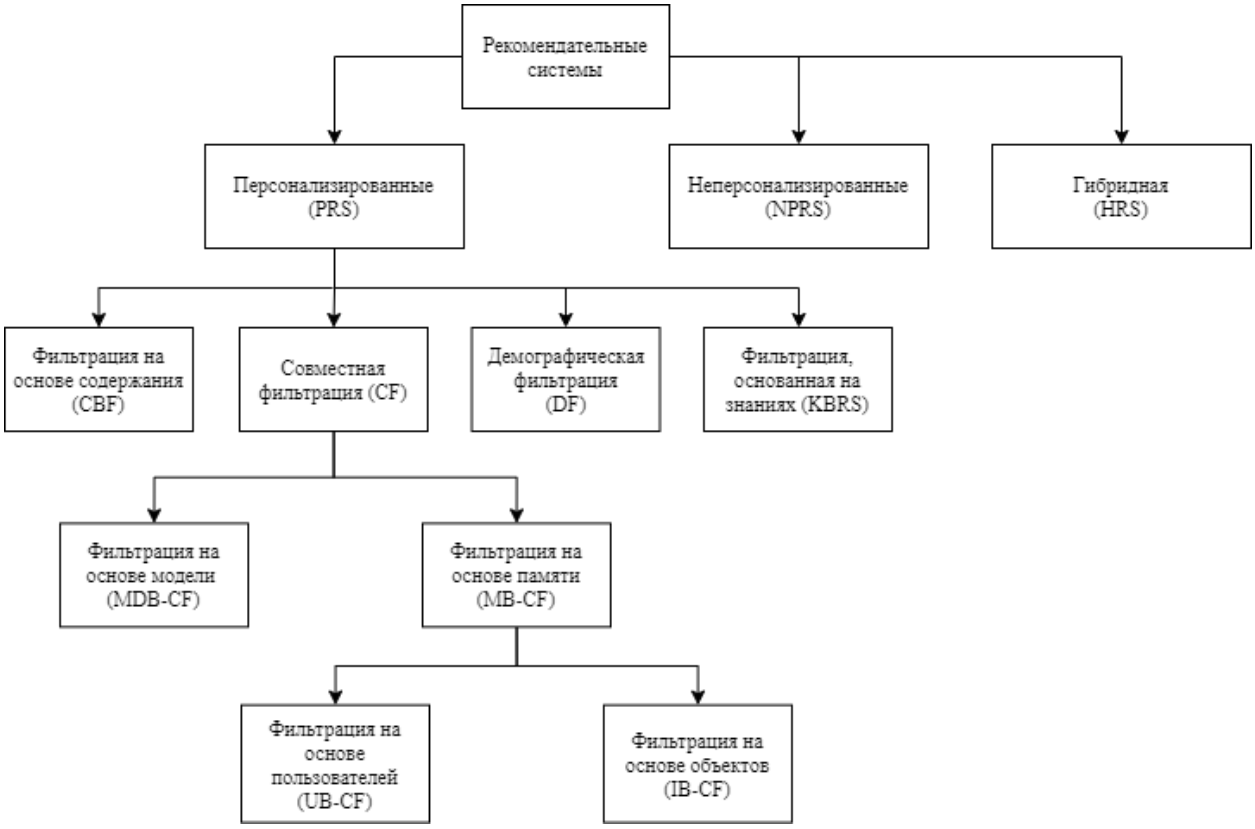


Рисунок 1 – Основные подходы к построению рекомендательных систем

Рекомендательные системы как видно из рисунка 1 бывают гибридными, неперсонализированными и персонализированными.

Гибридная рекомендательная система (Hybrid Recommender System (HRS)) сочетает в себе два и более рекомендательных подходов.

Неперсонализированные рекомендательные системы (Non-Personalized Recommender System (NPRS)) являются самым известным и простым видом

рекомендательных систем. Рекомендации в данном случае ориентированы на всех пользователей, а не на какого-то конкретного. Рекомендации могут быть основаны на количестве просмотров товара или его на среднем рейтинге и т.п.

Зачастую при неперсонализированном подходе рекомендательная система предлагает обратить пользователям на наиболее востребованные товары среди пользователей (рисунок 2).

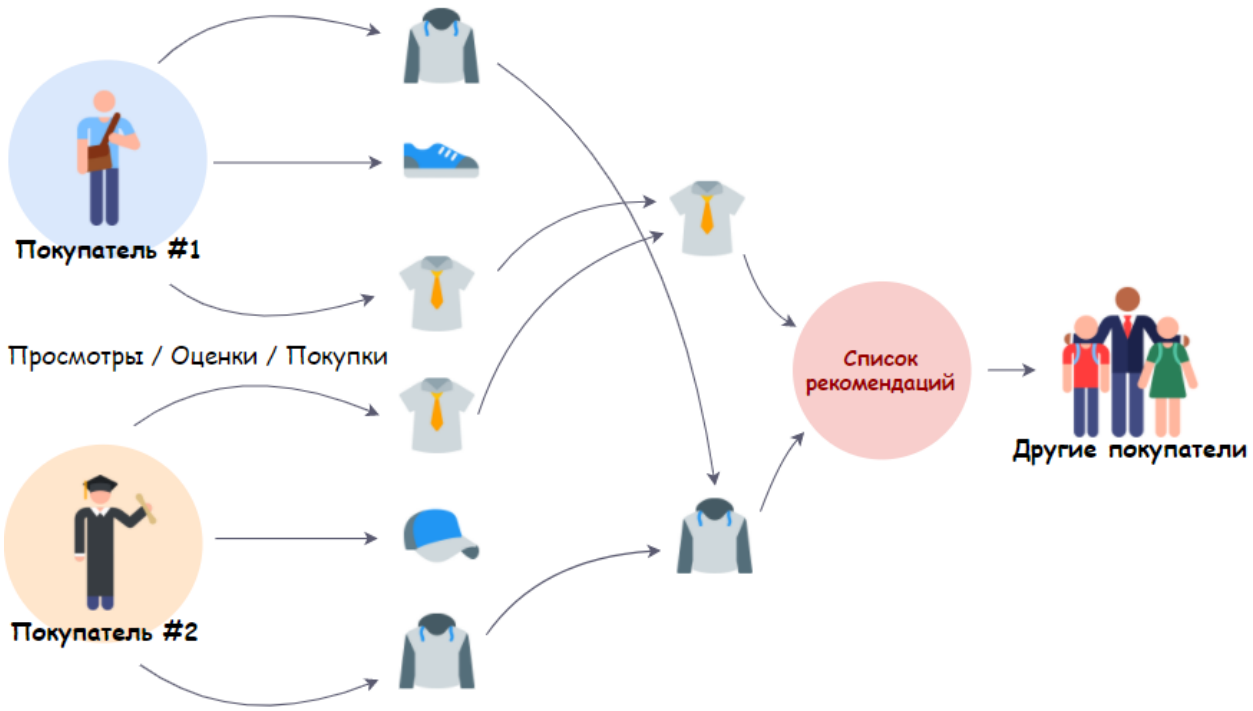


Рисунок 2 – Принцип работы неперсонализированного подхода

Ограничением модели является только то, что она является одномерной, то есть рекомендации остаются неизменными для каждого пользователя.

Персонализированными рекомендательными системы (Personalized Recommender System (PRS)) называются системы, формирующие список объектов, потенциально способных вызвать интерес пользователя, на основе его личных предпочтений. Данными в этом случае могут выступать, как и персональная (пол, возраст, город проживания и т. д.), так и поведенческая (взаимодействие с объектами системы) информация.

Персонализированные рекомендательные системы в свою очередь подразделяются на ряд подходов.

1.3.1 Фильтрация на основе содержания

Фильтрация на основе содержания (Content-based Filtering (CBF)) рекомендует объекты на основе профиля пользователя и профиля объекта. В данном случае для формирования списка рекомендаций используются описательные характеристики объекта и их соответствие предпочтениям пользователя. Для этого в рекомендательных системах используются ключевые слова, чтобы описать объекты, а профиль пользователя отражает оценку определенных тэгов или их совокупности (рисунок 3).

Рисунок 3 – Принцип работы фильтрации на основе содержания

Например, при выборе книги на сайте покупатель просматривал литературу преимущественно жанра «детектив». В результате чего на базе данной информации рекомендательная система сформирует для этого покупателя список книг аналогичного жанра, так как ранее покупатель неявным образом высказал своё предпочтение.

Преимуществом подхода является то, что отношения между объектами являются более стабильными, чем между пользователями, да и в целом

зачастую база объектов имеет меньший размер по сравнению с базой пользователей, в результате чего скорость работы системы остается хорошей, даже если количество пользователей увеличивается.

При данном подходе применяется байесовский классификатор, деревья решений, кластеризация и другие методы машинного обучения [32].

1.3.2 Совместная фильтрация

Идея совместной фильтрации (Collaborative Filtering (CF)) заключается в том, что схожим по интересам пользователям могут заинтересовать одни и те же объекты, хранящиеся в системе.

Совместная фильтрация подразделяется на два типа:

– фильтрация на основе памяти;

– фильтрация на основе модели.

Рассмотрим каждый вид совместной фильтрации более подробно.

1.3.2.1 Фильтрация на основе памяти

Совместная фильтрация на основе памяти (Memory-Based Collaborative Filtering (MB-CF)) используют всю базу данных для создания списка рекомендаций на основе предпочтений пользователей.

Фильтрация на основе памяти делится на следующие алгоритмы:

1. Совместная фильтрация на основе пользователей (User-Based Collaborative Filtering (UB-CF)) представляет собой поиск «друзей» по интересам для активного пользователя, с целью рекомендации тех объектов, которые высоко оценили «друзья» пользователя, но не он сам (рисунок 4).

Данный алгоритм находит сходство между пользователями на основе оценок, которые они ранее давали разным объектам.

Алгоритм показывает достаточно высокую эффективность, однако требует много времени и вычислительных ресурсов.

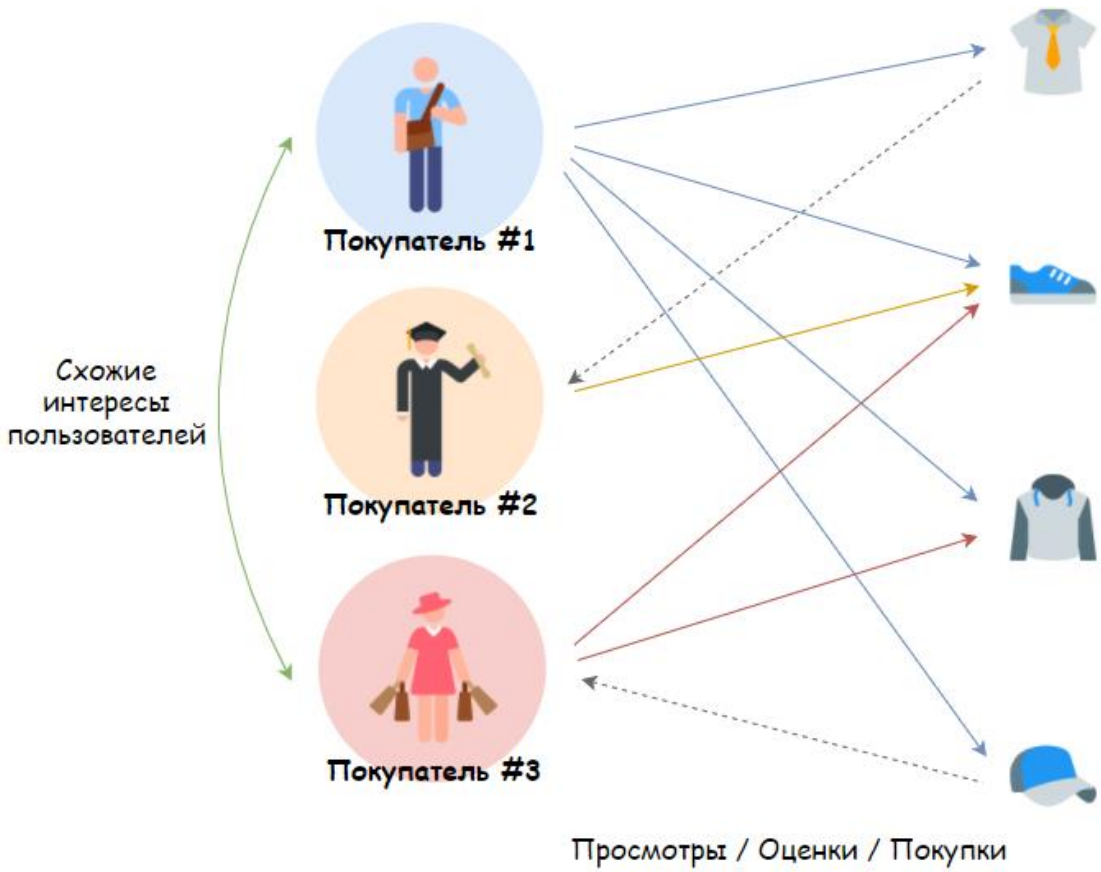


Рисунок 4 – Принцип работы совместной фильтрации на основе пользователей

2. Совместная фильтрация на основе объектов (Item-Based Collaborative Filtering (IB-CF)) достаточно похожа на фильтрацию на основе пользователей, однако в данном случае вместо поиска «друзей» по интересам, производится поиск похожих объектов на основе оценок пользователей (рисунок 5).

По своей сути данный алгоритм является абсолютно симметричен фильтрации на основе пользователей. Только теперь будем считать, что объект понравится пользователю, если ему понравились похожие объекты.

Совместная фильтрация на основе объектов требует гораздо меньше ресурсов, чем совместная фильтрация на основе пользователей, поскольку поиск похожих объектов занимает гораздо меньше времени, чем поиск похожих пользователей, по причине того, что база данных пользователей зачастую превышают по размеру базу данных объектов.

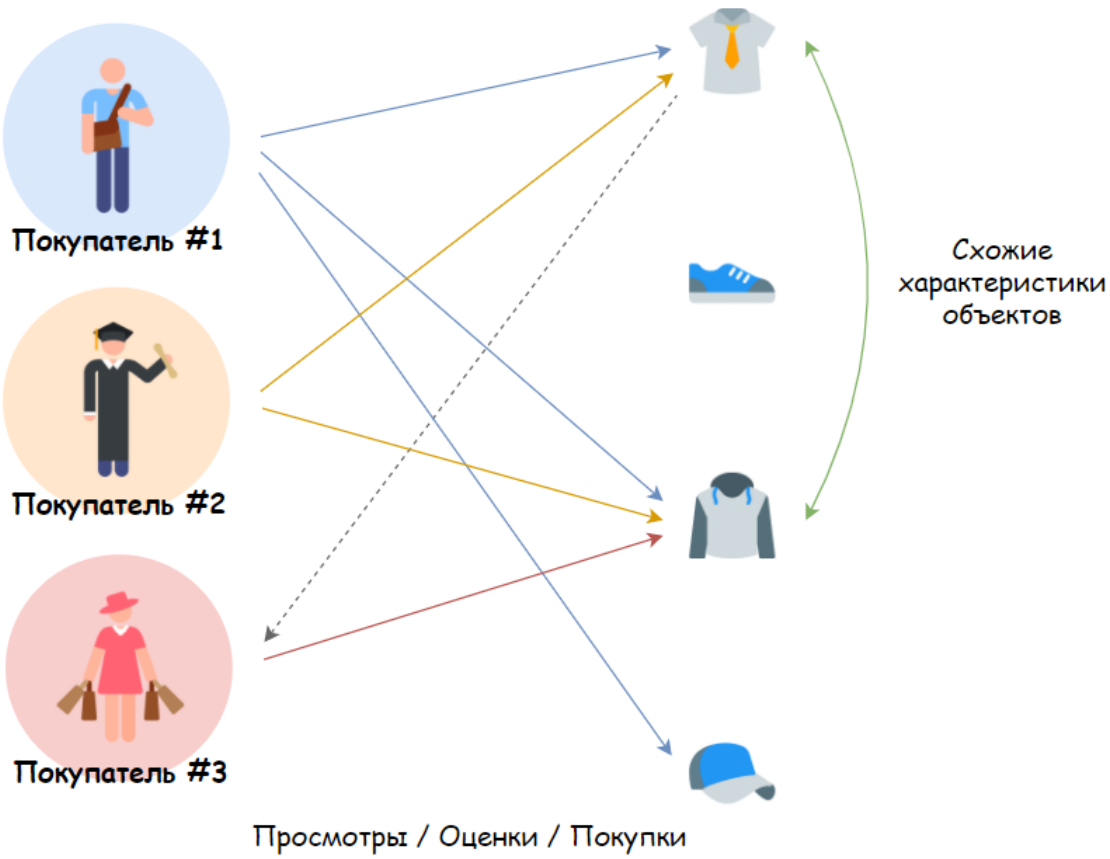


Рисунок 5 – Принцип работы совместной фильтрации на основе объектов

1.3.2.2 Фильтрация на основе модели

Совместная фильтрация на основе моделей (Model-based Collaborative Filtering (MDB-CF)) решает проблему продолжительной длительности вычислений пользовательских рекомендаций, свойственную совместной фильтрация на основе памяти, которая возникает при работе с большим количеством данных. Проблема решается путем извлечения подмножества пользователей и объектов в качестве репрезентативной «модели» для выработки рекомендаций. Уменьшенная размерность обеспечивает преимущества в виде скорости и масштабируемости.

Общие модели уменьшения размерности рейтинговой матрицы включают в себя байесовские сети, кластеризацию, поиск ассоциативных правил и многое другое. Цель этих методов заключается в выявлении скрытых

факторов, объясняющих взаимосвязь рассматриваемых оценок. Одним из недостатков подхода является то, что обобщения часто приводят к более низкому уровню точности, чем у подхода на основе памяти.

Для уменьшения размеров матрицы часто используется факторизация матрицы. Факторизацию матрицы можно рассматривать как разбиение большой матрицы на ряд более мелких. Одним из популярных алгоритмов факторизации матрицы является алгоритм сингулярного разложения (SVD).

1.3.3 Демографическая фильтрация

Демографическая фильтрация (Demographic Filtering (DF)) классифицирует пользователей на основе их демографической информации, помещая в соответствующую группу. Демографическая фильтрация также как и совместная формирует взаимосвязи между людьми, но используют для этого демографические данные о пользователях вместо поведенческих.

Демографическая фильтрация создает пользовательские группы на основе демографических характеристик, после чего отслеживая совокупное покупательское поведение пользователей в этих группах, формирует список рекомендаций. Системы демографической фильтрации используют демографическую информацию, такую как возраст, пол и образование, чтобы отнести пользователя к некой группе. Новому пользователю изначально рекомендуются объекты интересующую ту группу, к которой его определил алгоритм, однако при взаимодействии с системой он также начинает влиять на список рекомендаций своей группы (рисунок 6).

Примером демографической фильтрации может служить выдача объявления о продаже некого имущества в том же городе, из которого пользователь прошёл аутентификацию в приложении.

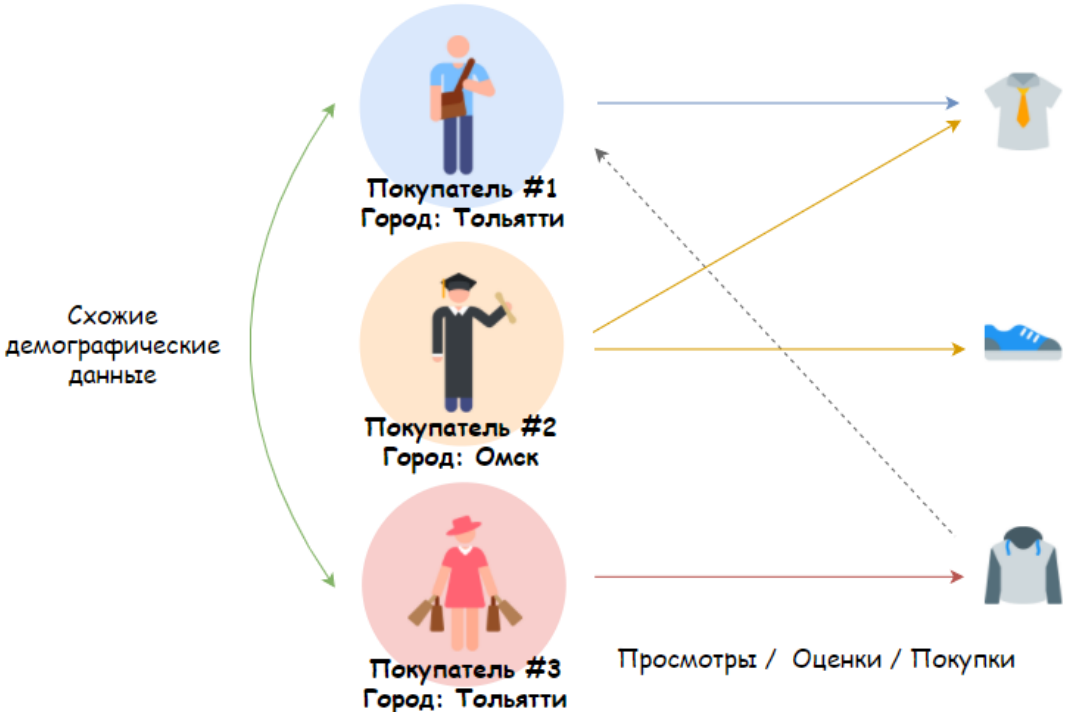


Рисунок 6 – Принцип действия демографической фильтрации

1.3.4 Рекомендательные системы, основанные на знаниях

Рекомендательная система, основанная на знаниях (Knowledge-based Recommender Systems (KBRS)) рекомендует объекты на базе явно указанных пользовательских предпочтений, например, с помощью анкеты (рисунок 7).

Рисунок 7 – Принцип действия рекомендательной системы, основанной на знаниях

В таких системах функция подобия оценивает, насколько рекомендации (решение проблемы) соответствуют потребностям пользователя (описание проблемы).

1.4 Выбор оптимального подхода к построению рекомендательных систем в рамках малой электронной коммерции

Для проведения сравнительного анализа сильных и слабых сторон, рассмотренных подходов к построению рекомендательных систем, были выделены основные проблемы, с которыми могут столкнуться рекомендательные системы в области малой электронной коммерции.

1. Холодный старт - проблема, возникающая, когда рекомендательная система на старте работы с пользователем не располагает достаточным количеством информации для составления списка рекомендаций.
2. Разреженность пользовательских данных - для выдачи полезных рекомендаций системе необходимо большое количество пользователей, активно взаимодействующих с товарами. В реальных же системах пользователи не так активны и в основном приобретают небольшое количество товаров, в результате матрица пользовательских элементов может быть чрезвычайно большой и разреженной.
3. Отсутствие разнообразия - проблема обусловлена тем, что генерируемые рекомендации похожи на уже приобретенные или просмотренные товары, тем самым препятствуя знакомству с чем-то новым, в результате чего снижается удовлетворенность качеством рекомендаций.
4. Обобщение – проблема, когда предлагаемые рекомендации больше основаны на общих интересах, чем на интересах пользователя.
5. Плохая адаптивность - порой случается такое, что вкусы пользователя изменились с течением времени, например, он стал вегетарианцем. В таком случае система не может адаптироваться к новым предпочтениям пользователя.

В таблице 1 представлен сравнительный анализ наличия выделенных проблем у рассмотренных подходов к построению рекомендательных систем в рамках предметной области за исключением гибридного подхода. Связано это с тем, что гибридный подход в зависимости от реализации может иметь разные достоинства и недостатки.

Таблица 1 – Сравнение наличия проблем, свойственных малой электронной коммерции, у основных рекомендательных подходов

Подход / Проблема	CBF	CF	DF	KBRS	NPRS
Холодный старт	+	+	-	-	-
Разреженность пользовательских данных	-	+	+	-	-
Отсутствие разнообразия	+	-	+	+	+
Обобщение	-	-	+	-	+
Плохая адаптивность	-	-	+	+	+

На основе проведенного сравнительного анализа в таблице 1 было решено выбрать гибридный подход, сочетающий в себе неперсонализированный подход и фильтрацию на основе содержания, для построения эффективной рекомендательной системы для малых предприятий электронной розничной торговли.

Данный выбор обусловлен тем, что в малой электронной коммерции пользователей может быть значительно меньше, чем ассортимента товара. При этом пользователи могут вести себя не активно в приложении, в следствии чего пользовательский журнал может быть сильно разряжен. Вполне логично в таком случае строить рекомендательную систему отталкиваясь от профилей товаров, а не пользователей, что в свою очередь отсеивает совместную и демографическую фильтрацию. Использование анкет при посещении подобных приложений кажется излишним, так как

избыточный опрос может оттолкнуть пользователя от покупок, так что отсеивается и рекомендательная система, основанная на знаниях. В связи с вышесказанным остаётся фильтрация на основе содержания и неперсонализированный подход. Собственно данные два подхода было решено положить в основу гибридной рекомендательной системы с целью устранения недостатков друг друга в соответствии с таблицей 1.

Идея решения заключается в том, что пользователю, впервые зашедшему в приложение, изначально, например, советуются наиболее популярные товары. Однако при расширении поведенческого профиля пользователя в работу вступает фильтрация на основе содержания, которая подбирает похожие товары на те, что заинтересовали пользователя ранее.

Помимо прочего на выбор гибридного подхода в такой вариации повлияли положительные результаты исследований M. Kaminskas и его коллег в статье «Product recommendation for small-scale retailers».

После выбора оптимального подхода к построению рекомендательной системы в области малой электронной коммерции была сформулирована постановка задачи.

1.5 Постановка задачи прогнозирования пользовательского интереса

Задачу прогнозирования пользовательского интереса, которая заключается в расчёте коэффициента сходства товаров с теми, что ранее заинтересовали пользователя, можно сформулировать следующим образом.

Пусть задано множество пользователей $U = \{u_1, \dots, u_{|U|}\}$ и множество товаров $I = \{i_1, \dots, i_{|I|}\}$, с которыми могут взаимодействовать пользователи.

Необходимо найти N наиболее похожих товаров на те, что ранее заинтересовали пользователя посредством вычисления сходства между товарами с помощью некого метода $\rho(i, i')$. Результатом работы метода

является сформированный для пользователя u список персональных рекомендаций товаров $r = \{i_j\}_{j=1}^{N}$, ранжированных по релевантности.

Целью является описание и реализация метода, который бы максимально близко оценивал сходство между товарами на основе их содержания.

Выводы по первой главе

В данной главе была обоснована актуальность рекомендательных систем для малых предприятий электронной коммерции, а также проведен обзор ряда существующих исследований в рамках предметной области.

Ключевой частью главы является рассмотрение и изучение подходов к построению рекомендательных систем. В ходе работы были рассмотрены такие подходы, как фильтрация на основе содержания, совместная фильтрация, демографическая фильтрация, рекомендательная система на основе знаний, гибридный и неперсонализированный подход.

В результате исследования был проведен сравнительный анализ подходов к построению рекомендательных систем в области малой электронной коммерции, на основе которого был выбран гибридный подход, сочетающий в себе неперсонализированный подход и фильтрацию на основе содержания.

Глава 2 Описание принципа работы, предлагаемой гибридной рекомендательной системы

2.1 Краткое описание принципа работы гибридной рекомендательной системы в области малой электронной коммерции

Предлагаемый гибридный подход к построению рекомендательной системы в области малой электронной коммерции, выбранный в первой главе диссертации, сочетает в себе неперсонализированный подход и фильтрацию на основе содержания.

Неперсонализированный подход используется для решения проблемы, когда на старте работы с пользователем система не располагает достаточным количеством информации о пользовательских интересах для составления списка рекомендаций.

Как только пользователь останавливает своё внимание на минимальном количестве товаров, в работу вступает фильтрация на основе содержания. Для переключения подхода формирования списка пользовательских рекомендаций с неперсонализированного подход на персонализированный, пользователю необходимо осуществить взаимодействие хотя бы с одним товаром.

Фильтрации на основе содержания была выбрана по причине того, что в приложениях малой электронной коммерции пользователи редко возвращаются и оставляют свои оценки товарам, в результате чего профиль пользователя не обладает достаточным количеством информации. Исходя из озвученных условий логично формировать список рекомендаций на основе профилей товаров, вызвавших у пользователя интерес в прошлом, а не на основе профиля пользователя.

На рисунке 8 представлена блок-схема предлагаемого гибридного подхода к построению рекомендательной системы в области малой электронной коммерции.

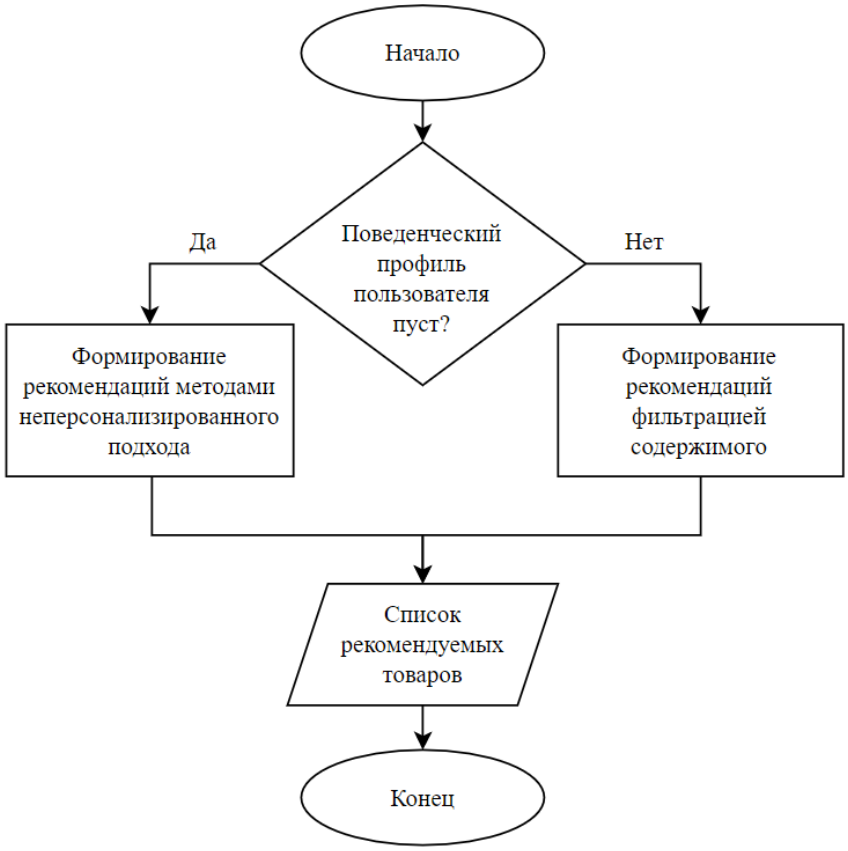


Рисунок 8 – Предлагаемый гибридный подход к построению рекомендательной системы в области малой электронной коммерции

Рассмотрим реализацию каждой части рекомендательной системы по отдельности.

2.2 Неперсонализированная рекомендательная система

Рекомендательные системы, построенные на неперсонализированном подходе, преимущественно рекомендуют всем пользователям одно и тоже, в следствие чего система продвигает только определенные товары, а пользователь получает не всегда интересующие его рекомендации [17]. Тем не

менее преимуществом является минимальное взаимодействие пользователя с системой, что позволяет последней обслуживать новых пользователей сразу при их первом посещении приложения, не дожидаясь от тех каких-либо действий.

Неперсонализированный подход может быть реализован многими методами, среди которых было выделено два основных:

1. Метод статистических данных.

Формирование списка рекомендаций на основе статистических данных работает по принципу: «если что-то нравится всем, то и вам это тоже понравится». При отборе товаров для их рекомендации могут учитываться совокупные явные и неявные предпочтения пользователей к товарам. Например, может использоваться средний оценка товара среди пользователей.

2. Метод продвижения товаров по признаку.

При продвижении товаров по признаку список рекомендаций может состоять из «новинок» или из тех товаров, на которые ритейлер в ручном режиме предлагает обратить внимание пользователям.

Оба рассмотренных выше метода реализации неперсонализированный рекомендательной системы было решено включить в предлагаемую рекомендательную систему по причине того, что каждый из них способен успешно решить проблему холодного старта на начальном этапе работы пользователя внутри приложения. В реализации метода статистических данных используется средняя оценка товаров, у метода продвижения товаров по признаку – дата добавления товара в приложение и популярность товара среди пользователей.

В разрабатываемой рекомендательной системе планируется учесть возможность выбора метода неперсонализированной рекомендательной системы, исходя из пожеланий ритейлера.

2.3 Фильтрация на основе содержания

Рекомендательные системы, использующие фильтрацию на основе содержания, предполагают, что товары похожие на те, что ранее заинтересовали пользователя также смогут вызвать у него интерес [14].

К содержанию товара, из которого можно извлечь характеризующие атрибуты, относятся вся та информация, которую обычно можно найти на странице товара в приложении:

- описание;
- характеристики;
- изображение;
- прочее.

Фильтрация на основе содержания сопоставляет содержание всех товаров с содержанием последнего товара, что заинтересовал пользователя ранее. Результатом такого сравнения является коэффициент, отображающий степень сходства между товарами или, иначе говоря, коэффициент потенциальной заинтересованности пользователем товаром.

Для возможности функционирования фильтрации на основе содержания необходимо реализовать следующие компоненты [24]:

1. Поведенческий профиль пользователя – ряд таблиц в базе данных, хранящих поведенческую активность пользователя с товарами.

2. Анализатор содержания – преобразовывает характеризующую информацию о товаре к удобному для восприятия машины виду, например, к вектору. После чего рассчитывает схожесть между товарами на основе полученного вектора.

На рисунке 9 представлена схема работы фильтрации на основе содержания.

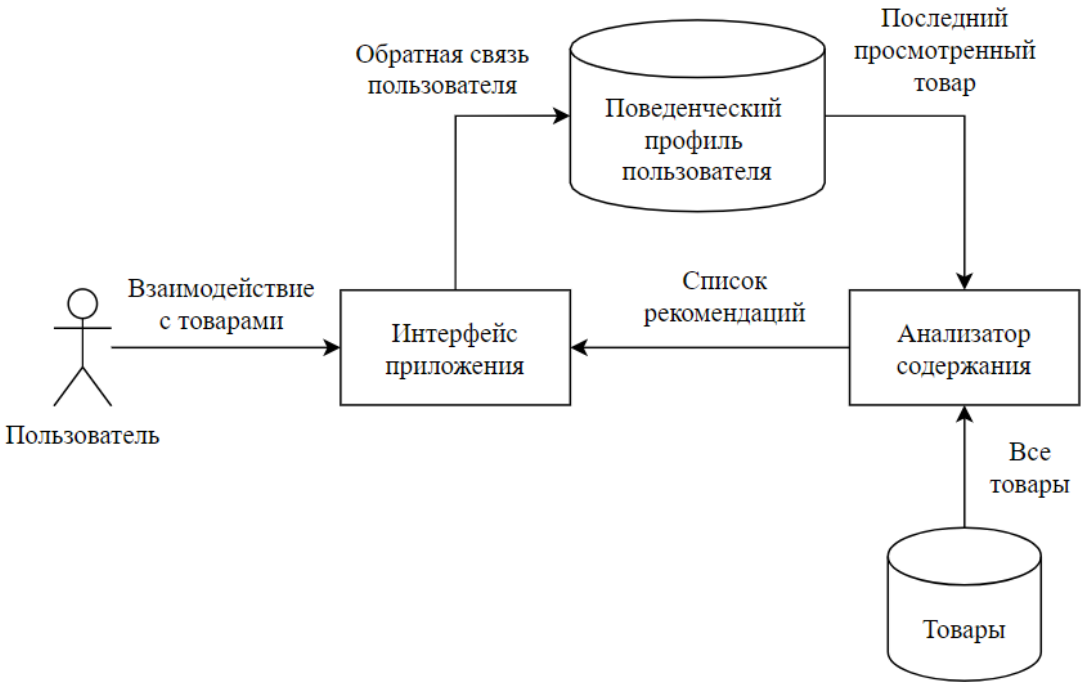


Рисунок 9 – Схема работы фильтрации на основе содержания

В предлагаемой рекомендательной системе планируется реализовать работу с описанием, характеристиками и изображением товаров (рисунок 10).

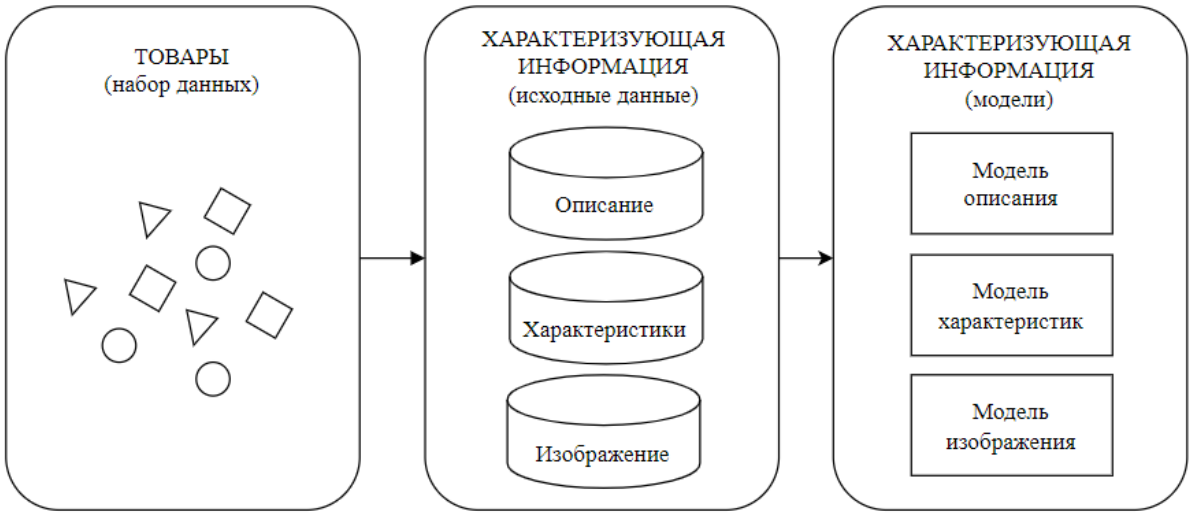


Рисунок 10 – Виды характеризующей информации, с которой работает предлагаемая реализация фильтрации на основе содержания

Основной упор в данной реализации фильтрации на основе содержания планируется делать на описание и характеристики товара. Однако в случае, если ритейлеру необходимо при формировании рекомендаций учитывать цвета на изображении товара, то он сможет включить использование данной характеризующей информации в ручном режиме.

Коэффициентом схожести товаров является сумма значений сходства для каждой составляющей содержания товара, деленная на количество используемых составляющих содержания товара для сравнения. Также на основании значения данного коэффициента будет происходить ранжирование товаров в списке рекомендаций.

Рассмотрим реализацию каждого компонента фильтрации на основе содержания по отдельности.

2.3.1 Поведенческий профиль пользователя

Данный компонент рекомендательной системы фиксирует различные пользовательские активности с товарами для формирования поведенческого профиля пользователя.

Поведенческий профиль пользователя является фундаментальным компонентом каждой рекомендательной системы, так как рекомендательная система не сможет формировать пользовательские рекомендации без знания к каким товарам пользователь проявлял интерес ранее и каким образом [19].

Реализация данного компонента происходит посредством фиксирования на уровне кода действий пользователя с товарами и сохранения этой информации в специально созданные для этого таблицы в базе данных.

Поведенческий профиль пользователя работает с двумя типами входных данных:

– явная обратная связь – представляет собой активности, при которых пользователь в явной форме высказывает свой интерес к товару, например, это может быть оценка или добавление товара в «мне нравится/избранное»;

– неявная обратную связь – определяется путем косвенного высказывания предпочтений пользователя посредством наблюдения за его поведением, например, это может быть просмотр страницы товара или добавление его в корзину.

В предлагаемом решении планируется работать с обоими типами обратной связи, высказанной как явным, так и неявным образом:

– добавление в избранное (явная обратная связь);
– оценка товару (явная обратная связь);
– отзыв товару (неявная обратная связь);
– просмотр страницы товара (неявная обратная связь);
– добавление в корзину (неявная обратная связь).

Хотя явная обратная связь по своей сути является более точно отображающей заинтересованность пользователя товаром, тем не менее ни каждый пользователь проставляет в приложениях малой электронной коммерции проставляет оценки понравившимся товарам или добавляет товар в избранное. Исходя чего логично ориентироваться также и на неявную обратную связь пользователя. Однако те же оценки товаров можно использовать при формировании рекомендаций неперсонализированной рекомендательной системой по среднему рейтингу товара среди пользователей.

2.3.2 Создание модели описания товара

Главными атрибутами моделей, представляющих текста, являются ключевые слова, которые характеризуют содержимое описания. Благодаря сравнению ключевых слов двух разных описаний товаров можно решить задачу их сходства между собой. Одной из таких моделей, основанной на ключевых словах является векторно-пространственная модель (Vector Space Model (VSM)).

Пусть имеется набор описаний $D = (d_1, d_2, \dots, d_{|D|})$ и словарь, сформированный из уникальных слов этого набора, $T = (t_1, t_2, \dots, t_{|T|})$.

Словарь T формируется путем применения ряда операций обработки естественного языка, таких как токенизация, удаление стоп-слов, стемминг и т.п. [27]. Каждое описание из набора в свою очередь представлено в виде n −мерного вектора, пространствами которого являются весовые коэффициенты, как показана на формуле (1).

$$d_j = (w_{ji}, w_{j2}, \dots, w_{jn}) \quad (1)$$

где d_j – это векторное представление $j-$ го описания;

w_{ij} – вес $i-$ го слова в $j-$ ом описании;

n – общее количество различных слов во всех описаниях рассматриваемого набора.

Каждый такой весовой коэффициент указывает на степень связи между описанием товара и словом из словаря рассматриваемого набора описаний.

Располагая таким представлением описаний, можно решить задачу их подобия, просто найдя расстояние между точками пространства. При этом чем ближе расположены точки, тем больше похожи соответствующие описания.

Прежде чем создавать векторную модель описания товара, первонаперво необходимо провести его предварительную обработку (рисунок 11).

Предварительная обработка один из наиболее важных шагов в подготовке текстов перед тем, как проводить с ними какие-либо операции. Качественная обработка способна не только уменьшить вычислительную сложность, но и улучшить результаты рекомендательной системы.

Область, занимающаяся обработкой текстов, написанных на естественных языках, с помощью информационных технологий, называется Обработкой Естественного Языка или Natural Language Processing (NLP) [7].

Рисунок 11 – Блок-схема предварительной обработки описания

Для осуществления предобработки описания было решено выполнять следующие шаги:

1. Приведение слов к нижнему регистру. Для повышения качества предобработки текста в первую очередь все слова приводятся к нижнему регистру. Делается это с целью избегания неверного толкования одинаковых слов, представленных символами в разных регистрах.

2. Токенизация. При работе с текстом в первую очередь необходимо сначала разбить предложения на массив отдельных слов (токены), отфильтровав при этом символы, не несущие смысловой нагрузки, такие как цифры, операторы, разделители, знаки пунктуации [2].

3. Исключение стоп-слов. Во всех текстах содержатся большое количество слов, предназначенных для того, чтобы сделать текст понятным читателю, это могут быть предлоги, причастия, междометия или частицы [5]. Однако, если речь заходит о преобразовании текста в массив слов, то, например, слово «что» не несет никакой информации, отображающей содержание описания, и в таком случае его можно спокойно удалить. Таким

образом, стоп-слова – это слова, не несущие смысловой нагрузки, иначе говоря, шум.

4. Нормализация описания. При обработке естественного языка может возникнуть ситуация, когда в тексте содержатся однокоренные слова или разные грамматические формы одного и того же слова. Например, слова «музыкальный» и «музыкальная» несут один и тот же смысл, однако машиной будут восприняты как абсолютно разные, что в контексте извлечения ключевых слов является неудовлетворительным результатом. Для разрешения подобной ситуации используется нормализация текста, которая проводит слова к нормальной словарной форме [20].

Для нормализации текста используются два подхода – стемминг и лемматизация.

Стемминг – это подход, основанный на правилах. Данный подход заключается в поиске лингвистической формы слова (корня) посредством обрезания его начала и конца на основе заданных правил. Правила обрезания задаются заранее и чаще всего представляют из себя набор регулярных выражений. Существенным недостатком такого подхода является возможная потеря информации при отрезании частей слова.

Лемматизация – это словарный подход. Сутью подхода является морфологический анализ слов, путём использования словарей, в которые машина может подсмотреть, чтобы сравнить текущую форму слова с ее словарной формой (леммой).

Если сравнивать принцип работы этих двух подходов, то можно заявить, что разработка стеммера намного проще, чем создание лемматизатора, так как в последнем случае необходимы глубокие лингвистические знания для создания словарей, позволяющих подбирать правильную форму слова [10].

По причине отсутствия человека с глубокими лингвистические знаниями, который бы помог в формировании словаря, который бы лег в основу лемматизатора, было решено использовать стемминг для нормализации описания товара.

Классической реализацией данного процесса является стеммер Портера. Алгоритм не использует баз основ слов, а работает, последовательно применяя ряд правил отсечения окончаний и суффиксов [33]. Блок-схема алгоритма для русского языка представлена в приложении А.

После выполнения предварительной обработки описания происходит построение его частотной модели с помощью статистической меры TF–IDF.

TF-IDF – статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов. Вес каждого слова пропорционален количеству употребления этого слова в документе, и обратно пропорционален частоте употребления слова в других документах коллекции.

К очевидным плюсам метрики можно отнести:

– метрика проста в вычислениях;

– скорость вычисления.

TF-IDF уже давно интегрирован в формулу ранжирования Яндекс и Google, а теперь и лёг в основу более продвинутых алгоритмов вроде векторного анализа текста [30].

Показатель TF рассчитывается как отношение числа вхождения некоторого слова к общему количеству слов текста. Благодаря TF точность определения значимости слова не зависит от длины текста. На одном и том же количестве вхождений слова длинный текст будет иметь показатель TF меньше, чем у короткого текста.

Показатель IDF – это инверсия частоты, с которой определенное слово фигурирует в коллекции текстов. Учет IDF позволяет снизить вес слов, употребляемых часто. Для каждого уникального слова в пределах конкретной коллекции текстов существует только одно значение IDF.

Вычисления веса рассматриваемого слова мерой TF-IDF в контексте документа осуществляется посредством перемножения показателей TF и IDF.

После вычисления весов слов описания выполняется нормализация, для того чтобы все веса попадали в интервал $[0; 1]$.

Для определения сходства между двумя описаниями товаров на основе полученных весов слов применяются различные меры сходства. Одной из самых популярных мер в текстовом анализе, использующейся для измерения схожести между двумя текстами, является косинусная мера [25].

На рисунке 12 представлена блок-схема вычисления сходства между описаниями товаров, сформированная на основе вышеописанных действий.

Рисунок 12 – Блок-схема вычисления сходства товаров на основе их описаний

Таким образом, в предлагаемой рекомендательной системе происходит расчет сходства между описаниями товаров.

2.3.3 Создание модели характеристик товара

Характеристики товара планируется принимать на входе анализатора содержания в виде вектора размерности n, измерения которого содержат ассоциативные массивы с наименованием характеристики в виде ключа и вектором значений этой характеристики размерности m в виде значения. В данном случае, n – это общее количество характеристик у товара, а m – это количество значений, которое может принимать конкретная характеристика.

На рисунке 13 представлено графическое изображение представления характеристик.

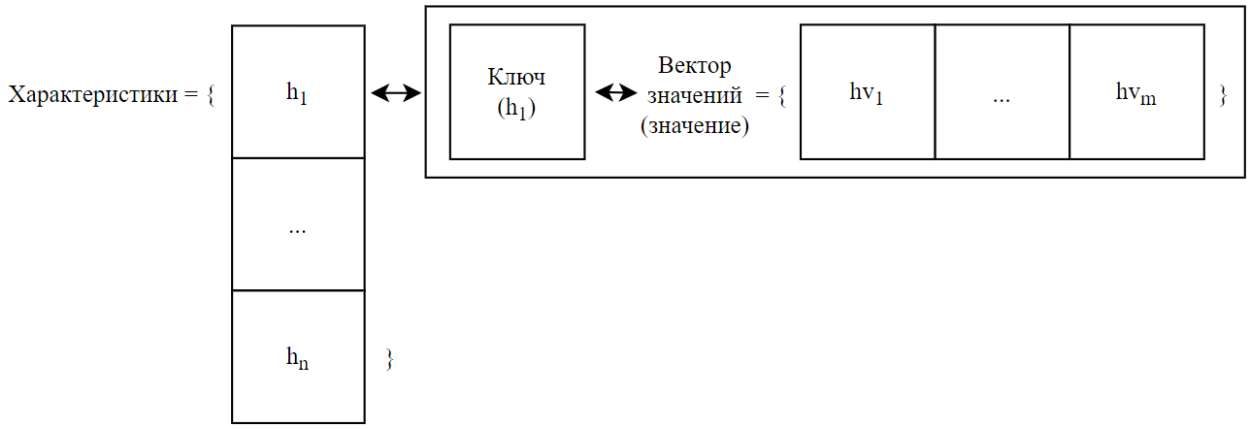


Рисунок 13 – Представление характеристик

Характеристики товаров в онлайн-магазинах зачастую уже представлены в одной и той же форме. Поэтому значение рассматриваемой характеристики можно определять 0 или 1, в зависимости от того присутствует ли рассматриваемое значение у того товара, что заинтересовал пользователя ранее. Значения характеристик предварительно подвергаются обработки аналогичной, что у содержания.

Посредством суммирования всех значений измерений, полученного вектора в ходе преобразований выше, и разделив вычисленное число на размерность вектора, получаем коэффициент схожести товаров в рамках i характеристики (формула (2)).

$$h_i = \frac{\sum_{j=0}^{m-1} hv_{ij}}{m} \quad (2)$$

где hv_{ij} – это $j-$ое значение $i-$ой характеристики товара;

m — это количество значений, которое может принимать i характеристика.

Проделав данную операцию для каждой характеристики, имеем вектор коэффициентов h, которые суммируем и делим на общее количество характеристик у товара (формула (3)).

$$w = \frac{\sum_{i=0}^{n-1} h_i}{n} \quad (3)$$

где h_i – это коэффициент схожести товаров в рамках i характеристики;

n – это общее количество характеристик у товара.

Результатом вычислений является коэффициент схожести характеристик рассматриваемого товара с тем, что ранее заинтересовал пользователя.

2.3.4 Создание модели изображения товара

При создании списка рекомендаций на основе текстовой информации может возникнуть ситуация, когда ряд товаров, способных заинтересовать пользователя, обладают достаточно близкими показателями схожести.

Например, если пользователь ищет в онлайн-магазине обувь интересующего цвета, то в этом случае имеет смысл предложить ему товар аналогичной расцветки [16]. В результате выше сказанного возникает сложность с определением релевантности товаров. Для разрешения таких спорных ситуаций предлагается использовать сравнение изображений.

С точки зрения визуальной психологии, когда человек видит некий образ впервые, то первое на что он обращает внимание – это преобладающие цвета, пренебрегая второстепенными [1]. Поэтому небольшое количество

преобладающих цветов зачастую вполне достаточно для характеристики цветовой информации изображения.

Для извлечения преобладающих цветов из изображения товара было решено воспользоваться алгоритмом кластеризации k-means, так как среднее значение всех цветов на изображении оказалось менее эффективным в решении данной задачи [22].

Алгоритм k-means – это алгоритм кластеризации, который определяет и помещает в один из k кластеров входные данные на основании общих признаков. Количество кластеров k при этом указывается заранее.

В качестве входных данных k-means предлагается использовать пиксели изображения. Пиксель – это минимальная единица изображения, которая содержит информацию о цвете. Сам по себе пиксель является одноцветным, однако этот цвет обычно представлен тремя или четырьмя компонентами в зависимости от цветовой модели.

Для дальнейшей работы с цветами изображения было решено выбрать цветовую модель HSB, названную так по трем компонентам, лежащих в ее основе: Hue (цветовой тон), Saturation (насыщенность), Brightness (яркость). В памяти компьютера каждая составляющая характеризуется следующими числами [13]:

– цветовой тон – варьируется в пределах $0–360°$, однако иногда приводится к диапазону 0–100 или 0–1;

– насыщенность – варьируется в пределах 0–100% или 0–1;

– яркость – варьируется в пределах 0–100% или 0–1.

Ключевым доводом, склонившим к выбору данной цветовой модели, является то, что цветовая модель HSB ближе человеческому восприятию цветов, чем тот же RGB, который определяют цвет как комбинацию основных цветов красного, зелёного, в то время как компоненты цвета в HSV отображают информацию о цвете в более привычной человеку форме: что это за цвет? насколько он насыщенный? насколько он яркий или тёмный? В

контексте ситуации, где нам важно ориентироваться на цветовые предпочтения клиента данная цветовая модель выглядит более подходящей.

В свою очередь, представление цвета тремя компонентами в HSB системе позволяет обрабатывать пиксели изображения как точки в трехмерном пространстве.

Принцип работы алгоритма k-means в данных условиях выглядит следующим образом [4]:

1. Задается количество k кластеров.
2. Из исходного множества данных случайным образом выбираются k наблюдений, которые будут служить начальными центрами кластеров.
3. Для каждого пикселя исходного множества определяется ближайший центр кластера с помощью расстояния, измеренного, например, Евклидовой метрикой [8]. Расстояние Евклида между двумя пикселями, представленных в виде 3-мерного пространства, рассчитывается по формуле (4).

$$sim(x, y) = \sqrt{\left(H_x - H_y\right)^2 + \left(S_x - S_y\right)^2 + \left(B_x - B_y\right)^2} \quad (4)$$

где x, y – сравниваемые пиксели изображения в цветовой модели HSB;

H – значение цветового тона пикселя;

S – значение насыщенности пикселя;

B – значение яркости пикселя.

4. После того, как все пиксели отнесены к кластерам происходит перевычисления центроида для каждого кластера, посредством расчета среднего значения всех пикселей, отнесенных к рассматриваемому кластеру.
5. Алгоритм итеративно повторяется с 3 по 4 шаг, до тех пор, пока пиксели не останутся в одних и тех же кластерах. Зачастую такого может и не случиться, так как при большом количестве данных центры будут перемещаться в малом радиусе, и пиксели по краям кластеров будут прыгать

из одного кластера в другой. Для избежание подобной ситуации задается максимальное количество итераций.

Последние вычисленные центры k кластеров и являются преобладающими цветами изображения.

Результатом выполнения вышеописанных действий является вектор, содержащий в себе набор преобладающих цветов изображения товара в цветовой модели HSB.

Для расчета сходства между преобладающими цветами на изображениях товаров, представленных в виде числового вектора, предлагается использовать евклидово расстояние, вычисляемое по формуле (4).

Выводы по второй главе

В данной главе был сформулирован принцип работы гибридной рекомендательной системы посредством рассмотрения способов реализации её отдельных частей, неперсонализированного подхода и фильтрации на основе содержания.

Для неперсонализированного подхода были выбраны следующие методы реализации – метод статистических данных и метод продвижения товаров по признаку.

Для фильтрации на основе содержания был описан принцип вычисления коэффициентов сходства между двумя товарами на основании описания, характеристик и преобладающих цветов на изображении товара. Текстовая составляющая содержания товаров (описание и характеристики) используется по умолчанию при формировании списка рекомендаций, тем не менее, если ритейлер считает, что при создании рекомендаций также необходимо учитывать цвета изображения товара, то в планируемой реализации он сможет активировать использование данной характеризующей информации.

Глава 3 Практическая реализация гибридной рекомендательной системы

3.1 Стек технологий, использованный при реализации предлагаемой рекомендательной системы

Для реализации предложенной рекомендательной системы использовался объектно-ориентированный язык программирования Java. Выбор данного языка программирования был обусловлен тем, что многие web и android приложения электронной коммерции написаны именно на данном языке программирования.

В ходе начала работы над практической реализаций рекомендательной системы было решено сразу ориентироваться на её потенциальную возможность интеграции с приложениями, написанными на базе фреймворка Spring.

На основе выбранных решений был сформирован следующий стек технологий:

– Maven – инструмент для управления и сборки проекта;

– JUnit – фреймворк автоматического тестирования написанного кода;

– Spring Data JPA – библиотека, предоставляющая возможность для взаимодействия с сущностями базы данных;

– Spring Boot – проект, упрощающий процесс создания web-приложений на основе Spring, требуя от разработчиков минимум усилий по настройке и написанию кода;

– Slf4J – библиотека для логирования, предоставляющая мощный фасад для различных систем логирования на Java;

– Lombok – библиотека по добавлению дополнительной функциональности в Java с помощью изменения исходного кода перед Java компиляцией.

Программный код писался в интегрированной, единой среде разработки, IntelliJ IDEA. Для работы с реляционными базами данных использовался DBeaver, свободный кроссплатформенный менеджер баз данных.

В качестве хостинга для исходного кода проекта и хранилища информации обо всех изменениях этого самого кода был использован GitHub, сервис на базе системы контроля версий Git.

С помощью вышеназванного набора инструментов, включающего в себя языки программирования, фреймворки и библиотеки была осуществлена реализация предлагаемой рекомендательной системы в области малой электронной коммерции.

3.2 Классы «UserSupercalss» и «ProductSuperclass»

В первую очередь при разработке гибридной рекомендательной системы были созданы два ключевых класса «UserSupercalss» и «ProductSuperclass».

Каждый из озвученных классов являются классом-сущностью, то есть связан с некой таблицей в базе данных. Осуществляется эта связь благодаря спецификации JPA, которая реализовывает технологию объектно-реляционного отображения (ORM).

Класс «UserSupercalss» описывает все необходимые поля пользователя для функционирования рекомендательной системы. В данном случае атрибутом сущности пользователь выступает всего одно поле и это идентификационный номер пользователя (id).

Класс «ProductSuperclass» описывает все необходимые поля товара для функционирования рекомендательной системы. Необходимыми атрибутами класса «ProductSuperclass» являются:

- идентификационный номер (id);
- наименование (name);
- описание (description);
- изображение (image);

– преобладающие цвета на изображении товара (imageColors);
– характеристики (features);
– средний рейтинг (averageRating);
– дата добавления (addedDate).

Используя Spring Data JPA, был написан репозиторий для взаимодействия с сущностью товара, который помимо стандартных CRUD операций позволяет получать товары по среднему рейтингу и по дате добавления, что в свою очередь реализует методы неперсонализированной рекомендательной системы – метод статистических данных и метод продвижения товаров по признаку.

Также был создан класс-сервис для работы всё с той же сущность.

3.3 Поведенческий профиль пользователя

Фильтрация на основе содержания рекомендует пользователю различные товары, на которые стоит обратить внимание, используя для этого его поведенческий профиль.

Для пользовательских событий был создан абстрактный класс «Event». Данный класс содержит общие поля, свойственные всем активностям пользователя:

– идентификационный номер события в таблице (id);
– пользователь, совершивший действие (user);
– товар, с которым взаимодействовал пользователь (product);
– дата события (date).

Класс «Event» наследуют пять классов-сущностей, каждый из которых ответственен за соответствующую активность пользователя:

– «Favorite» – добавление товара в избранное;
– «View» – просмотр пользователем товара;
– «Cart» – добавление товара в корзину;
– «Rating» – пользователь поставил оценку товару;

– «Review» – пользователь оставил отзыв товару.

Выше названные пользовательские действия были выбраны ввиду того, что чаще всего могут встречаться в малой электронной коммерции [26].

На приложении Б представлена UML-диаграмма классов, которая демонстрирует атрибуты и взаимосвязи между классами, описывающих поведенческую активность пользователя, а также между классами-сущностями «UserSupercalss» и «ProductSuperclass». Методы классов на рассматриваемой UML-диаграмме классов было решено не отображать, так как все они являются геттерами и сеттерами представленных атрибутов.

Для каждого класса, описывающего пользовательскую активность, были реализованы репозитории с использованием Spring Data JPA, а также соответствующие классы-сервисы, которые выполняют основную бизнес-логику по работе с данными классами. Методы разработанных сервисов необходимо будет использовать ритейлеру при внедрении гибридной рекомендательной системы в своё web-приложение для обработки действий пользователя, чтобы предлагаемое решение могло успешно функционировать.

Для фиксирования каждого пользовательской активности был создан класс-сущность «UserEvent», который по факту собой представляет поведенческий профиль пользователя.

UML диаграмма-класса «UserEvent», отображающая его атрибуты и связи с другими классами-сущностями, изображена на приложении В. Методы классов также было решено не отображать на UML-диаграмме классов, так как все они геттеры и сеттеры представленных атрибутов.

Для класса-сущности «UserEvent» также был реализован репозиторий с использованием Spring Data JPA и класс-сервис, с помощью которого можно получить список наиболее популярных товаров среди пользователей, последний просмотренный пользователем товар, все пользовательские события с товаром или все поведенческие активности пользователя.

При фиксировании пользовательского действия с товаром посредством использования соответствующего сервиса, в таблицу «user_events» также

автоматически сохраняется факт пользовательского события. Иначе говоря, формируется поведенческий профиль пользователя.

В приложении В на UML-диаграмме классов фигурирует ранее не озвученный класс-перечисление «EventWithProduct». «EventWithProduct» содержит список констант действий пользователя.

3.4 Анализатор содержания описаний

В первую очередь при работе с описанием товара необходимо предварительно его обработать.

Первым делом выполняется токенизация описания, посредством приведения слов к нижнему регистру и разбиению предложения на массив отдельных слов с фильтрацией лишних символов (цифры, операторов и т.д.).

Для осуществления вышеупомянутых действий был написан класс «Tokenizer» со специальным методом «tokenize».

После токенизации из описания исключаются стоп-слова с помощью метода «stopWords» класса «StopWords». Список стоп-слов был взят у команды «CountWordsFree», занимающейся составлением стоп-слов для разных языков мира и выкладывающих свои результаты исследований в свободный доступ [21].

Как только исключаются все стоп-слова из описания, выполняется нормализация текста. Для этого используется класс «Normalizer», который содержит реализацию стеммера Портера.

UML-диаграмма классов, участвующих в предварительной обработке описания товара, представлена на рисунке 14.

Следующим шагом после выполнения предварительной обработки описания, является построение его частотной модели.

Для осуществления расчета статистической меры TF-IDF были написаны три класса «TF», «Idf» и «TfIdf».

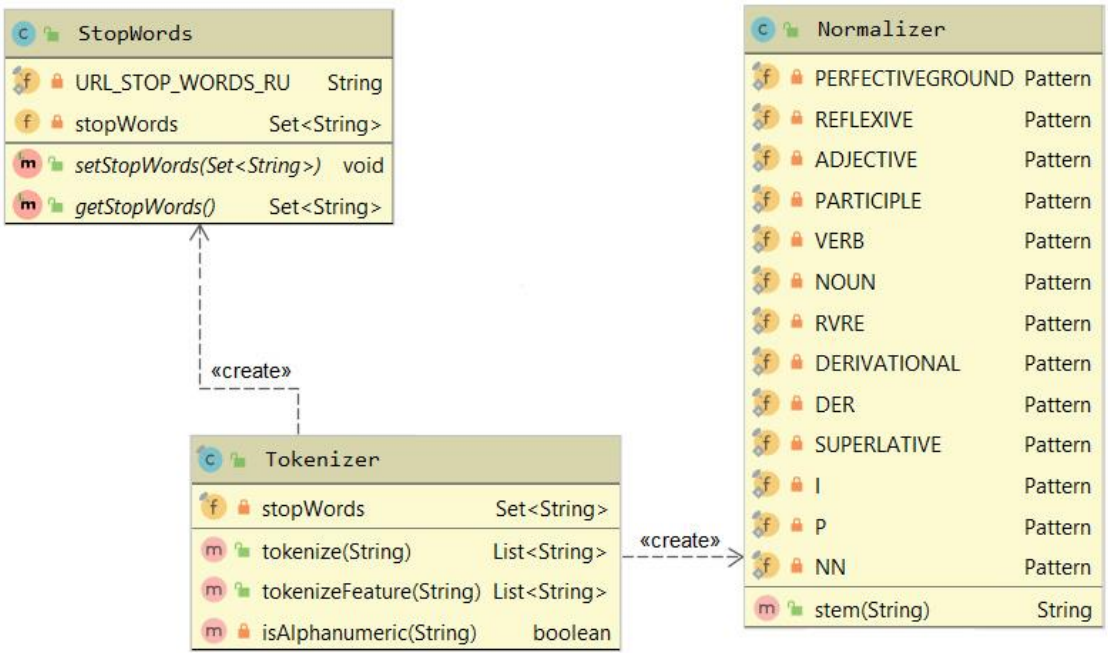


Рисунок 14 – UML-диаграмма классов, участвующих в предварительной обработке описания товара

На рисунке 15 представлена UML-диаграмма классов, реализующих статистическую меру TF-IDF.

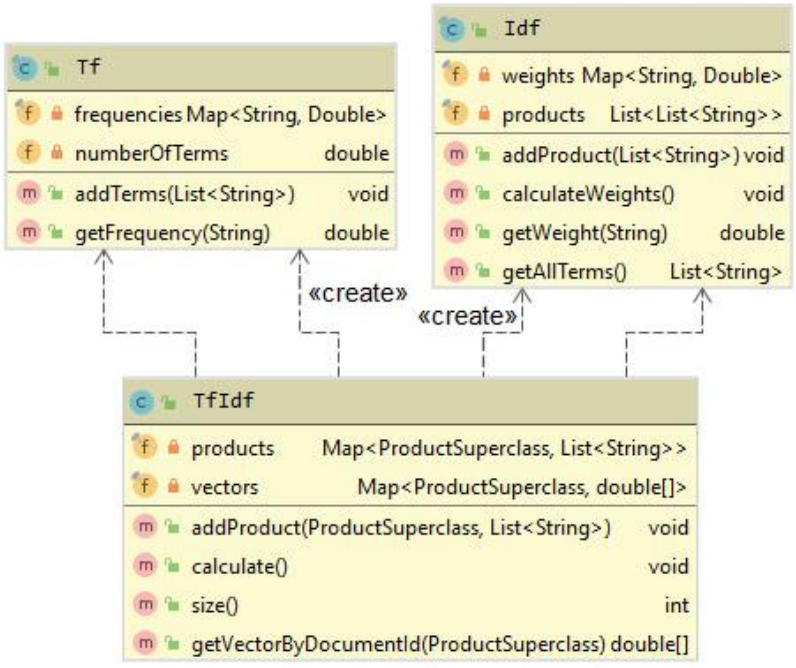


Рисунок 15 – UML-диаграмма классов, реализующих меру TF-IDF

Метод «calculate» класса «TfIdf» рассчитывает веса слов описания товара, отражающих важность того или иного слова в контексте этого самого описания.

Для вычисления сходства описания товаров был реализован класс «SimilarityDescription» с ключевым методом «getSimilarProducts». Целью данного метода является сравнение описания последнего заинтересовавшего пользователя товара с каждым описанием товара из базы данных, используя функционал вышеупомянутых классов. Результатом работы метода является ассоциативный массив, где ключом выступает – идентификационный номер товара, а значением – коэффициент сходства описания рассматриваемого товара с тем товаром, что ранее заинтересовал пользователя.

3.5 Анализатор содержания характеристик

Обработка характеристик товара почти идентична обработке описания за единственным исключением, что в классе «Tokenizer» вызывается метод «tokenizeFeture», так как токенизация характеристик слегка отличается от токенизации описания. Например, нет нужды удалять слова маленькой длины или цифры, так как в характеристиках это может быть валидным значением.

Также как и для расчета сходства описания товаров, для вычисления сходства характеристик был создан класс «SimilarityFeatures» с ключевым методом «getSimilarProducts». Итогом работы данного метода является ассоциативный массив, где ключом выступает – идентификационный номер товара, а значением – коэффициент сходства характеристик рассматриваемого товара с тем товаром, что ранее заинтересовал пользователя.

3.6 Анализатор содержания изображений

Во второй главе для извлечения преобладающих цветов на изображении товара было решено воспользоваться алгоритмом кластеризации k-means.

Стоит напомнить, что в пункте 2.3.4 говорилось о том, что для работы с цветами изображения было решено выбрать цветовую модель HSB, так как она ближе человеческому восприятию цветов. Исходя из этого был создан класс «HSBPixel», который принимает на вход RGB пиксели и преобразовывает их в конструкторе с параметрами в HSB пиксели.

Для преобразования пикселей изображения из RGB модели в HSB использовался алгоритм, представленный в приложении Г.

Реализации алгоритма k-means была осуществлена в рамках метода «getDominantColors» класса «DominantColor».

В приложении Д представлена блок-схема алгоритма кластеризации k-means, с помощью которого вычисляются преобладающие цвета на изображении товара.

Прежде, чем изображение передается k-means оно предварительно сжимается до размера $64x64$. Данная операция служит для уменьшения количества обрабатываемых пикселей, что существенно сокращает время, необходимое для извлечения преобладающих цветов изображения, при этом данная операция не сильно влияет на итоговую картину.

Итогом выполнения метода «getDominantColors» является список преобладающих цветов изображения.

В реализованной гибридной рекомендательной системе присутствует возможность задать искомое количество преобладающих цветов на изображении товара с помощью параметра «numColors» (по умолчанию значение равняется трём).

Для работы с изображением был написан специальный утилитный класс «ImageUtils», с помощью которого можно получить изображение по URL, изменить размер изображения, а также ряд других возможностей.

Для расчёта сходства преобладающих цветов на изображении товаров был реализован класс «SimilarityImage» с методом «getSimilarProducts». Итогом работы данного класса является ассоциативный массив, где ключом выступает – идентификационный номер товара, а значением – коэффициент

сходства преобладающих цветов на изображении рассматриваемого товара с тем товаром, что ранее заинтересовал пользователя.

3.7 Реализация меры подобия

Для вычисления сходства товаров на основании коэффициентов сходства описаний, характеристик и преобладающих цветов на изображении товара был написан интерфейс «Distance».

Интерфейс «Distance» реализовывают два класса:

1. «CosineImageDistance» – класс, рассчитывающий сходства между двумя товарами, используя для этого косинусную меру.

2. «EuclideanImageDistance» – класс, рассчитывающий расстояние между двумя товарами, используя для этого евклидово расстояние.

На рисунке 16 представлена UML-диаграмма классов интерфейса «Distance» и его реализаций.

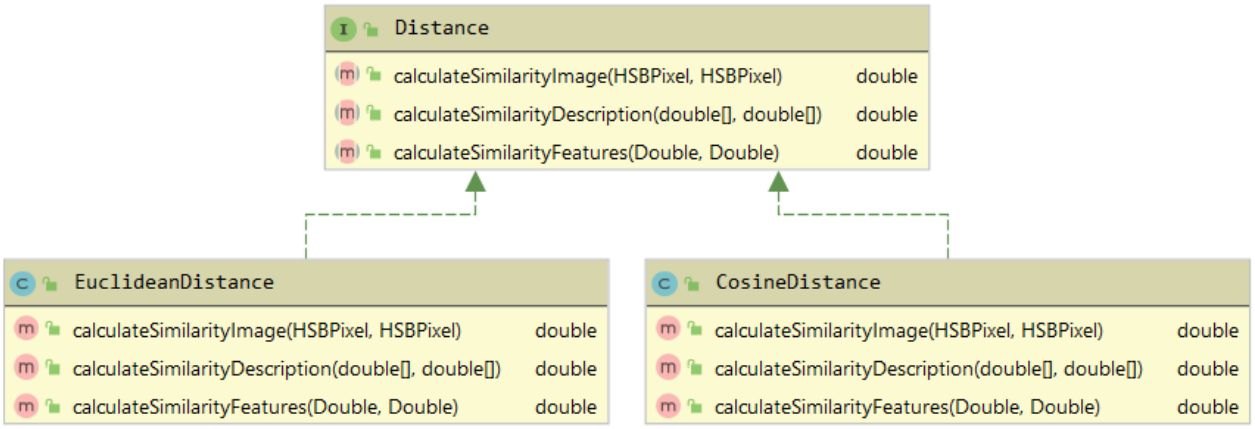


Рисунок 16 – UML-диаграмма классов интерфейса «Distance» и его реализаций

Воспользовавшись методам необходимой реализации интерфейса «Distance», можно рассчитать сходства необходимой составляющей содержания между двумя товарами.

3.8 Формирование пользовательских рекомендаций

Главным классом, к которому можно обратиться, чтобы получить список рекомендаций для конкретного пользователя, является класс «RecommendationSystem».

«RecommendationSystem» содержит три ключевых метода:

1. «getRecommendationsForUser» – публичный метод для получения списка рекомендаций. В случае если поведенческий профиль пользователя пуст, то вызывает метод «getNonPersonalRecommendations», если же пользователь взаимодействовал хотя бы с одним товаром, то вызывается «getPersonalRecommendations». Задав необходимый параметр при вызове данного метода, можно настроить то, какие составляющие содержания товара будут участвовать в поиске товаров похожих на последний товар, заинтересовавший пользователя. Позволяет настроить количество действий, которые должен совершить пользователь, чтобы в работу включились персонализированные рекомендации (фильтрация на основе содержания) или вовсе их отключить, оставив только в эксплуатации неперсонализированную рекомендательную систему. Также предоставляет возможность настроить количество возвращаемых пользователю товаров, которые могут его заинтересовать.

2. «getPersonalRecommendations» – закрытый метод. Осуществляет вычсиление похожих товаров на тот товар, что ранее заинтересовал пользователя, используя для этого фильтрацию на основе содержания.

3. «getNonPersonalRecommendations» – закрытый метод. Реализует неперсонализированные рекомендации. По умолчанию возвращает наиболее популярные товары в системе, посредством просмотра таблицы в базе данных «user_event». Позволяет сменить рекомендации с самых популярных товаров на новинки или на самые высокоценные среди пользователей, задав необходимое значение параметра «typeNonPersonalRecommendations» при вызове метода.

В ходе выполнения метода getRecommendationsForUser() гибридная рекомендательная система возвращает релевантный список товаров, на основе заданных ритейлером параметров.

3.9 Требования к пользовательской стороне рекомендательной системы

Для того чтобы разработанная рекомендательная система работала корректно необходимо, чтобы сторона, которая ей собирается пользоваться, обеспечила ряд требуемых условий, в ином же могут возникать неполадки или же рекомендательная система может в принципе не заработать.

Так как разработанная рекомендательная система реализована на языке Java для web-приложений, построенных на Spring Framework или Spring Boot, то очевидно, что в первую онлайн-приложение ритейлера должно быть написано на языке программирования Java и использовать возможности указанного фреймворка. Также, помимо этого, веб-приложение должно иметь в своем стеке технологий реляционную базу данных PostgreSQL.

Для того, чтобы интегрировать разработанное решение в web-приложение необходимо выполнять ряд следующих действий:

1. Импортировать библиотеку с гибридной рекомендательной системой в своё веб-приложение.

2. Выполнить наследование классов «UserSuperclass» и «ProductSuperclass» классами, которые отвечают за товары и пользователей в приложении, соответственно.

3. Добавить обработку пользовательской активности посредством вызова методов соответствующих сервисов.

4. При добавлении нового товара в базу данных или обновлении его изображения необходимо заполнять поле «imageColors» класса «ProductSuperclass» результатом работы метода «getDominantColors» класса

«DominantColors», с целью сохранения преобладающих цветов на изображении товара.

5. Использовать метод «getRecommendationsForUser» класса «RecommendationSystem», задав необходимые параметры, для получения списка пользовательских рекомендаций товаров.

Решение о том, в какой информационный блок веб-приложения выводить список пользовательских рекомендаций, сформированный разработанной рекомендательной системой, остается за ритейлером.

Выводы по третьей главе

В данной главе было представлено подробное рассмотрение реализации, предлагаемой гибридной рекомендательной системы в области малой электронной коммерции.

Был определён стек технологий, который использовался при реализации рекомендательной системы, а также представлены UML-диаграммы ключевых классов и представлено описание их предназначения.

Также были сформированы требования к пользовательской стороне рекомендательной системы.

Глава 4 Тестирование разработанной рекомендательной системы

4.1 Тестирование качества прогнозирования интереса пользователя рекомендательной системы

Для определения качества прогнозирования рекомендательных систем используются регрессионные и классификационные метрики. Пригодность каждой метрики зависит от характеристик набора данных и типа задач, которые рекомендательная система будет выполнять.

Регрессионные метрики оценивают точность рекомендательной системы путем непосредственного сравнения прогнозируемых оценок с фактическими оценками пользователей.

Классификационные метрики определяют, то насколько правильно рекомендательная система отличает подходящие для пользователя товары от неподходящих, не учитывая при этом какой товар удовлетворит интерес пользователя лучше.

По причине того, что в предлагаемой рекомендательной системе прогнозируется потенциальная заинтересованность пользователя в товаре, а не его оценка ему, то логично в таком случае применять классификационные метрики.

Для определения качества прогнозирования интереса пользователя гибридной рекомендательной системой для области малой электронной коммерции было решено рассмотреть наиболее популярные классификационные метрики:

- доля правильных ответов (accuracy);
- точность (precision);
- полнота (recall);
- F-мера (F-measure).

Для вычисления метрик введём такое понятие, как матрица ошибок, представляющую собой способ разбиения товаров на четыре категории в

зависимости от комбинации истинного ответа и ответа рекомендательной системы (таблица 2).

Таблица 2 – Матрица ошибок

Категория		Фактическая оценка	
		Положительная	Отрицательная
Оценка системы	Положительная	Истинно положительный (TP)	Ложно положительный (FP)
	Отрицательная	Ложно отрицательный (FN)	Истинно отрицательный (TN)

Доля правильных ответов является самой простой оценкой классификации, высчитывающей вероятность того, что класс будет предсказан правильно (5).

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

Точность системы в пределах класса обозначает долю записей, действительно принадлежащих данному классу относительно всех объектов, которые система отнесла к этому классу (6).

$$precision = \frac{TP}{TP+FP} \quad (6)$$

Полнота системы отображает долю найденных классификатором записей, принадлежащих классу относительно всех записей этого класса в тестовой выборке (7).

$$recall = \frac{TP}{TP+FN} \quad (7)$$

Очевидно, что чем выше точность и полнота, тем лучше. Однако в реальной жизни максимальная точность и полнота не достижимы одновременно, что вынуждает искать некий баланс. Именно для это используется F-мера, которая объединяет в себе информацию о точности и полноте рассматриваемого алгоритма, что облегчает принятие решения о том какую реализацию использовать. F-мера вычисляется по формуле 8.

$$\text{Fmeasure} = 2 \times \frac{precision \times recall}{precision + recall} \quad (8)$$

Данная формула придает одинаковый вес точности и полноте, поэтому F-мера будет падать одинаково при уменьшении и точности и полноты.

В качестве входных данных для тестирования разработанной рекомендательной системы было решено воспользоваться общедоступным набором Goodreads Datasets, который содержит информацию о пользователях и их книжных предпочтениях с сайта, предоставляющего свободный доступ к обширной базе данных книг, аннотаций, различных обзоров, Goodreads [15].

Данный набор был выбран по причине наличия в нём таких сущностей, как объект потребления, пользователь и фиксирование факта взаимодействия пользователя с объектом (оценка книге). Также на выбор повлияла обширность корректных данных и присутствие необходимой характеризующей информации у объектов для тестирования предлагаемой рекомендательной системы.

Выбранный набор представляет собой ряд файлов характеризующую информацию о книгах в различных жанрах и о взаимодействиях пользователей с этими самыми книгами.

Из набора Goodreads Datasets были отобраны следующие файлы, посвященные книгам в жанре «поэзия»:

– goodreads_interactions_poetry – содержит около двух миллионов семисот тысяч пользовательских оценок книгам в жанре «поэзия»;

– goodreads_books_poetry – содержит информацию о тридцати шести тысячах книг в жанре «поэзия».

В рамках написания данной главы для тестирования качества прогнозирования интереса пользователя в программную реализацию разработанной гибридной рекомендательной системы был добавлен специальный пакет с классами для работы с тестовыми данными, а также для определения качества рекомендаций с помощью рассмотренных ранее метрик.

Прежде чем проводить эксперименты все данные из выбранных файлов предварительно были обработаны и распределены по соответствующим таблицам и полям в базе данных.

Для оценивания качества прогнозирования рекомендательной системой пользовательского интереса случайным образом выбирался пользователь из набора данных goodreads_interactions_poetry, у которого поведенческий профиль имеет размерность больше двадцати. После чего у этого пользователя произвольно бралась самая высоко оцененная им книга, относительно которой рекомендательная система уже формировала список рекомендаций равный размерности поведенческого профиля пользователя. Преобладающие цвета обложек книг при этом не использовались.

Исходя из вышеописанного, элементы матрицы ошибок определялись следующим образом:

– TP – книга присутствует в списке рекомендаций, пользователь проставлял ей оценку;

– FP – книга присутствует в списке рекомендаций, пользователь не проставлял ей оценку;

– FN – книга отсутствует в списке рекомендаций, пользователь проставлял ей оценку;

– TN – книга отсутствует в списке рекомендаций, пользователь не проставлял ей оценку.

Суммарно было проведено десять экспериментов, результаты которых можно наблюдать в таблице 3.

Таблица 3 - Экспериментальные исследования

№	Доля правильных ответов	Точность	Полнота	F-мера
1	0,74	0,74	0,77	0,75
2	0,68	0,69	0,69	0,69
3	0,71	0,74	0,80	0,72
4	0,74	0,73	0,74	0,76
5	0,67	0,60	0,78	0,75
6	0,78	0,65	0,73	0,72
7	0,71	0,67	0,82	0,74
8	0,81	0,81	0,86	0,84
9	0,73	0,66	0,79	0,62
10	0,79	0,77	0,84	0,80

На рисунке 17 изображен график, основывающийся на данных, полученных по результатам экспериментов.

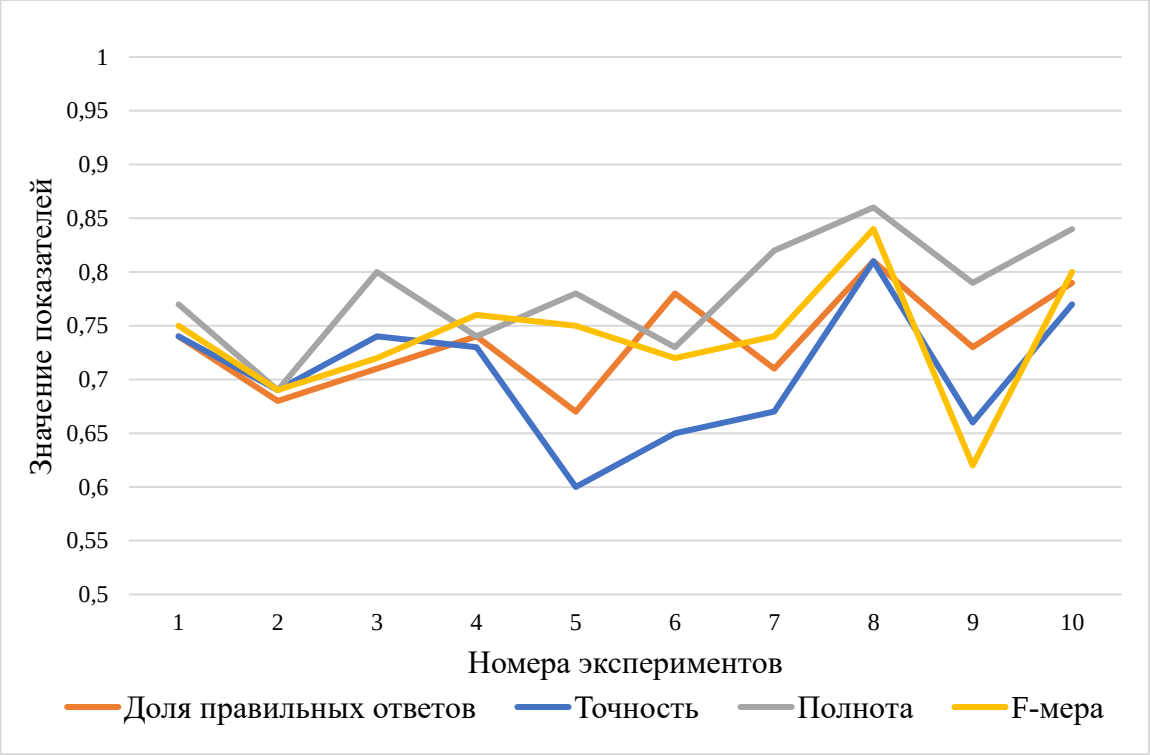


Рисунок 17 – График экспериментальных исследований

Как видно из результатов, реализованная рекомендательная система имеет достаточно высокий показатель охвата, что даёт понять параметр полноты. Показатель точности имеет показатели чуть меньше, что означает, что присутствовали ложные срабатывания, однако, благодаря метрики F-мера можно сделать вывод, что рекомендательная система показывает удовлетворительные результаты в решении поставленной задачи.

4.2 Тестирование возможности интеграции рекомендательной системы в приложения малой электронной коммерции

Для тестирования возможности интеграции гибридной рекомендательной системы с приложениями малой электронной коммерции был разработан простой прототип онлайн-магазина, который выступил площадкой для испытаний предлагаемого решения прогнозирования интересов пользователя.

При разработке приложения для тестирования рекомендательной системы, а также для удобства его развертывания использовалась IDE IntelliJ IDEA. Для работы с реляционной базой данных использовался DBeaver, свободный менеджер баз данных.

Разработка веб-приложения велась на объектно-ориентированном языке программирования Java. Данный выбор обусловлен тем, что рекомендательная система была написана на этом языке программирования. Сборку проекта обеспечивал Maven, фреймворк для автоматизации сборки проектов.

В качестве фреймоворка для написания серверной логики веб-приложения был выбран Spring Boot, так как данное решение позволяет наиболее простым способом создать web-приложение, требуя при этом от разработчика минимум усилий по его настройке и написанию кода. Благодаря своей компонентной архитектуре, Spring Boot особенно подходит для разработки различных приложений, например, таких как проекты электронной коммерции.

Для реализации интерфейса веб-приложения использовались html-теги, которые сообщают браузеру каково содержание страницы, и css – язык описания внешнего вида html-документа.

Также для создания адаптивного дизайна веб-приложения применялись популярные библиотеки jQuery и Bootstrap 4.

Для хранения данных использовалась свободная объектно-реляционная система управления базами данных PostgresSQL. Выбор остановился на PostgresSQL по причине того, что именно данная СУБД чаще всего ассоциируются с веб-приложениями. Тестовые данные о товарах была взяты у онлайн-магазина H&M.

Концепция разработанного прототипа онлайн-магазина предполагает работу с пользователем со стандартными правами и правами администратора.

Стоит учесть, что пользователь не сможет просматривать товары до тех пор, пока не зарегистрируется в приложении. Такое решение было принято для обеспечения удобства тестирования. В реальных условиях очевидно, что пользователь, зашедший в приложение малой электронной коммерции, может быть не зарегистрирован и его в таком случае нужно фиксировать, например, по кукам. Однако, чтобы не выполнять необязательную в текущей ситуации работу, было решено позволить работать с приложением только зарегистрированным пользователям.

Диаграмма вариантов использования приложения зарегистрированным пользователем и администратором представлена на рисунке 18.

Первое, что встречает пользователь при открытии веб-приложения для тестирования предложенной гибридной рекомендательной системы, так это страницу аутентификации, на которой можно ввести данные для входа в приложение.

Если же пользователь ранее не был зарегистрирован в веб-приложении, то он может это сделать, перейдя на страницу регистрации по ссылке в верхней части сайта «Перейти к регистрации».

В случае, если пользователь обладает стандартными правами в приложении, то при аутентификации его перебрасывает на страницу с товарами.

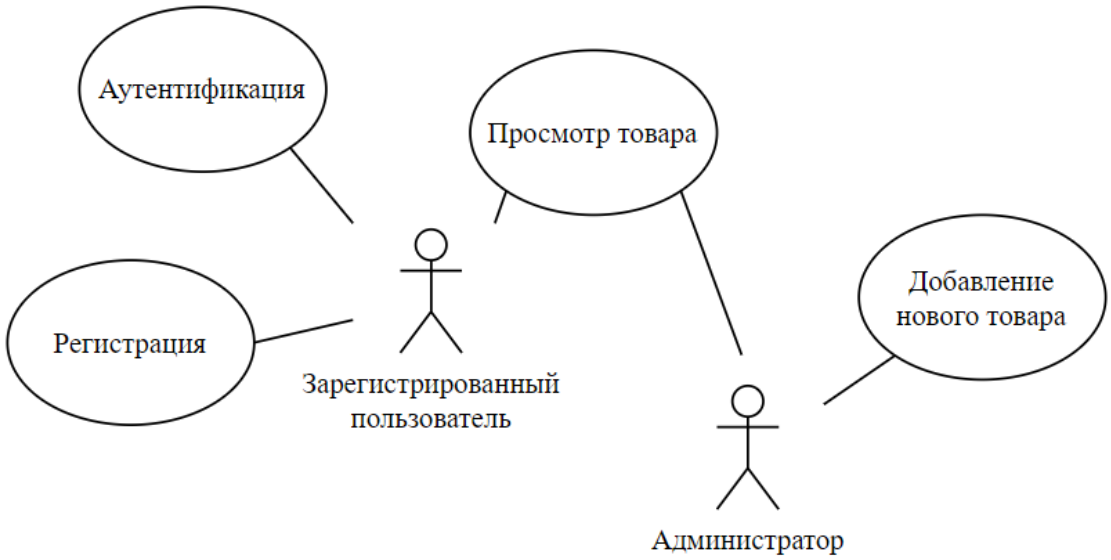


Рисунок 18 – Диаграмма вариантов использования прототипа онлайн-магазина для тестирования рекомендательной системы

Для упрощения разработки было решено не создавать страницу с детальной информацией о товаре, переход пользователя на которую бы фиксировался при открытии заинтересовавшей его продукции. В качестве альтернативы данному действию под каждой карточкой товара была добавлена кнопка «Просмотр товара» (рисунок 19).

В случае, если пользователь обладает правами администратора, то он переходит на страницу добавления нового товара в базу данных.

Первым делом для подключения разработанной рекомендательной системы в новый проект, необходимо в pom.xml приписать репозиторий и саму зависимость, чтобы Maven знал откуда выкачивать библиотеку с рекомендательной системой.

После того, как рекомендательная система была подключена в проект, было выполнено наследование классами «User» и «Product» классов «UserSuperclass» и «ProductSuperclass», соответственно.

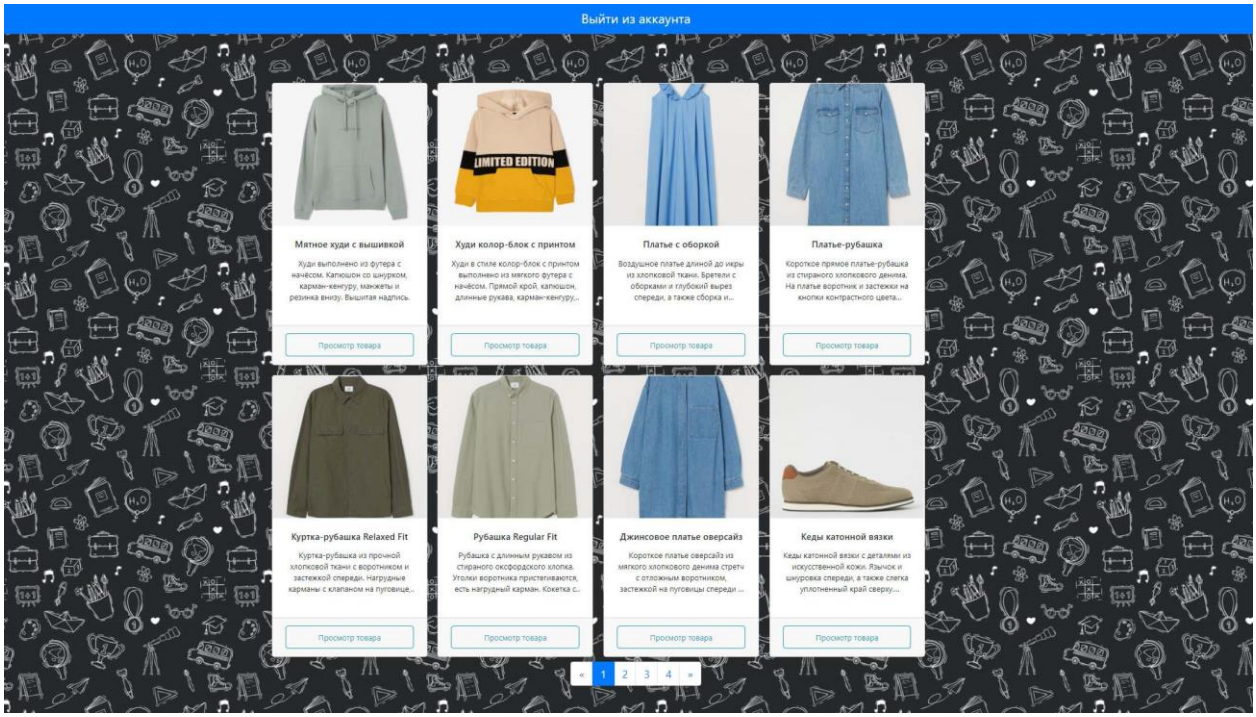


Рисунок 19 – Страница с товарами

Помимо этого, была произведена замена механизма вывода товаров на страницу с общим ассортиментом. Теперь вместо беспорядочного вывода всего доступного ассортимента используется метод рекомендательной системы «getRecommendationsForUser()», который использует описание, характеристики и преобладающие цвета на изображении товара для создания рекомендаций пользователю.

Также была добавлена обработка кнопки «Просмотр товара», которая записывает факт просмотра товара пользователем в таблицы «user_event» и «users_views», посредством специального метода рекомендательной системы.

На странице администратора при создании нового товара был использован метод для получения преобладающих цветов из изображения товара с последующим сохранением их в базу данных.

В результате запуска приложения после интеграции рекомендательной системы, благодаря технологии JPA, в базе данных были созданы таблицы на основе существующих классов сущностей необходимых для работы рекомендательной системы (рисунок 20).

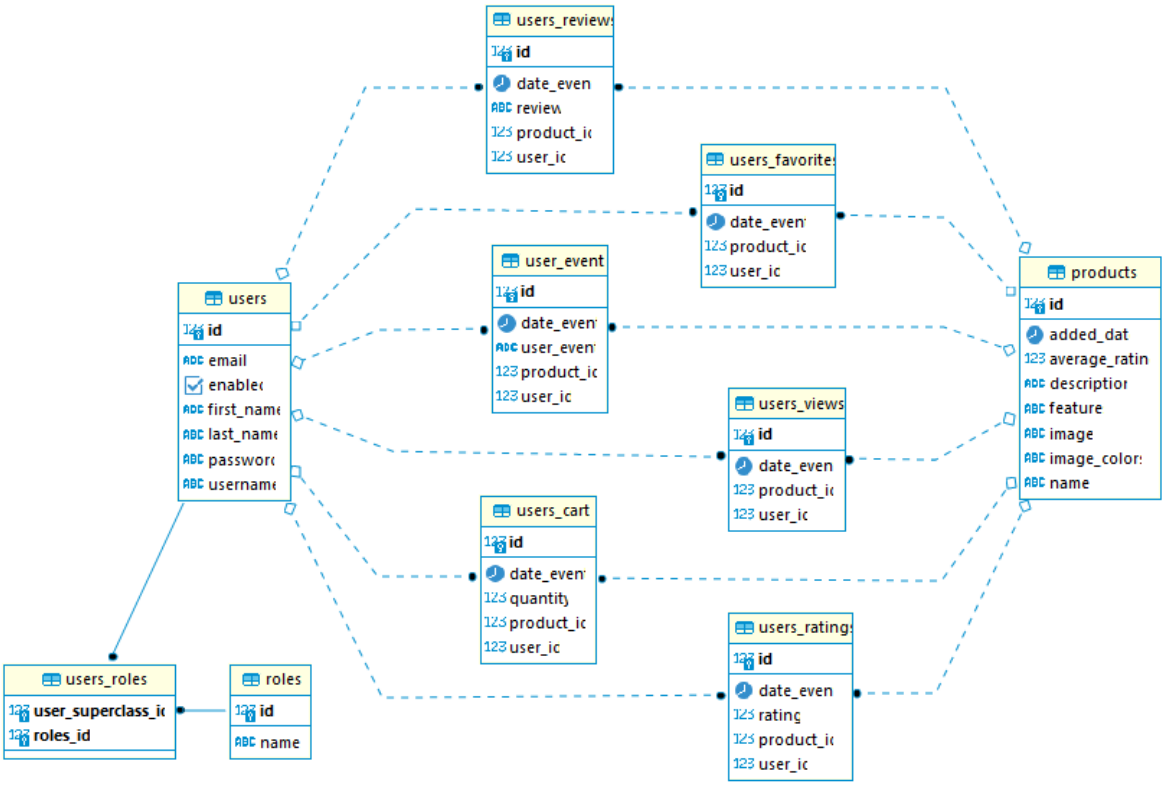
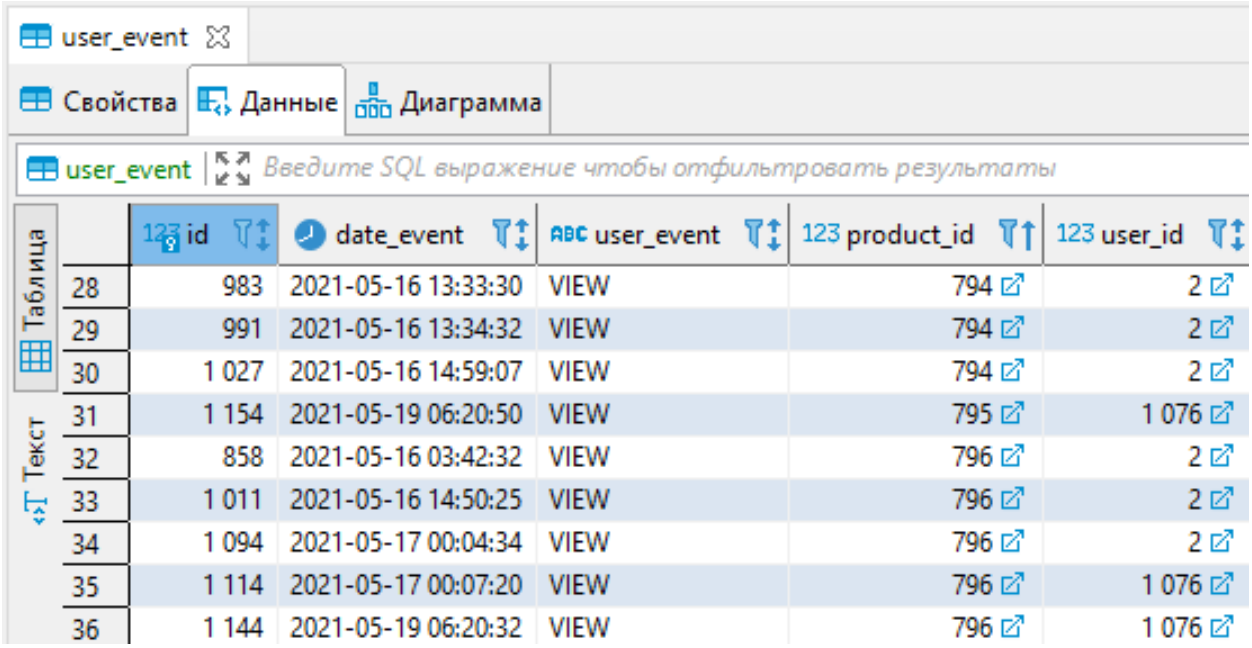


Рисунок 20 – Схема базы данных после интеграции рекомендательной системы

Добавление всех необходимых таблиц в базу данных для работы гибридной рекомендательной системы, а также тот факт, что приложение всё также успешно работает при запуске и открытии страницы с товарами, говорит о том, что интеграция прошла успешно.

После успешного процесса интеграции рекомендательной системы необходимо проверить её работоспособность. Для более естественного тестирования рекомендательной системы был предварительно заполнен поведенческий журнал пользователей. Сделано это было намеренно для того, чтобы неперсонализированная рекомендательная, которая является частью

предложенного решения, могла корректно продвигать наиболее популярные товары среди пользователей (рисунок 21).



	id	date_event	user_event	product_id	user_id
28	983	2021-05-16 13:33:30	VIEW	794	2
29	991	2021-05-16 13:34:32	VIEW	794	2
30	1 027	2021-05-16 14:59:07	VIEW	794	2
31	1 154	2021-05-19 06:20:50	VIEW	795	1 076
32	858	2021-05-16 03:42:32	VIEW	796	2
33	1 011	2021-05-16 14:50:25	VIEW	796	2
34	1 094	2021-05-17 00:04:34	VIEW	796	2
35	1 114	2021-05-17 00:07:20	VIEW	796	1 076
36	1 144	2021-05-19 06:20:32	VIEW	796	1 076

Рисунок 21 – Заполненный поведенческий журнал

Для начала был создан абсолютно новый пользователь. При первом посещении пользователя веб-приложения сработала неперсонализированная рекомендательная система, так как его поведенческий профиль был пуст.

В результате работы выше названной части предложенного решения пользователю вернулись наиболее востребованные товары среди всех пользователей приложения (рисунок 22).

Однако, после того как пользователь обратил внимание на «Кеды» посредством кнопки «Просмотр товара», поведенческий журнал пользователя больше не пуст, а это значит, что в работу вступает фильтрация на основе содержания.

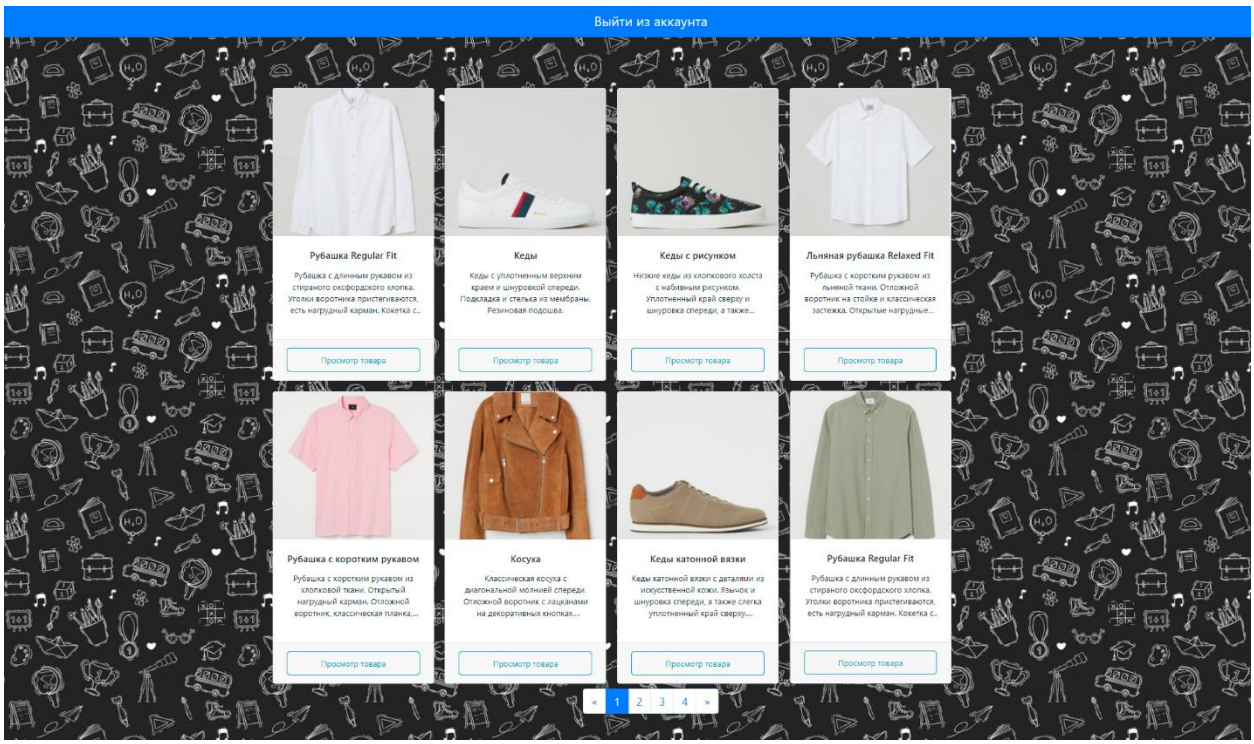


Рисунок 22 – Наиболее популярные товары среди пользователей

На рисунке 23 представлен последний просмотренный пользователем товар и коэффициенты схожести других товаров с ним.

Сходство на базе описания, характеристик и изображения:

1221	Кеды	1,00000
1223	Высокие кеды	0,63985
888	Кеды катонной вязки	0,63434
1184	Кеды	0,62553
1158	Массивные кеды	0,59141
1159	Высокие холщовые кеды	0,58799
1156	Кеды с рисунком	0,56662
1186	Холшовые ботинки	0.55350

Рисунок 23 – Коэффициенты схожести товаров с тем, что заинтересовал пользователя ранее

Список рекомендаций для пользователя, сформированный фильтрацией на основе содержания, представлен на рисунке 24.

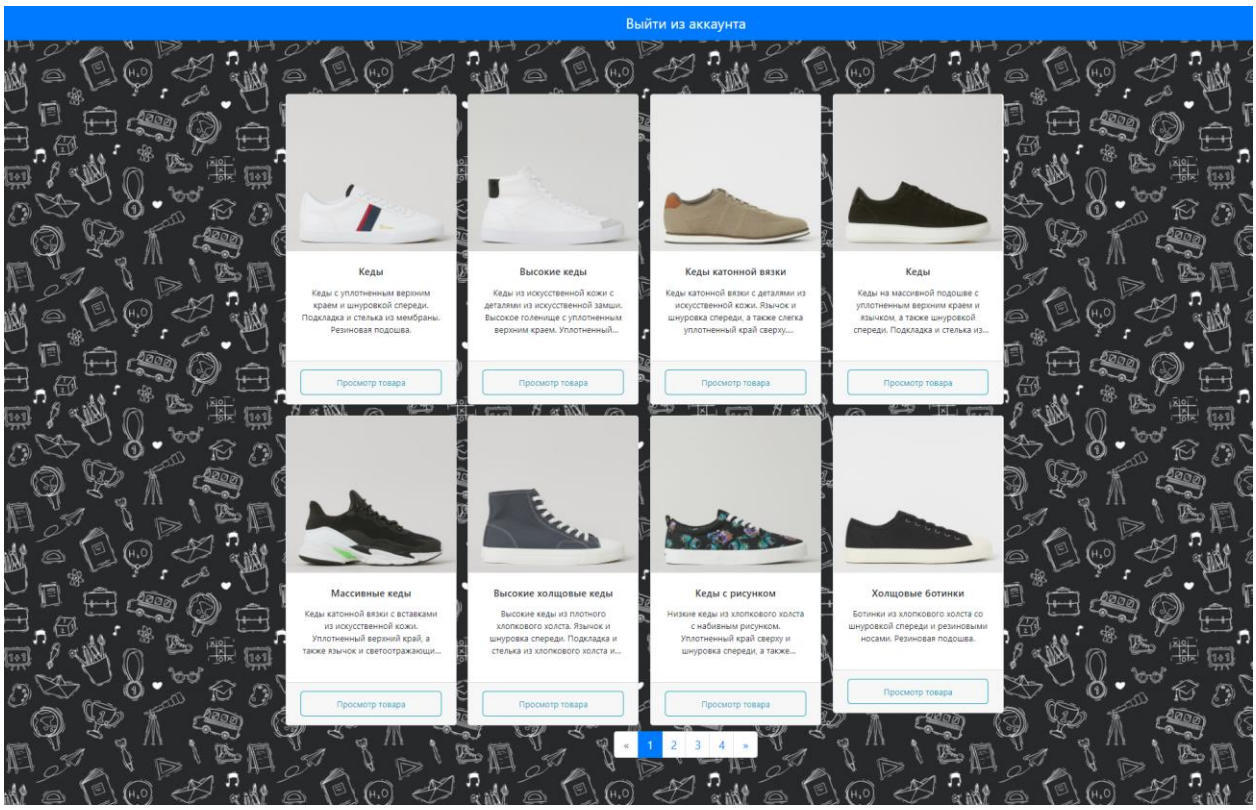


Рисунок 24 – Результат работы фильтрации на основе содержания

Как видно на рисунке 24 пользователю теперь рекомендуются схожие товары на кеды, что ему понравились ранее. В самое начало списка товаров вынеслись наиболее релевантные товары для текущего пользователя, а менее интересующие по мнению рекомендательной системы ушли в конец.

4.3 Анализ результатов проведенных экспериментов

В ходе испытаний, помимо указанных в данной работе, было выявлено, что рекомендательная система демонстрирует наилучшие показатели, когда описание максимально подробно описывает содержание товара, характеристики наиболее точно отображают параметры товара, а на изображении присутствует только товар с однородным фоном.

Также в процессе ряда испытаний было установлено оптимальное количество искомых преобладающих цветов на изображении товара,

равняемое трем. Первоначально количество, вынимаемых преобладающих цветов из изображения товара, равнялось десяти. Однако экспериментальным путем было установлено, что при большом количестве, извлекаемых преобладающих цветов из изображения, алгоритм k-means вынимал значительную долю цветов из фона. В результате чего показатели сходства, преобладающих цветов изображений товаров, были довольно высоки, в то время как цвета самих товаров, представленных на изображениях, объективно различались.

Стоит признать, что в случае рекомендации одежды использование сходства, преобладающих цветов изображения, безусловно имеет смысл. Тем не менее, если речь, например, идет о книгах, то,

конечно, в таком случае сравнивать цвета обложек не рационально. Для таких случаев предусмотрена возможность отключения сравнения преобладающих цветов товаров, оставив тем самым только использование текстовой информации о товарах. Это говорит о том, что прежде, чем использовать предлагаемую гибридную рекомендательную систему ритейлеру лучше предварительно ознакомиться с параметрами настройки, которые используются для формирования списка рекомендаций.

4.4 Развитие предлагаемой рекомендательной системы в дальнейшем

Хотя предлагаемая рекомендательная система уже может внести свой вклад в развитие малой электронной коммерции, тем не менее, можно провести дополнительную работу по расширению её функциональных возможностей с учетом потенциальной эксплуатации.

В частности, было бы очень полезно получить отзывы реальных пользователей об их опыте работы с рекомендательной системой, что помогло понять, что можно улучшить в её взаимодействии с пользователями.

Кроме того, планируется разработать версию для мобильных приложений на платформе Android. А также добавить поддержку английской локали.

В дальнейшем планируется улучшать и дополнять новым функционалом разработанную рекомендательную систему для области малой электронной коммерции.

Выводы по четвертой главе

В ходе тестирования разработанной рекомендательной системой были проведены экспериментальные исследования, посвященные оценки её качества прогнозирования интереса пользователя. В результате ряда экспериментов были получены следующие среднеарифметические результаты: точность – 0,706, полнота – 0,78, F-мера – 0,739 и доля правильных ответов – 0,736, что позволяет судить о эффективности рекомендательной системы в решении задачи прогнозирования интереса пользователя в области малой электронной коммерции.

При тестировании интеграции гибридной рекомендательной системы в Spring-приложения сбоев обнаружено не было, все разработанные модули работоспособны.

На основе проведенных экспериментов и полученных результатов был проведен анализ и сделаны выводы о том, что от ритейлера требуется максимально развернутое и корректное заполнение данных о товаре. Особенно это касается описания, так как это напрямую влияет на результативность поиска похожих товаров на те, что ранее заинтересовали пользователя. Также были озвучены размышления на тему, то куда можно развивать и улучшать данное решение.

Без сомнений внедрение подобной технологии в малую электронную коммерцию поспособствует увеличению объёма продаж товара и лояльности покупателей.

Заключение

В процессе выполнения диссертация была определена актуальность исследования, а также проанализировано назначение рекомендательных систем и подходы их построения.

Основной частью данной работы является исследование алгоритмов построения рекомендательных систем. В качестве предметной области была выбрана область малой электронной коммерции.

Исследованный алгоритм действий по созданию рекомендательной системы представляет собой следующий ряд шагов:

– определение наиболее оптимального подхода к построению рекомендательной системы в рамках рассматриваемой предметной области с помощью сравнительного анализа сильных и слабых сторон различных подходов;

– выбор и описание методов и алгоритмов для реализации выбранного ранее подхода;

– осуществление программной реализация предлагаемой рекомендательной системы с использованием различных инструментов и программных средств;

– тестирование качества прогнозирования пользовательского интереса разработанной рекомендательной системой, а также испытание возможности её интеграции в сторонние приложения.

Реализация предложенного в данной работе способа решения поставленной задачи представляет собой гибридную рекомендательную систему с неперсонализированным подходом и фильтрацией на основе содержания. Фильтрация на основе содержания подбирает похожие товары на тот, что ранее заинтересовал пользователя, используя такие составляющие содержания, как описание, характеристики и преобладающие цвета на изображении товара. По умолчанию при формировании списка рекомендаций применяется только текстовая составляющая содержания товаров (описание и

характеристики), однако, если ритейлер считает, что при создании рекомендаций необходимо учитывать цвета изображения товара, то он сможет активировать использование данной характеризующей информации вручную.

Для того, чтобы фильтрация на основе содержания вступила в работу необходимо, чтобы пользователь взаимодействовал хотя бы с одним с товаром. До тех пор же, пока поведенческий профиль пользователя пуст, за формирование списка пользовательских рекомендаций отвечают методы неперсонализированного подхода – метод статистических данных и метод продвижения товаров по признаку. По умолчанию используется реализация метода продвижения товаров по признаку, заключающаяся в продвижение наиболее популярных товаров среди пользователей.

В ходе работы предлагаемая рекомендательная система была практически реализована на объектном-ориентированном языке программирования Java с возможностью интеграции в Spring приложения. Доля правильных ответов и F-мера при тестировании качества прогнозирования интереса пользователя фильтрацией на основе содержания на тестовых данных составили 0,736 и 0,739, соответственно, что позволяет сделать выводы об эффективности способа решения поставленной задачи. При тестировании возможности интеграции рекомендательной системы в сторонние приложения и при её функциональном тестировании программных сбоев обнаружено не было.

Разработанную гибридную рекомендательную систему можно использовать для совершенствования продвижения товаров в онлайн магазинах малой электронной коммерции.

Исходный код был представлен в виде библиотеки и выложен в свободный доступ на GitHub для того, чтобы другие разработчики могли использовать его в своих программных продуктах.

В качестве перспективы развития можно назвать создание рекомендательной системы на основе описанной модели для приложений электронной коммерции, разработанных на платформе Android.

Список используемой литературы

1. Байбардина, Т.Н. Торговая реклама непродовольственных товаров: учеб. пособие / Т.Н. Байбардина, О.А. Бурцева, Т.Л. Процко. – М.: Высшая. школа, 2016. - 207 с.
2. Гречачин, В.А. К вопросу о токенизации текста // Международный научно-исследовательский журнал. – 2016. – №6 (48). – С. 25-27.
3. За 2018 год российский рынок онлайн-торговли вырос на 59% [Электронный ресурс] // Ассоциации компаний Интернет-торговли – Режим доступа: https://www.akit.ru/аналитика-акит2018 (дата обращения: 12.02.2020).
4. Метод k-средних (K-means) [Электронный ресурс] // Loginom – Режим доступа: https://wiki.loginom.ru/articles/k-means.html (дата обращения: 14.02.2020).
5. О программировании, алгоритмах и не только [Электронный ресурс] // Стоп-символы русского языка – Режим доступа: http://www.algorithmist.ru/2010/12/stop-symbols-in-russian.html (дата обращения: 18.02.2020).
6. Новый мир e-commerce в России: как изменился рынок в 2020 году и что его ждет [Электронный ресурс] // Sostav – Режим доступа: https://www.sostav.ru/publication/cityads-45530.html (дата обращения: 24.05.2021).
7. Плавное введение в Natural Language Processing (NLP) [Электронный ресурс] // DataStart – Режим доступа: https://datastart.ru/blog/read/plavnoe-vvedenie-v-natural-language-processing-nlp (дата обращения: 09.04.2020).
8. Реализация алгоритма k-means (k-средних) на примере работы с пикселями [Электронный ресурс] // Хабр – Режим доступа: https://habr.com/ru/post/427761 (дата обращения: 07.02.2020).
9. Рекомендательные системы: You can (not) advise [Электронный ресурс] // Хабр – Режим доступа: https://habr.com/ru/post/176549 (дата обращения: 27.04.2020).

10. Стеммер Портера для русского языка [Электронный ресурс] // Medium – Режим доступа: https://medium.com/@eigenein/стеммер-портера-для-русского-языка -d41c38b2d340 (дата обращения: 18.05.2020).

11. Часть 1. Введение в подходы и алгоритмы [Электронный ресурс] // IBD Developer – Режим доступа: https://www.ibm.com/developerworks/ru/library/os-recommender1/index.html (дата обращения: 11.12.2019).

12. Часть 2. Механизмы с открытым исходным кодом [Электронный ресурс] // IBD Developer – Режим доступа: URL: https://www.ibm.com/developerworks/ru/library/os-recommender2/index.html (дата обращения: 25.03.2020).

13. Цвет и его модели [Электронный ресурс] // КомпьюАрт – Режим доступа: https://compuart.ru/article/23772 (дата обращения: 27.02.2020).

14. Beginners Guide to learn about Content Based Recommender Engines Python [Электронный ресурс] // Analytics Vidhya – Режим доступа: https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems (дата обращения: 17.05.2020).

15. Goodreads Datasets [Электронный ресурс] // UCSD Book Graph – Режим доступа: https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home (дата обращения: 07.06.2021).

16. Image Based Product Recommendation System [Электронный ресурс] // Roberto Reif – Режим доступа: https://www.robertoreif.com/blog/2018/05/14/product-recommendations-using-image-similarity-yy76x (дата обращения: 12.03.2020).

17. Non-Personalised Recommender System in Python [Электронный ресурс] // Medium – Режим доступа: https://medium.com/@tomar.ankur287/non-personalised-recommender-system-in-python-42921cd6f971 (дата обращения: 18.03.2020).

18. Online: UK, Europe & N. America [Электронный ресурс] // Centre for Retail Research – Режим доступа: https://www.retailresearch.org/online-retail.html (дата обращения: 14.02.2020).

19. Recommender systems [Электронный ресурс] // User Profile – Режим доступа: http://recommender-systems.org/user-profile (дата обращения: 20.05.2020).

20. Stemming? Lemmatization? What? [Электронный ресурс] // Towards Data Science – Режим доступа: https://towardsdatascience.com/stemming-lemmatization-what-ba782b7c0bd8 (дата обращения: 12.05.2020).

21. The list of stop words [Электронный ресурс] // CountWordsFree: Text Processing Tools – Режим доступа: https://countwordsfree.com/stopwords (дата обращения: 17.05.2021).

22. Using python and k-means to find the dominant colors in images [Электронный ресурс] // charlesleifer.com – Режим доступа: https://charlesleifer.com/blog/using-python-and-k-means-to-find-the-dominant-colors-in-images (дата обращения: 10.04.2020).

23. Chen, J. Product recommendation system for small online retailers using association rules mining / J. Chen, C. Miller, G. Dagher // In Proceedings of the International Conference on Innovative Design and Manufacturing. – 2014. – P. 71-77.

24. Falk K. Practical Recommender Systems, Manning Publications: 1st edition. – 2019. – 432 p.

25. Huang A. Similarity measures for text document clustering, In Proceedings of the Sixth New Zealand Computer Science Research Student Conference, 2008 – 49-56 pp.

26. Jawaheer, G. Modeling user preferences in recommender systems: a classification framework for explicit and implicit user feedback / G. Jawaheer, P. Weller, P. Kostkova // ACM Transactions on Interactive Intelligent Systems. – 2014. – Vol. 4. – №2. – P. 1-26.

27. Jones, K.S. A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation. – 2004. – Vol. 60. – №5. – P. 493-502.

28. Kaminskas, M. Product recommendation for small-scale retailers / M. Kaminskas, D. Bridge, F. Foping, D. Roche // In International Conference on Electronic Commerce and Web Technologies. – 2015. – P. 17- 29.

29. Kaminskas, M. Product-Seeded and Basket-Seeded Recommendations for Small-Scale Retailers / M. Kaminskas, D. Bridge, F. Foping, D. Roche // Journal on Data Semantics. – 2016. – Vol. 6. – №1. P. 3-14.

30. Manning, C. D. An Introduction to Information Retrieval Draft / C. D. Manning, P. Raghavan, H. Schütze // Information Retrieval Journal. – 2010. - Vol. 13. – № 2. – P. 192-195.

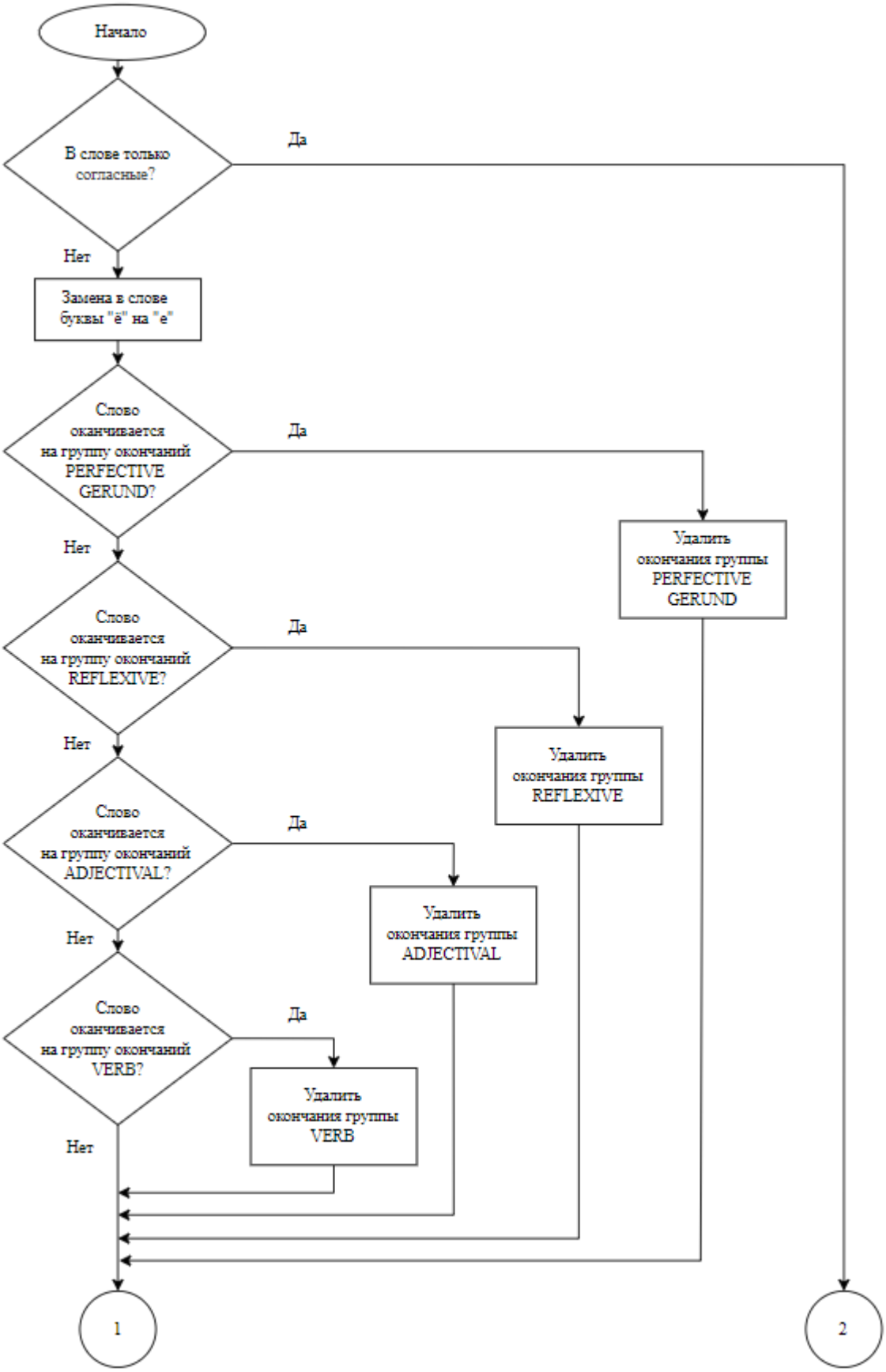
31. Ming, L. Grocery shopping recommendations based on basket-sensitive random walk / L. Ming, B. M. Dias // In Proceedings of the 15th International Conference on Knowledge Discovery and Data mining. – 2009. – P. 1215–1224.

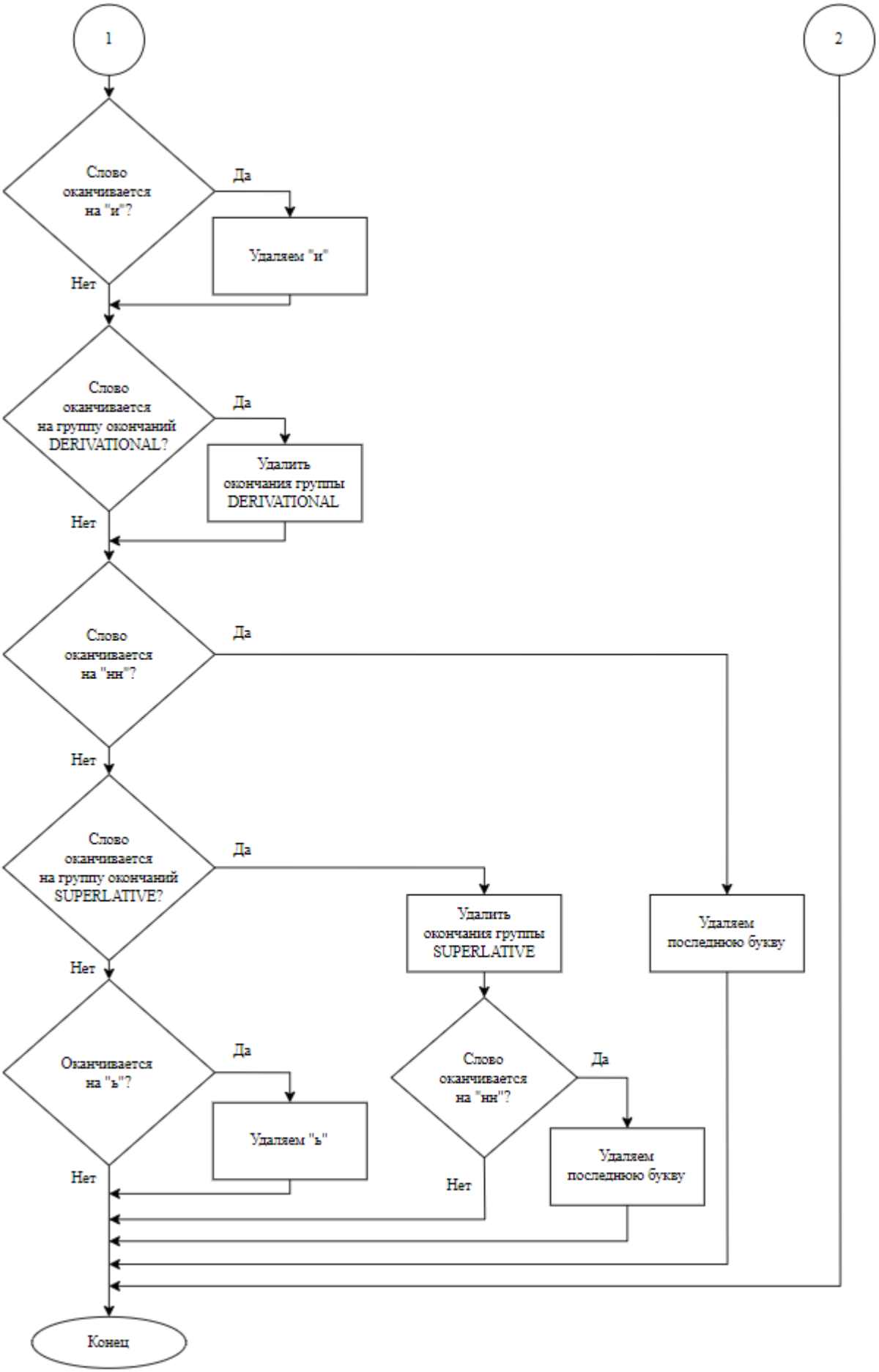
32. Pazzani, M. Learning and revising user proles: The identication of interesting web sites/ M. Pazzani, D. Billsus // Machine Learning. – 1997. – № 27, P. 313-331.

33. Porter, M.F. An algorithm for suffix stripping // Program: electronic library and information systems. – 1980. - Vol. 14. – № 3. – P. 130-137.

Приложение А

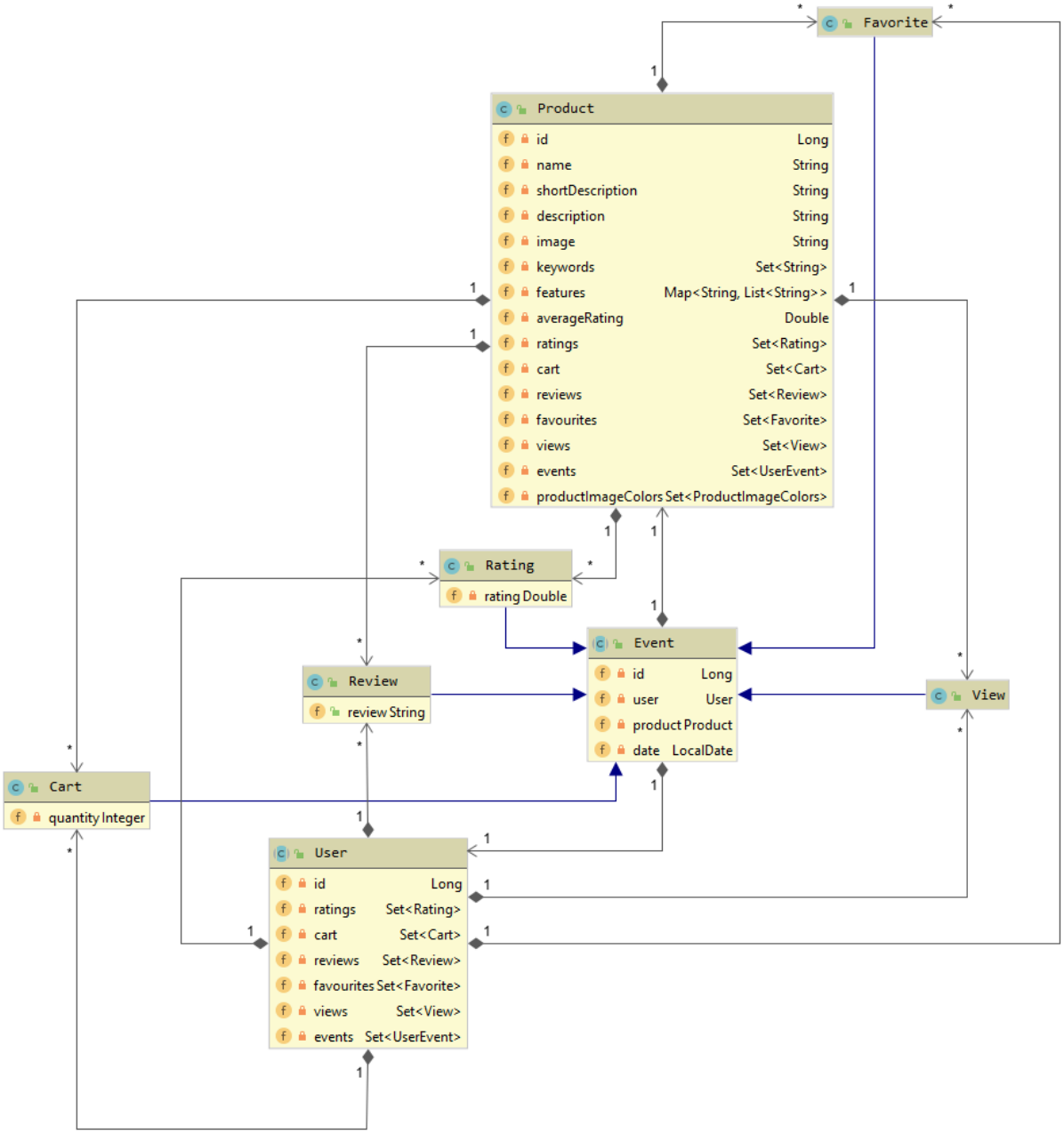
Блок-схема алгоритма стемминга Портера




1
2
Слово оканчивается на "и"?
Да
Удаляем "и"
Нет
Слово оканчивается на группу окончаний DERIVATIONAL?
Да
Удалить окончания группы DERIVATIONAL
Нет
Слово оканчивается на "нн"?
Да
Нет
Слово оканчивается на группу окончаний SUPERLATIVE?
Да
Удалить окончания группы SUPERLATIVE
Удаляем последнюю букву
Нет
Оканчивается на "ь"?
Да
Удаляем "ь"
Слово оканчивается на "нн"?
Да
Удаляем последнюю букву
Нет
Нет
Конец

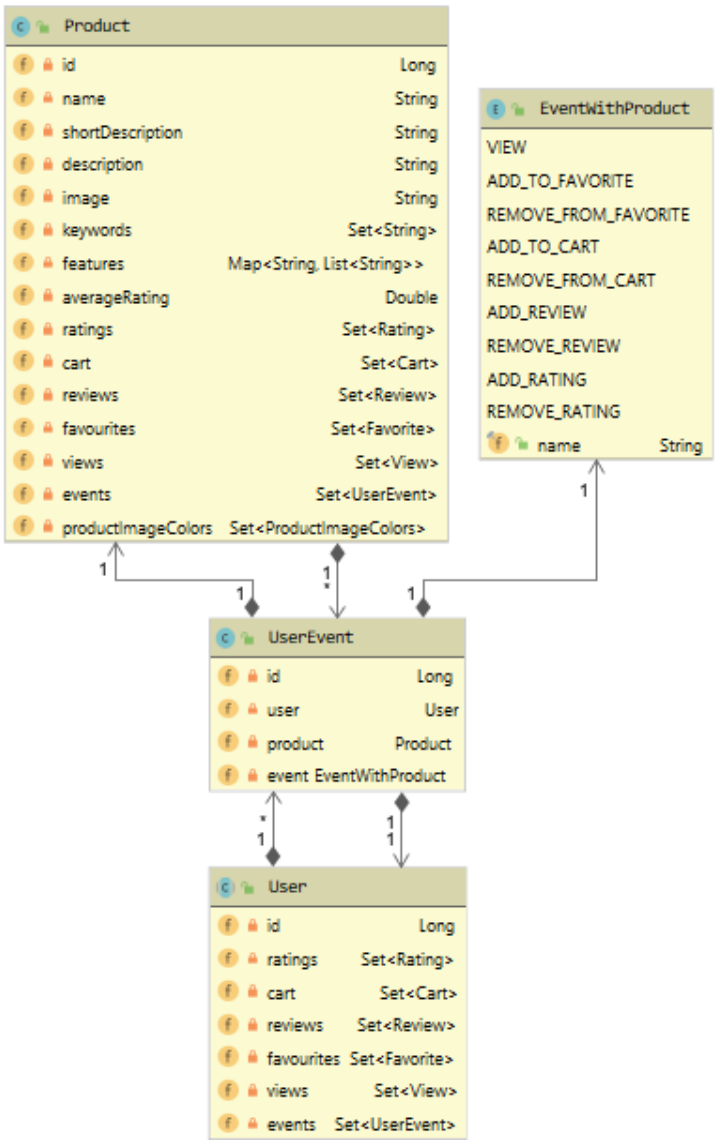
Приложение Б

UML-диаграмма классов, описывающих действия пользователя с товарами в приложении



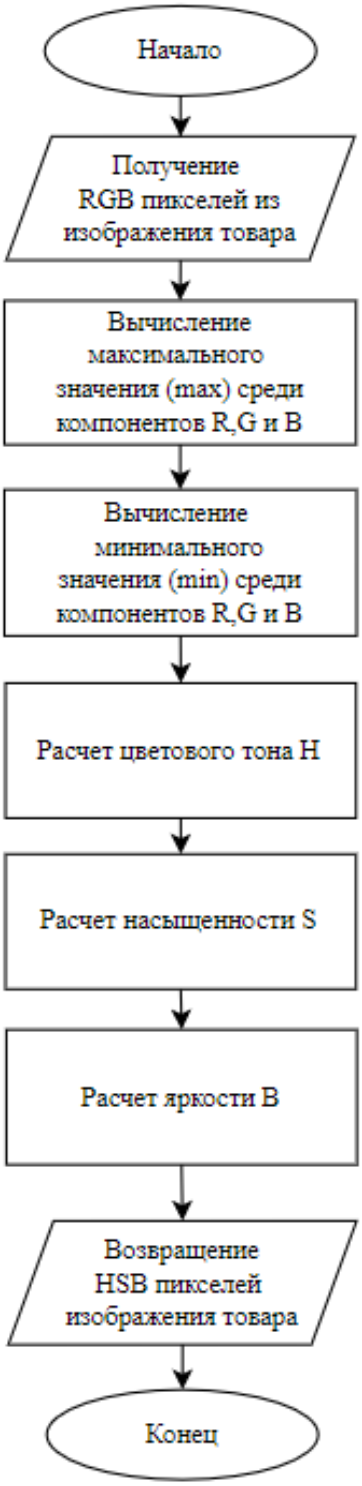
Приложение В

UML диаграмма-классов, образующих поведенческий профиль пользователя



Приложение Г

Блок-схема алгоритма преобразования пикселей изображения из RGB модели в HSB



Приложение Д

Блок-схема алгоритма k-means