Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.04.48

Шнейдер
Вероника Владимировна

Использование облачных технологий и сервисов при реализации системы подбора литературы

АВТОРЕФЕРАТ

на соискание степени магистра информатики и вычислительной техники
по специальности 1-40 81 04 «Обработка больших объемов информации»

_____

Научный руководитель
Теслюк Владимир Николаевич
доцент кафедры Информатики
кандидат физико-математических наук

_____

Минск 2017

ВВЕДЕНИЕ

Современное Интернет-пространство предоставляет пользователю огромное количество разнообразной информации, в которой становится все сложнее ориентироваться, поэтому применение классических средств поиска и систематизации не может полностью удовлетворить потребности пользователя: невозможно просмотреть все материалы, чтобы выбрать для себя подходящие. В связи с этим стало появляться все больше так называемых РС, которые ориентированы на предоставление информации, наиболее полно удовлетворяющие интересы пользователя и отвечающие его запросу.

Рекомендательные системы – сравнительно новый класс ПО, в чью задачу входит изучение вкусов пользователя путем анализа его действий и оценок и являются одним из важных разделов интеллектуального анализа данных – Data Mining. Это программы, которые пытаются предсказать, какие объекты (фильмы, музыка, книги, новости, веб-сайты, услуги) будут интересны пользователю, имея определенную информацию о его профиле.

РС используются, как правило, в коммерческих целях (Интернет-магазины, каталоги), что увеличивает интерес к ним с экономической точки зрения. Помимо этого РС используют в сервисах, производящих и публикующих контент (новостные порталы, журналы) и в контекстной рекламе.

РС служит разным целям: сократить время поиска подходящих товаров, среднего чека покупки, количество товаров в корзине, время пребывания посетителя на сайте, глубину просмотра и вовлеченность.

Предпоссылками для популяризации РС стало увеличение данных не только в Интернет-среде и реальной жизни, но и невозможность человека пропустить через себя столь большой объем информации и выбрать необходимое.

Поэтому необходимо было разработать технологию, которая помогла бы найти пользователю то, в чем он нуждается, и помочь избежать того, на что он не желает тратить свое время и внимание.

Целью данной работы является изучение РА, выбор оптимального и на его основе создание РС по подбору литературы.

Для достижения данной цели будет проведен анализ существующих РА, выявлены их положительные стороны и недостатки. Будут исследованы существующие облачные технологии для хранения БД и API, которые продоставляют готовые БД книг. Итогом магистерской диссертации станет приложение, которое на основании личных предпочтений пользователя будет рекомендовать ему к прочтению определенные книги.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Цель и задачи исследования

Целью диссертационной работы является исследование типов РС, их алгоритмов, выбор оптимального и на его базе создание приложения по подбору литературы с использованием облачных технологий и сервисов.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ существующих РА, выявить достоинства и недостатки.
2. Выбрать оптимальный РА на основании полученных результатов.
3. Провести исследования необходимых ОТиС и выбрать оптимальные для приложения.
4. Реализовать приложение на основании полученных выводов по ОТиС и при использовании РА.
5. Провести экспериментальные исследования разработанного приложения.

Объектом исследования является использование методов обработки больших объемов данных при разработке рекомендательных систем.

Предметом исследования является рекомендательная системы.

Основной *гипотезой*, положенной в основу диссертационной работы, является возможность использования алгоритмов предоставления схожих по параметрам объектов на основании определеной выборки. В данный момент существует большое количество алгоритмов, позволяющих проводить данный анализ. Каждый из них предполагает различную методику анализа и опирается на собственную математическую модель. Из-за того, что существующие модели незначительно учитывают различные данные, которые важны, некоторые РА являются неточными. Путем рассмотрения этих данных можно добиться хорошей выборки, что приведёт к более эффективному РА.

Связь работы с приоритетными направлениями научных исследований и запросами реального сектора экономики

В связи с большим ростом интернет сервисов в сфере продаж (онлайн-магазины) большое внимание компании стали уделять рекомендательным системам. РА в их основе анализирует данные пользователя на основе его персональной информации или профилей из социальных сетей и создает выборку предполагаемых товаров, которые могут быть ему интересны.

Интернет сервисы, в которые интегрирована РС или ее модуль, заметно превосходят другие сервисы. Так как данная РС помогает реализовывать

больше товаров и увеличивает доход компании или владельца. Существует множество видов РА и зачастую владельцам интернет сервисов и разработчикам не удается интегрировать верный модуль РС, что приводит к снижению доходов. С учётом данных факторов, в реальном секторе экономики существует необходимость разработки удобной, простой и полной РС .

Личный вклад соискателя

Результаты, приведенные в диссертации, получены соискателем лично. Вклад научного руководителя В. Н. Теслюка, заключается в формулировке целей и задач исследования.

Апробация результатов диссертации

Основной алгоритм рекомендательной системы был представлен на 53-й научной конференции студентов, магистрантов и аспирантов БГУИР.

Один из модулей системы был интегрирован в предыдущую работу соискателя «Интерактивная среда онлайн-обучения» и был представлен на 53-й научной конференции студентов, магистрантов и аспирантов БГУИР.

Опубликованность результатов диссертации

По теме диссертации опубликовано 2 печатных работ, из них 2 работы в сборнике трудов и материалов 53-й научной конференции студентов, магистрантов и аспирантов БГУИР.

Структура и объем диссертации

Диссертация состоит из введения, общей характеристики работы, трех глав, заключения, списка использованных источников, списка публикаций автора и приложений.

В первой главе представлен анализ предметной области и алгоритмов РС, выявлены основные существующие проблемы в рамках тематики исследования, показаны направления их решения.

Вторая глава посвящена обзору программных средств и технологий, необходимых для реализации РС.

Третья глава посвящена обзору облачных ОТиС, созданию БД и интеграции Google API Books.

В четвертой главе идет речь о разработке РС на основе полученных выводов и результатов.

Общий объем работы составляет 120 страниц, из которых основного текста – 85 страниц, 41 рисунков на 18 страницах, 5 таблиц на 2 страницах, список использованных источников из 36 наименований на 3 страницах и 1 приложение на 12 страницах.

КРАТКОЕ СОДЕРЖАНИЕ

Глава 1 Исследование предметной области

Первая глава диссертационной работы имеет обзорный характер. Полностью описана история развития рекомендательных систем, которая хоть и началась в начале 90-х годов, однако серьезнейшим толчком к их развитию явился конкурс Netflix Prize, организованный в 2006 г. компанией Netflix.

Помимо этого были разобраны классификация рекомендательных систем и подходы к их формированию. Классификация состоит из четырех основных критериев — тип сбора данных (явный, неявный), тип схожести (на основании содержания, на основании транзакций), использование модели предметной области (основанные на памяти, основанные на модели, комбинированный).

Данные критерии определяют базовую составляющую существующих фильтров, однако помимо них есть еще 4 подхода, которые также закладываются в основу фильтров и составляют четвертый критерий — подход к формированию рекомендательной системы. Существует четыре типа:

1. основанные на действиях пользователей (англ. user-based);
2. основанные на свойствах объектов (англ. item-based);
3. основанные на знаниях (англ. knowledge-based);
4. гибридный (англ. Hybrid).

В магистерской диссертации приводится их полное описание и сравнение, на основании его был сделан вывод о том, что гибридный подход наиболее эффективен.

Далее подробно рассмотрены рекомендательные алгоритмы. Одним из самых простых и базовых алгоритмов является коэффициент сходства Пола Жаккра [1-А]. Более сложные алгоритмы – это корреляция Пирсона, алгоритмы кластеризации, байесовские сети доверия, цепи Маркова, метод Монте-Карло, классификация по методу Роккио, сингулярное разложение матрицы и BRISMF.

Помимо этого разобраны критерии оценки рекомендательных систем. Критериев оценки достаточно много: точность, ограничение, доверие, неожиданность, назнообразие, устойчивость к атакам, приватность, адаптивность, масштабируемость, покрытие, скорость обучения, степень новизны. Все они зависят от сложности алгоритма.

Также существуют методы качества прогнозов. К ним относятся средняя абсолютная процентная ошибка прогнозирования (Mean Absolute Percent Error – MAPE), средняя абсолютная ошибка прогнозирования (Mean Absolute Error –

MAE), корень квадратный из средней квадратичной ошибки прогнозирования (Root Mean Squared Error – RMSE).

В конце главы приводятся основные проблемы и преимущества РС.

В заключении поставлена задача – реализовать алгоритм, который бы:

1) легко масштабировался до значительно большего, чем Netflix, размера;
2) работал быстро;
3) адаптировался бы к разным наборам данных;
4) работал бы как с явными рейтингами (оценки), так и с неявными (просмотры);
5) позволял бы на лету добавлять новые рейтинги, пользователей, объекты.

В главе 2 подробно рассмотрены необходимые средства и технологии для реализации РС по теме магистерской диссертации.

Глава 2 Обзор программных средств и технологий

В качестве операционной системы использована ОС Ubuntu. Ubuntu - одна из многих бесплатных ОС на ядре Linux. Операционная система имеет понятный интерфейс и ориентирована на обычных пользователей. По умолчанию в Ubuntu входит набор необходимых приложений для работы с документами и Интернетом. Одним словом, Ubuntu хорошая бесплатная альтернатива Microsoft Windows. В начале главы подробно рассмотрены причины выбора данной системы и приведено ее описание.

Для оформления веб-страниц использованы такие языки как HTML, CSS. Они предназначены для оформления верного расположения объектов, стилизации станицы и выполнения интерактивных задач, либо анимации.

HTML (от англ. HyperText Markup Language – «язык гипертекстовой разметки») – стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML. HTML5 (англ. HyperText Markup Language, version 5) – язык для структурирования и представления содержимого всемирной паутины. Это пятая версия HTML.

CSS (англ. Cascading Style Sheets – каскадные таблицы стилей) – формальный язык описания внешнего вида документа, написанного с использованием языка разметки. CSS3 (англ. Cascading Style Sheets 3 – каскадные таблицы стилей третьего поколения) – активно разрабатываемая спецификация CSS.

Язык программирования JavaScript (JS) – прототипно-ориентированный сценарный язык программирования. Является реализацией языка ECMAScript

(стандарт ECMA-262). Согласно рейтингу голландской компании TIOBE на январь 2017 года JavaScript занимает седьмую позицию среди языков программирования. С лета 2015 года для языка JavaScriptt ES 6, или EcmaScript6, является официально принятым стандартом.

Однако в связи с тем, что стандарт новый, не все браузеры его поддерживают. Дабы новый код заработал в старых браузерах используют транскомпилятор Babel.

Babel - это JavaScript транскомпилятор. Транскомпилятор отличается от компилятора тем, что он переводит код из новой версии в более старую - стабильный.

Для сборки проекта использован Webpack. Webpack – это невероятно мощный, гибкий и популярный сборщик модулей JavaScript. Вся конфигурация Webpack представляет из себя обычный CommonJS-модуль, расположенный в файле webpack.config.js. Для того, чтобы познакомиться с этим инструментом, необходимо понимать базовые термины и концепции, лежащие в его основе, четыре кита, на которых всё держится. Верхнеуровневых понятий, на которых всё это работает, всего четыре: точка входа (Entry), место назначения, куда будет производиться вывод результатов (Output), загрузчики (Loaders) и плагины (Plugins).

В качестве БД использован MongoDB, который реализует новый подход к построению баз данных, где нет таблиц, схем, запросов SQL, внешних ключей и многих других вещей, которые присущи объектно-реляционным базам данных. В отличие от реляционных баз данных MongoDB предлагает документо-ориентированную модель данных, благодаря чему MongoDB работает быстрее, обладает лучшей масштабируемостью, ее легче использовать.

WebStorm – среда для разработки на JavaScript, которая подходит как для front-end-разработки, так и для создания приложений на Node.js.

Главное достоинство WebStorm – это удобный и умный редактор JavaScript, HTML и CSS, который также поддерживает языки, такие как TypeScript, CoffeeScript, Dart, Less, Sass и Stylus и фреймворки, например, Angular, React и Meteor.

WebStorm позволяет эффективно разрабатывать приложения на Node.js и поддерживает полноценную отладку Node.js приложений. Новое приложение можно создать, используя шаблон Node.js Express, а необходимые модули установить с помощью встроенного в WebStorm менеджера npm.

Node или Node.js – программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения.

Express.js, или просто Express, каркас web-приложений для Node.js, реализованный как свободное и открытое программное обеспечение под лицензией MIT.

На основании обзора программных средств было сделано заключение:

1) магистерская диссертация будет реализована в операционной системе Ubuntu 17.04;

2) для разработки будет использована среда WebStorm 2017;

3) фреймворком послужит Node.js Express для быстроты реализации приложения;

4) в качестве языка программирования будет использован последний релиз JavaScript – EcmaScript 6 с возможностью создавать классы и модули;

5) для веб-части приложения будут использованы HTML5 и CSS3;

6) дабы старые браузеры могли поддерживать новую версию EcmaScript 6 будет установлен babel, который компилирует код в более старую версию;

7) весь проект будет собран посредством Webpack, что позволит пользовательской стороне отдать большой собраный файл, а разработчику разбить приложение так как удобно;

8) БД для приложения будет реализована с помощью MongoDB.

Глава 3 Использование облачных технологий и сервисов

Идея того, что сейчас мы называем облачными вычислениями, впервые была озвучена Джозефом Карлом Робнеттом Ликлайдером (J.C.R. Licklider) в 1970году, когда он был ответственным за разработку ARPANET (Advanced Research Projects Agency Network). Идея Линклайдера заключалась в том, что каждый человек будет подключен к сети, из которой он будет получать не только данные, но и программы. Другой ученый Джон Маккарти (John McCarthy) говорил о том, что вычислительные мощности будут предоставляться пользователям как услуга (сервис).

Ее развитию поспособствовали ряд факторов [30]. Стремительное развитие сети Интернет, а именно пропускной способности. Хотя в начале 90-х глобальных прорывов в области облачных технологий не произошло, сам факт «ускорения» Интернета дал толчок к скорейшему развитию технологии.

В 1999 году появилась компания Salesforce.com, которая предоставила доступ к своему приложению через сайт. Эта компания стала первой

компанией, предоставившей свое программное обеспечение по принципу «программное обеспечение как сервис» (SaaS).

В 2002 году Amazon запустила свой облачный сервис, где пользователи могли хранить информацию и проводить необходимые вычисления.

В 2006 году Amazon запустила сервис Elastic Compute cloud (EC2), где пользователи могли запускать свои собственные приложения. Таким образом, сервисы Amazon EC2 и Amazon S3 стали первыми сервисами облачных вычислений.

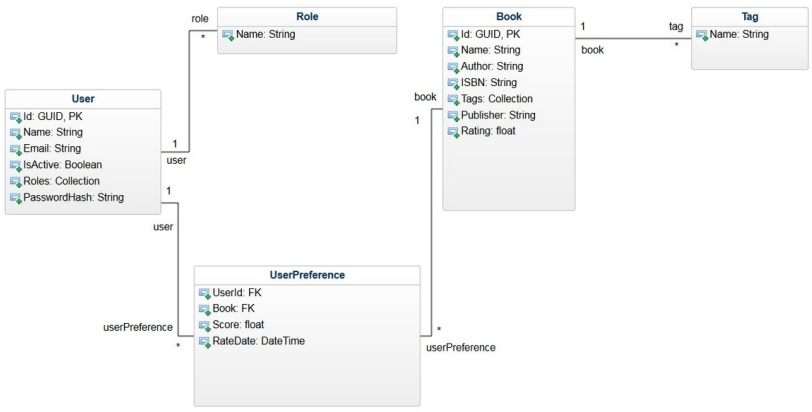
Основное отличие «облачного» программного решения от обычного в том, что вся информация, с которой Вы работаете, сохраняется не на Вашем жестком диске, а на удаленном сервере. Аналогично с производимыми операциями: они нагружают не персональный компьютер или ноутбук, а мощности серверов компании, предоставляющей то или иное приложение. Вы же получаете лишь результат, отправляемый на монитор через интернет.

В данный момент наиболее популярны 5 сервисов для хранения данных в облаке:

1) Amazon Web Service;
2) Google cloud;
3) IBM cloud;
4) Rackspace;
5) Azure.

В 3 главе подробно рассмотрены Amazon Web Service, Google cloud и Azure. Проведено их сравнение и принято решение использовать сервис от Amazon.

Также приведена схема БД приложения. Несмотря на то, что БД кажется небольшой, она позволяет проводить все необходимые операции, необходимые



PC.

Рисунок 3.1 – БД приложения

Полученная БД по теме магистерской диссертации была развернута в MongoDB Atlas. Данный сервис позволяет интегрировать БД на любое облако от трех ведущих компаний: Amazon, Microsoft, Google.

В качестве БД для работы с книгами выбран сервис Google API Books. Google Книги («англ. Google Books») – сервис полнотекстового поиска по книгам, оцифрованным компанией Google (свыше 10 миллионов книг из крупнейших библиотек США).

Служба поиска книг Google Books позволяет разработчикам Web-приложений получать списки книг и метаданных через API на базе архитектуры REST. Модуль Gdata среды Zend Framework предоставляет возможность обрабатывать XML-каналы, генерируемые этим API, и использовать его в контексте настраиваемых Web-приложений. Эта статья знакомит читателя с API данных Google Book Search, демонстрируя, как его можно использовать для поиска книг по ключевым словам, автору и названию; получать эскизы обложек и анонсы, а также добавлять отзывы и библиографические данные в библиотеки пользователей.

Служба Google Books интересна не только для читателя, но и с точки зрения разработчика – своим API данных. Этот API позволяет выполнять чтение и поиск в базе данных Google Books тех книг, которые соответствуют заданным пользователем критериям, и применять результаты этого поиска в других Web-приложениях. Доступ к этому API, который следует модели REST, можно получить с помощью любого инструментария разработки, поддерживающего XML. У этого API уже есть клиентские библиотеки для PHP, JavaTM и других распространенных языков программирования.

В конце главы приведены листинги кода по интеграции сервиса в приложение по магистерской диссертации.

Глава 4 Реализация программного средства

В качестве базы для PC был взят алгоритм BRISMF, который позволяет работать как с неявными, так и с явными данными.

В качестве входных данных система получает информацию о действиях, совершенных пользователем, или об оценках, выставленных им объектам из набора данных. Поскольку данные предполагается как получать, так и использовать на веб-проектах, основной внешний интерфейс был реализован поверх протокола HTTP, по которому в формате JSON передаются как действия и оценки пользователя, так и запросы рекомендаций для него.

Эти запросы могут быть совершены как с помощью AJAX из Javascript-кода, размещаемого на страницах, так и, при желании, с сервера проекта, пользующегося услугами рекомендательной системы.

Такой подход позволяет начать использование рекомендательной системы на новом проекте с минимальными усилиями, сравнимыми с установкой на проект счетчика.

Далее приведены методы предварительной фильтрации для объектов в РА и оптимизация параметров.

В 1 главе мыли рассмотрены две меры качества РС, однако в приложении использована третья – ARP.

В заключении описания алгоритма была приведены примеры его скорости на разных объемах данных, дабы показать его преимущество по сравнению с другими РС.

Представлена диаграмма цикла работы РС. Данный цикл состоит из 8 этапов. Первый этап состоит из наполнения БД РС необходимыми объектами, свойствами и знаниями:

1) сбор данных о действиях пользователей;
2) ввод данных о свойствах объектов;
3) ввод данных о знаниях в сфере рекомендаций.

Второй этап состоит из работы с самим сервисом рекомендаций:

4) отправка данных о текущем пользователе и запрос рекомендаций:;
5) стар основного гибридного алгоритма;
6) использование данных;
7) объединение результатов;
8) выдача рекомендаций.

В конце главы она разобрана подробнее.

На рисунке 4.1 представлены основные окна приложения:

1) витрина (главная страница приложения);
2) фасетный поиск (окно поиска);
3) окно результатов по поиску;
4) страница книги;
5) библиотека пользователя;
6) уведомления;
7) социальный фид (фидбэк от пользователей).

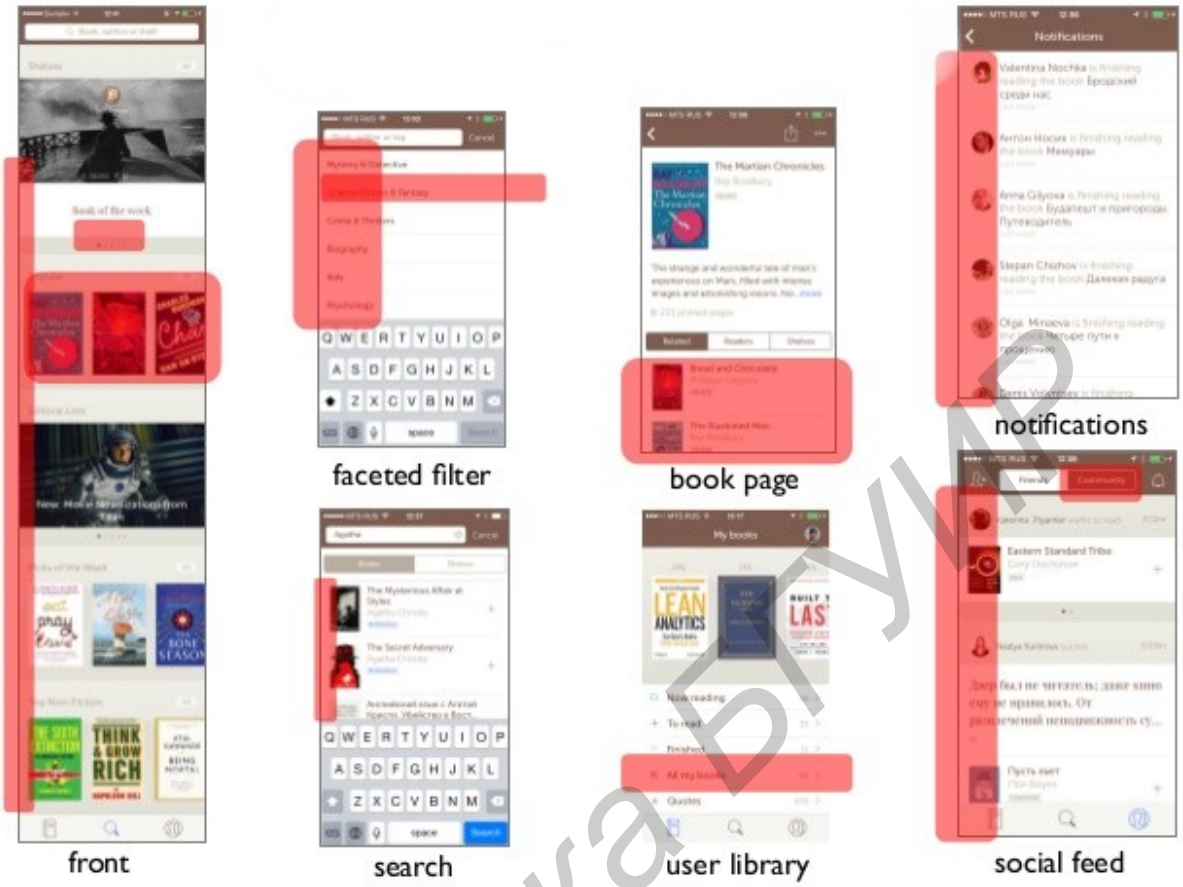


Рисунок 4.1 – Внешний вид РС

На главной странице приложения сразу размещаются книги по определенным тематикам, дабы пользователь мог выбрать определенную и уже на основании ее начала работу РС. При этом книги распределены в блоках двух видов – слайдер постеров и слайдер обложек.

Фасетный поиск – это строка поиска, при нажатии на которую пользователю высвечиваются определенные названия книг (выделено красным). Если не было ничего введено, то показывают наиболее популярные запросы.

Результаты поискать можно сортировать по списку и по сетке. Списком показывается обложка, название, автор и ссылка на книгу. При сетке показываются обложки и названия.

На странице книги также размещена гибридная выборка. Как только пользователь просматривает книгу, то система анализирует не только что он добавил в свою библиотеку, но и учитывает интересы с текущей книгой. Таким образом, если у пользователя в библиотеке детские книги и он просматривает фантастику, то выборка рекомендаций будет осуществляться по двум данным тематикам.

Библиотека пользователя позволяет добавлять новые книги, удалять и сортировать их.

Страница нотификаций показывает какие книги в данный момент лайкнули (поставили хороший отзыв) пользователи, на которых подписан текущий.

Социальный фидбэк: как только кто-то из круга пользователя выставляет рецензию для книги, то эта рецензия попадает в ленту дабы пользователь мог увидеть отзыв тех, за кем он следит.

ЗАКЛЮЧЕНИЕ

С развитием технологий растет и объем данных. Теперь он достигает таких больших размеров, которые человек не в силах через себя пропустить, потому необходимо использовать алгоритмы, которые упростили бы некоторые аспекты нашей жизни. В частности — рекомендательные системы.

Рекомендательные системы помогают экономить время и быстро получать то, что необходимо пользователю. На основе его данных, и, при необходимости, других, они составляют выборку объектов, которые могли бы понравится пользователю.

В магистерской диссертации была описана история разбития РС, их классификация и подходы к формированию, преимущества и недостатки. Было рассмотрено большинство рекомендательных алгоритмов. Для реализации РС диссертационной работы был выбран алгоритм BRISMF. В качестве метода оценки был использован метод ARP.

В результате, взяв в виде базы алгоритм BRISMF, получена система, которая, помимо решения задачи MF, оптимизирует настраиваемые параметры алгоритма, причем учитывает при оптимизации, помимо качества, стабильность результатов и скорость сходимости. Кроме того, был разработан механизм конвертации неявных данных в явные оценки, что значительно расширяет спектр алгоритмов, применяемых для анализа наборов неявных данных. Система показала высокую скорость работы: на полном наборе данных Netflix с 50 уходит около 12 минут при работе в один поток.

Также было исследовано применение метода случайных проекций LSH для ускорения формирования выдачи рекомендаций. Метод показал хорошие результаты работы и может быть рекомендован к использованию, особенно при работе с большим количеством объектов.

Архитектура системы позволяет как пересчитывать вектора профилей пользователей и объектов, обеспечивая при этом постоянную работоспособность, так и формировать рекомендации для новых пользователей на лету, без полного пересчета матриц профилей.

Помимо этого в систему был интегрирован сервис Google Api Books, который, при желании или необходимости, можно заменить на любой другой, и создана БД приложения, которая была размещена в облачном сервисе от фирмы Amazon.

Таким образом, поставленная на магистерское проектирование задача решена в полном объеме, ПО разработано с учетов всех требований, которые

были описаны в главе 1, работоспособность подтверждена в главе 4 и практическим применением. При некоторой доработке ПО может быть использовано в любой области, подразумевающее применение РС.

СПИСОК ОПУБЛИКОВАНЫХ РАБОТ

1-А. Шнейдер, В.В. Механизм рекомендательной системы / В.В. Шнейдер // Компьютерные системы и сети: материалы 53-ей научной конференции аспирантов, магистрантов и студентов. – Минск: БГУИР, 2017. – [в печати].

2-А. Шнейдер, В.В. О пользе и преимуществах интерактивного онлайн-обучения / В.В. Шнейдер, Е.Н. Побыванец // Компьютерные системы и сети: материалы 53-ей научной конференции аспирантов, магистрантов и студентов. – Минск: БГУИР, 2017. – [в печати].