

О.С. Самохвалова, К.Д. Уляшев, Д.Е. Попова

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»
им. В. И. Ульянова (Ленина)

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПОСТРОЕНИЯ РЕКОМЕНДАЦИЙ СООБЩЕСТВ СОЦИАЛЬНОЙ СЕТИ "ВКОНТАКТЕ"

Целью проекта являлась разработка рекомендательной системы для социальной сети ВКонтакте. Предметом рекомендаций были выбраны сообщества в социальной сети ВКонтакте. В качестве алгоритма рекомендации была выбрана коллаборативная фильтрация, но в более простом виде: рекомендации составляются на основе знаний о сообществах, в которых состоят друзья пользователя. В результате выполнения работы было разработано приложение с интерфейсом пользователя, позволяющее пользователю получить список рекомендованных сообществ социальной сети ВКонтакте.

The main goal of the project was to develop a recommendation system for the social network VK. The Communities in the VK were chosen as the subject of recommendations. As a recommendation algorithm, collaborative filtering was chosen, but in a simpler form: recommendations are based on knowledge of the communities in which user's friends take part. As a result, we developed an application with a user interface which allows the user to get a list of recommended communities of the social network VK.

Социальные сети, рекомендации, построение рекомендаций

Введение

Рекомендательные системы – инструменты автоматической генерации предложений товаров и услуг на основе знаний о потребностях клиентов, стали появляться еще в 2000-х, однако настоящий подъем в этой области случился примерно 5-10 лет назад, когда состоялось соревнование Netflix Prize. Тогда компания Netflix давала в прокат не цифровые копии, а рассылала VHS-кассеты и DVD, поэтому для них было очень важно повысить качество рекомендаций, ведь чем лучше Netflix рекомендует своим пользователям фильмы, тем больше фильмов они берут в прокат.

На данный момент существует немало успешных самостоятельных рекомендательных сервисов, таких как Last.fm, Pandora и IMDb. Помимо этого, все крупные социальные сети используют подобные алгоритмы для рекомендаций контента своим пользователям, например, Instagram предлагает потенциально интересные пользователю профили, а ВКонтакте составляет плейлисты с теми музыкальными композициями, которые, по мнению рекомендательной системы, должны понравиться пользователю.

В социальной сети "ВКонтакте" созданы тысячи сообществ по интересам, популярность которых стремительно набирает обороты, но инструменты сети не предоставляют пользователям возможность получить рекомендации в этой области. Также, на просторах интернета нет удобных и доступных решений этой проблемы. Социальная сеть "ВКонтакте" не предоставляет пользователю возможность получить рекомендации сообществ. Приложение, реализующее построение рекомендаций на основе интересов друзей, решает эту проблему.

Для решения данной задачи было решено использовать упрощенную версию коллаборативной фильтрации. Коллаборативная фильтрация предполагает расчет рекомендаций на основе оценок других пользователей. В данном случае мы учитываем информацию о тех сообществах, в которых состоят друзья пользователя, составляем граф интересов и делаем рекомендацию.

Предполагалось разработать веб-приложение с простым и удобным интерфейсом пользователя, с помощью которого пользователь мог бы получить 2 версии корректного списка рекомендованных сообществ: на основе информации о подписках его друзей с учетом и без учета количества общих интересов с пользователем.

Задачи данной работы:

- Исследование существующих решений по рекомендациям сообществ пользователям ВКонтакте;
- Реализация приложения, позволяющего пользователям получать список рекомендуемых к просмотру сообществ ВКонтакте.

Обзор предметной области

Перед созданием приложения был произведен поиск аналогичных инструментов для построения рекомендаций сообществ популярных социальных сетей, но подобных решений не нашлось. По завершению работы были рассмотрены научные статьи, которые так же исследовали рекомендательные алгоритмы для социальных сетей.

Сравнение аналогов

РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ДАННЫХ ИЗ ПРОФИЛЯ СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ» [1]

В статье рассматривается проблема автоматизации процесса веб-сёрфинга и фильтрации контента, а также, описано проектирование и реализация мультиагентной рекомендательной системы «EZSurf», обеспечивающей анализ интересов и предоставление рекомендаций пользователям социальной сети «ВКонтакте» на основе данных из профиля конкретного пользователя. Спроектированная авторами статьи рекомендательная система, реализующая метод фильтрации содержимого (content-based), использующая профиль пользователя социальной сети «ВКонтакте».

РЕКОМЕНДАЦИИ ТРЕКОВ В СОЦИАЛЬНЫХ СЕТЯХ [2]

В этой работе исследована применимость алгоритма Random Walk with Restarts в крупных социальных сетях для рекомендации музыкальных треков. Предложены модификации оригинального алгоритма, позволяющие улучшить качество рекомендаций. Предложен способ составления рекомендации в режиме реального времени по запросу пользователя. Так как оригинальный алгоритм при работе использует структуры данных, которые приобретают колоссальные размеры в реальных социальных сетях, автором был представлен «облегченный» вариант алгоритма, описаны качество и время его работы.

АЛЬТЕРНАТИВНАЯ РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА ДЛЯ ПОЛЬЗОВАТЕЛЕЙ STEAM [3]

Статья посвящена разработке рекомендательной системы для пользователей интернет-сервиса Steam, специализирующемся на продаже компьютерных игр. Разрабатываемая система использует гибридную модель фильтраций, состоящую из «collaboration filtration» (коллаборативная фильтрация), а также «content-based» (фильтрация, основанная на контенте). Такие системы направлены на то, чтобы компенсировать недостатки одного подхода внедрением другого.

Критерии сравнения аналогов

Доступность приложения

Основной идеей было создание простого рабочего приложения: любой пользователь сможет без проблем с ним взаимодействовать.

Скорость построения рекомендаций

Предполагается, что для пользователя имеет значение время, требуемое для решения поставленной задачи. Это говорит о том, что одной из главных целей должна быть быстрота работы алгоритмов рекомендаций.

Простота алгоритма

Будем считать, что для поставленной задачи необходимо найти наиболее оптимальный вариант алгоритма. Под простотой подразумевается низкая сложность вычислений, производимых данным алгоритмом.

Таблица сравнения по критериям

Название статьи	Доступность приложениям	Скорость построения рекомендаций	Простота алгоритма
Построение рекомендаций сообществ социальной сети «ВКонтакте»	+	Алгоритм срабатывает за несколько секунд, но загрузка данных занимает несколько минут.	Примитивный алгоритм. Основная сложность заключается в запросах к базе Neo4j.
РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ДАННЫХ ИЗ ПРОФИЛЯ СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ»	-	-	Сложный мультиагентный механизм
РЕКОМЕНДАЦИИ ТРЕКОВ В СОЦИАЛЬНЫХ СЕТЯХ	-	Требует значительных временных ресурсов	Алгоритм несет в себе нетривиальные расчеты
АЛЬТЕРНАТИВНАЯ РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА ДЛЯ ПОЛЬЗОВАТЕЛЕЙ STEAM	-	-	Алгоритм несет в себе нетривиальные расчеты

Выводы по итогам сравнения

Рассмотренные нами рекомендательные системы используют более сложные алгоритмы, из чего можно предположить, что время работы этих алгоритмов значительно выше значений, полученных нами (в одной из статей автор указывает на то, что алгоритм не справляется за приемлемое для пользователя время). Надо отметить, что не было найдено аналогов, связанных с построением рекомендаций именно сообществ сети ВКонтакте. Это говорит о том, что созданное нами решение потенциально найдет отклик среди пользователей социальной сети "ВКонтакте".

Обзор подходов и выбор метода решения

Существует множество подходов для поиска рекомендаций, самыми популярными из которых являются коллаборативная фильтрация, content - based и гибридный подход.

- Коллаборативная фильтрация - самый популярный вид рекомендательных систем на данный день. Системы коллаборативной фильтрации – это модели, которые пытаются предсказать, насколько вам понравится тот или иной продукт, получая на вход данные о том, как вы и другие пользователи оценивали этот и другие продукты в прошлом. Хотя такие системы достаточно эффективны, их точность сильно зависит от того, на сколько пересекаются оцениваемые объекты между отдельными пользователями. Чем больше и чаще эти пересечения, тем точнее будет система. Этот факт ограничивает сферу применения данной модели; если каждый пользователь будет свою уникальную информацию, которую никто другой не видел, коллективные рекомендательные системы будут бессильны. Также, для одним из основных недостатков коллаборативной фильтрации является проблема холодного старта: такая система не может построить рекомендации для нового пользователя, так как еще ничего про него не знает, аналогично и с новыми продуктами - они никому не будут рекомендоваться, пока не накопится информация о них.
- Content - based - это система, которая ищет объекты,похожие на те, что пользователь уже положительно оценил. Описываемый подход чаще всего используется для текстов:

документов, сайтов, блогов; или объектов, которые можно описать ключевыми словами. В рекомендательных системах контентного типа полезность рекомендуемых объектов находится из оценок пользователя, данных схожим объектам. Например, для того чтобы посоветовать человеку документ, контентная система пытается найти сходство между различными объектами, которые ранее получили у пользователя высокую оценку (сходство может заключаться в тематике, авторе и даже в похожих названиях глав в документе).

- Демографический подход - сравнивает характеристики объекта с характеристиками пользователя. Рассматриваться могут простые дискретные свойства: возраст, пол, место проживания, национальность так и более сложные, например, интерес пользователя к определенному объекту. Некоторые из них система может легко определить самостоятельно: страну и даже город, в котором проживает пользователь, можно определить по IP, предпочитаемый язык можно узнать из свойств браузера используя JavaScript. Другие данные можно попросить пользователя указать явно. Несмотря на то, что объем полученной информации о пользователе будет достаточно ограничен, она может помочь принять важные решения, например, кому рекомендовать куклу, а кому видео для взрослых.
- Статистический подход - это системы, которые основываются на статистических данных, собранных с пользователей. Проще говоря, это привычные нам рейтинги самых-самых: самые скачиваемые, самые читаемые, самые популярные и тп. Данный подход решает проблему холодного старта.
- Ассоциативный подход - строит рекомендации на основе данных о том, какие объекты используются вместе. Яркий пример применения таких систем — это анализ покупок пользователей. Например, кто-то купил себе новый телефон и, вполне вероятно, ему понадобятся к нему наушники, чехол, зарядка, дополнительная карта памяти и прочие аксессуары. Прочие рекомендательные системы здесь могут быть бессильны, ведь у этих объектов нет никаких общих параметров и сравнить их не представляется возможным, их объединяет лишь то, что они используются вместе.
- Гибридный подход объединяет в себе основные принципы сразу нескольких подходов. Концепция данного подхода заключается в том, что разные системы могут эффективно взаимодействовать, тем самым компенсируя недостатки друг друга.

Многие из алгоритмов, используемых рекомендательными системами, пришли из области машинного обучения, которая занимается алгоритмами для обучения, прогнозирования и принятия решений. Эти алгоритмы реализуют нетривиальные математические вычисления, так как в их задачу входит не только построение списка рекомендаций, но и масштабируемость, анализ поведения пользователей и предсказание их дальнейших действий.

В рамках выполнения данной работы решение вышеперечисленных задач избыточно, поэтому использование данных алгоритмов нецелесообразно.

В качестве базы данных была выбрана нереляционная база данных Neo4j. Вся логика построения рекомендаций реализована через запросы к базе. Такой подход позволяет максимально облегчить алгоритм и ускорить его.

По принципу построения рекомендаций данный способ имеет общие черты, как со статистической фильтрацией, так и с коллаборативной.

Описание решения и используемые технологии

Приложение разработано с использованием технологии Spring Boot. Интерфейс пользователя реализован с помощью фреймворка Angular 5.

Для работы приложения необходимо загрузить данные. Загрузка данных в программу происходит автоматически после авторизации пользователя с помощью профиля ВКонтакте. Авторизация реализована с помощью средств Spring Security и протокола OAuth. Используя VK

API, приложение получает информацию об авторизованном пользователе и его друзьях, а также о сообществах, в которых они состоят. Стоит отметить, что также было разработано API для импорта данных вручную. После загрузки данные сохраняются в базу данных Neo4j.

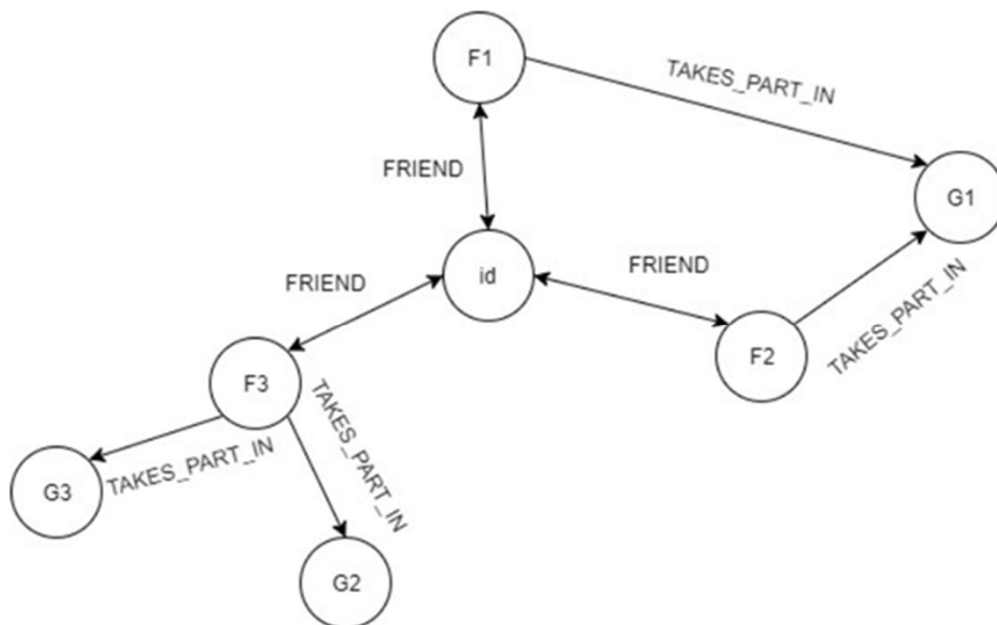


рис. 1 Графическое представление NoSQL модели данных

Вершины графа бывают двух типов: Person и Group. Корневая вершина имеет тип Person и хранит идентификатор пользователя, остальные вершины типа Person хранят идентификаторы его друзей, вершины – листья имеют тип Group и хранят идентификатор сообщества. Вершины имеют следующие свойства:

- id(int);
- name(String);
- vkId(int);
- avatar(String).

Ребра между корнем и Person – вершинами создают связи FRIEND (связь пользователя с другом). Ребра между Person – вершинами и Group – вершинами создают связи TAKES_PART_IN (связь человека с сообществом). Каждый режим построения алгоритмов реализуется посредством одного запроса к базе данных. Код запросов приведен в приложении.

Рекомендательная система работает в двух режимах:

Первый режим забирает из базы данных все Group – вершины, которые не имеют связей с корневой вершиной, и ранжирует их в порядке убывания количества связей типа TAKE_PART_IN. Иными словами, рассматриваются сообщества, в которых состоят друзья пользователя, но не состоит сам пользователь, составляется рейтинг по количеству вхождений в сообщество друзей.

Второй режим сначала анализирует назначает каждой некорневой Person - вершине вес: чем больше количество Group - вершин, с которыми и корневая вершина, и рассматриваемая Person - вершина имеют связи типа TAKE_PART_IN, тем больше вес. После назначения весов повторяются действия, описанные в пункте 1, но ранжирование происходит в порядке убывания суммы всех весов Person - вершин, которые имеют с рассматриваемой Group - вершиной связь типа TAKE_PART_IN. Проще говоря, рассматриваются сообщества, в которых состоят друзья пользователя, но не состоит сам пользователь, составляется рейтинг по количеству вхождений в сообщество друзей, но с учетом информации о том, сколько общих сообществ у пользователя с

другом. Чем больше общих сообществ у пользователя и друга, тем больший вес в рейтинге имеют сообщества, в которых состоит этот друг.

После завершения поиска рекомендаций полученная информация выводится на интерфейс пользователя в удобном для восприятия виде.

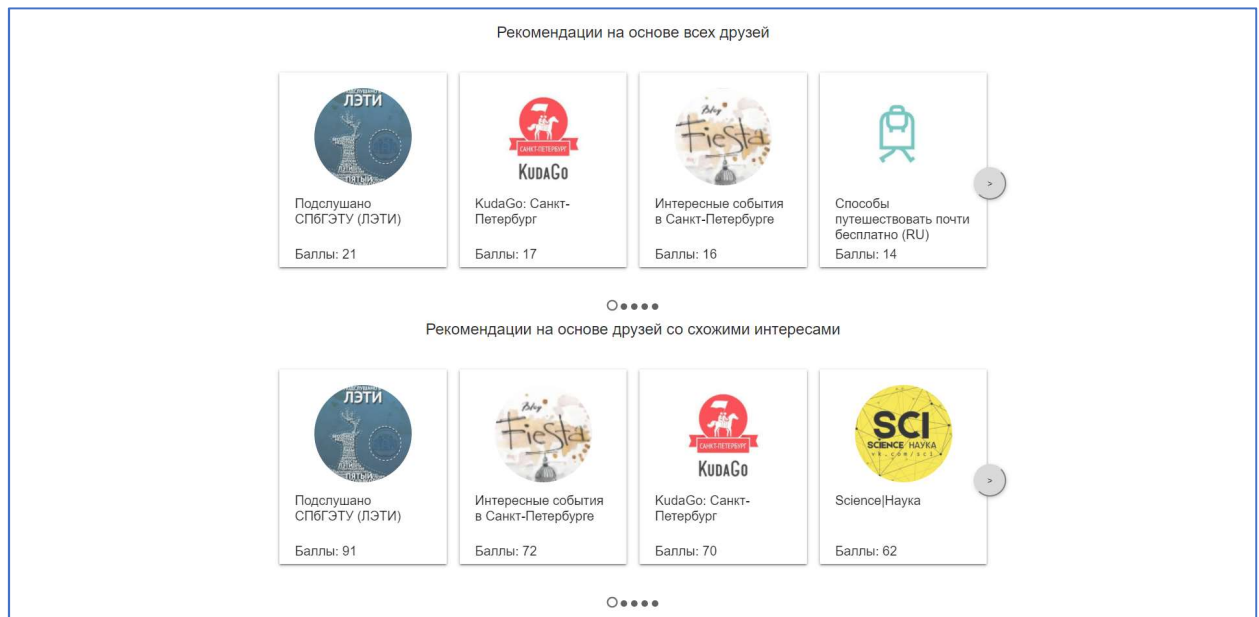


рис. 2 Интерфейс пользователя

Главным минусом данного решения является время, требуемое на загрузку данных в базу: при больших объемах данных приложение не сможет отработать за приемлемое для пользователя время, несмотря на то, что сам алгоритм построения рекомендаций занимает всего несколько секунд. Для пользователя с небольшим количеством друзей время работы программы в целом занимает от полутора до трех минут. Под небольшим количеством подразумеваем до 200 человек.

Исследование метода решения

Было произведено исследование аналогов выбранной модели данных для SQL СУБД, в ходе которого было выяснено, что для решения поставленной задачи больше подходит NoSQL модель.

Данный вывод был сделан по следующим причинам:

- NoSQL модель занимает меньше ресурсов памяти по сравнению с SQL моделью. Neo4j и SQL модель не подразумевают коллекций, однако если говорить о коллекции как о наборе однородных данных, то стоит отметить, что в нашей NoSQL модели различается 2 набора: множество вершин графа и множество ребер, в то время как SQL модель содержит 4 набора: таблицы Person, Friend, Group, Take_Part_In.
- SQL-запросы были бы более громоздкими и сложными, чем NoSQL-запросы.

Таблица «Person» содержит информацию о пользователе и его друзьях:

- personId (int);
- name (String);
- vkId (int);
- avatar (String).

Таблица Group содержит информацию о сообществах, в которых состоят пользователь и/или его друзья:

- groupId (int);
- name (String);
- vkId (int);
- avatar (String).

Таблица Friend описывает связи между пользователем и его друзьями:

- personId (int);
- userId (int).

Таблица Takes_Part_In описывает связи между людьми и сообществами:

- personId (int);
- groupId (int).

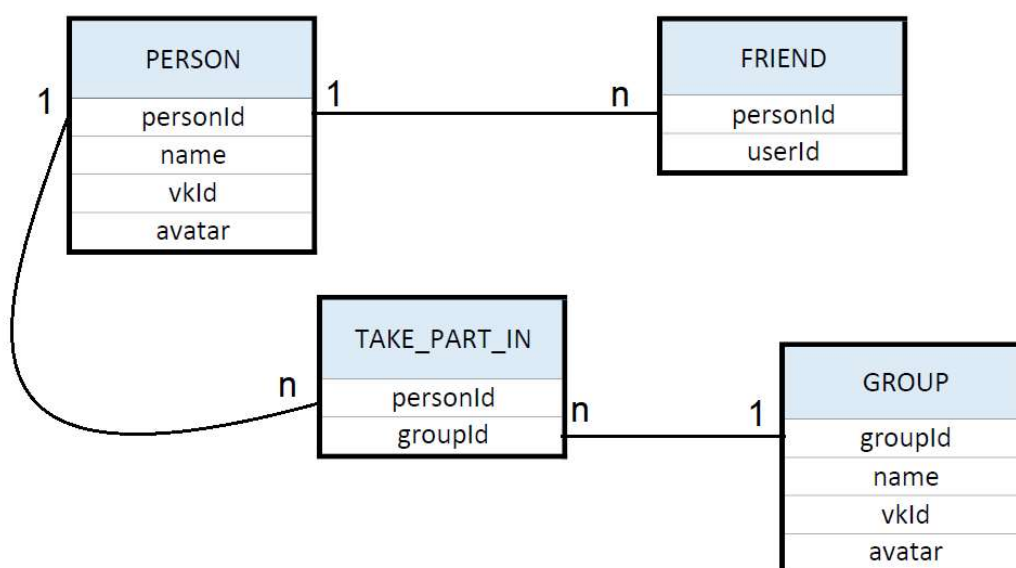


рис. 3 Графическое представление SQL модели данных

Заключение

В рамках создания рекомендательной системы для социальной сети "ВКонтакте" были рассмотрены наиболее распространённые подходы для поиска рекомендаций. Было выяснено, что большинство рассмотренных алгоритмов используют сложные математические вычисления и решают ряд побочных задач, что говорит о нецелесообразности их использования для решения поставленных задач.

Разработанное приложение успешно решает задачу построения рекомендаций сообществ для пользователей социальной сети "ВКонтакте". Используемый в рекомендательной системе алгоритм является оптимальным, так как при своей простоте реализации он гарантирует минимальные временные затраты. В дальнейшем планируется оптимизировать загрузку данных из социальной сети "ВКонтакте", чтобы максимально сократить время работы программы.

Приложение

Запросы к базе данных:

- Рекомендации на основе количества друзей:

MATCH (g:Group), (g)-[r]-(p2:Person), (p:Person) WHERE NOT (g)-[:TAKES_PART_IN]-(p) AND p.vkId = {0} RETURN g as group, COLLECT(p2) as friends, COUNT(r) AS rating ORDER BY rating DESC LIMIT {1};

Рекомендации на основе общих интересов с друзьями:

MATCH (g:Group), (g)-[r]-(p3:Person), (p:Person) WHERE NOT (g)-[:TAKES_PART_IN]-(p) and p.vkId = {0} WITH g, p3 MATCH (p1:Person)-[:TAKES_PART_IN]-(g1:Group)-[:TAKES_PART_IN]-(p3) WHERE p1.vkId = {0} " + "WITH g, p3, COUNT(g1) as weight RETURN g as group, COLLECT(p3) as friends, SUM(weight) as rating ORDER BY rating DESC LIMIT {1};

- Ссылка на готовое приложение:

https://github.com/moevm/nosql-2017-social_network_recomendations

СПИСОК ЛИТЕРАТУРЫ

1. АВХАДЕЕВ БУЛАТ РИНАТОВИЧ, ВОРОНОВА ЛИЛИЯ ИВАНОВНА, ОХАПКИНА ЕЛЕНА ПАВЛОВНА. (2014). РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ДАННЫХ ИЗ ПРОФИЛЯ СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ», Вестник Нижневартковского Государственного Университета №3 стр. 68-76
2. А.А. Дзюба. (2012)РЕКОМЕНДАЦИИ ТРЕКОВ В СОЦИАЛЬНЫХ СЕТЯХ. Магистерская работа. СПБГУ
3. ГАЖА КОНСТАНТИН ВЛАДИМИРОВИЧ, КАЛМЫКОВ ВЛАДИСЛАВ ВЯЧЕСЛАВОВИЧ, МИТЯНИНА АНАСТАСИЯ ВЛАДИМИРОВНА. (2017) АЛЬТЕРНАТИВНАЯ РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА ДЛЯ ПОЛЬЗОВАТЕЛЕЙ STEAM. Научная дискуссия №1/2017 стр. 78-93
4. IBM// URL <https://www.ibm.com/developerworks/ru/library/os-recommender1/index.html> (дата обращения 08.12.2017)
5. ЛОМАШ ДМИТРИЙ АЛЕКСЕЕВИЧ, ХЛОПИН КОНСТАНТИН ВИКТОРОВИЧ (2013). Реализация рекомендательных систем на основе алгоритмов коллаборативной и контекстной фильтрации. Вестник РГУПС №1/2013 стр. 75-84
6. Хабрахабр // habrahabr URL: <https://habrahabr.ru/post/58309/> (дата обращения 08.12.2017)

O.S. Samokhvalova, K.D. Uliashev, D.E. Popova