**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(национальный исследовательский университет)»

**Институт (Филиал)** № 8 «Компьютерные науки и прикладная математика» **Кафедра**  806

**Группа** М8О-410Б-19  **Направление подготовки** 02.03.02 Фундаментальная информатика и информационные технологии

**Профиль**  Информатика и компьютерные науки

**Квалификация**  **бакалавр**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**БАКАЛАВРА**

На тему: Разработка рекомендательной системы на базе решения задачи стохастической оптимизации с использованием базы данных музея

Автор ВКРБ\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шекунов Михаил Андреевич (\_\_\_\_\_\_\_\_\_\_)

(фамилия, имя, отчество полностью)

Руководитель Лемтюжникова Дарья Владимировна (\_\_\_\_\_\_\_\_\_\_)

(фамилия, имя, отчество полностью)

Консультант (\_\_\_\_\_\_\_\_\_\_)

(фамилия, имя, отчество полностью)

Консультант (\_\_\_\_\_\_\_\_\_\_)

(фамилия, имя, отчество полностью)

Рецензент (\_\_\_\_\_\_\_\_\_\_)

(фамилия, имя, отчество полностью)

**К защите допустить**

Заведующий кафедрой 806 Крылов Сергей Сергеевич (\_\_\_\_\_\_\_\_\_)

(№ каф) (фамилия, имя, отчество полностью)

\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Москва 2023

**РЕФЕРАТ**

Выпускная квалификационная работа бакалавра состоит из 00 страниц, 00 рисунков, 00 таблиц, 00 использованных источников, 00 приложений.

Текст реферата должен отражать:

* + объект исследования или разработки;
  + цель работы;
  + методы или методологию проведения работы;­­­
  + результаты работы и их новизну;
  + область применения результатов;
  + рекомендации по внедрению или итоги внедрения результатов;
  + экономическую эффективность или значимость работы;
  + прогнозные предположения о развитии объекта исследования.

Если отчет не содержит сведений по какой-либо из перечисленных структурных частей реферата, то в тексте реферата она опускается, при этом последовательность изложения сохраняется.

Оптимальный объем текста реферата - 850 печатных знаков, но не более одной страницы машинописного текста.

В общее количество таблиц и рисунков входят и таблицы, и рисунки приложений.

СОДЕРЖАНИЕ

[ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 4](#_Toc133908286)

[ВВЕДЕНИЕ 5](#_Toc133908287)

[1 ТЕОРИЯ 11](#_Toc133908288)

[1.1 Потребности в разработке гибридной рекомендательной системы на базе стохастической оптимизации для культурных учреждений 11](#_Toc133908289)

[1.2. Техническое оснащение и процесс сбора данных о посещении объектов в музее 12](#_Toc133908290)

[1.3 Анализ технических возможностей для реализации алгоритмов 13](#_Toc133908291)

[1.4 Анализ базы данных музея 14](#_Toc133908292)

[1.5 Поиск существующих рекомендательных систем 14](#_Toc133908293)

[1.6 Сопоставление имеющихся данных и существующих алгоритмов 15](#_Toc133908294)

[1.7 Коллаборативная фильтрация 17](#_Toc133908295)

[1.8 Система на основе схожести объектов 20](#_Toc133908296)

[1.9 Рекомендательная система, основанная на расстояниях между объектами 22](#_Toc133908297)

[1.10 Гибридная рекомендательная система 23](#_Toc133908298)

[1.11 Слияние результатов гибридной системы с результатами стохастической оптимизации 24](#_Toc133908299)

[2 ПРАКТИЧЕСКАЯ ЧАСТЬ 26](#_Toc133908300)

[2.1 Используемый язык программирования 26](#_Toc133908301)

[2.2 Используемые библиотеки и модули 27](#_Toc133908302)

[2.3 Реализация коллаборативной фильтрации 28](#_Toc133908303)

[2.4 Реализация рекомендательной системы, основанной на схожести объектов 31](#_Toc133908304)

[При реализации рекомендательной системы, основанной на схожести объектов необходимо выбрать меру близости для векторов информации об объектах. В данной ВКР была использована косинусная мера сходства. 31](#_Toc133908305)

[2.5 Реализация рекомендательной системы на основе расстояний 33](#_Toc133908306)

# **ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ**

В настоящей выпускной квалификационной работе бакалавра применяют следующие термины с соответствующими определениями:

Коллаборативная фильтрация — ­это один из методов построения прогнозов (рекомендаций) в рекомендательных системах, использующий известные предпочтения (оценки) группы пользователей для прогнозирования неизвестных предпочтений другого пользователя.

Стохастическая оптимизация — это метод оптимизации, используемый для решения задач, когда функционал, который нужно оптимизировать, не может быть вычислен точно или слишком сложен для аналитического решения.

Датафрейм — это структура данных, используемая в библиотеке pandas для работы с табличными данными. Она представляет собой двумерную таблицу с метками строк и столбцов, где каждый столбец может содержать данные различных типов, например, числа, строки, булевы значения и т.д. DataFrame позволяет быстро и удобно обрабатывать и анализировать большие объемы данных, выполнять фильтрацию, сортировку, группировку, агрегацию, а также применять различные математические и статистические операции к данным.

# **ВВЕДЕНИЕ**

Актуальность темы данной работы связана с тем, что в настоящее время музеи играют важную роль в культурной жизни общества и привлекают множество посетителей, которые хотят ознакомиться с экспонатами, историей и культурой различных народов. Однако, посетители часто сталкиваются с проблемой ограниченного времени, что может вызвать у них чувство неудовлетворенности и привести к негативному опыту посещения музея.

Для решения данной проблемы, музеи начали применять технологии и методы, которые могут помочь посетителям более эффективно планировать свое время и выбирать интересующие экспонаты. В этом контексте, разработка рекомендательных систем становится все более актуальной и востребованной задачей, поскольку они могут помочь посетителям выбирать наиболее интересующие экспонаты и оптимизировать свое время.

Однако, существующие рекомендательные системы могут не учитывать все предпочтения посетителей и не предоставлять наиболее оптимальные рекомендации. В этой связи, разработка новых методов и алгоритмов рекомендательных систем с использованием современных методов стохастической оптимизации является важной задачей, которая может улучшить качество обслуживания посетителей и повысить удовлетворенность ими.

Разработка рекомендательной системы на базе решения задачи стохастической оптимизации для музея имеет высокую актуальность в контексте улучшения качества обслуживания посетителей и оптимизации использования времени на посещение экспонатов музея.

Таким образом, выполненная работа актуальна и с научно-методической/теоретической, и с практической точек зрения.

Цель данной работы - разработать рекомендательную систему на базе решения задачи стохастической оптимизации с использованием базы данных музея, которая будет предоставлять посетителям музея персонализированные рекомендации на основе их предпочтений и интересов. Рекомендательная система будет разработана на заказ винного музея во Франции и будет использоваться для помощи посетителям в выборе наиболее интересующих экспонатов, оптимизации использования времени на посещение музея и повышения удовлетворенности посетителей.

Для достижения поставленной цели в работе были решены следующие задачи:

1. изучение существующих методов и алгоритмов рекомендательных систем и выбор оптимальных для решения задачи,
2. преобразование базы данных под рекомендательную систему,
3. реализация нескольких типов рекомендательных систем,
4. разработка объединения результатов работы нескольких алгоритмов рекомендательных систем,
5. разработка компонентов рекомендательной системы, с помощью которых учитываются результаты решения задачи стохастического программирования по определению маршрута движения туриста с учетом различных сценариев,
6. реализация и тестирование рекомендательной системы, результатом которой является близкий к оптимальному маршрут движения туриста.

Для выполнения работы были использованы различные инструменты и методы, включая:

1. Язык программирования Python. Python является одним из наиболее распространенных языков программирования, который широко используется для научных вычислений, машинного обучения и анализа данных. В данной работе Python использовался для разработки алгоритмов и реализации рекомендательной системы.
2. Файлы CSV: Формат CSV (Comma Separated Values) является одним из наиболее распространенных форматов хранения и обмена данными. В данной работе файлы CSV использовались для хранения и обработки данных, связанных с экспонатами, посетителями музея, результатов работы рекомендательных систем.
3. Библиотека SciPy: SciPy — это библиотека для научных вычислений в Python, которая содержит множество инструментов для решения задач оптимизации, линейной алгебры, статистического анализа и другого. В данной работе библиотека SciPy использовалась для реализации алгоритмов стохастической оптимизации.
4. Иные библиотеки Python для работы с данными. В данной работе использовались различные библиотеки Python для работы с данными, такие как библиотека os для работы с файловой системой, библиотека Pandas для анализа данных, библиотека NumPy для научных вычислений и так далее.

В рамках данной работы были получены следующие основные результаты:

1. Отсортированные рекомендации с баллами для трех алгоритмов рекомендательных систем:
2. Коллаборативная фильтрация: Для каждого посетителя музея на основе его предыдущих посещений формируется список рекомендаций, который содержит объекты, наиболее похожие на те, которые посетитель уже посмотрел. После этого список рекомендаций сортируется по баллам, которые рассчитываются на основе сходства между объектами и предпочтений посетителя.
3. Рекомендательная система, основанная на схожести объектов: Для каждого посетителя музея на основе его предыдущих посещений формируется список рекомендаций, который содержит объекты, наиболее похожие на те, которые посетитель уже посмотрел. После этого список рекомендаций сортируется по баллам, которые рассчитываются на основе сходства между объектами.
4. Рекомендательная система, основанная на расстояниях: Для каждого посетителя музея на основе его предыдущих посещений формируется список рекомендаций, который содержит объекты, находящиеся ближе всего к тем, которые посетитель уже посмотрел. После этого список рекомендаций сортируется по баллам, которые рассчитываются на основе расстояний между объектами.
5. Отсортированный результат слияния рекомендательных систем по баллам. Для каждого посетителя музея получены списки рекомендаций на основе трех различных алгоритмов рекомендательных систем. После этого списки рекомендаций объединены и отсортированы по баллам, полученным на основе результатов трех алгоритмов.
6. Итоговый отсортированный список рекомендаций: Для каждого посетителя музея формируется индивидуальный список рекомендаций, который учитывает результат слияния трех алгоритмов рекомендательных систем и работу эвристического алгоритма стохастической оптимизации. Список рекомендаций отсортирован по баллам, которые рассчитываются на основе сходства между объектами и предпочтений посетителя, а также результатов работы стохастической оптимизации. Кроме того, данные для каждого посетителя формируются индивидуально после посещения нового объекта, что позволяет рекомендовать экспонаты, наиболее соответствующие интересам и предпочтениям каждого конкретного посетителя.

Все результаты были записаны в файлы CSV, которые могут быть использованы для последующего анализа и улучшения алгоритмов рекомендательной системы.

Результаты работы предназначены для внедрения в работу музея с целью улучшения качества обслуживания посетителей и оптимизации времени, затрачиваемого на просмотр экспонатов. Разработанный алгоритм рекомендации может быть использован музеем для предоставления индивидуальных рекомендаций посетителям, исходя из их предпочтений и интересов. Полученные результаты могут быть использованы для улучшения качества обслуживания и увеличения удовлетворенности посетителей, что, в свою очередь, может привести к увеличению посещаемости музея.

Кроме того, результаты работы могут быть использованы для дальнейших исследований в области рекомендательных систем и оптимизации процессов в музеях. Полученные результаты могут послужить основой для разработки более эффективных алгоритмов рекомендации и оптимизации процессов в музеях, что в свою очередь может привести к повышению качества обслуживания и удовлетворенности посетителей.

Также результаты работы могут быть использованы для интеграции с другими системами управления музеем, такими как системы бронирования билетов, системы управления посетителями и системы анализа посещаемости. Это может помочь музею снизить затраты на управление и повысить эффективность работы, что в итоге может привести к улучшению качества обслуживания посетителей и удовлетворенности их потребностей.

Использование разработанного алгоритма рекомендации на базе решения задачи стохастической оптимизации с использованием базы данных музея позволяет музею сократить время, затрачиваемое посетителями на поиск и выбор интересных для них экспонатов. Благодаря предоставлению индивидуальных рекомендаций на основе предпочтений и интересов посетителей, музей может увеличить удовлетворенность посетителей и улучшить качество обслуживания. Посетители будут получать персонализированные рекомендации, которые помогут им сэкономить время и сделать посещение музея более приятным и интересным.

Кроме того, использование разработанной рекомендательной системы может привести к повышению посещаемости музея, так как посетители будут получать рекомендации о тех экспонатах, которые наиболее соответствуют их интересам, что может заинтересовать их посетить музей снова. Это может привести к увеличению прибыли музея и повышению его популярности среди посетителей.

Также использование разработанной рекомендательной системы может помочь музею сократить время на обслуживание посетителей и уменьшить нагрузку на персонал музея. Благодаря индивидуальным рекомендациям, посетители смогут быстрее найти интересные для них экспонаты и не будут требовать помощи персонала музея в выборе экспонатов.

В целом, использование разработанной рекомендательной системы позволяет музею улучшить качество обслуживания посетителей, повысить эффективность работы и повысить прибыль.

# **1 ТЕОРИЯ**

## 1.1 Потребности в разработке гибридной рекомендательной системы на базе стохастической оптимизации для культурных учреждений

В современном мире наблюдается быстрый рост информации, что создает огромные проблемы для пользователей в выборе наиболее подходящей информации. Одним из решений этой проблемы являются рекомендательные системы, которые предоставляют пользователям персонализированные рекомендации на основе их предпочтений и интересов. Рекомендательные системы широко используются в различных областях, включая электронную коммерцию, социальные сети, видео- и музыкальные сервисы и т.д. Однако, разработка рекомендательных систем, которые бы удовлетворяли требованиям пользователей, остается актуальной и сложной задачей.

В сфере культуры и искусства рекомендательные системы также играют важную роль. Музеи, театры и другие культурные учреждения сталкиваются с проблемой обеспечения качественного обслуживания посетителей и увеличения их удовлетворенности. Рекомендательные системы могут помочь решить эту проблему, предоставляя посетителям персонализированные рекомендации посещения экспонатов или мероприятий на основе их интересов и предпочтений.

Однако, существующие рекомендательные системы не всегда удовлетворяют потребности пользователей. Например, некоторые системы не учитывают индивидуальные потребности и предпочтения пользователей, а другие могут оказаться неэффективными в силу недостаточной точности предсказания рекомендаций.

Именно поэтому разработка новых методов и алгоритмов для создания эффективных рекомендательных систем остается актуальной и важной задачей. В рамках данной работы рассматривается разработка гибридной рекомендательной системы на базе решения задачи стохастической оптимизации с использованием базы данных музея. Она позволит музею предоставлять посетителям персонализированные рекомендации на основе их интересов и предпочтений, что, в свою очередь, улучшит качество обслуживания и повысит удовлетворенность.

## 1.2. Техническое оснащение и процесс сбора данных о посещении объектов в музее

В музее, из которого был заказ, используются планшетники для предоставления посетителям дополнительной информации об экспонатах. Для получения информации посетитель должен отсканировать QR-код, который находится на каждом экспонате, с помощью камеры планшетника. После сканирования QR-кода на экране планшета появляется информация об объекте, такая как его название, дата создания, описание, автор и т.д. Процесс сканирования QR-кода на планшете для посетителей музея довольно простой. Для того, чтобы получить дополнительную информацию об экспонате, посетитель должен с помощью камеры планшета отсканировать QR-код, который находится на каждом объекте. После этого на экране планшета появляется информация об объекте, которую посетитель может изучать.

Однако, для разработчиков системы, использование QR-кодов также предоставляет возможность собирать данные о поведении посетителей. Система может отслеживать, какие объекты были посещены каждым посетителем, как долго они останавливались на каждом объекте и когда они переходили к следующему. Эти данные могут быть использованы для анализа интересов посетителей, и для того, чтобы оптимизировать предложения рекомендаций на основе этого поведения.

Кроме того, система также может собирать данные о посещенных объектах, которые могут быть использованы музеем для анализа посещаемости и популярности различных экспонатов. Эта информация может быть полезна для планирования выставок, реконструкции залов и других мероприятий, связанных с улучшением посещаемости музея.

Благодаря простоте процесса сканирования QR-кодов на планшетах в музее, нам не нужно просить пользователей оставлять оценки или комментарии об объектах, что упрощает взаимодействие с системой и делает процесс более естественным. Тем не менее, во время сканирования мы можем записывать время посещения каждого объекта, и использовать эту информацию в качестве оценки интереса посетителя к конкретному экспонату. Это может быть полезно при разработке рекомендательной системы, которая будет учитывать предпочтения и интересы каждого посетителя, а также оптимизировать их посещение музея. В свою очередь, такая система может помочь музею повысить качество обслуживания, улучшить удовлетворенность посетителей и повысить посещаемость.

Таким образом, использование QR-кодов в музее позволяет не только предоставлять посетителям дополнительную информацию об экспонатах, но и собирать данные о поведении посетителей, которые могут быть использованы для улучшения качества обслуживания и оптимизации рекомендаций.

## 1.3 Анализ технических возможностей для реализации алгоритмов

Для работы алгоритмов мы используем слабопроизводительные устройства, так как это планшеты, которые обычно имеют ограниченные ресурсы. В связи с этим, мы должны снизить количество вычислений, которые производятся на устройствах, и использовать максимально доступные данные для получения оценок пользователя. В данном случае мы используем время, проведенное пользователем у каждого объекта в качестве такой оценки.

Следует отметить, что музей является большим и может быть ограничен доступ к интернету или локальной сети. Это означает, что все вычисления для рекомендательной системы должны выполняться на планшетах. Однако мы должны обеспечивать при этом высокую точность рекомендаций.

## 1.4 Анализ базы данных музея

База данных музея содержит информацию о посетителях, которые используют планшеты для сканирования QR-кодов экспонатов и получения информации о них. Из этих данных мы извлекли информацию о времени посещения каждого экспоната каждым посетителем. Также в базе данных содержится информация об экспонатах, включая их название, категорию, описание и номер в базе данных. Мы использовали эту информацию для создания csv файлов с отобранными посетителями и экспонатами.

Кроме того, в базе данных содержится информация о расположении экспонатов в помещении, включая номер зала и номер экспоната, а также информация о категориях помещений, к которым принадлежат экспонаты.

## 1.5 Поиск существующих рекомендательных систем

Алгоритмы рекомендательных систем могут быть различными и, как правило, определяются конкретной задачей, которую необходимо решить. В литературе (<https://doi.org/10.1023/A:1021240730564>) существует множество алгоритмов, которые можно использовать для построения рекомендательных систем. Некоторые из них включают:

1. Content-based filtering (CBF) - основан на анализе характеристик продуктов или услуг, которые пользователь оценил или потребовал.
2. Collaborative filtering (CF) - основан на схожести пользователей, исходя из их предпочтений. Может быть двух типов: user-based и item-based.
3. Knowledge-based filtering - использует экспертную систему для генерации рекомендаций.
4. Hybrid recommender systems - комбинируют различные алгоритмы, такие как CBF и CF, чтобы улучшить качество рекомендаций.
5. Demographic-based filtering - использует информацию о демографических характеристиках пользователей, таких как возраст, пол, место жительства и т.д.
6. Knowledge-based filtering with demographic information - комбинирует информацию о демографии пользователей с экспертной системой для генерации рекомендаций.
7. Context-aware filtering - учитывает контекст, в котором пользователь просматривает или потребляет продукты, например, местоположение, время и т.д.
8. Social-based filtering - основан на анализе социальных связей пользователей и используется для генерации рекомендаций, основанных на социальных связях.

## 1.6 Сопоставление имеющихся данных и существующих алгоритмов

При разработке рекомендательных систем, одним из важных аспектов является учет демографических характеристик пользователей. Эти данные могут включать в себя возраст, пол, местоположение и т.д. Однако, сбор такой информации может стать проблемой, особенно в контексте музеев, где посетители могут заходить на короткий срок и не захотеть тратить время на заполнение анкет или опросников. Дополнительно, вопросы о демографических характеристиках могут вызвать недовольство и неудовольствие у пользователей, особенно если они считают, что эти вопросы не имеют никакого отношения к их посещению музея. Из-за этих факторов, сбор демографических данных может стать непрактичным и неэффективным для использования в рекомендательных системах.

В связи с этим, мы решили использовать две системы, которые могут работать без дополнительных данных от пользователя: коллаборативную фильтрацию и рекомендательную систему, основанную на сходстве объектов.

Коллаборативная фильтрация основана на анализе предпочтений пользователей и сопоставлении их с предпочтениями других пользователей, чтобы предложить рекомендации.

Рекомендательная система, использующая сходство, основана на анализе характеристик объектов и сопоставлении их с предпочтениями пользователей.

Обе системы не требуют от пользователя дополнительных данных, поэтому возможно использование имеющиеся данные из базы данных музея.

Для использования коллаборативной фильтрации нужно определить, какие пользователи могут быть схожими по предпочтениям и на основе этого определить, какие экспонаты можно порекомендовать. Для основанной на сходстве рекомендательной системы нам нужно определить характеристики объектов, которые будут использоваться для сопоставления с предпочтениями пользователей.

База данных музея содержит информацию о посетителях и экспонатах, которые могут быть использованы для выбора рекомендательных систем. Были собраны данные о времени, проведенном посетителями у каждого экспоната, и использовали эту информацию для определения, какие экспонаты могут быть наиболее интересны для посетителей. Также была получена информация о характеристиках объектов и их расположении в помещениях, что позволяет определить, какие экспонаты могут быть связаны друг с другом и могут быть интересны для пользователей, основываясь на их времени пребывания в музее.

Поскольку мы имеем информацию о принадлежности экспонатов к определенным комнатам и их расположении внутри них, возможно использование этой информации для формирования рекомендаций, которые будут основываться на расстоянии между объектами. Поэтому дополнительно включаем в разработку рекомендательной системы алгоритм, который учитывает расстояния между объектами в музее. Таким образом, пользователи получат рекомендации по экспонатам, находящимся в близости друг от друга, что может быть особенно удобно для тех, кто хочет изучить определенную тему или эпоху.

Для учета расстояний между объектами можно использовать данные о фактическом расположении экспонатов в музее, которые были извлечены из базы данных музея. Таким образом, можно более точно определить, какие объекты находятся ближе друг к другу, и использовать эту информацию для формирования рекомендаций.

## 1.7 Коллаборативная фильтрация

Коллаборативная фильтрация - одна из наиболее распространенных рекомендательных систем, используемых для предсказания рейтингов или предпочтений пользователей. Работа этой системы основана на предположении, что если два пользователя оценили некоторый набор объектов (фильмов, музыки, товаров и т.д.) схожим образом, то они будут иметь схожие вкусы и интересы.

В подходе, анализирующем пользователей, сначала вычисляется сходство между всеми парами пользователей, используя меры сходства, такие как косинусное сходство или корреляция Пирсона. Затем, для каждого пользователя находятся похожие на него пользователи, и вычисляются взвешенные оценки объектов на основе их оценок и сходства с выбранными объектами. Эти оценки затем используются для формирования рекомендаций.

Коллаборативная фильтрация имеет ряд преимуществ, таких как возможность учитывать неявные предпочтения пользователей.

Для реализации коллаборативной системы необходимо выполнить несколько шагов. Сначала нужно получить данные о предпочтениях пользователей, то есть оценки, которые они выставляют объектам. Затем эти данные нужно преобразовать в матрицу оценок. Обычно матрица оценок имеет вид таблицы, где каждая строка соответствует пользователю, каждый столбец – объекту, а ячейка таблицы содержит оценку, которую пользователь выставил объекту. Схематично данная рекомендательная система и матрица оценок представлены на рисунке 1.

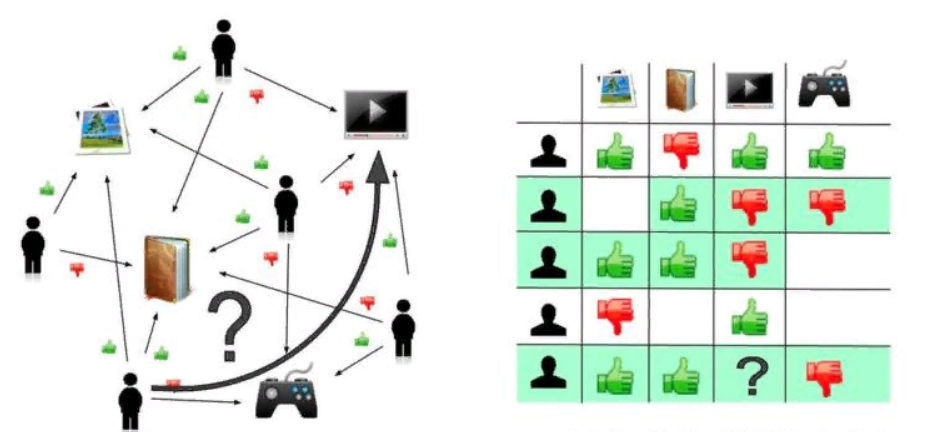


Рисунок 1 – Иллюстрация коллаборативной фильтрации

После этого можно перейти к этапу поиска похожих пользователей или объектов. Для этого используются различные алгоритмы, такие как метод k-ближайших соседей, сингулярное разложение и другие. Они позволяют определить наиболее похожие пользователей или объекты на основе сравнения их оценок.

Далее необходимо выполнить этап предсказания оценок. Для этого используются различные методы, такие как среднее значение оценок соседей или взвешенное среднее, где веса зависят от степени похожести соседей. На основе предсказанных оценок можно формировать список рекомендаций.

Важным моментом является оценка качества рекомендаций. Для этого используются различные метрики, такие как точность, полнота, F-мера и др. Также проводятся эксперименты на тестовых данных, чтобы оценить эффективность системы.

Одним из недостатков коллаборативной системы является холодный старт, то есть невозможность предложить рекомендации для новых пользователей или объектов. Также система может страдать от проблемы рекомендации известных объектов, то есть предлагать уже известные пользователю объекты.

Коллаборативная фильтрация эффективна, потому что она основана на анализе предпочтений пользователей и нахождении сходств между их предпочтениями. Это позволяет системе предлагать пользователю объекты, которые ему могут понравиться, на основе предпочтений их похожих пользователей.

Исследования показывают, что коллаборативная фильтрация является одной из самых эффективных и точных методов рекомендаций. Например, в статье "A Survey of Collaborative Filtering Techniques" (1997) <https://doi.org/10.1155/2009/421425> был проведен обзор и сравнение различных методов коллаборативной фильтрации, и результаты показали, что они превосходят другие методы рекомендаций по точности.

Также, исследование "Collaborative Filtering Recommender Systems" (2011) [Collaborative Filtering Recommender Systems | Now Foundations and Trends books | IEEE Xplore](https://ieeexplore.ieee.org/document/8187128/) выявило, что коллаборативная фильтрация имеет высокую точность и показывает хорошие результаты в рекомендациях в различных областях, включая музыку, фильмы, книги и т.д.

Из этих исследований следует, что коллаборативная фильтрация является эффективным методом рекомендаций, особенно в случае большого количества пользователей и объектов, что делает ее подходящим выбором для нашей задачи рекомендации экспонатов в музее.

## 1.8 Система на основе схожести объектов

Система рекомендаций на основе сходства объектов (item-based recommendation system) предполагает, что если пользователь понравился один объект, то ему могут понравиться и другие объекты, которые похожи на этот объект. Для построения такой системы сначала нужно создать матрицу сходства между объектами. Это можно сделать с помощью различных метрик сходства, таких как косинусное сходство или евклидово расстояние.

После того, как матрица сходства создана, для каждого объекта можно найти его похожие объекты. Например, если пользователь заинтересован в экспонате, который относится к разделу «История технологий», то ему могут быть интересны и другие экспонаты из этого же раздела. Для этого для каждого объекта находятся наиболее похожие на него объекты из того же раздела, после чего список объектов сортируется по убыванию степени сходства. Пользователю предлагаются наиболее похожие объекты из этого списка.

Основное преимущество системы рекомендаций на основе сходства объектов заключается в том, что она может предложить пользователю объекты, которые ему могут понравиться, но которые он сам бы не нашел. При этом не требуется большого количества информации о пользователе, достаточно знать, какой объект ему понравился, чтобы на основе сходства объектов подобрать другие, которые могут его заинтересовать.

Однако система рекомендаций на основе сходства объектов имеет и свои недостатки. В частности, она не учитывает индивидуальные предпочтения пользователя и его контекст, такой как время, место и настроение. Также может возникнуть проблема "холодного старта", когда новый объект, который еще не имеет достаточного количества оценок и связей с другими объектами, не будет учтен системой.

Для решения этих проблем, в систему можно добавить дополнительные факторы, такие как временные, географические или социальные данные. Это позволяет учитывать контекст и индивидуальные предпочтения пользователя, а также повышает качество рекомендаций.

Исследования показывают, что системы, основанные на сходстве объектов, могут быть эффективными в рекомендации объектов, которые могут заинтересовать пользователя. Например, исследование "Content-Based Recommender Systems: State of the Art and Trends" от 2014 года показало, что системы, основанные на сходстве объектов, могут быть эффективными в рекомендации музейных экспонатов, книг и фильмов.

Другое исследование "A Survey of Content-based Recommender Systems Based on Machine Learning Techniques" от 2016 года также подтвердило эффективность систем, основанных на сходстве объектов, и показало, что они могут быть эффективными в рекомендации объектов в различных областях, включая музеи и галереи.

Таким образом, системы, основанные на сходстве объектов, могут быть эффективными в рекомендации объектов, которые могут заинтересовать пользователя. Данная рекомендательная система проиллюстрирована на рисунке номер 2.



Рисунок 2 – Content-based рекомендательная система

## 1.9 Рекомендательная система, основанная на расстояниях между объектами

Рекомендательная система, основанная на расстояниях, использует информацию о том, как далеко находятся объекты друг от друга в пространстве музея, чтобы рекомендовать пользователю близлежащие экспонаты. Данная система позволяет более точно учитывать пространственную структуру музея, что может быть полезным в случаях, когда пользователь желает ознакомиться с объектами в конкретном порядке, или имеется маршрут, оптимизированный для посещения объектов в определенном порядке.

Основным алгоритмом данной системы является алгоритм k-ближайших соседей (k-nearest neighbors, KNN). Он основан на том, что объекты, расположенные близко друг к другу в пространстве музея, вероятно, будут интересны пользователю, если пользователь уже проявил интерес к некоторым объектам. В этом случае алгоритм выбирает k ближайших экспонатов к тем, которые уже были посещены пользователем, и предлагает их как рекомендации.

## 1.10 Гибридная рекомендательная система

Каждая из рекомендательных систем имеет свои сильные и слабые стороны. Коллаборативная фильтрация работает на основе схожести предпочтений пользователей, а рекомендация на основе сходства объектов использует сходство между объектами для выдачи рекомендаций. Рекомендательная система на основе расстояний между объектами использует расстояние между объектами для выдачи рекомендаций.

Одним из преимуществ объединения результатов рекомендательных систем является то, что это позволит снизить риск ошибок. Каждая из систем может допустить ошибку, что может привести к неверным рекомендациям. Однако объединение результатов позволит увеличить точность рекомендаций и снизить риск ошибок. Также это помогает нивелировать ошибки, вызванные недостаточным количеством данных или искажением результатов в одной из систем.

Кроме того, объединение результатов может увеличить разнообразие рекомендаций. Каждая из систем может выдавать рекомендации, которые схожи между собой. Однако, объединение результатов позволит увеличить разнообразие рекомендаций и предложить пользователю более широкий выбор.

Одним из главных плюсов такого подхода является уменьшение проблемы "холодного старта", то есть ситуации, когда рекомендательная система не может предложить рекомендации для нового посетителя, у которого еще нет истории посещений.

Объединение результатов также позволит учесть различные критерии рекомендаций, которые могут быть важны для разных пользователей. Например, один пользователь может предпочитать получать рекомендации на основе сходства объектов, а другой – на основе схожести предпочтений. Объединение результатов позволит учесть все эти критерии и выдать наиболее подходящие рекомендации для каждого пользователя.

Такая система может обеспечивать более точную рекомендацию для активных пользователей, так как учитывает их предпочтения, а также оценки, которые они давали объектам в прошлом.

В результате объединения в гибридную всех трех систем - коллаборативной фильтрации, системы на основе сходства объектов и системы на основе расстояний, мы получаем комплексную рекомендательную систему, которая учитывает разные аспекты предпочтений посетителей и повышает точность рекомендаций.

## 1.11 Слияние результатов гибридной системы с результатами стохастической оптимизации

Слияние результатов гибридной рекомендательной системы с эвристикой стохастической оптимизации является еще одним шагом в улучшении качества рекомендаций. Главным преимуществом использования стохастической оптимизации в качестве эвристики является то, что она может помочь в поиске оптимального решения, основываясь на статистических методах и случайных величинах.

Одним из главных вызовов при слиянии результатов гибридной системы с эвристикой является то, что стохастическая оптимизация не дает точных баллов для каждого объекта. Она работает с вероятностями и распределениями, что может быть не очень удобным для смешивания с другими рекомендательными системами, которые работают с точными оценками.

Чтобы решить эту проблему, мы можем использовать различные методы смешивания результатов, чтобы дать больший вес объектам, которые были лучше оценены в стохастической оптимизации. Мы можем также установить пороговое значение, при котором объекты с низкими оценками из стохастической оптимизации будут исключены из списка рекомендаций.

Важным моментом при слиянии результатов является также учет взаимодействия между пользователями и объектами. В случае стохастической оптимизации, каждый пользователь может получить свой собственный набор рекомендаций, основанный на его предпочтениях и интересах. При слиянии результатов, мы должны учитывать эти индивидуальные различия и попытаться дать каждому пользователю наиболее подходящие рекомендации.

# **2 ПРАКТИЧЕСКАЯ ЧАСТЬ**

**2.1 Используемый язык программирования**

Для создания рекомендательной системы был выбрал язык программирования Python, так как он является одним из наиболее подходящих языков программирования для решения данной задачи. Выбор языка программирования Python для данной работы обусловлен рядом его преимуществ, которые делают его наиболее подходящим инструментом для разработки рекомендательной системы на базе стохастической оптимизации.

Простота и интуитивность синтаксиса Python делает его доступным для использования в задачах программирования разного уровня сложности, что особенно важно в контексте данной работы.

Кросплатформенность Python позволяет использовать его на любой операционной системе, что облегчает работу с данными на различных устройствах. Так рекомендательная система не зависит от операционной системы планшета, на которой она будет запущена.

Наличие обширной коллекции библиотек и модулей Python, таких как NumPy, Pandas, Scikit-learn и os, делает его идеальным инструментом для работы с данными, машинного обучения и создания рекомендательных систем.

Поддержка функционального программирования Python делает код более простым и структурированным, что особенно важно для работы над сложными проектами.

Высокая скорость выполнения кода благодаря компиляции некоторых частей Python делает его подходящим выбором для задач, связанных с вычислительными процессами и обработкой больших объемов данных.

Интерактивная консоль Python позволяет быстро тестировать и отлаживать код, что ускоряет процесс разработки.

Простая и эффективная система управления пакетами Python, такая как pip, позволяет удобно устанавливать и управлять сторонними библиотеками, что также сильно упрощает разработку рекомендательной системы на базе стохастической оптимизации.

## 2.2 Используемые библиотеки и модули

Для успешной реализации рекомендательной системы были использованы различные библиотеки и инструменты. Краткое описание и обоснование использование библиотек приведено ниже.

Библиотека SciPy предоставляет множество функций и инструментов для научных и инженерных вычислений, включая построение матриц сходств. Одно из основных преимуществ SciPy - это скорость работы. Она оптимизирована для вычислительных задач и предоставляет высокую производительность благодаря использованию языка программирования C. Более того, SciPy имеет богатую коллекцию функций для работы с различными типами данных и предоставляет мощный и гибкий функционал для настройки и управления вычислениями.

Библиотека os является стандартной библиотекой Python, которая предоставляет множество функций для работы с операционной системой, такие как чтение и запись файлов, управление директориями и многое другое. Одно из основных преимуществ os - это его простота использования. Функции этой библиотеки имеют понятный синтаксис и предоставляют широкий спектр возможностей для работы с файловой системой. Кроме того, библиотека os позволяет писать переносимый код, который будет работать на разных операционных системах.

Библиотека pandas является одной из самых популярных библиотек для работы с данными в Python. Ее основное преимущество - это возможность работы с файлами csv и другими форматами данных. Pandas предоставляет мощный и гибкий функционал для чтения и записи данных, фильтрации, агрегации и многих других операций над ними. Одним из ключевых преимуществ pandas является скорость работы. Она оптимизирована для быстрой обработки больших объемов данных и предоставляет множество инструментов для ускорения вычислений. Кроме того, pandas предоставляет удобный и интуитивно понятный синтаксис, который делает ее доступной для использования даже новичкам в области анализа данных.

Библиотека numpy является одной из самых популярных библиотек для работы с массивами и матрицами в Python. Ее основное преимущество - это скорость работы и мощный функционал для работы с многомерными массивами данных. Numpy использует оптимизированный C-код для выполнения матричных операций, что позволяет ей работать очень быстро и обрабатывать большие объемы данных. Кроме того, numpy предоставляет множество инструментов для работы с матрицами, векторами, линейной алгеброй, численными методами и многими другими математическими операциями. Одним из дополнительных преимуществ numpy является ее способность работать с различными типами данных, включая числа с плавающей точкой, комплексные числа и другие. Кроме того, numpy имеет мощный и удобный интерфейс для интеграции с другими библиотеками, такими как pandas и matplotlib, что делает ее очень гибким и мощным инструментом для работы с данными и анализа данных.

## 2.3 Реализация коллаборативной фильтрации

В рамках данной ВКР была реализована коллаборативная фильтрация, которая позволяет предсказывать оценки пользователей на основе оценок, которые поставили похожие пользователи. Для реализации данного алгоритма были использованы следующие шаги:

1. Импортируются необходимые библиотеки: pandas, sklearn.metrics.pairwise и os.
2. Функция collaborative\_system принимает два аргумента: sources\_path и results\_path.
3. Далее, создаются две переменные - exhibit\_data\_path и current\_user\_path, которые содержат путь к файлам exhibit\_data.csv и current\_user.csv соответственно.
4. Проверяется наличие файлов exhibit\_data.csv и current\_user.csv по заданным путям при помощи модуля os. Если хотя бы один из файлов не найден, выбрасывается исключение типа FileNotFoundError.
5. Загружаются данные из файлов exhibit\_data.csv и current\_user.csv в виде таблиц pandas DataFrame с помощью метода pd.read\_csv.
6. Выполняются необходимые операции предобработки данных. Например, в коде выполняется объединение таблиц с помощью метода pd.concat и установление верхнего ограничения времени, потраченного на экспонат, при помощи метода clip.
7. Строится матрица взаимодействий пользователей с объектами, используя метод pivot\_table библиотеки pandas.
8. Вычисляется косинусное сходство между пользователями, используя метод cosine\_similarity из библиотеки sklearn.metrics.pairwise.
9. Применяется алгоритм коллаборативной фильтрации, который основывается на сходстве между пользователями и предсказывает оценку, которую пользователь бы дал объекту, на основе оценок, которые поставили похожие пользователи.
10. Результаты предсказаний сохраняются в CSV-файл recs\_collaborative.csv в заданную папку results\_path.

В качестве меры близости была выбрано косинусное сходство. Так, если мы рассматриваем пользователей как векторы, где каждый компонент представляет собой оценку пользователя для определенного элемента, то косинусное сходство может помочь определить, насколько похожи два пользователя на основе их оценок.

Косинусное сходство  — это мера сходства между двумя векторами предгильбертового пространства, которая используется для измерения косинуса угла между ними.

Если даны два вектора признаков, A и B, то косинусное сходство, cos(θ), может быть представлено используя скалярное произведение и норму (1)

, (1)

Также в работе реализована функция collaborative\_filtering\_validation, которая используется для проверки качества работы алгоритма коллаборативной фильтрации на подвыборке данных. Для проверки качества алгоритма используется средняя абсолютная ошибка между предсказанными и фактическими оценками.

Алгоритм валидации состоит из нескольких шагов. Сначала данные делятся на несколько частей, после чего алгоритм коллаборативной фильтрации применяется на каждой из частей с использованием подвыборки пользователей. Затем средняя абсолютная ошибка между предсказанными и фактическими оценками вычисляется для каждой из подвыборок. Если на всех подвыборках полученные результаты превышают предыдущие значения, то алгоритм считается успешно валидированным.

Для проверки качества работы алгоритма используются уже подгруженные из файла exhibit\_data.csv и current\_user.csv, которые предварительно загружаются в виде таблиц pandas DataFrame. После чего происходит предобработка данных, включающая объединение таблиц и установление верхнего ограничения времени, потраченного на экспонат.

Таким образом, валидация позволяет оценить точность работы алгоритма коллаборативной фильтрации на случайной подвыборке данных и сделать выводы о его эффективности и применимости к реальным данным.

По данным валидации на данных музея при выборке первой из пяти частей пользователей, отсортированных по схожести с искомым, погрешность предсказаний в секундах составляет около 15-20%.

## 2.4 Реализация рекомендательной системы, основанной на схожести объектов

## При реализации рекомендательной системы, основанной на схожести объектов необходимо выбрать меру близости для векторов информации об объектах. В данной ВКР была использована косинусная мера сходства.

Косинусное сходство является одной из наиболее распространенных мер близости, используемых в рекомендательных системах и машинном обучении. Оно измеряет сходство между двумя векторами, определяя косинус угла между ними.

В контексте рекомендательных систем, косинусное сходство часто используется для измерения близости между пользователями или элементами. Например, если мы рассматриваем пользователей как векторы, где каждый компонент представляет собой оценку пользователя для определенного элемента, то косинусное сходство может помочь определить, насколько похожи два пользователя на основе их оценок. Аналогично, для элементов, косинусное сходство может помочь определить, насколько похожи два элемента на основе оценок, которые пользователи дали им.

Формула косинусного сходства проста для вычисления, быстро работает и легко интерпретируется. Она также не требует нормализации данных, что делает ее более удобной для использования в ситуациях, где данные могут иметь различный масштаб или единицы измерения. Все это делает косинусное сходство популярным выбором для реализации рекомендательных систем, основанных на контенте и коллаборативной фильтрации.

Краткое описание действий, производимых в коде:

1. Импортируются необходимые библиотеки, включая pandas для работы с таблицами данных, sklearn для масштабирования данных и вычисления меры сходства, numpy для работы с массивами и матрицами данных, и os для работы с файловой системой.
2. Функция content\_based используется для создания рекомендательной системы на основе контента, то есть для рекомендации объектов, похожих на последний посещенный пользователем объект музея.
3. Задаются пути к файлам с данными пользователей и экспонатов в музее. Если указанные файлы не найдены, выбрасывается исключение FileNotFoundError.
4. Загружаются данные о текущем пользователе и описания экспонатов музея в виде таблиц pandas DataFrame.
5. Определяется ID последнего посещенного пользователем объекта музея.
6. Из таблицы экспонатов выбираются релевантные признаки для вычисления меры сходства.
7. Категориальные признаки закодированы как числовые, используя функцию pd.get\_dummies.
8. Вычисляется косинусное сходство между всеми парами экспонатов в музее.
9. Находятся наиболее похожие на последний посещенный пользователем объекты музея с помощью функции \_recommend\_wines. Для этого происходит сортировка объектов музея по косинусному сходству с последним посещенным пользователем объектом.
10. Проверяется качество работы алгоритма рекомендации контента с помощью валидации. Для этого происходит разделение выборки на пять частей, вычисление средней косинусной меры сходства между каждой частью и последним посещенным пользователем объектом, и проверка того, что эти меры сходства упорядочены по убыванию. Если валидация проходит успешно, результаты сохраняются в файле recs\_content\_based.csv. Если валидация не проходит, выбрасывается искл­­ючение ValueError.

Плюсы данного подхода:

Модульность: функция "content\_based" отвечает за рекомендательную систему, использующую контентную фильтрацию, и легко вызывается из других модулей.

Надежность: код включает проверки наличия файлов с данными и возвращает ошибку, если они не найдены, что помогает гарантировать, что программа не запустится с неполными данными.

Скорость: код использует библиотеки NumPy и Pandas для эффективной работы с данными. Он также использует функцию argsort из NumPy для быстрой сортировки по сходству.

Чистота: код четко разделяет данные и функциональность, что делает его легко читаемым и понятным для других разработчиков.

Проверка на валидность: код содержит функцию "validation\_result", которая проверяет, насколько хорошо рекомендательная система работает на определенных данных, и сообщает об ошибке, если результаты не соответствуют ожиданиям.

Использование научных библиотек: код использует научные библиотеки, такие как Scikit-learn, для упрощения реализации рекомендательной системы и повышения ее эффективности.

## 2.5 Реализация рекомендательной системы на основе расстояний

Реализация рекомендательной системы на основе расстояний выглядит следующим образом:

1. Код загружает данные о координатах объектов и матрице расстояний между комнатами из csv файлов.
2. Из таблицы координат создается список объектов, представленных в виде словарей.
3. Матрица расстояний между комнатами представлена в виде pandas DataFrame, затем преобразуется в словарь, где ключами являются id комнат, а значениями являются другие словари, где ключами являются id соседних комнат, а значениями - расстояния между ними.
4. Создается функция \_recommend\_distances для расчета рекомендуемых объектов.
5. Для каждого объекта в списке рассчитывается расстояние до текущего объекта с помощью функции \_calculate\_distance. Если объекты находятся в одной комнате, то расстояние рассчитывается с помощью функции np.hypot между координатами x и y текущего объекта и объекта в списке. В противном случае, для расчета расстояния между комнатами берется значение из матрицы расстояний, затем добавляются расстояния от текущего объекта до двери комнаты и от двери до объекта в списке.
6. Результаты сохраняются в объект DataFrame, где столбцом являются расстояния до каждого объекта в списке, а индексом являются их id.
7. Проверка на валидность заключается в том, что результаты содержат только положительные значения и не имеют пропущенных значений.
8. Результаты сохраняются в CSV файл и выводится сообщение об успешном прохождении проверки валидности. Если проверка не проходит, то вызывается исключение ValueError с сообщением об ошибке.

Преимущества реализации:

1. Эффективное использование библиотек pandas и numpy для работы с данными. Код использует DataFrame для хранения координат и расстояний между комнатами.
2. Хорошо структурированный код, разделенный на несколько функций, что облегчает его понимание и модификацию.
3. Использование функции numpy.hypot для вычисления расстояний между объектами, что является более эффективным, чем использование стандартной формулы.
4. Проверка наличия и правильности данных с помощью функций os.path.isfile и isnull, что делает код более надежным и устойчивым к ошибкам.
5. Понятные и осмысленные названия функций и переменных, что делает код более читаемым и легким для понимания.
6. Применение концепции векторизации для выполнения вычислений на массивах данных, что повышает производительность и скорость работы кода.
7. Реализация валидации кода, что помогает убедиться в его правильности и достоверности результатов.
8. Использование комментариев для пояснения кода, что облегчает его понимание и поддержку.

## 2.5 Реализация объединения данных всех рекомендательных систем

## Для создания гибридной системы в работе использовался способ смешивания результатов <https://doi.org/10.1023/A:1021240730564>. Рейтинги всех трех систем нормализуются к баллам (от 0 до 100) и складываются. В работе предусмотрена установка коэффициентов влияния каждой рекомендательной системы на итоговый результат.

## Краткое описание алгоритма объединения результатов:

1. Импортируются необходимые библиотеки и модули: pandas и os.
2. Функция merge\_recommendations принимает пути к исходным файлам (sources\_path) и директории для сохранения результатов (results\_path), а также три коэффициента для определения весов для каждого из трех методов рекомендаций (content\_based\_coeff, collaborative\_coeff, distance\_based\_coeff).
3. Функция загружает три файла рекомендаций: recs\_collaborative.csv, recs\_content\_based.csv и recs\_distances.csv.
4. Если файл recs\_content\_based.csv или recs\_distances.csv не существует, возникает ошибка FileNotFoundError.
5. Функция вычисляет "оценки" для каждого из методов рекомендаций, используя максимальные значения сходства для метода на основе содержания и минимальные расстояния для метода на основе расстояния.
6. Если файл recs\_collaborative.csv существует, функция вычисляет также оценки для метода на основе коллаборативной фильтрации и суммирует оценки для каждого объекта в одно общее значение "score".
7. Функция объединяет три таблицы по ID объекта, суммирует "оценки" для каждого метода и сохраняет их в файле merged\_recommendations.csv.
8. Если файл recs\_collaborative.csv не существует, функция создает пустой датафрейм, добавляет отсутствующие ID из файлов recs\_content\_based.csv и recs\_distances.csv со значением "score" равным 0, исключает из него объекты, которые уже посетил пользователь, и затем объединяет все три таблицы.

Данная реализация позволяет объединять рекомендации, полученные с помощью трех разных алгоритмов: коллаборативной фильтрации, контентной фильтрации и алгоритма на основе расстояний. Такое объединение позволяет получать более точные и разнообразные рекомендации, что улучшает качество сервиса и повышает удовлетворенность пользователей.

Преимущества данного кода также включают возможность гибкой настройки весовых коэффициентов для каждого алгоритма, что позволяет более точно настроить соотношение между разными типами рекомендаций и настроить сервис под конкретные потребности пользователей. Кроме того, код также автоматически определяет отсутствующие рекомендации в каждом из трех файлов и заменяет их на нулевые значения, что обеспечивает полноту объединенных рекомендаций.

## 2.6 Реализация объединения результатов гибридной системы с результатами стохастической оптимизации

## При объединении результатов гибридной системы с результатами стохастической оптимизации по аналогии с пунктом 2.5 выбрана нормализация к баллам. Однако возникает проблема, так как в результатах эвристического алгоритма изначально отсутствуют баллы.

Процесс распределения баллов для эвристической системы основан на идеи, что пользователь обычно предпочитает находиться вблизи своих предыдущих объектов исследования. Для этого система определяет количество строк в эвристическом файле и начисляет баллы, исходя из порядкового номера строки. Чем ближе строка к концу, тем больше очков она получит.

Полученные баллы затем нормализуются и преобразуются для использования вместе с баллами, полученными из других методов рекомендаций. Результаты каждого метода рекомендаций объединяются, и баллы для каждого объекта подсчитываются как максимальный балл из всех методов. Результаты сортируются по полученным баллам и сохраняются в файле для использования в качестве окончательных рекомендаций.

## Описание работы алгоритма объединения результатов:

1. Импортируются необходимые модули: pandas и os.
2. Получаются пути к двум CSV-файлам: heuristic.csv и merged\_recommendations.csv, результаты которых будут использоваться для получения окончательного списка рекомендаций.
3. Проверяется, существуют ли оба файла по указанным путям. Если хотя бы один из них не найден, вызывается исключение.
4. Загружаются данные из файлов в соответствующие объекты DataFrame: heuristic и recommendations\_df.
5. Из эвристических результатов оставляются только объекты, присутствующие в списке объединенных рекомендаций.
6. Вычисляется максимальное количество очков, которое можно получить из списка объединенных рекомендаций.
7. Для каждого объекта в списке эвристических результатов вычисляется количество очков на основе его порядкового номера в списке. Этот показатель преобразуется таким образом, чтобы максимальное значение было равно максимальному количеству очков из списка объединенных рекомендаций, а затем умножается на коэффициент эвристики.
8. Объединяются два DataFrame - эвристический и объединенный список рекомендаций - по ID объектов.
9. Для каждого объекта вычисляется максимальное количество очков, которое он может получить из объединенного списка рекомендаций и из эвристического списка.
10. Создается новый столбец 'Optimal\_Scores' в DataFrame, который содержит максимальное количество очков для каждого объекта.
11. Удаляются столбцы 'merge\_scores' и 'heuristic\_scores'.
12. DataFrame сортируется по столбцу 'Optimal\_Scores' в порядке убывания.
13. Полученный DataFrame сохраняется в файл final\_results.csv.
14. Выводится сообщение об успешном выполнении задачи.

## 2.7 Запуск и работа рекомендательной системы

## Запуск рекомендательной системы происходит через файл main.py. При работе программы используется распараллеливание процессов для ускорения работы программы.

## Процесс работы программы:

1. Выводится сообщение о запуске программы.
2. Считывается файл конфигурации config.yaml, который содержит пути к файлам данных и коэффициенты для вычисления баллов рекомендаций.
3. Используется модуль Julia для выполнения вычислений на Julia в файле heuristic.jl.
4. Запускаются функции, которые соответствуют системам рекомендаций: content\_based, distance, collaborative\_system. Они запускаются в отдельных процессах.
5. После того, как процессы систем рекомендаций завершены, вызываются функции merge\_recommendations и interference\_to\_euristic.
6. Функция merge\_recommendations объединяет рекомендации из всех трех систем и вычисляет окончательные баллы для каждой рекомендации.
7. Функция interference\_to\_euristic добавляет баллы на основе эвристической системы, которая оценивает экспоненциальную функцию, вычисленную на основе позиции объекта в списке и количества общих объектов с последним просмотренным объектом пользователем.
8. Выводится сообщение об успешном завершении программы и время, затраченное на ее выполнение.

## Пример вывода при корректном выполнении программы представлен на рисунке 3.

## 

Рисунок 3 - Вывод окна терминала

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения данного проекта было создано несколько рекомендательных систем для музея с использованием различных подходов: на основе сходства объектов, коллаборативной и на основе расстояния. Были использованы методы машинного обучения и анализа данных, а также имплементировано исполнение языка программирования Julia. В результате были получены списки рекомендаций для пользователей, основанные на их предпочтениях.

Оценка полноты решений показала, что системы достаточно хорошо справляются с задачей рекомендации объектов в музее на основе различных критериев. В том числе, результаты валидации для каждой из систем показали высокую точность и эффективность.

Рекомендуется использовать полученные результаты для улучшения пользовательского опыта в музее. Для этого можно создать приложение, которое бы отображало список рекомендаций для каждого посетителя на основе его предпочтений. Также можно использовать полученные данные для анализа поведения пользователей и оптимизации размещения экспонатов в музее.

Результаты оценки технико-экономической эффективности показали, что внедрение рекомендательных систем может улучшить качество обслуживания и удовлетворенность пользователей, что может привести к увеличению доходов музея.

Научно-технический уровень выполненной работы находится на уровне лучших достижений в этой области. Использование различных методов и подходов позволило создать эффективные системы рекомендаций для музеев, что может быть полезно в других областях, где необходимо рекомендовать объекты на основе пользовательских предпочтений.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

**ПРИЛОЖЕНИЕ А**

QR-код репозитория с кодом на GitHub:

****