



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

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №2 по курсу «Проектирование Рекомендательных Систем»

Тема Сравнение алгоритмов коллаборативной фильтрации

Студент Якуба Д. В.

Группа ИУ7-33М

Оценка (баллы) _____

Преподаватели Быстрицкая А.Ю.

Оглавление

Введение	3
1 Аналитический раздел	4
1.1 Коллаборативная фильтрация	4
1.2 Коллаборативная фильтрация по пользователю	4
1.3 Коллаборативная фильтрация по объекту	4
2 Конструкторский раздел	6
2.1 MovieLens Rating	6
3 Технологический раздел	7
3.1 Средства реализации	7
3.2 Библиотеки	7
4 Исследовательский раздел	8
4.1 Условия исследований	8
4.2 Зависимости времени выполнения коллаборативной фильтрации от параметра количества используемых похожих объектов для каждой из мер близости	8
4.3 Зависимости значений ROC-AUC от параметра количества используемых похожих объектов	10
ЗАКЛЮЧЕНИЕ	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	14

ВВЕДЕНИЕ

Цель работы – сравнение алгоритмов коллаборативной фильтрации по пользователю и по объекту.

Для достижения поставленной цели потребуется:

- привести описание алгоритмов;
- привести описание используемых для исследования данных;
- провести сравнение алгоритмов по времени работы и точности предсказания.

1. Аналитический раздел

1.1 Коллаборативная фильтрация

Коллаборативная фильтрация – это метод рекомендации, при котором анализируется только реакция пользователей на объекты: оценки, которые выставляют пользователи объектам. [1]

Оценки могут быть как явными (выставленная пользователем оценка), так и неявными (например, длительность нахождения пользователя на странице товара). Целью коллаборативной фильтрации является предсказание оценки пользователем по оценкам других. Чем больше оценок имеется, тем точнее получатся рекомендации. [1]

1.2 Коллаборативная фильтрация по пользователю

В данном методе предполагается, что пользователи, которые в прошлом имели похожие предпочтения, будут иметь похожие предпочтения и в будущем. Для построения рекомендаций с использованием этого метода выполняются следующие шаги:

1. Создание матрицы, где строки представляют пользователей, а столбцы объекты; значения в ячейках матрицы отражают оценки или действия пользователей по отношению к объектам;
2. Вычисление сходства между пользователями (обычно используются косинусное сходство или корреляция Пирсона);
3. Для конкретного пользователя генерируются рекомендации на основе сходства с другими пользователями через агрегацию оценок или действий похожих пользователей.

1.3 Коллаборативная фильтрация по объекту

В данном методе предполагается, что пользователи будут интересоваться теми объектами, которые похожи на уже положительно оцененные объекты. Для построения рекомендаций с использованием метода выполняются следующие шаги:

1. Создание матрицы, где строки представляют объекты, а столбцы пользователей; значения в ячейках матрицы определяют оценки или действия пользователей по отношению к объектам;
2. Вычисление сходства между пользователями;
3. Генерация рекомендаций с опорой на сходство между объектами, кото-

рые пользователь уже оценил.

2. Конструкторский раздел

2.1 MovieLens Rating

В качестве источника данных был взят датасет, располагающийся в свободном доступе на веб-сайте MovieLens [2]. Набор данных включает в себя множество записей, поля которых описывают идентификатор пользователя, объект, оценку (от 1 до 5) и время появления рецензии. Предобработки проводить не потребовалось.

3. Технологический раздел

3.1 Средства реализации

В качестве используемого был выбран язык программирования Python [3].

Данный выбор обусловлен следующими факторами:

- Большое количество исчерпывающей документации;
- Широкий выбор доступных библиотек для разработки;
- Простота синтаксиса языка и высокая скорость разработки.

При написании программного продукта использовалась среда разработки Visual Studio Code. Данный выбор обусловлен тем, что данная среда распространяется по свободной лицензии, поставляется для конечного пользователя с открытым исходным кодом, а также имеет большое число расширений, ускоряющих разработку.

3.2 Библиотеки

При анализе и обработке датасета, а также для решения поставленных задач использовались библиотеки:

- pandas;
- numpy;
- matplotlib;
- sklearn [4];
- LibRecommender [5].

Данные библиотеки позволили полностью покрыть спектр потребностей при выполнении работы.

4. Исследовательский раздел

4.1 Условия исследований

Исследование проводилось на персональном вычислительной машине со следующими характеристиками:

- процессор Apple M1 Pro,
- операционная система Ventura 13.5.2,
- 32 Гб оперативной памяти.

Временные затраты определялись с использованием библиотеки `time`. Значение ROC-AUC определялось с использованием библиотеки `LibRecommender`.

В данном исследовании значение параметра, отвечающего за количество используемых похожих объектов, менялось от 1 до 30. Также для каждого алгоритма и значения параметра использовались косинусная мера, критерий Пирсона и мера Жаккара.

4.2 Зависимости времени выполнения коллаборативной фильтрации от параметра количества используемых похожих объектов для каждой из мер близости

На рисунке 4.1 представлен график зависимости времени выполнения коллаборативной фильтрации по пользователю от параметра количества используемых похожих объектов для каждой из мер близости.

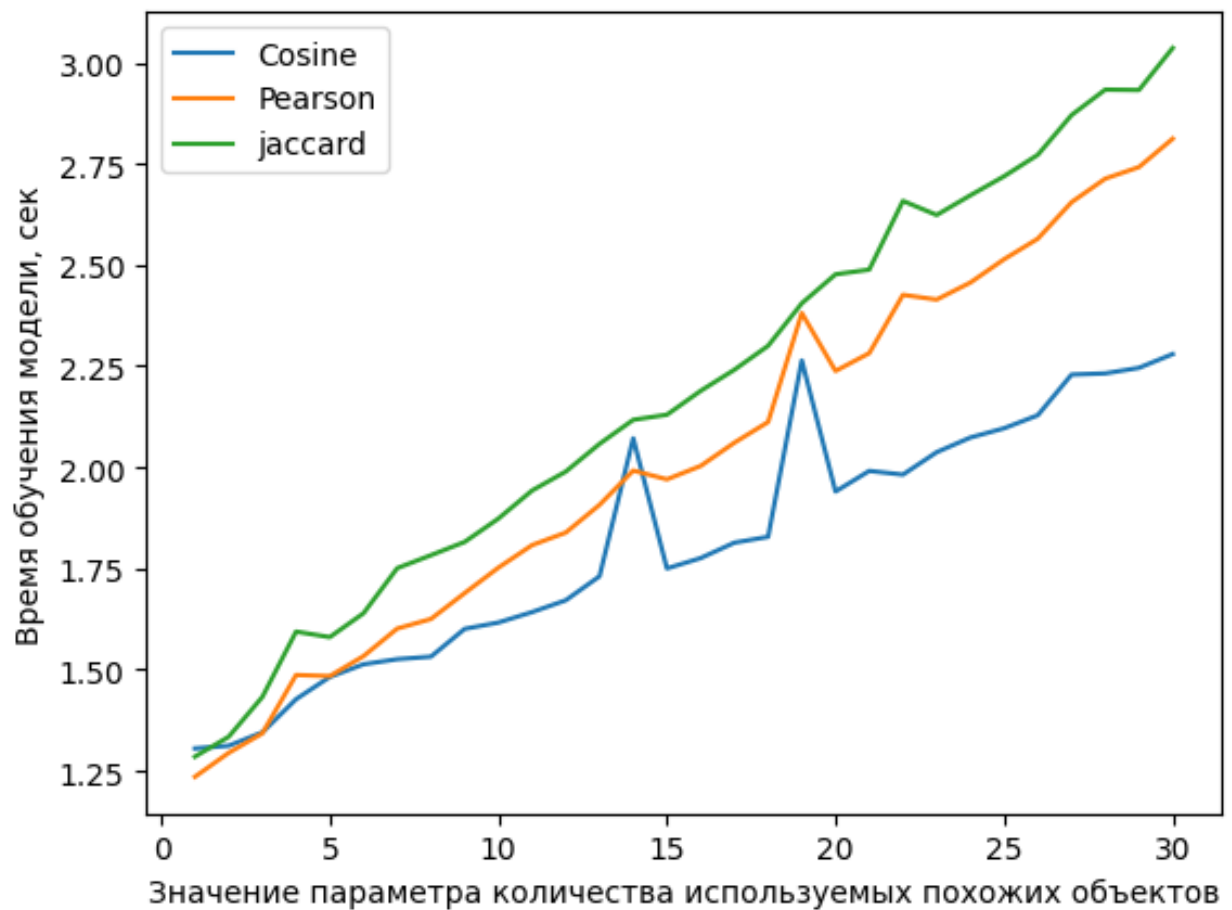


Рис. 4.1: График зависимости времени выполнения коллаборативной фильтрации по пользователю от параметра количества используемых похожих объектов для каждой из мер близости.

На рисунке 4.2 представлен график зависимости времени выполнения коллаборативной фильтрации по объекту от параметра количества используемых похожих объектов для каждой из мер близости.

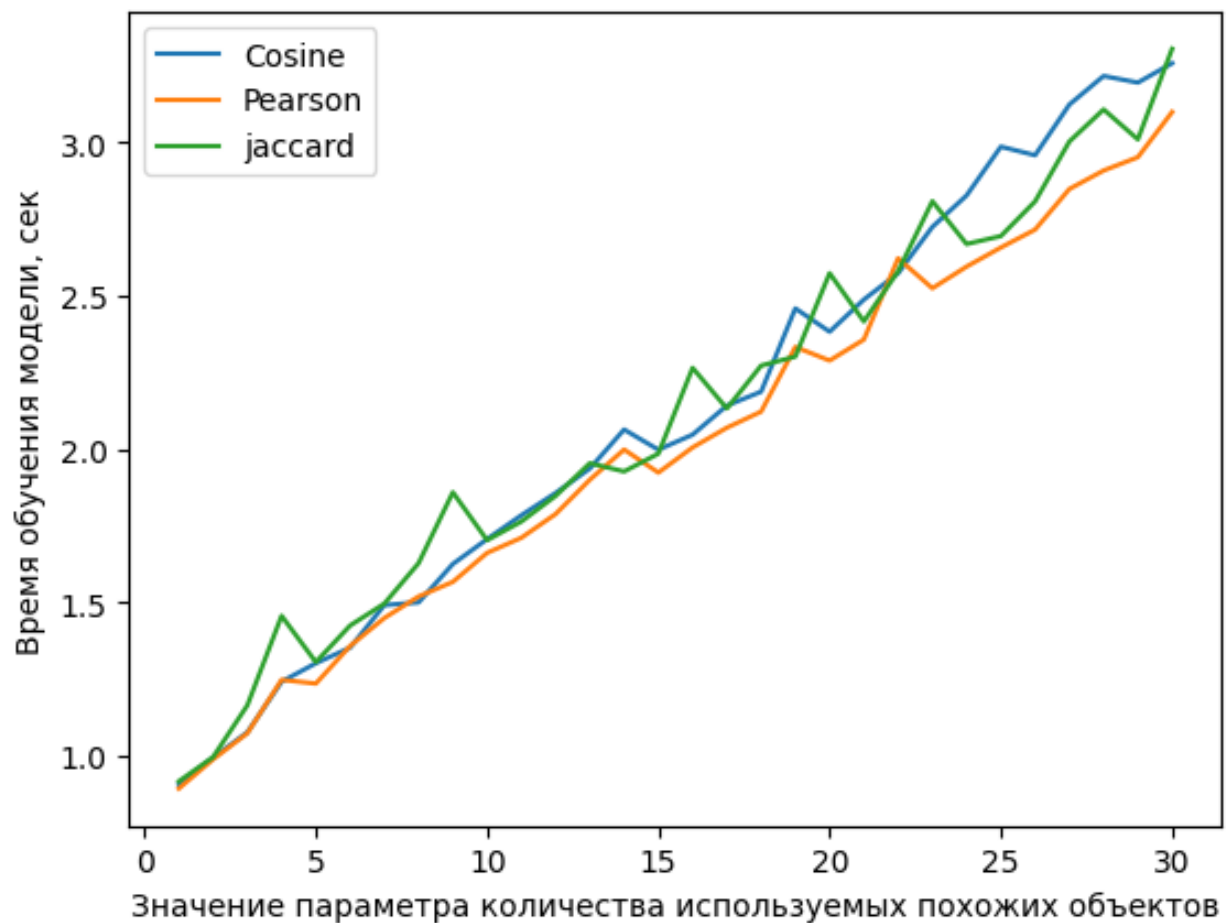


Рис. 4.2: График зависимости времени выполнения коллаборативной фильтрации по объекту от параметра количества используемых похожих объектов для каждой из мер близости.

4.3 Зависимости значений ROC-AUC от параметра количества используемых похожих объектов

На рисунке 4.3 представлен график зависимости ROC-AUC (при фильтрации по пользователю) от параметра количества используемых похожих объектов для каждой из мер близости.

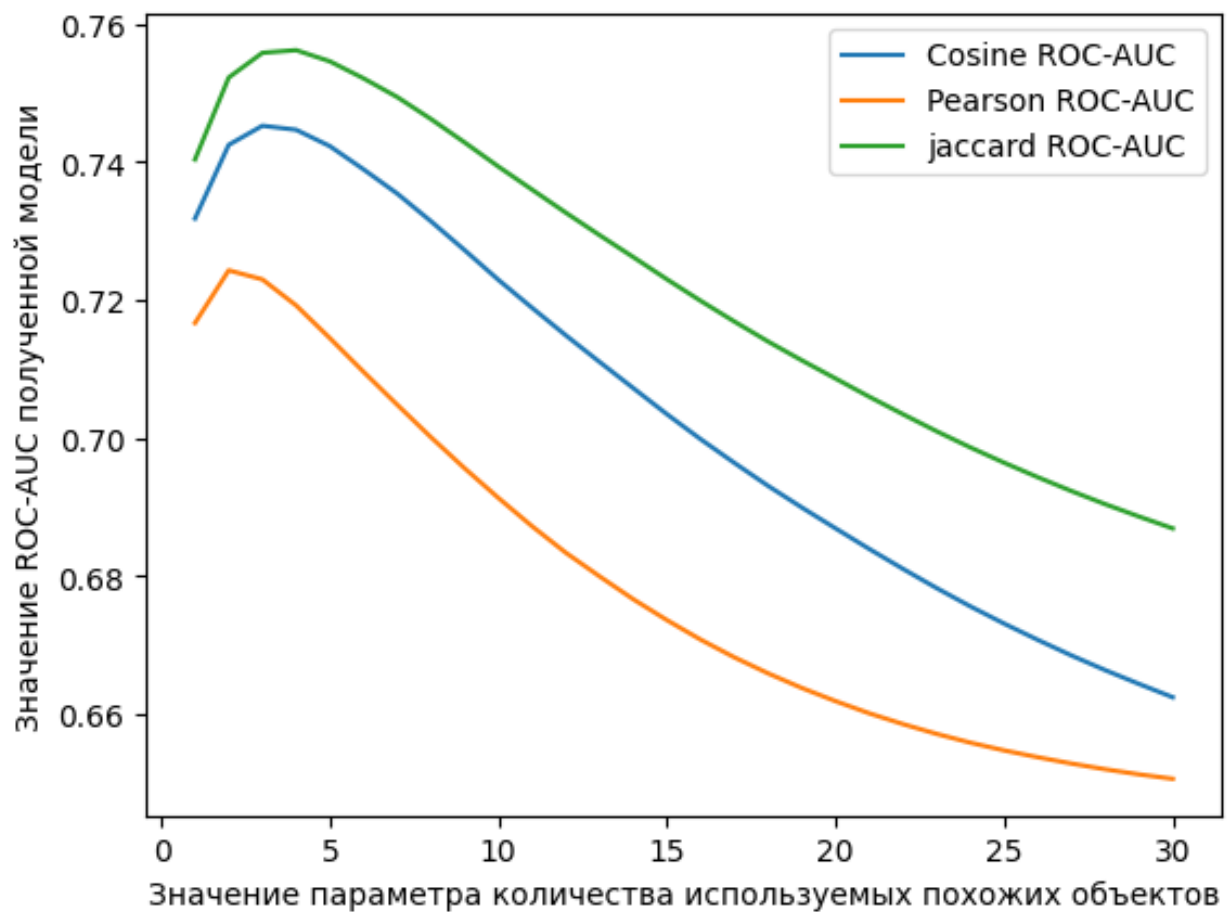


Рис. 4.3: График зависимости ROC-AUC (при фильтрации по пользователю) от параметра количества используемых похожих объектов для каждой из мер близости.

На рисунке 4.4 представлен график зависимости ROC-AUC (при фильтрации по объекту) от параметра количества используемых похожих объектов для каждой из мер близости.

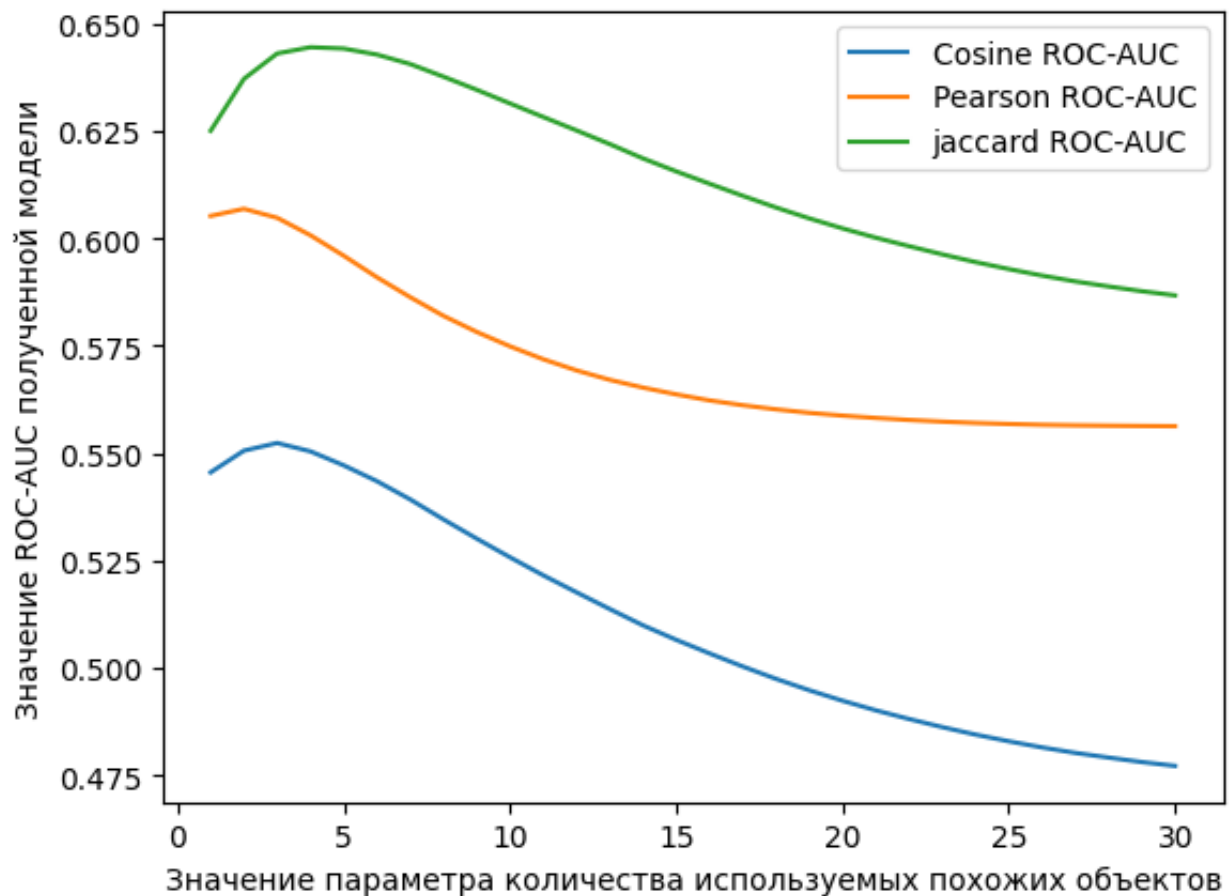


Рис. 4.4: График зависимости ROC-AUC (при фильтрации по объекту) от параметра количества используемых похожих объектов для каждой из мер близости.

Заключение

В результате проведенных исследований заметно, что при фильтрации по пользователю, лучше всего по времени себя показала косинусная мера на значениях параметра больше 5. В случае фильтрации по объекту, сложно выделить какую-то из мер близости.

Также стало понятно, что на рассматриваемом датасете с точки зрения ROC-AUC лучшим значением параметра количества используемых похожих объектов при фильтрации и по пользователю, и по объекту, будет находится в интервале $[3, 5]$, причем лучше всего себя показывает именно мера Жаккара.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы было проведено сравнение алгоритмов коллаборативной фильтрации по пользователю и по объекту.

Были решены следующие задачи:

- приведено описание алгоритмов;
- приведено описание используемых для исследования данных;
- проведено сравнение алгоритмов по времени работы и точности предсказания.

Проведенные исследования показали, что на использованном датасете с точки зрения ROC-AUC в двух вариациях лучше всего себя показывает фильтрация с параметром используемых похожих объектов, лежащим в интервале от 3 до 5, и используемой мерой Жаккара. Также стало понятно, что некоторый выигрыш по времени за счет использования различных мер близости, можно получить лишь при фильтрации по пользователю.

Список литературы

1. Владимировна Смоленчук Татьяна. Метод коллаборативной фильтрации для рекомендательных сервисов // Вестник науки и образования. 2019. № 22-1.
2. MovieLens [Электронный ресурс]. Режим доступа: <https://grouplens.org/datasets/movielens/> (дата обращения 16.09.2023).
3. Python official page [Электронный ресурс]. Режим доступа: <https://www.python.org/> (дата обращения 10.05.2023).
4. Scikit-learn official page [Электронный ресурс]. Режим доступа: <https://scikit-learn.org/stable/> (дата обращения 10.05.2023).
5. LibRecommender: official PyPI project page [Электронный ресурс]. Режим доступа: <https://pypi.org/project/LibRecommender/> (дата обращения 16.09.2023).