|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа №2**

**по курсу «Проектирование рекомендательных систем»**

|  |  |
| --- | --- |
| **Тема: «Сравнение алгоритмов коллаборативной фильтрации»**  **Студент Горячев В. Г.**  **Группа ИУ7-33М**  **Преподаватель Быстрицкая А. Ю.** |  |

Москва

2023 г

## Описание алгоритмов

***Фильтрация по пользователям***

Алгоритм основан на использовании мер близости. В матрице оценок пользователями товаров ищутся пользователи, схожие друг с другом. Подразумевается, что людям со схожими покупками/вкусами можно предположить те объекты, которые один из похожих пользователей ещё не попробовал

***Фильтрация по объектам***

Тот же принцип, только наоборот – ищутся похожие по оценкам предметы, поскольку если они понравились людям, то, возможно, и похожие объекты можно им предложить.

***Библиотеки***

В качестве языка программирования был выбран язык Python вместе с интерактивной средой Jupyter Notebook, поскольку они предоставляют удобный инструментарий для исследования, в частности, для выполнения лабораторных работ. Это определило выбор библиотек — нужно было найти совместимые с языком программирования.

В качестве источника алгоритмов использовались материалы из интернета для написания алгоритмов, подлежащих тестированию, поскольку не получилось найти библиотеки, предоставляющие их в готовом виде. Все реализации используют стандартные библиотеки – numpy, pandas и sklearn.

***Данные***

Для сравнения алгоритмов был использован учебный вариант набора данных MovieLens (<https://grouplens.org/datasets/movielens/>). Он содержит оценки пользователей по фильмам и специально составлен для подобного рода учебных задач.

***Сравнение***

Чтобы упростить сравнение, использовался стандартный подход из машинного обучения – разделение набора данных на тренировочную и обучающую части и применение метрики, сравнивающей предсказания модели и истинные ответы. В данном случае использовалась метрика RMSE (корень среднеквадратической ошибки), которая сравнивала прогнозы оценок пользователей для объектов, отсутствующих в обучающем наборе, с оценками из тестового. Также была возможность посмотреть время работы алгоритмов в зависимости от количества записей в наборе.

Между собой сравнивались также результаты работы алгоритмов в зависимости от выбранной метрики близости – евклидовой, косинусной и корреляционной.

Время на обработку данных:

* косинусная мера: 2.1 c;
* евклидово расстояние: 1.5 c;
* корреляция: 23.5 c.

Рисунок 1 — Зависимость качества работы от метрики

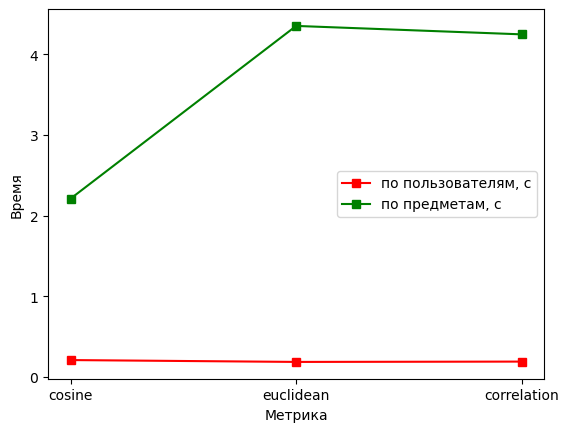
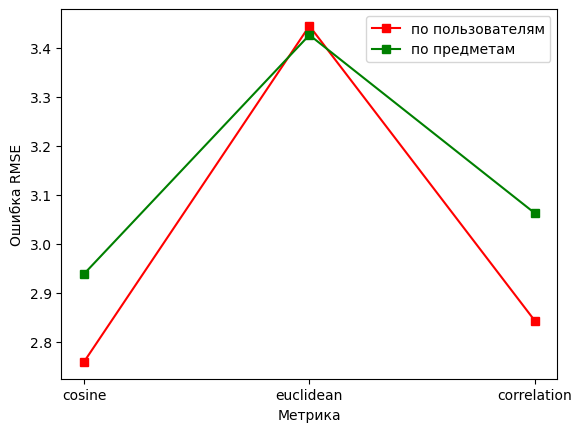


Рисунок 2 — Зависимость времени работы от метрики

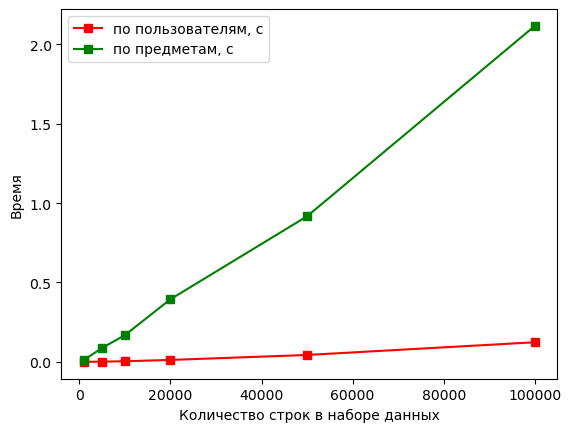


Рисунок 3 — Зависимость времени работы от размера набора

**Вывод**

В ходе лабораторной работы были сравнены алгоритмы коллаборативной фильтрации по пользователям и по предметам. Исходя из проведённого сравнения лучшие результаты показывает алгоритм фильтрации по пользователям с косинусной мерой близости как по качеству работы, так и по времени. Стоит только отметить, что алгоритм фильтрации по объектам уступает по времени только в силу того, что количество фильмов в данном наборе данных существенно выше количества пользователей, однако, судя по графику,