|  |  |
| --- | --- |
| 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 №4**

**по курсу «Проектирование рекомендательных систем»**

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

Москва

2023 г

# Описание алгоритмов

Матричная факторизация — разложение матрицы на произведение двух или более матриц.

***SVD в рекомендательных системах***

Разложение матрицы осуществляется на такие матрицы: *,* что элементы, лежащие на главной диагонали матрицы — это сингулярные числа, а все элементы остальные элементы являются нулевыми. Это позволяет получить приближённую матрицы к матрице М, с известным рангом. Это позволяет выделять главные факторы, способствующие выбору того или иного товара и осуществлять, на основе этого приближения, предсказания-рекомендации для пользователей.

***NMF***

Факторизация неотрицательных матриц (NMF) — это представление матрицы V в виде произведения матриц W и H, в котором все элементы трёх матриц неотрицательны.

# 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, а также сами библиотеки. Использованы стандартные библиотеки – numpy, pandas и sklearn, а также к ним nltk для набора стоп-слов, matplotlib, umal-learn.

# 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.

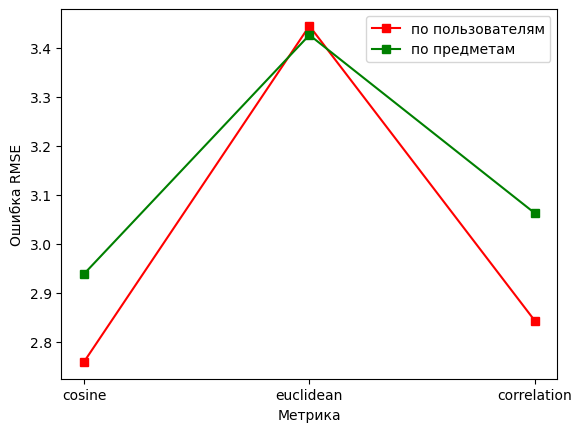
# Сравнение

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

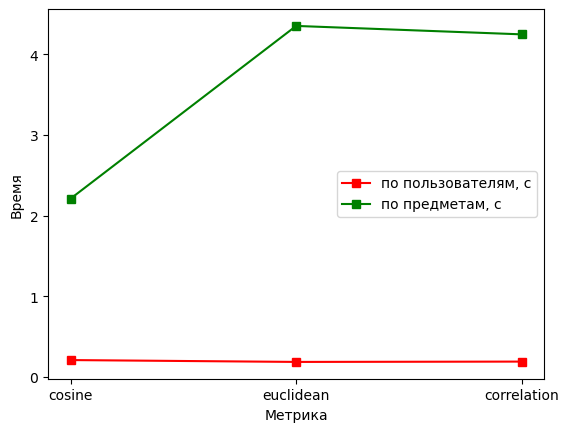


Рисунок 2 — Зависимость времени работы от метрики

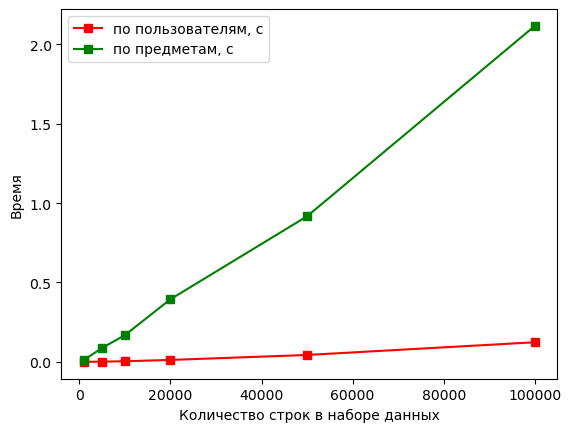


Рисунок 3 — Зависимость времени работы от размера набора

"4/~$\321\200\321\217\321\207\320\265\320\262 4.docx"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.