|  |  |
| --- | --- |
| 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 №3**

**по курсу «Проектирование рекомендательных систем»**

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

Москва

2023 г

# Описание алгоритмов

***Фильтрация по контенту: общая часть***

Алгоритм основан на анализе текста, переводящем естественный язык описания продукта в некое векторное представление, которое может быть сравнено любыми разумными в данной задаче способами.

***Использование tf-idf в качестве векторайзера***

TF-IDF — статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. Вес некоторого слова пропорционален частоте употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции.

***Использование скрытого распределения Дирихле (LDA)***

Сложно, не могу объяснить… Тоже статистика, тоже поиск множественных тематик документа. По сути, тоже позволяет получить вектор.

# 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 набор данных игр из Steam (https://www.kaggle.com/datasets/trolukovich/steam-games-complete-dataset/data). Он содержит описания игр, теги, краткие описания и другую информацию из магазина. Использовались только половина строка из 40 000, поскольку для таких матриц не хватало уже оперативной памяти.

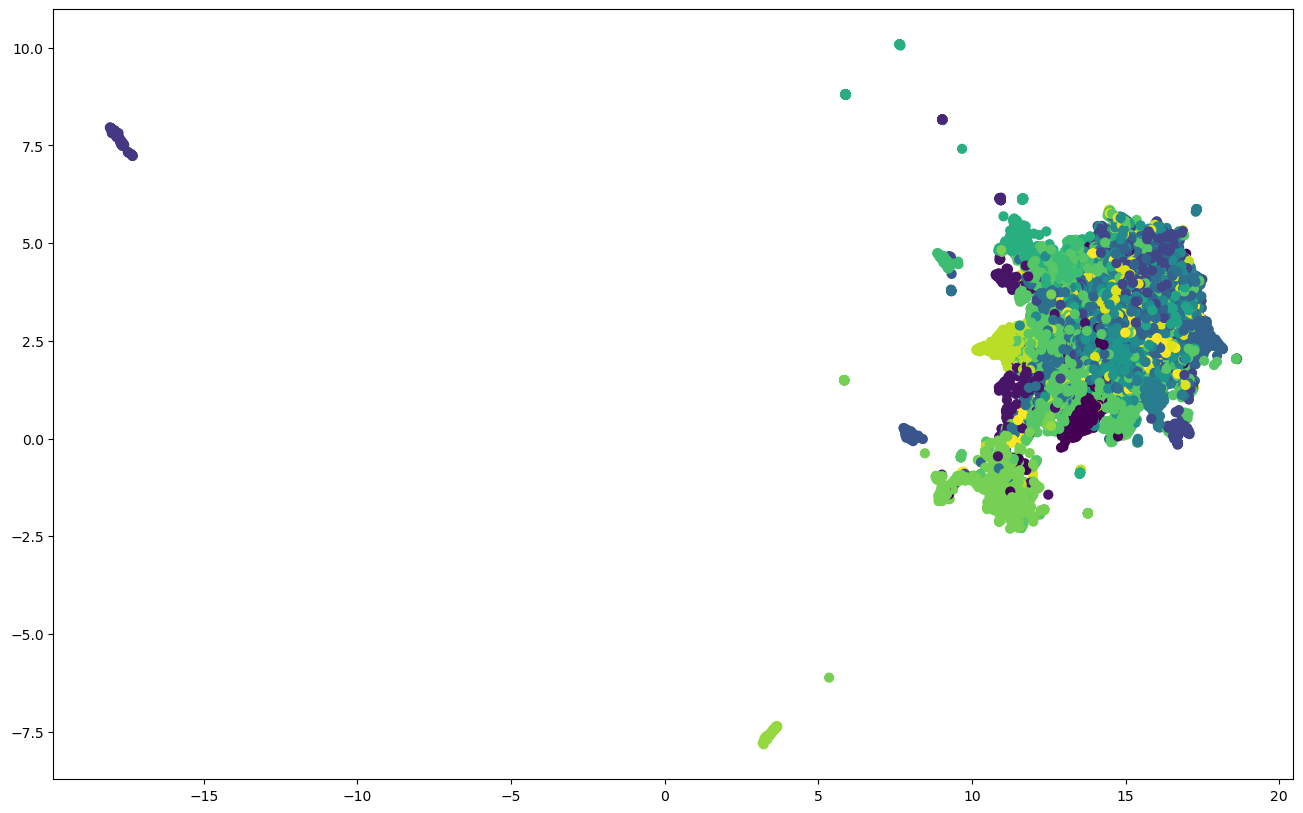


Рисунок 1 — Неудачная попытка посмотреть характер данных в датасете с помощью кластеризации по k-средних и визуализации через umap

# Сравнение

Специфика сравнения этих алгоритмов состоит в том, что, в отличие от предыдущей задачи, сложно автоматизировать оценку качества с помощью метрик, поскольку необходимо было бы найти датасет, содержащий правильные ответы о сходстве и различии предметов в наборе данных. Но выбранный набор данный позволяет, за счёт собственного игрового опыта, хотя бы примерно оценить качество работы алгоритма.

Сравнение времени работы алгоритмов тоже не имеет смысла, поскольку они порождают фактически одинаковый результат – квадратную матрицу размера, соответствующего количеству предметов в наборе данных.

Время на подготовку алгоритмов к работе:

* tf-idf:
  + косинусная мера: 28 c;
  + евклидово расстояние: 24 c;
* LDA:
  + косинусная мера: 111 c;
  + евклидово расстояние: 97 c.

Использованную в прошлой работе метрику корреляции представляется невозможным применить здесь, поскольку она не рассчитана на работу с разреженными матрицами, а без них, учитывая размеры матриц объектов, время работы будет существенно возрастать.

Что касается смыслового качества работы, то невозможно выделить явного лидера. tf-idf с косинусной мерой близости выдаёт действительно похожие игры, может быть, даже чересчур. LDA с той же мерой выдаёт более странные результаты, но некоторые рекомендации от этого алгоритма были бы интереснее и менее очевидны. С евклидовой мерой близости качество предложений от обоих алгоритмов падает, что, на мой взгляд, не компенсируется сниженным временем на настройку системы.

**Вывод**

В ходе лабораторной работы были сравнены алгоритмы коллаборативной фильтрации по пользователям и по предметам. Исходя из проведённого сравнения более стабильные результаты показывает подход, основанный на tf-idf – он позволяет достаточно быстро настроить систему и получить правильные, но очевидные предсказания. LDA же требует больше времени, имеет больше параметров, тонкая настройка которых позволит, по идее, получать рекомендации более высокого качества, чем tf-idf, но не так стабильно (например, только 2 игры из 10 предложенных с сомнительной похожестью остальных 8).