МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

**ОТЧЁТ**

по лабораторной работе

«Поиск частых наборов»

|  |  |
| --- | --- |
|  | Автор работы, студент группы КЭ-129  \_\_\_\_\_\_\_\_\_\_ Д.В. Кутюшкин  Проверил  \_\_\_\_\_\_\_\_\_\_\_\_\_ А.И. Гоглачев  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |

Челябинск, 2023 г.

ОГЛАВЛЕНИЕ

[ЗАДАНИЕ 3](#_Toc130562579)

[ПУНКТ 1 4](#_Toc130562580)

[ПУНКТ 2 6](#_Toc130562581)

[ПУНКТ 3 7](#_Toc130562582)

# ЗАДАНИЕ

1. Выполните поиск частых наборов объектов в трех различных наборах данных с помощью следующих алгоритмов (или их модификаций): Apriori, FP Growth, ECLAT. Наборы данных должны существенно отличаться друг от друга по количеству транзакций и/или типичной длине транзакции (количеству объектов). Варьируйте пороговое значение поддержки (например: 1%, 3%, 5%, 10%, 15%, 20%). Проверьте идентичность результатов, полученных с помощью различных алгоритмов.
2. Подготовьте список частых наборов, в которых не более семи объектов (разумное количество). Проанализируйте и изложите содержательный смысл полученного результата.
3. Выполните визуализацию полученных результатов в виде следующих диаграмм:
   * сравнение быстродействия алгоритмов на фиксированном наборе данных при изменяемом пороге поддержки;
   * общее количество частых наборов объектов на фиксированном наборе данных при изменяемом пороге поддержки;
   * максимальная длина частого набора объектов на фиксированном наборе данных при изменяемом пороге поддержки;
   * количество частых наборов объектов различной длины на фиксированном наборе данных при изменяемом пороге поддержки.
4. Подготовьте отчет о выполнении задания и загрузите отчет в формате PDF в систему. Отчет должен представлять собой связный и структурированный документ со следующими разделами:
   * формулировка задания;
   * гиперссылка на каталог репозитория с исходными текстами, наборами данных и др. сопутствующими материалами;
   * рисунки с результатами визуализации;
   * пояснения, раскрывающие смысл полученных результатов.

# ПУНКТ 1

В работе используются 3 набора данных:

1. kosarak;
2. retail;
3. market basket analysis.

В качестве реализаций алгоритмов в работе использовались библиотеки:

1. apyori – алгоритм Apriori;
2. pyECLAT – алгоритм ECLAT;
3. mlxtend – алгоритм FP Growth.

Проверим идентичность результатов работы алгоритмов на одном и том же наборе данных, выберем для этой цели набор данных «market basket analysis».



Рисунок 1 – Результат работы алгоритма Apriori при заданном пороге поддержки 15%



Рисунок 2 – Результат работы алгоритма FP Growth при заданном пороге поддержки 15%



Рисунок 3 – Результат работы алгоритма ECLAT при заданном пороге поддержки 15%

Судя по результатам можно сказать, что все алгоритмы находят одни и те же наборы при одинаковом пороге поддержки.

Найдем наборы для каждого датасета, варьируя значение порога поддержки. Также параллельно найдем наборы на фиксированном датасете (market basket analysis) и измерим время работы для каждого алгоритма, использовать будем те же значения порога поддержки. Эти результаты будут использованы в пунктах 2 и 3.

# ПУНКТ 2

Подготовим списки частых наборов, в которых разумное количество объектов (не более 7) для каждого набора данных. Порог поддержки выберем 20%.

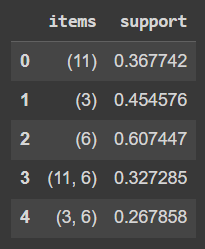


Рисунок 4 – Kosarak

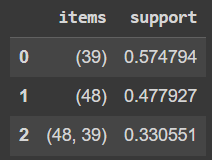


Рисунок 5 – Retail

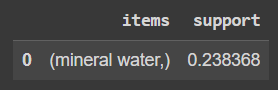


Рисунок 6 – Market basket analysis

Смысл полученных результатов заключается в том, что эти комбинации чаще всего встречаются среди всех транзакций, при этом частоту выражает величина поддержки, которая определяется как отношение числа транзакций, в которых имеется данный набор, к общему числу всех транзакций.

# ПУНКТ 3

Выполним визуализацию результатов в виде диаграмм.

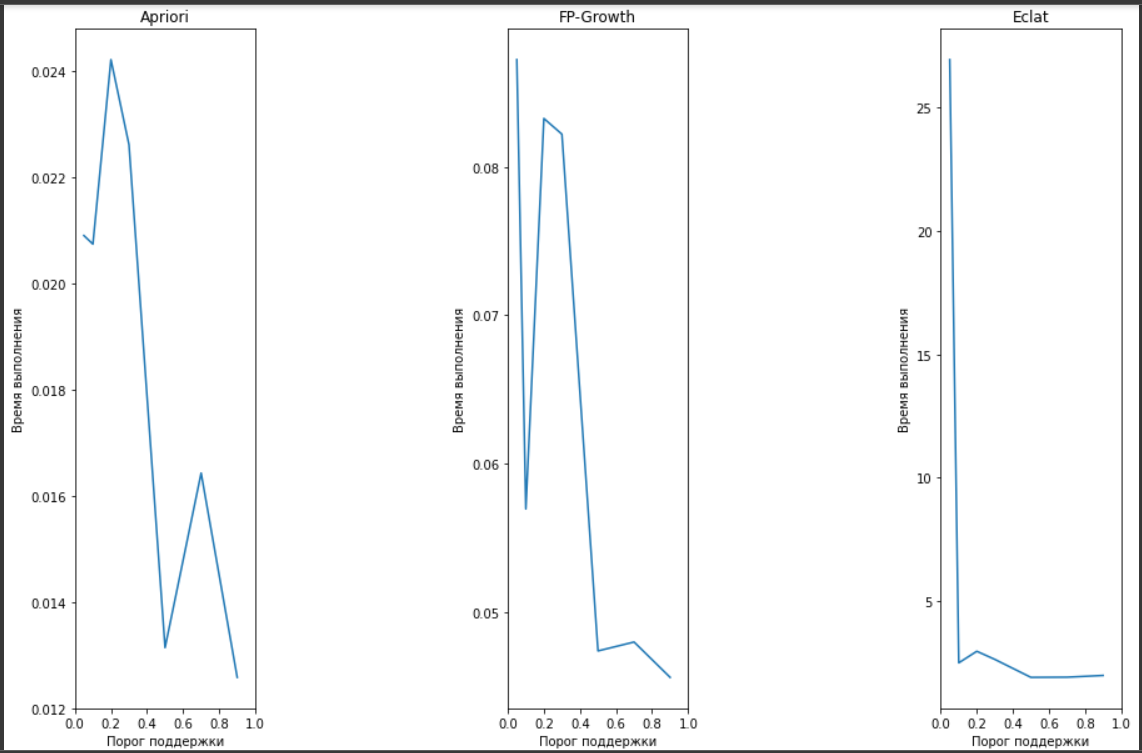


Рисунок 7 – Сравнение быстродействия алгоритмов на фиксированном наборе данных при изменяемом пороге поддержки

Самой быстрой является реализация алгоритма apriori, чуть медленнее является FP Growth, и на несколько порядков медленнее оказался ECLAT. Можно заметить, что при увеличении порога поддержки время выполнения всех алгоритмов уменьшается.

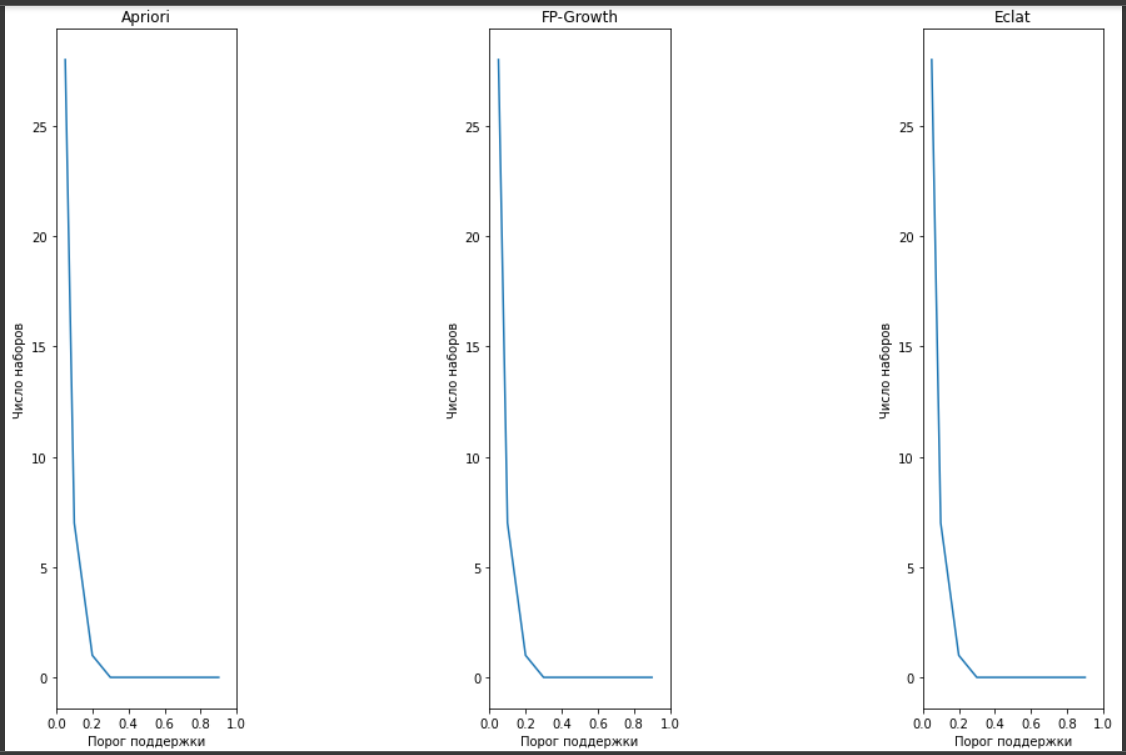


Рисунок 8 – Общее количество частых наборов объектов на фиксированном наборе данных при изменяемом пороге поддержки

Все алгоритмы находят одинаковое число наборов для всех значений порога поддержки. Видно, что чем выше порог поддержки, тем меньше будет найдено наборов.

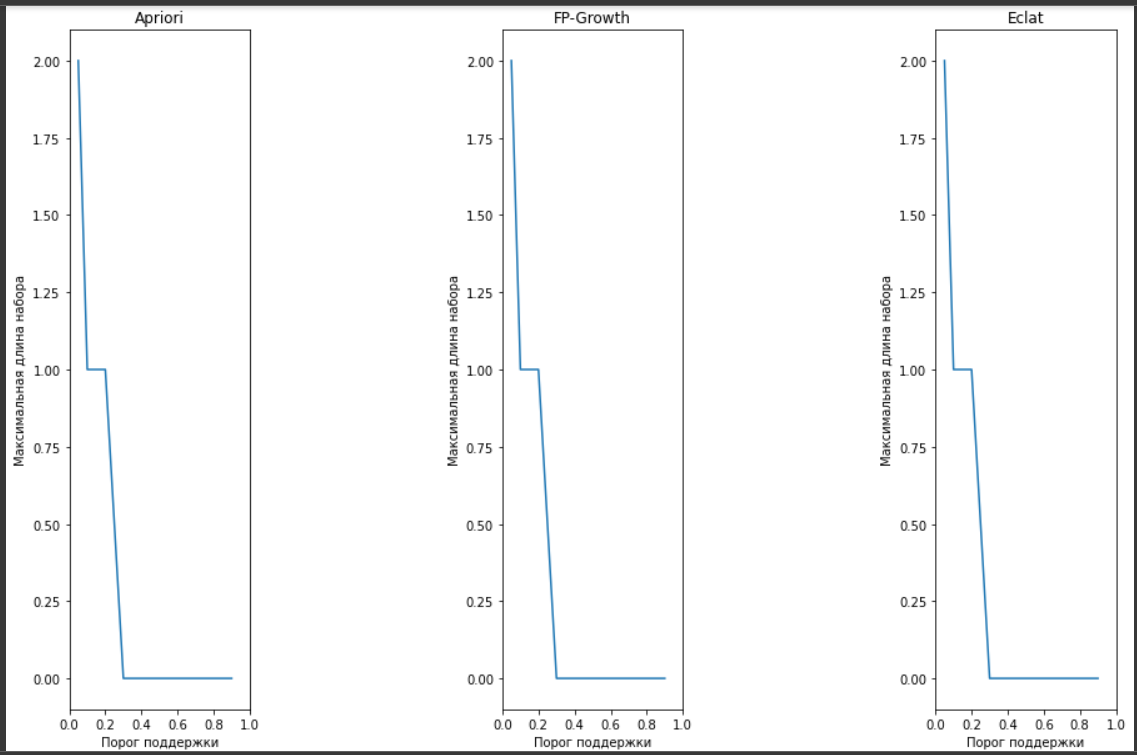


Рисунок 9 – Максимальная длина частого набора объектов на фиксированном наборе данных при изменяемом пороге поддержки

Т.к. алгоритмы находят одни и те же наборы, максимальные длины наборов также будут одинаковы. При этом чем выше порог поддержки, тем меньшей длины алгоритмы находят наборы.

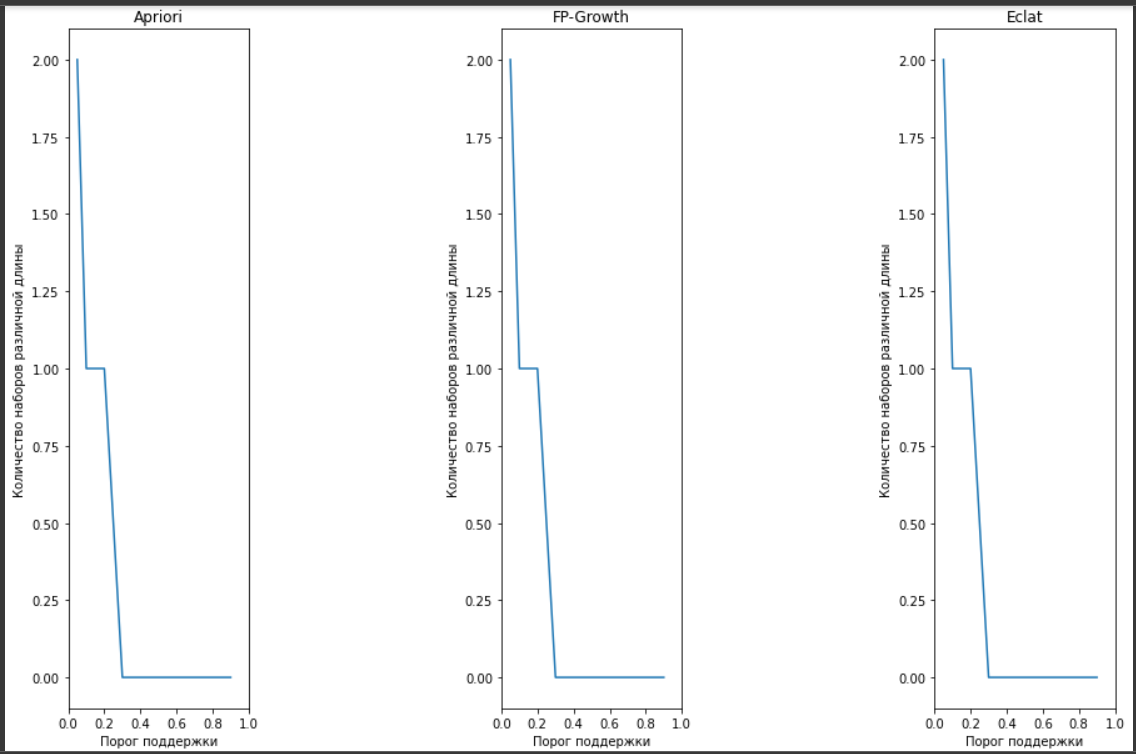


Рисунок 10 – Количество частых наборов объектов различной длины на фиксированном наборе данных при изменяемом пороге поддержки

Видно, что графики на рисунках 9 и 10 идентичны. Из этого можно сделать вывод, что если был найден набор длины n, то будет найден набор и длины n-1 и т.д.

Ссылка на репозиторий с исходным кодом и наборами данных: