**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

**“Большие данные и распределенная цифровая платформа”**

**ОТЧЕТ**

**по лабораторной работе №5**

**по дисциплине «Алгоритмы и структуры данных»**

**на тему «Кластеризация данных»**

**Студент гр. 23Б16-пу**

**Горынцев Р.Н.**

**Преподаватель**

**Дик А.Г.**

**Санкт-Петербург**

**2025 г.**

Оглавление

1. [**Цель работы** 3](#_Toc198944415)
2. [**Описание задачи (формализация задачи)** 3](#_Toc198944416)
3. [**Теоретическая часть** 4](#_Toc198944417)
4. [**Основные шаги программы** 6](#_Toc198944418)
5. [**Блок схема программы** 7](#_Toc198944419)
6. [**Описание программы** 8](#_Toc198944420)
7. [**Рекомендации пользователя** 9](#_Toc198944421)
8. [**Контрольный пример** 10](#_Toc198944422)
9. [**Анализ результатов работы** 16](#_Toc198944423)
10. [**Вывод** 21](#_Toc198944424)

# **Цель работы**

Целью данной лабораторной работы является исследование влияния отбора наиболее информативных признаков и обезличивания данных на качество кластеризации алгоритмом ISODATA. Сравнить результаты кластеризации исходных данных, отобранных и обезличенных данных. Сделать вывод.

# **Описание задачи (формализация задачи)**

1. **Подготовка данных**Сформировать или найти реальный датасет, содержащий в себе больше 15 столбцов
2. **Кластеризация и оценка**Выполнить кластеризацию алгоритмом ISODATA, используя в качестве способа измерения расстояния – расстояние Чебышева. Оценить качество кластеризации методом отделимости кластеров.
3. **Отбор информативных признаков**Применить алгоритм последовательного добавления (алгоритм Аdd) для выбора наиболее значимых признаков.
4. **Повторная кластеризация**

Выполнить ISODATA на данных с отобранными признаками.

1. **Обезличивание данных**

Обезличить датасет: добавление гауссова шума к числовым признакам, хэширование SHA-256 с солью для категориальных данных.

1. **Кластеризация обезличенных данных**
2. **Сравнить качество кластеризации и сделать выводы.**

# **Теоретическая часть**

1. **Кластеризация методом ISODATA**

ISODATA (Iterative Self-Organizing Data Analysis Technique) – это итеративный алгоритм кластеризации, расширяющий возможности k-means за счёт автоматической адаптации числа кластеров. В отличие от k-means, где количество кластеров задаётся заранее, ISODATA динамически объединяет и разделяет кластеры на основе параметров:

* – минимальное число точек в кластере
* – максимальное стандартное отклонение
* – минимальное расстояние между центрами кластеров

1. **Cпособ измерения расстояния**

Для кластеризации используется расстояние Чебышева:

Применение в ISODATA:

* Определение принадлежности точки к кластеру
* Расчёт внутрикластерной дисперсии
* Объединение близких кластеров

1. **Выбор наиболее информативные признаков**

Алгоритм последовательного добавления – метод обёртки, который пошагово выбирает признаки, максимизирующие качество кластеризации.

Этапы работы:

1. Начинает с пустого набора признаков.
2. На каждой итерации добавляет признак, дающий максимальный прирост результата.
3. Останавливается при достижении заданного числа признаков или отсутствии улучшения.

Критерий оценки:

где

a –внутрикластерное расстояние,

*b*–межкластерное расстояние.

1. **Оценка качества кластеризации**

Метод отделимости кластеров:

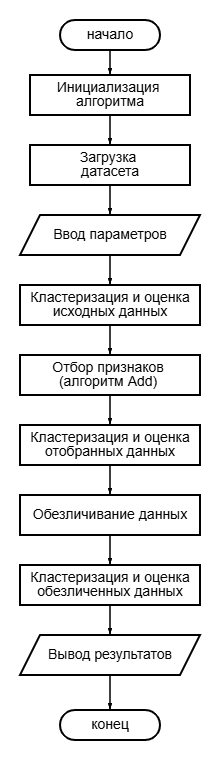
Интерпретация:

* Более 1.5 – отличная разделимость
* 0.8–1.5 – удовлетворительная
* Менее 0.8 – кластеры перекрываются

# **Основные шаги программы**

* 1. **Инициализация интерфейса и загрузка датасета**
* Создаётся интерфейс с вкладками, кнопками, таблицами и холстом.
* Пользователь выбирает CSV/Excel файл, нажав на кнопку "Загрузить датасет".
* Набор данных загружается с помощью Pandas.
  1. **Обезличивание данных**
* При нажатии на кнопку "Обезличить данные" идентификаторы и категориальные признаки хешируются.
* Результаты отображаются в текстовой области, а данные сохраняются для дальнейшего использования.
  1. **Отбор наиболее информативных признаков**
* Пользователь нажимает кнопку "Выбрать признаки (Add)" для запуска алгоритма последовательного добавления признаков.
* Наиболее значимые признаки выводятся в текстовое поле.
  1. **Кластеризация данных**
* Пользователь выбирает тип данных (исходные, отобранные признаки или обезличенные) и запускает алгоритм ISODATA.
* Параметры кластеризации задаются через интерфейс.
* Результаты кластеризации отображаются в текстовой области.
  1. **Оценка качества кластеризации**
* При нажатии на кнопку "Оценить качество" вычисляется индекс качества на основе внутрикластерных и межкластерных расстояний.
* Результат оценки выводится в текстовое поле и на соответствующую метку в интерфейсе.

# **Блок схема программы**



*Рис 1: Блок схема программы*

# **Описание программы**

Программная реализация выполнена на языке Python 3.12 с использованием библиотек tkinter, pandas, numpy, hashlib. Программа организована в виде единого файла, основная цель которого — кластеризация данных. В рамках программы реализован класс DatasetViewer, который объединяет всю логику работы с графическим интерфейсом. В процессе разработки программы использовались функции, каждая из которых имеет чётко определённое назначение:

*Таблица 1. Описание функций*

|  |  |  |
| --- | --- | --- |
| Функция | Описание | Возвращаемое значение |
| \_\_init\_\_ | Инициализирует графический интерфейс), создаёт атрибуты для хранения датасетов и параметров кластеризации. | None |
| setup\_ui | Настраивает интерфейс: кнопки управления, поля параметров ISODATA, текстовую область вывода и информационные метки. | None |
| load\_dataset | Загружает датасет из CSV/Excel-файла, преобразует категориальные признаки (One-Hot Encoding). | None |
| anonymize\_data | Обезличивает данные: хеширует идентификаторы (student\_id) и категориальные признаки с помощью SHA-256. | None |
| prepare\_data | Подготавливает данные для кластеризации: удаляет ID, применяет One-Hot Encoding к категориальным признакам. | None |
| optimized\_add\_feature\_selection | Реализует алгоритм последовательного отбора признаков (Add). Оценивает значимость каждого признака. | None |
| run\_isodata | Запускает кластеризацию алгоритмом ISODATA. | None |
| isodata\_algorithm | Динамически изменяет число кластеров, объединяет/разделяет их на основе параметров (мин. точек, макс. дисперсии). | tuple (число кластеров) |
| evaluate\_clusters | Оценивает качество кластеризации | float |
| display\_clustering\_results | Выводит датасет с метками кластеров в текстовое поле и обновляет метку числа кластеров. | None |
| update\_data\_labels | Обновляет цветовые индикаторы в интерфейсе (тип данных, признаки) в зависимости от текущего состояния. | None |
| show\_error | Отображает сообщение об ошибке в текстовом поле. | None |
| reset\_state | Сбрасывает состояние программы: очищает выбранные признаки, метки кластеров и оценки. | None |

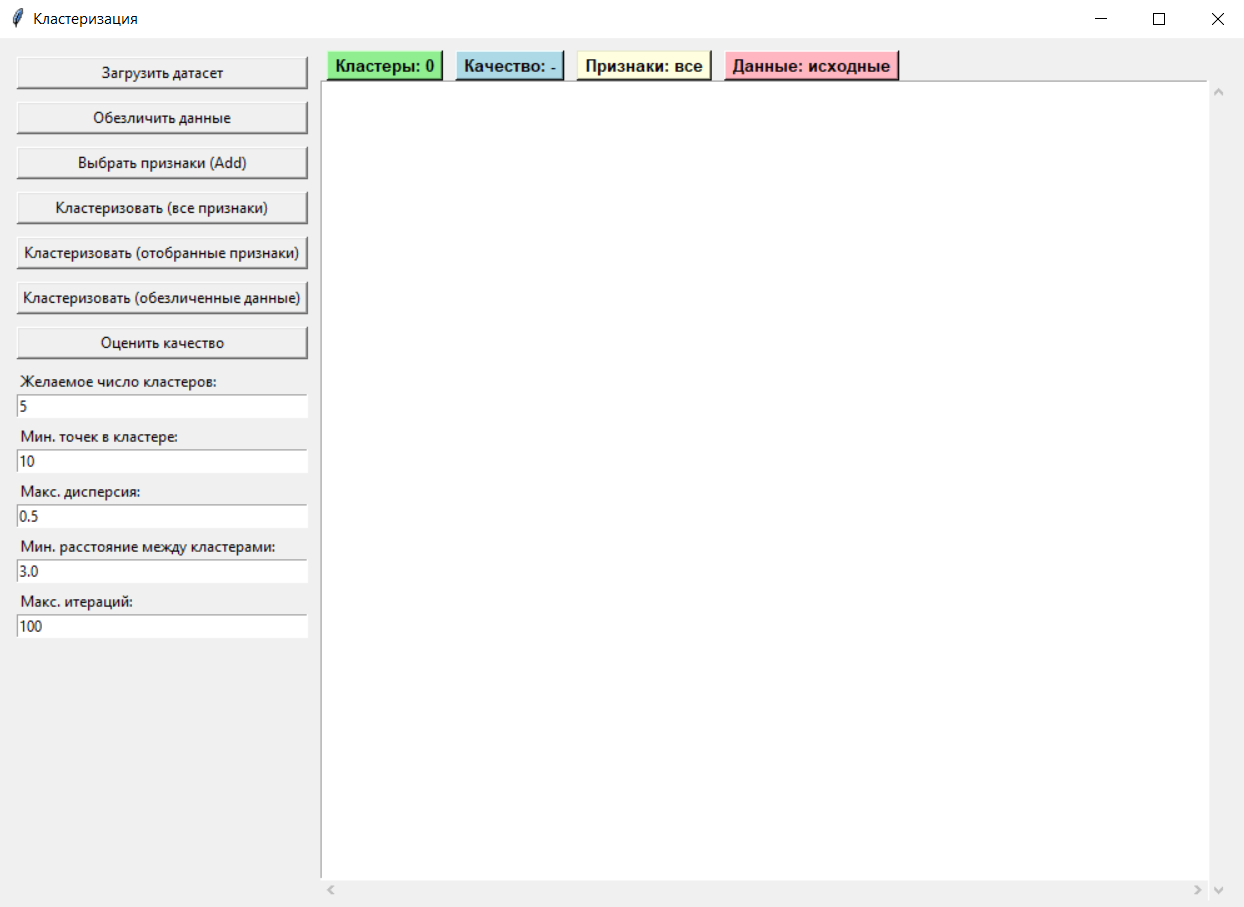
# **Рекомендации пользователя**

1. Убедитесь, что у вас установлен Python 3.12 и библиотеки tkinter, pandas, numpy, hashlib.
2. Запустите программу.
3. Нажмите кнопку "Загрузить датасет" и выберите файл в формате .xlsx или .csv.
4. Нажмите кнопку **"Обезличить данные",** чтобы хешировать идентификаторы и категориальные признаки.
5. Нажмите "Выбрать признаки (Add)", чтобы запустить алгоритм отбора топ-признаков.
6. Настройте параметры кластеризации в правой панели и запустите кластеризацию.
7. Нажмите "Оценить качество", чтобы получить оценку кластеризации.

# **Контрольный пример**

**1. Запуск программы:**

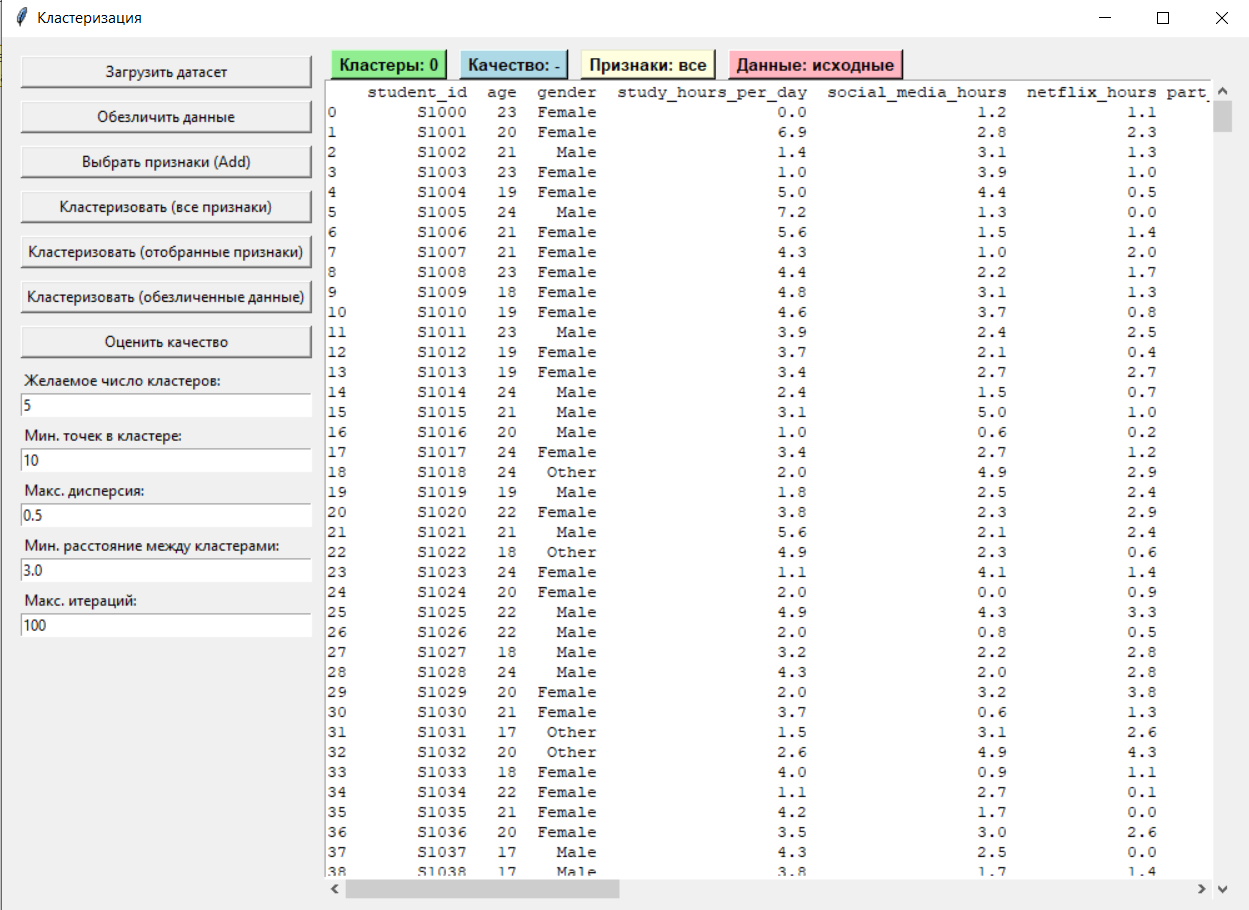
* Пользователь запускает программу. На экране появляется главное окно интерфейса с возможностью загрузки датасета, обезличивания данных, отбора признаков, кластеризации, оценки качества и областью для вывода результатов (Рис. 2).



*Рис 2: Интерфейс программы*

**2. Загрузка датасета и анализ статистики:**

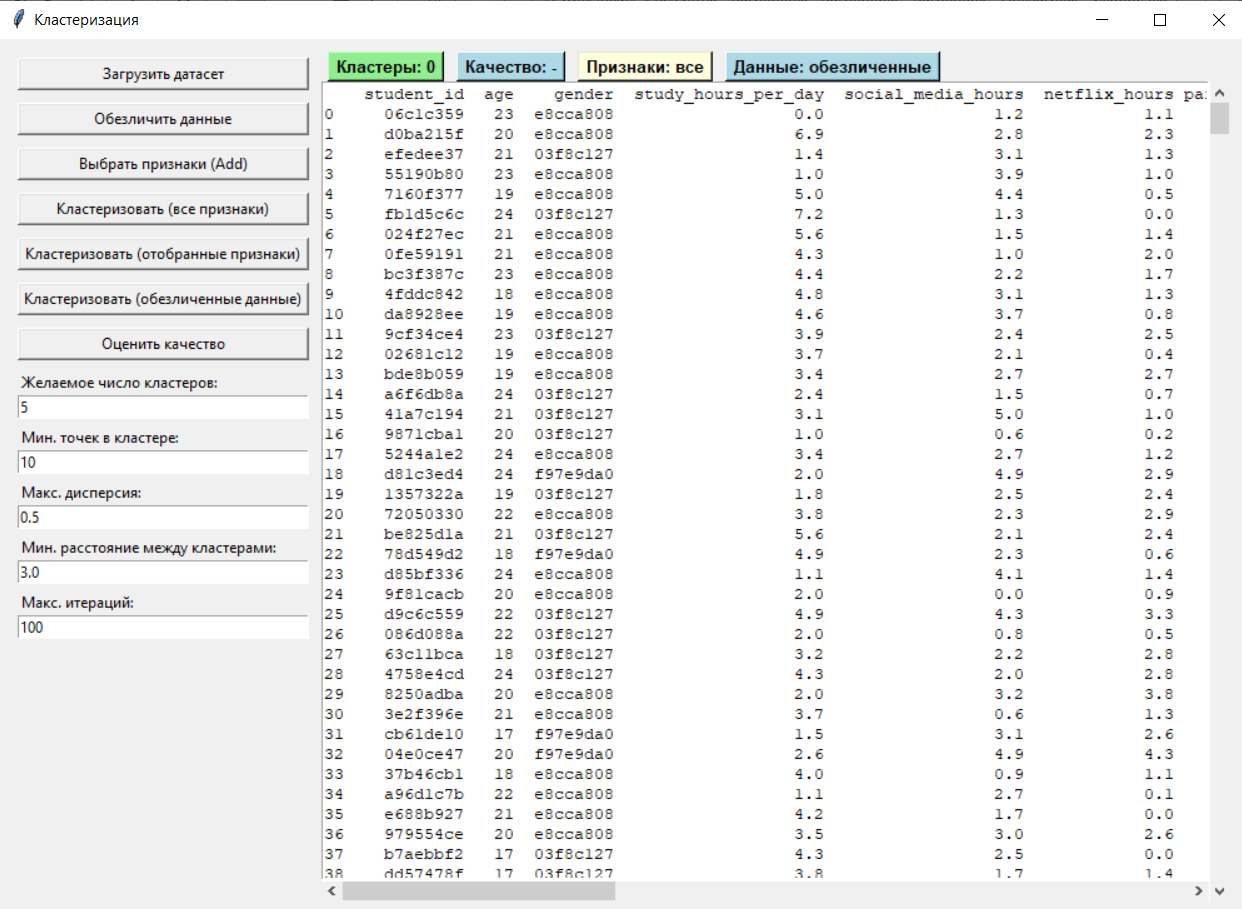
* Пользователь нажимает кнопку "Загрузить датасет" и в открывшемся окне выбирает файл.
* Датасет успешно отобразится в окне.



*Рис 3: Загрузка датасета*

**3. Обезличивание данных:**

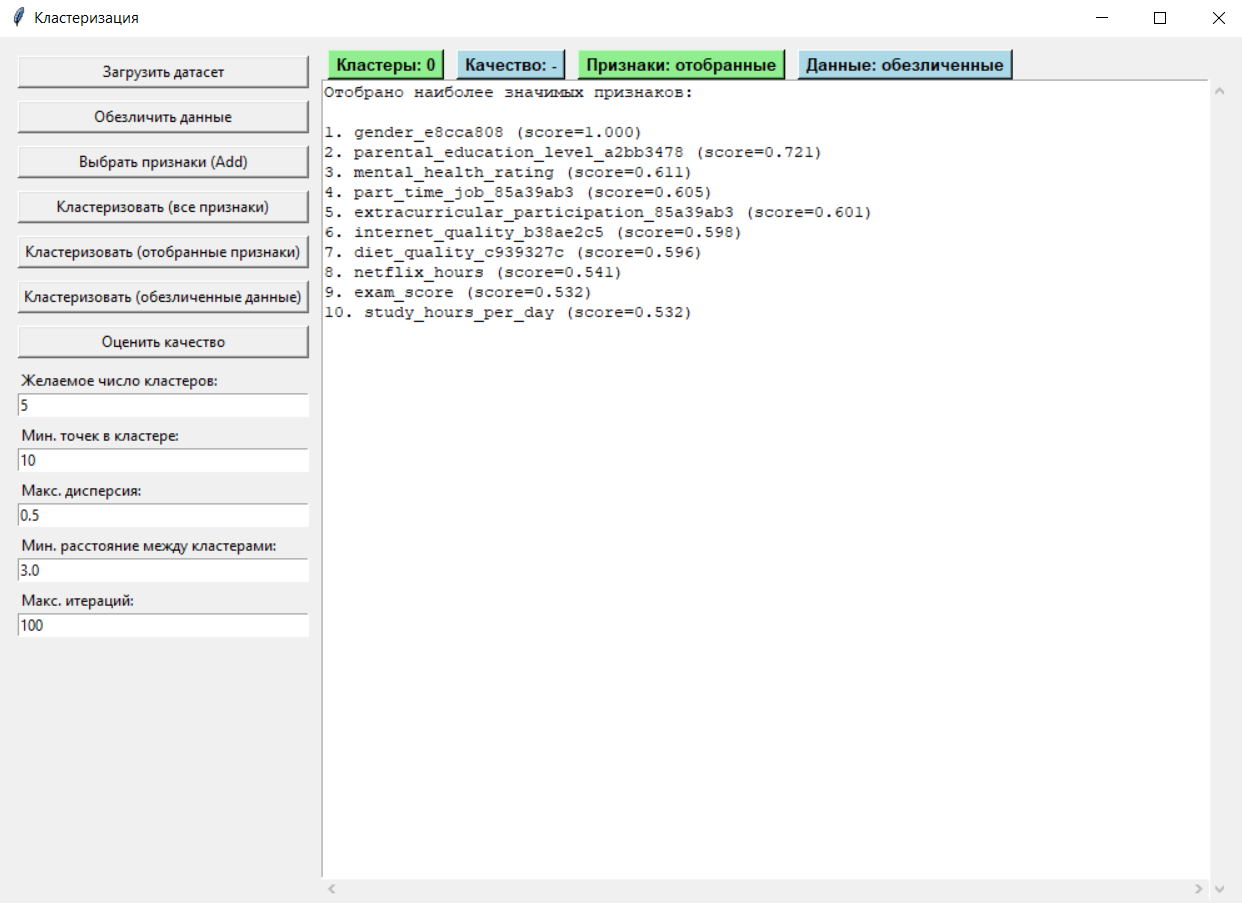
* Пользователь нажимает "Обезличить данные".
* В текстовом поле появятся хешированные значения (Рис. 4).



*Рис 4: пример обезличивания*

**4. Отбор признаков:**

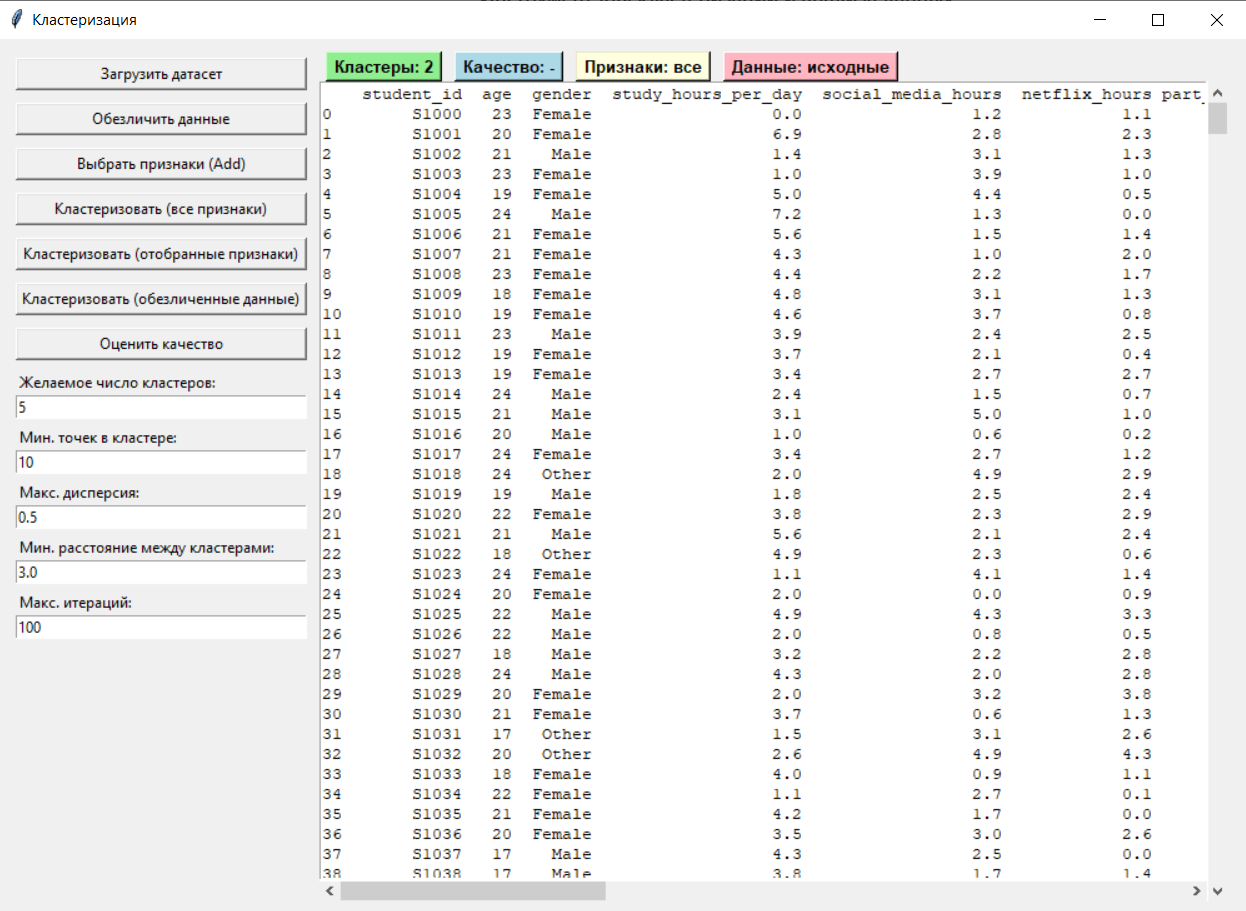
* Пользователь выбирает кнопку **"Выбрать признаки (Add)".**
* **Программа выведет 10 наиболее информативных признаков (Рис. 5).**



*Рис 5: пример отбора признаков*

**5. Кластеризация:**

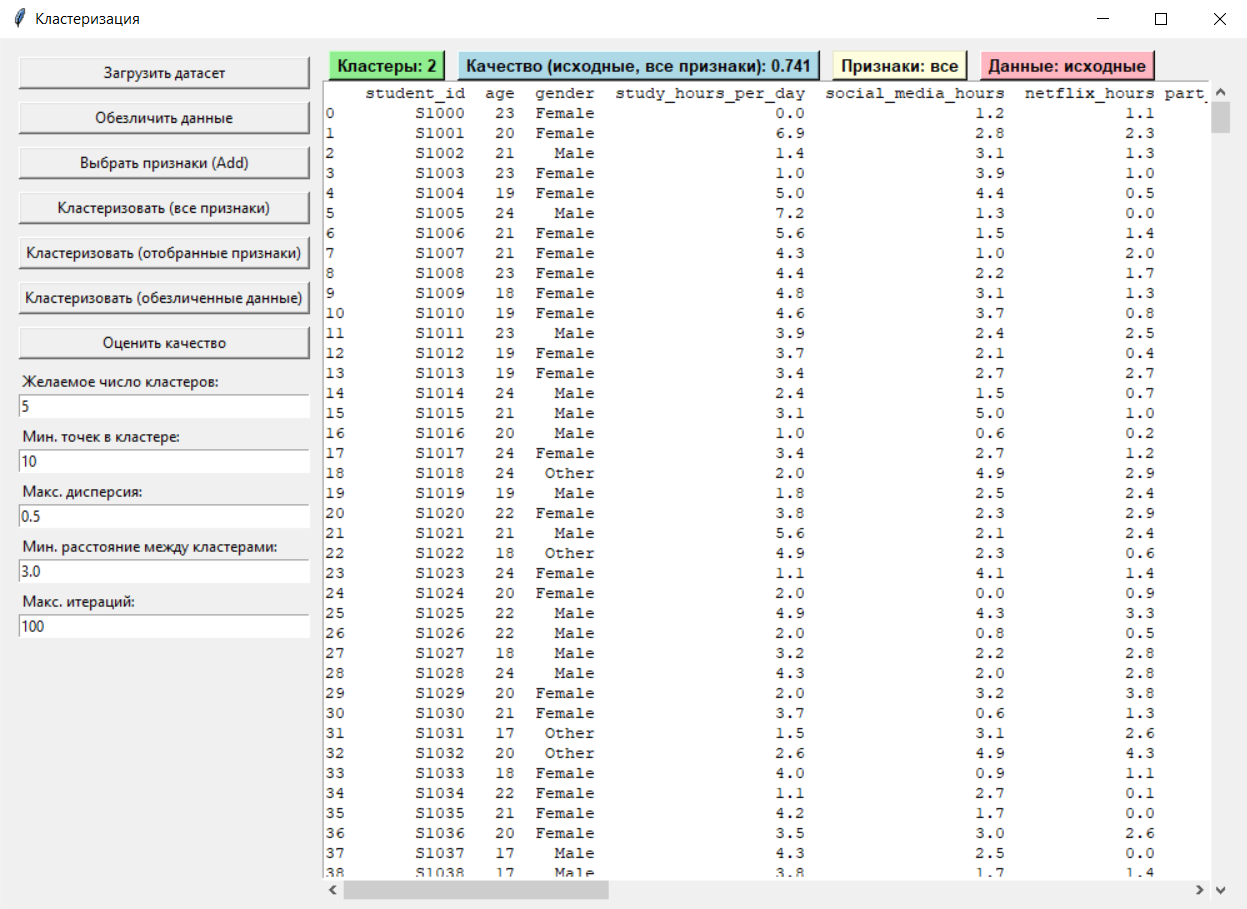
* Пользователь устанавливает параметры:
* Желаемое число кластеров
* Мин. точек в кластере
* Макс. дисперсия
* Мин. Расстояние между кластерами
* Макс. итераций
* Нажимает на кнопку **"Кластеризовать" в зависимости от того, какие признаки нужно кластеризовать (Рис. 6).**



*Рис 6: пример кластеризации*

**6. Оценка качества:**

* После нажатия "Оценить качество" в текстовом поле отображается результат (Рис. 7)



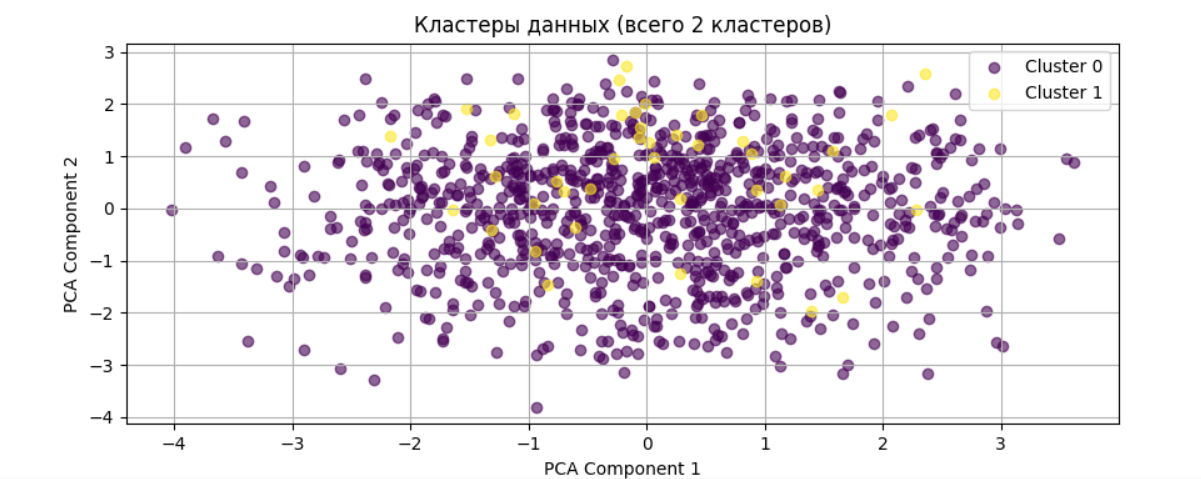
*Рис 7: пример оценки качества*

# **Анализ результатов работы**

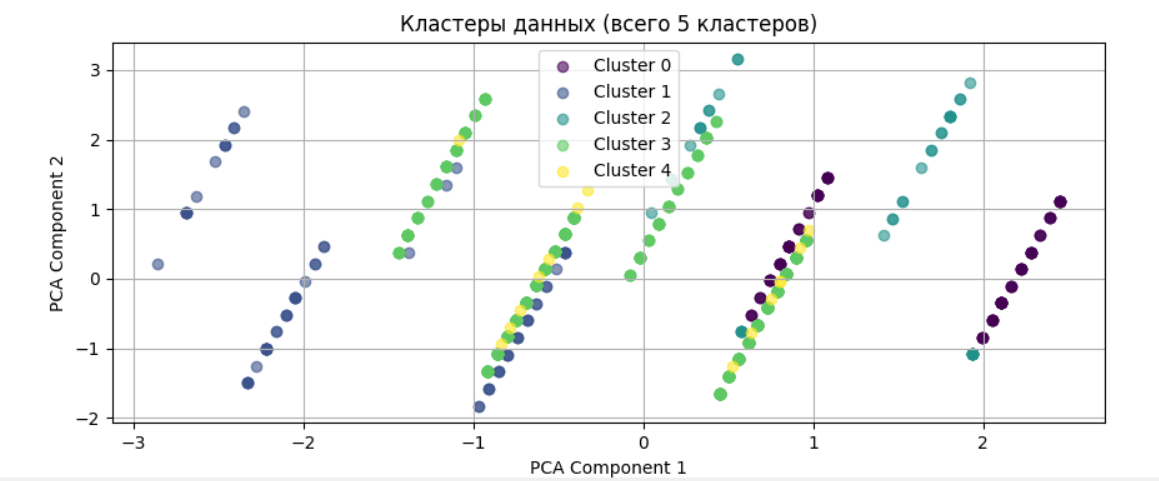
Проведём анализ на различных параметрах кластеризации:

*Таблица 2. Анализ результатов*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметры  кластеризации | Количество  информативных  признаков | Оценка исходных  признаков | Оценка отобранных  признаков | Оценка обезличенных  признаков |
| * Желаемое число кластеров: 2 * Мин. точек в кластере: 3 * Макс. Дисперсия: 0,6 * Мин. Расстояние между кластерами: 1,0 * Макс. Итераций: 80 | 3 | 0,258 | 0,629 | 0,318 |
| * Желаемое число кластеров: 3 * Мин. точек в кластере: 5 * Макс. Дисперсия: 0,5 * Мин. Расстояние между кластерами: 1,5 * Макс. Итераций: 100 | 5 | 0,697 | 0,689 | 0,741 |
| * Желаемое число кластеров: 4 * Мин. точек в кластере: 5 * Макс. Дисперсия: 0,4 * Мин. Расстояние между кластерами: 1,8 * Макс. Итераций: 120 | 7 | 0,574 | 0,596 | 0,741 |
| * Желаемое число кластеров: 5 * Мин. точек в кластере: 4 * Макс. Дисперсия: 0,3 * Мин. Расстояние между кластерами: 1,2 * Макс. Итераций: 150 | 10 | 0,287 | 0,544 | 0,328 |
| * Желаемое число кластеров: 6 * Мин. точек в кластере: 4 * Макс. Дисперсия: 0,25 * Мин. Расстояние между кластерами: 1,0 * Макс. Итераций: 180 | 12 | 0,217 | 0,233 | 0,226 |
| * Желаемое число кластеров: 7 * Мин. точек в кластере: 3 * Макс. Дисперсия: 0,2 * Мин. Расстояние между кластерами: 0,8 * Макс. Итераций: 200 | 15 | 0,231 | 0,229 | 0,251 |
| * Желаемое число кластеров: 8 * Мин. точек в кластере: 3 * Макс. Дисперсия: 0,15 * Мин. Расстояние между кластерами: 0,6 * Макс. Итераций: 220 | 10 | 0,221 | 0,259 | 0,225 |
| * Желаемое число кластеров: 5 * Мин. точек в кластере: 5 * Макс. Дисперсия: 0,35 * Мин. Расстояние между кластерами: 1,5 * Макс. Итераций: 150 | 8 | 0,697 | 0,562 | 0,741 |



*Рис 8: пример визуализации кластеризации исходных данных*



*Рис 9: пример визуализации кластеризации отобранных данных*



*Рис 10: пример визуализации кластеризации обезличенных данных*

* **Оптимальное число кластеров**  
  Наилучшие оценки качества (особенно для обезличенных признаков) наблюдаются при **3, 4 и 5 кластерах** (оценки: 0.741, 0.741, 0.741 и 0.328). Это говорит о том, что данные хорошо разделяются на 3–5 групп. Увеличение числа кластеров свыше 5 приводит к снижению качества (например, для 6–8 кластеров оценки падают до 0.226–0.251).
* **Влияние параметров кластеризации**
  + **Макс. дисперсия**: Более строгие значения (0.2–0.4) дают лучшие результаты, особенно для 3–5 кластеров. При увеличении дисперсии выше 0.5 качество ухудшается.
  + **Мин. расстояние между кластерами**: Оптимальный диапазон — 1.5–1.8. Слишком малые значения (0.6–1.0) снижают четкость разделения.
  + **Мин. точек в кластере:** Лучшие результаты при 5 точках, что указывает на устойчивость кластеров. Меньшие значения (3–4) могут приводить к шуму.
* **Отбор признаков**
  + **Информативные признаки**: Наибольшее количество (15) не гарантирует высокого качества (оценка 0.229–0.251 для 7 кластеров). Лучшие результаты достигаются при 5–10 признаках.
  + **Оценка признаков**: Обезличенные признаки почти всегда показывают более высокие оценки (например, 0.741 против 0.689 для 3 кластеров), что подчеркивает важность предварительной обработки данных (например, нормализации или PCA).
* **Итерации**  
  Увеличение числа итераций (до 150–200) не всегда улучшает результат, особенно если другие параметры (дисперсия, расстояние) не оптимальны. Например, для 8 кластеров при 220 итерациях оценка всего 0.225.
* **Итог**
* **Оптимальное число кластеров (3-5)** - даёт наивысшие оценки качества (до 0.741)
* **Строгие параметры кластеризации**:
  + Макс. дисперсия 0.2-0.4
  + Мин. расстояние между кластерами 1.5-1.8
  + Мин. 5 точек в кластере
* **Предварительная обработка данных**:
  + Обезличивание признаков существенно улучшает результаты
  + Отбор 5-10 наиболее информативных признаков
* **Умеренное число итераций (80-150)** - достаточно для сходимости алгоритма

# **Вывод**

В ходе лабораторной работы было исследовано влияние отбора информативных признаков и обезличивания данных на качество кластеризации с использованием алгоритма ISODATA. Основной задачей являлось сравнение результатов кластеризации исходных, отобранных и обезличенных данных.

Отбор наиболее информативных признаков (5-10 показателей) позволил повысить точность кластеризации за счет устранения шумовых и избыточных переменных. В то же время процедура обезличивания, включавшая добавление гауссова шума к числовым данным и хеширование категориальных признаков, обеспечила лучшую разделимость кластеров.

Наибольшая эффективность алгоритма ISODATA наблюдалась при следующих параметрах: количество кластеров 3-5, максимальная дисперсия 0.2-0.4, минимальное расстояние между центрами кластеров 1.5-1.8. Эти настройки обеспечивали оптимальный баланс между точностью группировки и устойчивостью результатов.

Полученные результаты подчеркивают важность выбора методов предварительной обработки данных в зависимости от решаемой задачи.