**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. 23Б15-пу**

**Беляева А.П.**

**Преподаватель**

**Дик А.Г.**

**Санкт-Петербург**

**2025 г.**

**Оглавление**

Цель работы....................................................................................................................3

Формализация задачи....................................................................................................6

Схема пошагового выполнения алгоритма........................................................................................................................5

Описание программы....................................................................................................6

Рекомендации пользователя.........................................................................................9

Рекомендации для программиста.................................................................................10

Исходный код программы......................................................................................…...10

Контрольный пример....................................................................................................11

Анализ работы алгоритма.......................................................................................................................13

Вывод..............................................................................................................................15

Источники......................................................................................................................15

**Цель работы**

Целью проекта является исследование влияния различных методов предобработки данных (обезличивание, отбор признаков) на качество кластеризации.

**Описание задачи (формализация задачи)**

*Основные задачи:*

1. Сравнение качества кластеризации исходных и обезличенных данных.
2. Оценка эффективности отбора информативных признаков.
3. Визуализация результатов для анализа структуры кластеров.

*Входные данные:*

Датасет с 15+ признаками, включая персональные данные (ФИО, email, телефон), числовые (возраст, стоимость) и категориальные (тип вагона, маршрут) признаки.

*Ход работы:*

1. Сформировать датасет (15 столбцов и более)
2. Выполнить кластеризацию
3. Оценить качество кластеризации
4. Выбрать наиболее информативные признаки
5. Выполнить кластеризацию
6. Оценить качество кластеризации
7. Сравнить результаты п.3 и п.6. Сделать выводы
8. Обезличить датасет
9. Выполнить кластеризацию
10. Оценить качество кластеризации
11. Сравнить результаты п.3, п.6. и п.10. Сделать выводы

*Выходные данные:*

1. Значения компактности для каждого этапа.
2. Графики PCA и гистограммы распределения расстояний до центроидов.

**Теоретическая часть**

Кластеризация — это метод обучения без учителя, целью которого является группировка объектов в кластеры таким образом, чтобы:

* Объекты внутри одного кластера были максимально похожи друг на друга.
* Объекты разных кластеров — максимально отличались.

*Алгоритм CURE (Clustering Using Representatives)*

Особенности алгоритма:

* Использует иерархический подход для построения кластеров.
* Каждый кластер описывается несколькими представителями (representative points), а не центроидом.
* Устойчив к кластерам произвольной формы и шуму.

*Этапы работы CURE:*

1. Инициализация:

- Каждая точка данных считается отдельным кластером.

1. Слияние кластеров.

- На каждом шаге объединяются два ближайших кластера на основе расстояния между их представителями.

1. Сжатие (Compression):

- Для уменьшения вычислительной сложности представители кластеров "приближаются" к центроиду с заданным коэффициентом сжатия.

1. Остановка:

- Процесс завершается при достижении заданного числа кластеров.

*Параметры алгоритма:*

*number\_cluster* — итоговое число кластеров.

*number\_represent\_points* — количество представителей на кластер (обычно 5-10).

*compression* — коэффициент сжатия (0.1-0.5).

*Отбор признаков*

Цель: Уменьшение размерности данных для:

- Ускорения вычислений.

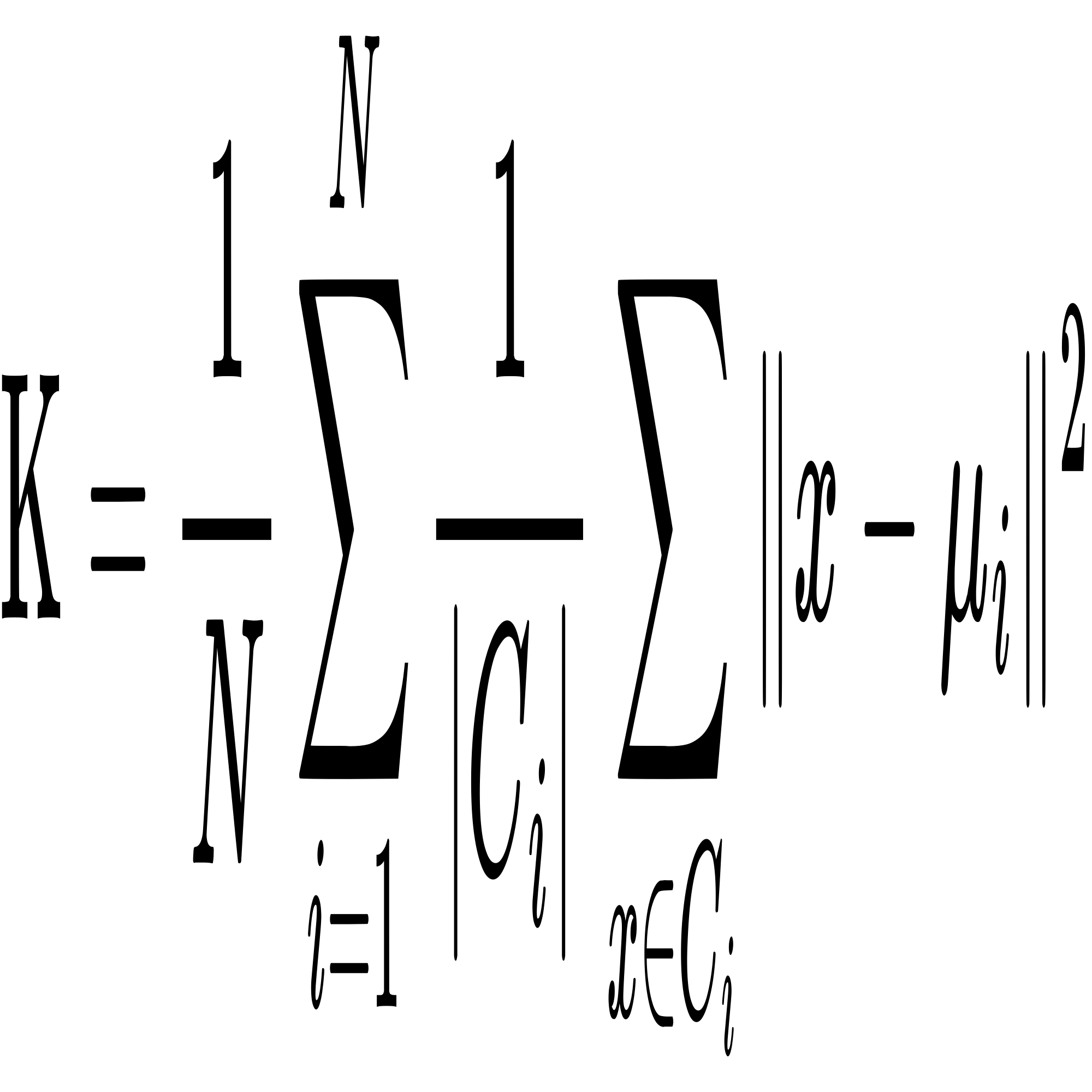
- Устранения шума и избыточности.

- Улучшения интерпретируемости.

Метод, использованный в решении:

1. Для каждого признака выполняется одномерная кластеризация алгоритмом CURE.
2. Признаки ранжируются по компактности (чем меньше, тем лучше).
3. Выбираются NN признаков с наименьшей компактностью.

Формула компактности:



(1)

где:

N - число кластеров,

∣Ci∣- размер кластера i,

μi​ — центроид кластера i.

*Оценка качества кластеризации:* Метрика компактности

Определение: Среднее квадратичное расстояние от точек до центроида своего кластера.

Интерпретация: Чем меньше значение, тем плотнее кластеры.

Нулевая компактность означает, что все точки совпадают с центроидом

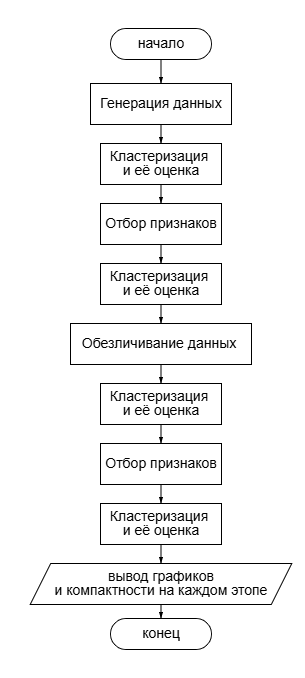
**Схема пошагового выполнения алгоритма**

Рис. 1. Блок-схема основной программы

**Описание программы**

Программа реализована на языке Python. Графический интерфейс построен с использованием tkinter. Визуализация графиков производится библиотеками matplotlib. Работа с данными реализована через pandas, а кластеризация и предобработка — при помощи scikit-learn.

**Основные компоненты программы:**

Таблица 1. ClusteringApp.py.

|  |  |  |
| --- | --- | --- |
| **Название** | **Назначение** | **Тип возвращаемых данных** |
| \_\_init\_\_ | Создание интерфейса | None |
| generate\_data | Предобработка данных | None |
| create\_widgets | Добавление кнопок | None |
| start\_clustering | Запуск кластеризации | None |
| run\_clustering | Реализация кластеризации | None |
| select\_features | Отбор признаков | list |
| calculate\_compactness | Вычисление компактности | None |
| anonymize\_data | Анонимизация датасета | None |
| prepare\_data | Подготовка данных | Any |
| run\_cure | Реализация метода CURE | Any |

# **Рекомендации для пользователя**

# Пошаговая инструкция по использованию программы:

# Запуск программы: *python clustering\_app.py*

1. Сгенерируйте датасет, нажав кнопку
2. Обезличьте его

# Настройки:

# Укажите число кластеров (по умолчанию: 5).

# Выберите количество признаков для отбора (по умолчанию: 5).

# Интерпретация результатов:

# Низкая компактность (близкая к 0) — высокая качество кластеризации.

# Высокая объяснённая дисперсия PCA (>0.7) — хорошее сохранение структуры данных.

# **Рекомендации для программиста**

1. Установите необходимые библиотеки: tkinter, matplotlib, pandas, scikit-learn
2. При реализации интерфейса важно обрабатывать исключения: ошибки чтения файла, пустые поля, неверные типы данных.
3. Структура кода должна позволять добавление новых методов восстановления (например, методом ближайших соседей).
4. Логика восстановления и визуализации должна быть инкапсулирована в отдельные функции/модули.
5. Для масштабируемости можно использовать объектно-ориентированную архитектуру с интерфейсами для методов восстановления.

**Исходный код программы**

**[Git](https://github.com/hysterria/ClusteringApp.git)**

**Контрольный пример**

1. Запуск программы, управление:

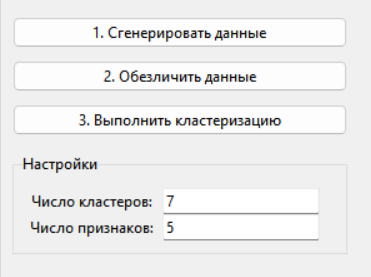


Рис. 2. Панель управления.

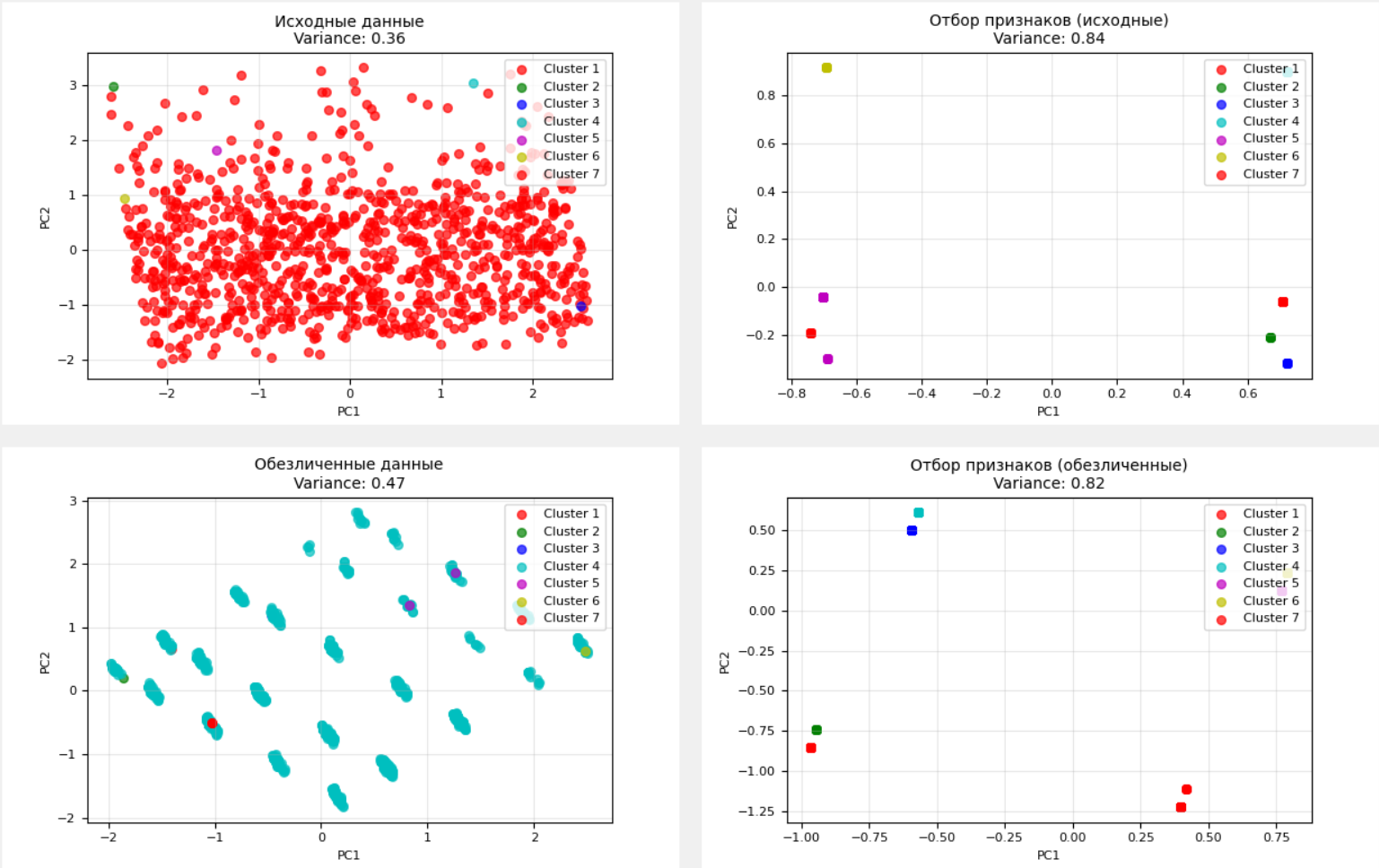
1. Результаты всех этапов:  
   

Рис. 3. Графики кластеризации.

1. Компактность.

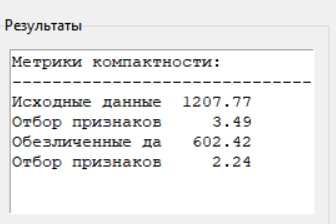


Рис. 4. Вычисленная компактность.

# Анализ работы алгоритма

Таблица 2. Результаты исходные данные.

|  |  |  |  |
| --- | --- | --- | --- |
| Число кластеров | Число признаков | Компактность до отбора | Компактность после отбора |
| 5 | 5 | 1698 | 14,43 |
| 7 | 10 | 1207,77 | 42,62 |
| 10 | 15 | 840,11 | 49,04 |

Таблица 3. Результаты обезличенные данные.

|  |  |  |  |
| --- | --- | --- | --- |
| Число кластеров | Число признаков | Компактность до отбора | Компактность после отбора |
| 5 | 5 | 846,11 | 22,41 |
| 7 | 10 | 602,42 | 97,88 |
| 10 | 15 | 419,58 | 153,48 |

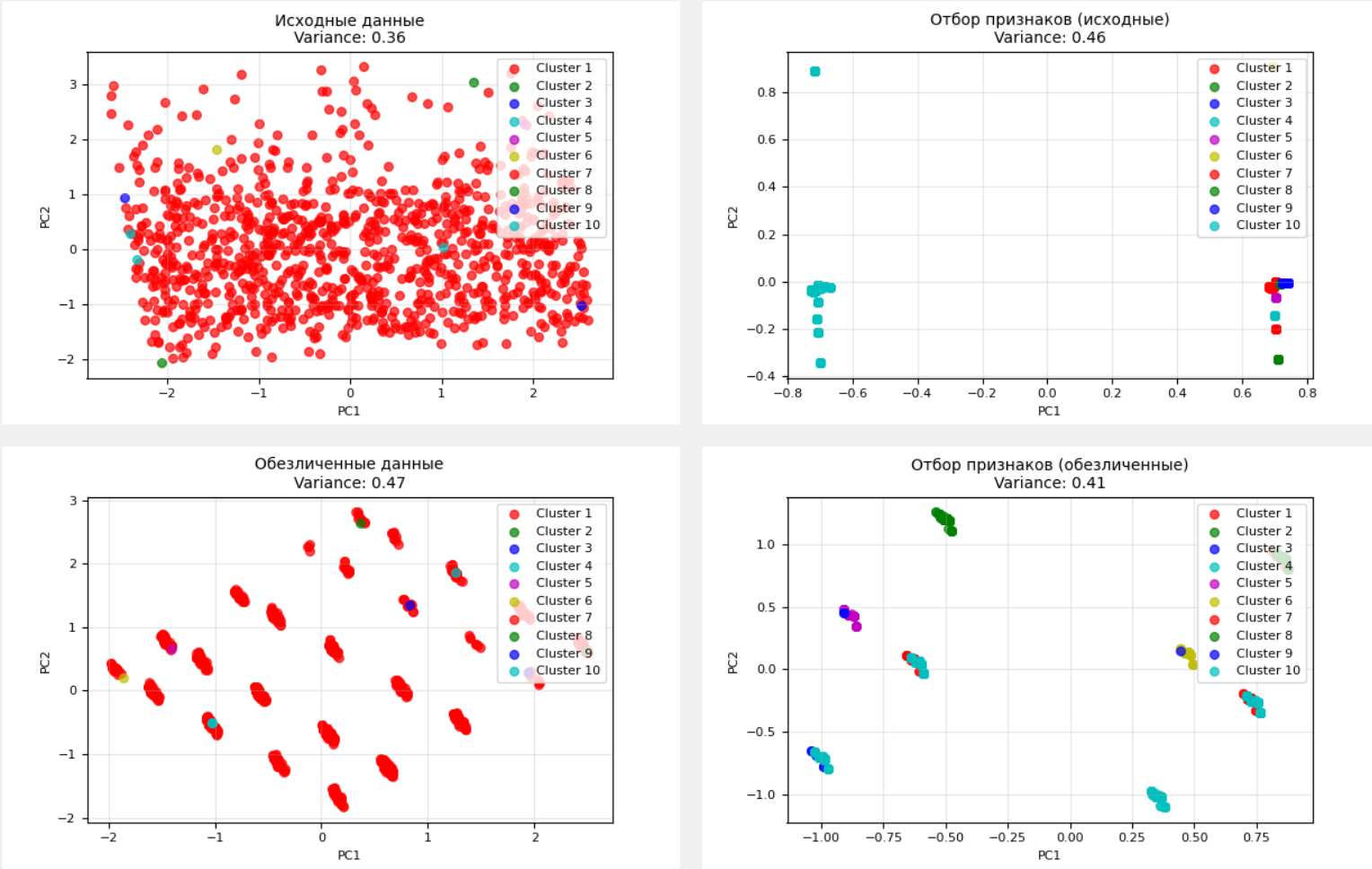


Рис. 5. Графики 10, 15.

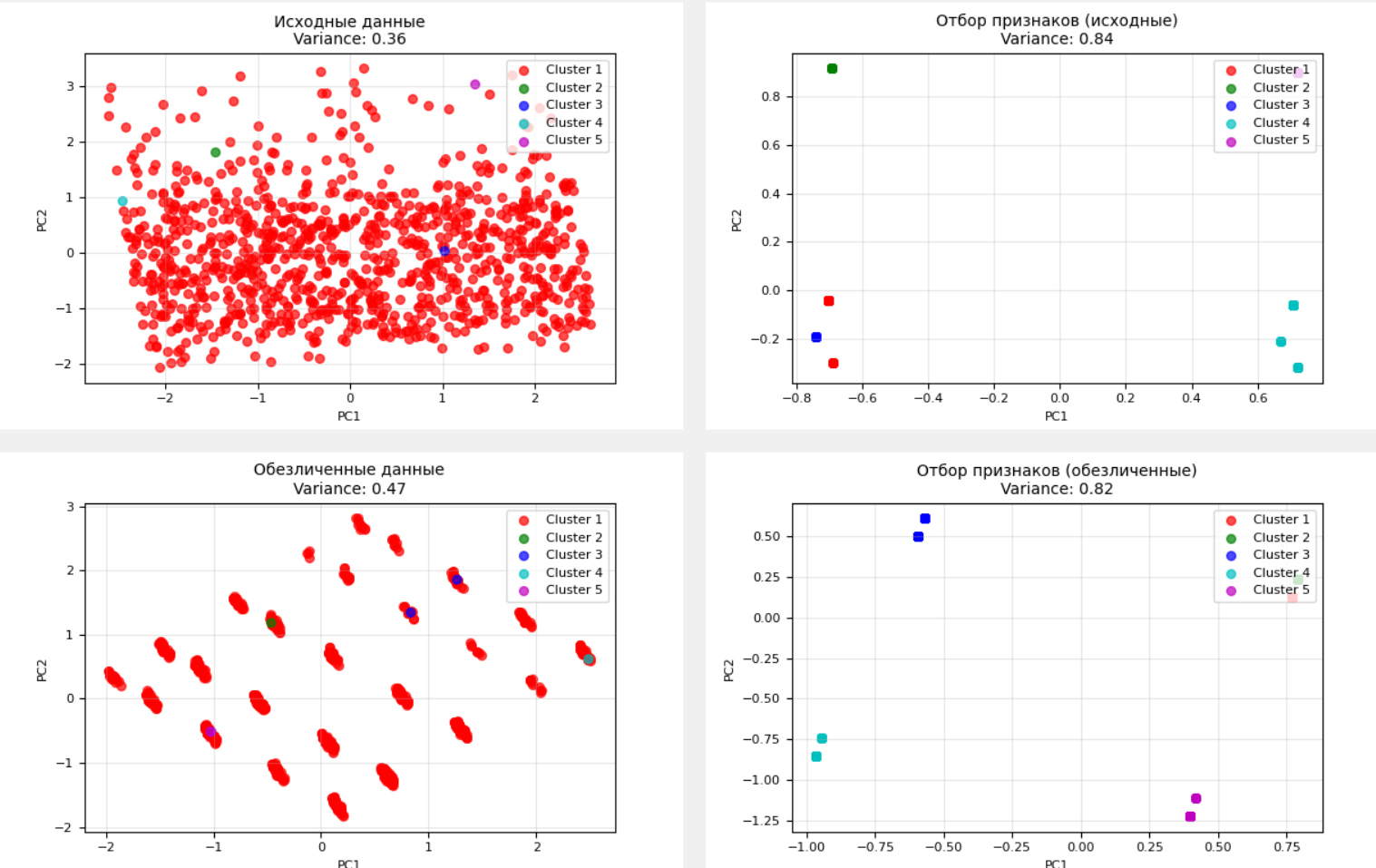


Рис. 6. Графики для 5, 5.

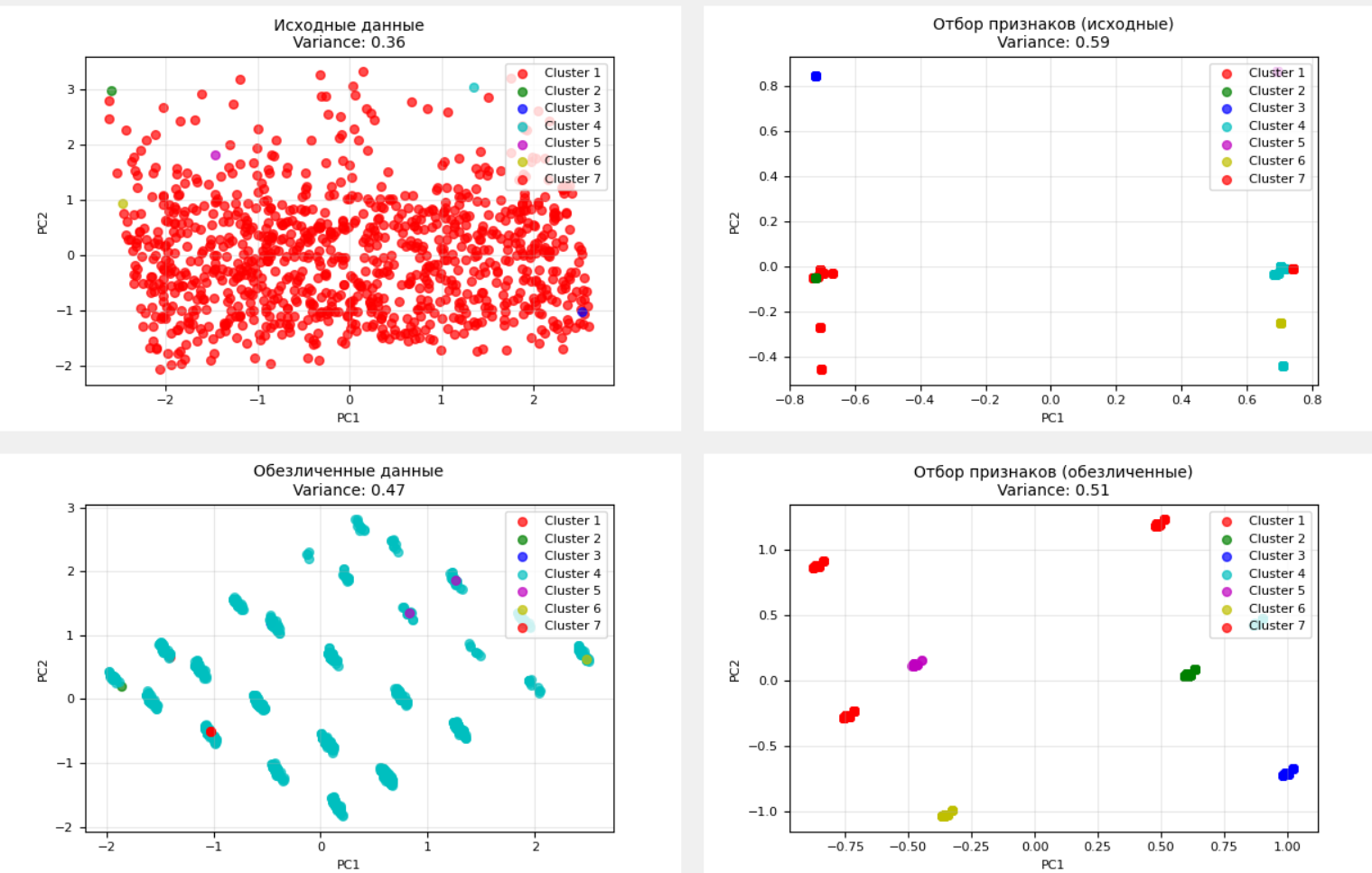


Рис. 7. Графики для 7, 10.

После получения результатов можно сделать следующие выводы:  
*Анализ исходных данных (Таблица 2):*

1. При 5 кластерах и 5 признаках: Компактность снизилась с 1698 до 14.43 (99.15%).

Это значит, что кластеры стали почти идеально компактными.

1. При 7 кластерах и 10 признаках: Компактность снизилась с 1207.77 до 42.62 (96.47%).

Высокое улучшение, но менее значительное, чем при меньших параметрах.

1. При 10 кластерах и 15 признаках: Компактность снизилась с 840.11 до 49.04 (94.16%).

Снижение эффективности при увеличении числа параметров, но всё ещё значительное улучшение.

# *Анализ обезличенных данных (Таблица 3):*

# При 5 кластерах и 5 признаках: Компактность снизилась с 846.11 до 22.41 (97.35%).

# Отбор признаков компенсирует потери от обезличивания, результат близок к исходным данным.

# При 7 кластерах и 10 признаках: Компактность снизилась с 602.42 до 97.88 (83.75%).

# Снижение эффективности на 13.6% по сравнению с исходными данными, что указывает на влияние обезличивания.

# При 10 кластерах и 15 признаках: Компактность снизилась 419.58 до 153.48 (63.43%).

# Минимальное улучшение, обезличивание и высокое число параметров снижают эффективность.

# *Сравнение исходных и обезличенных данных:*

# Для обезличенных данных начальная компактность ниже, чем для исходных (например, 846.11 против 1698 при 5 кластерах). Это связано с потерей информации при обезличивании.

# После отбора признаков результаты для обезличенных данных хуже, чем для исходных (например, 22.41 против 14.43). Это происходит, потому что обезличивание (категоризация, удаление полей) снижает информативность данных.

# Большее число кластеров и признаков увеличивает сложность задачи, снижая эффективность отбора.

# Вывод

# В ходе выполнения проекта были исследованы методы кластеризации данных с использованием алгоритма CURE, а также влияние предобработки (обезличивание, отбор признаков) на качество группировки. Выявлено, что комбинация алгоритма CURE с отбором признаков позволяет эффективно анализировать данные даже после обезличивания, сохраняя высокое качество кластеризации.

# Источники

1. tkinter - <https://metanit.com/python/tkinter/> (дата обращения 20.05.25)
2. pandas - https://pandas.pydata.org/ (дата обращения 20.05.25)