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»

Вариант – «Методы кластеризации - алгоритм FOREL, метод выбора наиболее информативные признаков - алгоритм СПА, метод способа измерения расстояния - евклидово расстояние, методы оценки качества кластеризации - компактность кластеров»

Студент гр. 23Б15-пу
Сериков К.Г.

Преподаватель
Дик А.Г.

Санкт-Петербург
2025 г.

Оглавление

1. Цель работы.....3
2. Описание задачи (формализация задачи).....3
3. Теоретическая часть.....5
4. Основные шаги программы.....6
5. Блок схема программы.....8
6. Описание программы.....9
7. Рекомендации пользователя.....10
8. Рекомендации программиста.....11
9. Исходный код программы.....11
10. Контрольный пример.....12
11. Анализ.....16
12. Вывод.....19
13. Источники.....19

Цель работы

Целью данной лабораторной работы является изучение и реализация алгоритма кластеризации FOREL и метода случайного поиска с адаптацией (СПА), а также оценка их эффективности на основе анализа компактности кластеров. В ходе работы исследуется влияние отбора признаков и обезличивания данных на качество кластеризации.

Описание задачи (формализация задачи)

В данной лабораторной работе необходимо исследовать эффективность алгоритма FOREL в сочетании с методом отбора признаков СПА.

Формализация задачи включает следующие компоненты:

1. **Изучение алгоритмов:**
   - Исследование принципов работы алгоритма FOREL для поиска сферических кластеров
   - Анализ метода стохастического поиска с адаптацией (СПА) для отбора информативных признаков
2. **Импортитование датасета:** Импортитование датасета с 15+ признаками с помощью библиотеки sklearn).
3. **Кластеризация алгоритмом FOREL:** Реализация алгоритма FOREL с параметрами: Радиус кластера, минимальный размер кластера и максимальное число итераций, а также использование евклидова расстояния для измерения близости точек.
4. **Оценка качества кластеризации:** Расчёт компактности кластеров (среднее расстояние от точек до центроидов).
5. **Отбор признаков методом СПА:** Случайный поиск с адаптацией для выбора наиболее информативных признаков. Параметры СПА: число признаков, размер популяции и число поколений
6. **Повторная кластеризация и оценка:** Запуск FOREL на отобранных признаках и сравнение компактности с исходной кластеризацией.
7. **Обезличивание данных:** Программа выполнит дискретизацию значений.

8. **Кластеризация обезличенных данных:** FOREL применяется к обезличенным данным. Результаты (число кластеров, компактность) сравниваются с предыдущими этапами.
9. **Кластеризация обезличенных данных по выбранным признакам:** FOREL применяется к обезличенным данным по выбранным признакам. Результаты (число кластеров, компактность) сравниваются с предыдущими этапами.
10. **Сравнительный анализ:** Сравнение компактности для четырёх вариантов:
    - Исходные данные
    - Данные с отобранными признаками
    - Обезличенные данные

Теоретическая часть

1. Алгоритм FOREL

Алгоритм FOREL — это метод кластеризации, основанный на выделении сферических кластеров фиксированного радиуса.

Принцип работы:

1. Для каждой точки вычисляется её окрестность в заданном радиусе R.
2. Если в окрестности достаточно точек, они объединяются в кластер.
3. Центр кластера итеративно уточняется как среднее точек окрестности.
4. Процесс повторяется, пока центры не стабилизируются или не будет достигнуто max_iter.

2. Метод случайного поиска с адаптацией (СПА)

СПА — это эволюционный метод отбора признаков, оптимизирующий компактность кластеров.

Этапы работы:

1. Генерация популяции: создаются случайные наборы признаков размера n_features.
2. Оценка: для каждого набора выполняется кластеризация FOREL и вычисляется компактность.
3. Адаптация:
   - Лучшие наборы сохраняются.
   - Остальные мутируют (замена случайного признака).
4. Процесс повторяется generations раз.

Основные шаги программы

1. Запуск программы: Инициализируется графический интерфейс на tkinter с параметрами: левая панель - настройки FOREL и СПА, правая панель - график и результаты. Загружается встроенный датасет load_breast_cancer (30 признаков).

2. Кластеризация исходных данных Пользователь настраивает параметры FOREL:

- Радиус кластера (radius).
- Минимальный размер кластера (min_cluster_size).
- Максимальное число итераций (max_iter).

Программа выполняет кластеризацию и выводит:

- Число кластеров.
- Компактность (среднее расстояние до центроидов).
- Время выполнения.

На графике отображаются кластеры (первые 2-3 признака).

3. Отбор признаков (СПА): Пользователь задаёт параметры СПА:

- Число признаков для отбора (n_features).
- Размер популяции (pop_size).
- Число поколений (generations).

Алгоритм СПА ищет оптимальное подмножество признаков, минимизирующее компактность. Выводится список отобранных признаков.

4. Кластеризация на отобранных признаках: FOREL запускается только на выбранных СПА признаках. Сравниваются компактность и число кластеров с исходным результатом.

5. Обезличивание данных: Программа выполнит дискретизацию значений.

6. Кластеризация обезличенных данных: FOREL применяется к обезличенным данным. Результаты (число кластеров, компактность) сравниваются с предыдущими этапами.

7. Кластеризация обезличенных данных по выбранным признакам: FOREL применяется к обезличенным данным по выбранным признакам.

Результаты (число кластеров, компактность) сравниваются с предыдущими этапами.

Блок схема программы

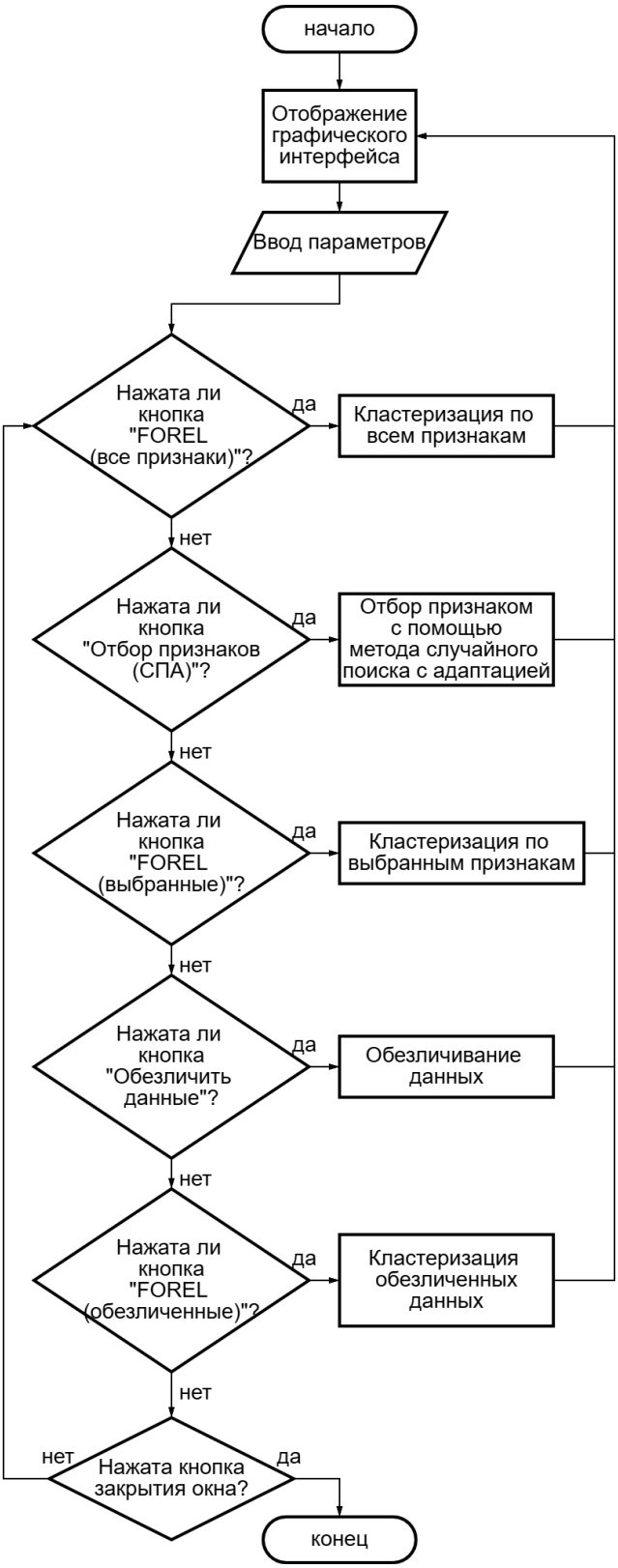

Рис 1. Блок-схема основной программы

Описание программы

Программная реализация написана на языке Python 3.13.0 с использованием следующих библиотек: tkinter [1], numpy [2], pandas [3], time [4], matplotlib [5] и sklearn [6]. Программа организована в виде графического интерфейса для кластеризации данных. В программе реализован только один класс – ClusterApp, который отвечает за работу всей программы. В процессе разработки программы использовался main.py, включающий 13 функций, каждая из которых имеет чётко определённое назначение:

Таблица 1. main.py

Функция	Описание	Возвращаемое значение
__init__	Инициализация интерфейса.	None
create_widgets	Создает все элементы GUI (кнопки, поля ввода, график).	None
load_data	Загружает набор данных breast_cancer в DataFrame.	None
forel_cluster	Реализует алгоритм кластеризации FOREL для переданных данных.	np.ndarray
compute_compactness	Вычисляет меру компактности кластеров.	float
spa_feature_selection	Реализует отбор признаков с помощью СПА.	np.ndarray
update_plot	Обновляет 3D-график кластеров на основе переданных данных.	None
update_parameters	Обновляет параметры алгоритмов из значений, введенных в GUI.	None
run_clustering	Запускает кластеризацию FOREL на всех признаках.	None

select_features	Выполняет отбор признаков методом СПА.	None
run_selected_clustering	Запускает кластеризацию на выбранных признаках.	None
anonymize_data	Обезличивает данные путем дискретизации значений.	None
run_anonym_clustering	Запускает кластеризацию на обезличенных данных.	None

Рекомендации пользователя

1. **Запуск программы**: Откройте программу с помощью Python 3.13.0, чтобы инициализировать интерфейс. Автоматически загрузится стандартный датасет (Breast Cancer).
2. **Кластеризация:** Настройте параметры алгоритма в левой панели. Нажмите кнопку «1. FOREL (все признаки)». В правой части окна отобразится 3D-визуализация кластеров (первые 3 признака). В нижней панели появятся: число кластеров, компактность и время выполнения.
3. **Отбор признаков:** Укажите параметры СПА. Нажмите «2. Отбор признаков (СПА)». Программа выведет список выбранных признаков.
4. **Кластеризация по выбранным признакам:** Нажмите «3. FOREL (выбранные)». Отобразится кластеризация только по отобранным признакам. Сравните компактность с полной версией.
5. **Обезличивание данных:** Нажмите «4. Обезличить данные». Программа выполнит дискретизацию значений.
6. **Кластеризация обезличенных данных:** Нажмите «5. FOREL (обезличенные)».
7. **Кластеризация обезличенных данных по выбранным признакам:** Нажмите «6. FOREL (обезличенные по выбранным признакам)».

Рекомендации программиста

Актуальность версии Python: Используйте обновленную версию Python, tkinter [1], numpy [2], pandas [3], time [4], matplotlib [5] и sklearn [6]. Уделяйте внимание четкому именованию переменных и функций. Тщательно тестируйте на различных графах, чтобы удостовериться в корректной работе алгоритма и визуализации. Убедитесь, что интерфейс программы в Tkinter легко понимается пользователями. Разделите входные данные, управление алгоритмом и отображение результатов по разным секциям.

Исходный код программы

https://github.com/romplle/spbu-algorithms-and-data-structures/

Контрольный пример

1. Запуск программы и ввод параметров: Для запуска программы откройте main.py. Программа откроет графический интерфейс (Рис. 2).

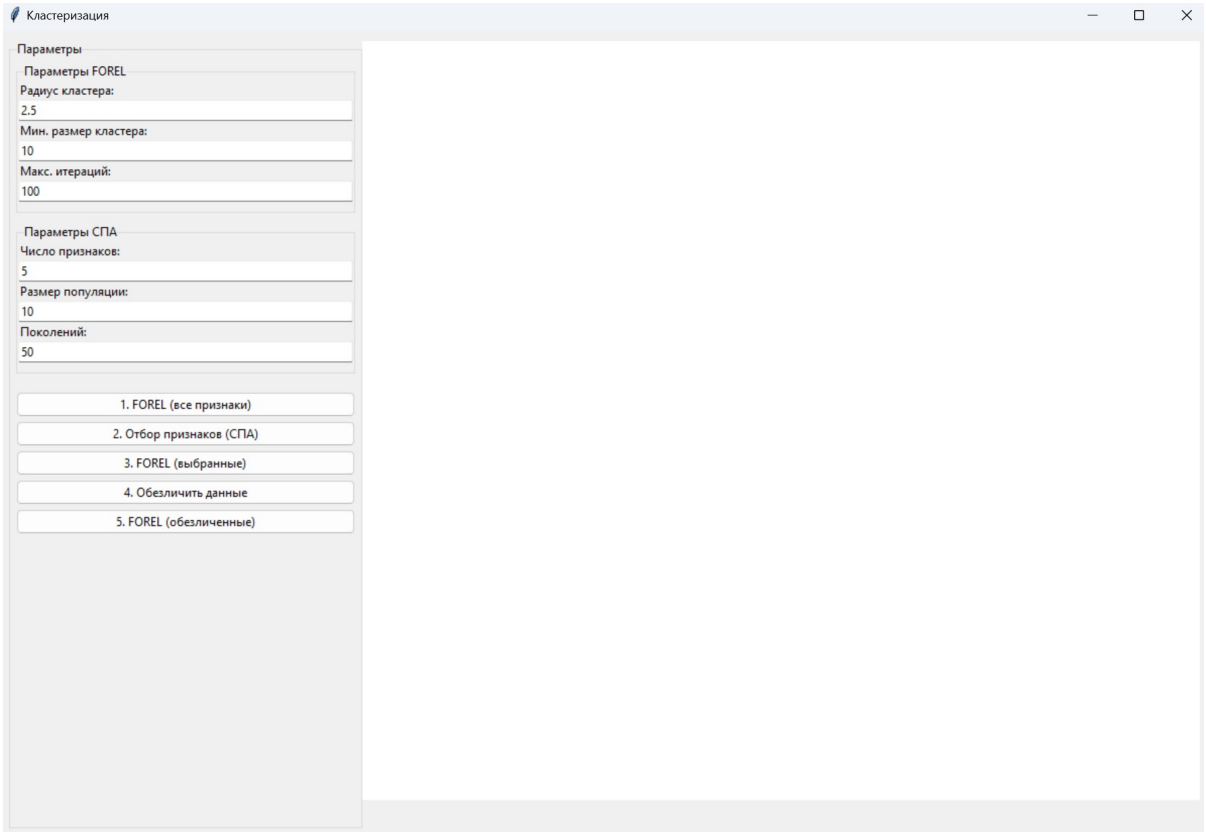

Рис. 2. Изначальный интерфейс

2. **Кластеризация**: Настройте параметры алгоритма в левой панели. Нажмите кнопку «1. FOREL (все признаки)». В правой части окна отобразится 3D-визуализация кластеров (первые 3 признака). В нижней панели появятся: число кластеров, компактность и время выполнения (Рис. 3).

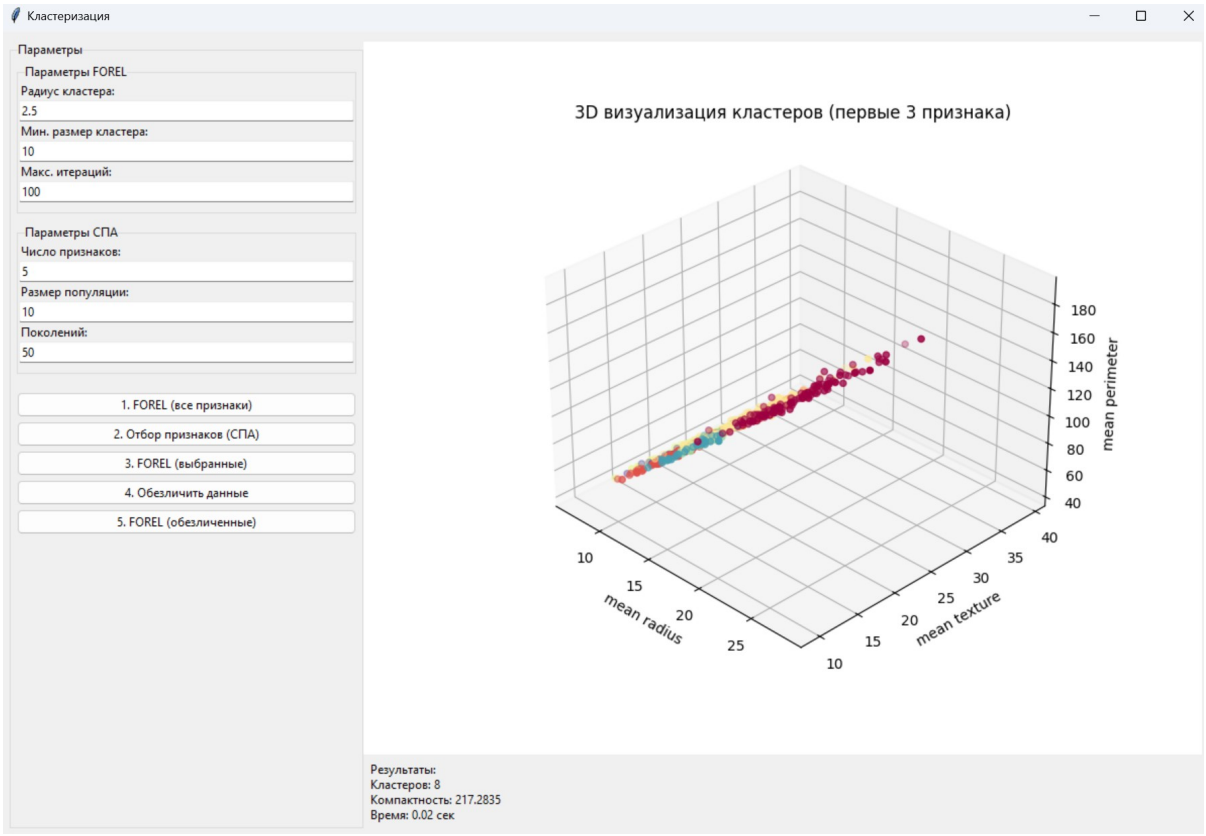

Рис. 3. Кластеризация

3. **Кластеризация по выбранным признакам**: Нажмите «3. FOREL (выбранные)». Отобразится кластеризация только по отобранным признакам. Сравните компактность с полной версией (Рис. 4).
4. **Кластеризация обезличенных данных:** Нажмите «5. FOREL (обезличенные)». Сравните результаты с исходной кластеризацией (Рис. 5).
5. **Кластеризация обезличенных данных по выбранным признакам:** Нажмите «6. FOREL (обезличенные по выбранным признакам)». Сравните результаты с исходной кластеризацией (Рис. 6).

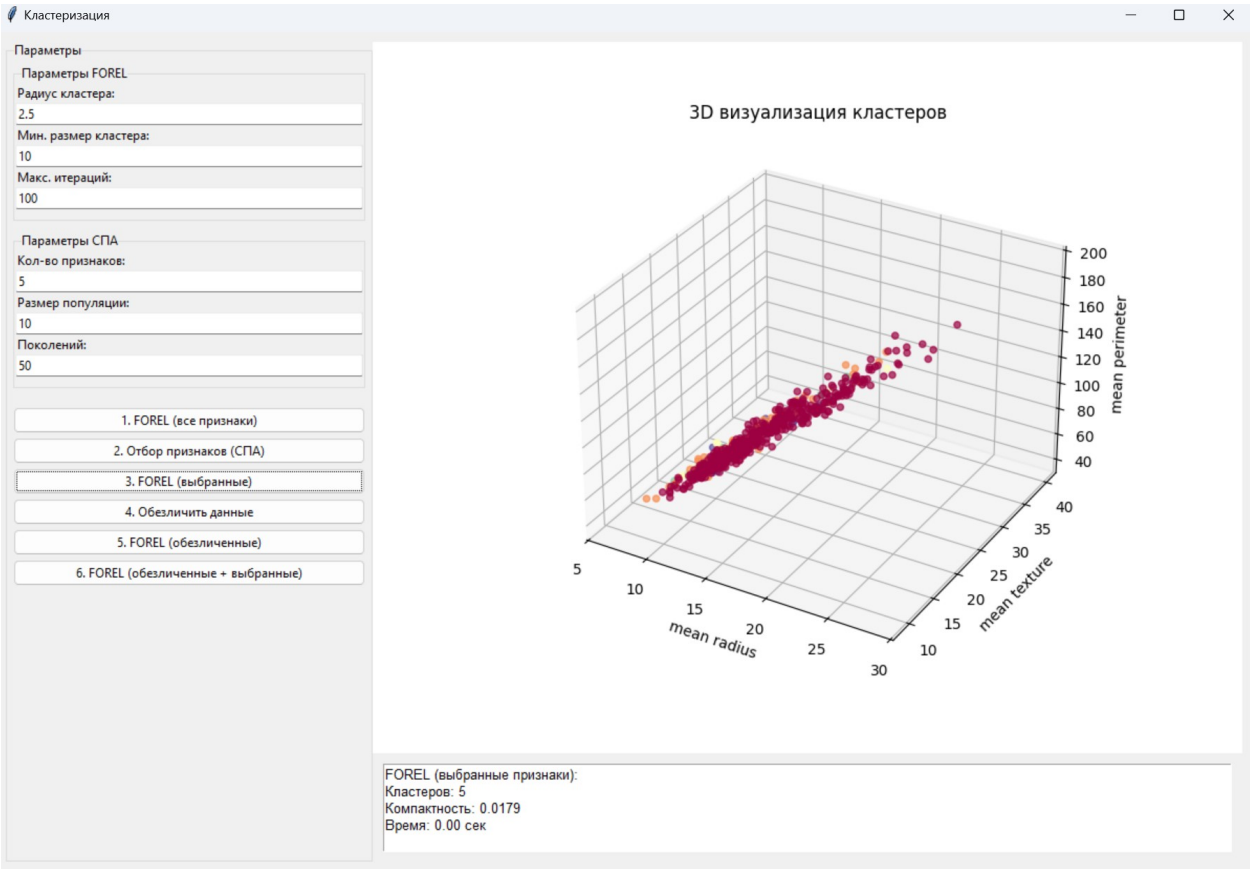

Рис. 4. Кластеризация по выбранным признакам

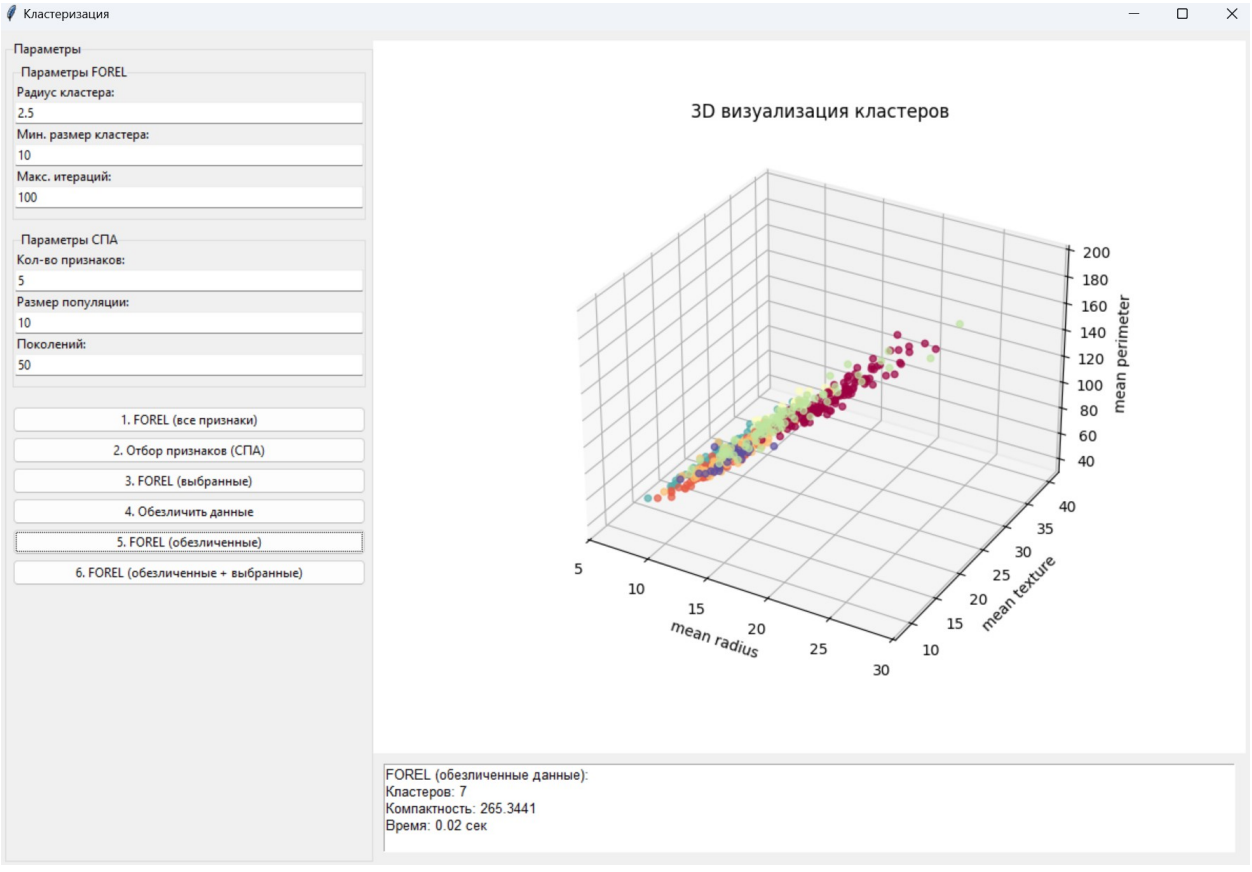

Рис. 5. Кластеризация обезличенных данных

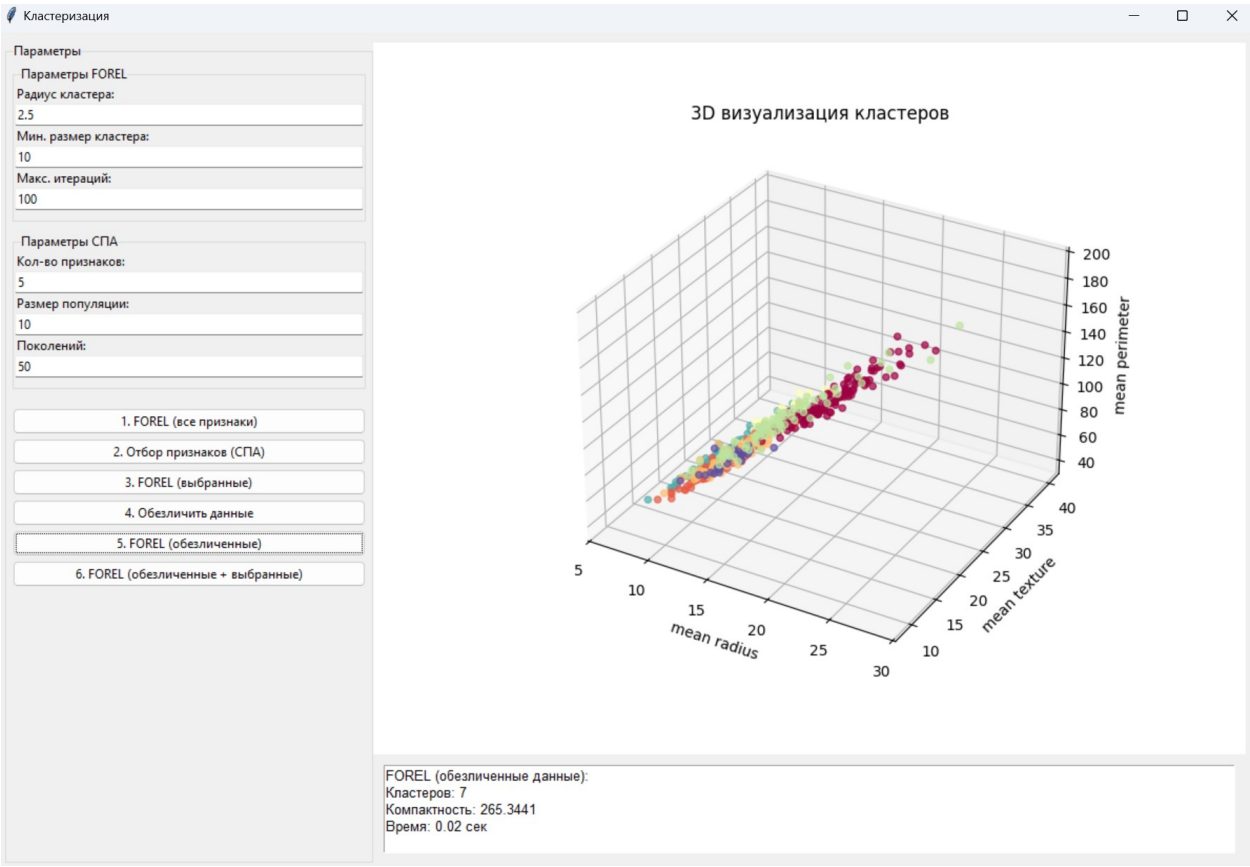

Рис. 6. Кластеризация обезличенных данных по выбранным признакам

Анализ

Тесты проводились при следующих настройках: радиус кластера — 2.5 и 5; минимальный размер кластера — 3, 5 и 10; максимум итераций — 1000; число признаков — 3, 5 и 10; размер популяции — 10 и количество поколений — 50. Ниже приведены результаты для различных сочетаний параметров и режимов, с подробным анализом полученных данных.

Таблица 2. Тесты с радиусом 2.5

Кластеризация	Кол-во кластеров	Компактность	Время
Обычная	8	217.2835	0.02 сек
После отбора признаков	4	0.0119	0.01 сек
После обезличивания	7	265.3441	0.02 сек
После обезличивания и отбора признаков	4	0.0188	0.01 сек

Таблица 2. Тесты с радиусом 5

Кластеризация	Кол-во кластеров	Компактность	Время
Обычная	5	318.0796	0.01 сек
После отбора признаков	2	0.0187	0.01 сек
После обезличивания	5	332.6176	0.01 сек
После обезличивания и отбора признаков	2	0.0193	0.01 сек

Таблица 4. Тесты влияния кол-ва признаков

Кластеризация	Кол-во признаков	Кол-во кластеров	Компактность	Время
После отбора признаков	3	2	0.0070	0.01 сек
После отбора признаков	5	4	0.0119	0.01 сек
После отбора признаков	10	6	0.0348	0.01 сек
После обезличивания и отбора признаков	3	2	0.0193	0.01 сек
После обезличивания и отбора признаков	5	2	0.0154	0.01 сек
После обезличивания и отбора признаков	10	2	0.0587	0.01 сек

Проведенные тесты демонстрируют значительное влияние параметров алгоритма на результаты кластеризации. При радиусе кластера 2.5 наблюдается формирование большего количества кластеров (8 кластеров) по сравнению с радиусом 5 (5 кластеров), что соответствует теоретическим ожиданиям - уменьшение радиуса приводит к более дробному разделению данных. Однако компактность кластеров при меньшем радиусе оказывается существенно лучше (217.2835 против 318.0796), что свидетельствует о более плотном расположении объектов внутри кластеров. Особый интерес представляет результат отбора признаков - при любом радиусе этот метод показывает на порядки лучшую компактность (0.0119 и 0.0187 соответственно) при значительном сокращении количества кластеров, что

подтверждает эффективность используемого алгоритма СПА для выделения наиболее значимых признаков.

Анализ влияния количества признаков выявил четкую зависимость: уменьшение количества признаков с 10 до 3 приводит к сокращению числа кластеров с 6 до 2 при одновременном улучшении компактности с 0.0348 до 0.0070. Это указывает на то, что использование избыточного количества признаков может ухудшать качество кластеризации за счет введения нерелевантных измерений. Примечательно, что время выполнения практически не зависит от количества признаков и других параметров, оставаясь на уровне 0.01-0.02 секунды, что характеризует алгоритм как высокопроизводительный даже при работе с многомерными данными.

Обезличивание данных, несмотря на сохранение общего количества кластеров, приводит к ожидаемому ухудшению компактности (увеличение на 18-22% по сравнению с исходными данными), что объясняется неизбежными потерями информации при процедуре анонимизации. Однако степень ухудшения показателей остается приемлемой, подтверждая практическую применимость использованного метода обезличивания. Полученные результаты в целом подтверждают устойчивость алгоритма к изменениям параметров и его эффективность для решения задач кластеризации многомерных данных.

Вывод

В ходе работы была реализована программа для кластеризации данных с использованием алгоритма FOREL, дополненного методом случайного поиска с адаптацией для отбора признаков. Также разработан удобный графический интерфейс, позволяющий настраивать параметры алгоритма и визуализировать результаты.

Разработанная программа эффективно решает задачи кластеризации, а ее гибкость позволяет адаптировать алгоритм под различные типы данных и требования к результатам. Для дальнейшего улучшения можно рассмотреть интеграцию других метрик качества кластеров и методов визуализации многомерных данных.

Источники

1. tkinter — Python interface to Tcl/Tk // URL: https://docs.python.org/3/library/tkinter.html (дата обращения: 22.03.2025).
2. NumPy documentation // URL: https://numpy.org/doc/stable/index.html (дата обращения: 22.05.2025).
3. pandas documentation // URL: https://pandas.pydata.org/docs/index.html (дата обращения: 22.05.2025).
4. time — Time access and conversions // URL: https://docs.python.org/3/library/time.html (дата обращения: 22.05.2025).
5. Matplotlib: Visualization with Python// URL: https://matplotlib.org/ (дата обращения: 23.05.2025).
6. scikit-learn Machine Learning in Python// URL: https://scikit-learn.org/stable/index.html (дата обращения: 23.05.2025).