**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

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

**ОТЧЕТ**

**по лабораторной работе №10**

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

**на тему «Кластеризация данных»**

**Студент гр. 23Б15-пу**

**Черевко М.Е.**

**Преподаватель**

**Дик А.Г.**

**Санкт-Петербург**

**2025 г.**

Оглавление

1. [Цель работы 3](#_Toc198475769)
2. [Описание задачи (формализация задачи) 3](#_Toc198475770)
3. [Теоретическая часть 5](#_Toc198475771)
4. [Основные шаги программы 7](#_Toc198475772)
5. [Блок схема программы 10](#_Toc198475773)
6. [Описание программы 11](#_Toc198475774)
7. [Рекомендации пользователя 13](#_Toc198475775)
8. [Рекомендации программиста 15](#_Toc198475776)
9. [Исходный код программы 17](#_Toc198475777)
10. [Используемый датасет 17](#_Toc198475778)
11. [Контрольный пример 18](#_Toc198475779)
12. [Анализ результатов работы алгоритма 21](#_Toc198475780)
13. [Вывод 35](#_Toc198475781)
14. [Источники 37](#_Toc198475782)

# Цель работы

Целью данного проекта является исследование и сравнение качества кластеризации табличных данных до и после отбора признаков и обезличивания с использованием иерархической кластеризации и алгоритма максиминного расстояния. Будет оценено влияние отбора признаков (по критерию компактности кластеров) и обезличивания на эффективность кластеризации с применением метрики Чебышева для расстояний и силуэтного коэффициента для оценки качества

# Описание задачи (формализация задачи)

1. Подготовка данных

* Сформировать синтетический или реальный датасет, содержащий не менее 15 признаков, включая категориальные (например, Category, Color) и числовые столбцы (например, Age, Purchase Amount (USD)). В качестве примера используется реальный датасет shopping\_trends.csv.
* Убедиться в отсутствии меток классов (данные относятся к задаче без учителя), но обеспечить возможность оценки качества кластеризации с помощью силуэтного коэффициента.

1. Первичная кластеризация

* Выполнить кластеризацию с использованием иерархического алгоритма максиминного расстояния (метод средней связи) на полном наборе признаков.
  + В качестве метрики расстояния использовать метрику Чебышева (Chebyshev), определяемую как максимальная разница между координатами двух точек:
  + Оценить качество кластеризации с помощью силуэтного коэффициента, который измеряет компактность и отделимость кластеров в пространстве с метрикой Чебышева.

1. Отбор признаков

* На основе критерия компактности кластеров (среднее расстояние от точек кластера до его центроида по метрике Чебышева) выбрать наиболее информативные признаки. Количество признаков определяется либо пользователем (через интерфейс), либо автоматически (все признаки, если не указано иное).
* Повторить кластеризацию на отобранных признаках с теми же параметрами (метрика Чебышева, метод средней связи).

1. Сравнение результатов

* Оценить качество кластеризации после отбора признаков с использованием силуэтного коэффициента.
* Сравнить результаты с первоначальной кластеризацией на полном наборе признаков и сделать выводы об эффективности отбора признаков, основываясь на изменении силуэтного коэффициента и числа кластеров.

1. Обезличивание данных

* Обезличить датасет с использованием методов биннинга числовых данных (например, Age и Purchase Amount (USD) разбиваются на интервалы), обобщения категориальных данных (например, редкие категории заменяются на Other) и k-анонимности (гарантия, что каждая комбинация квази-идентификаторов встречается не менее ( k ) раз, где ( k = 10 )).
* Выполнить кластеризацию на обезличенных данных и вновь рассчитать силуэтный коэффициент.

1. Финальный анализ

* Сравнить три результата кластеризации:
  + На полном наборе признаков.
  + На отобранных информативных признаках.
  + На обезличенных данных.
* Проанализировать влияние объёма признаков, качества отбора признаков и обезличивания на эффективность кластеризации, учитывая:
  + Изменение силуэтного коэффициента как индикатора качества разделения.
  + Число эффективных кластеров (кластеров с более чем одной точкой).
  + Визуальную интерпретацию с помощью метода главных компонент (PCA), отображающего данные в двумерном пространстве с долей объяснённой дисперсии для главных компонент (PC1 и PC2).
* Сделать выводы о том, как обезличивание и отбор признаков влияют на структуру данных и способность алгоритма выявлять осмысленные кластеры.

# Теоретическая часть

1. Обезличивание данных

Обезличивание данных — это процесс модификации данных с целью защиты конфиденциальности, при котором сохраняется возможность их анализа. Основная цель обезличивания — минимизировать риск идентификации отдельных лиц, сохраняя при этом полезность данных для задач, таких как кластеризация. В данном проекте обезличивание включает несколько методов:

* **Биннинг числовых данных**: Числовые признаки (например, возраст, сумма покупок) разделяются на интервалы (бины), а значения внутри каждого интервала заменяются средним значением. Это уменьшает точность данных, снижая вероятность идентификации, но может влиять на вариативность, необходимую для кластеризации.
* **Обобщение категориальных данных**: Категориальные признаки (например, категория товара, цвет) группируются в более общие классы. Редкие категории заменяются на общую категорию (например, Other), что также снижает риск идентификации.
* **Удаление прямых идентификаторов**: Признаки, которые могут напрямую идентифицировать человека (например, Customer ID), удаляются из набора данных.

Обезличивание неизбежно приводит к потере информации, что может повлиять на качество последующего анализа, включая кластеризацию. В данном проекте это проявляется в снижении числа кластеров после обезличивания по сравнению с исходными данными.

1. K-анонимность

K-анонимность — это модель защиты конфиденциальности, которая гарантирует, что каждая запись в наборе данных неотличима от как минимум ( k-1 ) других записей по набору квази-идентификаторов (признаков, которые в совокупности могут идентифицировать человека, например, возраст, пол, категория покупок).

* **Определение**: Набор данных удовлетворяет k-анонимности, если каждая комбинация значений квази-идентификаторов встречается как минимум ( k ) раз.
* **Реализация**: В данном проекте для достижения k-анонимности:
  1. Определяются квази-идентификаторы (например, Age, Gender, Category, Purchase Amount (USD)).
  2. Данные группируются по этим признакам, и подсчитывается количество записей в каждой группе.
  3. Группы с числом записей меньше ( k ) (в данном случае ( k = 10 )) заменяются на наиболее частую комбинацию значений.
* **Влияние на данные**: K-анонимность дополнительно уменьшает вариативность данных, так как редкие комбинации заменяются более частыми, что может привести к потере различий между записями и затруднить кластеризацию.

1. Иерархическая кластеризация с метрикой Чебышева

Кластеризация — это метод машинного обучения без учителя, который используется для группировки данных на основе их сходства. В данном проекте применяется иерархическая кластеризация с метрикой Чебышева.

* **Иерархическая кластеризация**: Этот метод строит иерархию кластеров, начиная с отдельных точек (агломеративный подход). На каждом шаге объединяются два наиболее близких кластера, пока не будет достигнуто желаемое число кластеров.
  + В проекте используется метод средней связи (method='average'), где расстояние между кластерами определяется как среднее расстояние между всеми парами точек из этих кластеров.
* **Метрика Чебышева**: Метрика Чебышева (или - метрика) определяется как максимальная разница между координатами двух точек:  
  Эта метрика подчёркивает наибольшее различие между точками в каком-либо измерении, что делает её чувствительной к выбросам и различиям в отдельных признаках. В данном проекте метрика Чебышева используется в соответствии с условием задачи для вычисления расстояний при построении дендрограммы и оценки отделимости кластеров.

1. Выбор числа кластеров с помощью силуэтного коэффициента

Для автоматического определения оптимального числа кластеров используется силуэтный коэффициент — метрика, оценивающая, насколько хорошо точки соответствуют своему кластеру по сравнению с другими кластерами.

* **Формула силуэтного коэффициента**:  
  Для каждой точки ( i ):  
  где:
  + ( a(i) ) — среднее расстояние от точки ( i ) до всех других точек в её кластере (компактность).
  + ( b(i) ) — минимальное среднее расстояние от точки ( i ) до точек в любом другом кластере (отделимость).  
    Средний силуэтный коэффициент для всего набора данных — это среднее ( s(i) ) по всем точкам.
* **Интерпретация**:
  + Значения близкие к 1 указывают на хорошее разделение кластеров.
  + Значения около 0 или отрицательные указывают на плохую кластеризацию (точки могут быть ближе к другим кластерам, чем к своему).
* **Реализация**: В проекте силуэтный коэффициент вычисляется с использованием метрики Чебышева для разного числа кластеров (от 2 до 10). Выбирается число кластеров, максимизирующее силуэтный коэффициент. Если силуэтный коэффициент низкий (< 0.1), выдаётся рекомендация кластеризовать данные до обезличивания.

1. Выбор признаков по компактности

Для уменьшения размерности данных и повышения качества кластеризации используется метод выбора признаков, основанный на компактности кластеров.

* **Компактность**: Компактность кластера измеряет, насколько близко точки внутри кластера расположены друг к другу. В данном проекте компактность вычисляется как среднее расстояние (по метрике Чебышева) от точек кластера до его центроида.
* **Алгоритм выбора признаков**:
  1. Изначально выбирается пустое подмножество признаков.
  2. На каждой итерации добавляется признак, который минимизирует среднюю компактность кластеров, вычисленную с использованием текущего подмножества признаков и нового признака.
  3. Процесс повторяется, пока не будет достигнуто заданное число признаков.
* **Цель**: Этот метод позволяет отобрать признаки, которые способствуют образованию более плотных и компактных кластеров, улучшая качество кластеризации.

1. Визуализация кластеров с помощью PCA

Для визуализации результатов кластеризации используется метод главных компонент (Principal Component Analysis, PCA).

* **PCA**: Это метод снижения размерности, который проецирует данные в пространство меньшей размерности, сохраняя максимальную дисперсию. В данном проекте данные проецируются на две главные компоненты (PC1 и PC2), чтобы их можно было визуализировать на двумерном графике.
* **Процесс**:
  1. Категориальные признаки кодируются с помощью LabelEncoder (преобразуются в числовые значения).
  2. Данные масштабируются с помощью MinMaxScaler, чтобы привести все признаки к диапазону [0, 1].
  3. Применяется PCA для получения двух главных компонент.
  4. На двумерном графике точки отображаются с цветовой кодировкой по кластерам.
* **Доля объяснённой дисперсии**: PCA предоставляет информацию о доле дисперсии, объяснённой каждой компонентой (PC1 и PC2). Высокая доля объяснённой дисперсии (например, >70%) указывает, что двумерная проекция хорошо отражает структуру данных.
* **Ограничения**: PCA не учитывает кластерную структуру и может не отражать разделимость кластеров, особенно если данные после обезличивания стали слишком однородными.

1. Проблемы и ограничения

* **Влияние обезличивания на кластеризацию**: Обезличивание снижает вариативность данных, что может привести к тривиальному разделению (например, 1 или 2 кластера вместо ожидаемых 4). Это связано с биннингом, обобщением и k-анонимностью, которые "сглаживают" данные.
* **Метрика Чебышева**: Использование метрики Чебышева делает кластеризацию чувствительной к максимальным различиям в признаках. После обезличивания такие различия уменьшаются, что затрудняет разделение на кластеры.
* **Силуэтный коэффициент**: Низкий силуэтный коэффициент после обезличивания (< 0.1) указывает на плохую разделимость данных. В таких случаях рекомендуется кластеризовать данные до обезличивания.
* **PCA**: Хотя PCA сохраняет дисперсию (например, по главной компоненте 1), она не гарантирует, что кластеры будут визуально различимы, особенно если данные однородны.

# Основные шаги программы

Программа DataClusteringApp, реализованная на Python с использованием библиотеки tkinter, предоставляет графический интерфейс для обработки, обезличивания и кластеризации табличных данных. Основные этапы работы программы включают загрузку данных, их обезличивание, отбор признаков, выполнение кластеризации с метрикой Чебышева, оценку качества и визуализацию результатов. Ниже описаны ключевые шаги.

* 1. Загрузка и предварительная обработка данных
* **Загрузка датасета**: Пользователь выбирает файл в формате CSV или ARFF (например, shopping\_trends.csv) через диалоговое окно filedialog.askopenfilename. Для ARFF-файлов данные декодируются из байтового формата с использованием scipy.io.arff, а для CSV — загружаются через pandas.read\_csv.
* **Проверка данных**: Программа проверяет, что датасет содержит не менее 2 строк и хотя бы 1 признак. При несоответствии выводится сообщение об ошибке, и таблица очищается.
* **Обработка пропущенных значений**: Пропуски заполняются:
  + Для числовых столбцов (например, Age, Purchase Amount (USD)) — медианным значением.
  + Для категориальных столбцов (например, Category, Color) — наиболее частым значением (модой).
* **Сохранение исходных данных**: Загруженные данные сохраняются как original\_data, а текущие данные — как current\_data. Данные отображаются в таблице интерфейса с помощью ttk.Treeview с динамической настройкой ширины столбцов.
  1. Обезличивание данных
* **Удаление идентификаторов**: Прямые идентификаторы, такие как Customer ID, удаляются из датасета.
* **Биннинг числовых признаков**: Числовые столбцы разбиваются на интервалы с заменой значений на средние:
  + Age: интервалы [0–10, 10–20, 20–30, 30–40, 40–50, 50–60, 60–70], значения заменяются средними (например, 5, 15, ..., 65).
  + Purchase Amount (USD): интервалы [0–20, 20–40, ..., 120–150], с адаптацией последнего интервала до максимального значения.
  + Review Rating: интервалы [0–1, 1–2, 2–3, 3–4, 4–5], значения заменяются средними (0.5, 1.5, ..., 4.5).
  + Previous Purchases: интервалы [-1–0, 0–5, 5–10, ..., 40–50], с адаптацией последнего интервала до максимального значения.
* **Обобщение категориальных признаков**:
  + Category: сохраняются Clothing, Footwear, остальные заменяются на Other.
  + Color: сохраняются Gray, Maroon, White, остальные заменяются на Other.
  + Size: объединяются в S/M и L/XL.
  + Payment Method: объединяются в Digital (Venmo, PayPal, Credit Card) и Other.
  + Frequency of Purchases: объединяются в Frequent (Weekly, Fortnightly, Bi-Weekly) и Infrequent.
* **K-анонимность**: Для квази-идентификаторов (Age, Gender, Category, Purchase Amount (USD)) группы с числом записей менее 5 заменяются на наиболее частую комбинацию. Результаты (k-анонимность, число изменённых строк, распределение размеров групп) отображаются в текстовом поле "Результаты".

1. Отбор признаков

* **Критерий компактности**: Признаки отбираются на основе минимизации компактности кластеров, вычисляемой как среднее расстояние (по метрике Чебышева) от точек кластера до его центроида.
* **Процесс**: Пользователь задаёт число признаков через поле ввода (0 — использовать все). Алгоритм итеративно добавляет признаки, минимизируя компактность, до достижения заданного числа или полного набора.
* **Результат**: Отобранные признаки используются для кластеризации, а их количество отображается в "Результаты".
  1. Кластеризация
* **Алгоритм**: Применяется иерархическая кластеризация с методом средней связи (method='average') и метрикой Чебышева.
* **Определение числа кластеров**:
  + Пользователь может задать число кластеров (≥2) или выбрать автоматический режим (0).
  + В автоматическом режиме число кластеров определяется максимизацией силуэтного коэффициента с метрикой Чебышева.
* **Предобработка**: Категориальные признаки кодируются с помощью LabelEncoder, данные масштабируются с MinMaxScaler.
* **Результаты**: Метки кластеров сохраняются в столбце Predicted\_Cluster. В "Результаты" выводятся: число признаков, число кластеров (и эффективных), силуэтный коэффициент, время выполнения, распределение точек. При низком силуэтном коэффициенте (< 0.1) выводится рекомендация кластеризовать до обезличивания.
  1. Оценка качества кластеризации
* **Силуэтный коэффициент**: Качество оценивается с помощью силуэтного коэффициента, вычисляемого с метрикой Чебышева, который отражает компактность и отделимость кластеров.
* **Диагностика**: Число эффективных кластеров (с более чем 1 точкой) проверяется для выявления тривиального разделени.
  1. Визуализация результатов
* **Снижение размерности**: Метод PCA применяется к масштабированным данным для получения двух главных компонент (PC1 и PC2).
* **График**: Точки отображаются на scatter-графике с цветовой кодировкой по кластерам. Доля объяснённой дисперсии для PC1 и PC2 указывается в заголовке, что позволяет оценить адекватность двумерной проекции.
  1. Сохранение и сброс данных
* **Сохранение**: Датасет сохраняется в CSV или Excel (XLSX) через filedialog.asksaveasfilename, с использованием openpyxl для Excel.
* **Сброс**: Возвращает current\_data к original\_data, обновляя таблицу и статус.

# Блок схема программы

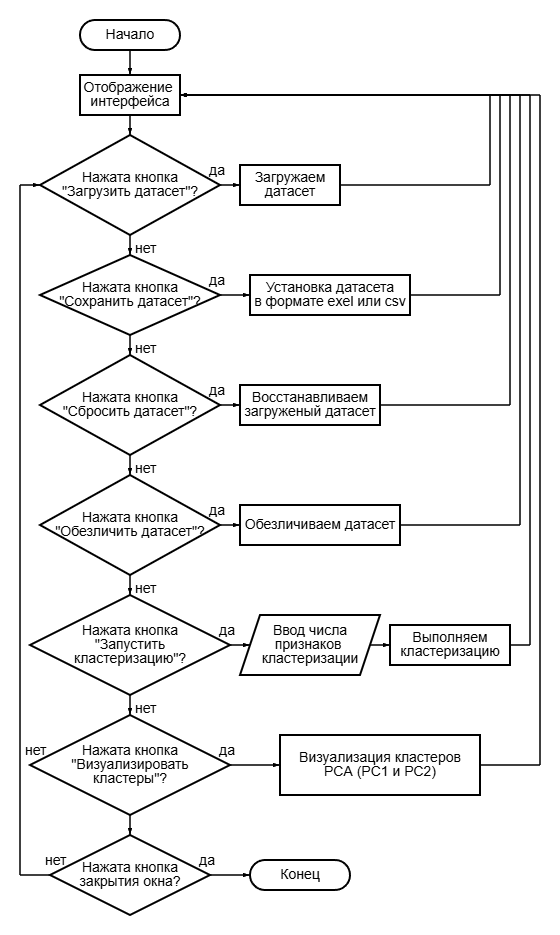


Рис 1. Блок-схема data\_clustering\_app.py

# Описание программы

Программа реализована на Python 3.12.6 с использованием pandas и numpy для обработки данных, tkinter для построения GUI и matplotlib для визуализации кластеров, а также scikit-learn и scipy для вычислений метрик и оптимизации, openpyxl для exel формата. Главный класс DataClusteringApp отвечает за:

Таблица 1. DataClusteringApp

|  |  |  |
| --- | --- | --- |
| **Метод** | **Описание** | **Возвращаемое значение** |
| \_\_init\_\_ | Инициализирует окно приложения, задаёт заголовок и минимальный размер окна, создаёт пустые DataFrame для исходных (original\_data) и текущих данных (current\_data), вызывает setup\_ui(). | None |
| setup\_ui | Создаёт графический интерфейс (GUI): добавляет кнопки для загрузки, сохранения, сброса, обезличивания, визуализации и кластеризации, поля ввода для числа признаков и кластеров, текстовое поле для результатов и таблицу для отображения данных с прокруткой. | None |
| load\_dataset | Загружает датасет из файла CSV или ARFF, проверяет его корректность (≥2 строки, ≥1 признак), заполняет пропущенные значения (медианой для числовых, модой для категориальных), сохраняет как original\_data и отображает в таблице. | None |
| save\_dataset | Сохраняет текущий датасет в файл CSV или Excel (XLSX) с помощью filedialog.asksaveasfilename, используя openpyxl для формата Excel, выводит статус сохранения. | None |
| reset\_dataset | Сбрасывает текущий датасет (current\_data) к исходному состоянию (original\_data), обновляет таблицу и текстовое поле статуса. | None |
| update\_table\_display | Обновляет таблицу ttk.Treeview, отображая текущий датасет с динамической настройкой ширины столбцов на основе максимальной длины значений. | None |
| update\_status | Обновляет текстовое поле status\_text, очищая его и вставляя новое сообщение (например, результаты обработки или ошибки). | None |
| visualize\_clusters | Выполняет визуализацию кластеров с помощью PCA: кодирует категориальные признаки с LabelEncoder, масштабирует данные с MinMaxScaler, снижает размерность до 2D, отображает scatter-график с долей объяснённой дисперсии. | None |
| compute\_compactness | Вычисляет компактность кластеров как среднее расстояние (по метрике Чебышева) от точек кластера до его центроида, используется для отбора признаков. | float |
| select\_features\_by\_compactness | Отбирает заданное число признаков, минимизируя компактность кластеров с использованием иерархической кластеризации (метод средней связи, метрика Чебышева). | list (индексы признаков) |
| compute\_separability | Вычисляет силуэтный коэффициент с метрикой Чебышева для оценки качества кластеризации, возвращает 0.0 при недостатке данных или ошибке. | float |
| find\_optimal\_clusters | Определяет оптимальное число кластеров (от 2 до 10+) путём максимизации силуэтного коэффициента с метрикой Чебышева, возвращает число кластеров, лучший коэффициент и список оценок. | tuple (число кластеров, коэффициент, список оценок) |
| perform\_clustering | Выполняет кластеризацию: кодирует категориальные признаки, масштабирует данные, отбирает признаки, применяет иерархическую кластеризацию (метод средней связи, метрика Чебышева), определяет число кластеров (вручную или автоматически), вычисляет силуэтный коэффициент, выводит результаты и время выполнения в текстовое поле. | None |
| deidentify\_dataset | Обезличивает датасет: выполняет биннинг числовых признаков (например, Age, Purchase Amount (USD)), обобщает категориальные (например, Category, Color), применяет k-анонимность (( k = 5 )), выводит результаты в текстовое поле. | None |

# Рекомендации пользователя

1. Запуск приложения
   * Убедитесь, что в консоли доступен Python >= 3.12.6.
   * Выполните: python data\_clustering\_app.pyb или щелкните по файлу дважды.
2. Загрузка датасета
   * Нажмите «Загрузить датасет».
   * Выберите файл в формате CSV или ARFF.
   * После успешной загрузки таблица отобразится в правой части окна, а в области статусных сообщений появится «Датасет загружен успешно.»
3. Сброс к оригиналу
   * Если вы изменили данные (кластеры, обезличивание и т.п.) и хотите вернуться к исходному виду, нажмите «Сбросить датасет».
   * Таблица вернётся к тому состоянию, в котором была сразу после загрузки.
4. Обезличивание
   * Чтобы скрыть подробные признаки (например, возраст, сумму покупок) и получить k-анонимный набор, нажмите «Обезличить датасет».
   * В тексте статуса вы увидите рассчитанное значение k-анонимности и распределение размеров групп.
5. Указание числа признаков
   * Введите в поле «Число признаков (если пусто — будут задействованы все)» целое число ( k ) (от 1 до ( d - 1 ), где ( d ) — общее число колонок за вычетом метки, если она присутствует) или оставьте 0 для использования всех признаков.
6. Кластеризация
   * Нажмите «Запустить кластеризацию».
   * Программа:
     + Закодирует все категориальные поля в числа с помощью LabelEncoder.
     + Отберёт ( k ) наиболее «компактных» признаков по критерию средней связи с метрикой Чебышева.
     + Проведёт иерархическую кластеризацию с методом средней связи и метрикой Чебышева.
     + Определит число кластеров (вручную, если указано ≥2, или автоматически от 2 до 10+ с максимизацией силуэтного коэффициента).
     + Рассчитает силуэтный коэффициент для оценки качества.
     + Выведет число кластеров, силуэтный коэффициент, время выполнения и распределение точек по кластерам.
7. Интерпретация результатов
   * Силуэтный коэффициент отражает компактность и отделимость кластеров: значения > 0.5 указывают на хорошее разделение, < 0.1 — на плохую разделимость (рекомендуется кластеризация до обезличивания).
   * Число эффективных кластеров (с более чем одной точкой) показывает, сколько кластеров действительно сформировано.
   * Время выполнения помогает оценить производительность на больших датасетах.

# Рекомендации программиста

1. Установка и зависимости
   * Требуется Python >= 3.12.6.
   * Установите библиотеки: pip install pandas numpy scipy scikit-learn matplotlib tkinter openpyxl
2. Архитектура кода
   * DataClusteringApp — основной класс, наследует tk.Tk.
   * GUI построен в методе setup\_ui():
     + кнопки для загрузки, сохранения, сброса, обезличивания, визуализации и запуска кластеризации;
     + поля для ввода числа признаков и кластеров;
     + ttk.Treeview для данных;
     + tk.Text для вывода отчёта.
   * Логика работы:
     + load\_dataset — загрузка CSV/ARFF;
     + reset\_dataset — откат к исходному DataFrame;
     + deidentify\_dataset — обобщение и k-анонимизация quasi-идентификаторов;
     + perform\_clustering — кодирование, отбор по компактности, иерархическая кластеризация, расчёт силуэтного коэффициента.
   * Вспомогательные методы:
     + update\_table\_display() — обновление модели таблицы;
     + update\_status() — обновление текстового поля статуса.
3. Отладка и расширение
   * Вывод status\_text удобно копировать для логов.
   * Для визуализации прогресса в консоли можно интегрировать tqdm, а для GUI — прогресс-бар ttk.Progressbar.
   * При необходимости добавьте настройки алгоритма (например, выбор метрики расстояния или метода связности) через дополнительные поля или комбобоксы.

# Исходный код программы

<https://github.com/FraaaM/Dataset_Clustering_and_Anonymizing>

# Используемый датасет

<https://www.kaggle.com/datasets/iamsouravbanerjee/customer-shopping-trends-dataset>

# Контрольный пример

1. Запуск приложения и первичная загрузка

Откроется окно программы с панелью управления слева и таблицей справа.

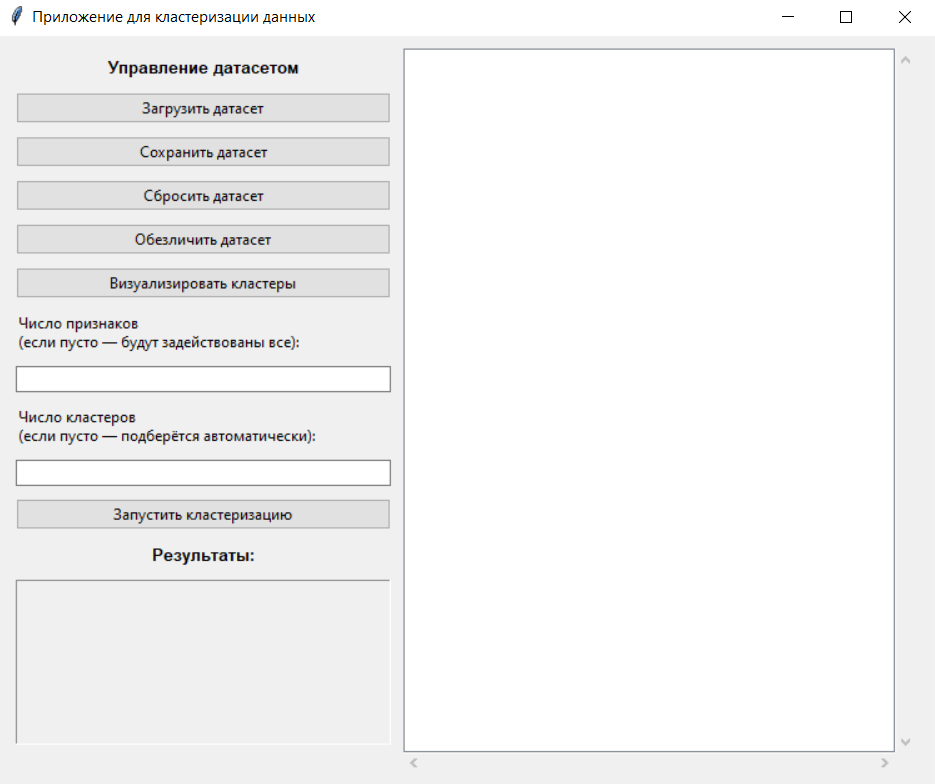


Рис 1. Главное окно data\_clustering\_app.py

1. Загрузка датасета

Нажмите «Загрузить датасет».

В диалоге выберите CSV или ARFF-файл с вашими данными (**не**обязательно наличие столбца `label`).

После загрузки таблица заполнится данными, а в поле статуса появится сообщение: Датасет загружен успешно.

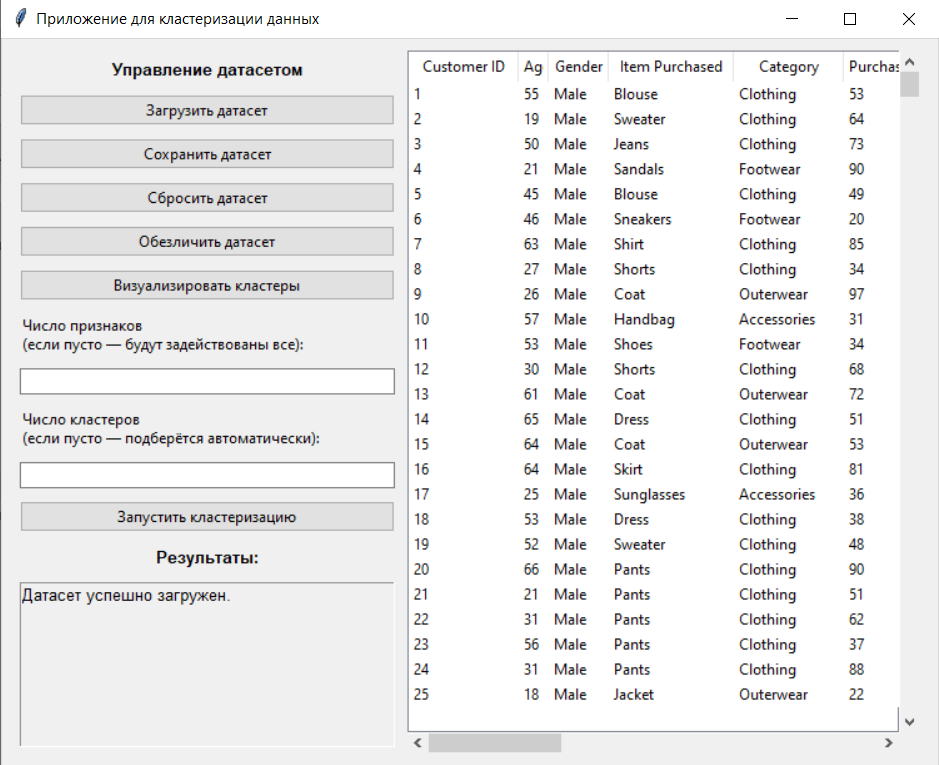


Рис 2. Пример загруженного датасета

1. Сброс к оригинальному виду

Если вы выполнили какие-то преобразования и хотите вернуться к изначальным данным:

* Нажмите «Сбросить датасет».
* Таблица вернётся к состоянию сразу после загрузки, в статусе появится: Датасет сброшен до оригинального состояния.

1. Обезличивание данных

* Нажмите «Обезличить датасет».
* Программа выполнит биннинг и группировку quasi-идентификаторов, затем рассчитает k-анонимность и распределение размеров групп.
* В статусе появятся строки вида:

k-анонимность датасета: 5

Размеры классов и их количество:

5: 8

6: 10

...

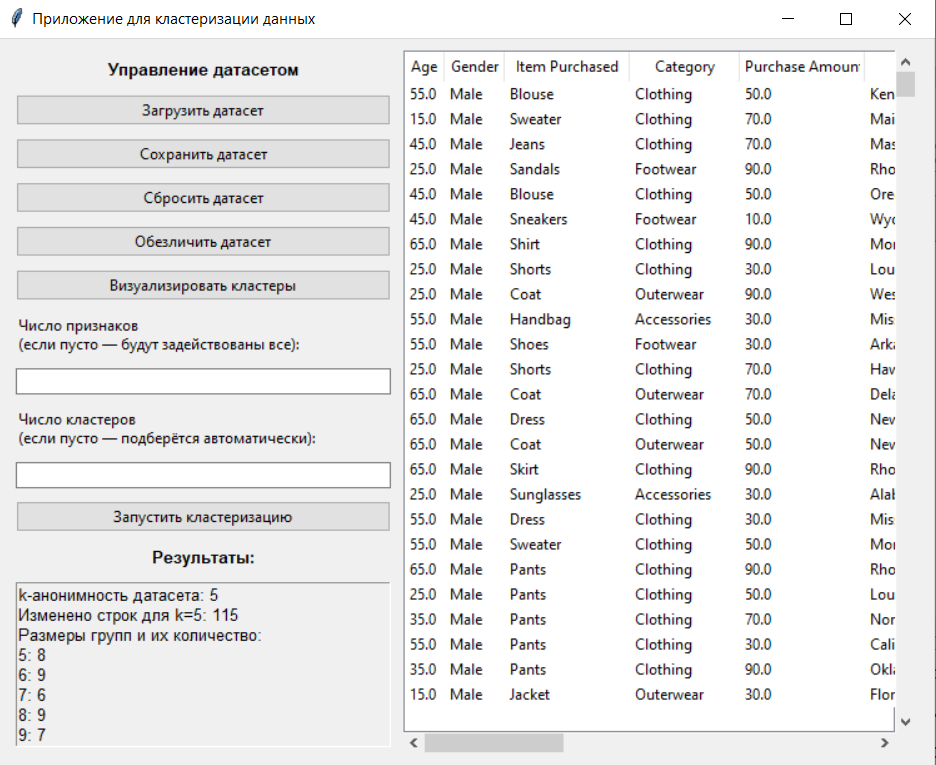


Рис 3. Пример обезличенного датасета

1. Отбор признаков и кластеризация

Введите в поле «Число признаков» желаемое число (например, 10) или оставьте 0 для использования всех признаков.  
Введите в поле «Число кластеров» желаемое число кластеров (например, 4) или не пишите ничего для автоматического выбора.  
Нажмите «Запустить кластеризацию».  
Программа выполнит:

* кодирование категориальных полей с помощью LabelEncoder;
* отбор ( k ) признаков по критерию компактности с использованием метрики Чебышева;
* иерархическую кластеризацию с методом средней связи и метрикой Чебышева;
* вычисление силуэтного коэффициента для оценки качества;
* замер времени.

В поле статуса отразятся результаты, например:

Отобрано признаков: 10

Число кластеров: 4 (эффективных: 4)

Отделимость кластеров: 0.4523

Время выполнения: 0.45 сек.

Распределение точек по кластерам:

Кластер 1: 150 точек

Кластер 2: 120 точек

Кластер 3: 100 точек

Кластер 4: 130 точек

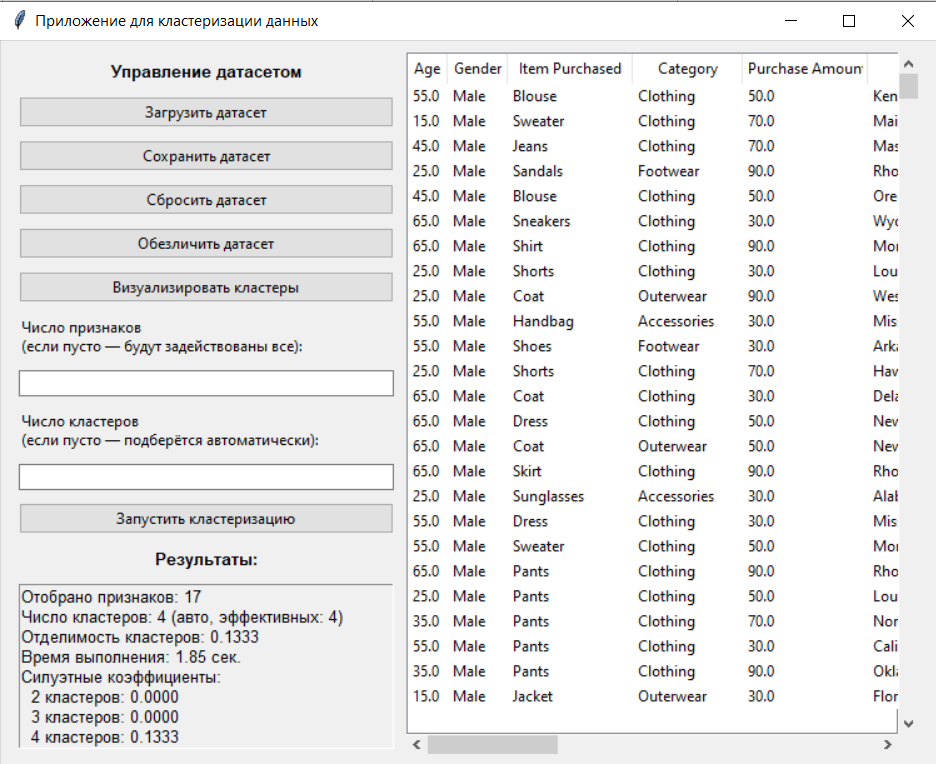


Рис 4. Пример кластеризации обезличенного датасета

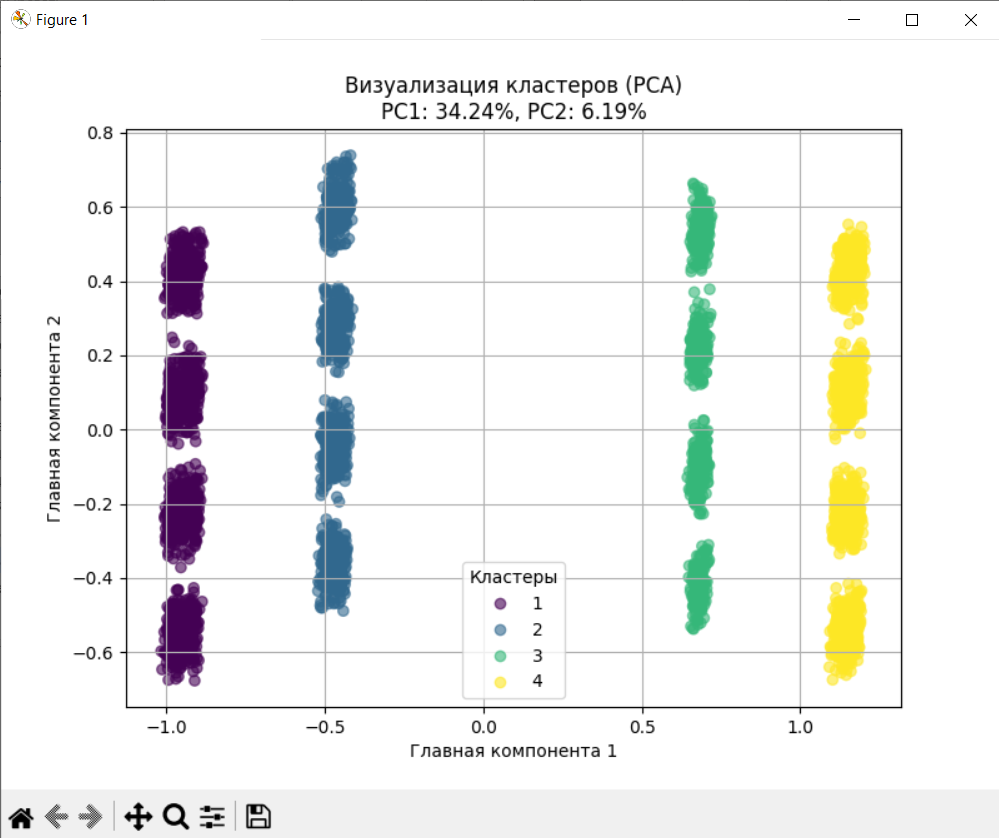
1. Повторная кластеризация и сравнение

Для сравнения Вы можете:

* Сразу после загрузки выполнить кластеризацию на всех признаках ();
* Повторить шаг 5 с отбором информативных признаков;
* Повторить кластеризацию после обезличивания и сверить показатели Отделимости кластеров и Силуэтных коэффициентов.

1. Визуализация кластеров

После кластеризации можно отобразить результат, нажав «Визуализировать кластеры» (см. Рис. 5.)

Рис 5. Визуализации вычисленных кластеров датасета

1. Завершение сеанса

Закройте окно приложения — программа автоматически завершит работу.

# Анализ результатов работы алгоритма

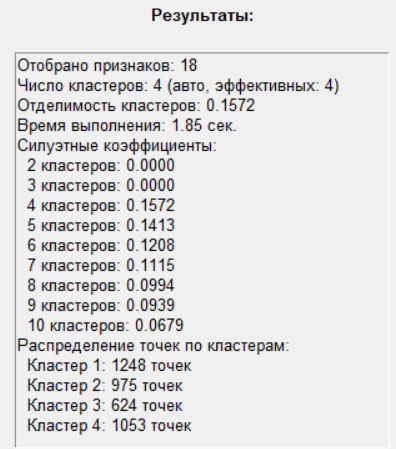
На основе Силуэтных коэффициентов и Распределения точек по кластером, программа вычисляет оптимальное число кластеров. Для большинства экспериментов наилучшее значение это 4 (Рис 6), однако в зависимости от количества выбранных наиболее информативных признаков вычисленное наилучшее число кластеров может меняться (Таблица 2).

Рис 6. Подбор оптимального числа кластеров

Таблица ниже составлена для изучения **влияния** числа признаков, подобранных через Компактность (плотность), и предварительной анонимизации на качество кластеризации Иерархическим методом (с расстоянием Чебышева (Chebychev)), анализ проводится с помощью оценки «Отделимость кластеров».

Таблица 2. Тесты кластеризации датасета

|  |  |  |  |
| --- | --- | --- | --- |
| **Число наиболее информативных признаков** | **Число подобранных кластеров** | **Отделимость кластеров** | **Время расчета (сек)** |
| **Без обезличивания** | | | |
| 2 | 2 | 0.6694 | 6.50 |
| 3 | 3 | 0.6694 | 9.02 |
| 4 | 3 | 0.6694 | 11.75 |
| 5 | 4 | 0.6693 | 14.70 |
| 6 | 4 | 0.4613 | 17.21 |
| 8 | 4 | 0.3436 | 22.37 |
| 12 | 4 | 0.2044 | 32.17 |
| Все (18) | 4 | 0.1572 | 1.91 |
| **С обезличиванием** | | | |
| 2 | 8 | 1.0000 | 5.82 |
| 3 | 10 | 0.9473 | 7.83 |
| 4 | 10 | 0.9473 | 10.11 |
| 5 | 9 | 0.7339 | 11.92 |
| 6 | 4 | 0.4672 | 14.23 |
| 8 | 4 | 0.2980 | 17.56 |
| 12 | 4 | 0.1793 | 24.99 |
| Все (17) | 4 | 0.1333 | 1.86 |

# 

Таблица содержит результаты кластеризации, выполненной программой data\_clustering\_app.py, для датасета с разным числом отобранных признаков, до и после обезличивания. Оцениваются следующие метрики: число наиболее информативных признаков, подобранное число кластеров, отделимость кластеров (силуэтный коэффициент с метрикой Чебышева) и время расчёта. Анализ разделён на две части: без обезличивания и с обезличиванием.

* **Анализ результатов без обезличивания**

1. **Качество кластеризации (отделимость кластеров)**:
   * При небольшом числе признаков (2–5) силуэтный коэффициент стабильно высок: 0.6694 для 2–4 признаков и 0.6693 для 5 признаков. Это указывает на хорошую компактность и отделимость кластеров: точки внутри кластеров близки друг к другу, а кластеры хорошо разделены.
   * С увеличением числа признаков качество кластеризации ухудшается:
     + Для 6 признаков силуэтный коэффициент падает до 0.4613, что всё ещё указывает на приемлемую, но уже менее чёткую кластеризацию.
     + Для 8 признаков значение снижается до 0.3436, а для 12 — до 0.2044, что говорит о слабой отделимости: кластеры начинают пересекаться, границы становятся нечёткими.
     + При использовании всех 18 признаков силуэтный коэффициент минимален (0.1572), что указывает на плохую кластеризацию: данные, вероятно, содержат избыточный шум, который мешает чёткому разделению.
   * **Вывод**: Оптимальное число признаков для кластеризации без обезличивания — 2–5, где достигается высокая отделимость (0.6693–0.6694). Увеличение числа признаков приводит к "проклятию размерности", снижая качество кластеризации.
2. **Число кластеров**:
   * Число кластеров увеличивается с ростом числа признаков: от 2 кластеров при 2 признаках до 4 кластеров при 5 и более признаках.
   * Это соответствует логике программы: при автоматическом подборе числа кластеров (методом максимизации силуэтного коэффициента) большее число признаков позволяет выявить более сложные структуры, но при этом качество кластеризации ухудшается из-за шума.
3. **Время расчёта**:
   * Время расчёта растёт с увеличением числа признаков: от 6.50 секунд для 2 признаков до 32.17 секунд для 12 признаков.
   * Однако при использовании всех 18 признаков время резко падает до 1.91 секунды. Это связано с тем, что отбор признаков (метод select\_features\_by\_compactness) не применяется, и программа сразу переходит к кластеризации, что снижает вычислительные затраты на отбор.

* **Анализ результатов с обезличиванием**

1. **Качество кластеризации (отделимость кластеров)**:
   * После обезличивания наблюдаются аномально высокие значения силуэтного коэффициента при малом числе признаков:
     + Для 2 признаков отделимость равна 1.0000, что указывает на идеальное разделение, но число кластеров (8) кажется нереалистичным для такого малого числа признаков.
     + Для 3–4 признаков отделимость составляет 0.9473, что также очень высокое значение, но число кластеров (10) выглядит завышенным.
   * С увеличением числа признаков качество начинает снижаться:
     + Для 5 признаков силуэтный коэффициент падает до 0.7339 (число кластеров — 9), что всё ещё указывает на хорошее разделение.
     + Для 6 признаков значение снижается до 0.4672, для 8 — до 0.2980, для 12 — до 0.1793, а при всех 17 признаках — до 0.1333, что указывает на плохую отделимость.
   * **Проблема**: Высокие значения силуэтного коэффициента при малом числе признаков (2–4) связаны с эффектом обезличивания: биннинг и k-анонимность (( k = 5 )) резко снижают вариативность данных, что приводит к искусственному "разделению". Однако большое число кластеров (8–10) при этом не отражает реальной структуры данных, а является артефактом метода подбора числа кластеров (find\_optimal\_clusters), который максимизирует силуэтный коэффициент.
2. **Число кластеров**:
   * После обезличивания число кластеров аномально велико при малом числе признаков: 8 для 2 признаков, 10 для 3–4 признаков, 9 для 5 признаков.
   * При 6 и более признаках число кластеров стабилизируется на 4, что совпадает с результатами без обезличивания.
   * **Вывод**: Обезличивание искажает структуру данных, особенно при малом числе признаков, что приводит к переоценке числа кластеров.
3. **Время расчёта**:
   * Время расчёта после обезличивания немного меньше, чем без обезличивания, для аналогичного числа признаков: например, 5.82 секунд для 2 признаков против 6.50 секунд.
   * Это связано с тем, что обезличивание уменьшает вариативность данных, упрощая вычисления в иерархической кластеризации.
   * При всех 17 признаках время минимально (1.86 секунд), как и в случае без обезличивания, из-за отсутствия этапа отбора признаков.

* **Сравнительный анализ**

1. **Влияние обезличивания**:
   * Обезличивание (биннинг числовых признаков, обобщение категориальных, k-анонимность) значительно влияет на качество кластеризации:
     + При малом числе признаков (2–4) обезличивание искусственно повышает силуэтный коэффициент (до 1.0000), но приводит к переоценке числа кластеров (8–10 вместо ожидаемых 2–4).
     + При большем числе признаков (6 и более) качество кластеризации после обезличивания сравнимо с результатами без обезличивания (например, 0.4672 против 0.4613 для 6 признаков), но всё ещё хуже при всех признаках (0.1333 против 0.1572).
   * **Вывод**: Обезличивание "сглаживает" данные, что может улучшить отделимость при малом числе признаков, но искажает реальную структуру данных, приводя к неадекватному числу кластеров.
2. **Оптимальное число признаков**:
   * Без обезличивания оптимальное число признаков — 2–5, где силуэтный коэффициент максимален (0.6693–0.6694).
   * С обезличиванием высокие значения силуэтного коэффициента (0.7339–1.0000) при 2–5 признаках вводят в заблуждение из-за завышенного числа кластеров. Более реалистичные результаты наблюдаются при 6 признаках (силуэтный коэффициент 0.4672, 4 кластера).
3. **Производительность**:
   * Время расчёта линейно растёт с увеличением числа признаков в обоих случаях, но обезличивание слегка ускоряет процесс за счёт уменьшения вариативности данных.
   * Использование всех признаков (без отбора) значительно быстрее (1.86–1.91 секунд), но даёт худшее качество кластеризации.

# Вывод

В рамках работы была разработана и исследована программа для кластеризации табличных данных с использованием:

* Метода кластеризации – иерархического алгоритма с методом средней связи.
* Метода измерения расстояния – метрики Чебышева (максимальное различие по признакам).
* Метода оценки качества – силуэтного коэффициента для оценки компактности и отделимости кластеров.
* Метода отбора признаков – компактности кластеров с использованием метрики Чебышева.
* **Основные результаты и выводы:**
* **Качество кластеризации**: Лучшие результаты без обезличивания достигаются при 2–5 признаках (силуэтный коэффициент 0.6693–0.6694, 2–4 кластера). После обезличивания высокие значения силуэтного коэффициента при малом числе признаков (2–4) являются артефактом, и реальное качество лучше оценивать при 6 признаках (0.4672, 4 кластера).
* **Влияние обезличивания**: Обезличивание искажает структуру данных, особенно при малом числе признаков, приводя к завышенному числу кластеров (до 10 при 2–4 признаках). Это подтверждает ранее выявленную проблему: после обезличивания данные часто группируются в 1 кластер, но метод подбора числа кластеров переоценивает их количество.
* **Производительность**: Отбор признаков увеличивает время расчёта (от 6.50 секунд для 2 признаков до 32.17 секунд для 12), но использование всех признаков (без отбора) даёт минимальное время (1.86–1.91 секунд) за счёт пропуска этапа отбора, хотя качество при этом страдает (силуэтный коэффициент 0.1572–0.1333).

Таким образом, сочетание иерархической кластеризации с метрикой Чебышева и отбора по компактности доказало свою эффективность и стабильность при выборе 2–5 признаков без обезличивания. Дополнительный этап обезличивания с отбором признаков позволяет достичь компромисса между приватностью данных и качеством кластеризации, однако требует оптимизации (увеличение числа бинов, снижение агрессивности k-анонимности) для минимизации искажений структуры данных.

# Источники

* **pandas** — Библиотека для анализа и обработки данных.
  + <https://pandas.pydata.org/docs/>
  + Дата обращения: 22.05.2025
* **numpy** — Библиотека для работы с многомерными массивами и математическими операциями.
  + <https://numpy.org/doc/stable/>
  + Дата обращения: 22.05.2025
* **scipy** — Библиотека для научных вычислений (включая загрузку ARFF-файлов, иерархическую кластеризацию и метрики расстояния).
  + <https://docs.scipy.org/doc/scipy/>
  + Дата обращения: 22.05.2025
* **tkinter** — Библиотека для создания графических интерфейсов в Python.
  + <https://docs.python.org/3/library/tkinter.html>
  + Дата обращения: 22.05.2025
* **scikit-learn** — Библиотека для машинного обучения (кластеризация, оценка качества с помощью силуэтного коэффициента, предобработка данных).
  + <https://scikit-learn.org/stable/>
  + Дата обращения: 22.05.2025
* **matplotlib** — Библиотека для визуализации данных (графики рассеяния для кластеров).
  + <https://matplotlib.org/stable/>
  + Дата обращения: 22.05.2025