DOI 10.58351/2949-2041.2026.34.5.049

Феофанов Артем Тимофеевич, Студент
Сахалинский государственный университет

Научный руководитель:
Осипов Геннадий Сергеевич,
д.т.н., профессор кафедры информатики,
Сахалинский государственный университет

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ КЛАСТЕРИЗАЦИИ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ SCIKIT-LEARN ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

Аннотация. в работе рассматриваются пять методов кластеризации для набора данных «Ирисы Фишера». Для обучения моделей использовалась библиотека scikit-learn языка программирования Python. Все методы были оценены по нескольким метрикам: время выполнения, коэффициент силуэта, индекс Калински-Харабаса, индекс Дэвиса-Болдина, скорректированный индекс Рэнда.

Ключевые слова: кластеризация, машинное обучение, K-Means, K-Means++, K-Medoids, центроидные методы кластеризации, метод Уорда, иерархические методы кластеризации, DBSCAN, пространственные методы кластеризации, метод силуэтов, метод «локтя».

Введение

В условиях стремительного роста количества цифровых данных возникает потребность в навыке эффективного и быстрого анализа и структурирования информации. Одной из известных проблем в области машинного обучения и data science является задача кластеризации. Кластеризация – это многомерная статистическая процедура, предназначенная для разбиения объектов на группы, таким образом, чтобы элементы, входящие в одну группу, были максимально «похожи» друг на друга, а элементы из разных групп максимально «отличались» между собой. Эта задача относится к классу обучения без учителя, так как размеченного набора данных с выходным полем нет.

В данный момент для работы с машинным обучением популярным выбором является высокоуровневый язык программирования Python. Он обладает огромным числом готовых библиотек и широко поддерживается сообществом. Для кластеризации мы воспользуемся инструментом scikit-learn. Он является бесплатной библиотекой для машинного обучения, в состав которой входят средства для классификации, регрессионного и кластерного анализа данных.

В рамках задачи кластеризации нами будут рассмотрены три группы методов (и их известные алгоритмы): иерархические (метод Уорда), пространственные (DBSCAN) и центроидные (K-Means, K-Means++, K-Medoids). Сравнение методов будет происходить по нескольким метрикам: время выполнения, коэффициент силуэта, индекс Дэвида-Болдина, индекс Калински-Харабаса, скорректированный индекс Рэнда [1].

Характеристика и инициализация методов кластеризации

Для анализа методов кластеризации нам необходим набор данных, подходящий для нашей задачи. Используем известные «Ирисы Фишера» – 150 экземпляров цветков ирисов, равномерно распределенных по трем биологическим видам (по 50 объектов на каждый вид): Iris, Iris Versicolor и Iris Virginica. Каждый объект характеризуется четырьмя признаками: длиной и шириной чашелистика, а также длиной и шириной лепестка. Scikit-learn предоставляет возможность воспользоваться этим датасетом с помощью функции load_iris() [2] (рис. 1).

```
# Загрузка данных
iris = load_iris()
X = iris.data
y_true = iris.target
```

Рисунок 1. Инициализация «Ирисов Фишера»

Из-за того, что алгоритмы используют евклидово расстояние для формирования кластеров, нам необходимо выполнить z-преобразование значений признаков, для которого существует формула:

$$x' = \frac{x - \overline{x}}{\sigma_x}$$

Используем функционал библиотеки для получения измененного датасета (рис. 2):

```
# Стандартизация данных
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

Рисунок 1. Стандартизация датасета

Теперь мы можем приступать к обучению моделей, используя следующие алгоритмы кластеризации:

- Метод K-Means, концептуально сформулированный Штейнгаузом и Ллойдом в 1957 году, является самым популярным алгоритмом кластеризации. В основе его функционирования лежат следующие положения: число кластеров жестко задается заранее; по умолчанию используется Евклидова метрика; глобальная математическая цель алгоритма – минимизировать целевую функцию, представляющую собой сумму квадратов внутрикластерных расстояний:

$$J = \sum_{i=1}^{k} \sum_{x \in S_i} (x - \mu_i)^2.$$

Для обучения с помощью алгоритма в scikit-learn есть встроенная функция KMeans(), для которой можно задать точное количество кластеров с помощью параметра n_cluster (рис. 3).

```
# K-Means со случайной инициализацией
k_means = KMeans(n_clusters=3, init='random', n_init=1, random_state=10)
k_means = k_means.fit_predict(X)
```

Рисунок 3. Инициализация K-Means

Алгоритм выполняется итеративно: случайным образом выбираются k начальных центроидов; каждый объект привязывается к ближайшему центру; центроиды пересчитываются как среднее арифметическое координат привязанных точек. Процесс повторяется до стабилизации центров. Главные достоинства – математическая простота и высокая скорость работы. Недостатки – чувствительность к выбросам, неспособность выделять невыпуклые формы и жесткая зависимость от начального случайного выбора центроидов из-за риска застревания в локальных минимумах.

- Метод K-Means++ призван устранить главный недостаток классического K-Means – хаотичную начальную инициализацию центров. Суть модификации заключается в специфической схеме предварительного «растаскивания» центроидов друг от друга. Первый центроид выбирается случайно, а каждый последующий – из оставшихся объектов выборки с вероятностью, пропорциональной квадрату расстояния от него до уже существующих

центроидов. Этот подход гарантирует, что начальные центры будут распределены по всему объему данных, что резко сокращает число итераций до сходимости и существенно повышает качество финального разбиения. Данный метод можно инициализировать с помощью параметра init = 'k-means++' в функции KMeans() (рис. 4):

```
# K-Means++
k_means_plus = KMeans(n_clusters=3, init='k-means++', n_init=1, random_state=10)
k_means_plus = k_means_plus.fit_predict(X)
```

Рисунок 4. Инициализация K-Means++

 Метод K-Medoids разработан для преодоления уязвимости K-Means к шуму и аномалиям. Вместо абстрактного математического центра (среднего арифметического), в качестве центра кластера, или, так называемой, медоиды, выбирается реальный объект из набора данных, для которого сумма расстояний до всех остальных точек кластера минимальна. Поскольку медоида является реальной точкой, экстремальный единичный выброс не способен сместить её положение так же радикально, как он смещает среднее значение в K-Means. Платой за помехоустойчивость является квадратичная вычислительная сложность, ограничивающая применение метода на больших данных. Для обучения данным методом используется функция KMedoids() (рис. 5).

```
# K-Medoids
k_medoids = KMedoids(n_clusters=3, random_state=10)
k_medoids = k_medoids.fit_predict(X)
```

Рисунок 2. Инициализация K-Medoids

 Метод Уорда является представителем иерархических агломеративные методов. Их суть заключается в том, на старте каждый объект объявляется отдельным самостоятельным кластером. Затем алгоритм начинает последовательно попарно объединять ближайшие кластеры, формируя древовидную структуру соподчиненности – дендрограмму. Процесс продолжается до тех пор, пока все объекты не сольются в один единичный кластер. Для определения расстояния между кластерами используются различные типы связи. Метод Уорда, в свою очередь, минимизирует прирост общей внутрикластерной дисперсии при объединении групп. Иерархическая кластеризация не требует случайной инициализации и не требует априорного задания числа кластеров. В scikit-learn этот метод определяется функцией AgglomerativeClustering() с параметром linkage = 'ward' (рис. 6).

```
# Метод Уорда
ward_method = AgglomerativeClustering(n_clusters=3, linkage='ward')
ward_method = ward_method.fit_predict(X)
```

Рисунок 6. Инициализация метода Уорда

 Алгоритм DBSCAN – один из ярких представителей пространственных методов. Основной принцип работы плотностных алгоритмов – это выделение областей высокой концентрации точек, которые разделяются областями с низкой плотностью [3]. Точки, оказавшиеся в области с низкой плотностью, определяются как «шум». Для DBSCAN основными параметрами на входе являются: радиус окрестности, где, если расстояние между любыми двумя точками меньше или равно заданному радиусу, эти точки считаются соседними, и минимальное количество точек, которое может быть в одном кластере (чем больше набор данных, тем больше точек должно быть выбрано). Инициализируем данный метод с помощью функции DBSCAN() (рис. 7).

```
# DBSCAN
dbscan = DBSCAN(eps=0.455, min_samples=5)
dbscan = dbscan.fit_predict(X)
```

Рисунок 7. Инициализация DBSCAN

Результаты кластеризации и сравнительный анализ

Для методов K-Means, K-Means++, K-Medoids и Уорда требуется задать количество кластеров для поиска. Существует два известных метода для подбора оптимального количества кластеров:

- Метод «Локтя» – оценивает изменение суммы квадратов внутрикластерных расстояний до центра кластера (данную метрику принято называть инерцией) на различных количествах кластеров и выбирает тот вариант, при котором данное расстояние перестает существенно уменьшаться. Для поиска нужного значения проведем несколько итераций кластеризации с помощью алгоритма K-Means, каждый раз изменяя количество кластеров от 1 до 11. Благодаря библиотеке Kneed и её метода KneeLocator() [4], можно автоматически определить оптимальное значение кластеров (рис. 8).

```
# Автоматически находим оптимальное количество кластеров
k_range = range(1, 11)
inertias = []

for k in k_range:
    k_means = KMeans(n_clusters=k, random_state=10, n_init=1, init='random')
    k_means.fit(X)
    inertias.append(k_means.inertia_)

kneedle = KneeLocator(list(k_range), inertias, curve="convex", direction="decreasing")
```

Рисунок 8. Обнаружение точки «локтя» с помощью библиотеки Kneed

Построим график, на котором будет отображено изменение инерции от количества кластеров, отобразим на нем полученное значение «локтя». Таким образом, оптимальное значение кластеров равно 3 (рис. 9).

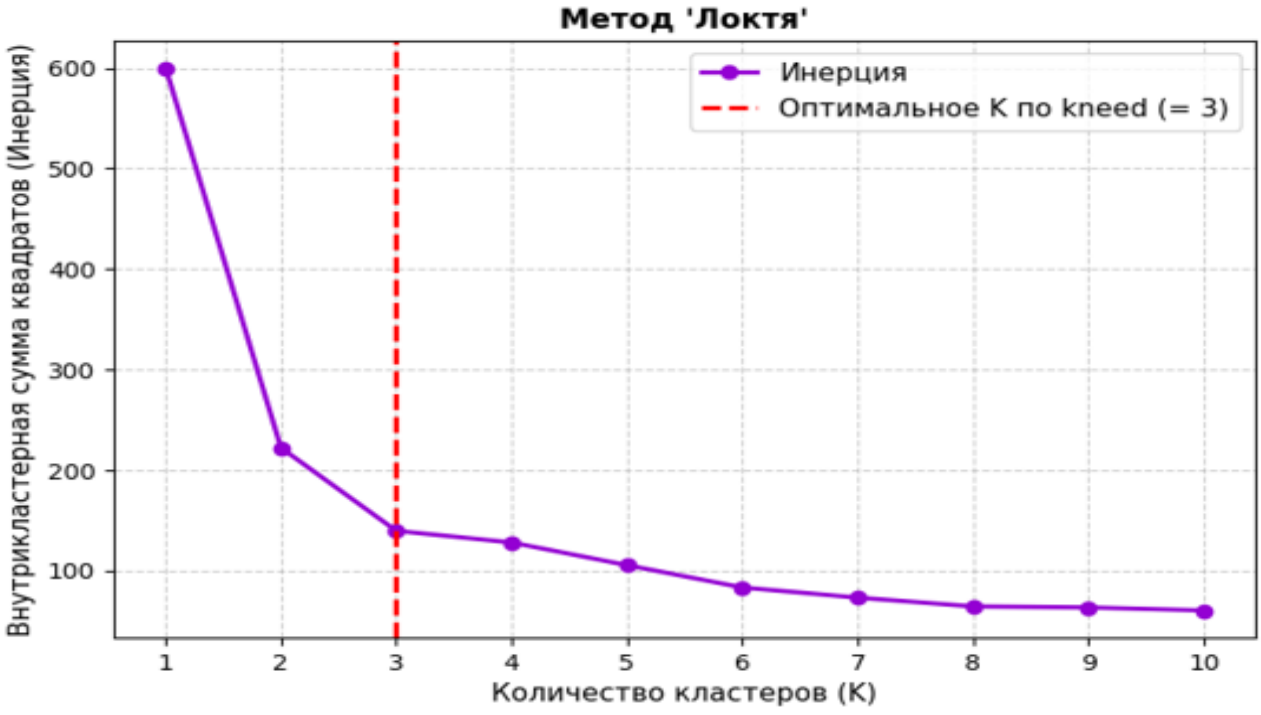

Рисунок 9. График метода «локтя», отображающий оптимальное количество кластеров

 Метод Силуэтов – показывает, какие объекты «уверенно» лежат внутри кластера, а какие занимают пограничное положение, вычисляя их силуэт по формуле:

$$s(i) = \frac{b(i) - a(i)}{\max[a(i), b(i)]},$$

где a – среднего внутрикластерного расстояние, b – среднее расстояние до ближайшего кластера. Построим графики силуэтов для количества кластеров от 2 до 5 (см. рис. 10 и рис. 11).

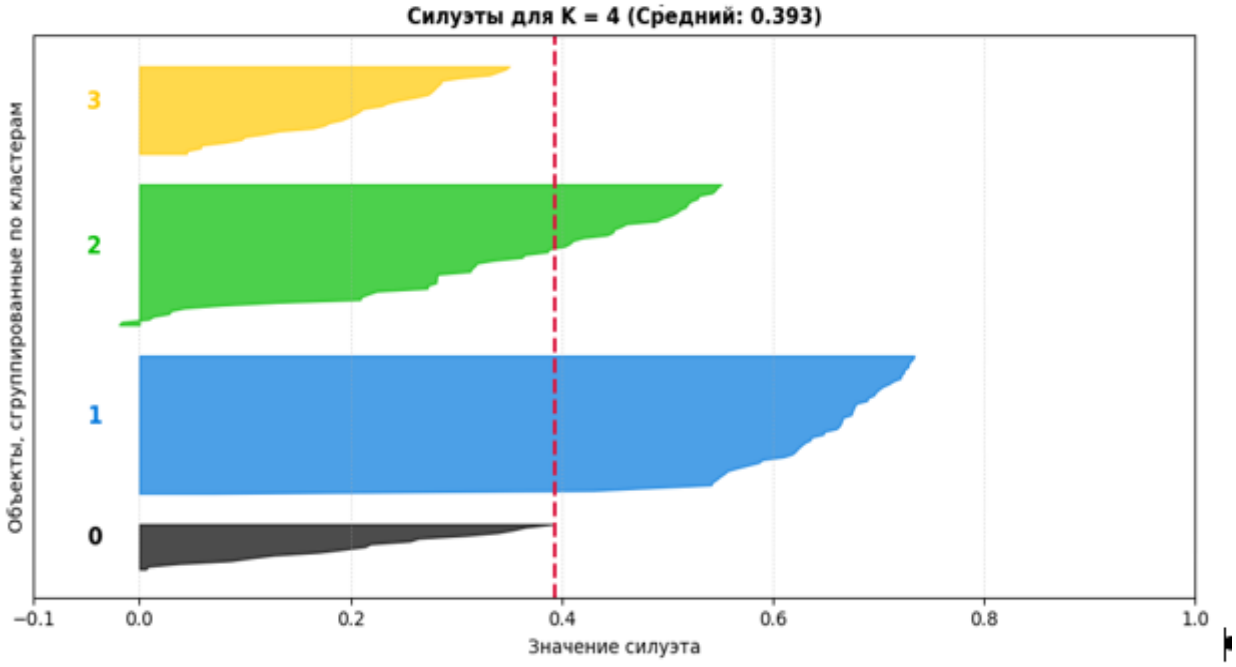

Рисунок 10. График силуэтов для k = 4

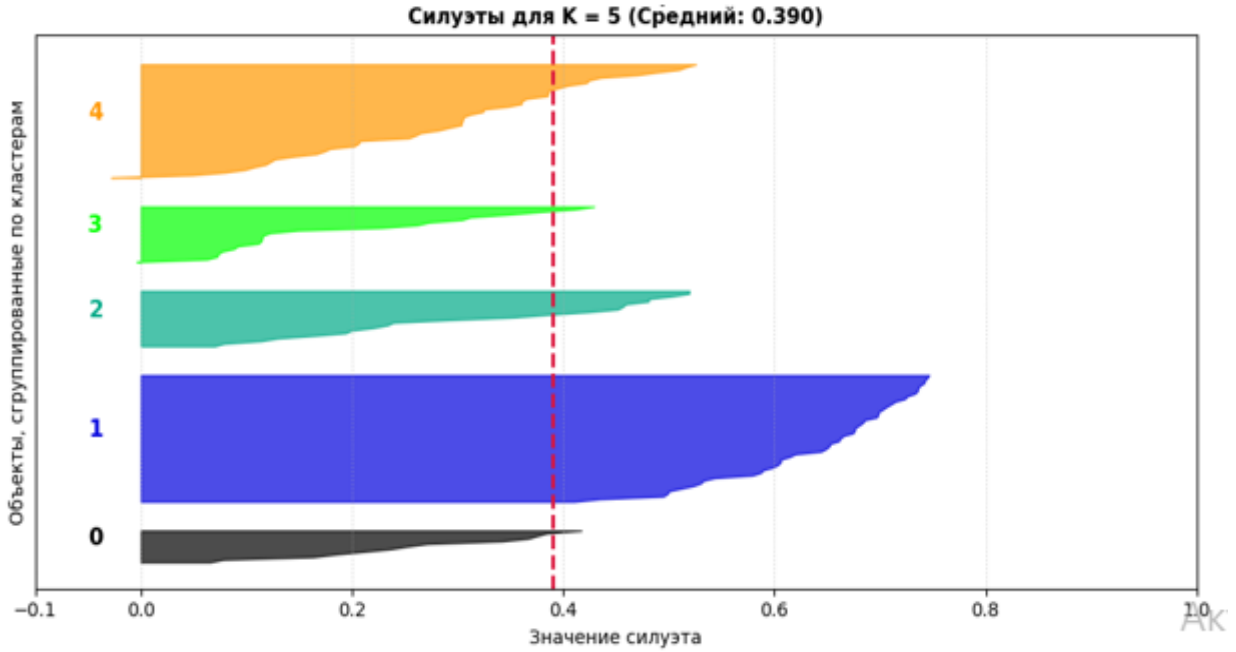

Рисунок 11. График силуэтов для k = 5

Как можно заметить, k = 4 (рис. 10) и k = 5 (рис. 11) можно смело отбрасывать, т. к. кластеры неоднородны и некоторые из них не пересекают среднюю линию силуэта. Остаются k = 2 (рис. 12) и k = 3 (рис. 13). Высота кластеров для k = 2 сильно отличается, в то время как в k = 3 все три кластера имеют сопоставимую толщину. Делаем вывод, что о том, что оптимальное количество кластеров равно 3.

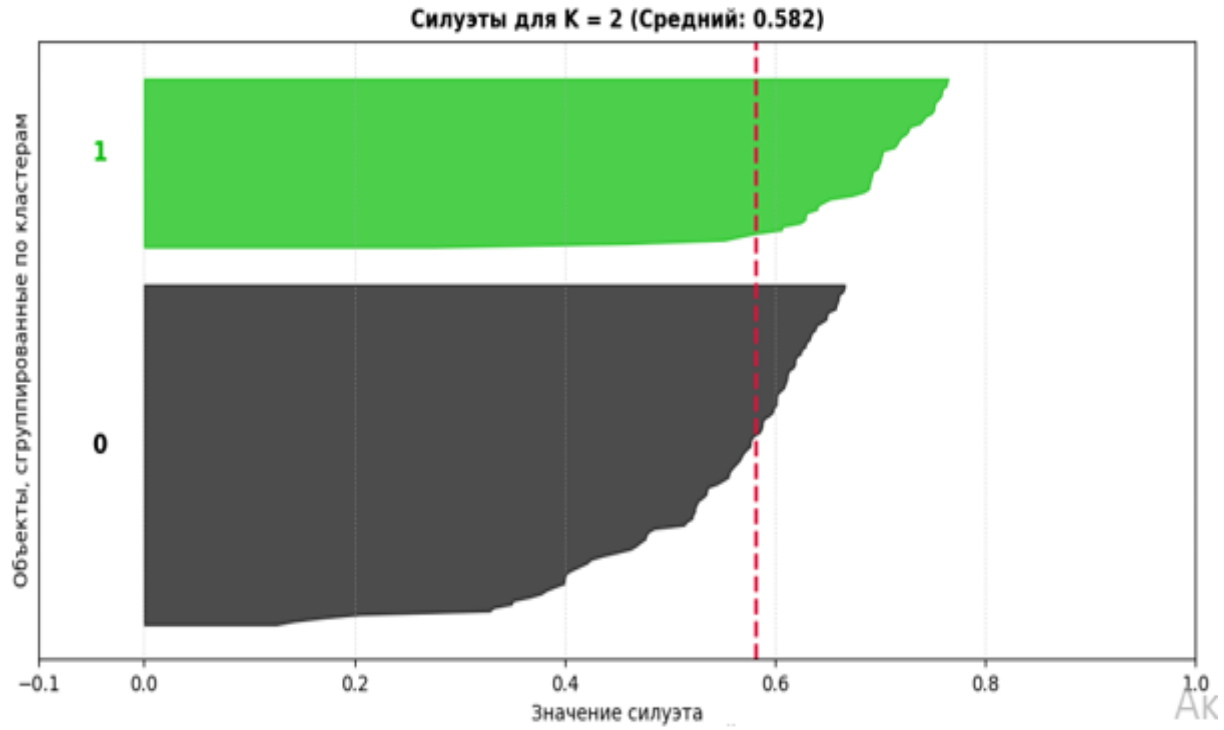

Рисунок 12. График силуэтов для k = 2

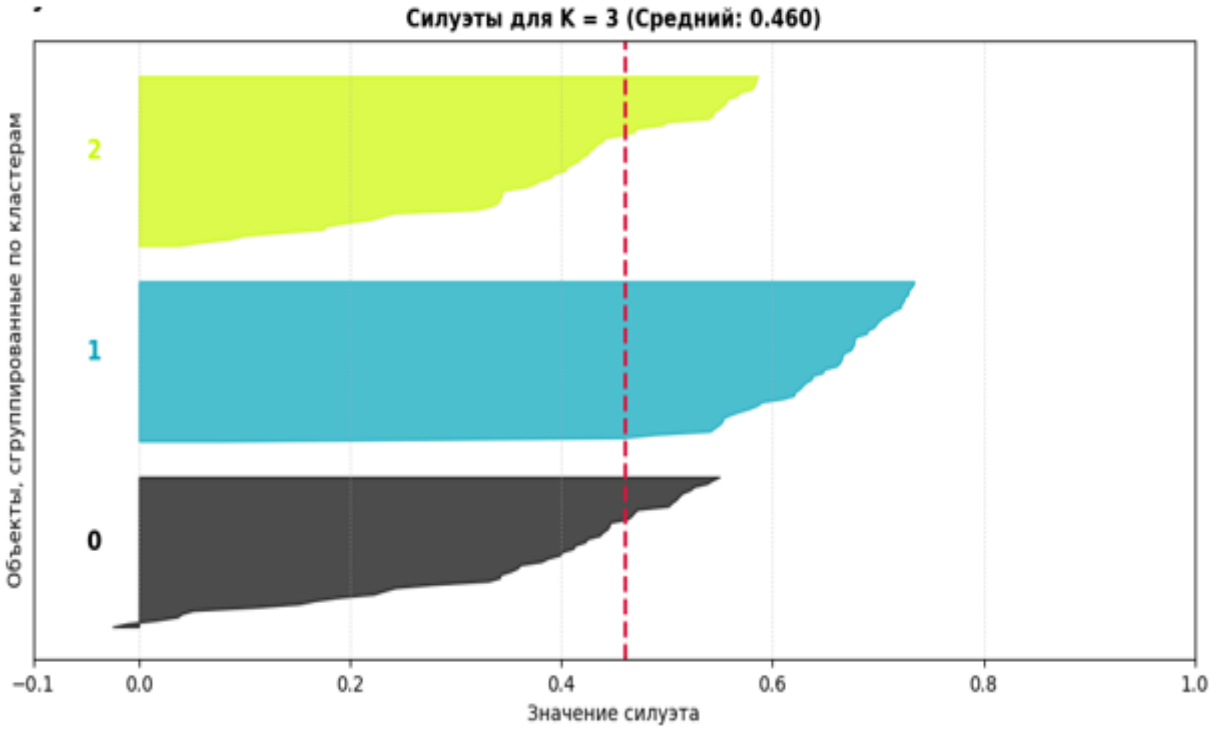

Рисунок 13. График силуэтов для k = 3

В случае метода DBSCAN, на вход требуются параметры eps (радиус окружности) и min_samples (минимальное количество точек в кластере). Методом перебора были выбраны значения 0.455 и 5 соответственно, чтобы получилось приблизительное разделение на 3 кластера. Результаты кластеризации всеми алгоритмами отображены на полученных графиках (рис. 14, рис. 15, рис. 16).

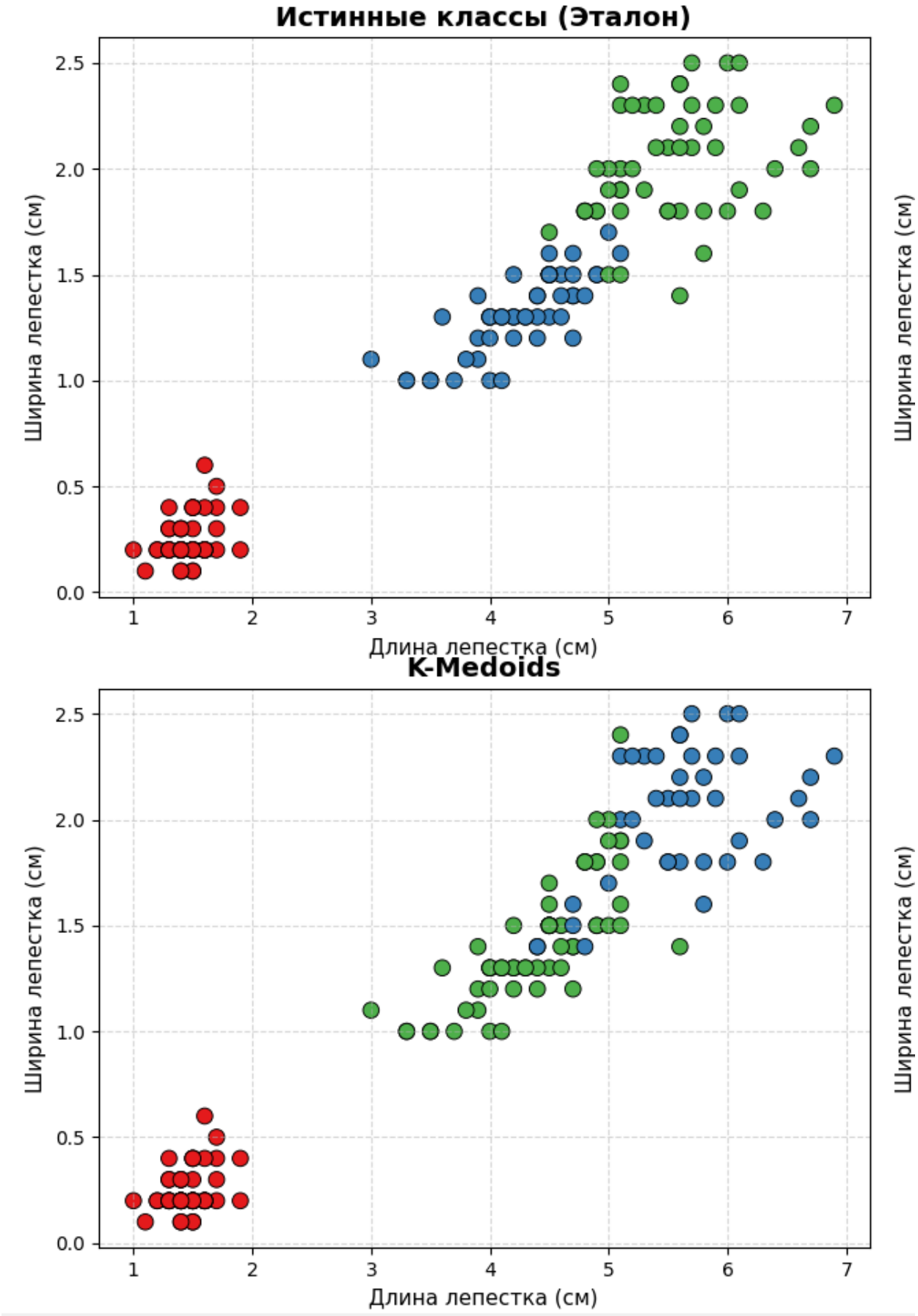

Рисунок 14. График разделения на кластеры для эталона и K-Medoids

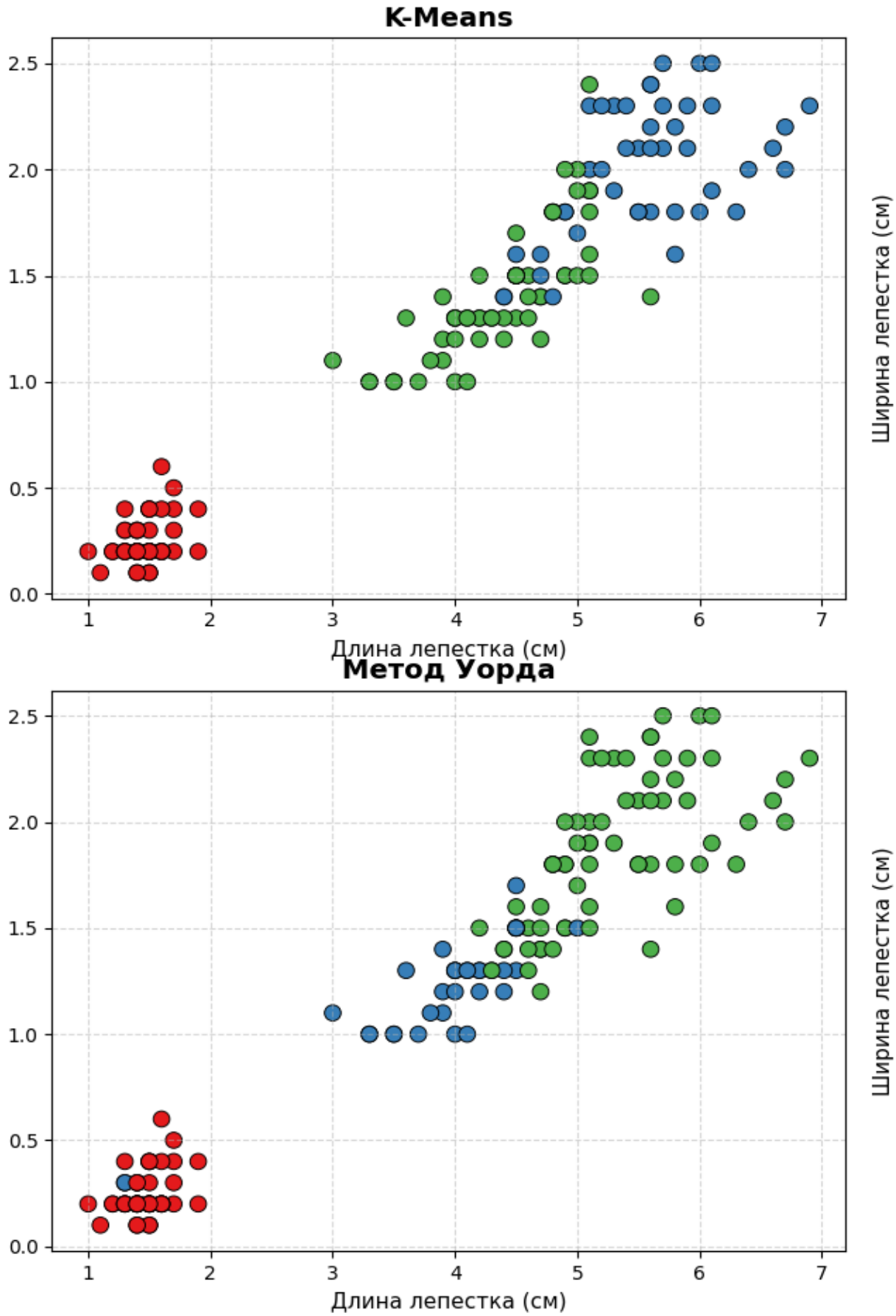

Рисунок 15. Графики разделения на кластеры для методов Уорда и K-Means

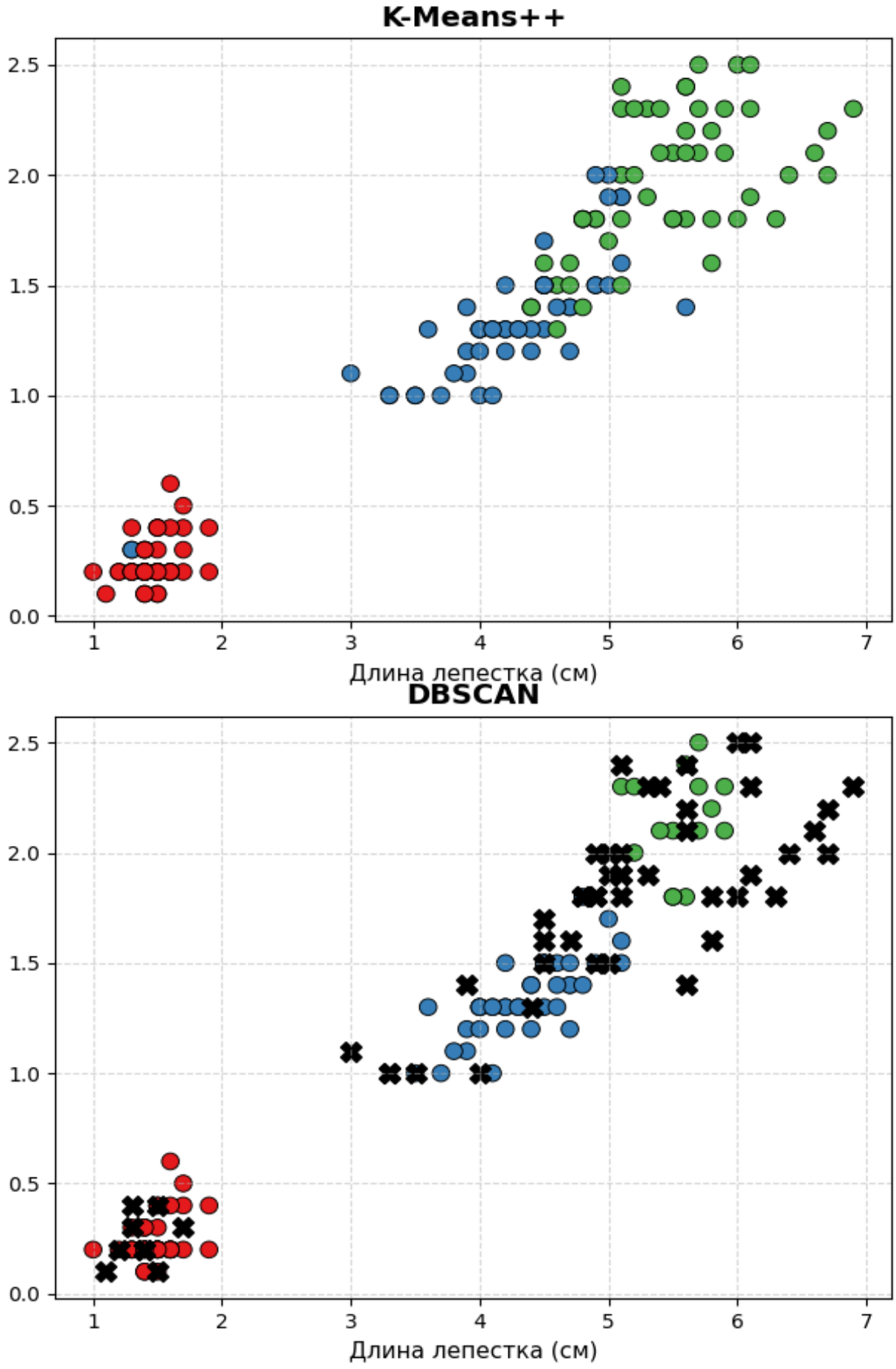

Рисунок 16. Графики разделения на кластеры для методов K-Means++ и DBSCAN

Чтобы сравнить каждый из методов, будем использовать следующие метрики:

- Время, затраченное на кластеризацию.
- Коэффициент силуэта, который служит критерием для оценки выраженности кластерной структуры. Лежит в диапазоне от -1 до 1. Чем ближе к единице, тем лучше кластеризация.
- Индекс Калински-Харабаса, который определяется как отношение суммы межкластерной дисперсии к сумме внутрикластерной дисперсии. Лежит в диапазоне от 0 до бесконечности. Чем больше, тем лучше.
- Индекс Дэвиса-Болдина, означающий среднее «сходство» между кластерами, где сходство – это мера, сравнивающая расстояние между кластерами с размером самих кластеров. Лежит в диапазоне от 0 до бесконечности. Чем ближе к нулю, тем лучше.
- Скорректированный индекс Рэнда, измеряющий сходство между двумя назначениями кластеров, учитывая все пары выборок и подсчитывая пары, которые либо сгруппированы в один и тот же кластер, либо в разные кластеры в обоих назначениях. Лежит в диапазоне от -1 до 1. Чем ближе к 1, тем лучше.

Выполнив кластеризацию каждым методом, получаем данные, записанные в таблице.

Таблица

«Сравнительный анализ методов кластеризации»

Метод кластеризации	**Время (мс)**	**Коэфф. силуэта**	**Калински-Харабас**	**Дэвис-Болдин**	**Скорректированный индекс Рэнда**
K-Means	2.19	0.4599	241.9	0.8336	0.6201
K-Means++	2.82	0.4565	239.48	0.8275	**0.6451**
K-Medoids	1.67	0.459	239.75	0.8385	0.6312
Метод Уорда	20.76	0.4467	222.72	0.8035	0.6153
DBSCAN	**1.6**	**0.545**	**246.94**	**0.6594**	0.5171

По времени выполнения лидирует метод DBSCAN (1.6 мс), сразу за ним с небольшим отставанием следует метод K-Medoids (1.67 мс). Хуже всех справился метод Уорда (20.76 мс), в силу своей схемы работы и необходимости пошагового вычисления древовидных структур.

В лидерах по коэффициенту силуэта оказался DBSCAN (0.545). Остальные методы не сильно отличаются друг от друга. Самое худшее значение – метод Уорда (0.4467).

Максимальное значение индекса Калински-Харабаса имеет DBSCAN (246.94). K-методы оказались близки друг к другу (241.9, 239.48 и 239.72). Минимальный индекс – метод Уорда (222.72).

По индексу Дэвиса-Болдина лучше всех оказался DBSCAN (0.6594). Наименее предпочтительными стали K-методы (0.8336, 0.8275 и 0.8385).

Лидером по скорректированному индексу Рэнда стал K-Means++ (0.6451), а худшим стал метод DBSCAN (0.5171), т.к. выделение «шума» приводит к расхождениям со структурой исходного набора данных.

Вывод

В ходе выполнения сравнительного анализа пяти методов кластеризации можно прийти к выводу, что ни один из методов не является универсальным по всем метрикам одновременно. Это доказывает теорему невозможности Клейнберга о том, что не существует оптимального алгоритма кластеризации.

Иерархический метод Уорда показал хорошее качество кластеризации по индексу Дэвиса-Болдина (0.8035), но из-за особенностей своей работы оказался самым долгим по времени выполнения (20.76 мс).

Пространственный метод DBSCAN стал лидером по времени выполнения (1.6 мс) и внутренне- и межкластерным метрикам, однако автоматическое определение «шума» привели к последнему месту по скорректированному индексу Рэнда (0.5171).

Центроидные методы (K-Means, K-Means++, K-Medoids) оказались лидерами по соответствию истинной разметке (индексу Рэнда) и продемонстрировали в среднем сбалансированные результаты. K-Medoids занял второе место по скорости выполнения (1.67 мс), а K-Means++ – лидер скорректированному индексу Рэнда (0.6451).

Таким образом, в зависимости от поставленных условий для задачи кластеризации, можно подобрать подходящий алгоритм кластеризации. Если у нас нет привязки к эталонной разметке, то лучше всех выглядит метод DBSCAN. Иначе, правильнее будет использовать алгоритм K-Means++, который демонстрирует самые сбалансированные показатели.

Список литературы:

1. Викиконспекты ИТМО. Оценка качества в задаче кластеризации [Электронный ресурс]. – URL: https://neerc.ifmo.ru/wiki/index.php?title=%D0%9E%D1%86%D0%B5%D0%BD%D0%BA%D0%B0_%D0%BA%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%B0_%D0%B2_%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B5_%D0%BA%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8 (Дата обращения 31.05.2026)

2. scikit-learn: официальная документация. The Iris Dataset [Электронный ресурс]. – URL: https://scikit-learn.org/1.4/auto_examples/datasets/plot_iris_dataset.html, (Дата обращения 31.05.2026)

3. Турашев А. С., Сухомлин В. А. Выявление аномалий в поведении ЦП с применением алгоритмов кластеризации библиотеки Scikit-Learn языка программирования Python [Текст]. – URL: https://www.elibrary.ru/download/elibrary_80596709_72761819.pdf, (Дата обращения 31.05.2026)

4. kneed: официальная документация. [Электронный ресурс]. – URL: https://kneed.readthedocs.io/en/stable, (Дата обращения 31.05.2026)