DOI 10.37539/2949-1991.2026.39.4.005

Гусев Владислав Михайлович, Студент,
Сахалинский государственный университет

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

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ОЦЕНКИ КАЧЕСТВА КЛАСТЕРИЗАЦИИ НА ПРИМЕРЕ АЛГОРИТМА K-MEANS И НАБОРА ДАННЫХ IRIS

Аннотация. В работе представлен систематический анализ внутренних метрик оценки качества кластеризации – коэффициента силуэта и индекса Данна. Теоретические положения проиллюстрированы практическим примером кластеризации классического датасета ирисов Фишера методом k-means. Проведён сравнительный анализ полученных значений коэффициента силуэта и индекса Данна, визуализированы результаты с помощью PCA-проекций и диаграмм силуэтов. Реализация моделей выполнена на языке Python.

Ключевые слова: Кластеризация, коэффициент силуэта, индекс Данна, оценка качества кластеризации.

Постановка задачи исследования

Цель данной работы: провести сравнительный анализ методов оценки качества кластеризации на примере алгоритма k-means и классического набора данных ирисы Фишера

Задачи исследования:

1. Реализовать или применить алгоритм k-means для кластеризации данных ирисов, используя только признаки объектов (без использования информации о видах).
2. Выполнить кластеризацию при различных значениях числа кластеров k (от 2 до 5).
3. Оценить качество полученных кластерных структур с помощью двух внутренних метрик:
4. Коэффициент силуэта (Silhouette coefficient), оценивающий компактность и отделимость кластеров на уровне отдельных объектов.
5. Индекс Данна (Dunn index), оценивающий отношение минимального межкластерного расстояния к максимальному внутрикластерному разбросу.
6. Провести сравнительный анализ полученных оценок и определить «оптимальное» число кластеров с точки зрения каждой метрики.

Сопоставить полученные результаты с известной истинной структурой данных (3 вида ирисов) и объяснить возможные расхождения, связанные с особенностями метода k-means и выбранных индексов качества.

Введение в задачу кластеризации

Алгоритмы кластеризации направленны на организацию данных в группы или кластеры на основе внутренних закономерностей и сходств внутри данных [1]

Кластеризация – это одна из задач машинного обучения без учителя, которая заключается в группировке множества объектов на подмножества таким образом, чтобы объекты внутри одного кластера были максимально похожи друг на друга, а объекты из разных кластеров максимально различны. Алгоритм кластеризации предполагает, что данные, находящиеся в одном кластере, должны иметь похожие свойства, а точки данных, находящиеся в разных кластерах, должны сильно различаться.

Кластеризация довольно часто применяется для решения широкого круга задач:

- Сегментация клиентов: разбивка аудитории на группы для персонализации рекомендаций.
- Обработка изображений: сегментация изображений, сжатие цифровой палитры.
- Анализ социальных сетей: выделение сообществ по интересам.
- Кластеризация генов или видов растений или животных.
- Анализ текстов: тематическое моделирование, группировка новостей по темам [2].
- Поиск аномалий: объекты, не попавшие ни в один кластер или попавшие в очень маленькие кластеры могут быть выбросами.
- Алгоритмы приближённого поиска векторных представлений данных [3].

Результатом работы алгоритма кластеризации является разбиение исходной выборки на кластеры. Каждому объекту присваивается номер кластера, к которому он был отнесён. При этом от алгоритма не ожидается интерпретация кластеров, например красные или синие ирисы.

Виды метрик расстояния

Евклидово расстояние ($L2$-расстояние) в основном используется для вычисления расстояния между любыми двумя точками в произвольном пространстве. Согласно формуле евклидова расстояния, расстояние между любыми двумя точками (A, B) на плоскости с координатами (x, y) и определяется следующей формулой:

$$dist(A,B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \quad (1)$$

Визуализация евклидового расстояния приведена на рисунке 1.

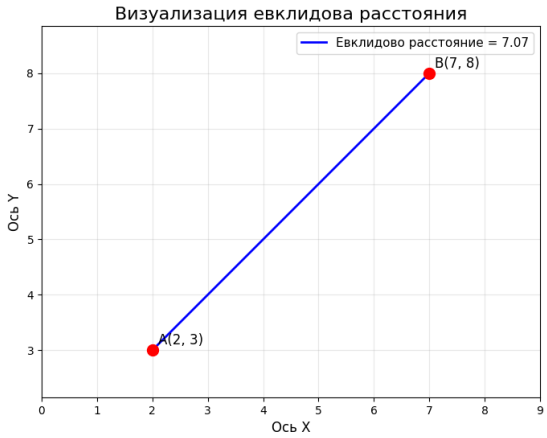

Рисунок 1. Евклидово расстояние

Эвклидово расстояние интуитивно понятно и представляет собой длину прямой линии между объектами и рассчитывается на основе теоремы Пифагора по их координатам. Для того чтобы подчеркнуть разницу между сильно удалёнными объектами можно использовать квадрат евклидова расстояния, который придаёт больший вес удалённым друг от друга точкам.

Манхэттенское расстояние ($L1$-расстояние) – определяется как сумма абсолютных разностей координат двух точек по всем измерениям. Название связанно с сетчатой структурой улиц Манхеттена. Кратчайшее расстояние между двумя точками в такой системе равно сумме пройденных кварталов.

$$dist(A,B) = |x_A - x_B| + |y_A - y_B| \quad (2)$$

Визуализация манхэттенского расстояния приведена на рисунке 2.

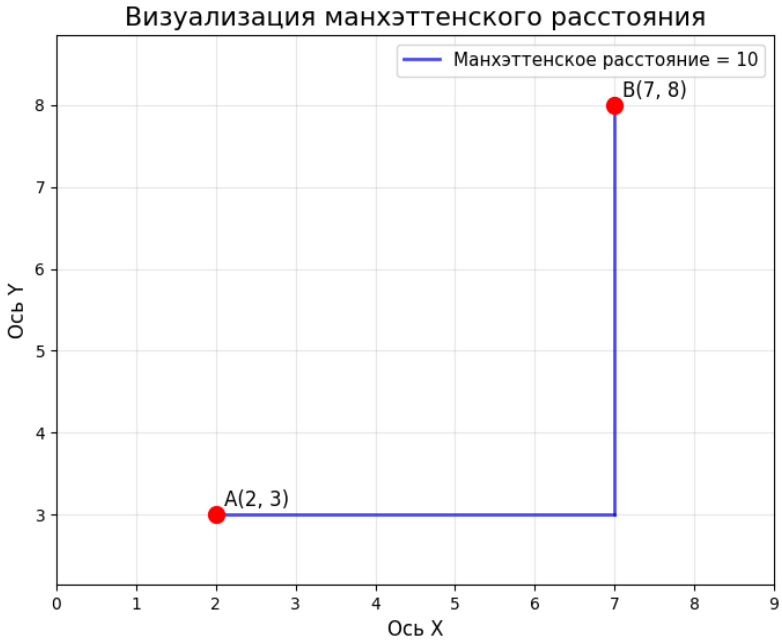

Рисунок 2. Манхэттенское расстояние

Недостатком Манхэттенского расстояния заключается в том, что атрибут с наибольшем значением может доминировать над итоговым значением расстояния.

Расстояние Чебышева (L_∞) – это максимальное абсолютное расстояние в одном измерении между двумя N-мерными точками. Формула для 2 точек записывается так:

$$dist(A,B) = \max(|x_A - x_B|, |y_A - y_B|) \quad (3)$$

Разница расстояния L1 и расстояния Чебышёва в том, что при переходе на одну клетку по диагонали в первом случае засчитывается два хода (например, вверх и влево), а во втором случае засчитывается всего один ход

Расстояние Минковского – определяет расстояние между двумя точками данных в нормированном векторном пространстве т.е. в N-мерном вещественном пространстве. Оно является общей формулой манхэттенского и евклидового расстояний.

$$dist(A,B) = (\sum_{i=1}^{N} |x_i - y_i|^p)^{\frac{1}{p}} \quad (4)$$

Эта формула:

- При $p = 1$ получаем формулу Манхэттенского расстояния.
- При $p = 2$ получаем формулу Евклидового расстояния.
- При $p = \infty$ получаем формулу расстояния Чебышева.

Таким образом, расстояние Минковского – это параметрическое семейство метрик, объединяющее *L1*, *L2* и другие *Lp*-расстояния.

Алгоритм кластеризации направлен на уменьшение внутрикластерного расстояния и увеличения межкластерного расстояния. Пример отличий межкластерного от внутрикластерного расстояния приведён на рисунке 3.

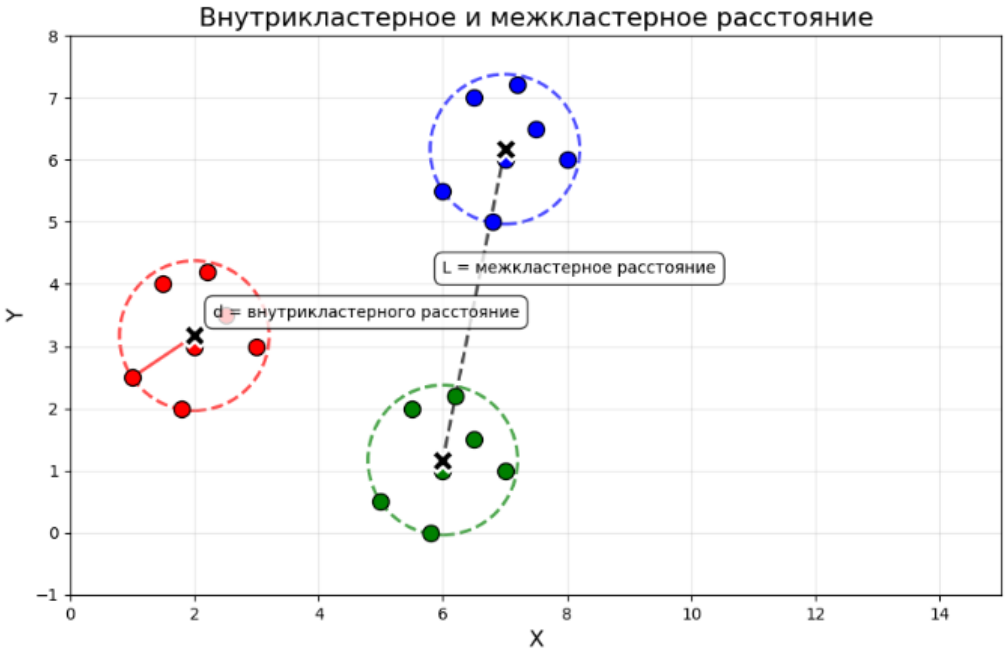

Рисунок 3. Внутрикластерное и межкластерное расстояние

Описание метода k-means

Алгоритм кластеризации k-means решает задачу распределения N наблюдений по K кластерам так, чтобы наблюдение принадлежало одному кластеру, который имеет наименьшее удаление от наблюдения [4].

К основным характеристикам алгоритма относятся:

• Число кластеров k – этот параметр алгоритма необходимо задать до начала его работы. Он определяет, на сколько групп будет разбито множество объектов.

• Центроид кластера – это точка в пространстве признаков, являющаяся «центром масс» кластера. Координаты центроида вычисляются как среднее значение координат всех точек, принадлежащих данному кластер

Метод k-means задаёт жёсткую кластеризацию, это означает, что каждая точка данных помещается только в один кластер. Целевой функцией алгоритма является:

$$min_{C_1,\dots,C_k,\mu_1,\dots,\mu_k} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (5)$$

Каждый кластер C_i содержит подмножество точек данных. Центроид кластера i равен среднему всех координат точек данных в кластере:

$$\mu_i = \sum_{x \in C_i} \frac{x}{n_i} \quad (6)$$

где n_i обозначает количество точек данных в кластере i [5].

Основными этапами работы алгоритма являются этапы распределения векторов по кластерам и пересчёт центроидов кластеров.

Распределение векторов по кластерам заключается в следующем: для каждого вектора $x_i \epsilon X, i = 1, \dots, n$ необходимо посчитать расстояние между этим вектором и центроидам кластера $\mu_j, j = 1, \dots, k$ так что:

$$C_i^{(t)} = min_{C_j^{(t)}} \left\| x - \mu_j^{(t)} \right\|^2, j = 1, \dots, k \quad (7)$$

В качестве метрики расстояния обычно используется евклидово расстояние, которое было записано ранее.

Пересчёт центроида μ_i кластера выполняется на каждом этапе для кластера C_i вычисляется как среднее арифметическое всех точек входящих в кластер.

В качестве базового примера можно создать простую программу на языке python. Создадим 10000 случайных данных и разобьём на 13 кластеров. Такая программа приведена на рисунке 4.

```
n_data = 10000
seed = 10
n_centers = 13

uniform = np.random.rand(n_data, 2)
clusters_uniform = KMeans(n_clusters=n_centers, random_state=seed).fit_predict(uniform)
figure = plt.figure(figsize=(20,10))

plt.subplot(121)
plt.scatter(uniform[:, 0], uniform[:, 1], edgecolors='k')
plt.title( label: "10000 случайно сгенерированных точек ", fontsize=16, weight='bold')
plt.axis('off')

plt.subplot(122)
plt.scatter(uniform[:, 0], uniform[:, 1], c=clusters_uniform, edgecolors='k', cmap='Spectral')
plt.title( label: "Кластеры выделенные k-means", fontsize=16, weight='bold')
plt.axis('off')
plt.show()
```

Рисунок 4. Пример программы создания кластеров на основе алгоритма k-means.

На рисунке 5 показан результат работы алгоритма с набором данных созданным случайным образом.

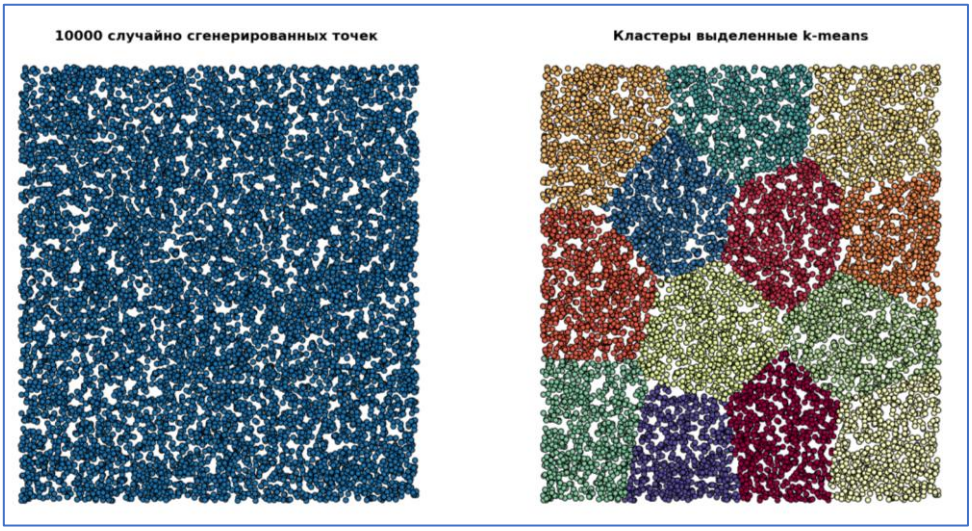

Рисунок 5. Пример работы алгоритма k-means.

Алгоритм k-means чувствителен к разбросу данных, потому что он опирается на евклидово расстояние. По этой формуле признаки с большими числами всегда перевешивают те, у которых числа маленькие.

Поэтому перед кластеризацией данные обычно нормируют. Самый распространенный способ – через z-преобразование (когда из значения вычитают среднее по выборке и результат делят на стандартное отклонение).

$$x' = \frac{x-\bar{x}}{\sigma_x} \quad (8)$$

Второй популярный способ – min-max нормализация, которая приводит все значения к интервалу от 0 до 1.

$$x' = \frac{x-x_{min}}{x_{max}-x_{min}} \quad (9)$$

Кластерные силуэты

Коэффициент силуэта (Silhouette coefficient) – это внутренняя метрика качества кластеризации, которая оценивает, насколько хорошо каждый объект соответствует своему кластеру по сравнению с другими кластерами. Метрика учитывает одновременно:

- Внутрикластерное расстояние (насколько близко объекты внутри кластера друг к другу)
- Межкластерное расстояние (насколько далеко кластеры находятся друг от друга)

Значение силуэта вычисляется для каждого объекта, а затем усредняется по всем объектам для получения общей оценки качества кластеризации:

$$s(i) = \frac{b(i)-a\,(i)}{\max\{a(i),b\,(i)\}} \quad (10)$$

где:

- $a\,(i)$ – среднее внутрикластерное расстояние;
- $b\,(i)$ среднее межкластерное расстояние;

Если рассмотреть предельные случаи, когда $a\,(i) = 0$ и $b\,(i)=0$, то $s\,(i)$ изменяется от -1 до 1.

Возможные значения коэффициента силуэта:

- $s\,(i) \approx 1$ объект хорошо кластеризован (далёк от других кластеров, близок к своему);
- $s\,(i) \approx 0$ объект находится на границе между кластерами;
- $s\,(i) \approx -1$ объект, вероятно, попал не в тот кластер;

Если кластер содержит единственный объект, то не ясно, как вычислить $a\,(i)$, и тогда установим $s\,(i) = 0$. Данный выбор является произвольным, но значение 0 оказывается «нейтральным».

Оценка для всей кластерной структуры достигается усреднением показателя по всем n объектам выборки или отдельного кластера:

$$S = \frac{1}{n}\sum_{i=1}^{n} s\,(i) \quad (11)$$

где: n – общее число объектов.

Кауфман и др. ввели термин «коэффициент силуэта» для обозначения максимального значения среднего $s\,(i)$ по всем данным всего набора данных:

$$SC = max_k \tilde{s}\,(k) \quad (12)$$

где: $\tilde{s}\,(k)$ представляет собой среднее значение $s\,(i)$) по всем данным всего набора данных для определенного количества кластеров k.

Коэффициент силуэта описывает наилучшую возможную кластеризацию для заданного числа кластеров, измеряемую наивысшим средним значением коэффициента силуэта для всех точек в наборе данных [6].

Индекс Данна

Индекс Данна – это одна из метрик для оценки алгоритмов кластеризации, определяемая как отношение минимального межкластерного расстояния к максимальному внутрикластерному расстоянию. Для заданного распределения кластеров более высокий индекс Данна указывает на лучшую кластеризацию. Одним из недостатков использования этого индекса является вычислительная стоимость по мере увеличения количества кластеров и размерности данных.

$$DI_m = \frac{min_{1\le i\le j\le m}\delta(C_i,C_j)}{max_{1\le k\le m}\Delta_k} \quad (13)$$

где:

- $\delta(C_i, C_j)$ – это межкластерное расстояние между кластерами C_i и C_j;
- Δ_k – это внутрикластерное расстояние;

Есть несколько способов определения внутрикластерного расстояния:

$\Delta_i = max_{x,y \epsilon\, C_i} d\,(x,y)$ вычисление максимального расстояния (была предложена Джозефом К. Данном)

$\Delta_i = \frac{2}{|C_i|(|C_i|-1)} \sum_{x,y \epsilon\, C_i, x \neq y} d\,(x,y)$ вычисление среднего расстояния между всеми парами точек внутри кластера.

$\Delta_i = \frac{\sum_{x \epsilon\, C_i} d\,(x,\mu)}{|C_i|}, \mu = \frac{\sum_{x \epsilon\, C_i} x}{|C_i|}$ вычисление среднего расстояния всех точек до центроида кластера.

Индекс Данна зависит от количества кластеров в множестве m. Если количество кластеров заранее не известно выбирается *m* при котором *DI* максимальный [7].

Кластеризация ирисов

Для исследования был использован классический набор данных Ирисы Фишера, содержащий 150 образцов ирисов трёх видов Iris setosa, Iris versicolor и Iris virginica. Каждый образец описывается четырьмя признаками:

- Длина чашелистика (sepal length);
- Ширина чашелистика (sepal width);
- Длина лепестка (petal length);
- Ширина лепестка (petal width);

В соответствии с задачей кластеризации столбец с видами не использовался.

Алгоритм k-means чувствителен к масштабу признаков, поэтому все данные были стандартизированы с помощью z-преобразования:

$$x' = \frac{x - \bar{x}}{\sigma_x} \quad (14)$$

Это позволило привести все признаки к одинаковому масштабу (среднее 0, стандартное отклонение 1) и устранить доминирование признаков с большими числовыми значениями. На рисунках с 6 – 8 представлен код программы для кластеризации и оценки качества.

```
iris = load_iris()
X = iris.data
X_scaled = StandardScaler().fit_transform(X)
y_true = iris.target
target_names = iris.target_names

k_values = list(range(2, 6))
results = []
kmeans_models = {}

for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels = kmeans.fit_predict(X_scaled)

    silhouette_avg = silhouette_score(X_scaled, labels)
    dunn_val = dunn_index(X_scaled, labels)

    results.append({
        'k': k,
        'Silhouette': silhouette_avg,
        'Dunn_Index': dunn_val
    })

    kmeans_models[k] = {'model': kmeans, 'labels': labels, 'silhouette': silhouette_avg}

results_df = pd.DataFrame(results)
print("\nРезультаты оценки качества кластеризации:")
print(results_df.to_string(index=False))

results_df.to_csv( path_or_buf: 'output/clustering_results.csv', index=False)
```

Рисунок 6. Оценка качества кластеризации датасета Iris методом k-means

```python
def dunn_index(X, labels):  1 usage
    k = len(np.unique(labels))
    clusters = [X[labels == i] for i in range(k)]

    deltas = []
    for cluster in clusters:
        if len(cluster) > 1:
            distances = cdist(cluster, cluster)
            n = len(cluster)
            upper_tri_indices = np.triu_indices(n, k=1)
            mean_dist = np.mean(distances[upper_tri_indices])
            deltas.append(mean_dist)
        else:
            deltas.append(0)

    max_delta = max(deltas) if max(deltas) > 0 else 1e-9

    centroids = np.array([cluster.mean(axis=0) for cluster in clusters])
    pairwise_distances = cdist(centroids, centroids)
    np.fill_diagonal(pairwise_distances, np.inf)
    min_delta = np.min(pairwise_distances)

    if min_delta == np.inf or max_delta == 0:
        return 0

    return min_delta / max_delta
```

Рисунок 7. Реализация индекса Данна

Функция dunn_index вычисляет индекс Данна – метрику качества кластеризации, которая оценивает компактность и разделимость кластеров.

```python
optimal_k_silhouette = results_df.loc[results_df['Silhouette'].idxmax(), 'k']
optimal_k_dunn = results_df.loc[results_df['Dunn_Index'].idxmax(), 'k']
optimal_k_silhouette = int(optimal_k_silhouette)
optimal_k_dunn = int(optimal_k_dunn)

print("ИТОГ:")
print(f"Оптимальное k по силуэту: {optimal_k_silhouette} "
      f"(силуэт = {results_df.loc[results_df['k'] == optimal_k_silhouette, 'Silhouette'].values[0]:.4f})")
print(f"Оптимальное k по индексу Данна: {optimal_k_dunn} "
      f"(индекс Данна = {results_df.loc[results_df['k'] == optimal_k_dunn, 'Dunn_Index'].values[0]:.4f})")
print(f"Истинное число кластеров (виды ирисов): 3")
```

Рисунок 8. Определение оптимального k и вывод в консоль

Для каждого значения числа кластеров $k = 2, 3, 4, 5$ были выполнены следующие шаги:

1. Кластеризация методом k-means с использованием евклидовой метрики. Для повышения устойчивости результатов задано n_init=10 (алгоритм запускается 10 раз с разными случайными начальными центроидами, выбирается лучшее разбиение);
2. Оценка качества с помощью двух внутренних метрик:

- Средний коэффициент силуэта $\tilde{s}$
- Индекс Данна DI_m

3. Визуализация результатов:

- Графики силуэтов для каждого k

- Проекция кластеризованных данных на первые две главные компоненты (PCA) для наглядного отображения структуры кластеров.

Численные значения метрик качества представлены в таблице 1.

Таблица 1.

Оценка качества кластеризации при различных *k*

k	Средний коэффициент силуэта	Индекс Данна
2	0.581750	2.006519
3	0.459948	1.387339
4	0.386941	1.101365
5	0.345901	1.147654

На рисунке 9 представлены графики зависимостей метрик от числа кластеров, а на рисунках 10 и 11 – проекции кластеризованных данных на первые две главные компоненты. На рисунках 12 – 13 показаны диаграммы силуэтов для каждого *k*.

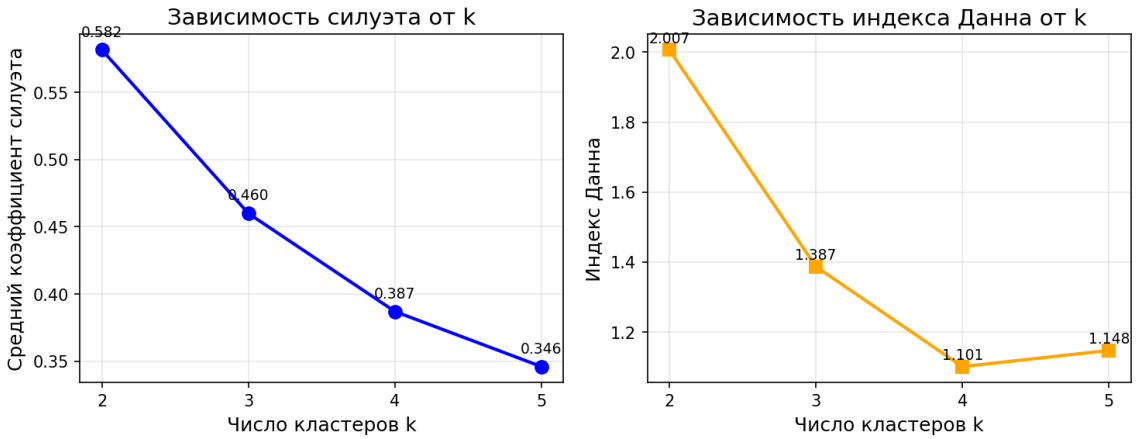

Рисунок 9. Зависимость среднего коэффициента силуэта и индекса Данна от числа кластеров

По обеим метрикам оптимальным является *k*=2, что расходится с истинным числом кластеров в датасете Iris (3 вида).

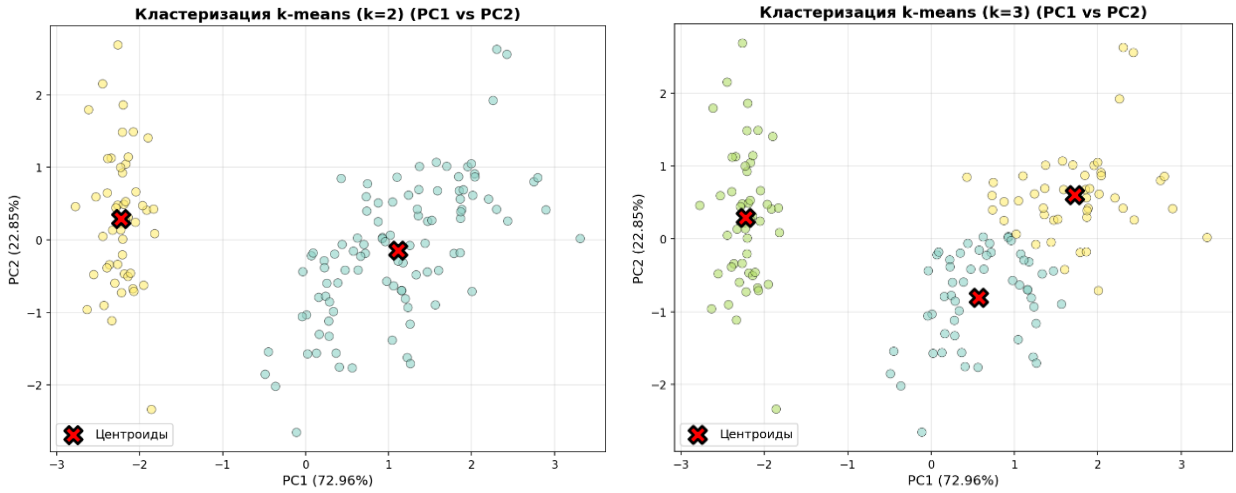

Рисунок 10. Визуализация кластеризации для *k* = 2, 3

На двух графиках показана кластеризация данных ирисов методом k-means с k=2 и k=3 после проецирования на главные компоненты (PC1 – 73%, PC2 – 23%). При k=2 точки делятся на два кластера: компактный слева и размытый справа. При k=3 правая группа разделяется на два отдельных кластера, что лучше соответствует естественной структуре данных. Красными крестами отмечены центроиды.

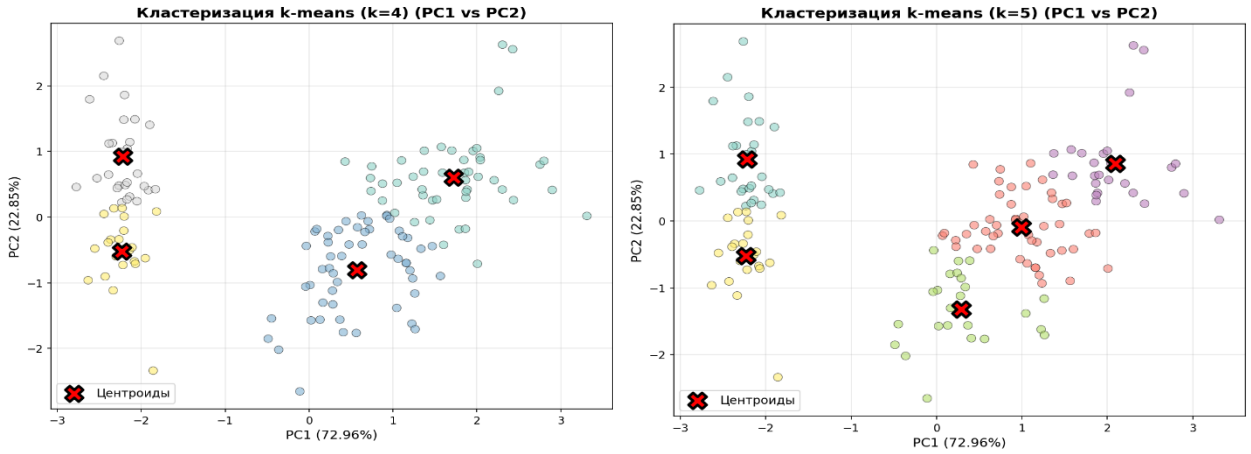

Рисунок 11. Визуализация кластеризации для k = 4, 5

При k=4 и k=5 алгоритм начинает выделять внутри плотных скоплений кластеры. Качество кластеризации ухудшается, появляются искусственные группы, которых нет в природе.

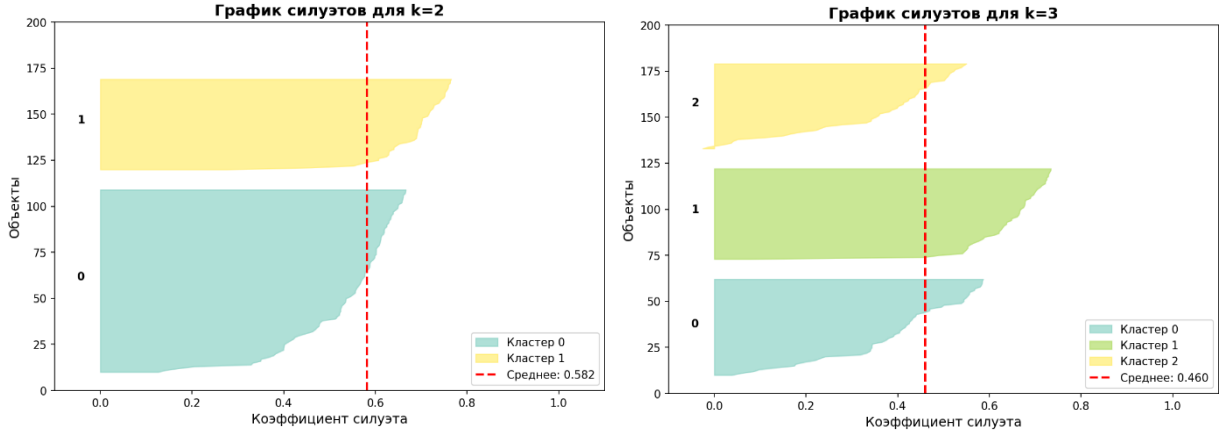

Рисунок 12. Графики силуэтов для k = 2, 3.

На первых двух графиках силуэтов для k=2 и k=3 показано, насколько каждый объект соответствует своему кластеру. При k=2 все кластеры имеют положительные коэффициенты силуэта, большинство значений выше 0.5, а среднее значение составляет 0.582 – это хороший результат. При k=3 средний силуэт снижается до 0.460, появляются объекты с низкими и даже нулевыми значениями, что указывает на некоторое перекрытие между кластерами, однако структура в целом сохраняется. Толщина полос отражает размер каждого кластера.

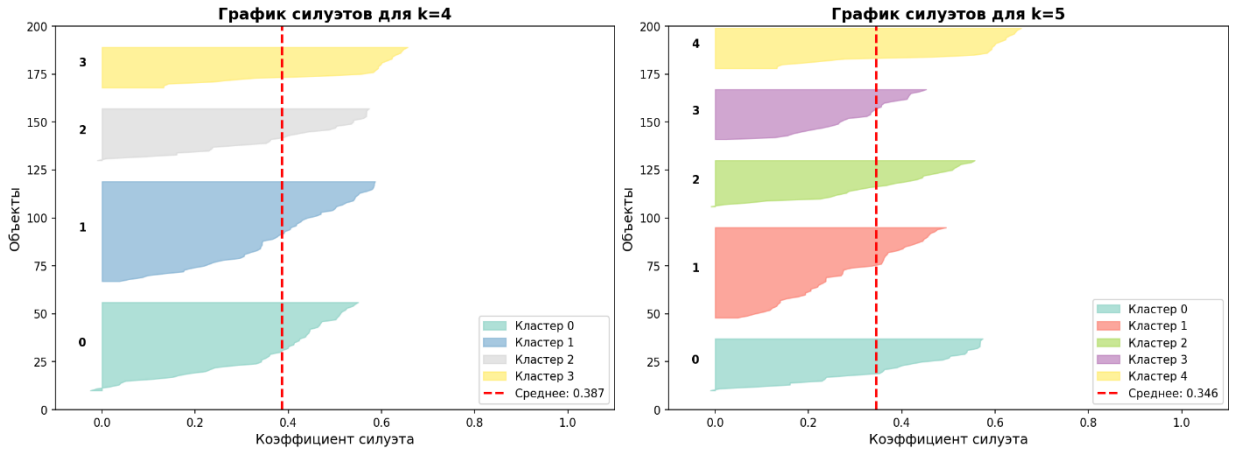

Рисунок 13. Графики силуэтов для k = 2, 3.

Выбор оптимального числа кластеров

По среднему коэффициенту силуэта максимальное значение достигается при k=2 ($\tilde{s}$=0.5818). Это указывает на то, что с точки зрения этого критерия наилучшим является разбиение на два кластера.

По индексу Данна максимальное значение индекса Данна также наблюдается при k=2 (DI=2.0065). При k=3 индекс снижается до 1.3873, а затем остается на уровне около 1.1.

Известно, что в датасете представлены три биологических вида ирисов. Однако ни одна из использованных метрик не указала на k=3 как на оптимальное число кластеров. Обе метрики отдали предпочтение k=2.

Анализ результатов оценки качества кластеризации набора данных Iris методом k-means выявил, что как средний коэффициент силуэта, так и индекс Данна указывают на оптимальность разбиения на два кластера, несмотря на наличие трёх биологических видов. Данное расхождение объясняется следующими факторами.

Вид Iris setosa является линейно разделимым и образует компактный, изолированный кластер в пространстве признаков. В то же время виды Iris versicolor и Iris virginica демонстрируют значительное перекрытие в области длины и ширины лепестков, формируя единое непрерывное облако точек. С геометрической точки зрения такая конфигурация делает естественным разбиение на два кластера: один соответствует setosa, второй – объединённому множеству versicolor и virginica.

Индекс Данна определяется как отношение минимального межкластерного расстояния к максимальному внутрикластерному диаметру и, следовательно, характеризует наихудший случай разделимости. При k=3 кластер, включающий пересекающиеся популяции versicolor и virginica, обладает увеличенным внутрикластерным разбросом, что приводит к росту знаменателя и снижению значения индекса.

Коэффициент силуэта представляет собой усреднённую оценку компактности и разделимости на уровне отдельных объектов. Наличие переходной зоны между versicolor и virginica уменьшает среднее значение силуэта при увеличении числа кластеров с 2 до 3, поскольку значительная часть точек оказывается вблизи границ кластеров.

Метод k-means предполагает, что кластеры имеют сферическую форму и сравнимый размер. В действительности Iris setosa образует компактную изолированную группу, тогда как versicolor и virginica обладают вытянутой, несферической структурой с частичным перекрытием. Такая конфигурация выходит за рамки предположений алгоритма, что приводит к некорректному выделению трёх биологических классов при k=3: один из видов оказывается искусственно разделённым на две части.

Несмотря на то, что обе внутренние метрики согласованно указывают на k=2, содержательная интерпретация требует учёта биологической структуры данных. При k=3, несмотря на более низкие значения метрик, достигается соответствие трём истинным видам, что подтверждается визуальным анализом PCA-проекции и распределением центроидов. В задачах, где априорная информация о числе кластеров доступна, предпочтительнее выбирать k=3 на основе содержательного обоснования, рассматривая значения метрик как показатель степени соответствия данным модельных предположений, а не как абсолютный критерий оптимальности.

Выводы

В результате выполнения работы был проведён сравнительный анализ двух внутренних метрик оценки качества кластеризации – коэффициента силуэта и индекса Данна – на примере алгоритма k-means и классического датасета ирисов Фишера.

В теоретической части работы показано, что кластеризация является задачей обучения без учителя, направленной на группировку объектов по принципу внутрикластерного сходства. Описан алгоритм k-means, его ключевые особенности, включая чувствительность к масштабу признаков и предположение о сферической форме кластеров. Рассмотрены коэффициент силуэта, оценивающий качество на уровне отдельных объектов и всей выборки, и индекс Данна, характеризующий отношение минимального межкластерного расстояния к максимальному внутрикластерному разбросу.

В экспериментальной части выполнена кластеризация данных ирисов при различных значениях числа кластеров с предварительной стандартизацией признаков. Результаты показали, что обе метрики дают согласованную оценку, указывая на оптимальность двух кластеров, что расходится с истинным числом биологических видов (три). Это расхождение объясняется структурой данных: вид Iris setosa хорошо отделён от двух других, тогда как Iris versicolor и Iris virginica значительно перекрываются в пространстве признаков. Кроме того, метод k-means предполагает сферическую форму кластеров, что не соответствует реальному распределению данных.

Таким образом, коэффициент силуэта и индекс Данна являются полезными инструментами внутренней оценки качества кластеризации, однако их рекомендации следует интерпретировать с учётом структуры данных и ограничений выбранного алгоритма. На практике для принятия обоснованного решения о числе кластеров целесообразно использовать совокупность нескольких метрик и визуальный анализ.

Список литературы:

1. Bishop C. M., Nasrabadi N. M. Pattern Recognition and Machine Learning. – New York: Springer, 2006. – Vol. 4, no. 4. – P. 738. – URL: https://arxiv.org/pdf/2401.07389 (дата обращения: 02.03.2026).

2. Wolfram Research. Clustering // Introduction to Machine Learning. – URL: https://www.wolfram.com/language/introduction-machine-learning/clustering/ (дата обращения: 12.03.2026).

3. FAISS: Facebook AI Similarity Search. – URL: https://metadesignsolutions.com/faiss-facebook-ai-similarity-search/ (дата обращения: 19.03.2026).

4. Алгоритм k-means // Algowiki-project. – URL: https://algowiki-project.org/w/ru/index.php?title=%D0%A3%D1%87%D0%B0%D1%81%D1%82%D0%BD%D0%B8%D0%BA:IanaV/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_k_means&oldid=6020 (дата обращения: 19.03.2026).

5. Чжен Э., Казари А. Машинное обучение: конструирование признаков: принципы и техники для аналитиков. – М.: Эксмо, 2022. – 145 с.

6. Kaufman L., Rousseeuw P. J. Finding Groups in Data: An Introduction to Cluster Analysis. – Hoboken, NJ: Wiley-Interscience, 1990. – P. 87.

7. Dunn index // Wikipedia. – URL: https://en.wikipedia.org/wiki/Dunn_index (дата обращения: 10.04.2026).