Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине «Прикладные интеллектуальные системы и экспертные системы»

Кластеризация данных

Студент Мамедов Р. В.

Группа М-ИАП-23-1

Руководитель Кургасов В.В.

доцент, канд. пед. наук

Липецк 2023 г.

Цель работы

Получить практические навыки решения задачи кластеризации фактографических данных в среде Jupiter Notebook. Научиться настраивать параметры методов и оценивать точность полученного разбиения.

Задание кафедры

1) Загрузить выборки согласно варианту задания.

2) Отобразить данные на графике в пространстве признаков. Поскольку решается задача кластеризации, то подразумевается, что априорная информация о принадлежности каждого объекта истинному классу неизвестна, соответственно, на данном этапе все объекты на графике должны отображаться одним цветом, без привязки к классу.

3) Провести иерархическую кластеризацию выборки, используя разные способы вычисления расстояния между кластерами: расстояние ближайшего соседа (single), дальнего соседа (complete), Уорда (Ward). Построить дендрограммы для каждого способа. Размер графика должен быть подобран таким образом, чтобы дендрограмма хорошо читалась.

4) Исходя из дендрограмм выбрать лучший способ вычисления расстояния между кластерами.

5) Для выбранного способа, исходя из дендрограммы, определить количество кластеров в имеющейся выборке. Отобразить разбиение на кластеры и центроиды на графике в пространстве признаков (объекты одного кластера должны отображаться одним и тем же цветом, центроиды всех кластеров – также одним цветом, отличным от цвета кластеров).

6) Рассчитать среднюю сумму квадратов расстояний до центроида, среднюю сумму межкластерных расстояний для данного разбиения. Сделать вывод о качестве разбиения.

7) Провести кластеризацию выборки методом k-средних. для k [1, 10].

8) Сформировать три графика: зависимость средней суммы квадратов расстояний до центроида, средней суммы средних внутрикластерных расстояний и средней суммы межкластерных расстояний от количества кластеров. Исходя из результатов, выбрать оптимальное количество кластеров.

9) Составить сравнительную таблицу результатов разбиения иерархическим методом и методом k-средних.

Вариант №12

Вариант	**12**
Вид классов	classificati on
Random_state	27
class_sep	1

Ход работы

Загрузим выборки согласно варианту:

```
X, y = make_classification(n_samples=100,
                           n_features=2,
                           n_redundant=0,
                           n_informative=2,
                           n_clusters_per_class=1,
                           n_classes=4,
                           random_state=27,
                           class_sep=1)
```

Отобразим на графике сгенерированные данные, для этого воспользуемся библиотекой matplotlib.pyplot

```
plt.scatter(X[:, 0], X[:, 1])
```

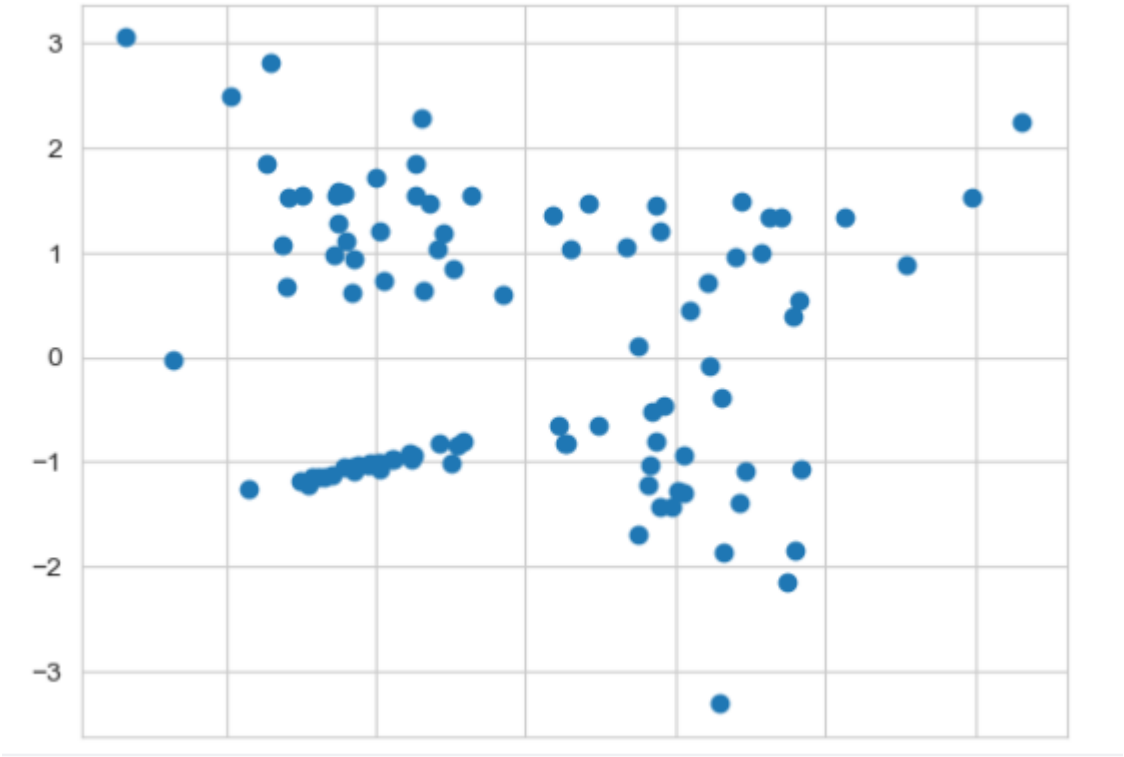

Рисунок 1 – Визуализация полученных данных

Воспользуемся иерархической кластеризацией выборки с использованием различных методов вычисления расстояния.

Метод вычисления расстояния ближайшего соседа (single). Полученная дендограмма представлена на рисунке 2

```
from scipy.cluster.hierarchy import linkage

clusters_single = linkage(X, method='single')
clusters_single
```

```
from scipy.cluster.hierarchy import dendrogram

dendrogram(clusters_single)
plt.show()
```

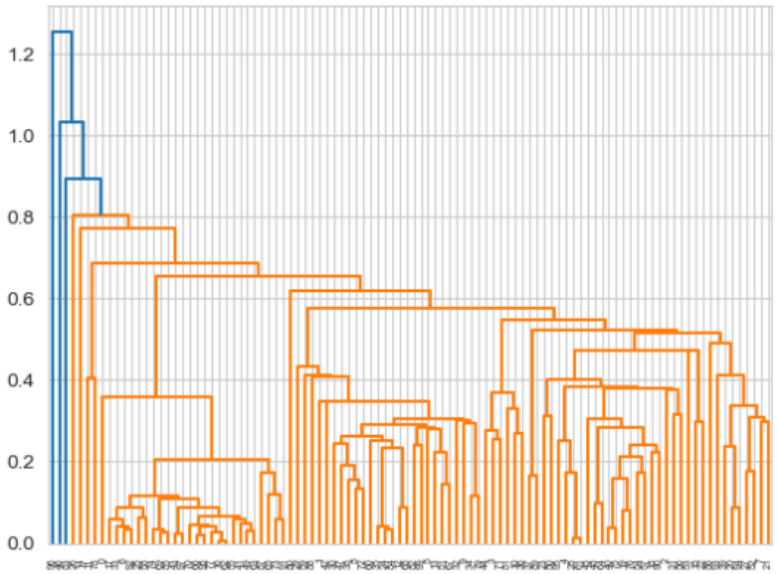

Рисунок 2 – Дендограмма, полученная при помощи метода ближайшего соседа

Метод вычисления расстояния дальнего соседа (complete). Полученная дендограмма представлена на рисунке 3.

```
 from scipy.cluster.hierarchy import linkage

clusters_complete = linkage(X, method='complete')
clusters_complete

from scipy.cluster.hierarchy import dendrogram

dendrogram(clusters_complete)
plt.show()
```

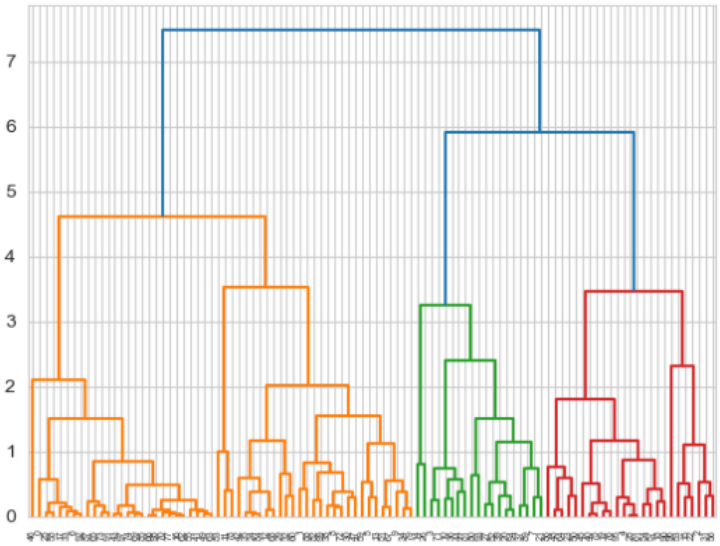

Рисунок 3 - Дендограмма для расстояния дальнего соседа (complete)

Метод вычисления расстояния Уорда (ward). Полученная дендограмма представлена на рисунке 4.

```
plt.show()
#%%
from scipy.cluster.hierarchy import linkage

clusters_ward = linkage(X, method='ward')
clusters_ward
from scipy.cluster.hierarchy import dendrogram

dendrogram(clusters_ward)
plt.show()
```

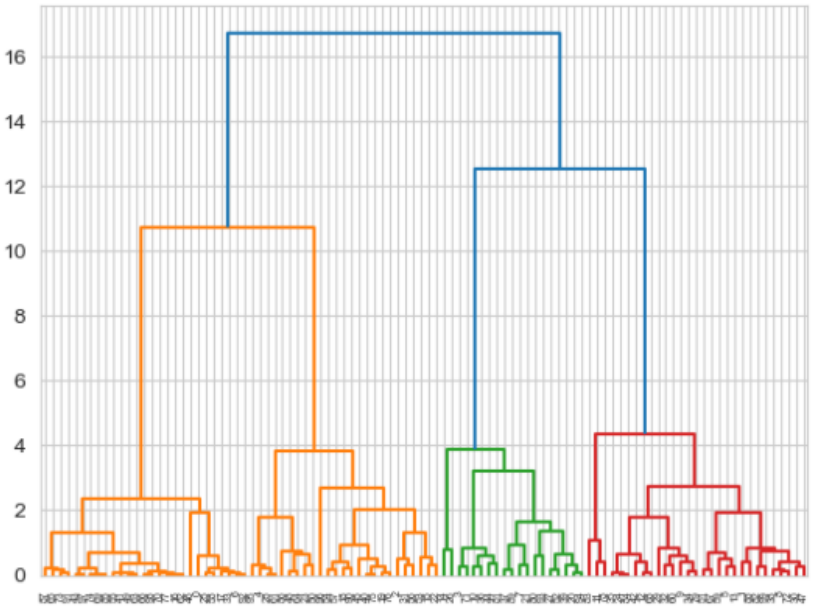

Рисунок 4 – Дендограмма для расстояния Уорда (ward)

Лучшим способом вычисления расстояния между кластерами является метод дальнего соседа (complete). Определим количество кластеров в имеющейся выборке с использованием данного способа и отобразим разбиение на кластеры и центроиды на графике в пространстве признаков. Полученное разбиение представлено на рисунке 5.

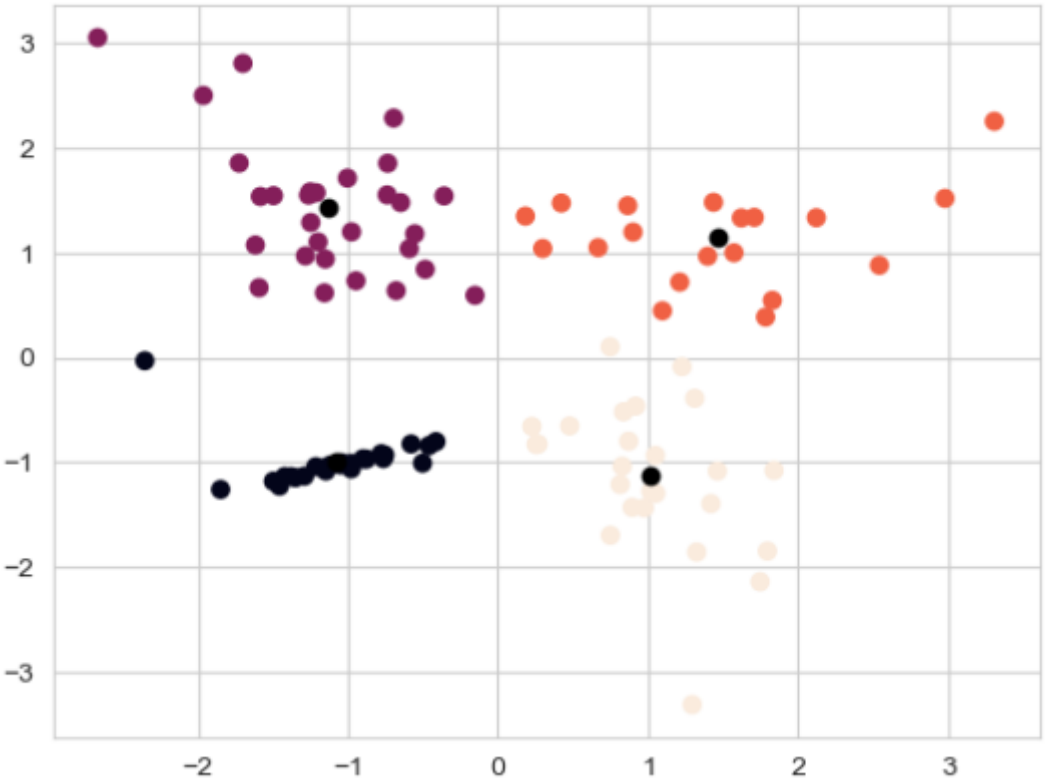

Рисунок 5 – График разбиения данных на кластеры

Рассчитаем среднюю сумму квадратов расстояний до центроида, среднюю сумму средних внутрикластерных расстояний и среднюю сумму межкластерных расстояний для данного разбиения.

Рассчитанное значение суммы квадратов расстояний до центроида и код для этого представлен на рисунке 6. Рассчитанное значение средних внутрикластерных расстояний и код для этого представлен на рисунке 7. Рассчитанное значение суммы межкластерных расстояний и код для этого представлен на рисунке 8.

```python
def get_inertia(X, clusters):
    sum_sq_dist = np.zeros(4)
    for i in range(1, 5):
        ix = np.where(T == i)
        sum_sq_dist[i - 1] = np.sum(euclidean_distances(*X[ix, :], [clusters[i - 1]]) ** 2)
    return np.sum(sum_sq_dist) / 4


get_inertia(X, clusters)
```

Executed at 2023.12.17 14:12:04 in 5ms

```
14.84018215047446
```

Рисунок 6 – Сумма квадратов расстояний до центроида

```python
def get_avg_incluster_distance(X, clusters):
    sum_avg_intercluster_dist = np.zeros(4)
    for i in range(1, 5):
        ix = np.where(T == i)
        sum_avg_intercluster_dist[i - 1] = np.sum(euclidean_distances(*X[ix, :],
         [clusters[i - 1]]) ** 2) / len(
            *X[ix, :])
    return np.sum(sum_avg_intercluster_dist) / 4


get_avg_incluster_distance(X, clusters)
```

Executed at 2023.12.17 14:12:04 in 5ms

```
0.6152937255671042
```

Рисунок 7 – Сумма средних внутрикластерных расстояний

```python
from sklearn.metrics.pairwise import euclidean_distances


def get_sum_incluster_distance(clusters):
    return np.sum(euclidean_distances(clusters, clusters))


get_sum_incluster_distance(clusters)
```

Executed at 2023.12.17 14:12:04 in 5ms

```
32.210648260196095
```

Рисунок 8 – Сумма межкластерных расстояний

Проведем кластеризацию выборки методом k-средних для k = [1, 10], а после построим три графика: зависимость средней суммы квадратов расстояний до центроида, средней суммы средних внутрикластерных

расстояний и средней суммы межкластерных расстояний от количества кластеров.

Код для расчета средней суммы квадратов расстояний до центроида представлен на рисунке 9, а построенный график на рисунке 10

```
from sklearn.cluster import KMeans

models = []
predicted_values = []

for k in range(1, 11):
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(X)
    models.append(kmeans)
    predicted_values.append(kmeans.predict(X))

sum_sq_dist_avg = []
for i, model in enumerate(models):
    sum_sq_dist_avg.append(model.inertia_ / (i + 1))

sum_sq_dist_avg
```

Executed at 2023.12.17 14:12:06 in 1s 612ms

```
C:\Users\Ruslan\anaconda3\envs\expert-systems\Lib\site

 [335.21064002846987,
  97.57214659193164,
  37.199023822789215,
  14.807226279540776,
  9.855232146203372,
  6.9809320711184215,
  4.840303207787193,
  3.6231104141131762,
  2.8513749174143985,
  2.242383732236761]
```

Рисунок 9 – Код для вычисления средней суммы квадратов расстояний до центроида

```
plt.plot(range(1, 11), sum_sq_dist_avg, '-o')
```
Executed at 2023.12.17 14:12:06 in 117ms

```
[<matplotlib.lines.Line2D at 0x2058de665d0>]
```

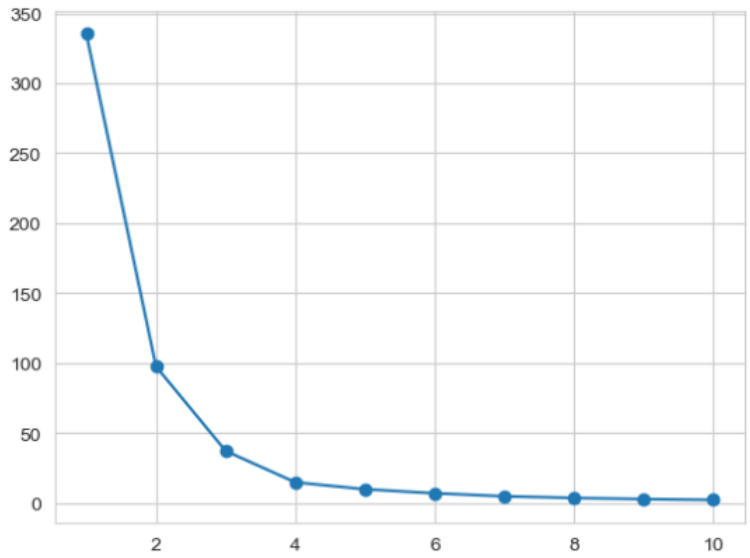

Рисунок 10 – График средней суммы квадратов расстояний до центроида

Код для расчета средней суммы средних внутрикластерных расстояний представлен на рисунке 11, а построенный график на рисунке 12.

```
new_centers = [model.cluster_centers_ for model in models]

sum_avg_intercluster_dist_avg = []
for k, kmean in enumerate(models):
    intercluster_sum = np.zeros(4)
    for i in range(4):
        ix = np.where(predicted_values[k] == i)
        if len(ix[0]) == 0:
            intercluster_sum[i - 1] = 0
        else:
            intercluster_sum[i - 1] = np.sum(
                euclidean_distances(*X[ix, :], [kmean.cluster_centers_[i - 1]]) ** 2) /
                 len(*X[ix, :])
    sum_avg_intercluster_dist_avg.append(np.sum(intercluster_sum) / (k + 1))
sum_avg_intercluster_dist_avg
```
Executed at 2023.12.17 14:12:06 in 5ms

```
[3.352106400284699,
 7.5798631038924285,
 8.061842054662742,
 9.035282937968375,
 8.073344481148713,
 4.726147695471854,
 5.678731558878901,
 3.9580210999303747,
 2.4660503149889195,
 4.498283713639709]
```

Рисунок 11 – Код для вычисления средней суммы средних внутрикластерных расстояний

```
plt.plot(range(1, 11), sum_avg_intercluster_dist_avg, '-o')
```
Executed at 2023.12.17 14:12:06 in 115ms

```
[<matplotlib.lines.Line2D at 0x2058dbed010>]
```

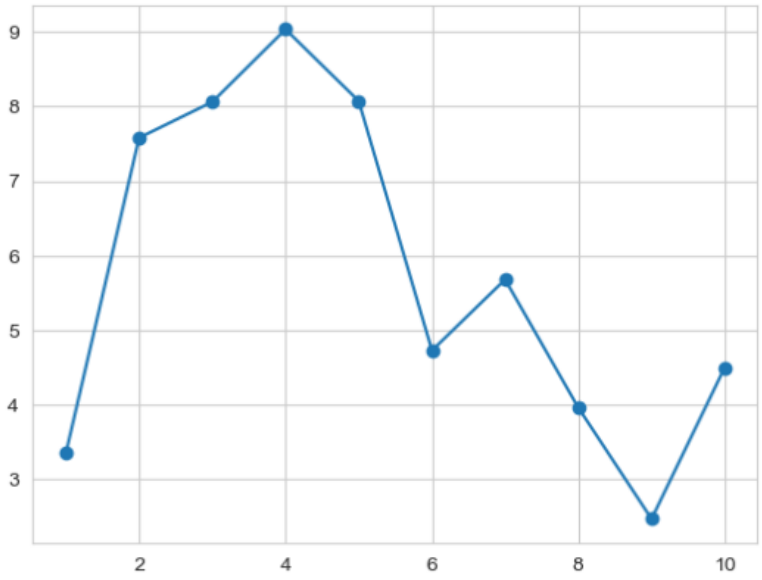

Рисунок 12 – График средней суммы средних внутрикластерных расстояний

Код для расчета средней суммы межкластерных расстояний от количества кластеров представлен на рисунке 13, а построенный график на рисунке 14.

```
sum_intercluster_dist_avg = []

for i, kmean in enumerate(models):
    value = np.sum(euclidean_distances(kmean.cluster_centers_, kmean.cluster_centers_))
    sum_intercluster_dist_avg.append(value / (i + 1))
sum_intercluster_dist_avg
```
Executed at 2023.12.17 14:12:06 in 4ms

```
[0.0,
 2.3688884494128217,
 5.295300320326329,
 8.076711981403047,
 11.866782849695927,
 15.592388909072584,
 16.506326224518535,
 21.118009291494424,
 22.89787619731213,
 26.075629954460076]
```

Рисунок 13 – Код для вычисления средней суммы межкластерных расстояний от количества кластеров

```
plt.plot(range(1, 11), sum_intercluster_dist_avg, '-o')
```
Executed at 2023.12.17 14:12:06 in 116ms

```
[<matplotlib.lines.Line2D at 0x2058ddb7550>]
```

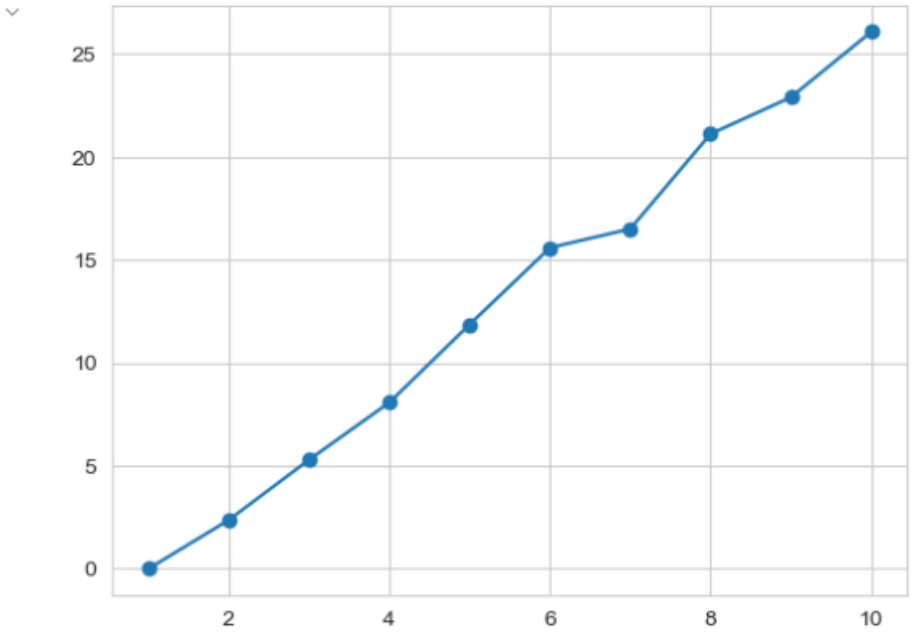

Рисунок 14 – График средней суммы межкластерных расстояний от количества кластеров

Составим сравнительную таблицу для ранее описанных метрик качества моделей для иерархического метода и метода k-средних. Составленная таблица представлена на рисунке 15

	Иерархический метод			Метод k-средних		
	Сумма квадратов расстояний до центроида	Сумма средних внутрикластерных расстояний	Сумма межкластерных расстояний	Сумма квадратов расстоян	Сумма средних внутрикластерны	Сумма межкластерных расстоян
0	14.84018215	0.615293726	32.21064826	335.2106	3.352106	0
1	14.84018215	0.615293726	32.21064826	97.57215	7.579863	2.368888
2	14.84018215	0.615293726	32.21064826	37.19902	8.061842	5.2953
3	14.84018215	0.615293726	32.21064826	14.80723	9.035283	8.076712
4	14.84018215	0.615293726	32.21064826	9.855232	8.073344	11.86678
5	14.84018215	0.615293726	32.21064826	6.980932	4.726148	15.59239
6	14.84018215	0.615293726	32.21064826	4.840303	5.678732	16.50633
7	14.84018215	0.615293726	32.21064826	3.62311	3.958021	21.11801
8	14.84018215	0.615293726	32.21064826	2.851375	2.46605	22.89788
9	14.84018215	0.615293726	32.21064826	2.242384	4.498284	26.07563

Рисунок 15 – Сводная таблица с метриками качества модели

Вывод

В ходе выполнения данной лабораторной работы мною были получены навыки кластеризации данных.

В рамках данной работы были применены различные методы кластеризации: иерархический метод и метод k-средних.

В ходе анализа метрик было определено, что оптимальное значение кластеров равняется четырем.

Также была составлена таблица сравнения метрик иерархическим методом кластеризации и методом k-средних.