Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине «Прикладные интеллектуальные системы и экспертные системы»

«Кластеризация данных»

Студент Косенков В.Д.

Группы М-ИАП-23

Руководитель Кургасов В.В.

Доцент

Липецк 2023 г

Цель работы

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

Задание кафедры

Задание:

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 средних внутрикластерных расстояний и 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.

Ход работы

Вариант 9

n_features=2, n_redundant=0, n_informative=2, n_clusters_per_class=1, n_classes = 4

9
classification
3
2

Рисунок 1 - Вариант для выполнения

Генерация данных для варианта представлена на рисунке 2.

```
from sklearn.datasets import make_classification
```

```
# Генерация выборки с использованием make_classification
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=3,
                           class_sep=2)
```

Рисунок 2 - Генерация данных

Отображение выборки на графике представлено на рисунке 3.

```
import matplotlib.pyplot as plt
```

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

```
<matplotlib.collections.PathCollection at 0x7d899e6896f0>
```

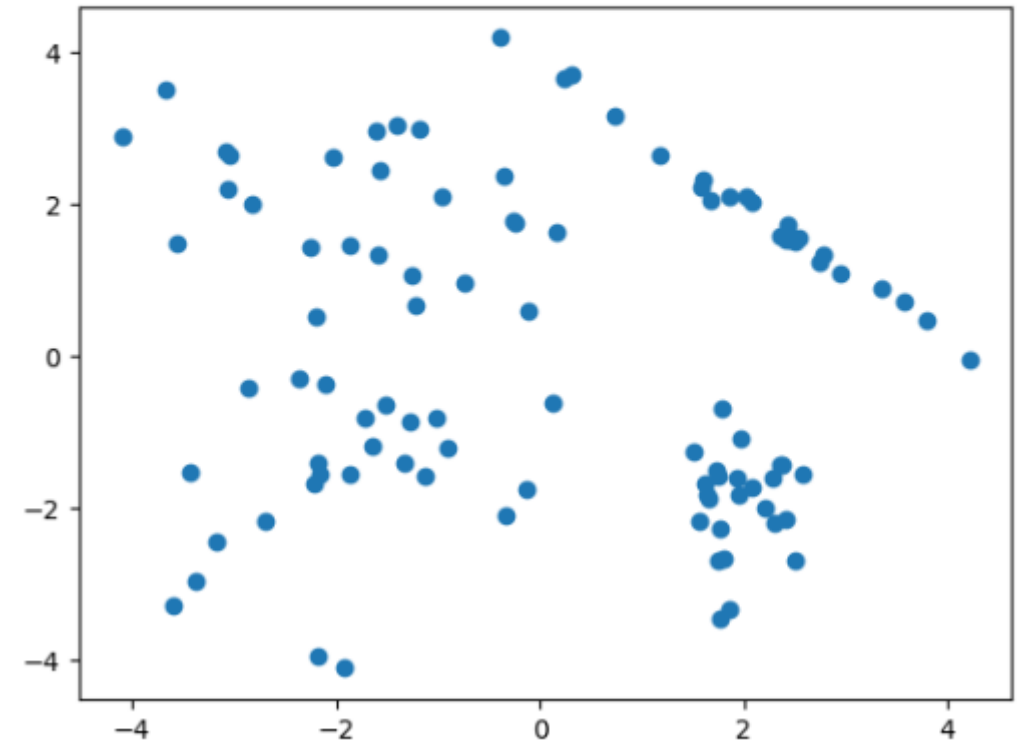

Рисунок 3 - Отображение выборки

Графики иерархической кластеризации представлены на рисунке 4.

```
mergings_single = linkage(X, method='single')
mergings_complete = linkage(X, method='complete')
mergings_ward = linkage(X, method='ward')
```

```
# Расстояние ближайшего соседа (single)
fig, axes = plt.subplots(1, 3, figsize=(15, 5))
dendrogram(mergings_single, ax=axes[0])
axes[0].set_title('Расстояние ближайшего соседа')

# Расстояние дальнего соседа (complete)
dendrogram(mergings_complete, ax=axes[1])
axes[1].set_title('Расстояние дальнего соседа')

# Расстояние Уорда (Ward)
dendrogram(mergings_ward, ax=axes[2])
axes[2].set_title('Расстояние Уорда')
```

```
Text(0.5, 1.0, 'Расстояние Уорда')
```

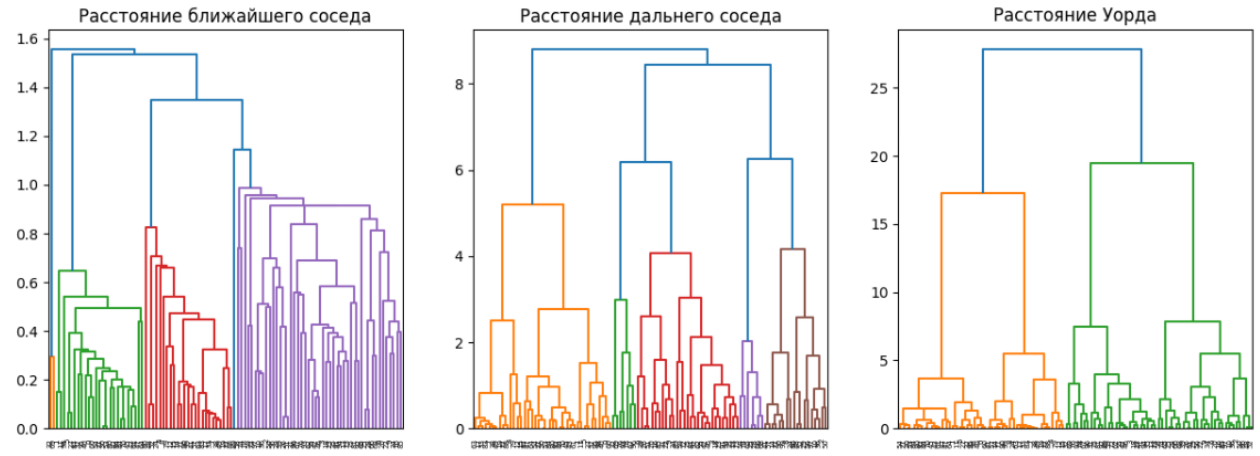

Рисунок 4 – Дендограммы

Выбор лучшего разбиения

```
mergings_ward = linkage(X, method='ward')
```

```
dendrogram(mergings_ward)
plt.show()
```

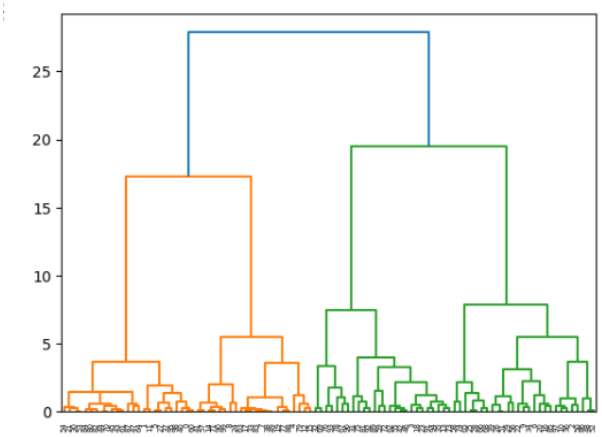

Рисунок 5 – Выбор лучшего разбиения

Лучшим способом вычисления расстояния между кластерами является расстояние Уорда (ward). 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 6.

```
plt.scatter(X[:, 0], X[:, 1], c=T)
plt.scatter(clusters[:, 0], clusters[:, 1], c='black')
```

```
<matplotlib.collections.PathCollection at 0x7d89ce363490>
```

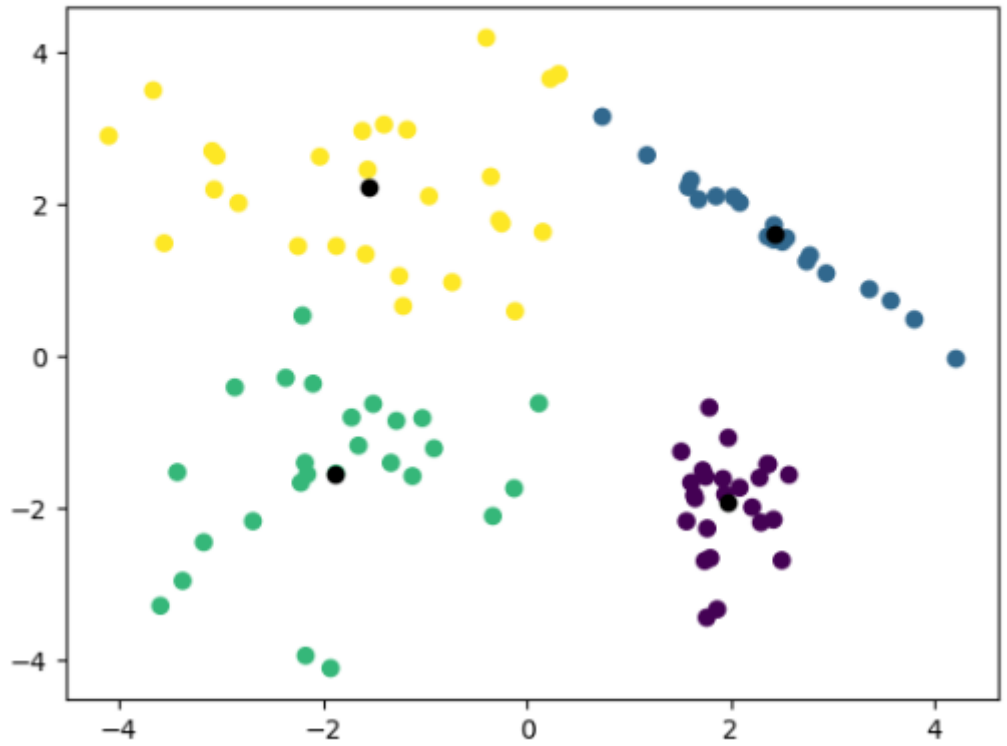

Рисунок 6 - График разбиения данных на кластеры

Рассчитаем среднюю сумму квадратов расстояний до центроида, среднюю сумму средних внутрикластерных расстояний и среднюю сумму межкластерных расстояний для данного разбиения. Расчеты представлены на рисунке 7.

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

```
#сумма квадратов расстояний до центроида
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)
sum_sq_dist = np.sum(sum_sq_dist) / 4
sum_sq_dist
```

```
40.265351641556066
```

```
#средняя сумма средних внутрикластерных расстояний
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, :])
sum_avg_intercluster_dist = np.sum(sum_avg_intercluster_dist) / 4
sum_avg_intercluster_dist
```

```
1.5744197361976733
```

```
#сумма межкластерных расстояний
sum_intercluster_dist = np.sum(euclidean_distances(clusters, clusters))
sum_intercluster_dist
```

```
52.045210189629415
```

Рисунок 7 - Рассчитанные характеристики

Далее надо провести кластеризацию выборки методом k-средних. для k [1, 10]. Средняя сумма квадратов расстояний до центроида показана на рисунке 8.

```
[18] from sklearn.cluster import KMeans

     import warnings
     warnings.simplefilter(action='ignore', category=FutureWarning)
```

```
[19] 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))
```

```
[20] sum_sq_dist_avg = []
     for it, kmean in enumerate(models):
         sum_sq_dist_avg.append(kmean.inertia_ / (it + 1))
     sum_sq_dist_avg
```

```
[887.898555329913,
 249.2200983189639,
 98.78077306138114,
 39.84843754995185,
 25.119665177735136,
 16.442316855271237,
 11.709878250041088,
 8.663695690980182,
 6.786461482361825,
 5.313407596705827]
```

```
plt.plot(range(1, 11), sum_sq_dist_avg, '-o')
```

```
[<matplotlib.lines.Line2D at 0x7d899abb3670>]
```

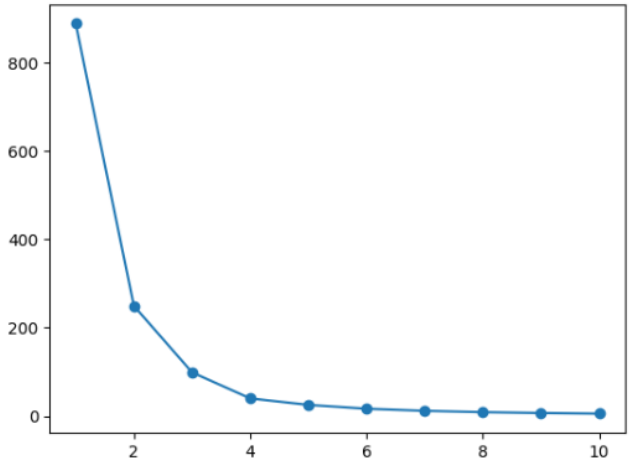

Рисунок 8 - Сумма квадратов расстояний до центроида

Средняя сумма средних внутрикластерных расстояний показана на рисунке 9.

```
[22] new_centers = [kmean.cluster_centers_ for kmean 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
```

```
[8.878985553299131,
 20.548525533365,
 20.768112953907305,
 23.927982520914885,
 24.30041412174638,
 12.428931811254477,
 16.701006390128313,
 16.269348864800193,
 7.617434346559047,
 9.315461734153725]
```

```
plt.plot(range(1, 11), sum_avg_intercluster_dist_avg, '-o')
```

```
[<matplotlib.lines.Line2D at 0x7d899ac23880>]
```

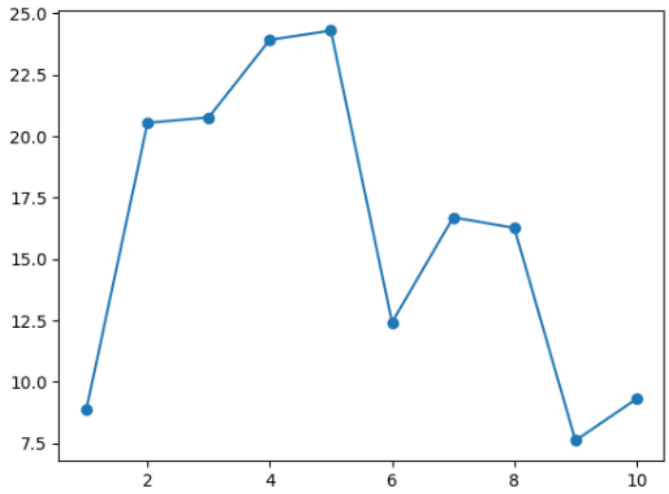

Рисунок 9 - Средняя сумма средних внутрикластерных расстояний

Средняя сумма средних межкластерных расстояний от количества кластеров показана на рисунке 10.

```
sum_intercluster_dist_avg = []

for k, kmean in enumerate(models):
    value = np.sum(euclidean_distances(kmean.cluster_centers_, kmean.cluster_centers_))
    sum_intercluster_dist_avg.append(value / (k + 1))
sum_intercluster_dist_avg
```

```
[0.0,
 3.947729496172375,
 8.472101291151235,
 13.033856141682863,
 17.1902998001549,
 22.290927238858018,
 26.65610231873841,
 30.419220049466055,
 33.3064882627388,
 38.73642980359689]
```

```
plt.plot(range(1, 11), sum_intercluster_dist_avg, '-o')
```

```
[<matplotlib.lines.Line2D at 0x7d8998a8d0f0>]
```

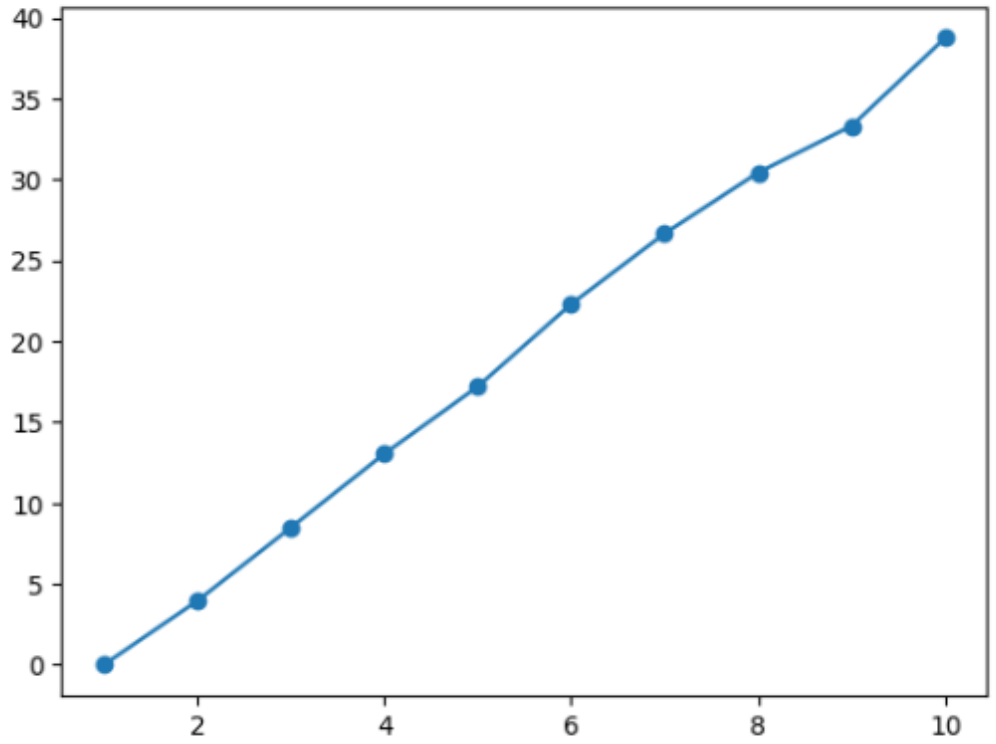

Рисунок 10 - Средняя сумма средних межкластерных расстояний от количества кластеров

Составим сравнительную таблицу результатов разбиения иерархическим методом и методом k-средних, показана на рисунке 11.

```python
import pandas as pd
```

```python
columns = pd.MultiIndex.from_product([['Иерархический метод', 'Метод k-средних'],
                                      ['Сумма квадратов расстояний до центроида', 'Сумма средних внутрикластерных расстояний', 'Сумма межкластерных расстояний
df = pd.DataFrame(columns=columns)
```

```python
df['Иерархический метод', 'Сумма квадратов расстояний до центроида'] = [sum_sq_dist for _ in range(len(sum_sq_dist_avg))]
df['Иерархический метод', 'Сумма средних внутрикластерных расстояний'] = [sum_avg_intercluster_dist for _ in range(len(sum_avg_intercluster_dist_avg))]
df['Иерархический метод', 'Сумма межкластерных расстояний'] = [sum_intercluster_dist for _ in range(len(sum_intercluster_dist_avg))]

df['Метод k-средних', 'Сумма квадратов расстояний до центроида'] = sum_sq_dist_avg
df['Метод k-средних', 'Сумма средних внутрикластерных расстояний'] = sum_avg_intercluster_dist_avg
df['Метод k-средних', 'Сумма межкластерных расстояний'] = sum_intercluster_dist_avg

df
```

	Иерархический метод			Метод k-средних		
	Сумма квадратов расстояний до центроида	Сумма средних внутрикластерных расстояний	Сумма межкластерных расстояний	Сумма квадратов расстояний до центроида	Сумма средних внутрикластерных расстояний	Сумма межкластерных расстояний
0	40.265352	1.57442	52.04521	887.898555	8.878986	0.000000
1	40.265352	1.57442	52.04521	249.220098	20.548526	3.947729
2	40.265352	1.57442	52.04521	98.780773	20.768113	8.472101
3	40.265352	1.57442	52.04521	39.848438	23.927983	13.033856
4	40.265352	1.57442	52.04521	25.119665	24.300414	17.190300
5	40.265352	1.57442	52.04521	16.442317	12.428932	22.290927
6	40.265352	1.57442	52.04521	11.709878	16.701006	26.656102
7	40.265352	1.57442	52.04521	8.663696	16.269349	30.419220
8	40.265352	1.57442	52.04521	6.786461	7.617434	33.306488
9	40.265352	1.57442	52.04521	5.313408	9.315462	38.736430

Рисунок 11 - Сравнительная таблица

Вывод

В результате выполнения работы были получены практические навыки решения задачи кластеризации фактографических данных в среде Jupiter Notebook, были настроены параметры методов и оценена точность полученного разбиения.