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 Цыганов Н.А.

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 по журналу 18, вариантов 12, следовательно: вариант 6 n_samples=100, n_features = 2, n_redundant = 0, n_informative = 2, n_cluster_per_class = 1, n_classes = 4. представлен на рисунке 1.

6
blobs
68
2
-
6

Рисунок 1 - Вариант для выполнения

Генерация данных для варианта представлена на рисунке 2.

```
[1]: from sklearn.datasets import make_blobs

[2]: X, y = make_blobs(n_samples=100,
                       centers=4,
                       n_features=2,
                       random_state=68,
                       cluster_std=2)
```

Рисунок 2 - Генерация данных

Отображение выборки на графике представлено на рисунке 3.

Отображение выборки на графике

```
]: import matplotlib.pyplot as plt

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

]: <matplotlib.collections.PathCollection at 0x13e7c8c90>
```

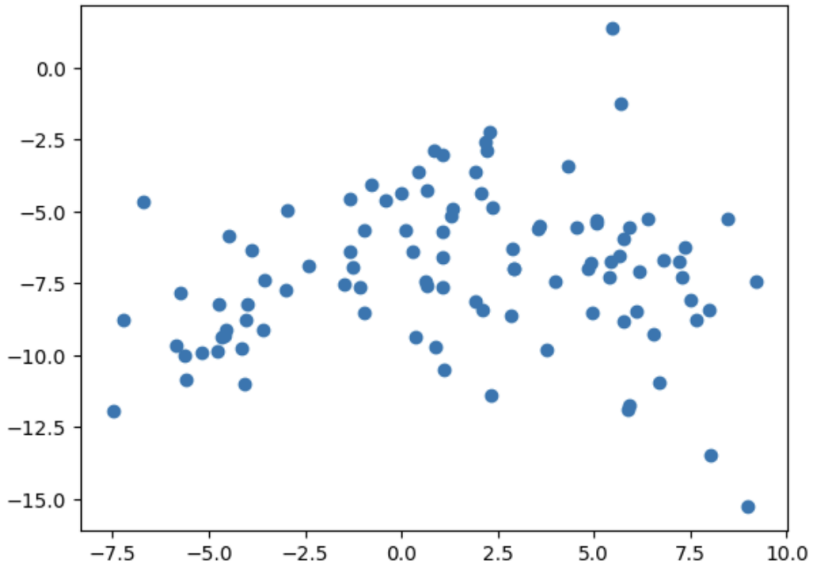

Рисунок 3 - Отображение выборки

Графики иерархической кластеризации представлены на рисунке 4.

Иерархическая кластеризация выборки

[6]:
```
mergings_single = linkage(X, method='single')
mergings_complete = linkage(X, method='complete')
mergings_ward = linkage(X, method='ward')
```

[7]:
```
# 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')
```

[7]:
```
Text(0.5, 1.0, 'Расстояние Уорда')
```

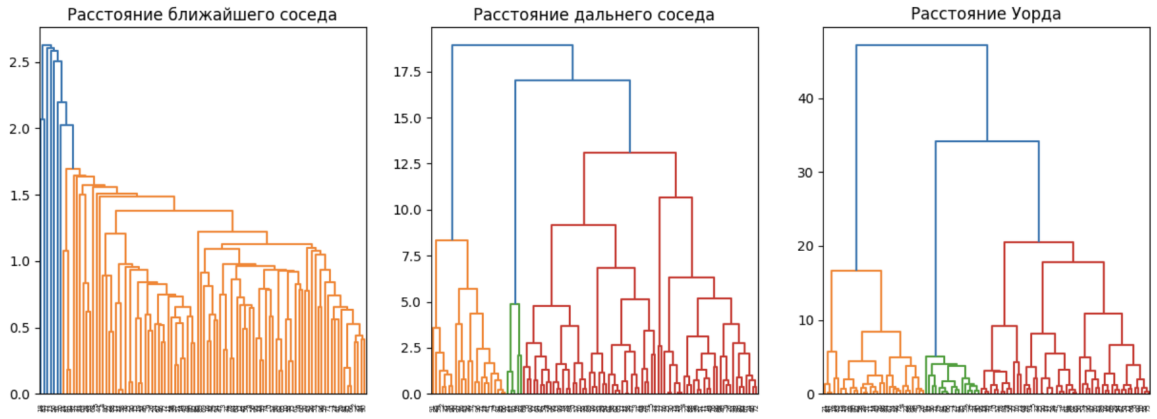

Рисунок 4 - Дендограммы

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 5.

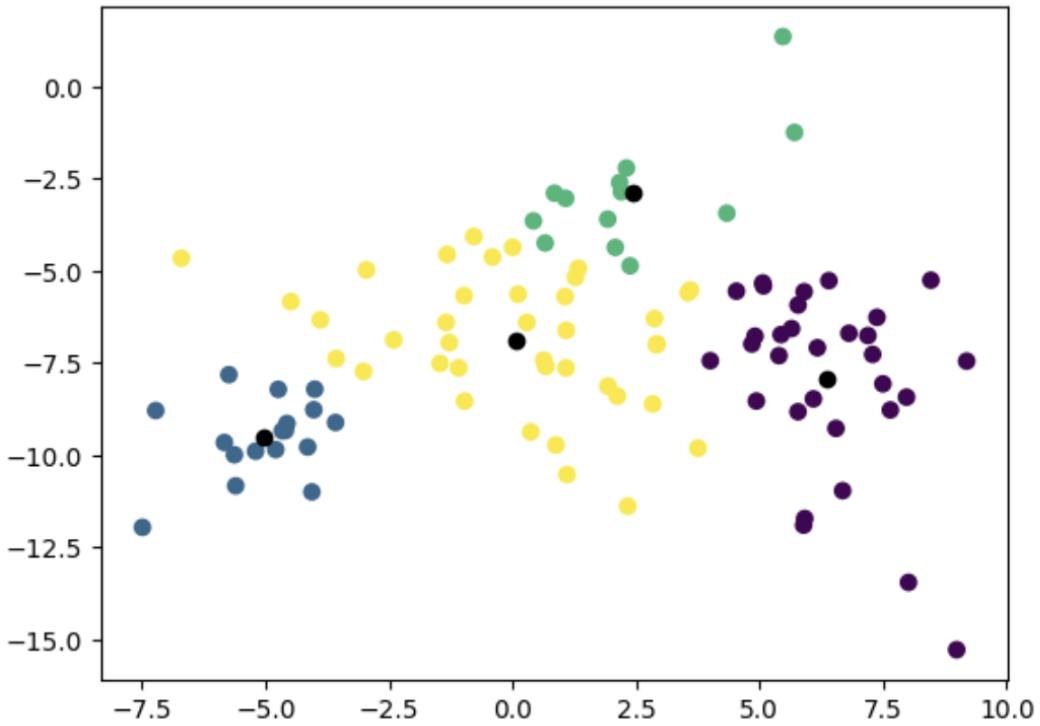

Рисунок 5 - График разбиения данных на кластеры

Рассчитаем среднюю сумму квадратов расстояний до центроида, среднюю сумму средних внутрикластерных расстояний и среднюю сумму межкластерных расстояний для данного разбиения. Расчеты представлены на рисунке 6.

```
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
```

```
171.04000058853853
```

```
#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
```

```
5.936502936490278
```

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

```
89.36575872390782
```

Рисунок 6 - Рассчитанные характеристики

Далее надо провести кластеризацию выборки методом k-средних. для k [1, 10]. Средняя сумма квадратов расстояний до центроида показана на рисунке 7.

```
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 it, kmean in enumerate(models):
    sum_sq_dist_avg.append(kmean.inertia_ / (it + 1))
sum_sq_dist_avg
```

```
[2592.844849872166,
 622.3067233497138,
 255.29521731231,
 146.7823357920375,
 90.6096691936283,
 61.27874740909241,
 44.43732421780813,
 32.57997194227077,
 25.75067263126529,
 20.1858098610387]
```

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

```
[<matplotlib.lines.Line2D at 0x1520190d0>]
```

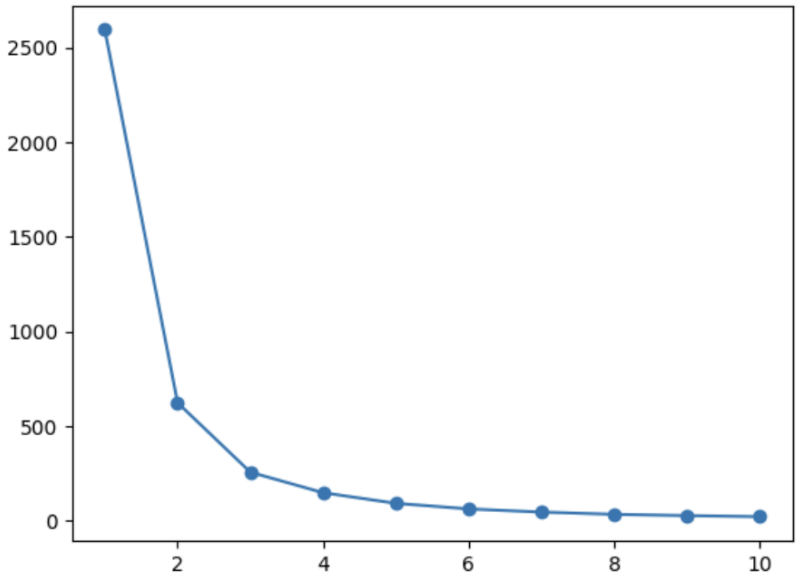

Рисунок 7 - Сумма квадратов расстояний до центроида

Средняя сумма средних внутрикластерных расстояний показана на рисунке 8.

```
]: 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) / ler
       sum_avg_intercluster_dist_avg.append(np.sum(intercluster_sum) / (k + 1))
   sum_avg_intercluster_dist_avg

]: [25.928448498721664,
    67.80377373606751,
    68.50494530182458,
    57.67590108365116,
    67.89976098520383,
    36.02398139770215,
    40.4686030903238,
    36.210584453484216,
    34.40324609073689,
    19.36411883829054]

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

]: [<matplotlib.lines.Line2D at 0x153c78cd0>]
```

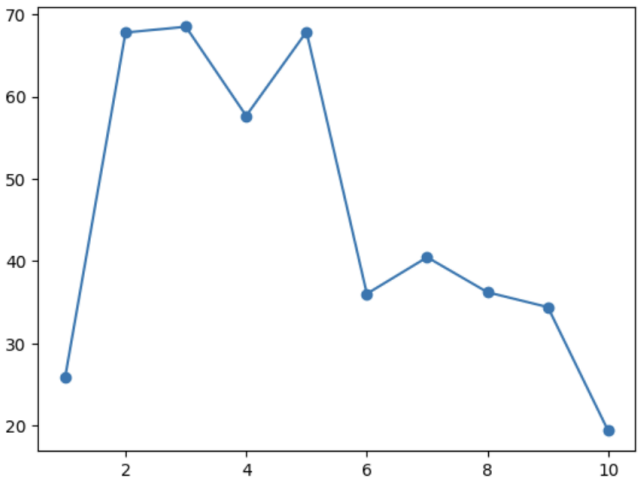

Рисунок 8 - Средняя сумма средних внутрикластерных расстояний

Средняя сумма средних межкластерных расстояний от количества кластеров показана на рисунке 9.

```
: 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,
   7.465592573607459,
   15.008098338276442,
   20.317430849561163,
   30.44979764840309,
   35.84223717885309,
   47.958427343806655,
   50.797728328930816,
   59.40927452228641,
   65.15978288636326]
```

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

```
: [<matplotlib.lines.Line2D at 0x153de8cd0>]
```

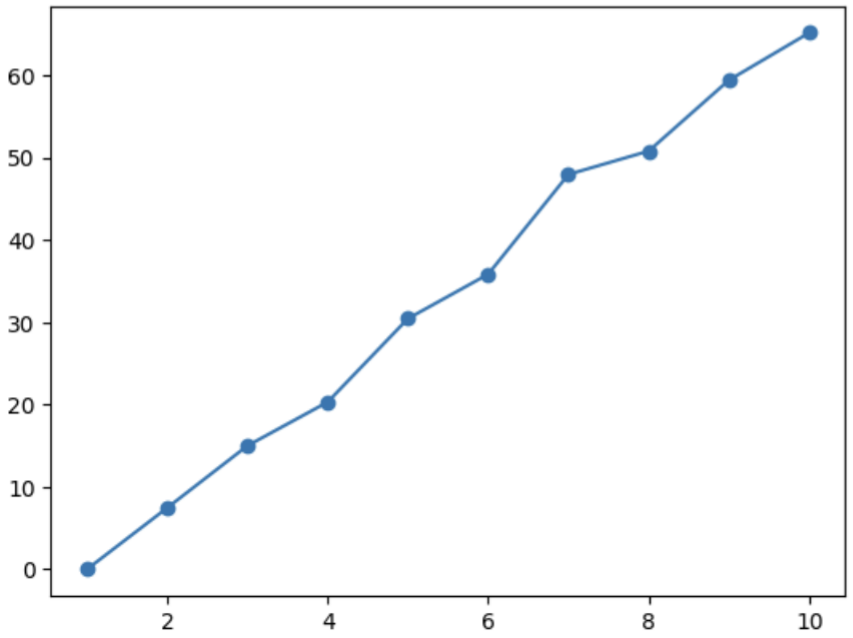

Рисунок 9 - Средняя сумма средних межкластерных расстояний от количества кластеров

Составим сравнительную таблицу результатов разбиения иерархическим методом и методом k-средних, показана на рисунке 10.

]:

	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	171.040001	5.936503	89.365759	2592.844850	25.928448	0.000000
1	171.040001	5.936503	89.365759	622.306723	67.803774	7.465593
2	171.040001	5.936503	89.365759	255.295217	68.504945	15.008098
3	171.040001	5.936503	89.365759	146.782336	57.675901	20.317431
4	171.040001	5.936503	89.365759	90.609669	68.907862	30.449798
5	171.040001	5.936503	89.365759	61.286112	36.157073	35.848507
6	171.040001	5.936503	89.365759	44.077527	30.888794	48.645945
7	171.040001	5.936503	89.365759	32.957975	28.678499	50.325094
8	171.040001	5.936503	89.365759	25.074366	28.322239	58.581943
9	171.040001	5.936503	89.365759	21.585000	30.081614	66.035509

Рисунок 10 - 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.