**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №7**

**по дисциплине «Статистические методы обработки экспериментальных данных»**

Тема: Кластерный анализ. Метод поиска сгущений.

|  |  |  |
| --- | --- | --- |
| Студентка гр. 7381 |  | Алясова А.Н. |
| Студент гр. 7381 |  | Кортев Ю.В. |
| Преподаватель |  | Середа А.-В.И. |

Санкт-Петербург

2021

**Цель работы.**

Освоение основных понятий и некоторых методов кластерного анализа, в частности, метода поиска сгущений.

**Основные теоретические положения.**

Метод поиска сгущений является еще одним итеративным методом кластерного анализа.

Основная идея метода заключается в построении гиперсферы заданного радиуса, которая перемещается в пространстве классификационных признаков в поисках локальных сгущений объектов.

Метод поиска сгущений требует, прежде всего, вычисления матрицы расстояний (или матрицы мер сходства) между объектами и выбора первоначального центра сферы.

В алгоритме поиска сгущений сначала выбирается начальный центр первого кластера. Выбор такого объекта может быть произвольным, а может основываться на предварительном анализе точек и их окрестностей. В рассматриваемом случае, центры выбираются вручную.

Как правило, на первом шаге центром сферы служит объект (точка), в ближайшей (заданной) окрестности которого расположено наибольшее число соседей. На основе заданного радиуса сферы (R) определяется совокупность точек внутри этой сферы, и для них вычисляются координаты центра (вектор средних для попавших в сферу значений признаков).

Когда очередной пересчет координат центра сферы приводит к такому же результату, как и на предыдущем шаге, перемещение сферы прекращается, а точки, попавшие в нее, образуют кластер, и из дальнейшего процесса кластеризации исключаются. Перечисленные процедуры повторяются для всех оставшихся точек. Работа алгоритма завершается за конечное число шагов, и все точки оказываются распределенными по кластерам. Число образовавшихся кластеров заранее неизвестно и сильно зависит от заданного радиуса сферы.

Для оценки устойчивости полученного разбиения целесообразно повторить процесс кластеризации несколько раз для различных значений радиуса сферы, изменяя каждый раз радиус на небольшую величину.

Существуют различные способы выбора начального радиуса сферы. В частности, если обозначить через расстояние между -м и -м объектами, то в качестве нижней границы значения радиуса сферы можно выбрать минимальное из таких расстояний, а в качестве верхней границы - максимальное:

Тогда, если начинать работу алгоритма с

и при каждом его повторении увеличивать значение на некоторую величину, то в конечном итоге можно найти значения радиусов, которые приводят к устойчивому разбиению на кластеры.

Следует отметить следующие существенные при реализации метода поиска сгущений моменты:

1. В случае разномасштабности квалификационных признаков необходимо проведение их нормировки перед началом работы метода;
2. Возможны два варианта реализации метода. Один из них не предполагает изменения заданного значения радиуса сферы до завершения кластеризации, а другой — предполагает изменение этого радиуса в процессе кластеризации при начале построения очередной сферы;
3. В отличие от метода k-средних метод поиска сгущений не требует задания количества кластеров, на которые предполагается разбить исходное множество объектов;
4. Качество полученного в результате применения метода итогового разбиения на кластеры оценивается, как и в методе k-средних, с помощью введенных на предыдущей лекции критериев качества разбиения , ,.
5. Получение в результате кластеризации пересекающихся кластеров (наличие спорных объектов) в принципе является неудовлетворительным результатом. На практике в этом случае необходимо скорректировать процесс, либо выбрать другой метод кластеризации.

После завершения многомерной классификации необходимо оценить полученные результаты. Для этой цели используются специальные характеристики – функционалы качества. Наилучшим разбиением считается такое, при котором достигается экстремальное (минимальное или максимальное) значение выбранного функционала качества.

В качестве таких функционалов могут быть использованы:

1. Сумма квадратов расстояний до центров кластеров



2. Сумма внутрикластерных расстояний между объектами



3. Сумма внутрикластерных дисперсий

Здесь  - дисперсия *j*-й переменной в *k*-м кластере.

Оптимальным следует считать разбиение, при котором сумма внутрикластерных (внутригрупповых) дисперсий будет минимальной.

Судить о качестве разбиения позволяют и некоторые простейшие приемы. Например, можно сравнивать средние значения признаков в отдельных кластерах (группах) со средними значениями в целом по всей совокупности объектов. Если групповые средние существенно отличаются от общего среднего значения, то это может являться признаком хорошего разбиения.

**Постановка задачи.**

Дано конечное множество из объектов, представленных двумя признаками (в качестве этого множества принимаем исходную двумерную выборку, сформированную ранее в лабораторной работе №4). Выполнить разбиение исходного множества объектов на конечное число подмножеств (кластеров) с использованием метода поиска сгущений. Полученные результаты содержательно проинтерпретировать.

**Порядок выполнения работы**

1. Нормализовать множество точек, отобразить полученное множество.
2. Реализовать алгоритм поиска сгущений, отобразить полученные кластеры, выделить каждый кластер разным цветом, отметить центроиды.
3. Проверить чувствительность метода к погрешностям. Сделать выводы.
4. Сравнить с методами из лабораторной работы №6. Сделать выводы.

**Выполнение работы.**

**1. Нормализовать множество точек, отобразить полученное множество.**

Исследуемая выборка представлена в табл. 1.

Таблица 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 480 | 153,3 | 25 | 408 | 110,0 | 49 | 405 | 103,6 | 73 | 465 | 127,7 | 97 | 487 | 146,0 |
| 2 | 510 | 129,4 | 26 | 331 | 74,1 | 50 | 434 | 140,4 | 74 | 390 | 108,1 | 98 | 532 | 158,7 |
| 3 | 426 | 119,0 | 27 | 467 | 113,0 | 51 | 344 | 86,8 | 75 | 463 | 129,2 | 99 | 330 | 71,1 |
| 4 | 482 | 139,9 | 28 | 545 | 145,3 | 52 | 415 | 119,7 | 76 | 468 | 128,9 | 100 | 438 | 134,1 |
| 5 | 393 | 103,2 | 29 | 396 | 83,8 | 53 | 463 | 136,7 | 77 | 488 | 134,1 | 101 | 593 | 187,4 |
| 6 | 510 | 162,3 | 30 | 351 | 102,9 | 54 | 475 | 143,6 | 78 | 443 | 137,4 | 102 | 445 | 124,7 |
| 7 | 403 | 123,9 | 31 | 503 | 148,5 | 55 | 463 | 144,9 | 79 | 505 | 155,8 | 103 | 518 | 154,0 |
| 8 | 506 | 158,4 | 32 | 402 | 120,8 | 56 | 392 | 82,7 | 80 | 395 | 109,1 | 104 | 496 | 141,7 |
| 9 | 393 | 122,8 | 33 | 542 | 146,1 | 57 | 452 | 140,5 | 81 | 474 | 132,5 | 105 | 473 | 136,4 |
| 10 | 442 | 115,4 | 34 | 437 | 124,3 | 58 | 504 | 143,8 | 82 | 490 | 139,9 | 106 | 522 | 154,5 |
| 11 | 411 | 112,9 | 35 | 453 | 119,5 | 59 | 443 | 122,9 | 83 | 396 | 90,1 | 107 | 547 | 154,7 |
| 12 | 514 | 153,6 | 36 | 386 | 105,8 | 60 | 461 | 138,6 | 84 | 362 | 97,9 | 108 | 560 | 169,8 |
| 13 | 525 | 156,5 | 37 | 434 | 122,3 | 61 | 340 | 85,1 | 85 | 566 | 175,7 | 109 | 412 | 127,8 |
| 14 | 543 | 155,4 | 38 | 418 | 118,4 | 62 | 438 | 134,9 | 86 | 418 | 109,3 | 110 | 444 | 130,0 |
| 15 | 412 | 116,3 | 39 | 391 | 107,5 | 63 | 523 | 148,7 | 87 | 502 | 132,5 | 111 | 437 | 121,8 |
| 16 | 449 | 124,5 | 40 | 399 | 100,0 | 64 | 416 | 120,5 | 88 | 500 | 155,5 | 112 | 462 | 138,8 |
| 17 | 482 | 136,4 | 41 | 486 | 139,4 | 65 | 483 | 143,4 | 89 | 359 | 71,9 | 113 | 438 | 122,2 |
| 18 | 569 | 157,4 | 42 | 421 | 124,2 | 66 | 440 | 128,5 | 90 | 443 | 135,7 | 114 | 406 | 110,1 |
| 19 | 484 | 147,5 | 43 | 496 | 143,1 | 67 | 423 | 131,1 | 91 | 421 | 118,0 | 115 | 413 | 106,7 |
| 20 | 472 | 134,2 | 44 | 463 | 121,2 | 68 | 386 | 95,5 | 92 | 433 | 128,2 | 116 | 458 | 121,7 |
| 21 | 453 | 124,2 | 45 | 508 | 159,0 | 69 | 321 | 86,1 | 93 | 514 | 174,6 | 117 | 408 | 117,0 |
| 22 | 422 | 117,9 | 46 | 419 | 105,3 | 70 | 433 | 131,5 | 94 | 320 | 72,6 |  |  |  |
| 23 | 320 | 64,5 | 47 | 434 | 108,7 | 71 | 351 | 89,0 | 95 | 406 | 113,8 |  |  |  |
| 24 | 547 | 164,4 | 48 | 440 | 126,7 | 72 | 481 | 148,3 | 96 | 465 | 140,9 |  |  |  |

Отображение исходной выборки представлено на рис. 1.

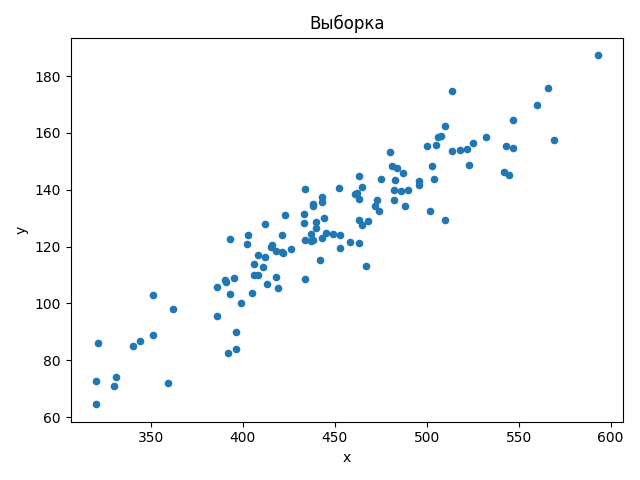


Рисунок 1 – Исходная выборка

Нормализация координат точек определяется по формуле:

Отображение нормализованной выборки представлено на рис. 2.



Рисунок 2 – Нормализованная выборка

**2. Реализовать алгоритм поиска сгущений, отобразить полученные кластеры, выделить каждый кластер разным цветом, отметить центроиды.**

Реализуем алгоритм поиска сгущений. Отобразим полученные кластеры, выделим каждый кластер разным цветом, отметим центроиды.

Определим нижнюю и верхнюю границы радиуса сферы:

Выберем из промежутка [] радиус .

Запустим алгоритм:

Формирование 1го кластера представлено на рис. 3.

|  |  |
| --- | --- |
|  |  |
|  |  |

Рисунок 3

Формирование 2го кластера представлено на рис. 4.

|  |  |
| --- | --- |
|  |  |
|  |  |

Рисунок 4

Формирование 3го кластера представлено на рис. 5-6.

|  |  |
| --- | --- |
|  |  |

Рисунок 5

|  |  |
| --- | --- |
|  |  |
|  |  |

Рисунок 6

Формирование 4го кластера представлено на рис. 7.

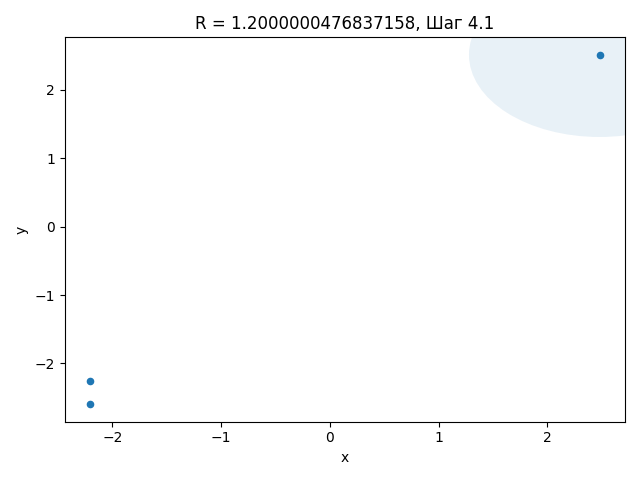


Рисунок 7

Формирование 5го кластера представлено на рис. 8.

|  |  |
| --- | --- |
|  |  |

Рисунок 8

Результат кластеризации представлен на рис. 9.

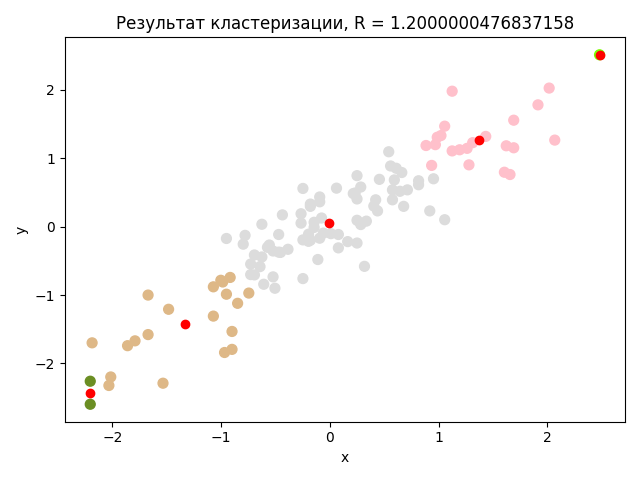


Рисунок 9

Таблица 2

|  |  |  |
| --- | --- | --- |
| Номер  кластера | Центр кластера | Количество элементов в кластере |
| 1 | (-0,004243736116897832 ; 0,05696773691479348) | 73 |
| 2 | (1,3728603917013382 ; 1,269402589075278) | 21 |
| 3 | (-1,3296908010805915 ; -1,423519995425545) | 20 |
| 4 | (2,4783418922683094 ; 2,5082120327432573) | 1 |
| 5 | (-2,2024006799255216 ; -2,4269556447966085) | 2 |

**3. Проверить чувствительность метода к погрешностям. Сделать выводы.**

Формирование 1го кластера с погрешностями представлено на рис. 10.

|  |  |
| --- | --- |
|  |  |
|  |  |

Рисунок 10

Формирование 2го кластера с погрешностями представлено на рис. 11.

|  |  |
| --- | --- |
|  |  |

Рисунок 11

Формирование 2го кластера с погрешностями представлено на рис. 12.

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |

Рисунок 12

Формирование 2го кластера с погрешностями представлено на рис. 13.

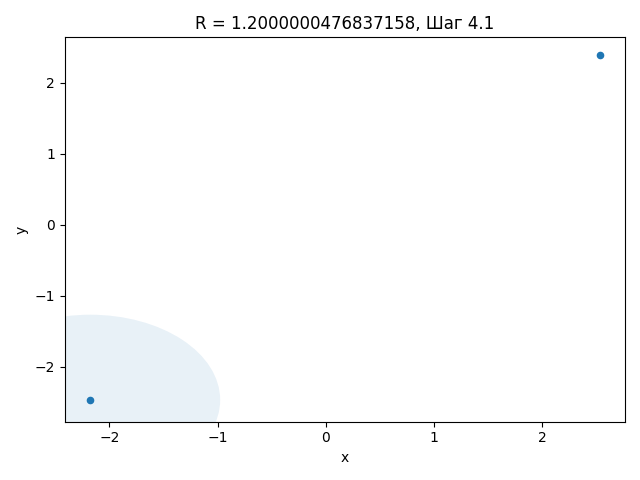
****

Рисунок 13

Формирование 5го кластера с погрешностями представлено на рис. 14.

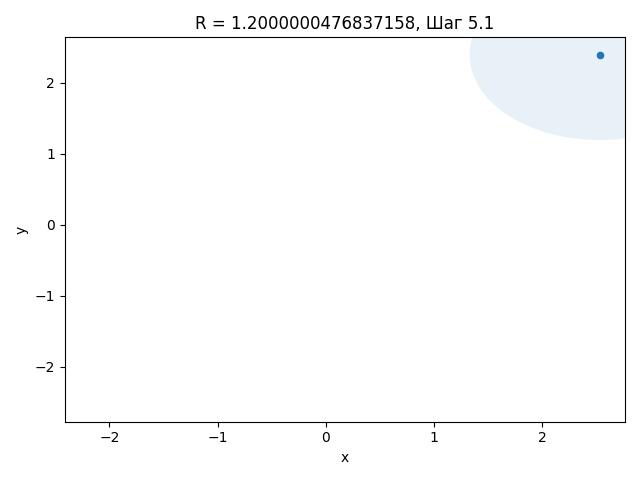
****

Рисунок 14

Результат кластеризации с погрешностями представлен на рис. 15.

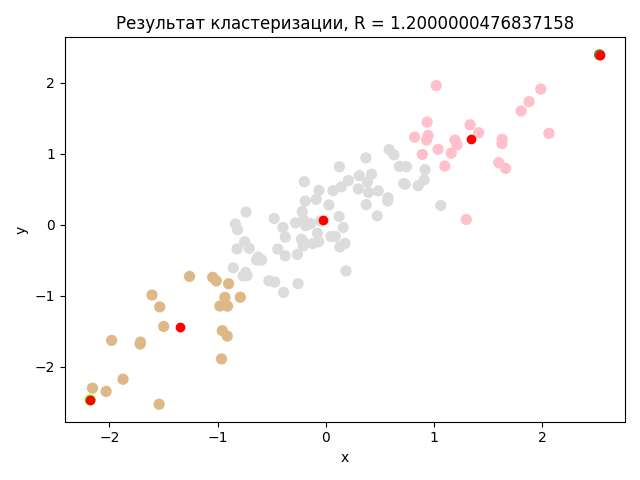
****

Рисунок 15

Таблица 3

|  |  |  |
| --- | --- | --- |
| Номер  кластера | Центр кластера | Количество элементов в кластере |
| 1 | (-0,0294042169854218 ; 0,06868813095241261) | 72 |
| 2 | (1,342186825428838 ; 1,210104850517267) | 22 |
| 3 | (-1,34686405930226 ; -1,4401333952484419) | 21 |
| 4 | (-2,1754394392808516 ; -2,465932878978102) | 1 |
| 5 | (2,528303696654842 ; 2,393057608215714) | 1 |

Для определения чувствительности метода поиска сгущений к погрешностям посчитаем функционалы качества:

1. Сумма квадратов расстояний до центров кластеров



2. Сумма внутрикластерных расстояний между объектами



3. Сумма внутрикластерных дисперсий

Здесь - дисперсия -й переменной в -м кластере.

Для метода поиска сгущений рассчитаем функционалы качества:

Для метода поиска сгущений c учетом погрешностей рассчитаем функционалы качества:

На основании этого можем сделать вывод, что метод несильно, но чувствителен к погрешностям, так как значения функционалов качества с учетом погрешностей возросли.

**4. Сравнить с методом из лабораторной работы №6. Сделать выводы.**

Для сравнения с методом k-средних возьмем значения функционалов качества при количестве кластеров из лабораторной работы №6.

Таблица 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Метод** | **Количество кластеров** |  |  |  |
| k-средних | 5 |  |  |  |
| поиска сгущений | 5 |  |  |  |

Визуальное представление работы двух методов:

|  |  |
| --- | --- |
| Метод поиска сгущений | Метод k-средних |
|  |  |

Рисунок 16

Рассмотрим результаты работы двух методов на рис. 16. Визуальная разница довольно большая. В данном случае алгоритм k-средних показал себя лучше так, как максимальное расстояние между точками в его кластерах меньше, чем у метода поиска сгущений. Сравним значения функционалов качества для данных разбиений (табл. 4). Аналогичный вывод можно сделать для сравнения по функционалам качества: метод k-средних лучше показал себя для исходных данных.

**Выводы.**

Таким образом, были освоены основные понятия кластерного анализа. С помощью алгоритма поиска сгущений исходная выборка была разбита на 5 кластеров. Метод поиска сгущений несильно, но чувствителен к погрешностям. При сравнении исследуемого метода с методом k-средних метод поиска сгущений оказался хуже для исходных данных.

По результатам работы, можно заметить, что алгоритм требует значительное количество итераций, также заранее неизвестно количество получаемых кластеров, оно зависит от выбранного радиуса.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД

import numpy as np

import sys

import math

import pandas as pd

import matplotlib.pyplot as plt

import matplotlib.colors as colors

import random

from itertools import combinations

from scipy.spatial import distance

random.seed(5)

np.random.seed(5)

df = pd.read\_csv('sample.csv', header=None)

df.columns = ['x', 'y']

ax = df.plot.scatter(x=0, y=1)

ax.set\_title('Выборка')

plt.show()

df = (df - df.mean(axis=0)) / df.std(axis=0)

noise = np.random.normal(0, 0.1, [len(df), 2])

df = df + noise

ax = df.plot.scatter(x=0, y=1)

ax.set\_title('Нормализованная выборка')

xlim = ax.get\_xlim()

ylim = ax.get\_ylim()

plt.show()

distances = distance.cdist(df, df)

distances = set(distances.flatten().tolist()) - {0}

R\_min = min(distances)

R\_max = max(distances)

def plot\_step(step\_points, circle, title):

ax = step\_points.plot.scatter(x=0, y=1)

ax.set\_title(title)

ax.set\_xlim(xlim)

ax.set\_ylim(ylim)

c = plt.Circle(circle, R, alpha=0.1)

ax.add\_patch(c)

plt.show()

points = df.copy()

R = np.float32(1.2) # 0.8

i = 0

df['Clusters'] = 0

while len(points):

circle = points.iloc[:, :2].sample(1)

j = 0

while True:

prev\_circle = circle

points\_in\_circle = points.apply(lambda x: np.linalg.norm(x - circle) <= R, axis=1)

circle = points[points\_in\_circle].mean(axis=0)

plot\_step(points, circle, 'R = {}, Шаг {}.{}'.format(R.round(1), i + 1, j + 1))

j += 1

if ((circle - prev\_circle).abs() < 0.0001).all().all():

break

points\_in\_circle = points.apply(lambda x: np.linalg.norm(x - circle) <= R, axis=1)

df.loc[points\_in\_circle.index, 'Clusters'] = points\_in\_circle \* i

points = points[~ points\_in\_circle]

i += 1

list\_colors = np.array([name for name, col in colors.CSS4\_COLORS.items()])

np.random.shuffle(list\_colors)

ax = df.plot.scatter(x=0, y=1, c=list\_colors[df['Clusters']], s=50)

for c in range(i):

circle = df[df['Clusters'] == c].iloc[:, :2].mean(axis=0)

ax.scatter(circle[0], circle[1], c='red')

ax.set\_title('Результат кластеризации, R = {}'.format(R.round(1)))

plt.show()

f1 = []

f2 = []

f3 = []

for c in range(i):

if np.isnan(df[df['Clusters'] == c].iloc[:, :2].var().mean()):

continue

circle = df[df['Clusters'] == c].iloc[:, :2].mean(axis=0)

cluster\_dists = []

f3.append(df[df['Clusters'] == c].iloc[:, :2].var().mean())

for comb in combinations(df[df['Clusters'] == c].iloc[:, :2].to\_numpy(), 2):

cluster\_dists.append(np.linalg.norm(comb[0] - comb[1]) \*\* 2)

f2.append(sum(cluster\_dists))

f1.append(sum(np.linalg.norm(df[df['Clusters'] == c].iloc[:, :2] - circle, axis=1) \*\* 2))

f2 = sum(f2)

f3 = sum(f3)

f1 = sum(f1)

print('R = {}:\nF1 = {}\nF2 = {}\nF3 = {}'.format(R, f1, f2, f3))

rows = []

for centroid\_id in range(i):

centroid = df[df['Clusters'] == centroid\_id].iloc[:, :2]

rows.append([centroid\_id + 1, '({} : {})'.format(\*(centroid.mean(axis=0))), len(centroid)])

res = pd.DataFrame(rows, columns=['Номер кластера', 'Центр кластера', 'Количество элементов в кластере'])

res.to\_csv('Таблица.csv', index=False)

print('R\_min = {}\nR\_max = {}'.format(R\_min, R\_max))