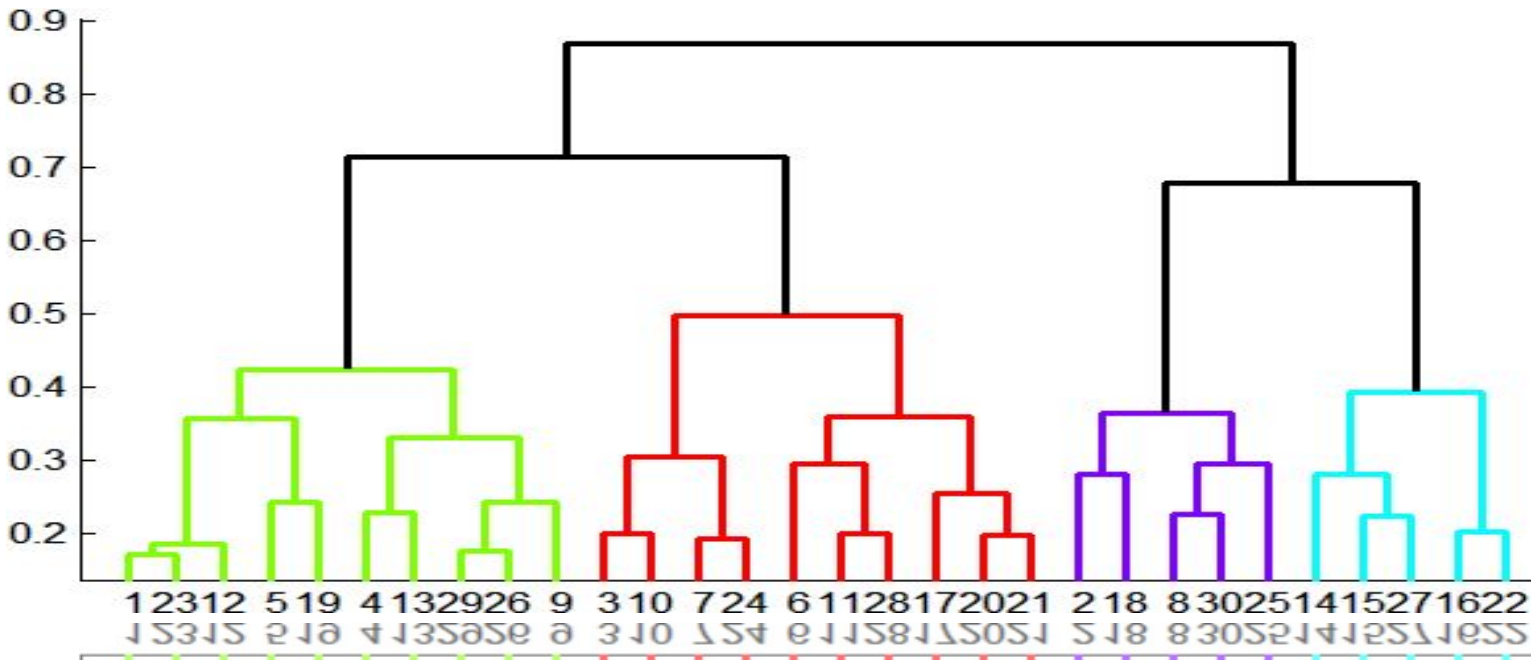
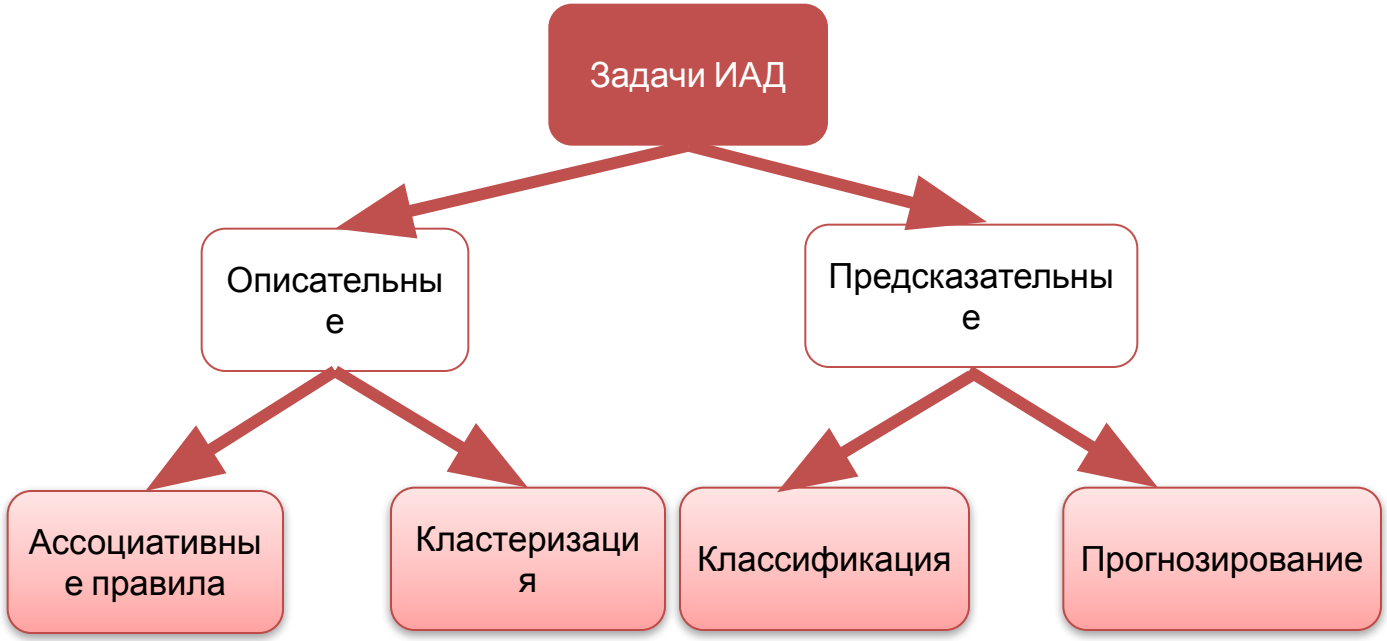



# Методы кластеризации

# Задачи интеллектуального анализа данных



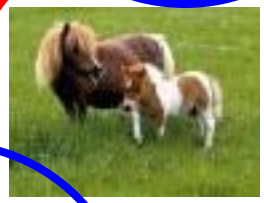
# Введение

- Задача кластеризации состоит в разделении исследуемого множества объектов на группы «похожих» объектов, называемых *кластерами*
- Решение задачи кластеризации называют *кластерным анализом*

- Кластеризация отличается от классификации тем, что этап обучения на примерах отсутствует
- В задачах классификации множество классов заранее известно, в кластеризации классы определяются в процессе анализа
- Поэтому кластеризация относится к задачам *обучения без учителя* (unsupervised learning)

- Задача кластеризации часто решается на начальных этапах исследования, когда о данных мало что известно
- Ее решение помогает лучше понять данные
- После определения кластеров применяются другие методы Data Mining, чтобы попытаться установить, что означает такое разбиение
- Кластерный анализ позволяет рассматривать достаточно большой объем информации и сжимать большие массивы информации, делать их компактными и наглядными

# ПРИМЕР –кластеризация результатов поиска



60

# Формальная постановка задачи

- Дано множество данных, состоящее из $N$ объектов (векторов):

$$S_1, S_2, \ldots, S_N$$

- Каждый объект описывается набором признаков:

$$x_1, x_2, \ldots, x_m,$$

где $m$ – размерность пространства признаков

# Формальная постановка задачи

- Таким образом, *i*-й объект можно записать в виде:

$$S_i = (x_{i1}, x_{i2}, \ldots, x_{im})$$

- Класс для каждого объекта неизвестен

# Формальная постановка задачи

Требуется:

- найти способ сравнения $d(S_p, S_q)$ объектов между собой (меру сходства, функцию расстояния)
- определить множество кластеров

$$C_1, C_2, \ldots, C_r$$

причем количество кластеров $r$ – неизвестно

- разбить данные по кластерам

# Метрики расстояния между объектами

- евклидово расстояние

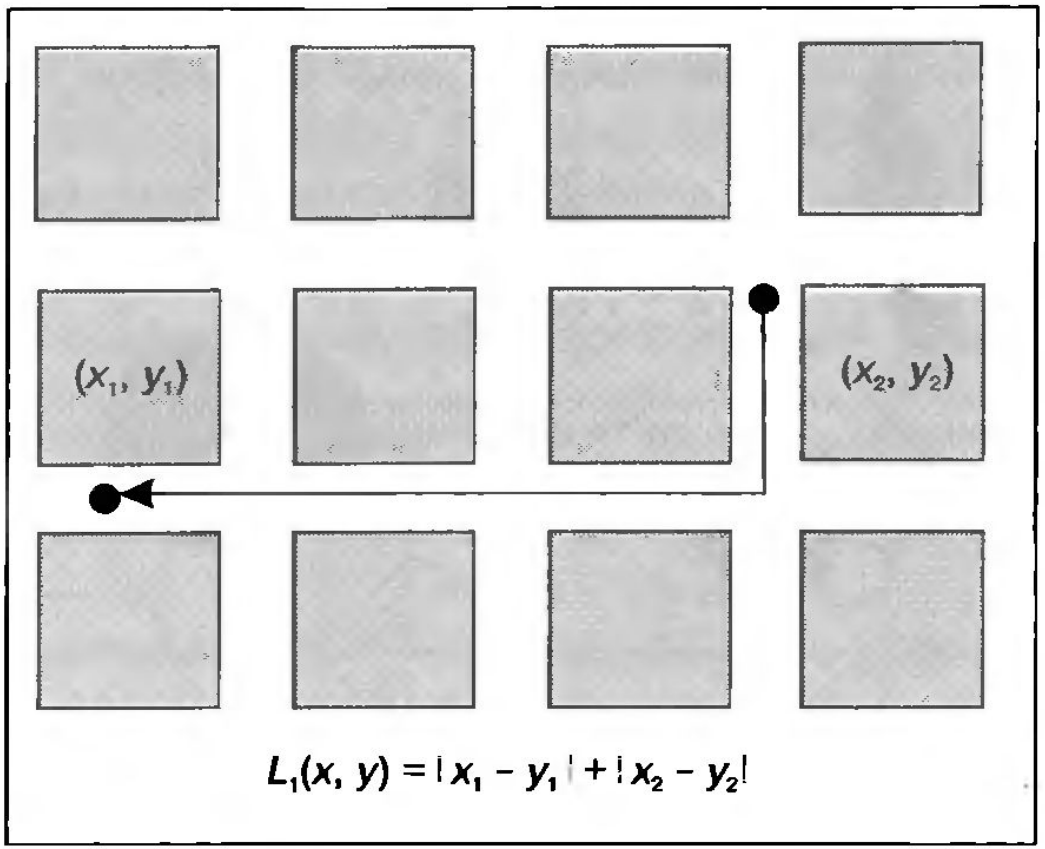
$$d(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^2\right)^{1/2}$$

- Манхэттенское расстояние

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

- расстояние Чебышева

$$d(x,y) = \max_{1 \le i \le n} |x_i - y_i|$$



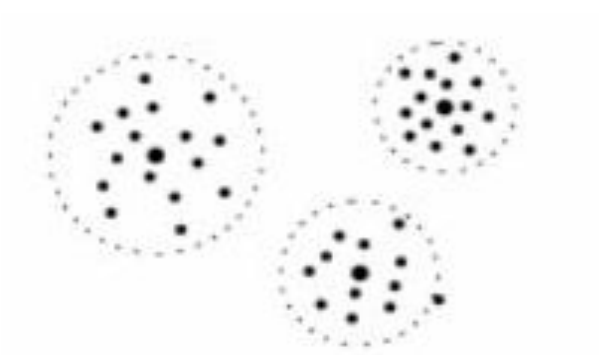

Методы кластерного анализа можно разделить на две группы:

- неиерархические
- иерархические

# Виды кластеров

Внутрикластерные расстояния, как правило, меньше межкластерных

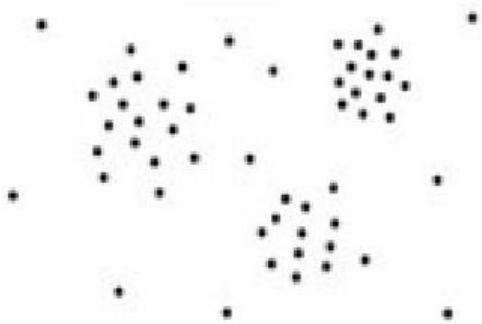
Но бывают ленточные кластеры, в которых внутрикластерные расстояния большие



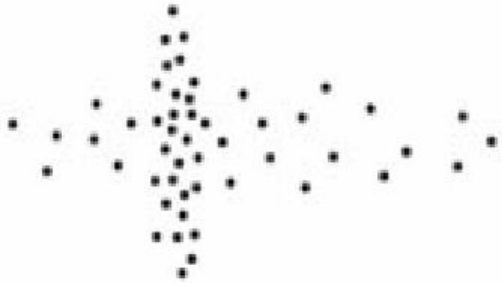
Идеальный случай- сферические кластеры с центром (встречаются редко)

# Разные виды кластеров ведут к проблеме выбора оптимального алгоритма кластеризации

кластеры могут соединяться перемычками



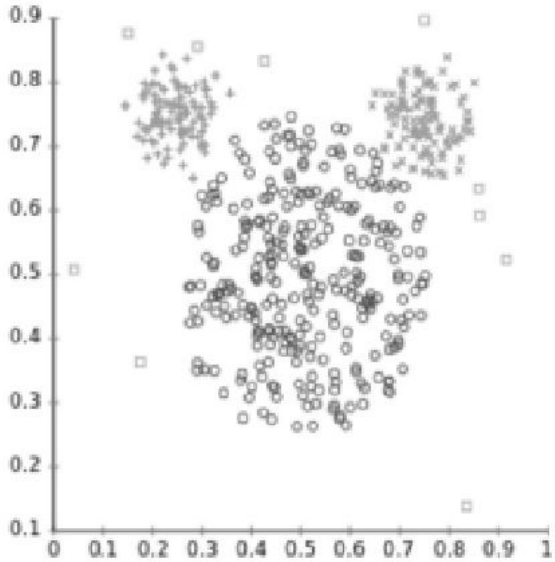
кластеры могут накладываться на разрежен-ный фон из редко расположенных объектов



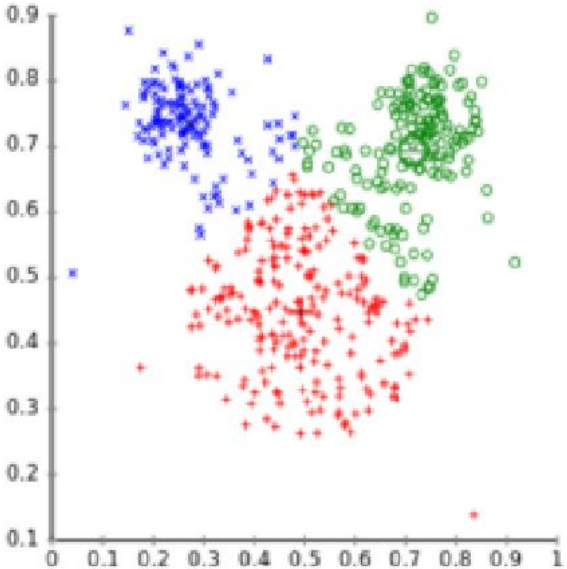
кластеры могут перекрываться

# Алгоритмы кластеризации

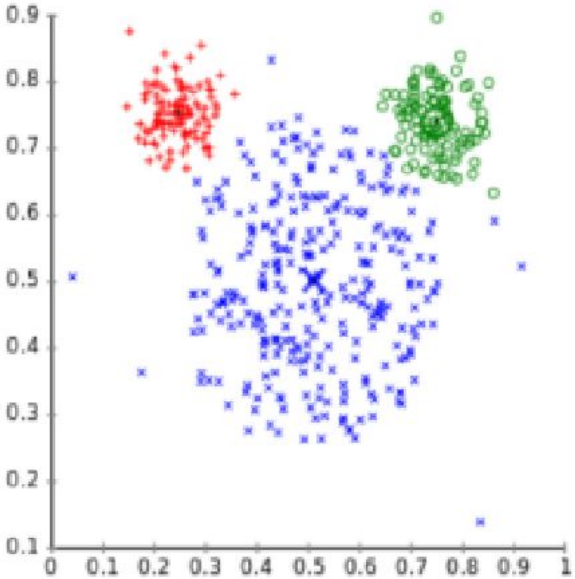
## Различия в результатах работы




Исходная выборка
("Mouse" dataset)




Метод 1




Метод 2

# Стандартизация данных

Как сделать признаки
равноправными в образовании кластеров?

$$d(x,y) = \max_{1 \le i \le n} |x_i - y_i|$$

ИТОГ: мы получим значения признаков, 95% которых находится в интервале (-2;2)

# Метод k-средних

- Неиерархическим методом кластеризации является метод k-средних (k-means)
- Предварительно необходимо выбрать вероятное число кластеров *k*

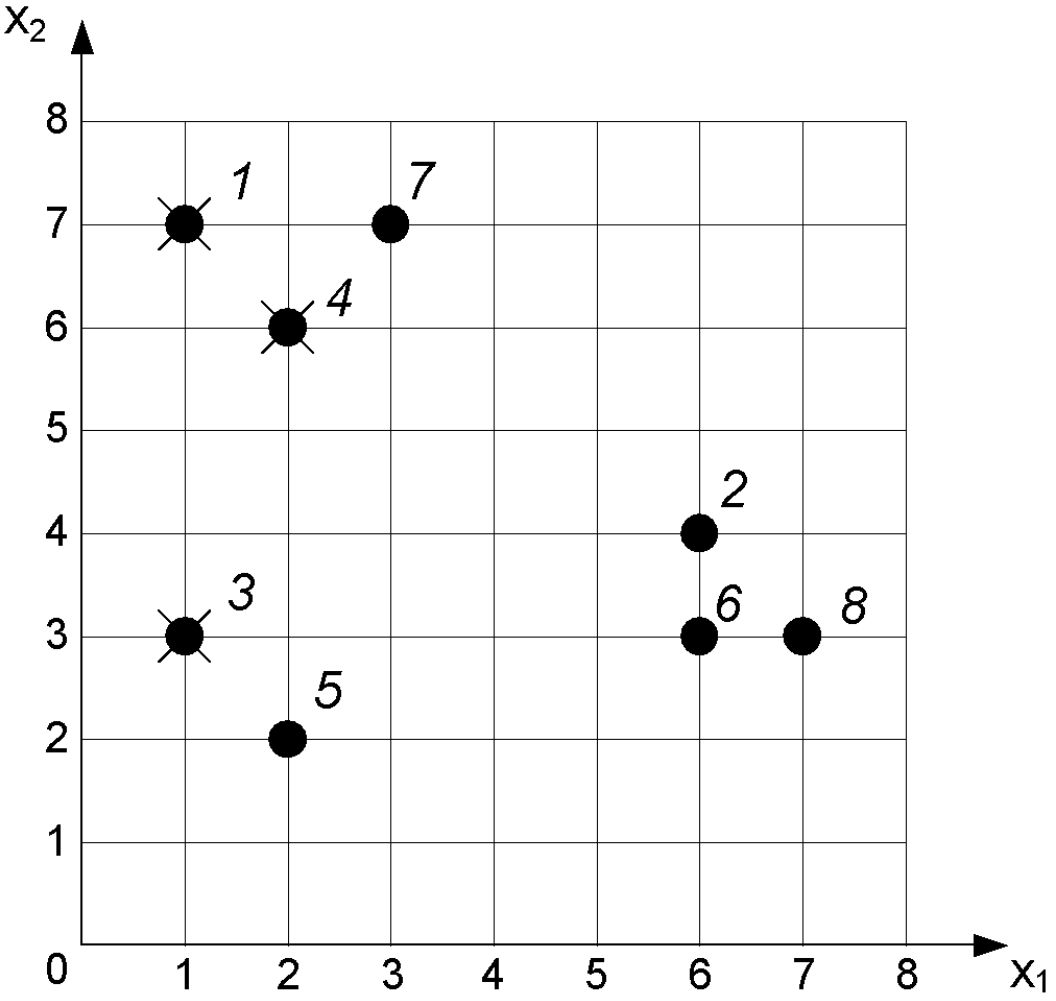
# Метод k-средних

1. Выбирается *k* произвольных исходных центров кластеров – обычно выбираются *k* объектов
2. Все объекты разбиваются на *k* групп, наиболее близких к одному из центров
3. Вычисляются новые центры кластеров
4. Проводится новое разбиение всех объектов на основании близости к новым центрам

Шаги 3 и 4 повторяются до тех пор, пока центры кластеров не перестанут меняться или пока не достигнуто максимальное число итераций

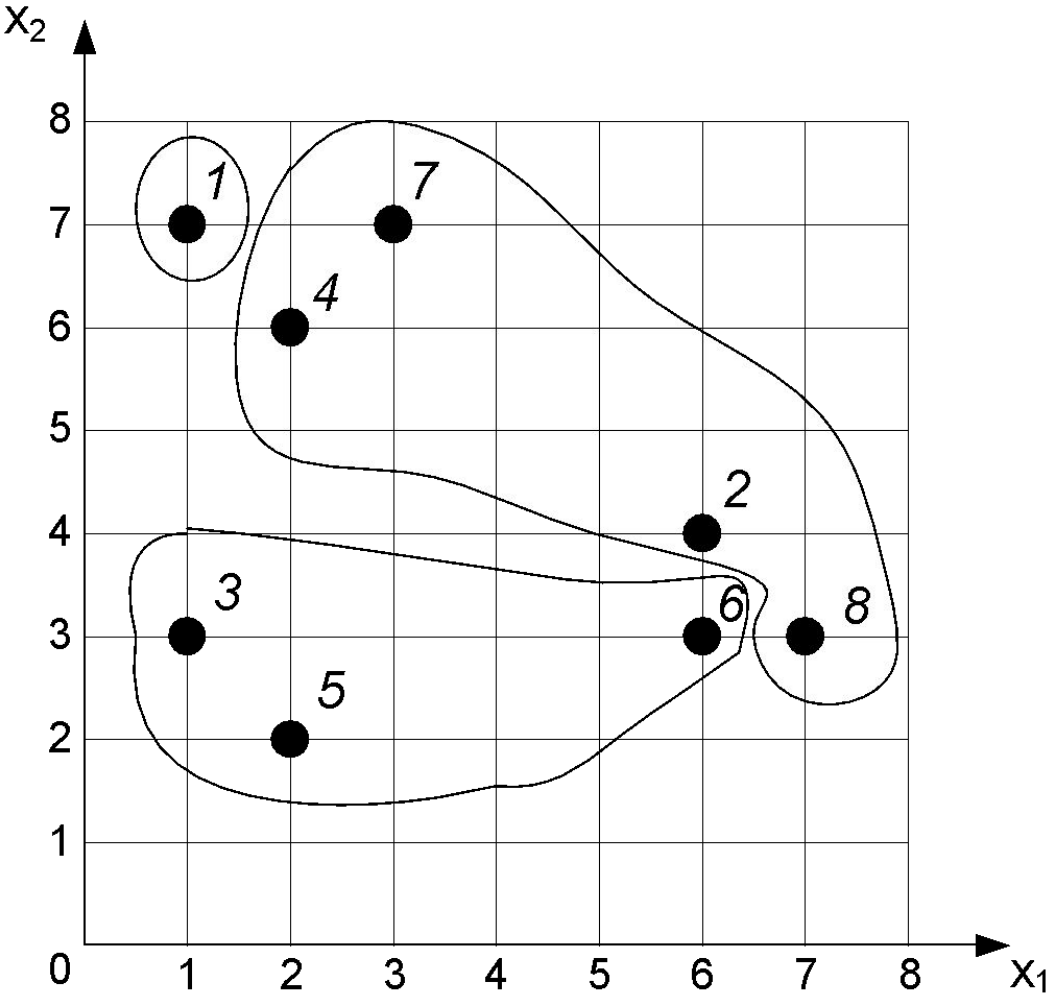
# Метод k-средних

- ***Пример***.
- Примем $k = 3$
- Начальные центры – объекты 1, 3, 4
- Разобьем все объекты по кластерам



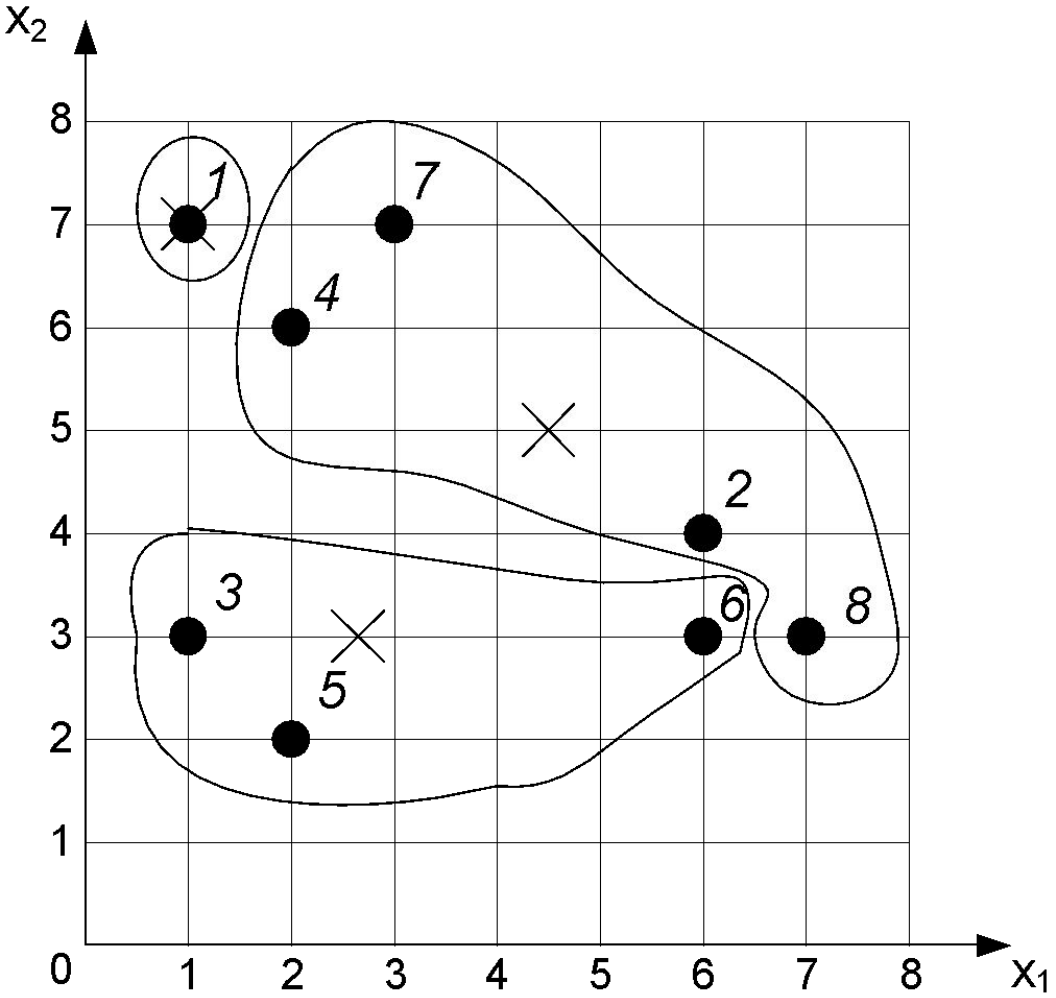
# Метод k-средних

- Найдем новые центры кластеров



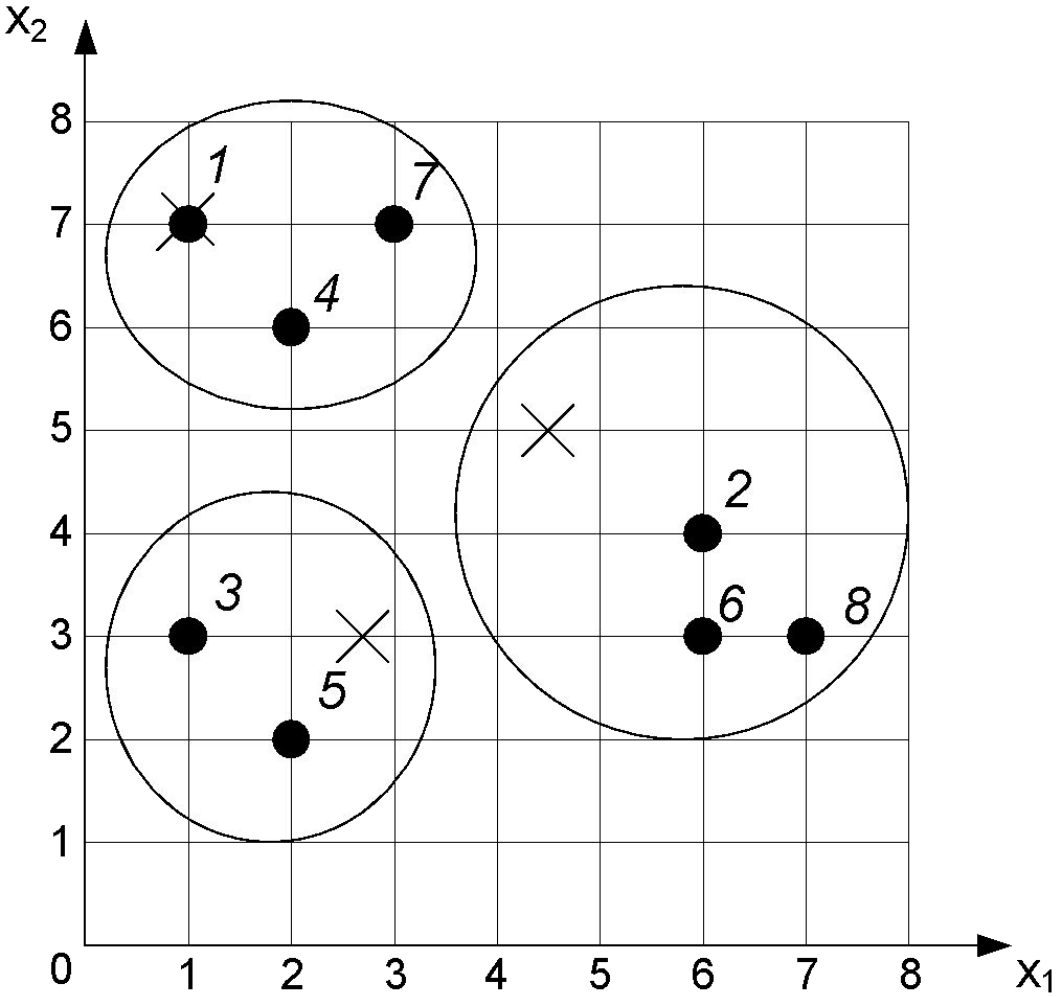
# Метод k-средних

- Найдем новые центры кластеров



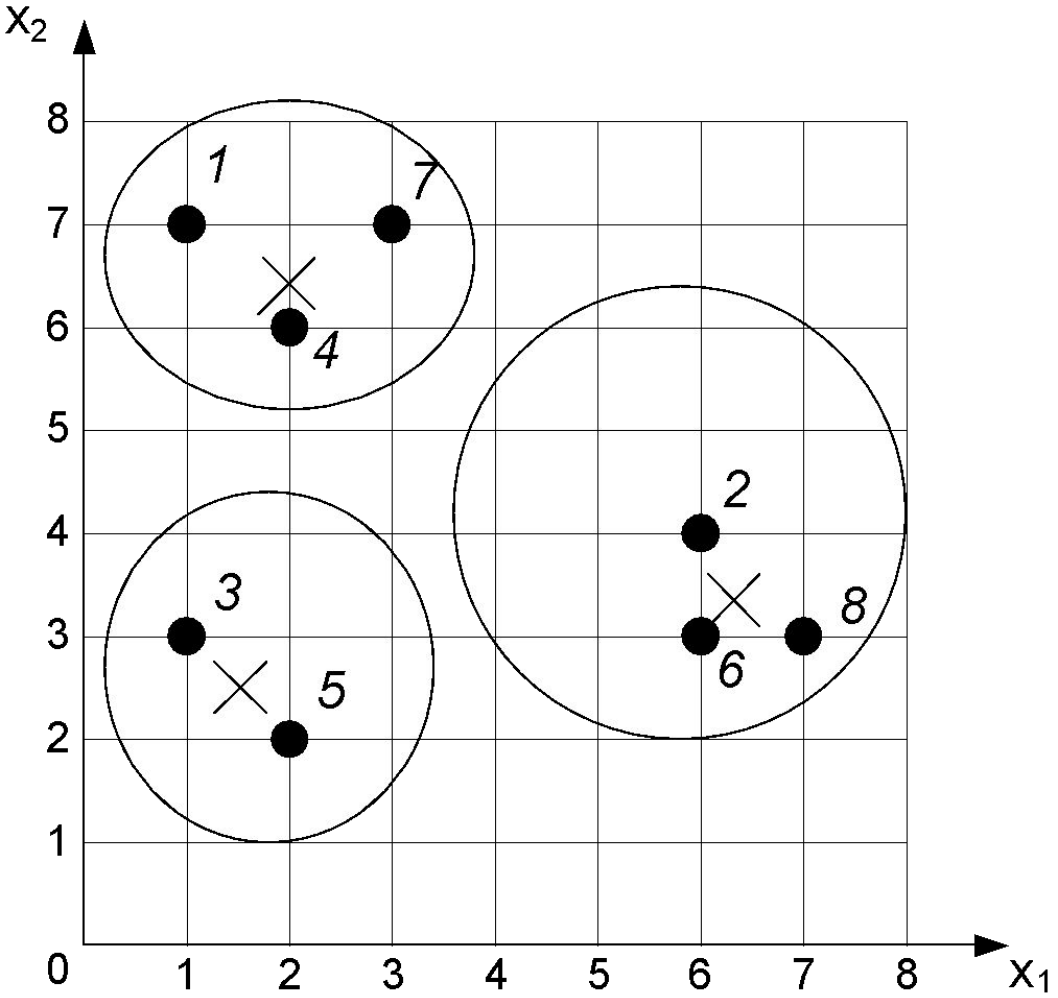
# Метод k-средних

- Разобьем все объекты по новым кластерам, относя каждый объект к кластеру с ближайшим центром



# Метод k-средних

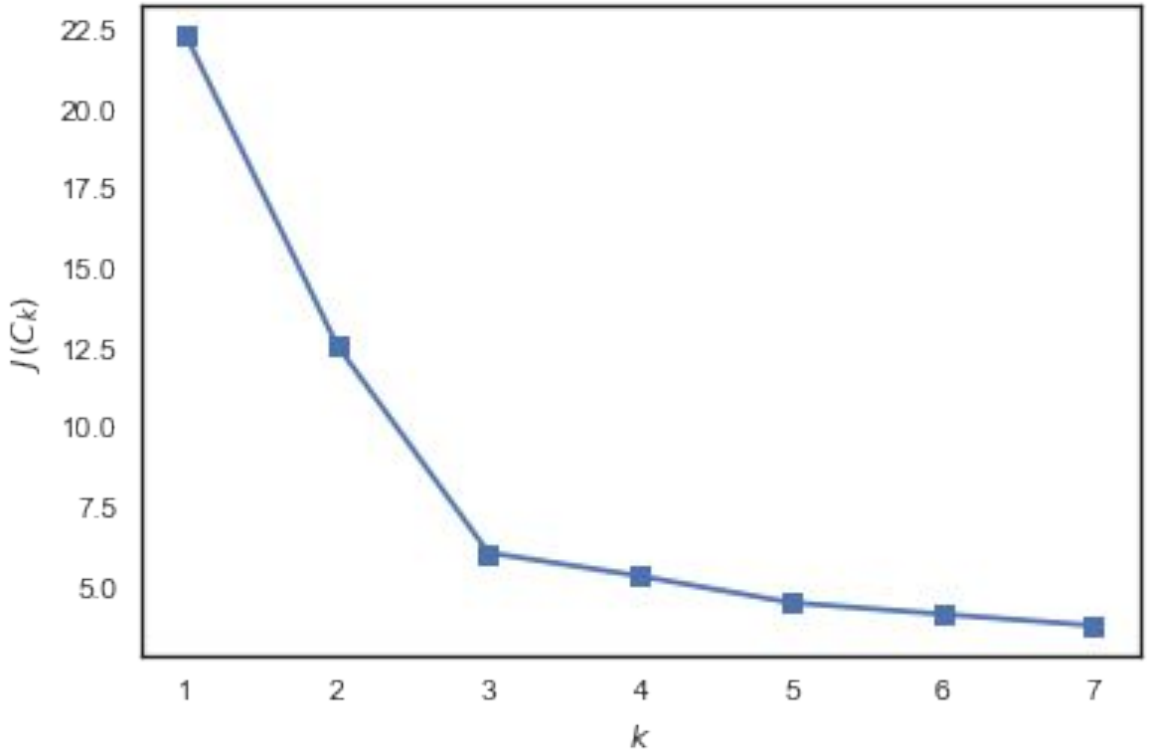
- Пересчитаем центры кластеров.
- Дальнейшая разбивка объектов по новым кластерам не меняет расположение центров



# Метод k-средних: определение k с помощью метода каменистой осыпи

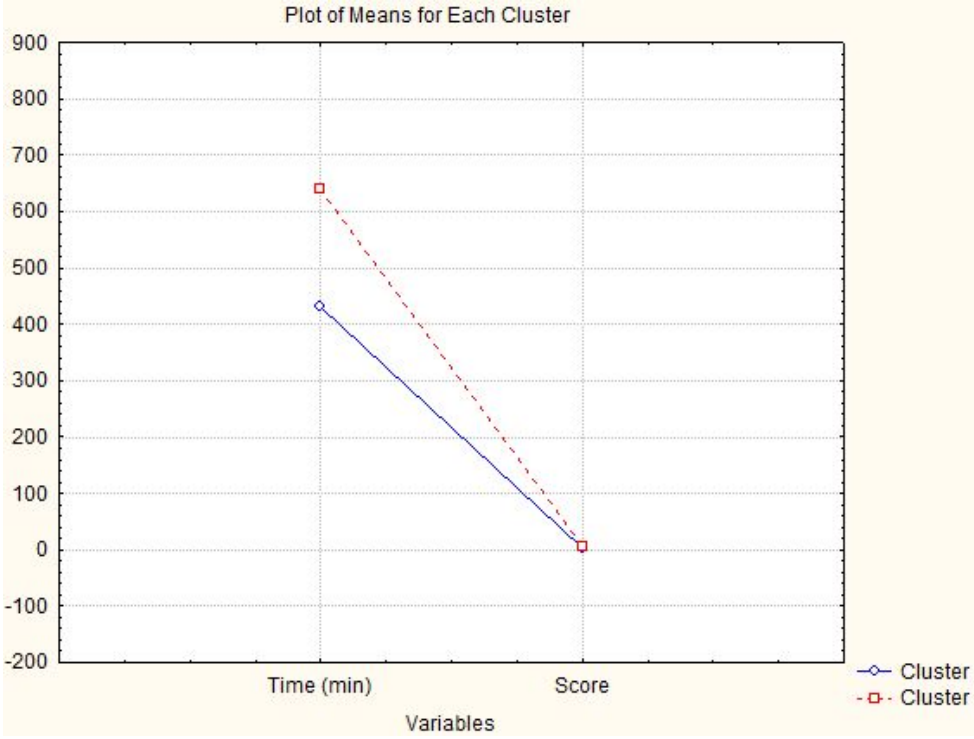
$$J(C) = \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \mu_k||^2 \to \min_C,$$

J (Ck) - сумма квадратов расстояний от точек до центроидов кластеров, к которым они относятся, k- количество кластеров

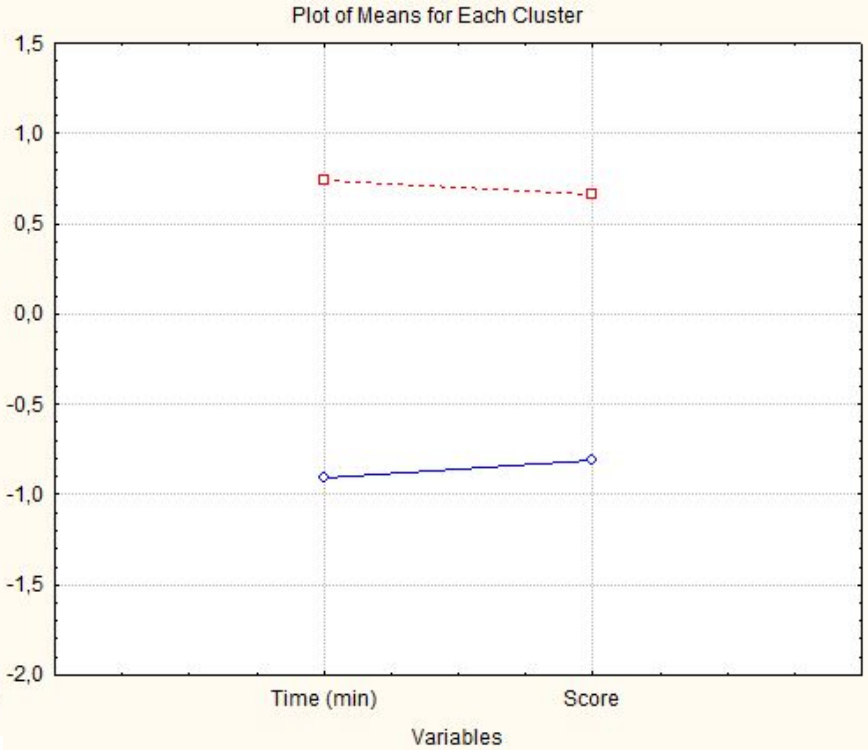


# График средних значений Признаков в кластерах

***До стандартизации***




***После***



# Иерархические методы

К иерархическим методам кластеризации относятся:

- агломеративный алгоритмы
- дивизимный алгоритмы

# Агломеративный метод

- В начале работы алгоритма все объекты являются отдельными кластерами
- На первом шаге наиболее похожие (близкие) два кластера объединяются в дин кластер
- На последующих шагах объединение продолжается до тех пор, пока все объекты не будут составлять один кластер
- На любом этапе объединение можно прервать, получив нужное число кластеров

# Вычисление расстояния между кластерами

1. **Метод ближайшего соседа** *(одиночная связь, Single linkage)*. Расстояние между двумя кластерами определяется расстоянием между двумя наиболее близкими объектами («ближайшими соседями») в различных кластерах.
2. **Метод наиболее удаленного соседа** *(полная связь, Complete linkage)***.** Расстояния между кластерами определяются наибольшим расстоянием между любыми двумя объектами в различных кластерах.
3. **Попарное среднее** *(Unweighted pair-group average)*. Расстояние между двумя различными кластерами вычисляется как среднее расстояние между всеми парами объектов в них.

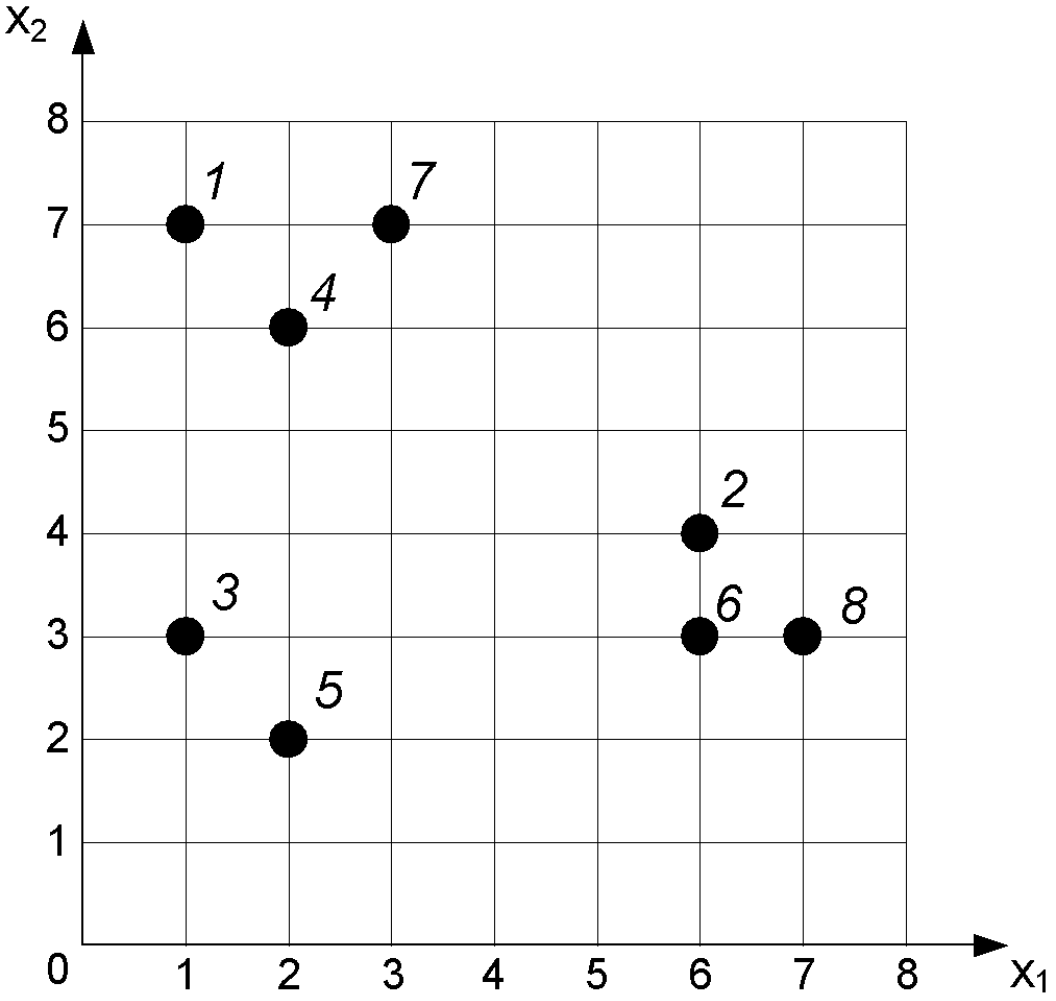
# Вычисление расстояния между кластерами

**4. Невзвешенный центроидный метод** (*Unweighted pair-group centroid)***.** В этом методе расстояние между двумя кластерами определяется как расстояние между их центрами.

**5. Метод Варда** (*Ward's method).* В качестве расстояния между кластерами берется прирост суммы квадратов расстояний объектов до центров кластеров, получаемый в результате их объединения В целом метод представляется очень эффективным, однако он стремится создавать кластеры малого размера.

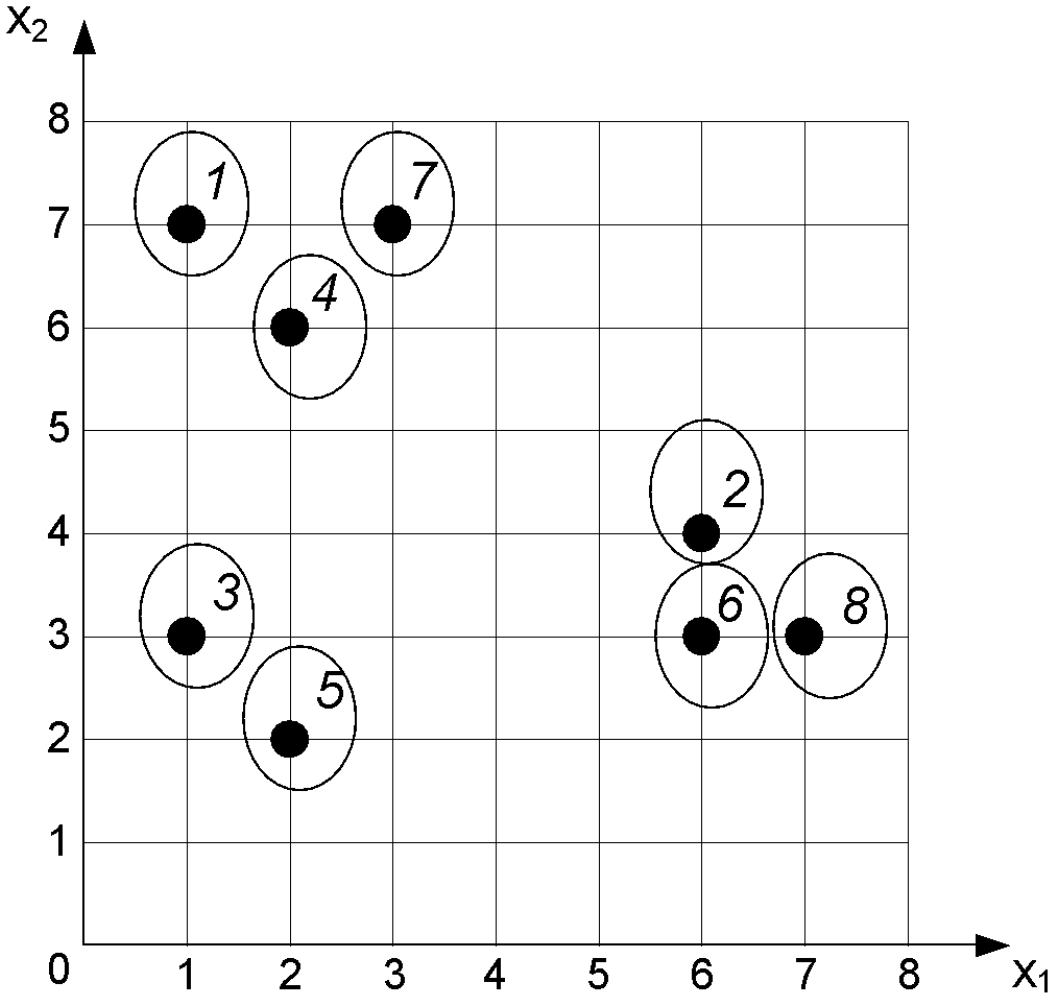
# Агломеративный метод

- ***Пример***.
  - Каждый объект формирует свой кластер



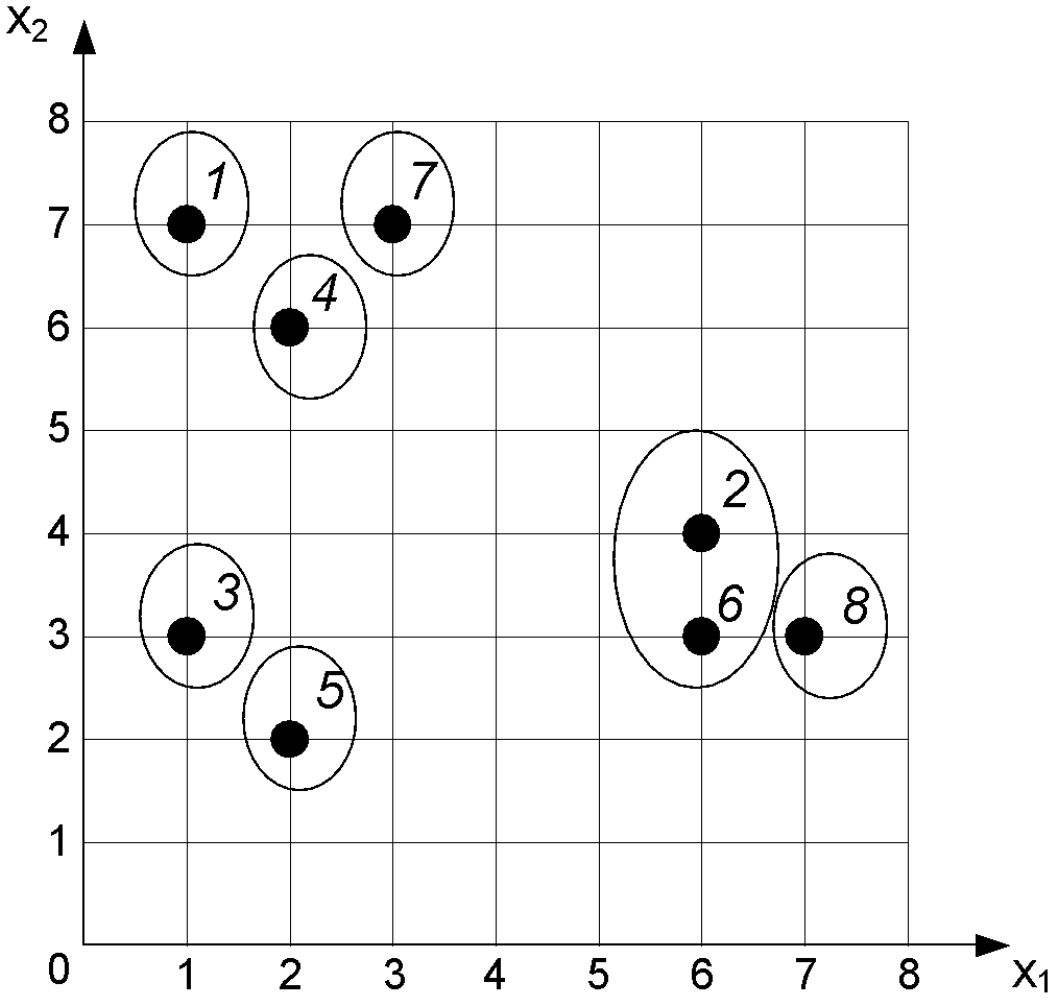
# Агломеративный метод

- Выбираем и объединяем два наиболее близких кластера



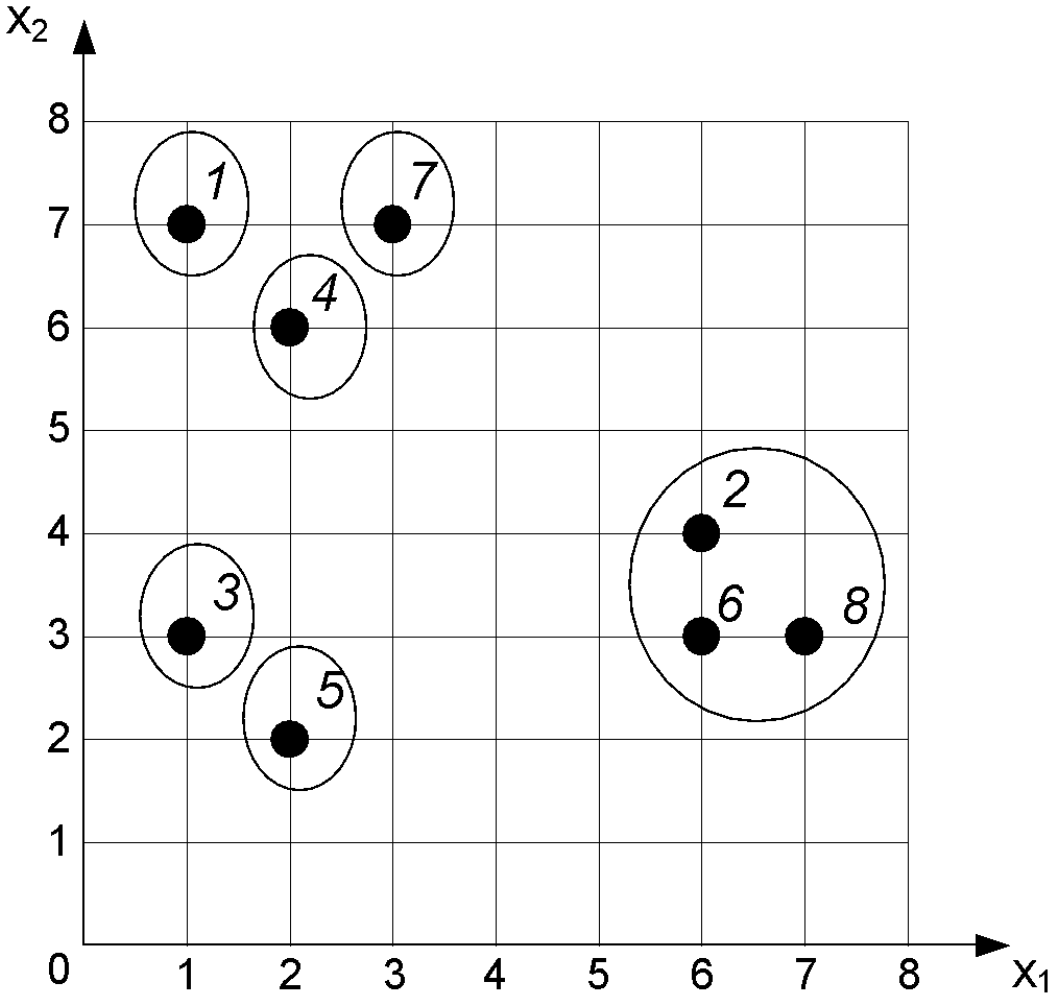
# Агломеративный метод

- Выбираем и объединяем два наиболее близких кластера



# Агломеративный метод

- Выбираем и объединяем два наиболее близких кластера



# Дивизимный метод

- На первом шаге все объекты помещаются в один кластер $C_1$
- Выбирается объект, у которого среднее значение расстояния до других объектов в этом кластере наибольшее:

$$\bar{d}\left(S_p\right)=\frac{1}{N_C}\cdot\sum_{i=1}^{Nc}d\left(S_p,S_i\right)$$

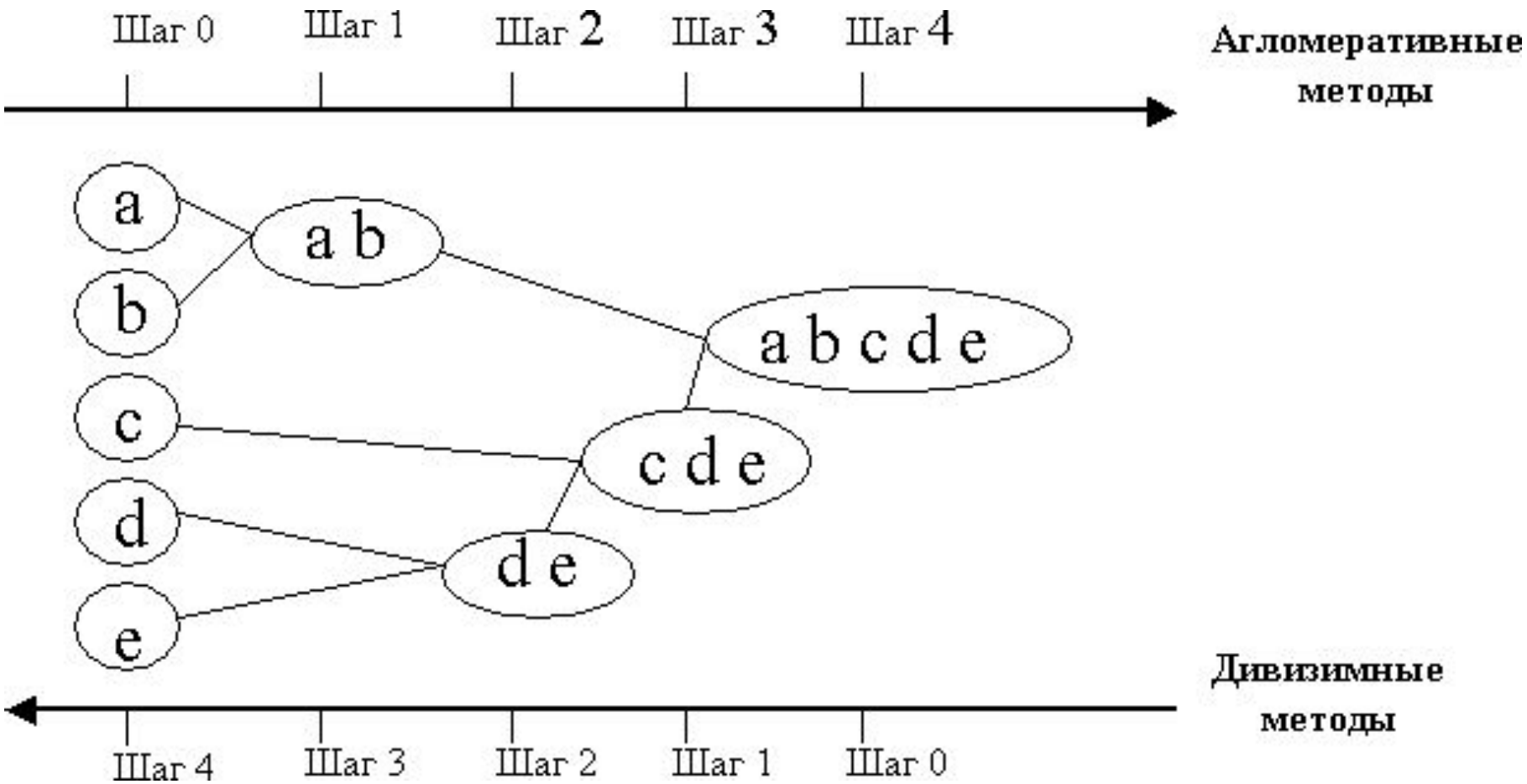
# Дивизимный метод

- Выбранный объект удаляется из кластера $C_1$ и формирует первый элемент второго кластера $C_2$
- На каждом последующем шаге объект в кластере $C_1$, для которого разность между средним расстоянием до объектов, находящихся в $C_2$ и средним расстоянием до объектов, остающихся в $C_1$, наибольшая, переносится в $C_2$

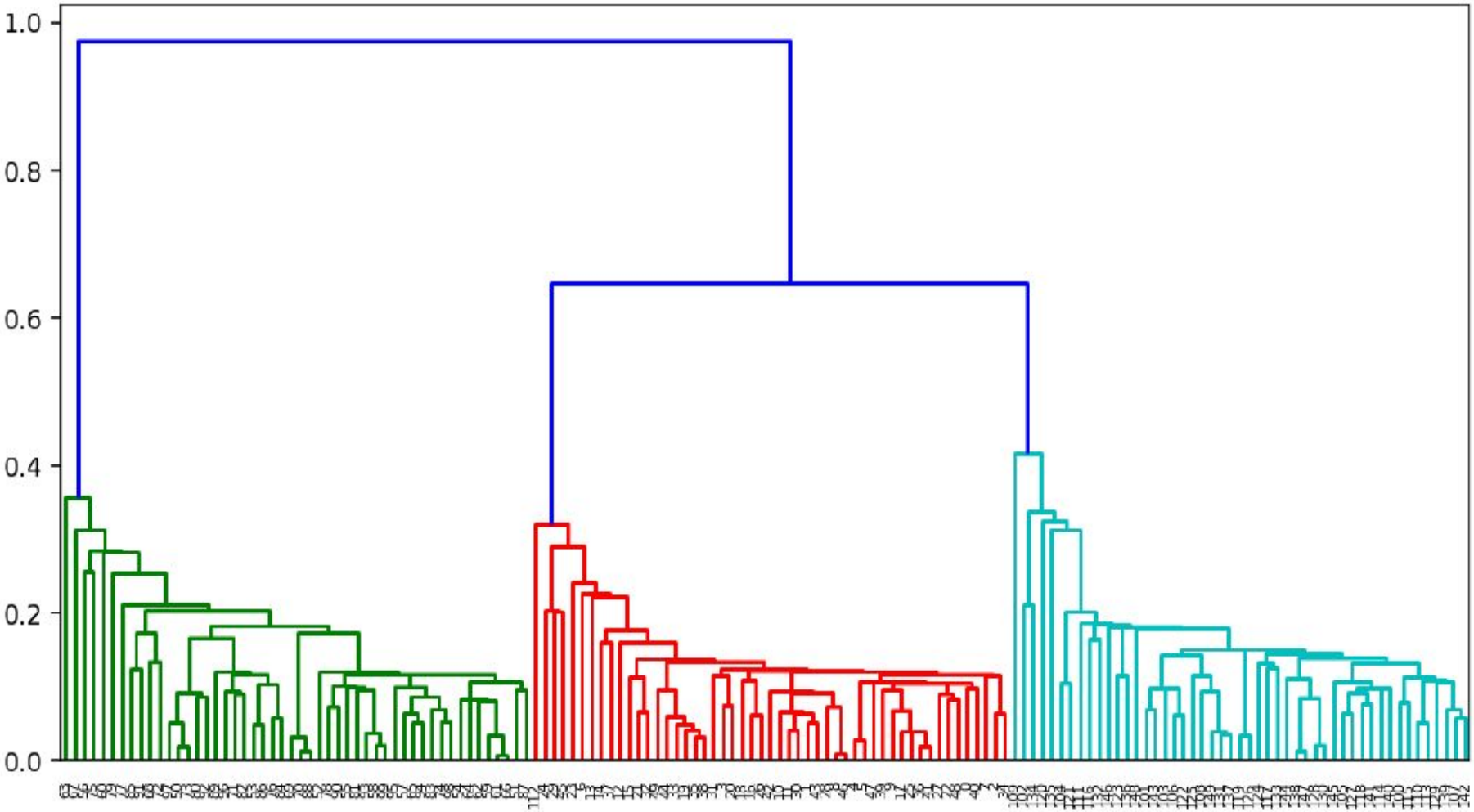
# Дивизимный метод

- В результате один кластер делится на два дочерних, один из которых расщепляется на следующем уровне иерархии
- Каждый последующий уровень применяет процедуру разделения к одному из кластеров, полученных на предыдущем уровне

# Иерархические методы



# ДЕНДРОГРАММА



# Метрики качества кластеризации

Коэффициент силуэта: $s = \frac{b - a}{\max(a, b)}$.

Здесь a — среднее внутрикластерное расстояние (то есть среднее расстояние между элементами, принадлежащими одному кластеру) , b— среднее межкластерное расстояние (среднее расстояние между элементами, принадлежащими разным кластерам).

Значение коэффициента силуэта лежит в диапазоне [−1,1]. Чем больше величина коэффициента, тем качественнее проведена кластеризация. Значения, близкие к -1, соответствуют плохим (неправильным) кластеризациям, значения, близкие к нулю, говорят о том, что кластеры пересекаются и накладываются друг на друга, значения, близкие к 1, соответствуют плотно сгруппированным кластерам.

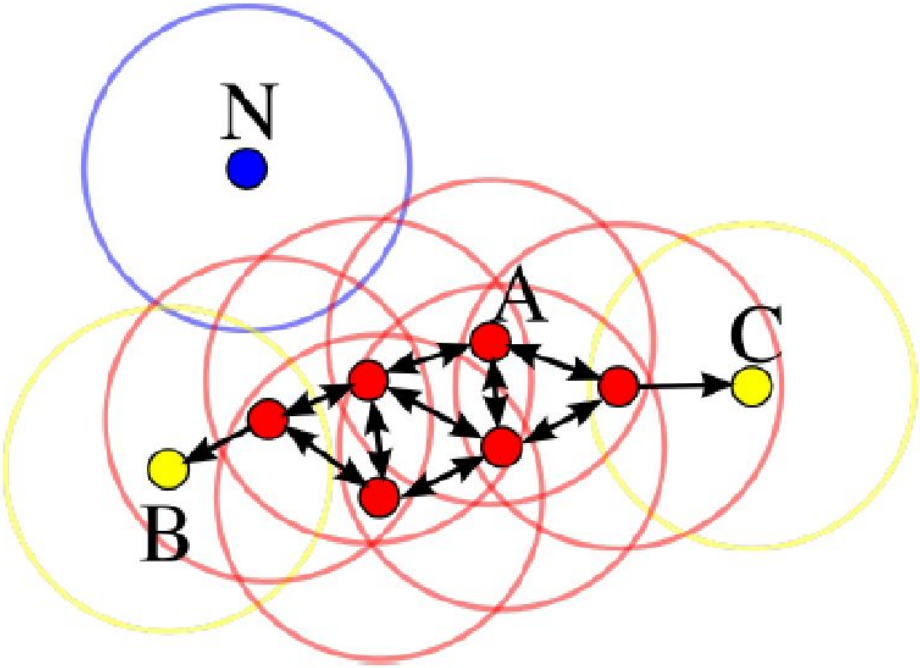
# Пример программы на Python

```
from sklearn import datasets
dataset = datasets.load_iris()
X = dataset.data
y = dataset.target
from sklearn.cluster import KMeans
model = KMeans(n_clusters=3).fit(X)
labels = model.labels_
from sklearn import metrics
metrics.silhouette_score(X, labels, metric='euclidean')
```

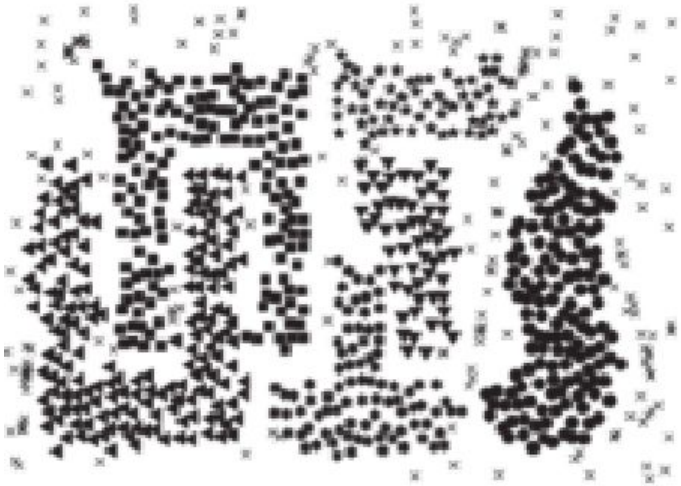
# DBSCAN

На вход алгоритму подаётся набор точек, параметры є (радиус окрестности) и m (минимальное число точек в окрестности).

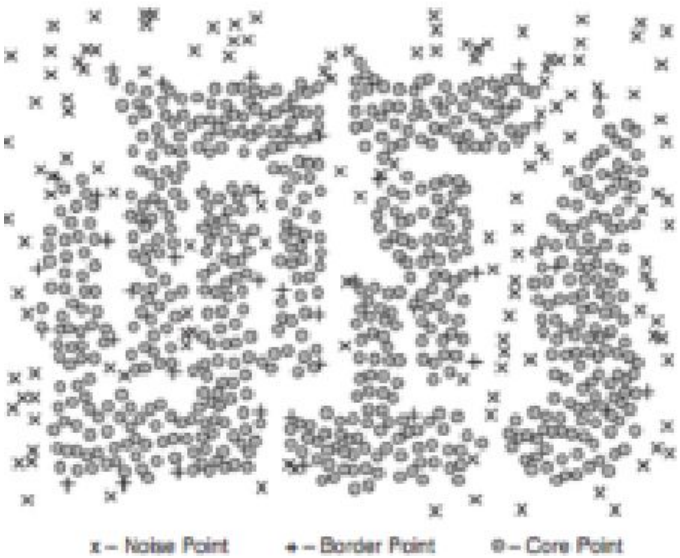
Основные, граничные и шумовые точки



# DBSCAN



(a) Clusters found by DBSCAN.




1: Пометить все точки, как основные, пограничные или шумовые.

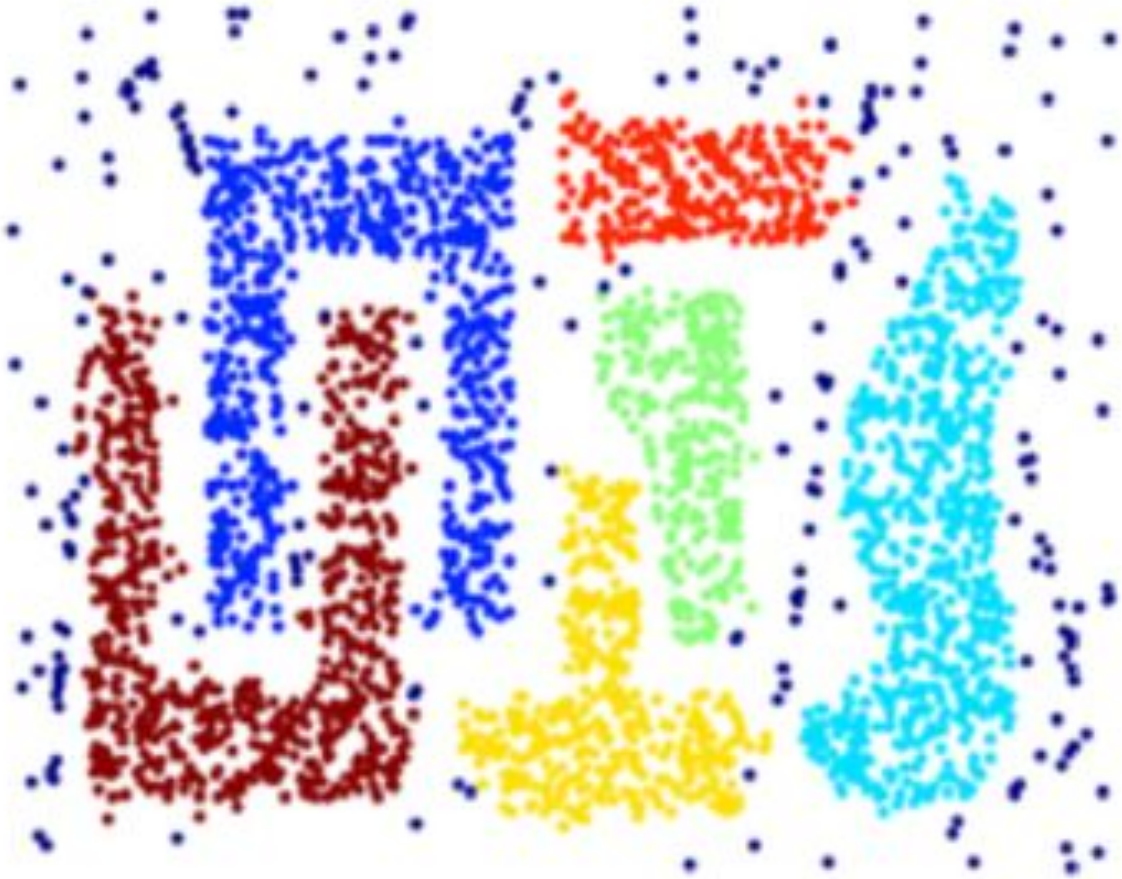
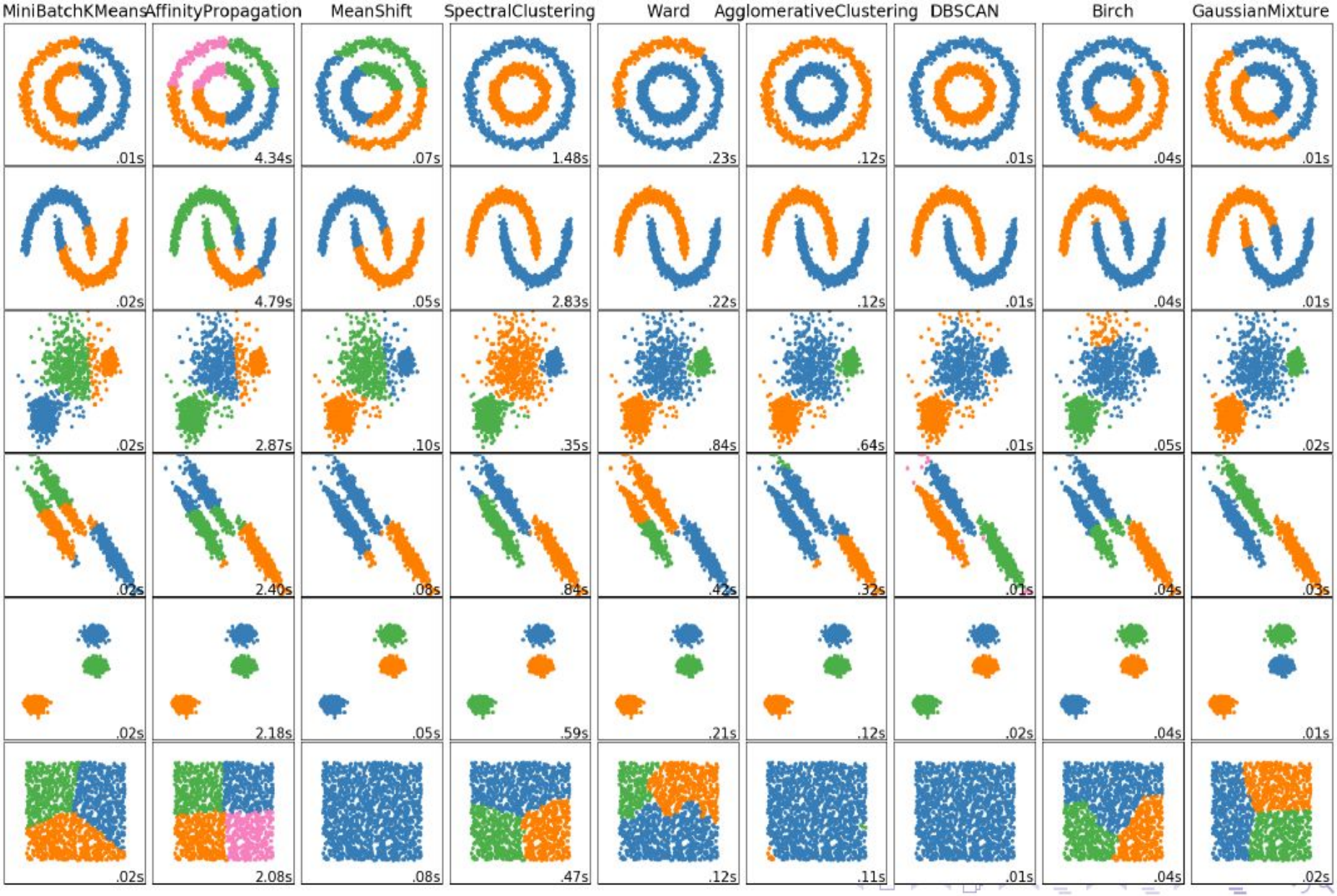
2: Отбросить точки шума.

3: Соединить все основные точки, находящиеся на расстоянии Eps радиуса одна от другой.

4: Объединить каждую группу соединенных основных точек в отдельный кластер.

5: Назначить каждую пограничную точку одному из кластеров, ассоциированных с ней основных точек.

# DBSCAN: результаты работы



# Примеры кластеризации с помощью различных методов