Обучение без учителя и кластеризация

Елена Кантонистова

Обучение с и без учителя

- Что такое обучение с учителем? Без учителя?

Обучение с учителем

- Обучение с учителем – это задачи, в которых алгоритм предсказывает целевую переменную (на исторических данных она также задана):

- классификация
- регрессия

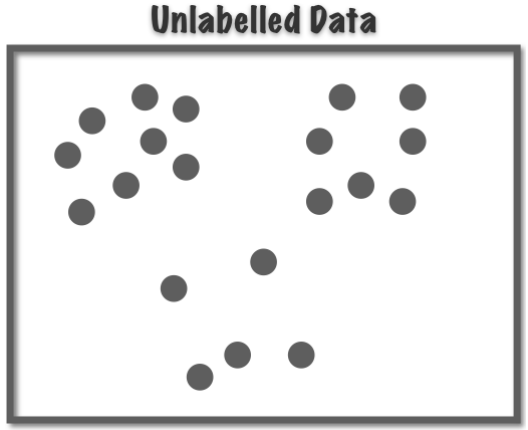
Обучение без учителя

- Обучение без учителем – это задачи, в которых алгоритм работает только с признаками объекта (целевой переменной нет или она не используется)

- кластеризация
- понижение размерности
- генерация изображений
- поиск аномалий

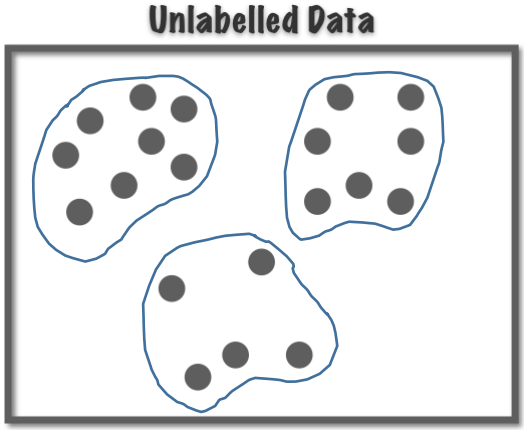
Кластеризация: K-means

K-Means



K-Means

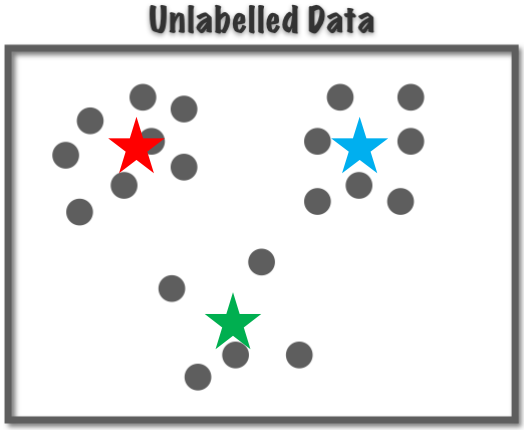
k=3



Вы видите 3 сгустка

K-Means

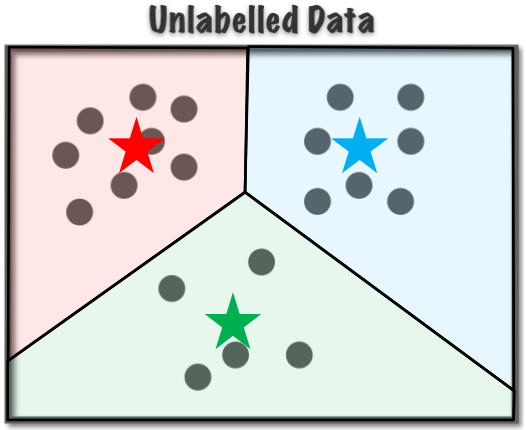
k=3



Опишем их центрами

K-Means

k=3



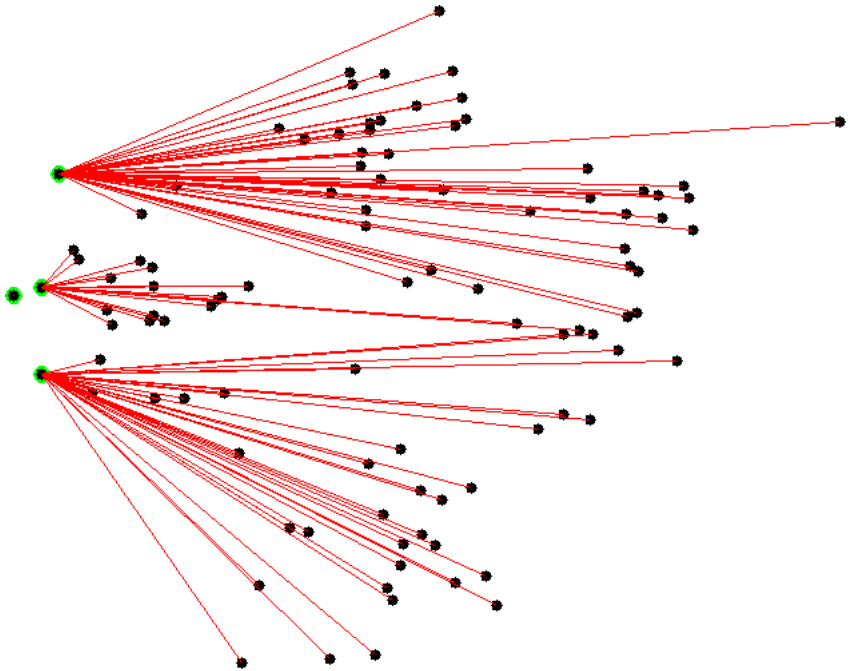
Каждая точка относится к ближайшему центру

https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c

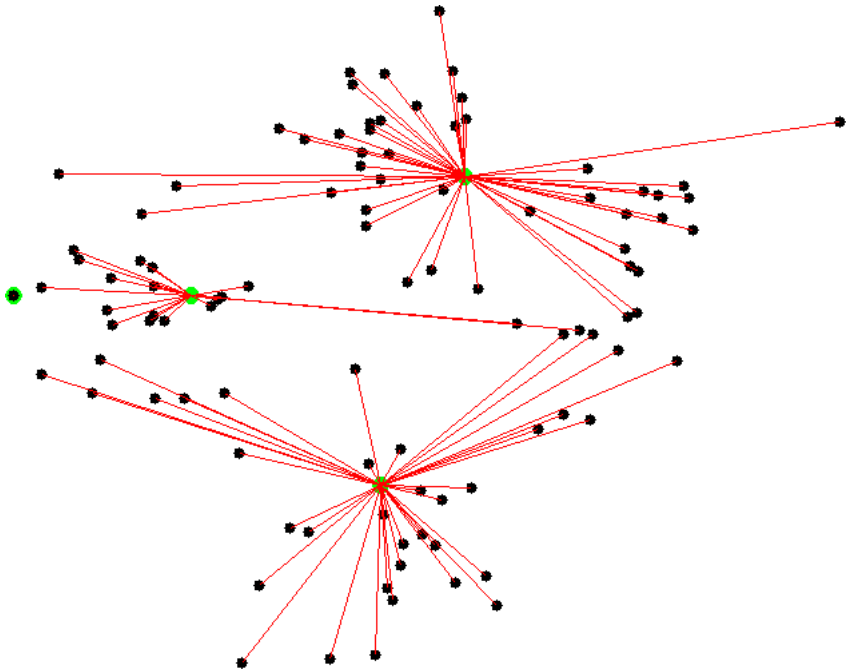
Кластеризация при помощи K-means

Возьмем
K=4
случайных
центра

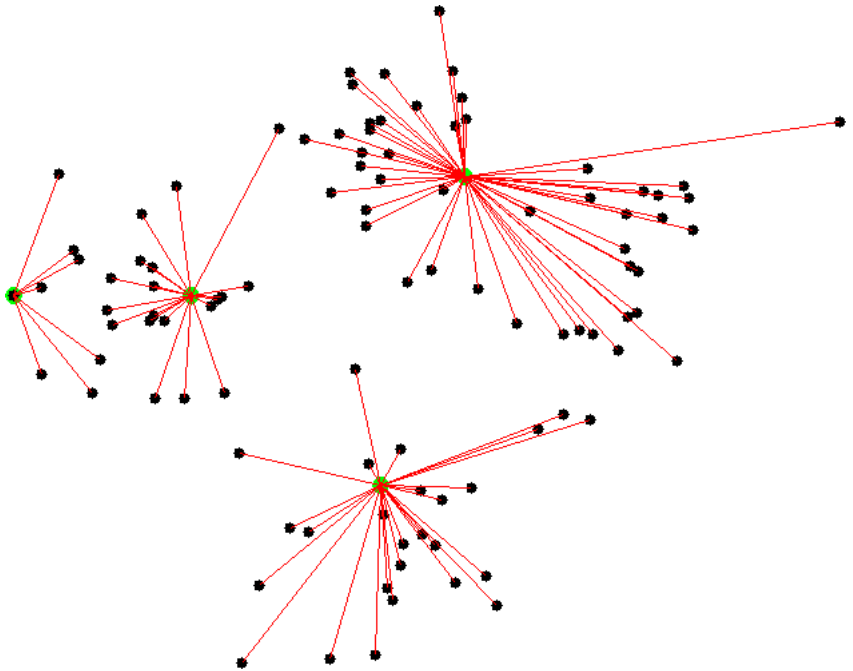
Для каждой точки находим ближайший центр



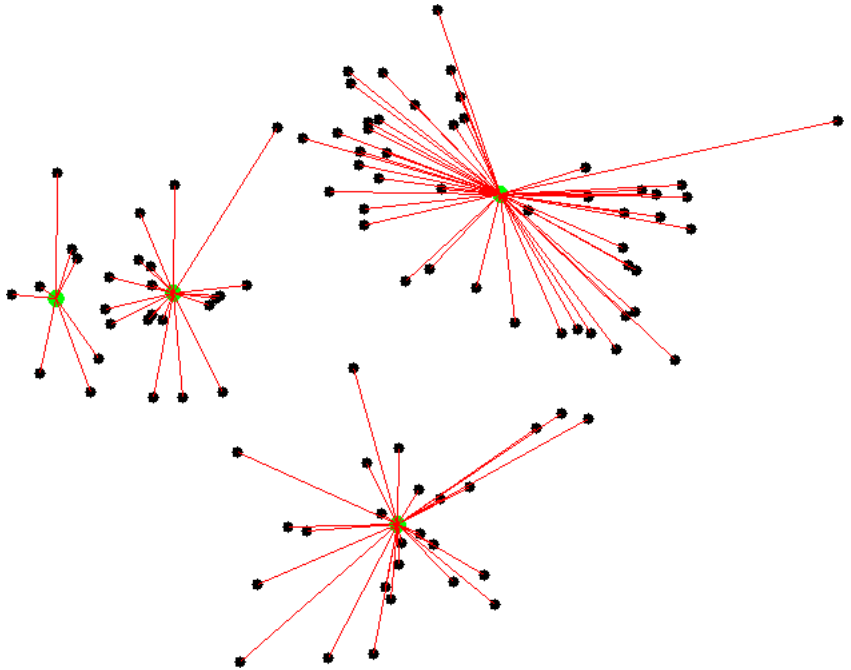
Пересчитываем центры



Опять ищем для каждой точки ближайший центр

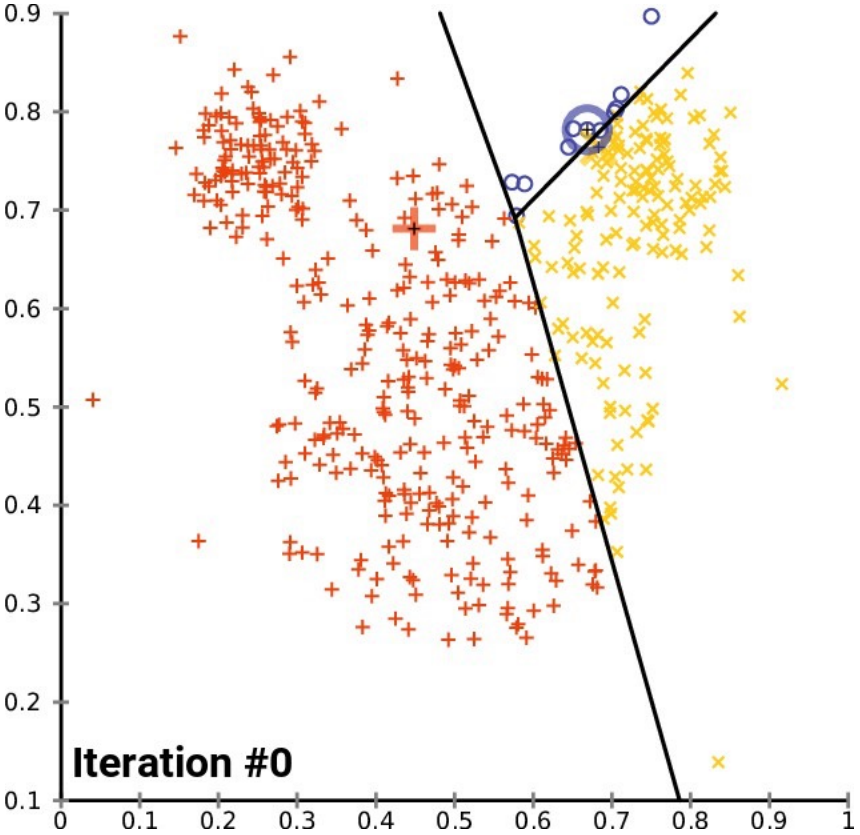


Опять пересчитываем центры



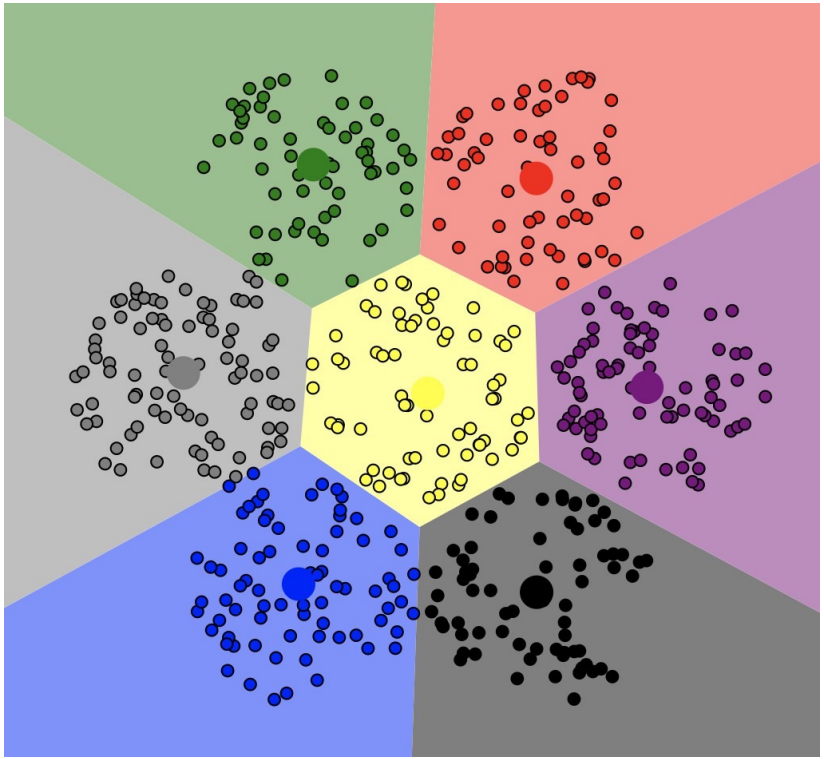
Анимация процесса K-means (K=4)

Неудачный пример



Демо K-Means

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/



Плюсы и минусы K-means

- **Плюсы:**
  - Очень простой алгоритм
  - Работает даже на больших данных

- **Минусы:**
  - Надо задавать число K руками
  - Не всегда находит кластеры правильно

Кластеризация: DBSCAN

DBSCAN: идея

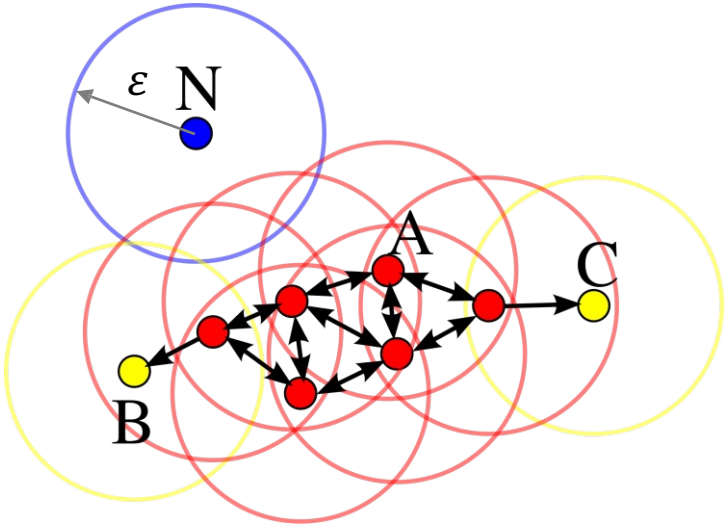
Хотим найти скопления точек, не зная заранее количество кластеров.

DBSCAN: идея

Хотим найти скопления точек, не зная заранее количество кластеров.

Параметры:

- ε – размер окрестности точки
- minPts – минимальное количество точек в ε-окрестности



DBSCAN: идея

Хотим найти скопления точек, не зная заранее количество кластеров.

Параметры:

- ε – размер окрестности точки
- minPts – минимальное количество точек в ε-окрестности

Все точки делятся на 3 типа:

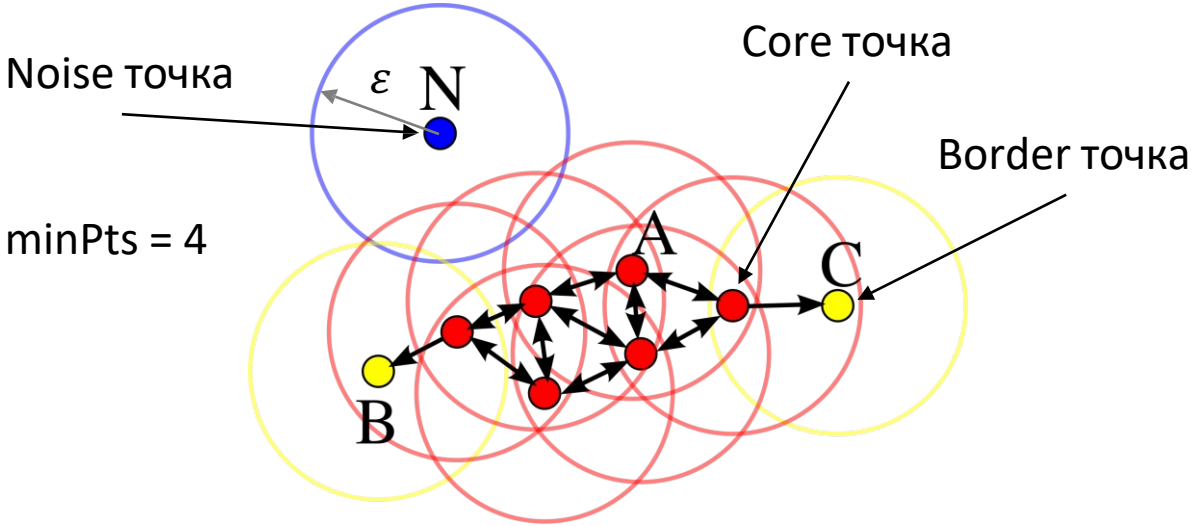
- **Core** точки – от minPts соседей в ε-окрестности
- **Border** точки – не Core точки, но достижимы из Core точек
- **Noise** точки – все остальные, меньше minPts соседей в ε-окрестности

DBSCAN: идея

Хотим найти скопления точек, не зная заранее количество кластеров.

Параметры:

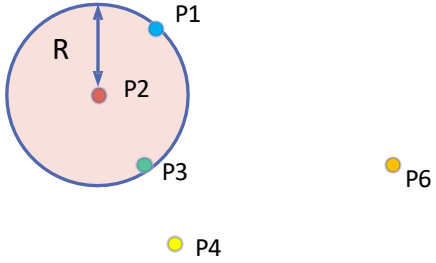
- ε – размер окрестности точки
- minPts – минимальное количество точек в ε-окрестности



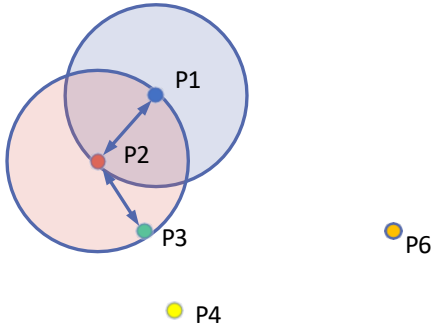
DBSCAN: алгоритм

- Для следующей произвольной точки ищем соседей в ε-окрестности
- Если их как минимум minPts, начинаем поиск связной компоненты из этой Core-точки
- Иначе помечаем точку как Noise, она может быть подключена позднее к какой-то Core-точке как Border точка

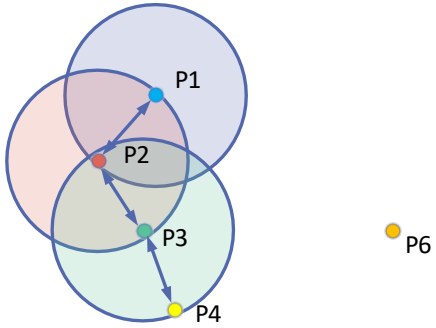
DBSCAN: Example (minPts = 2)



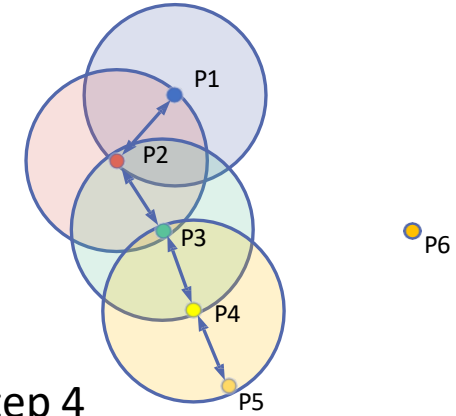
Step 1



Step 2

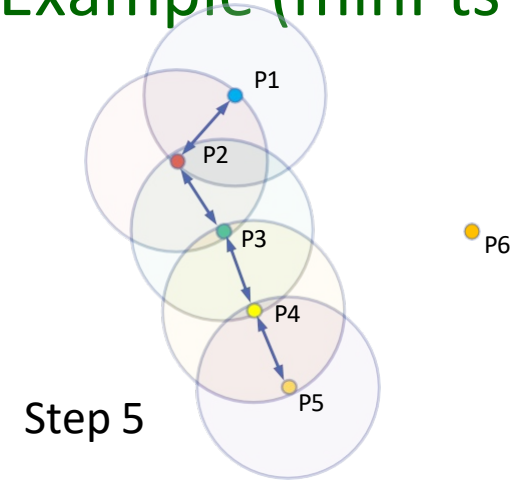


Step 3

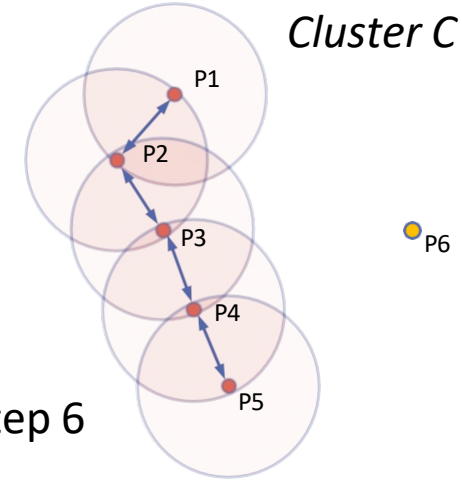


Step 4

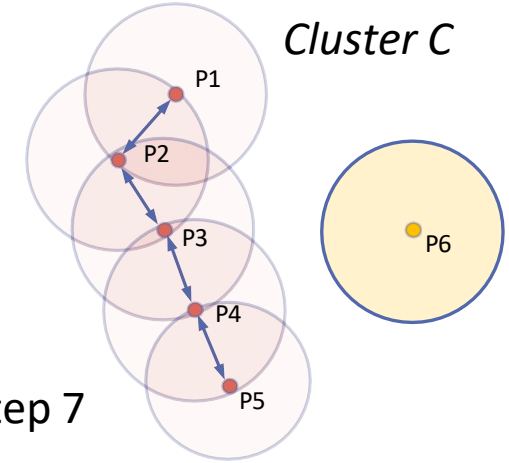
DBSCAN: Example (minPts = 2)



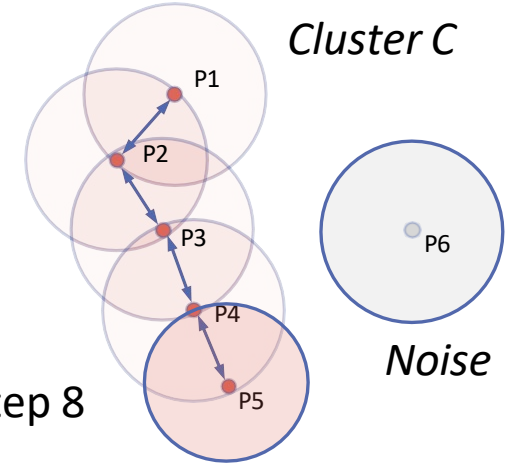
Step 5



Step 6

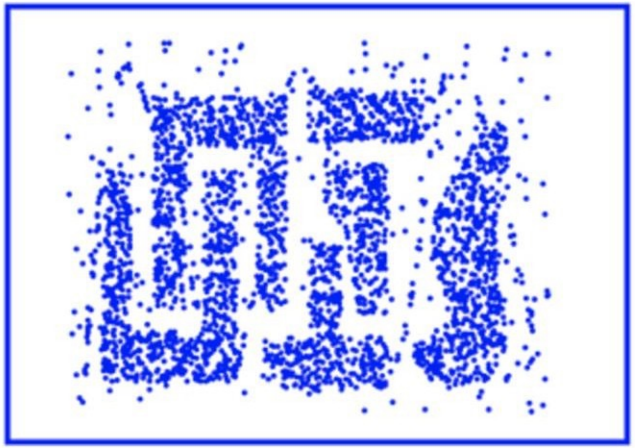


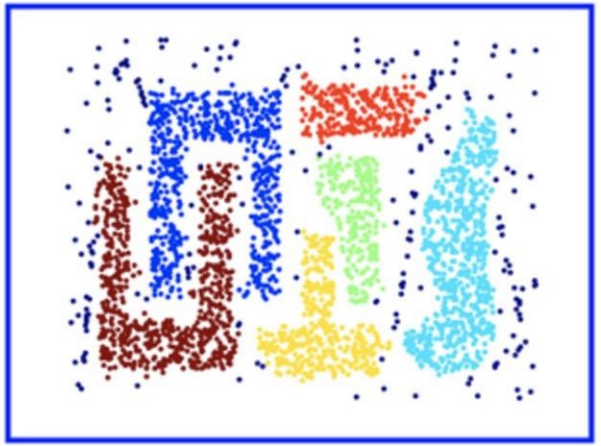
Step 7



Step 8

DBSCAN: пример

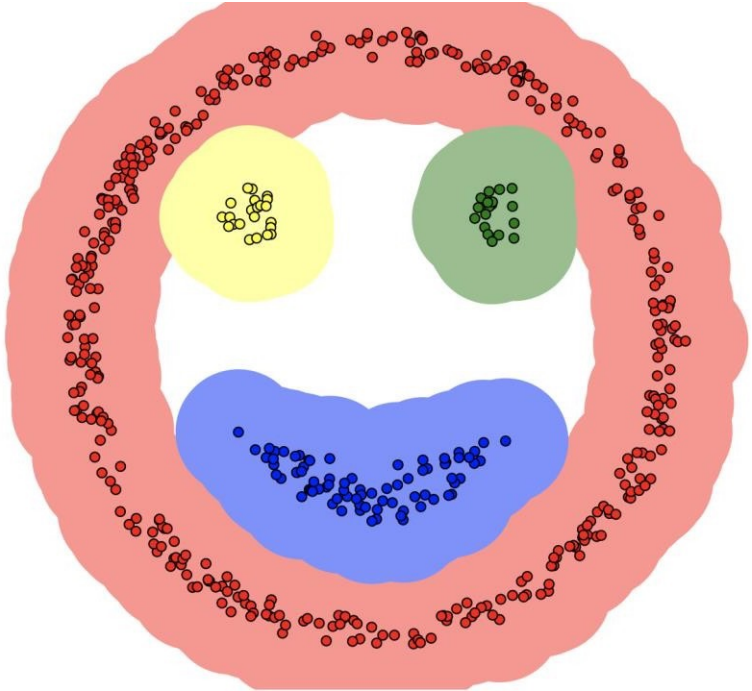




https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html

Демо DBSCAN

- https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/



epsilon = 1.98
minPoints = 4

Резюме: DBSCAN

Плюсы:

- Не нужно задавать кол-во кластеров
- Кластеры могут быть любой формы
- Может работать с шумными данными

Минусы:

- Долго работает на больших данных
- Чувствителен к выбору гипер-параметров