

# Лекция 3

## Кластеризация

---

Екатерина Тузова

# Разбор летучки

---

## Мотивирующий пример

---

# Мотивирующий пример



```
In [4]: pokemons.head()
```

```
Out[4]:
```

	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False

# Постановка задачи кластеризации

Кластеризация – задача разделения объектов одной природы на несколько групп так, чтобы объекты в одной группе обладали одним и тем же свойством.

# Постановка задачи кластеризации

Кластеризация – задача разделения объектов одной природы на несколько групп так, чтобы объекты в одной группе обладали одним и тем же свойством.

Кластеризация – это обучение без учителя.

# Постановка задачи кластеризации

$X$  – пространство объектов

$\rho : X \times X \rightarrow [0, \infty)$  – функция расстояния между объектами

Найти:

$Y$  – множество кластеров

$a : X \rightarrow Y$  – алгоритм кластеризации



# Гипотеза компактности

---

# Степени свободы в постановке задачи

---

- Критерий качества кластеризации

# Степени свободы в постановке задачи

- Критерий качества кластеризации
- Число кластеров неизвестно заранее

# Степени свободы в постановке задачи

- Критерий качества кластеризации
- Число кластеров неизвестно заранее
- Результат кластеризации существенно зависит от метрики

## Цели кластеризации

---

- Сократить объём хранимых данных

- Сократить объём хранимых данных
- Выделить нетипичные объекты

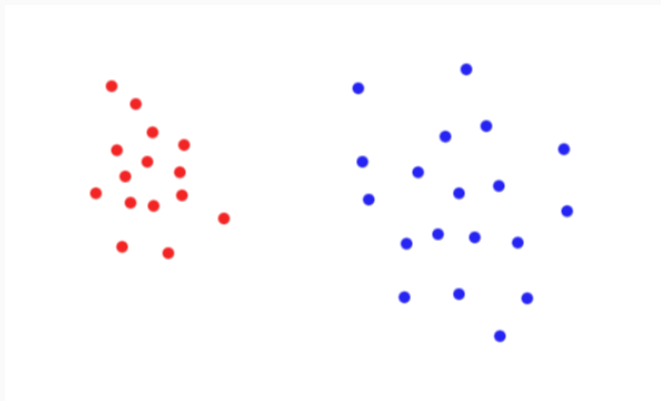


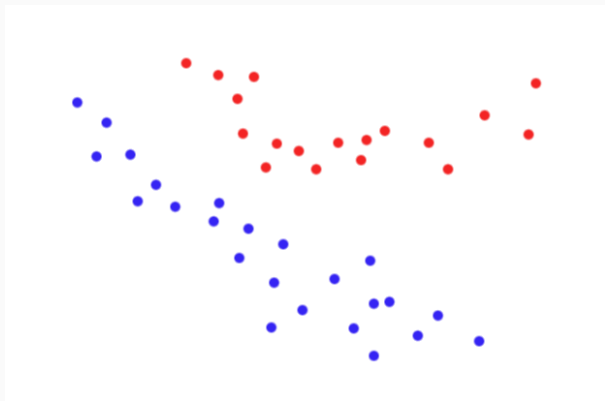
- Сократить объём хранимых данных
- Выделить нетипичные объекты
- Упростить дальнейшую обработку данных

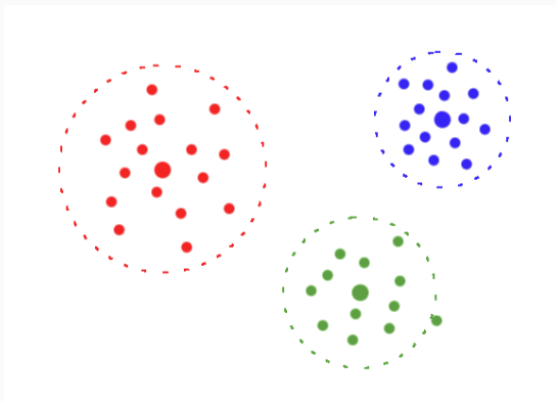
- Сократить объём хранимых данных
- Выделить нетипичные объекты
- Упростить дальнейшую обработку данных
- Построить иерархию множества объектов

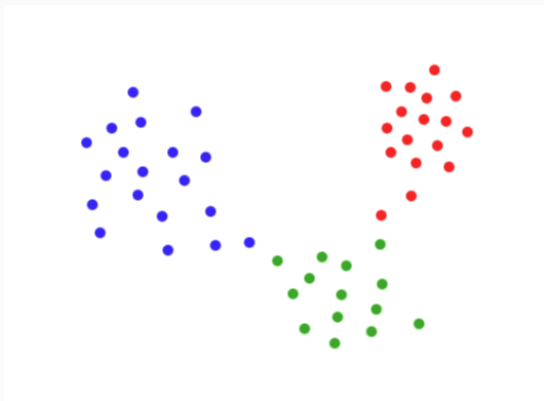
Какие бывают кластеры?

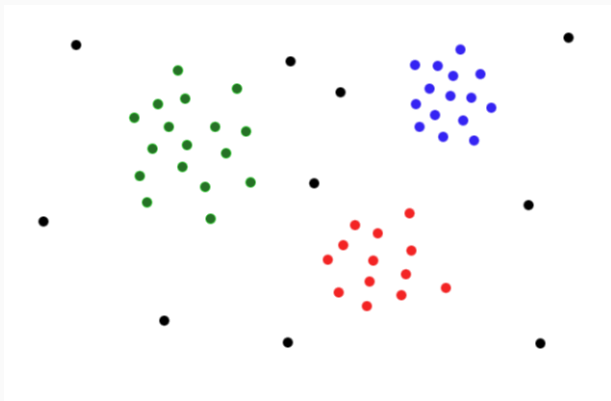
---



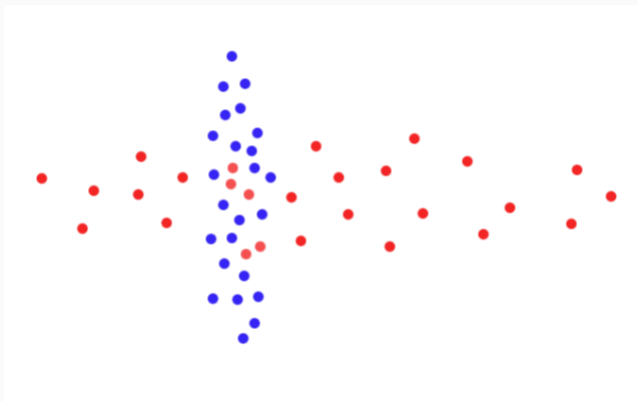








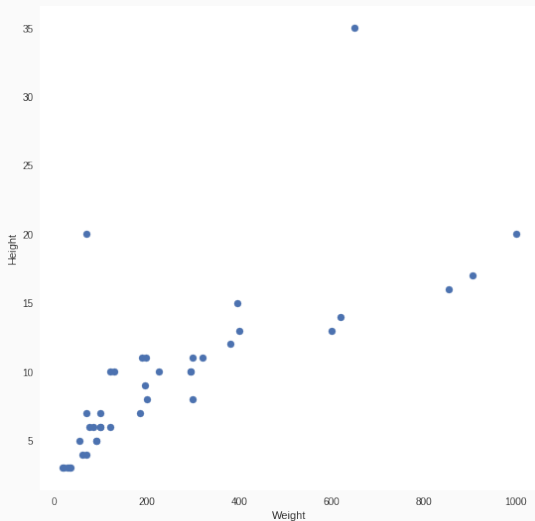




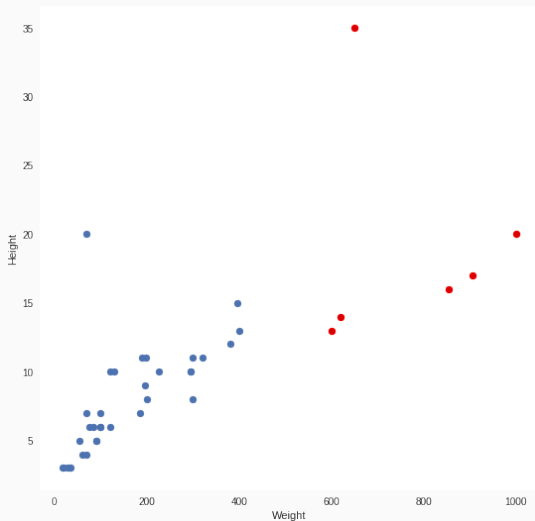
# Чувствительность к выбору метрики

---

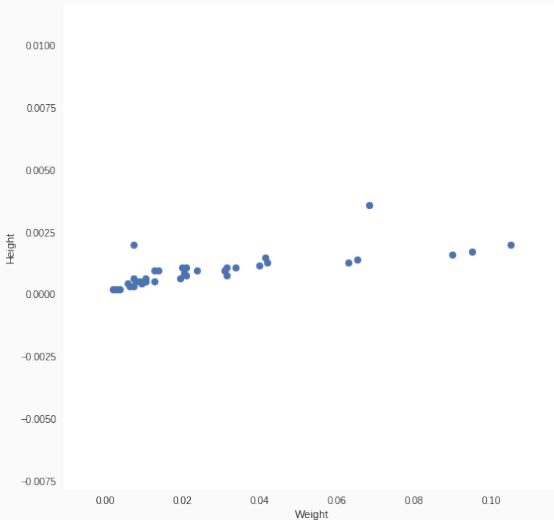
# Чувствительность к выбору метрики



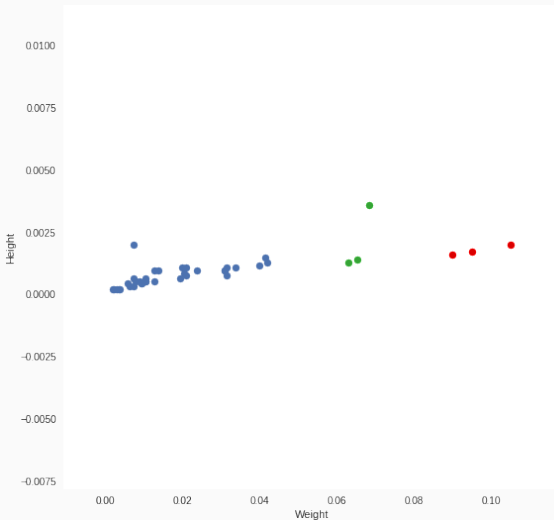
# Чувствительность к выбору метрики



# Чувствительность к выбору метрики



# Чувствительность к выбору метрики



# Оценка качества кластеризации

---

**Идея:** Минимизировать среднее внутрикластерное расстояние и при этом максимизировать среднее межкластерное расстояние.



**Идея:** Минимизировать среднее внутрикластерное расстояние и при этом максимизировать среднее межкластерное расстояние.

$$\frac{\sum_{a(x_i)=a(x_j)} \rho(x_i, x_j)}{\sum_{a(x_i)=a(x_j)} 1} \rightarrow \min$$

**Идея:** Минимизировать среднее внутрикластерное расстояние и при этом максимизировать среднее межкластерное расстояние.

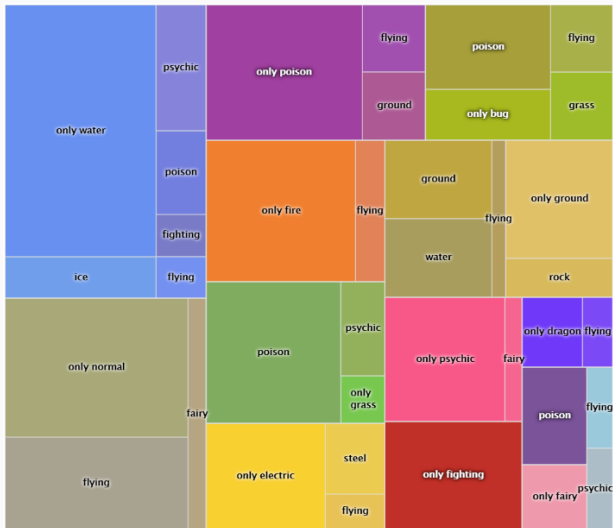
$$\frac{\sum_{a(x_i)=a(x_j)} \rho(x_i, x_j)}{\sum_{a(x_i)=a(x_j)} 1} \rightarrow \min$$

$$\frac{\sum_{a(x_i) \neq a(x_j)} \rho(x_i, x_j)}{\sum_{a(x_i) \neq a(x_j)} 1} \rightarrow \max$$

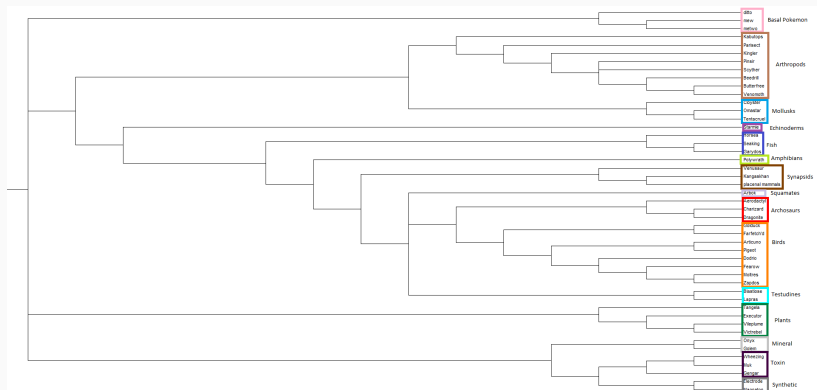
# Визуализация кластеров

---

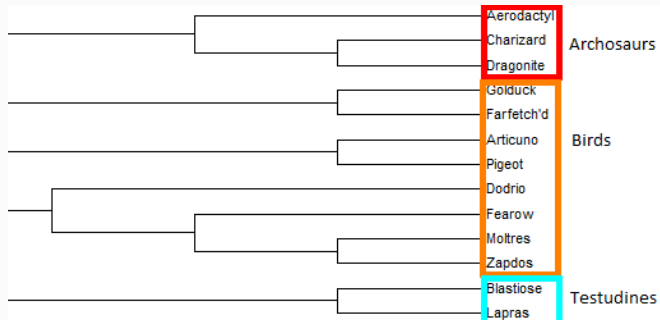
# Диаграмма вложения



# Дендрограмма



# Дендрограмма



Может ли так случиться, что дендрограмма имеет самопересечения?

Кластеризация монотонна, если на каждом шаге расстояние  $\rho$  между объединяемыми кластерами не уменьшается.

$$\rho_2 \leq \rho_3 \leq \dots \leq \rho_l$$



- Статистические
- Графовые
- Иерархические

# Статистические алгоритмы

---

Идея: минимизировать меру ошибки

$$E(X, C) = \sum_{i=1}^n \|x_i - \mu_i\|^2$$

$\mu_i$  – ближайший к  $x_i$  центр кластера

1 **function** KMEANS( $k$ )

2     Инициализировать  $\mu_i, i = 1 \dots k$

3     **repeat**[пока  $c_i$  не перестанет меняться]

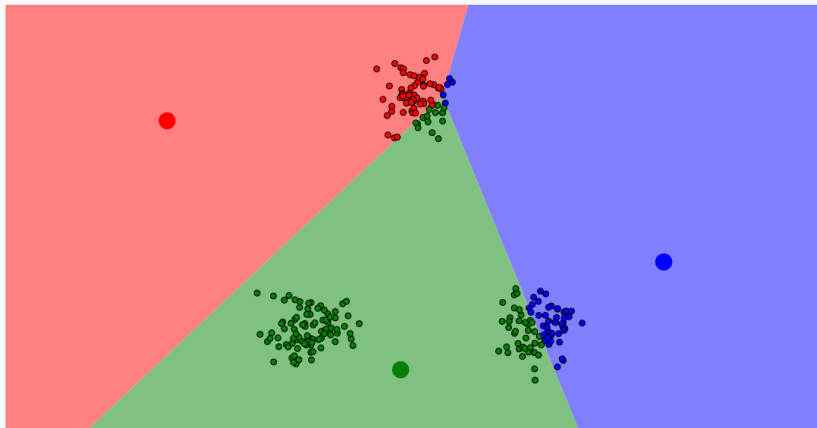
4          $c_i = \arg \min_{c \in C} \rho(x_i, \mu_c) \quad i = 1, \dots, l$

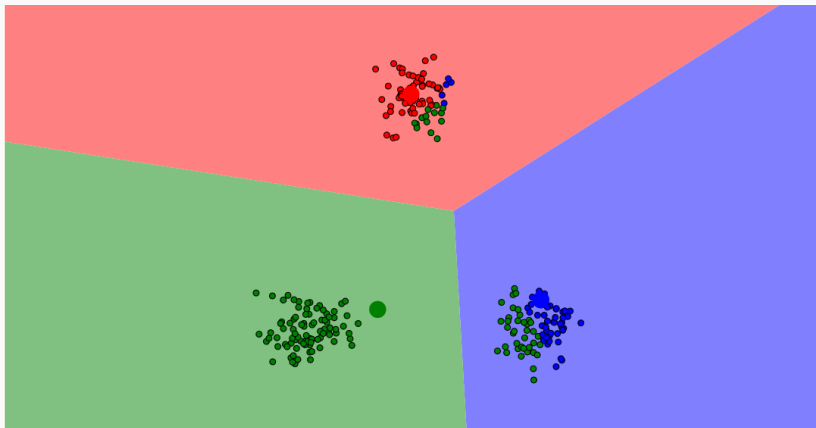
5          $\mu_c = \frac{\sum_{c_i=c} f_j(x_i)}{\sum_{c_i=c} 1} \quad j = 1, \dots, n, c \in C$

$\mu_c$  – новое положение центров кластеров

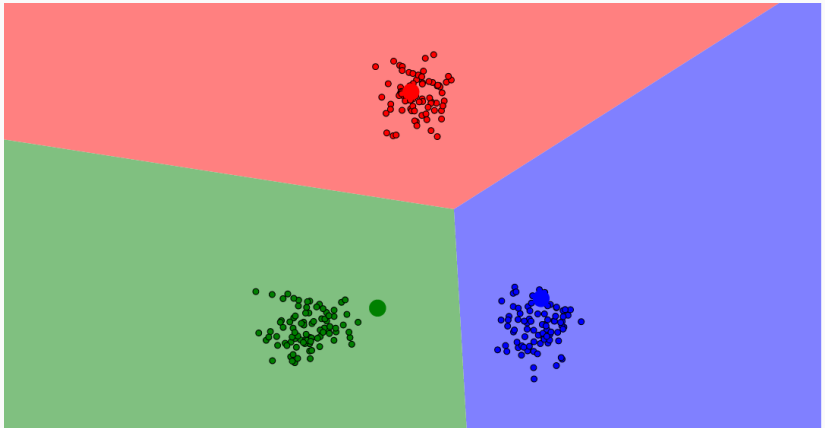
$c_i$  – принадлежность  $x_i$  к кластеру

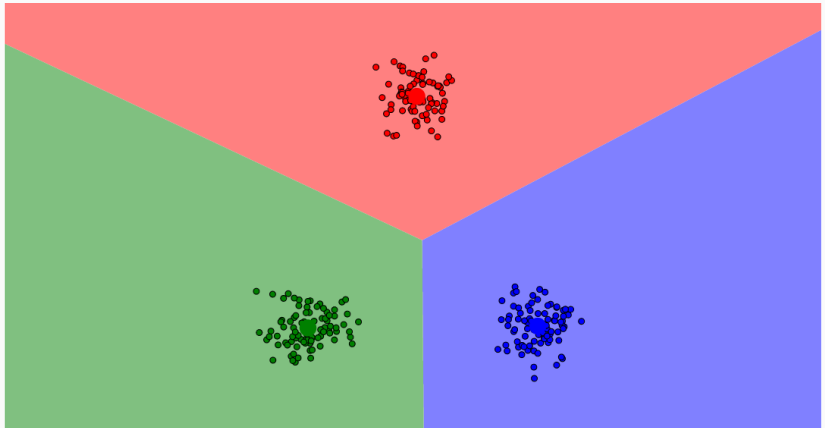
$\rho(x_i, \mu_c)$  – расстояние от  $x_i$  до центра кластера  $\mu_c$





# Метод $k$ -средних

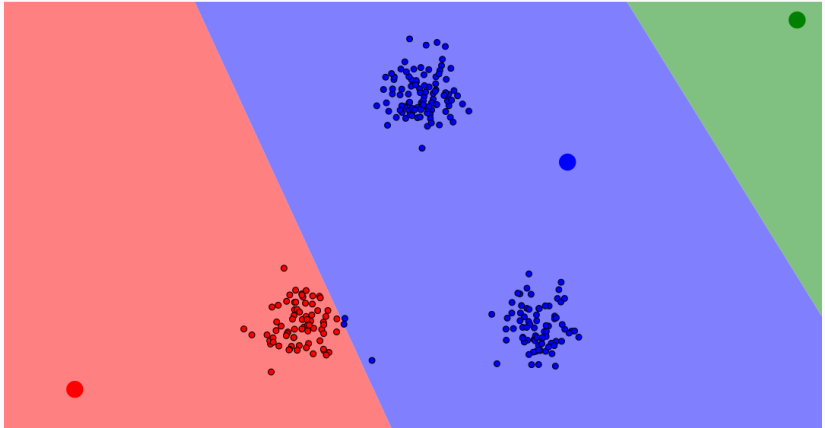




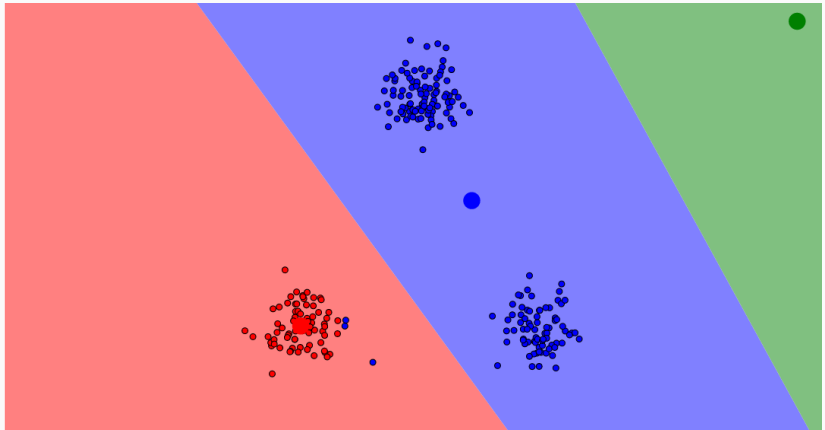


- Чувствительность к начальному выбору  $\mu_c$
- Необходимость задавать  $k$

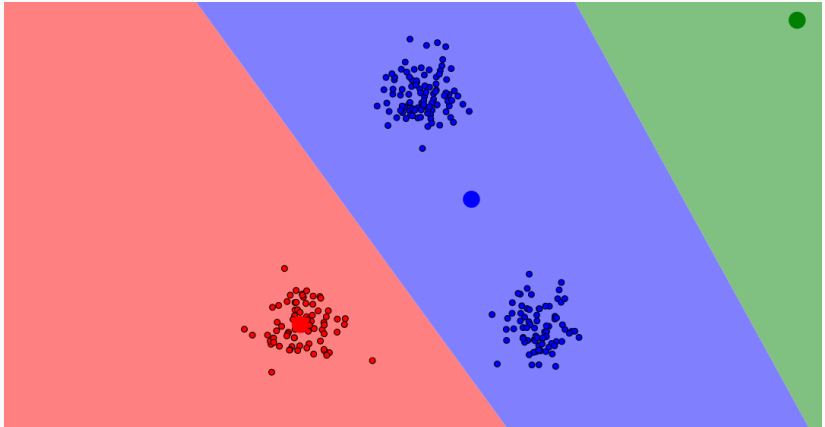
# Чувствительность к начальному выбору $\mu_c$



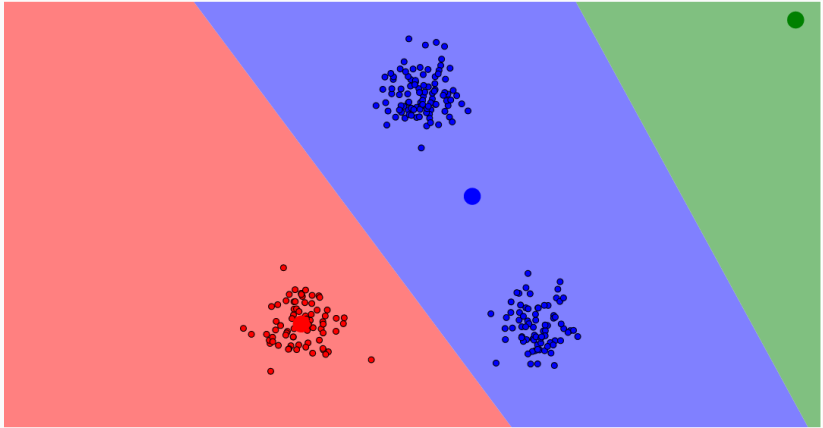
# Чувствительность к начальному выбору $\mu_c$



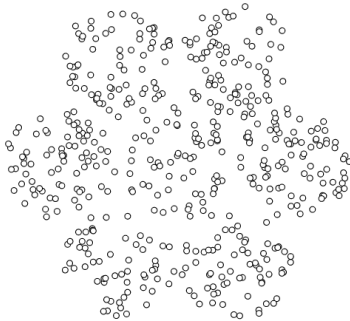
# Чувствительность к начальному выбору $\mu_c$



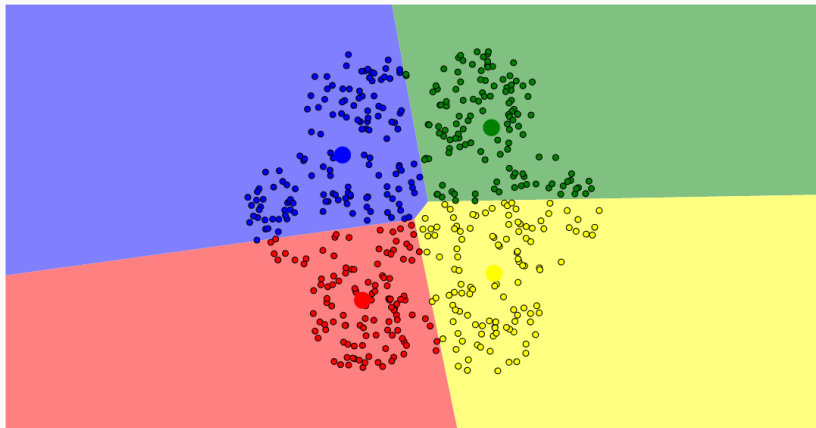
# Чувствительность к начальному выбору $\mu_c$



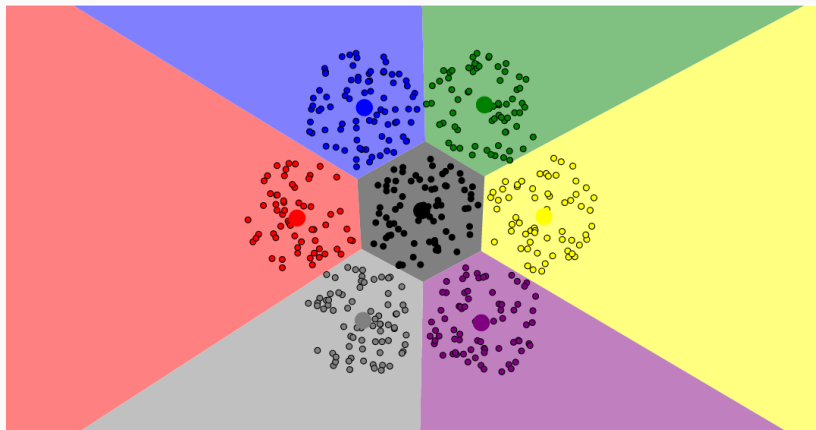
# Необходимость задавать $k$



# Необходимость задавать $k$



# Необходимость задавать $k$





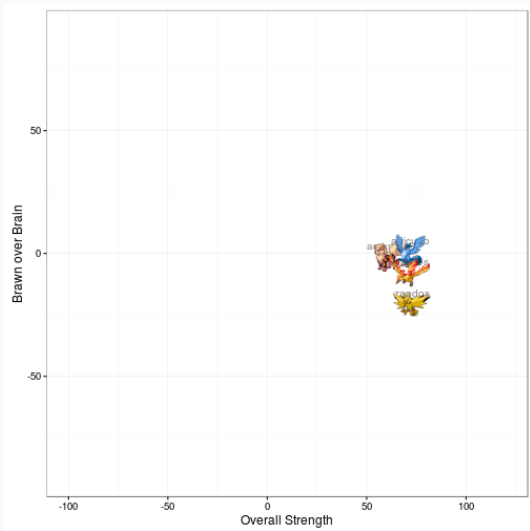
## Интересные результаты

---



Arcanine

# Интересные результаты



## Устранение недостатков

---

- Несколько случайных кластеризаций

- Несколько случайных кластеризаций
- Постепенное наращивание числа  $k$

- Несколько случайных кластеризаций
- Постепенное наращивание числа  $k$
- Использование `k-means++`

## Идея:

1. Выбрать первый центроид случайным образом
2. Для каждой точки найти значение квадрата расстояния до ближайшего центроида.
3. Выбрать из этих точек следующий центроид так, чтобы вероятность выбора точки была пропорциональна вычисленному для неё квадрату расстояния



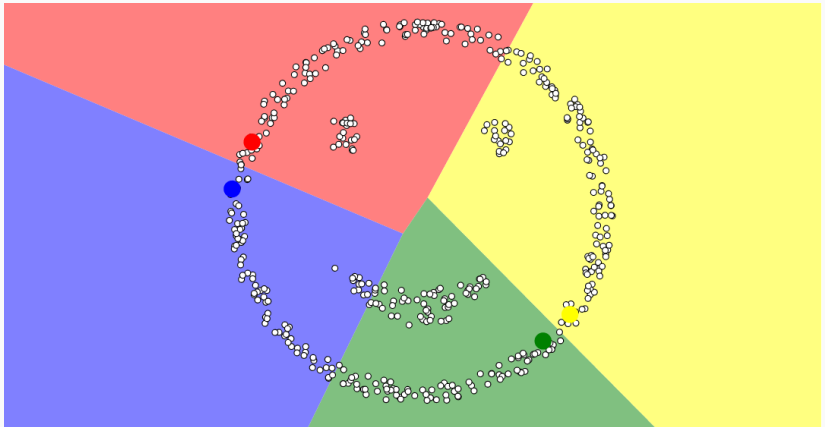
## Идея:

1. Получать на вход не  $k$ , а диапазон, в котором может находиться  $k$ .
2. Запустить  $k$ -means на самом маленьком значении из диапазона.
3. Разбить пополам полученные кластеры и проверить, не улучшилась ли кластеризация.

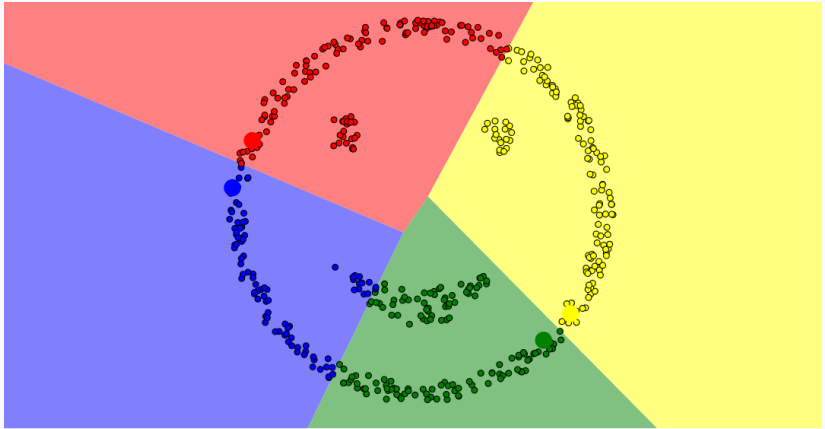
Когда k-means работает плохо

---

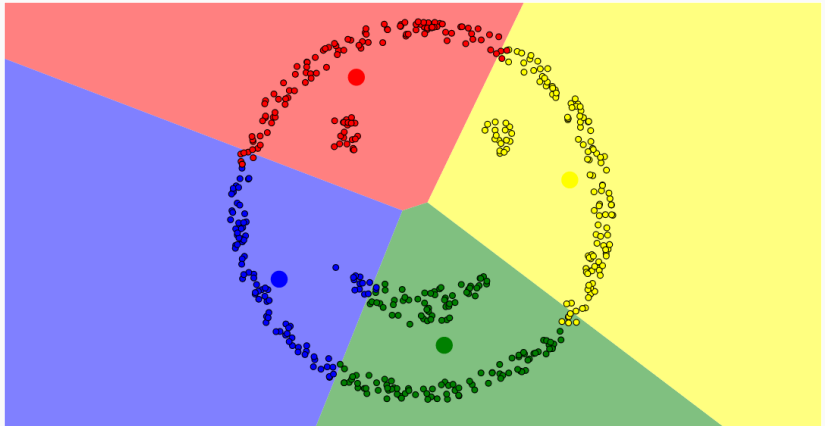
# "Не сферические данные"



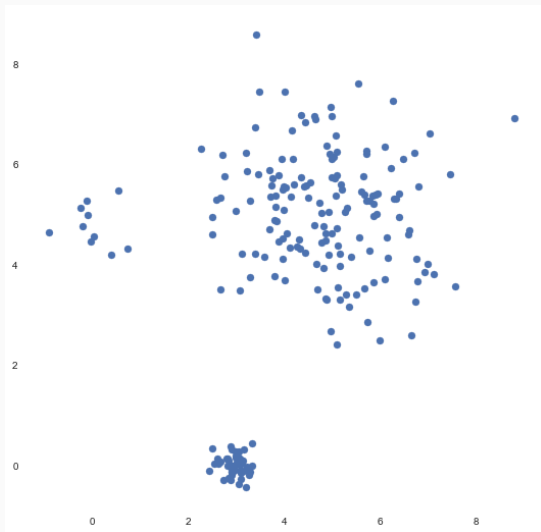
# "Не сферические данные"



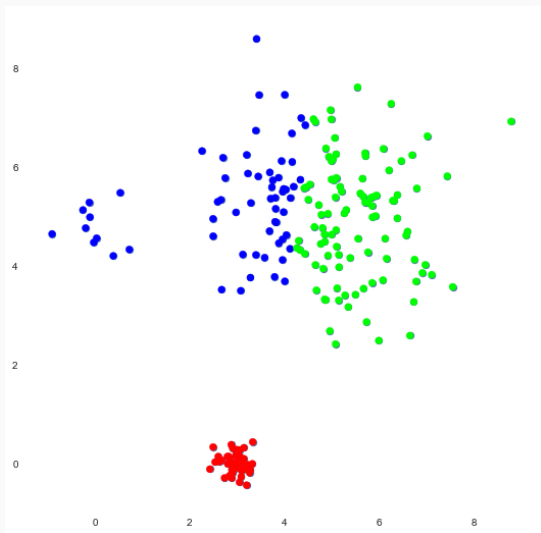
# "Не сферические данные"



# Разноразмерные кластеры



# Разноразмерные кластеры



**Идея:** Позволим объектам обновлять информацию друг о друге, для того, чтобы выбрать центр кластера.



**Идея:** Позволим объектам обновлять информацию друг о друге, для того, чтобы выбрать центр кластера.

$s(i, k)$  – "похожесть" объекта  $x_i$  на  $x_k$ ,  $s(k, k) < 0$ .

$r(i, k)$  – "ответственность",  $x_i$  решает насколько  $x_k$  подходит для того, чтобы быть центром кластера.

$a(i, k)$  – "доступность",  $x_k$  решает насколько  $x_i$  подходит для его кластера.

```
1 function AFFINITY_PROPAGATION( $S$ )
2    $R \leftarrow 0, A \leftarrow 0$ 
3   repeat[пока  $c_i$  не перестанет меняться]
4      $r(i, k) = s(i, k) - \max_{j \neq k}(a(i, j) + s(i, j))$ 
5      $i \neq k : a(i, k) = \min(0, r(k, k) + \sum_{j \neq i, j \neq k} \max(0, r(i, j)))$ 
6      $a(k, k) = \sum_{j \neq k} \max(0, r(i, j))$ 
7      $c_i = \arg \max_k (a(i, k) + r(i, k))$ 
```

+ Не нужно задавать количество кластеров

- + Не нужно задавать количество кластеров
- + "Выбросы" выделяются в отдельные кластеры

- + Не нужно задавать количество кластеров
- + "Выбросы" выделяются в отдельные кластеры
- Долгое время работы

# Метод распространения близости

- + Не нужно задавать количество кластеров
- + "Выбросы" выделяются в отдельные кластеры
- Долгое время работы
- Часто нуждается в постобработке

# Графовые алгоритмы

---

Какие есть две очевидные идеи?



Идеи:

1. Выделение связных компонент
2. Минимальное покрывающее дерево

1. Рисуем полный граф с весами, равными расстоянию между объектами
2. Выбираем лимит расстояния  $r$  и выкидываем все ребра длиннее  $r$
3. Компоненты связности полученного графа – наши кластеры

Как искать **компоненты связности**?

# Минимальное покрывающее дерево

Минимальное остовное дерево – дерево, содержащее все вершины графа и имеющее минимальный суммарный вес ребер.

Как использовать минимальное остовное дерево для разбиения на кластеры?

# Минимальное покрывающее дерево

Строим минимальное остовное дерево, а потом выкидываем из него ребра максимального веса.

Сколько ребер выбросим – столько кластеров получим.

# Иерархическая кластеризация

---

## Идея:

1. Считаем каждую точку кластером.
2. Затем объединяем ближайшие точки в новый кластер.
3. Повторяем.



```
1 function LANCE-WILLIAMS( $X^l$ )
2    $C_1 = \{\{x_1\}, \{x_2\}, \dots, \{x_l\}\}$ 
3   for  $t = 2, \dots, l$  do
4      $(U, V) = \arg \min_{U \neq V} \rho(U, V)$ 
5      $W = U \cup V$ 
6      $C_t = C_{t-1} \cup \{W\} \setminus \{U, V\}$ 
7     for each  $S \in C_t$  do
8       вычислить  $\rho(W, S)$ 
```

Чего-то не хватает?

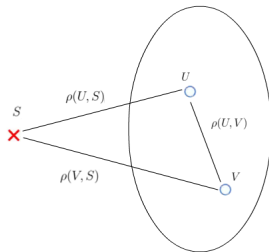
# Формула Ланса-Уильямса

$$W = \{U \cup V\}$$

Знаем:

$$\rho(U, S), \rho(V, S), \rho(U, V)$$

Расстояние  $\rho(W, S)$ ?



# Формула Ланса-Уильямса

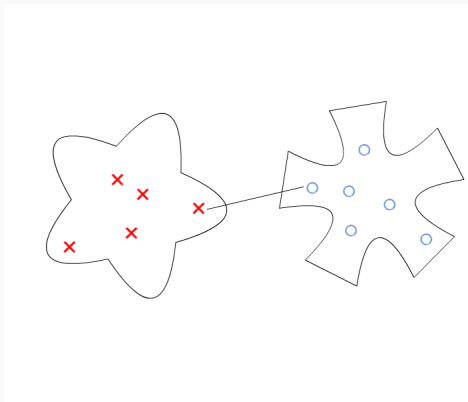
$$W = \{U \cup V\}$$

$$\rho(U \cup V, S) = \alpha_U \rho(U, S) + \alpha_V \rho(V, S) + \\ + \beta \rho(U, V) + \gamma |\rho(U, S) - \rho(V, S)|$$

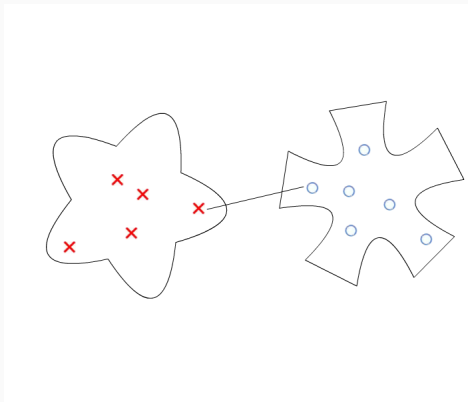
$\alpha_U, \alpha_V, \beta, \gamma$  – числовые параметры

Значения параметров  $\alpha_U, \alpha_V, \beta, \gamma$  ?

## Расстояние ближнего соседа



## Расстояние ближнего соседа

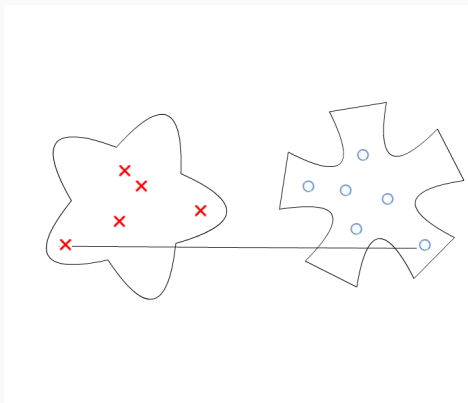


$$\alpha_U = \alpha_V = \frac{1}{2}$$

$$\beta = 0$$

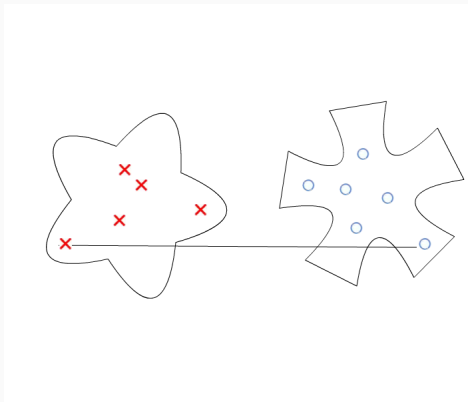
$$\gamma = -\frac{1}{2}$$

## Расстояние дальнего соседа





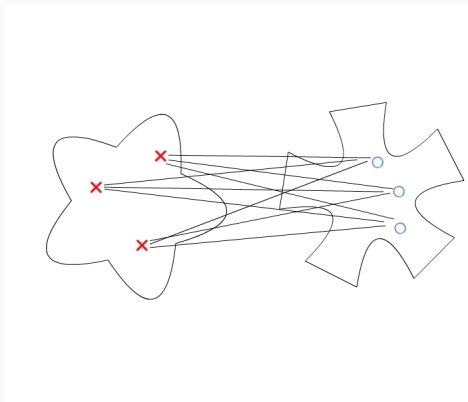
## Расстояние дальнего соседа

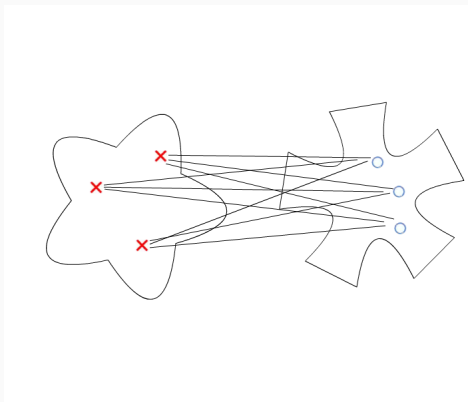


$$\alpha_U = \alpha_V = \frac{1}{2}$$

$$\beta = 0$$

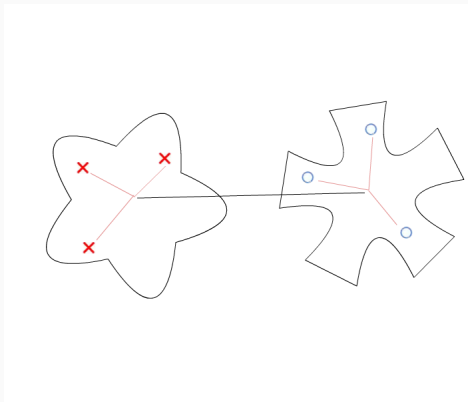
$$\gamma = \frac{1}{2}$$





$$\alpha_U = \frac{|U|}{|W|}$$
$$\alpha_V = \frac{|V|}{|W|}$$
$$\beta = \gamma = 0$$

# Расстояние Уорда



$$\alpha_U = \frac{|S|+|U|}{|S|+|W|}$$

$$\alpha_V = \frac{|S|+|V|}{|S|+|W|}$$

$$\beta = \frac{-|S|}{|S|+|W|}$$

$$\gamma = 0$$

# Обучение с частичным привлечением учителя

---

Вопросы?

## Что почитать по этой лекции

- Christopher M. Bishop "Pattern Recognition and Machine Learning" Chapter 9
- G. James, D. Witten, T. Hastie, R. Tibshirani "An Introduction to Statistical Learning" Chapter 10.3

# Что происходит сейчас в области

ICML'16: Interactive Bayesian Hierarchical Clustering

ICML'16: k-variates++: more pluses in the k-means++

NIPS'16: Clustering with Same-Cluster Queries

NIPS'16: Fast and Provably Good Seedings for k-Means



## На следующей лекции

- Деревья принятия решений
- Виды правил
- Поиск информативных закономерностей
- Подрезание деревьев
- Oblivious деревья
- Random forest