

---

# Задачи классификации и кластеризации





# Олег Булыгин

Lead Data scientist / Data analyst /  
developer, IT-тренер.

---

## Аккаунты в соц.сетях

@ obulygin91@ya.ru

vk.com/obulygin91

in linkedin.com/in/obulygin

@obulygin91

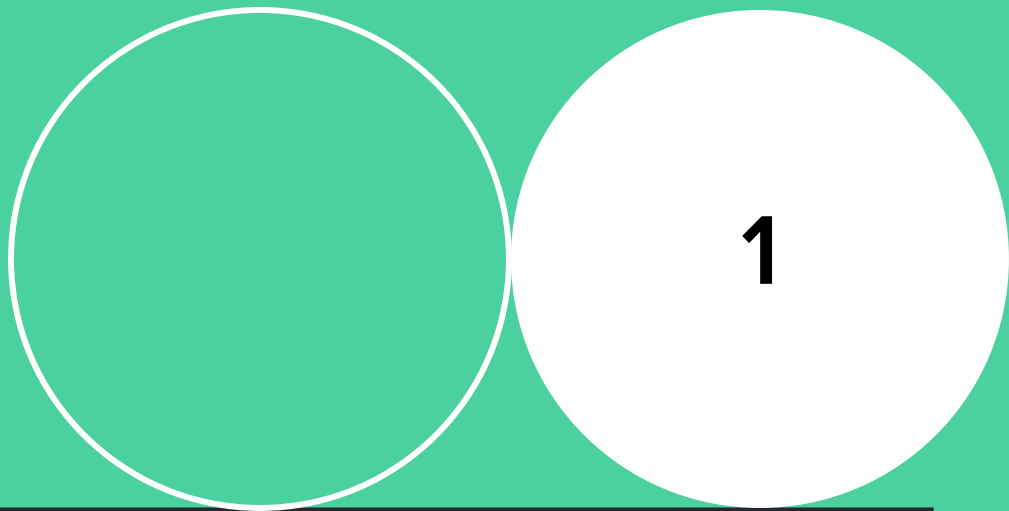


# Сегодня на лекции

1. Классификация
2. Кластеризация

---

# Дискриминантный анализ



## Выбираем метод анализа

Зависимая переменная	Независимая переменная	Метод
Непрерывная	Непрерывная	Регрессионный анализ
Категориальная	Непрерывная	Дискриминантный анализ

# Дискриминантный анализ

## Задача:

По значениям дискриминантных переменных для объектов получить значения классифицирующей переменной, то есть определить классы, в которые попадают эти объекты.

На основании некоторых признаков (независимых переменных) объект может быть причислен к одной из **заранее** заданных групп.

## Примеры

- Определить класс цветка на основе его характеристик (ширина и длина чашелистика и лепестка)
- Понять категорию заемщика - надежный или нет
- ...

# Входные данные

Класс (label)	X1 (petal_length)	X2 (petal_width)	X3 (sepal_length)	X4 (sepal_width)
Setosa	2.43	1.17	4.32	2.35
Setosa	1.42	5.23	2.32	3.31
Versicolor	2.12	3.12	4.87	4.34
...	...	...	...	...
Virginica	4.21	3.21	5.53	3.42
Virginica	5.65	5.34	4.32	2.23

# Дискриминантная функция

$$z=b_0+b_1*x_1 + b_2*x_2 + ... + b_n*x_n$$

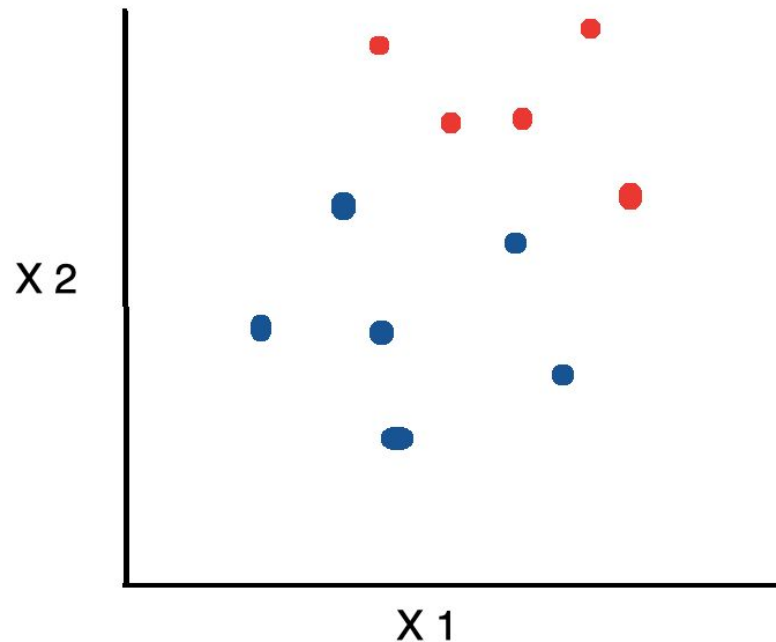
Задача - определить коэффициенты  $b$ , чтобы по значениям дискриминантной функции можно было с максимальной точностью провести разделение по группам.



# Пример для двух классов

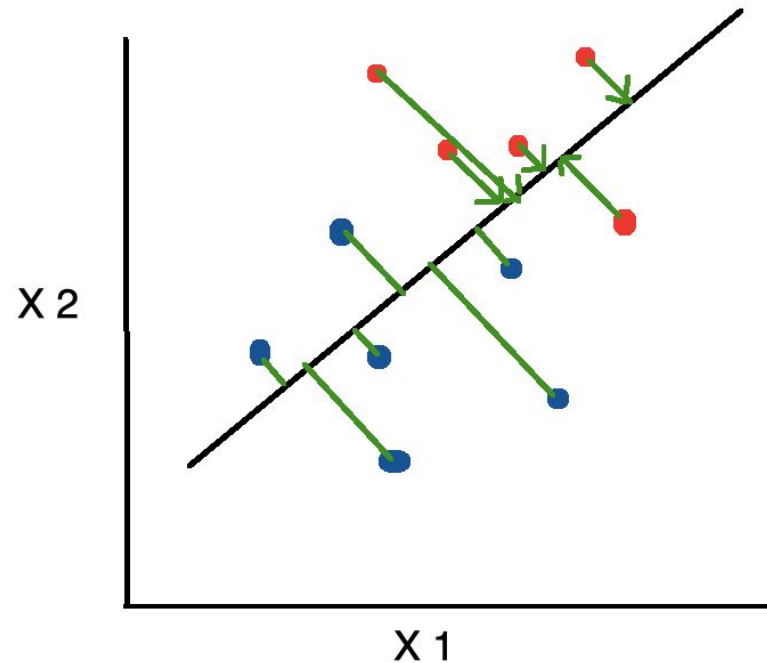
Хотим разделить красные и синие точки (классы) в зависимости от значений двух независимых переменных ( $x_1$  и  $x_2$ ).

Например: красные точки — благонадежные заемщики, а синие — нет.



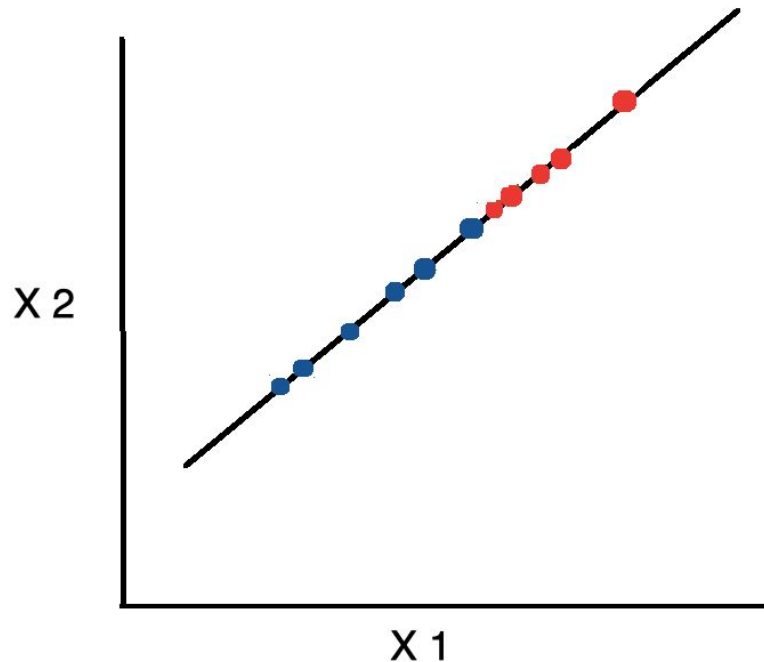
# Пример для двух классов

LDA строит линию, и проецирует на нее точки из нашей выборки



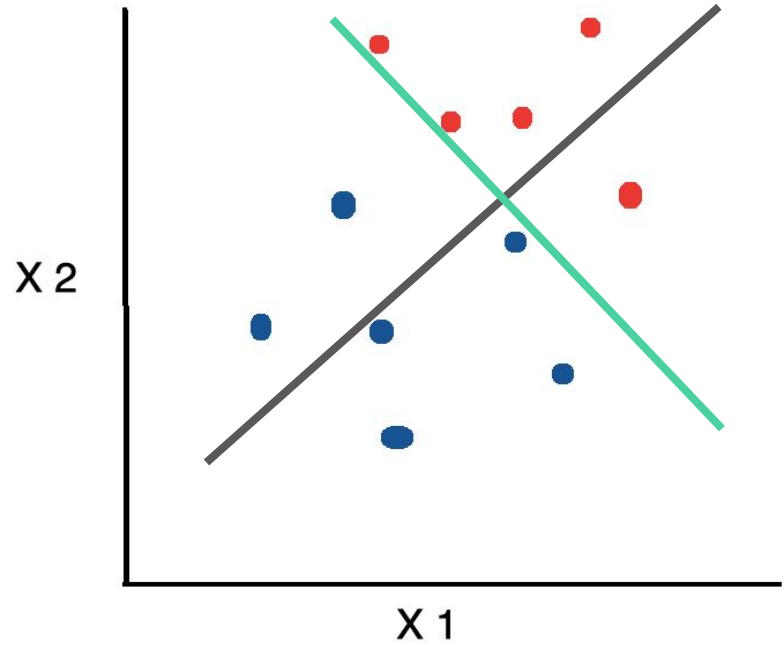
# Пример для двух классов

Эта линия строится специальным образом, чтобы “центр” области красных точек был максимально отдален от “центра” области синих точек



# Пример для двух классов

Дискриминантной функцией в данном случае будет перпендикуляр к этой линии



# Пример для двух классов

Более формально:

$$\mathbf{Z} = \mathbf{b0} + \mathbf{b1} * \mathbf{x1} + \mathbf{b2} * \mathbf{x2}$$

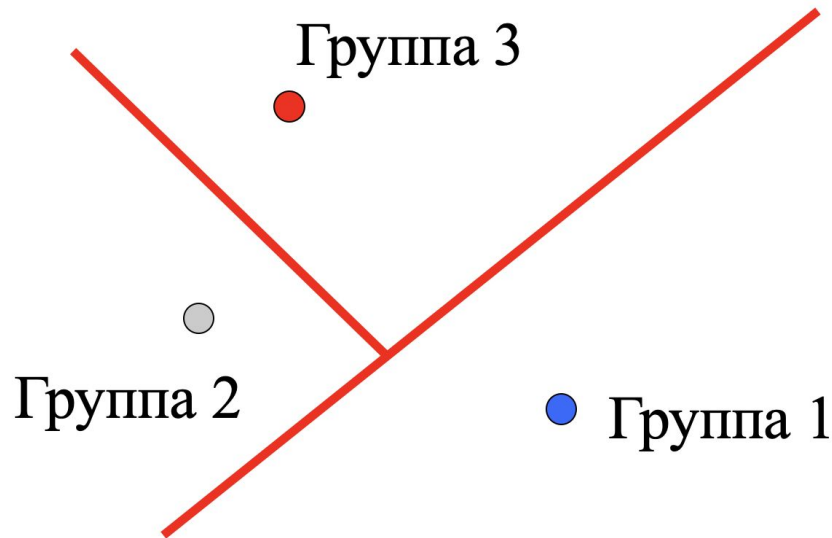
Пусть **Z1** и **Z2** - средние значения дискриминантной функции для синих и красных точек

Задача - подобрать такие коэффициенты **b0**, **b1**, **b2**, чтобы максимизировать **|Z1 - Z2|**

После этого можно найти **Zкр = (Z1 - Z2)/2** и построить решающее правило для нового образца (**Zn**): если **Zn - Zкр >= 0**, то точка синяя, иначе - красная

## Если классов больше 2-х:

Если классов больше двух - то нужна не одна дискриминантная функция, а несколько



# Предположения

- **Нормальное распределение.** Предполагается, что анализируемые переменные представляют выборку из многомерного нормального распределения. Отступление обычно не является критичным.
- **Однородность дисперсий/ковариаций.** Предполагается, что матрицы дисперсий/ковариаций переменных однородны. Как и ранее, малые отклонения не фатальны
- **Слабая скоррелированность признаков**

# Другие виды анализа

- С отбором признаков
  - Позволяет включать самые важные признаки
  - Позволяет исключать наименее информативные признаки
- Квадратичный дискриминантный анализ
  - В случае различия матриц ковариаций
- Логистическая регрессия
  - Еще один способ разделить данные

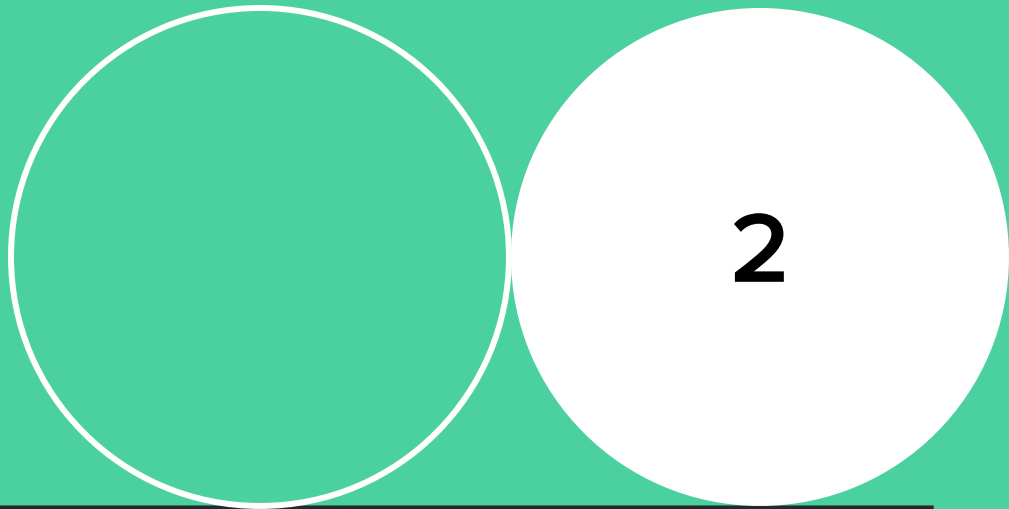


# Практика

1. Загрузим датасет по качеству вин
2. Применим к нему линейный дискриминантный анализ и посмотрим качество предсказания на отложенной выборке
3. Оставим два признака в данных - ash и flavanoids и два класса - 0 и 2
4. Попробуем еще раз применить LDA на такой ограниченной выборке
5. Посмотрим визуально, как выглядит линия разделения классов

---

# Кластерный анализ



# Цель

Цель кластерного анализа – нахождение групп схожих объектов в выборке данных. Эти группы удобно называть кластерами.

## **Пример:**

У нас есть список клиентов с характеристиками, которые у нас заказывали те или иные товары. Попробуем выделить в них группы (кластера) на основе признаков для проведения микротаргетинга.

# Отличие от других задач

- Отсутствие “правильных” ответов. Задача кластерного анализа - открытая по своей природе
- Сложность оценки качества

# Основные методы кластерного анализа

- Иерархический
- К-средних

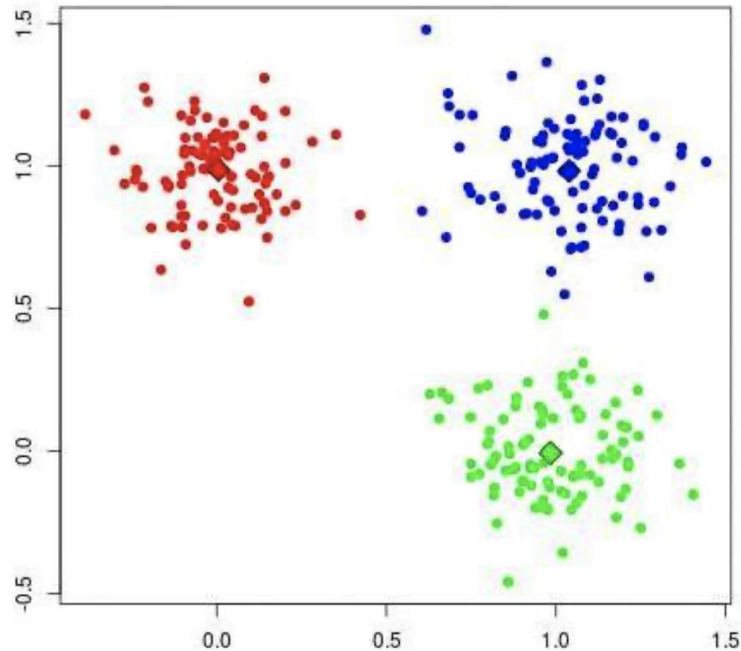
**К-средних**



# Метод К-средних

Итеративный метод, который работает непосредственно с объектами

Нужно заранее указать число кластеров



# Пошаговый пример

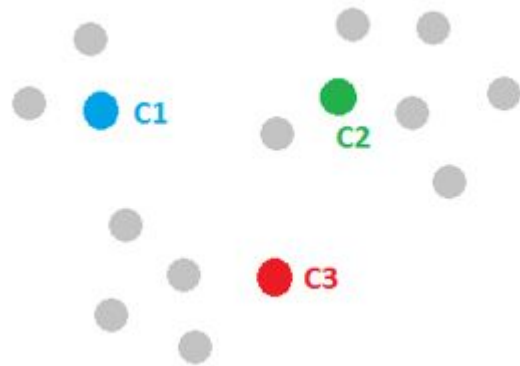
1. Нам дан набор объектов с их характеристиками





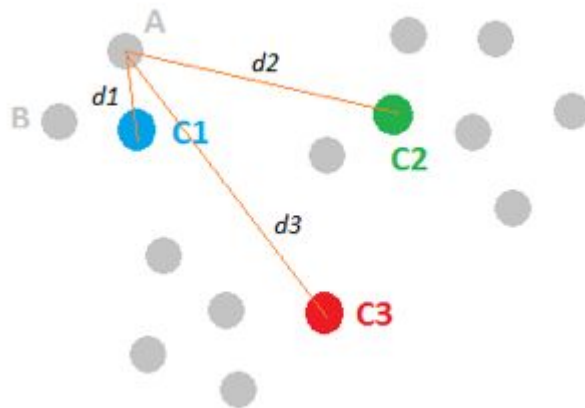
# Пошаговый пример

2. Выбираем количество кластеров  
(например, 3) и случайно располагаем  
их среди наших точек



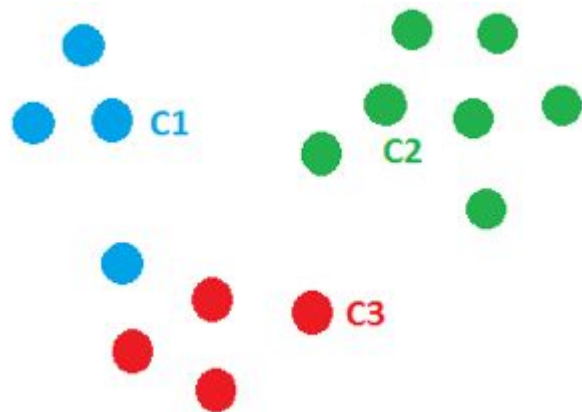
# Пошаговый пример

3. Берем одну точку из имеющихся у нас и считаем расстояние до каждого из центров. Относим точку к тому кластеру, расстояние до которого - минимально.



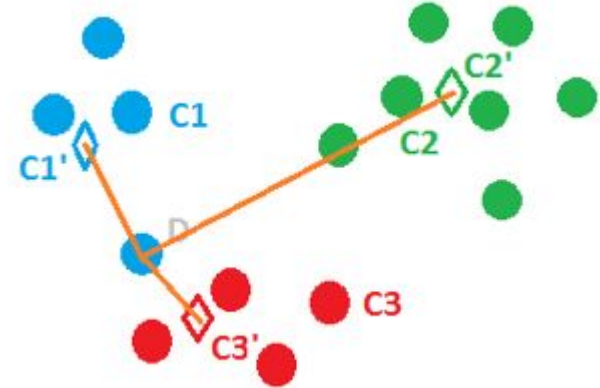
# Пошаговый пример

4. Таким образом относим все точки к одному из кластеров.



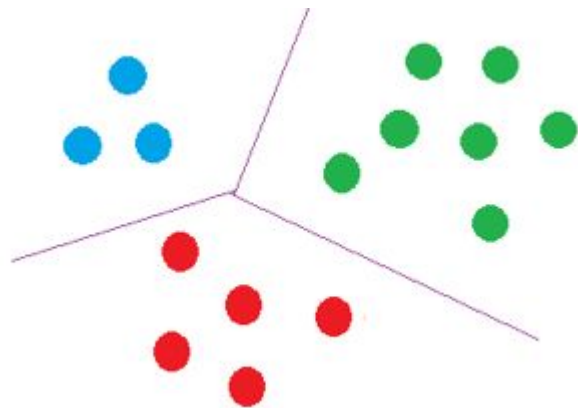
# Пошаговый пример

5. Передвигаем центры кластеров в центры точек. И запускаем шаги 3-5 заново.



# Пошаговый пример

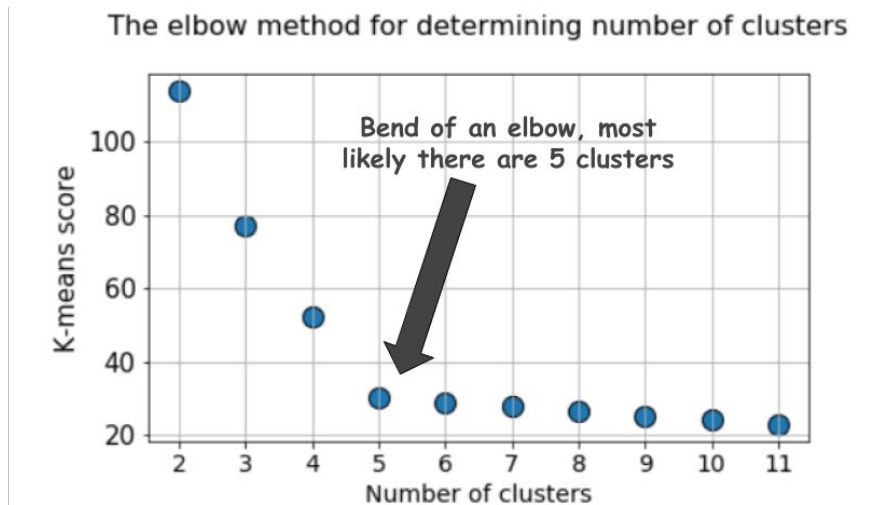
6. После стабилизации алгоритма получаем готовую кластеризацию.



# Выбор количества кластеров

## Метод локтя:

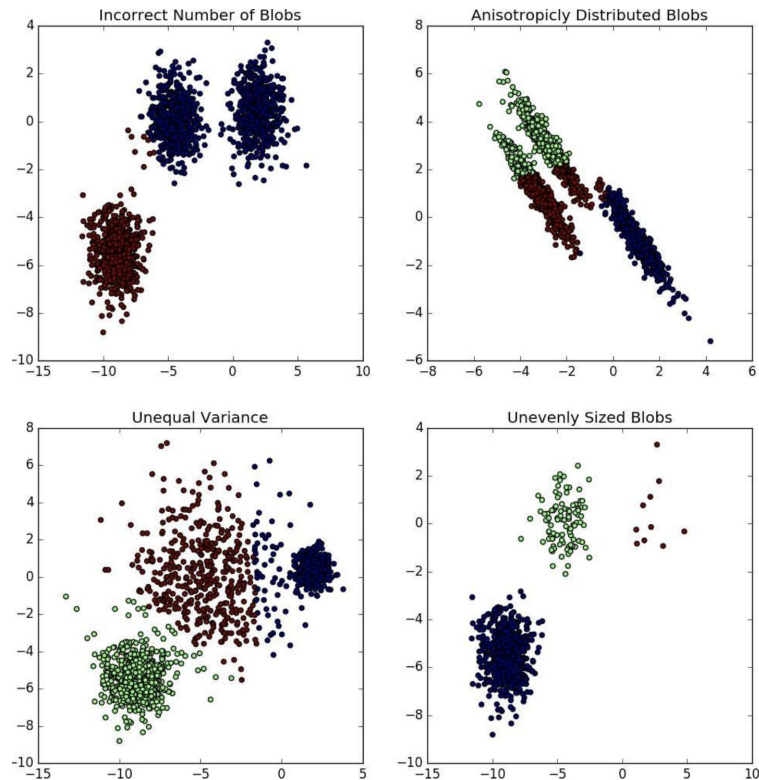
Смотрим на зависимость суммы внутриклассовых отличий от центра (в sklearn - inertia). Останавливаемся, когда при увеличении количества кластеров метрика перестает значительно улучшаться



# Ограничения

Алгоритм может выдавать контринтуитивные результаты, если:

1. Вы выбрали не то число кластеров
2. Кластеры - не выпуклые и близко расположены
3. Кластера обладают разной дисперсией



# Особенности работы

- Алгоритм k-средних сходится к локальному оптимуму. Следовательно, результат, полученный с помощью K-средних, не обязательно является самым оптимальным.
- Инициализация центров крайне важна для качества найденного решения. По умолчанию - используйте k-means++.
- Крайне чувствителен к масштабу признаков (будет дальше в лекции).



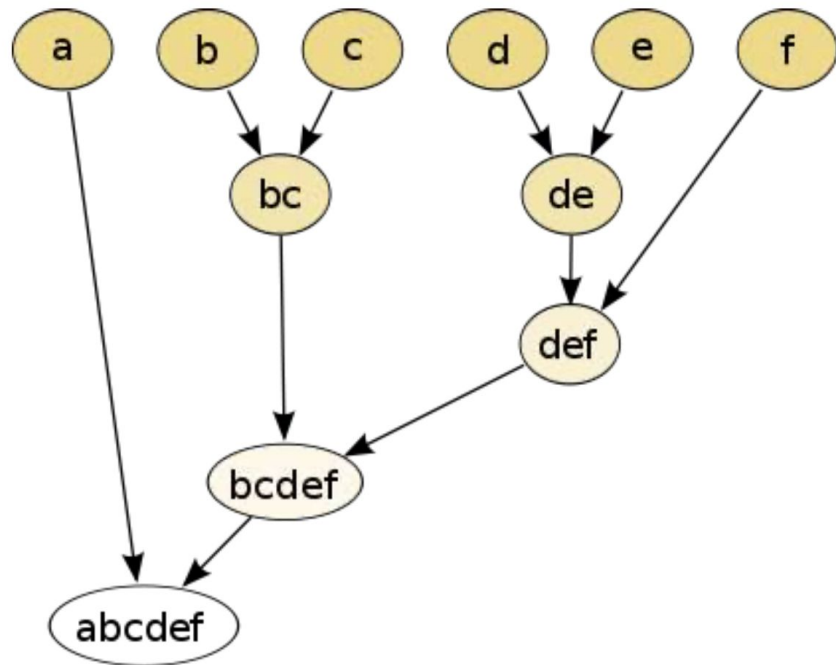
# Иерархический алгоритм



# Иерархический алгоритм

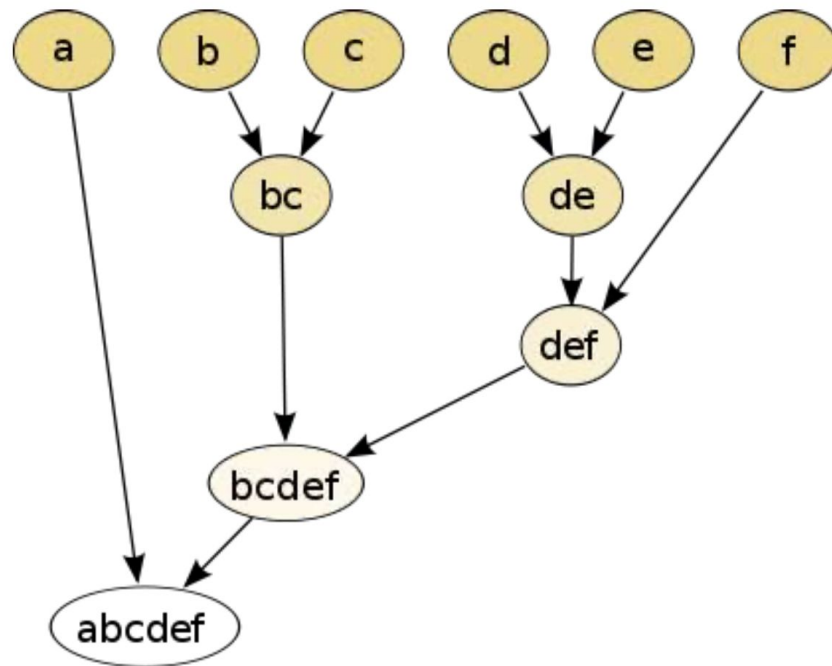
В иерархических методах происходит последовательное объединение наиболее близких объектов в один кластер.

Процесс объединения можно показать визуально в виде дендрограммы (дерева объединения).



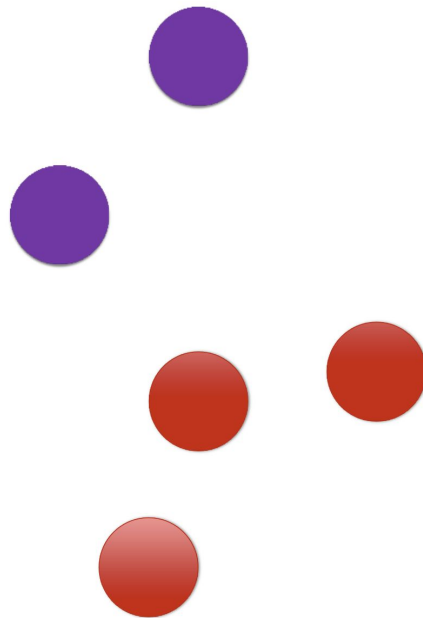
# Алгоритм

1. Все объекты - отдельные кластера
2. Повторять, пока у нас больше чем один кластер:
  - а. Соединить два ближайших кластера в один



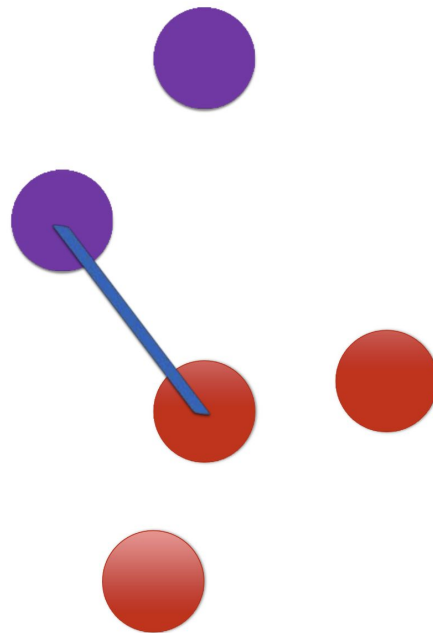
# Расстояние между кластерами

- Ближнего соседа
- Дальнего соседа
- Групповое среднее
- Расстояние между центрами
- Расстояние Уорда



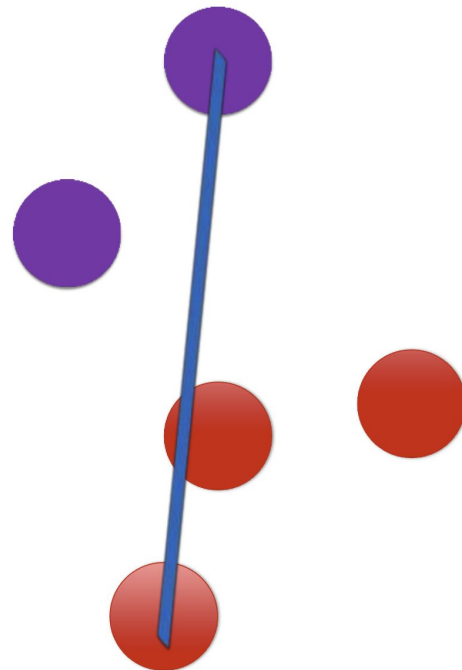
# Расстояние между кластерами

- **Ближнего соседа**
- Дальнего соседа
- Групповое среднее
- Расстояние между центрами
- Расстояние Уорда



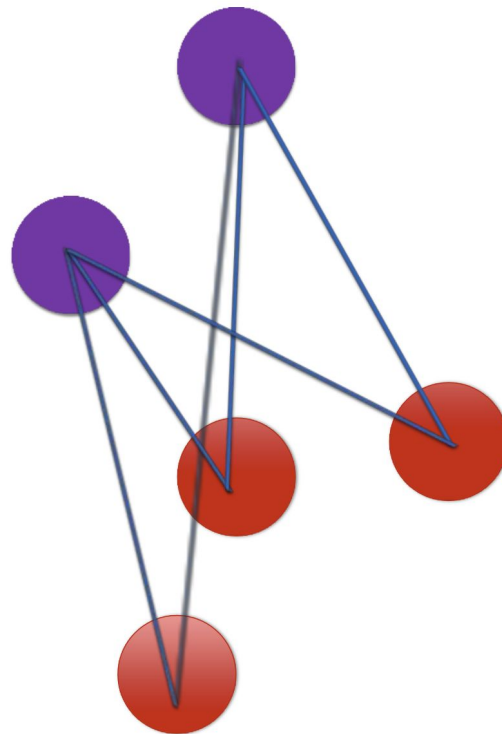
# Расстояние между кластерами

- Ближнего соседа
- **Дальнего соседа**
- Групповое среднее
- Расстояние между центрами
- Расстояние Уорда



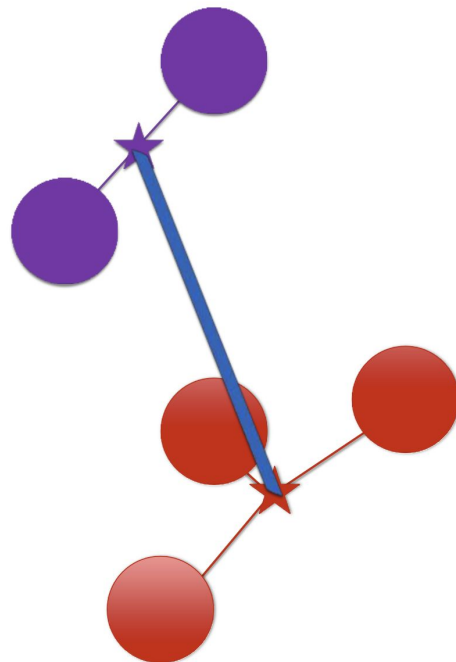
# Расстояние между кластерами

- Ближнего соседа
- Дальнего соседа
- **Групповое среднее**
- Расстояние между центрами
- Расстояние Уорда



# Расстояние между кластерами

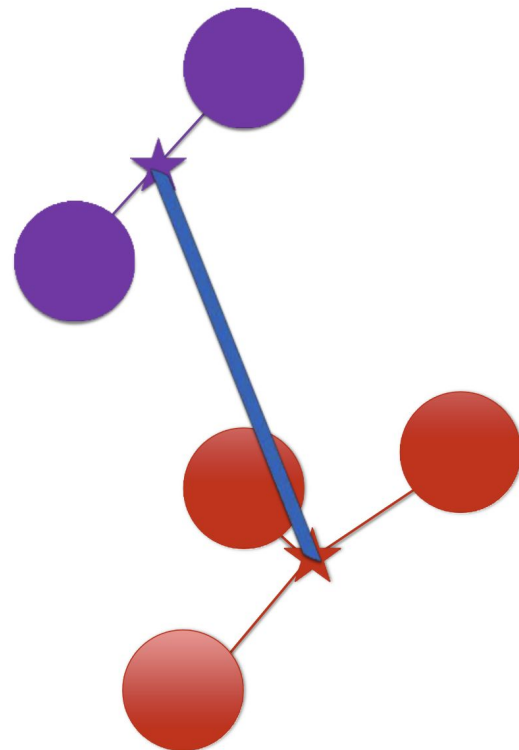
- Ближнего соседа
- Дальнего соседа
- Групповое среднее
- **Расстояние между центрами**
- Расстояние Уорда





# Расстояние между кластерами

- Ближнего соседа
- Дальнего соседа
- Групповое среднее
- Расстояние между центрами
- **Расстояние Уорда**



# Рекомендуемое расстояние

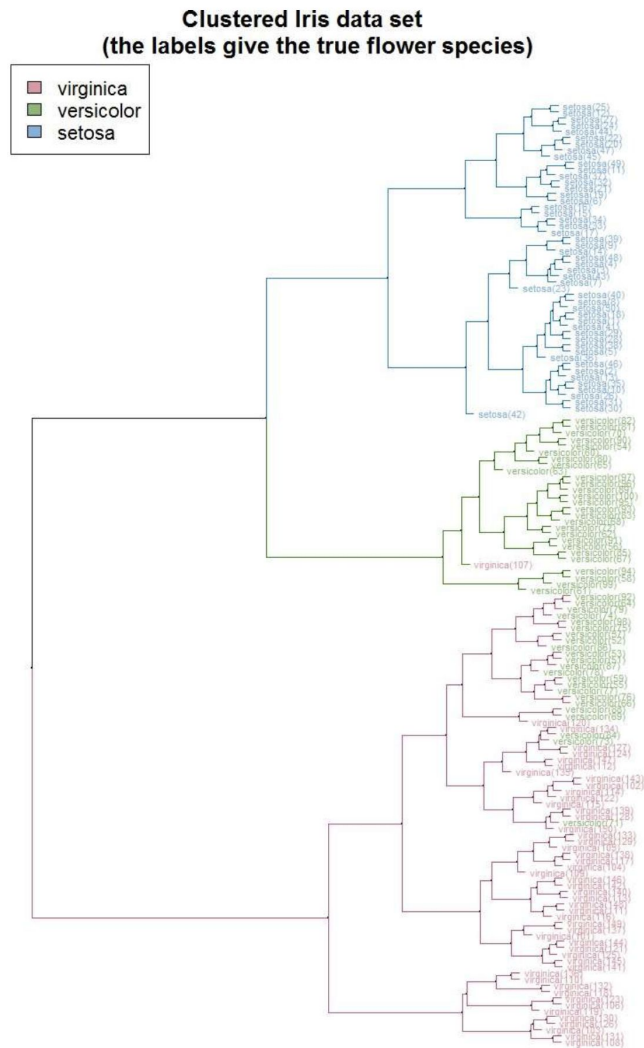
Используйте по умолчанию метод Уорда.

Причины:

- При каждом слиянии расстояние между кластерами растет
- Основано на центра кластеров
- При каждом объединении минимизирует внутрикластерную дисперсию

# Пример

## Кластеризация цветков Ириса



**О подсчете  
расстояний**



# Подсчет расстояний

Все алгоритмы кластеризации завязаны на подсчет расстояния между точками.

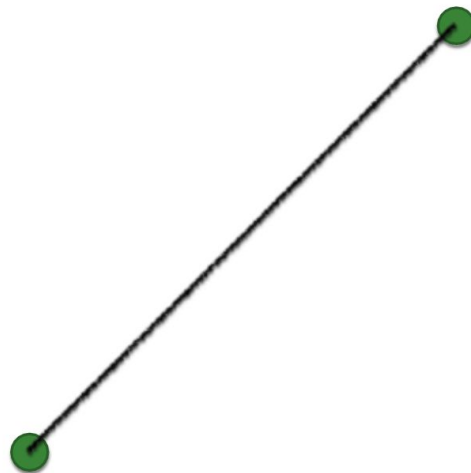
Расстояние между точками - это способ трактовать “похожесть” объектов в выборке.

Делать это можно различными способами. Рецепта как выбрать правильное расстояние - нет. Надо экспериментировать с учетом имеющихся данных и природы задачи.

# Евклидово

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{k=1}^n (p_k - q_k)^2}$$

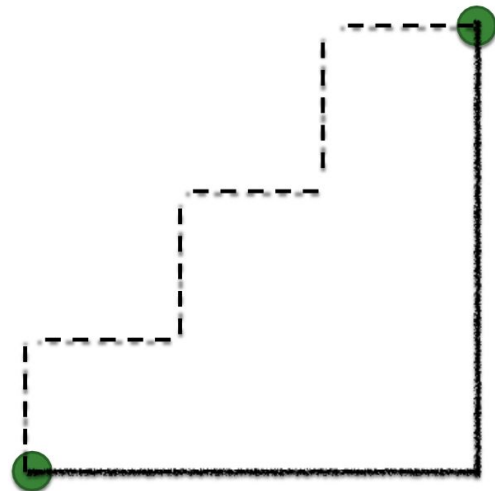
- Обычно мы все понимаем под расстоянием именно его
- Расстояние по прямой от точки А до точки Б



# Манхеттенское

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^n |p_i - q_i|$$

- Другое название - расстояние городских кварталов



# Другие

- Расстояние Чебышева
- Косинусное расстояние
- Расстояние Минковского
- 1 - коэффициент корреляции
- ...



# Нормализация данных

При выполнении кластерного анализа все признаки обычно приводят к единому масштабу, чтобы каждый фактор играл одинаковую роль при определении сходства объектов

## Выборка:

Человек	Рост (мм)	Вес (тонны)	Рост (см)	Вес (кг)
1	1800	0.085	180	85
2	1620	0.085	162	85
3	1800	0.093	180	93

## Евклидово расстояние от 1 до 2 и 3:

	Мм и тонны	См и кг
2	180	18
3	0.008	8

# Нормализация данных

$$Z = \frac{x_i - \bar{x}}{s}$$

Из каждого признака вычитаем среднее по этому признаку и делим на стандартное отклонение признака.

# Практика

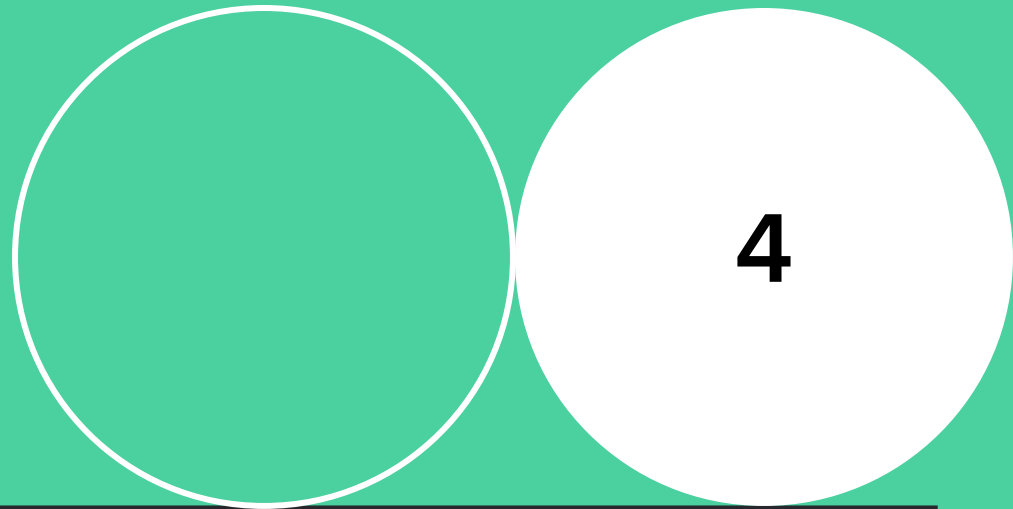
1. Создадим датасет при помощи функции `make_blobs`
2. Проведем кластеризацию методом k-средних и подберем оптимальное количество кластеров
3. Построим дендрограмму для иерархической кластеризации

# Важно помнить!

1. Большая часть методов кластерного анализа - набор эвристических правил
2. Не существует адекватного способа оценки “качества” кластеризации
3. Разные методы кластерного анализа и даже разные “настройки” одного и того же метода порождают разные результаты кластеризации

---

# Итоги



# Что мы узнали сегодня

- Разделять данные на категории при помощи дискриминантного анализа
- Искать кластера похожих объектов в выборке при помощи кластерного анализа



---

# Домашнее задание



# Домашнее задание

1. Возьмите датасет с цветками iris'a (функция `load_iris` из библиотеки `sklearn`)
2. Оставьте два признака - `sepal_length` и `sepal_width` и целевую переменную
3. Разделите данные на выборку для обучения и тестирования
4. Постройте модель LDA
5. Визуализируйте предсказания для тестовой выборки и центры классов (разделяющую плоскость можно не строить, но обратите внимание, что класса 3).
6. Отбросьте целевую переменную и оставьте только два признака - `sepal_length` и `sepal_width`
7. Подберите оптимальное число кластеров для алгоритма `kmeans` и визуализируйте полученную кластеризацию



---

# Дискриминантный и факторный анализ

Вопросы?