

ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

КАФЕДРА АВТОМАТИЗАЦИЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

ЛЕКЦИЯ № 03

Метрические модели

СОСТАВИТЕЛЬ: КАНД. ТЕХН. НАУК БЫКАДОР В.С.

Общее представление о метрическом пространстве

Пусть мы выберем 2-ва из 4-х признаков в ранее рассмотренной задаче с цветами ириса.

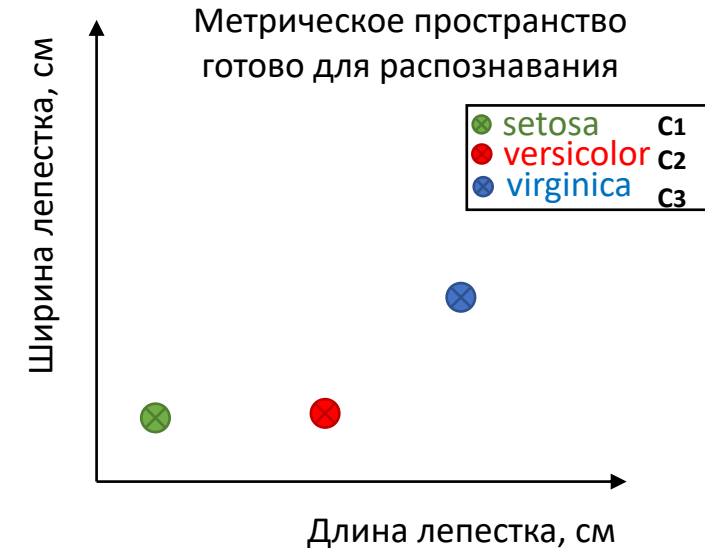
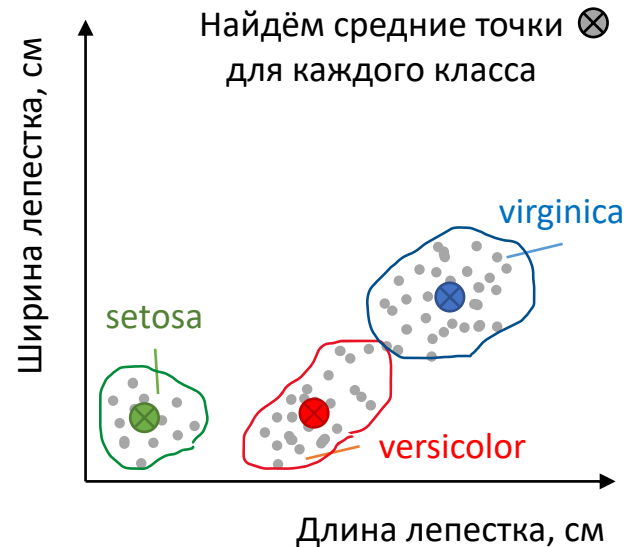
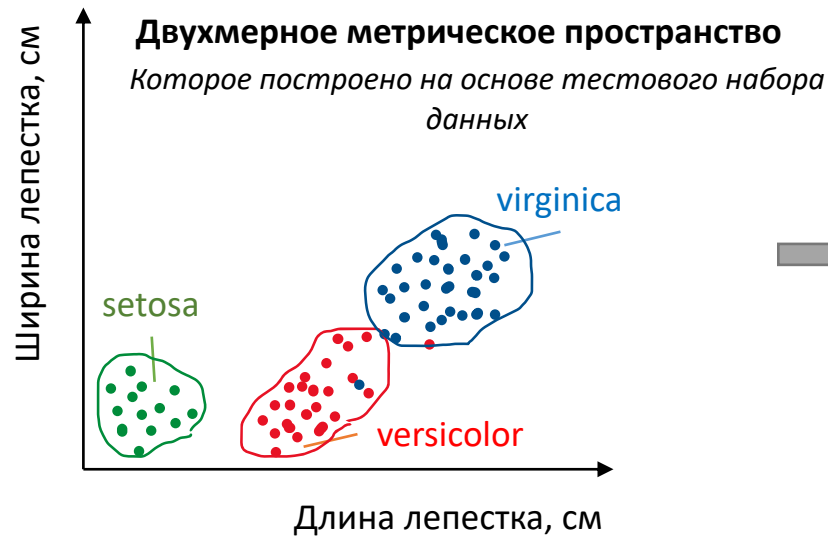


- Длина чашелистика.
- Ширина чашелистика.
- Длина лепестка.
- Ширина лепестка.

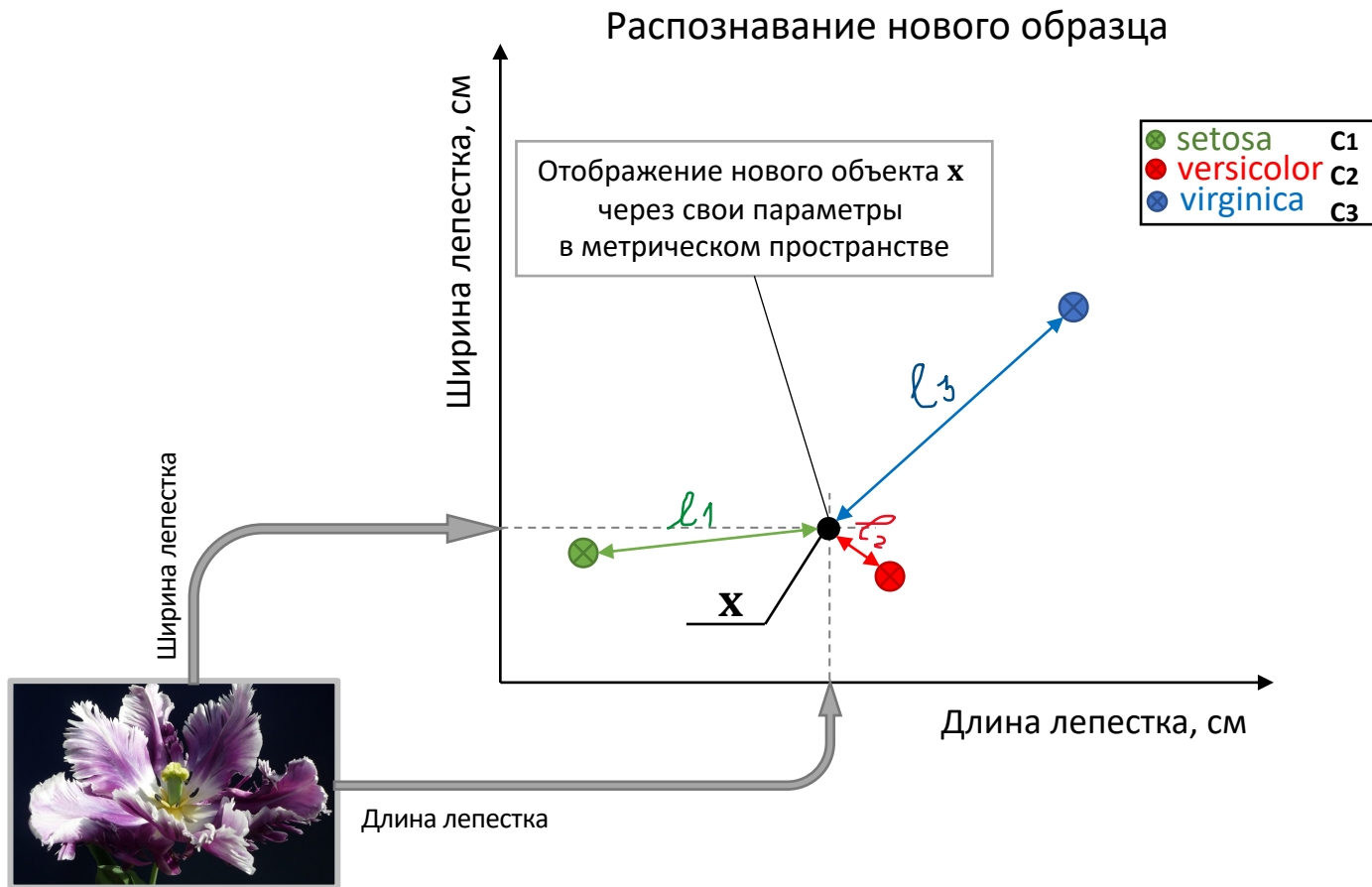
Выберем в качестве признаков **длину лепестков** и **ширину лепестков**.

Будем использовать каждый признак как отдельную размерность (ось) метрического пространства, так как выбрано 2-ва признака, то пространство будет двумерным. Если количество признаков будет равно d , то и метрическое пространство будет d -мерным.

В качестве единиц измерения размерностей могут быть использованы любые, а не только измеряющие физическое расстояние. Например, можно построить метрическое пространство на основе таких величин как плотность и сила тока.



Как выполняется распознавание в метрическом пространстве

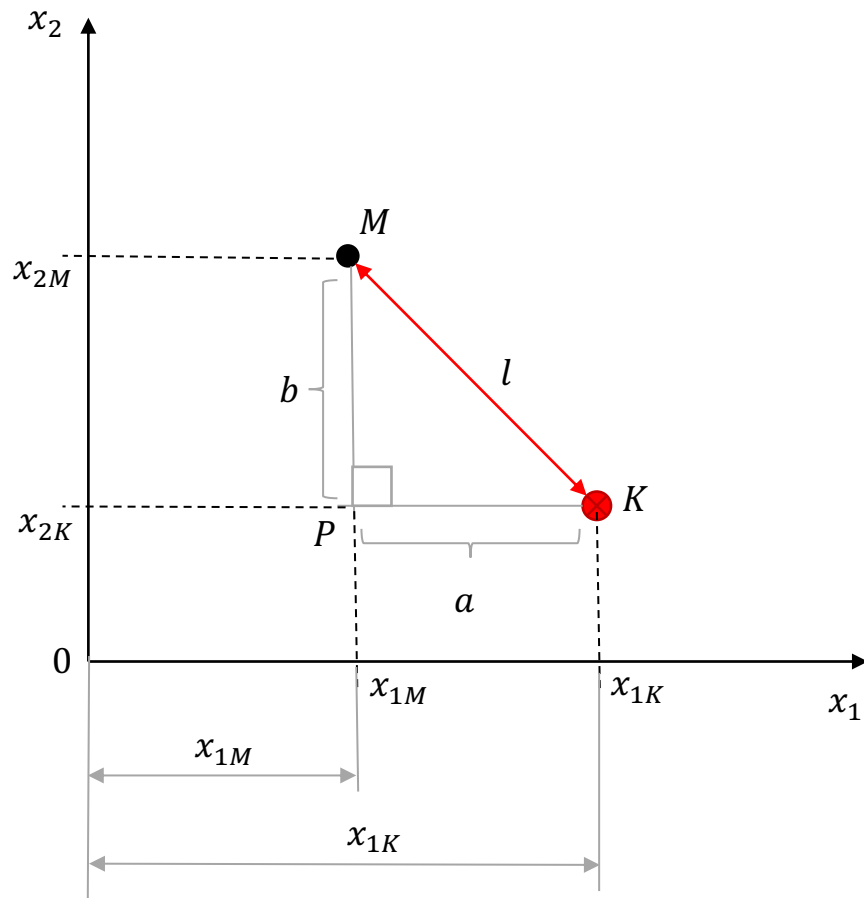


1) Выполняем измерение расстояние от новой точки до каждой средней точки, являющейся представителем соответствующего класса $L = \{l_1, l_2, l_3\}$.

2) Выполняется анализ расстояний от нового объекта до известных средних точек и выбирается минимальное расстояние из всех $\min(L)$.

В данном случае минимальным расстоянием будет являться $(l_{\min} = l_2) \Rightarrow x \in C_2$, то есть новый объект вероятнее всего принадлежит к ирису **versicolor**.

Как рассчитывается расстояние



Расстояние между точками ***KM*** является гипотенузой ***l*** прямоугольного треугольника ***PKM***

$$l^2 = a^2 + b^2 = \underbrace{(x_{1K} - x_{1M})^2}_a + \underbrace{(x_{2K} - x_{2M})^2}_b$$



$$l = \sqrt{(x_{1K} - x_{1M})^2 + (x_{2K} - x_{2M})^2}$$



$$l = ((x_{1K} - x_{1M})^2 + (x_{2K} - x_{2M})^2)^{\frac{1}{2}}$$



$$l = \left(\sum_{j=1}^2 (x_{jK} - x_{jM})^2 \right)^{\frac{1}{2}} \quad \text{- евклидово расстояние для двумерного пространства.}$$

Переход к расстоянию Минковского

Если количество признаков равно d (соответственно и количество измерений пространства, количество осей будет тоже равно d), а степень будет не только равно 2, а любому целому числу **больше 0**, то мы получим так называемое расстояние Минковского.

$$l = \left(\sum_{j=1}^2 (x_{jK} - x_{jM})^2 \right)^{\frac{1}{2}}$$

Diagram illustrating the formula for the Minkowski distance l . The formula is shown with handwritten red annotations: a red arrow points from d to the upper index 2 of the summation; a red arrow points from p to the lower index $\frac{1}{2}$ of the power; a red arrow points from the word "модуль" (module) to the vertical line segment between the summation and the power; and a red arrow points from the word "модуль" to the vertical line segment between the summation and the power.

т.к. p может быть нечётным числом (1, 3, 5,...).

Расстояние Минковского

Формальное определение

Если $\mathcal{X} = \mathbb{R}^d$, то **расстояние Минковского** порядка $p > 0$ определяется формулой:

$$\text{Dis}_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^d |x_j - y_j|^p \right)^{\frac{1}{p}}$$

Расстояние порядка p , которое обозначается как Dis_p называется **p -нормой**.

Координаты двух точек в d -мерном пространстве можно записать в виде векторов:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_d \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_d \end{pmatrix}$$

То есть в векторах \mathbf{x} и \mathbf{y} записаны значения признаков распознаваемого объекта и образца, который используется для распознавания.

Типовые нормы

1-норма (манхэттенское расстояние или расстояние городских кварталов): $\text{Dis}_1(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^d |x_j - y_j|$

2-норма (евклидово расстояние):

$$\text{Dis}_2(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^d |x_j - y_j|^2 \right)^{\frac{1}{2}} = \sqrt{\sum_{j=1}^d |x_j - y_j|^2} = \sqrt{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})}$$

Иногда полезно использовать разные шкалы для разных осей координат (если перемещение вдоль них происходит с разной скоростью), тогда можно перейти к расстоянию Махаланобиса:

$$\text{Dis}_M(\mathbf{x}, \mathbf{y} | \mathbf{M}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \mathbf{M} (\mathbf{x} - \mathbf{y})} \quad \text{Если } \mathbf{M} = \mathbf{I}, \text{ где } \mathbf{I} \text{ единичная матрица, то } \text{Dis}_I(\mathbf{x}, \mathbf{y} | \mathbf{I}) = \text{Dis}_2(\mathbf{x}, \mathbf{y})$$

(для справки)

0-норма (расстояние Хэмминга):

$$\text{Dis}_0(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^d I[x_j \neq y_j]$$

Определяет количество отличий между векторами \mathbf{x} и \mathbf{y} .

$$\begin{aligned} \mathbf{x} &= (1 \ 0 \ 0 \ 1 \ 1)^T \\ \mathbf{y} &= (1 \ 1 \ 0 \ 0 \ 1)^T \end{aligned} \quad \longrightarrow \quad \text{Dis}_0(\mathbf{x}, \mathbf{y}) = 2$$

∞ -норма (расстояние Чебышева):

$$\text{Dis}_\infty(\mathbf{x}, \mathbf{y}) = \max_j |x_j - y_j|$$

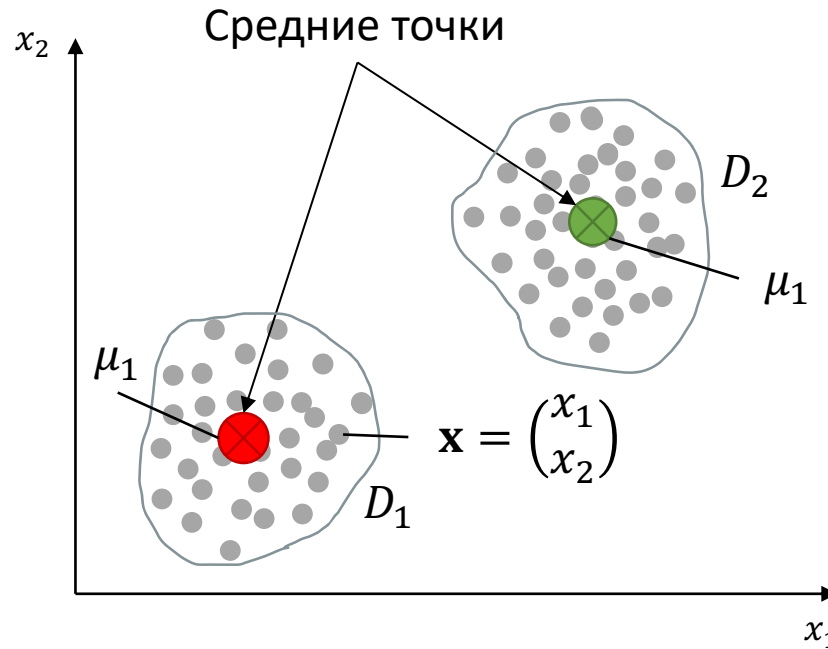
Это расстояние используется, когда необходимо определить два объекта в разные классы в зависимости от различий между ними по одному наиболее значимому признаку. Если один признак, по которому можно осуществить эффективное разделение классов существует.

Свойства расстояния

Свойства расстояния определяют так называемую **метрику** или **метрическое расстояние**:

- 1) Расстояние от точки до самой себя равно нулю: $\text{Dis}(\mathbf{x}, \mathbf{x}) = 0$;
- 2) Все остальные расстояния больше нуля: если $\mathbf{x} \neq \mathbf{y} \Rightarrow \text{Dis}(\mathbf{x}, \mathbf{y}) > 0$;
- 3) Расстояние симметрично: $\text{Dis}(\mathbf{x}, \mathbf{y}) = \text{Dis}(\mathbf{y}, \mathbf{x})$;
- 4) Объезд не может быть сократить расстояние (неравенство треугольника): $\text{Dis}(\mathbf{x}, \mathbf{z}) \leq \text{Dis}(\mathbf{x}, \mathbf{y}) + \text{Dis}(\mathbf{y}, \mathbf{z})$.

Фундаментальное свойство средней точки



В общем случае координаты средней точки не совпадают с координатами одной из точек области D . В этом случае средняя точка называется **центроида**.

Но можно наложить ограничение, чтобы координаты средней точки совпадали с координатами одной из точек области D . В этом случае средняя точка называется **медоида**.

Теорема

Среднее арифметическое μ множества точек D в евклидовом пространстве является единственной точкой, в которой сумма квадратов евклидовых расстояний до этих точек достигает минимума.

$$\mu = \frac{1}{|D|} \sum_{\mathbf{x} \in D} \mathbf{x}$$

, где $|D|$ - количество объектов в области D ;

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_d \end{pmatrix}.$$

Кластеризация алгоритмом K средних

Проблема K средних не имеет эффективного способа нахождения глобального минимума, поэтому приходится использовать эвристические алгоритмы. Самым известным из таких алгоритмов так и называется « K средних».

Это алгоритм машинного обучения **без учителя** задача которого распределить данные на количество заранее заданных кластеров.

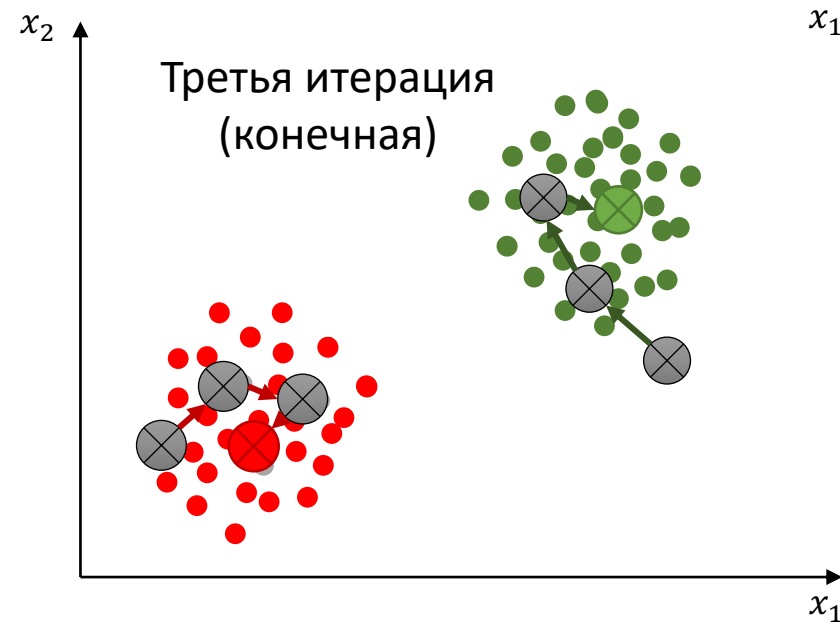
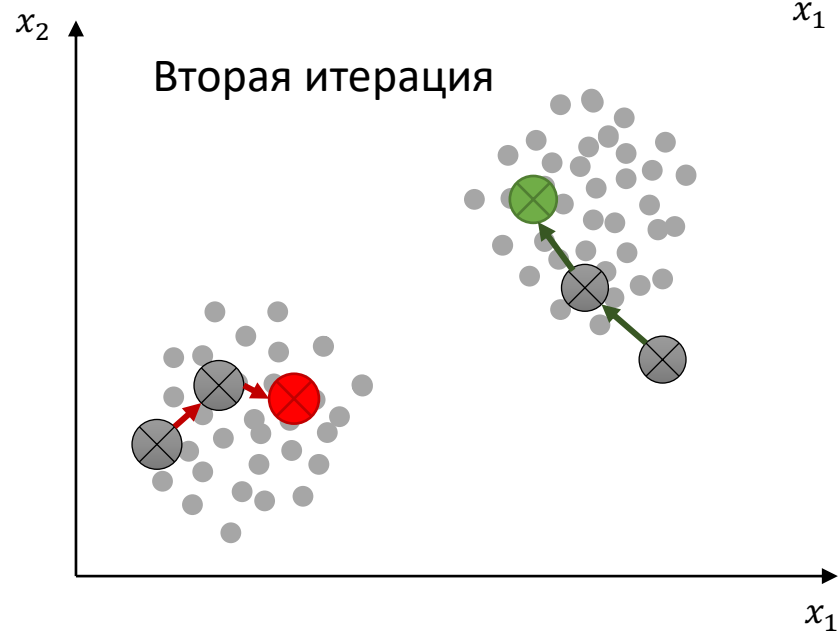
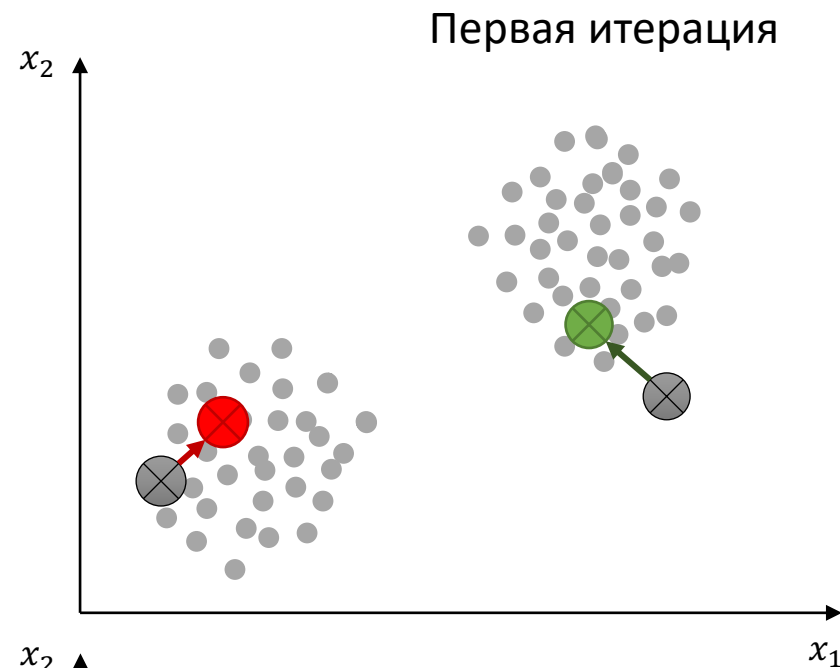
Основная идея данного алгоритма заключается в следующем:

- 1) задаётся требуемое количество кластеров;
- 2) первоначальные координаты центроидов кластеров, в общем случае, генерируются случайно в заданном диапазоне;
- 3) алгоритм поочередно разбивает данные на кластеры, применяя решающее правило ближайшего центроида;
- 4) на каждой итерации разбиения данных выполняется пересчёт координат центроидов.

В общем случае, алгоритм K средних сходится к стационарному состоянию за конечное время, но не гарантирует что найденное состояние является глобальным минимумом, и даже не известно на сколько близко найденные центроиды будут близки к глобальным минимумам.

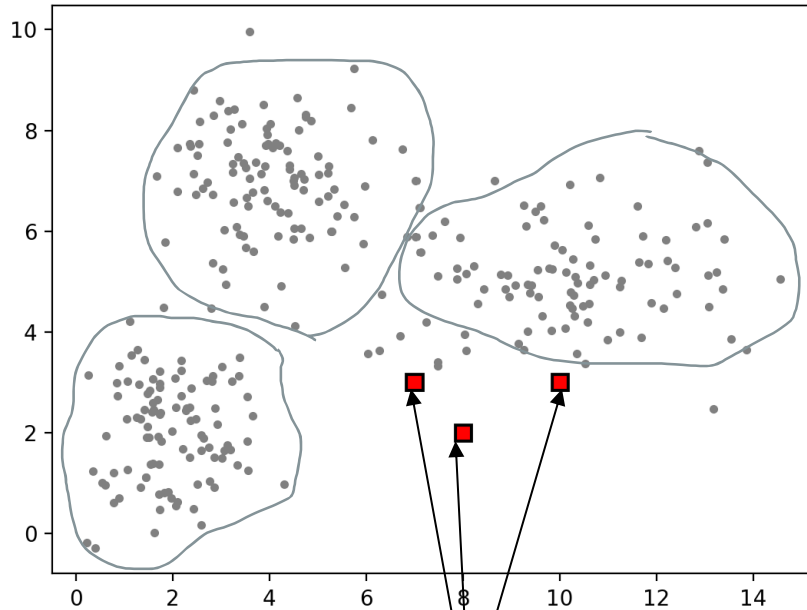
На практике рекомендуется прогнать алгоритм на данных несколько раз и выбрать лучшее решение.

Кластеризация алгоритмом K средних



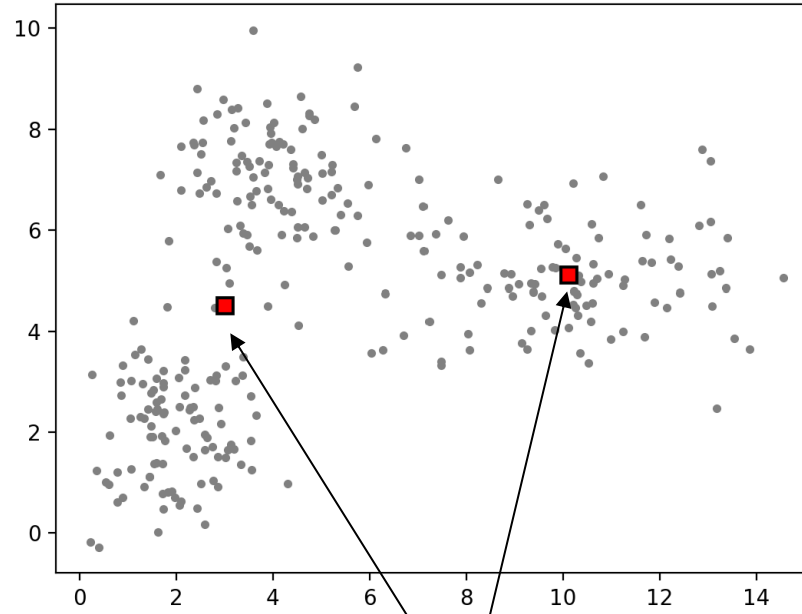
Кластеризация алгоритмом K средних (пример неудачной кластеризации на три кластера)

Произвольные значения центроидов



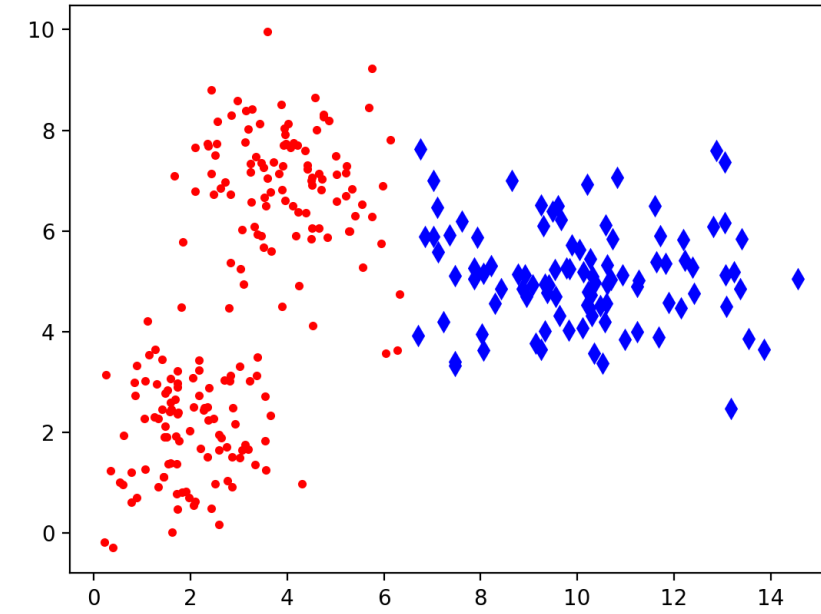
Неудачно сгенерированных
случайные координаты
центроидов

Найденные значения центроидов классификатором



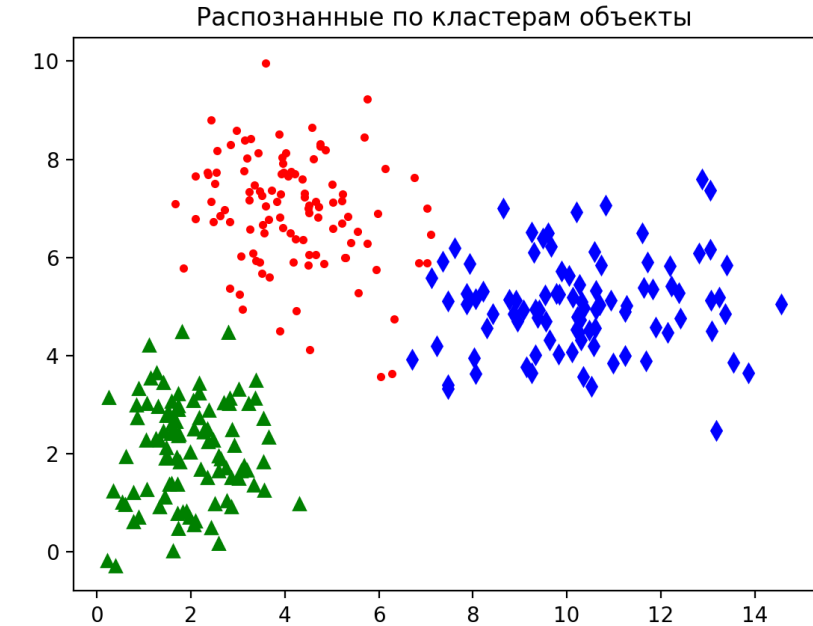
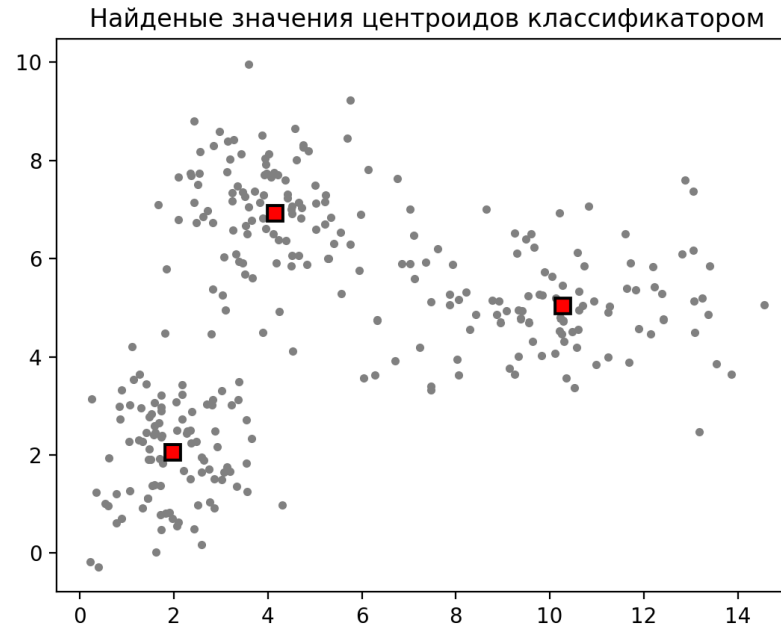
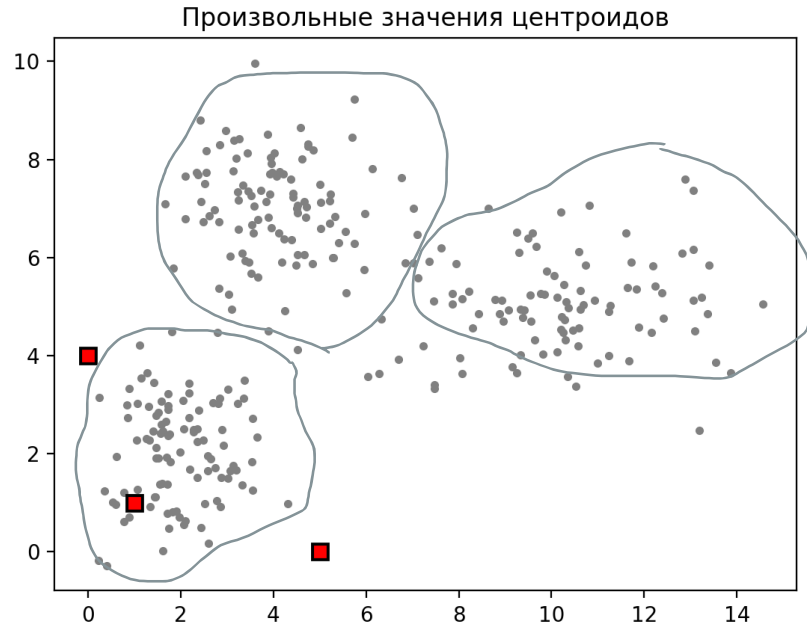
Один из центроидов
выродился

Распознанные по кластерам объекты



В результате данные
были разделены на два
кластера

Кластеризация алгоритмом K средних (пример успешной кластеризации на три кластера)



Казалось бы первоначальные координаты центроидов сгенерированы неудачно. Тем не менее последующие итерации привели к верному вычислению конечных координат центроидов и соответственно к ожидаемому распределению данных.

Формальный алгоритмом K средних

Алгоритм $KMeans(D, K)$ – кластеризация методом K средних с применением евклидова расстояния Dis_2

Вход: данные $\mathbf{x} \in D$, число кластеров K .

Выход: центроиды кластеров $\mu_1, \mu_2, \dots, \mu_K$ и кластеры D_1, D_2, \dots, D_K .

*/*Случайная инициализация $\mu_1, \mu_2, \dots, \mu_K$ */*

for $j \leftarrow 1$ **to** K **do**

$\mu_j \leftarrow \text{random}();$

end

*/*пересчёт $\mu_1, \mu_2, \dots, \mu_K$ и разделение $\mathbf{x} \in D$ на кластеры D_1, D_2, \dots, D_K */*

while $\mu_1, \mu_2, \dots, \mu_K$ не перестанут изменяться **do**

for \mathbf{x} **in** D **do**

$D_j \leftarrow \{\mathbf{x} \in D \mid \mathbf{x} \text{ отнесена к кластеру } j \Leftrightarrow \operatorname{argmin}_j Dis_2(\mathbf{x}, \mu_j)\};$

end

$\mu_j \leftarrow \frac{1}{|D_j|} \sum_{\mathbf{x} \in D_j} \mathbf{x};$

end

return $\mu_1, \mu_2, \dots, \mu_K, D_1, D_2, \dots, D_K;$

Классификация по K ближайшим соседям

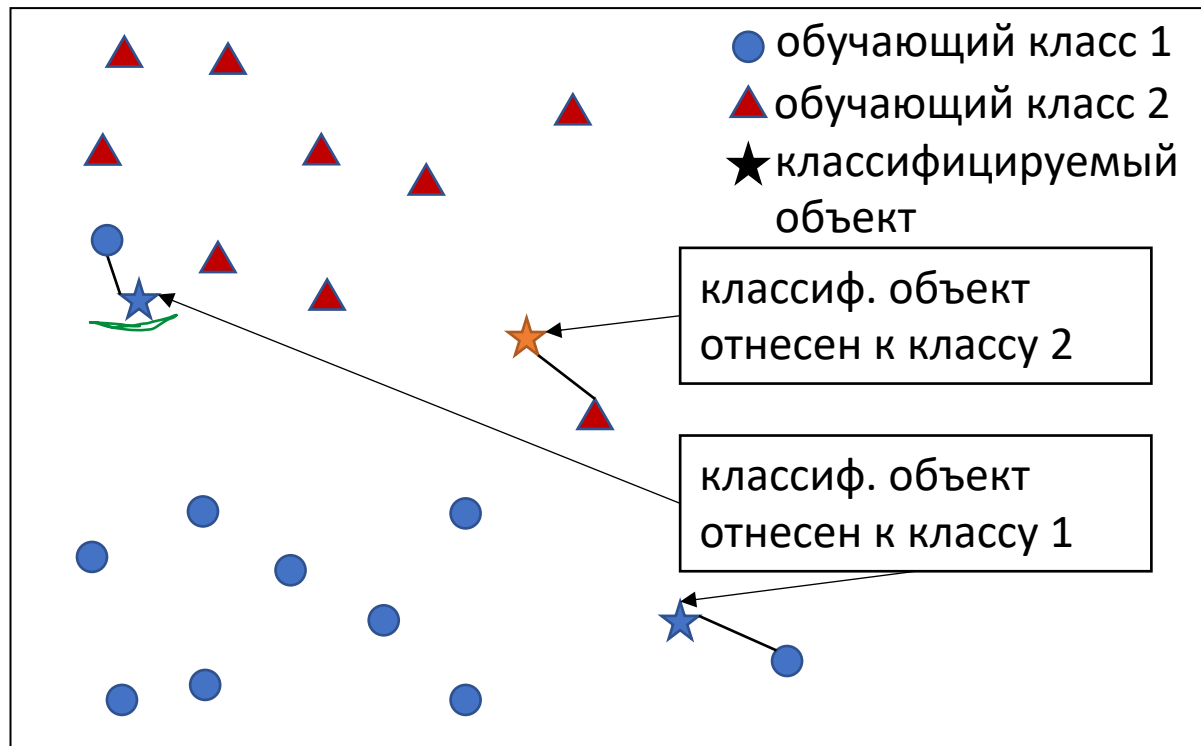
Это алгоритм машинного обучения с **учителем** задача которого выполнить классификацию новых данных по обучающему набору данных .

Самый простейший вариант реализации классификации по k ближайшим соседям сводится к запоминанию всех тестовых данных – это и есть обучение классификатора.

Далее, классификатор по k ближайшим соседям проводит голосование между $k \geq 1$ соседями, ближайшим к классифицированному объекту, и выполняет прогнозирование класса, получившего большинство голосов (так называемый мажоритарный класс).

Классификация по K ближайшим соседям

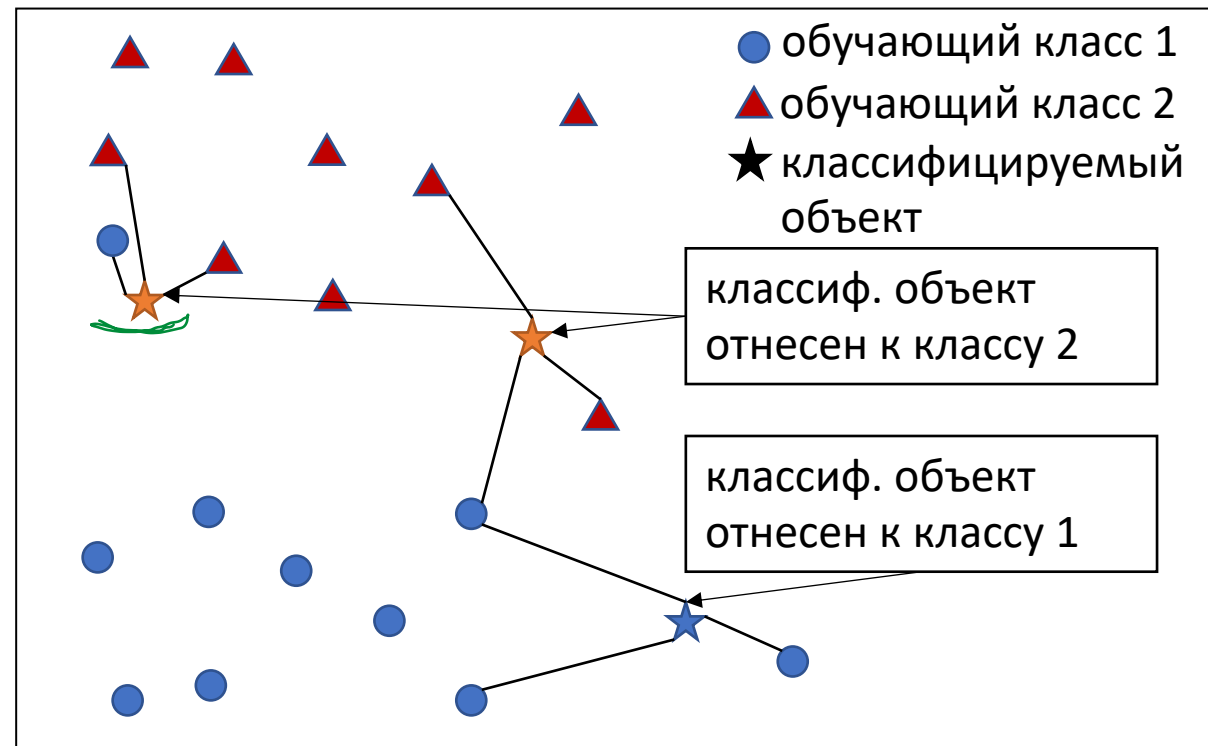
Классификация по **одному** ближайшему соседу



1) Необходимо вычислить расстояния между всеми элементами обучающего набора и классифицируемым объектом;

- 2) найти минимальное расстояние между классифицируемым объектом и объектом обучающего набора и получить метку его класса;
3) отнести классифицируемый объект к соответ. классу.

Классификация по **трём** ближайшим соседям



- 2) найти **3-ри** минимальных расстояний между классифицируемым объектом и объектами обучающего набора;
3) выполнить «голосование» и получить метку класса большинства объектов, т.е. так называемый «мажоритарный класс»;
3) отнести классифицируемый объект к соответ. классу.

Особенности и варианты классификация по K ближайшим соседям

- 1) При $K = 1$ модель на обучающем наборе как правило не допускает ошибок, но это потому что модель будет переобученной.
- 2) При увеличении числа соседей K обобщающая способность у модели машинного обучения в начале улучшается, но с дальнейшим увеличением числа K соседей модель становится недообученной.

Не существует правила, которое позволило бы однозначно сказать сколько соседей нужно использовать для конкретного набора данных, чтобы обеспечить хорошую обобщающую способность модели.

Но можно улучшить ситуацию, если подсчитывать не просто количество голосов, а проводить ещё и взвешивание голосов: то есть чем ближе сосед находится к классифицируемому объекту, тем его голос более весомее. Например, можно вычислять величину важности голоса как обратную величине расстояния от соседа до классифицированного объекта. Так как веса голосов убывают по мере увеличения расстояния от соседа до классифицируемого объекта, то эффект от увеличения количества голосующих соседей уменьшается.

Проклятие размерности. Основная проблема классификатора по K ближайшим соседям связана с тем, что такой классификатор плохо работает с входными данными большой размерности. Это получило название проклятие размерности.

Количественная оценка правильности модели

Существуют различные показатели количественной оценки правильности модели машинного обучения.

Приведем одну из таких оценок, которая используется в первую очередь – **верность, правильность (accuracy)**

$$acc = \frac{1}{|Te|} \sum_{\mathbf{x} \in Te} I[\hat{c}(\mathbf{x}) = c(\mathbf{x})]$$

где Te – тестовый набор данных;
 $I[...]$ - индикаторная функция, которая равна 1, если аргумент принимает истинное значение и 0, если аргумент принимает ложное значение;
 $\hat{c}(\mathbf{x})$ - функция возвращающая оценочную метку класса для классифицируемого объекта \mathbf{x} ;
 $c(\mathbf{x})$ - функция возвращающая истинную метку класса классифицируемого объекта \mathbf{x} .

Использованные информационные источники

1. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2015. – 400 с.: ил.
2. Мэрфи К. П. Вероятностное машинное обучение: введение / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2022. – 990 с.: ил.
3. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными.: Пер. с англ. - СПб.: ООО "Альфа-книга", 2017. - 480 с.: ил.