

# Категоризация объектов



- Объектов бесконечное множество, а сколько существует классов объектов?
- 1500-3000 основных существительных, ~10 подкатегорий
- Выделяют и больше – до 100 000 категорий

# Классификация объектов

---



## Задачи классификации изображений:

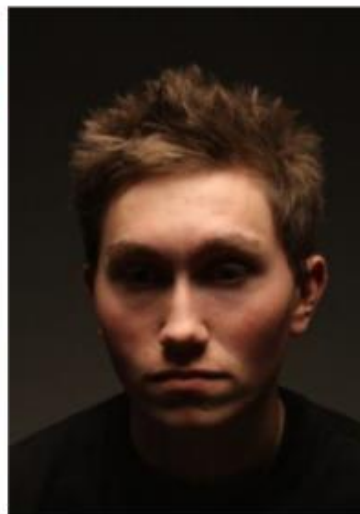
- ✓ Есть ли на изображении объект заданного класса
- ✓ Относится ли изображение к заданному классу



# Сложность задачи



изменчивость угла зрения



изменчивость освещения, теней, текстур



изменчивость масштаба  
/размеров

# Сложность задачи



изменчивость формы

Закрытия



изменчивость позы





Viola & Jones system, CVPR 2001



«Обычный» фотоаппарат сегодня

# Локализация пешеходов

Пример «нерешенной» задачи

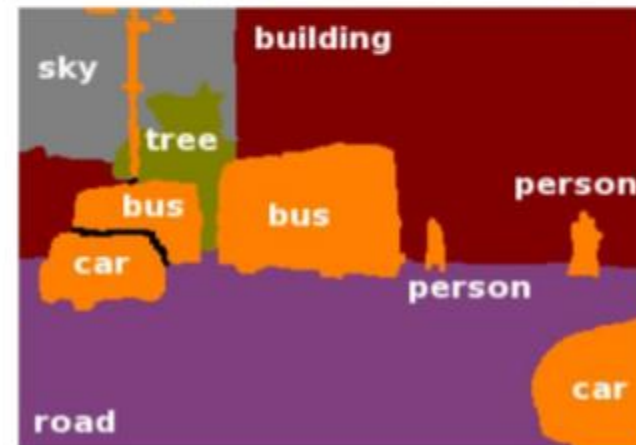


Enzweiler & Gavrila. Monocular Pedestrian Detection: Survey and Experiments. TPAMI'09

- Недостаточная скорость/аккуратность
- можно улучшить за счёт стерео камер и анализа видео

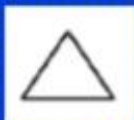
# Схожие задачи

1. Нахождение конкретных объектов
2. Локализация для видео
3. Другие модальности:
  - Спутниковые снимки
  - Медицинские изображения
  - Стерео/range maps
4. Семантическая сегментация
5. Оценка конфигурации  
(выражение лица, поза человека...)



*Gould, Fulton, Koller, ICCV'09*

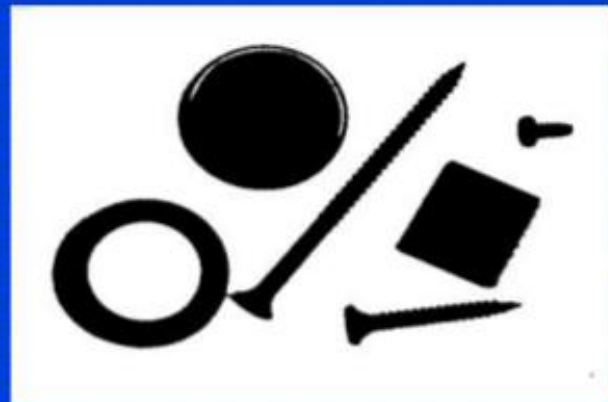




Поиск шаблонов



Использование  
локальных  
особенностей

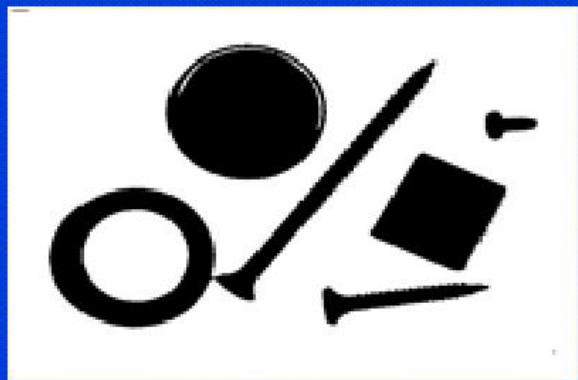


Инвариантные  
признаки для  
выделенных  
сегментов

- Все эти методы нашли своё применение, но решить задачу распознавания не получилось

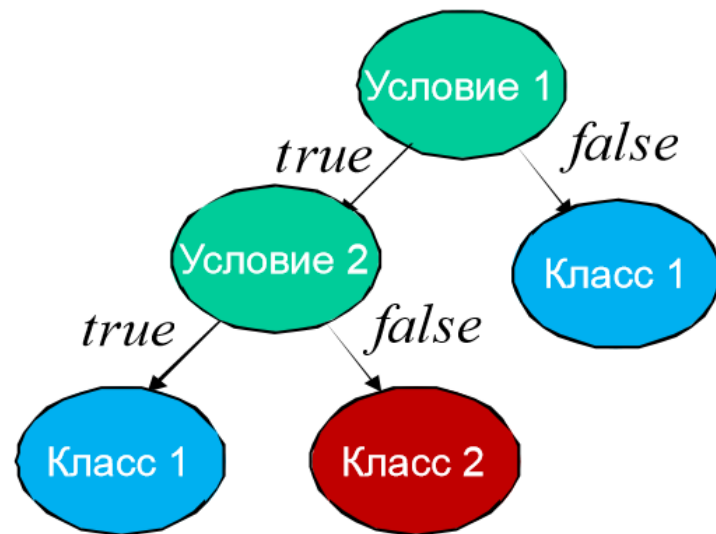


# Проблемы распознавания



Геометрические  
признаки для  
выделенных  
сегментов

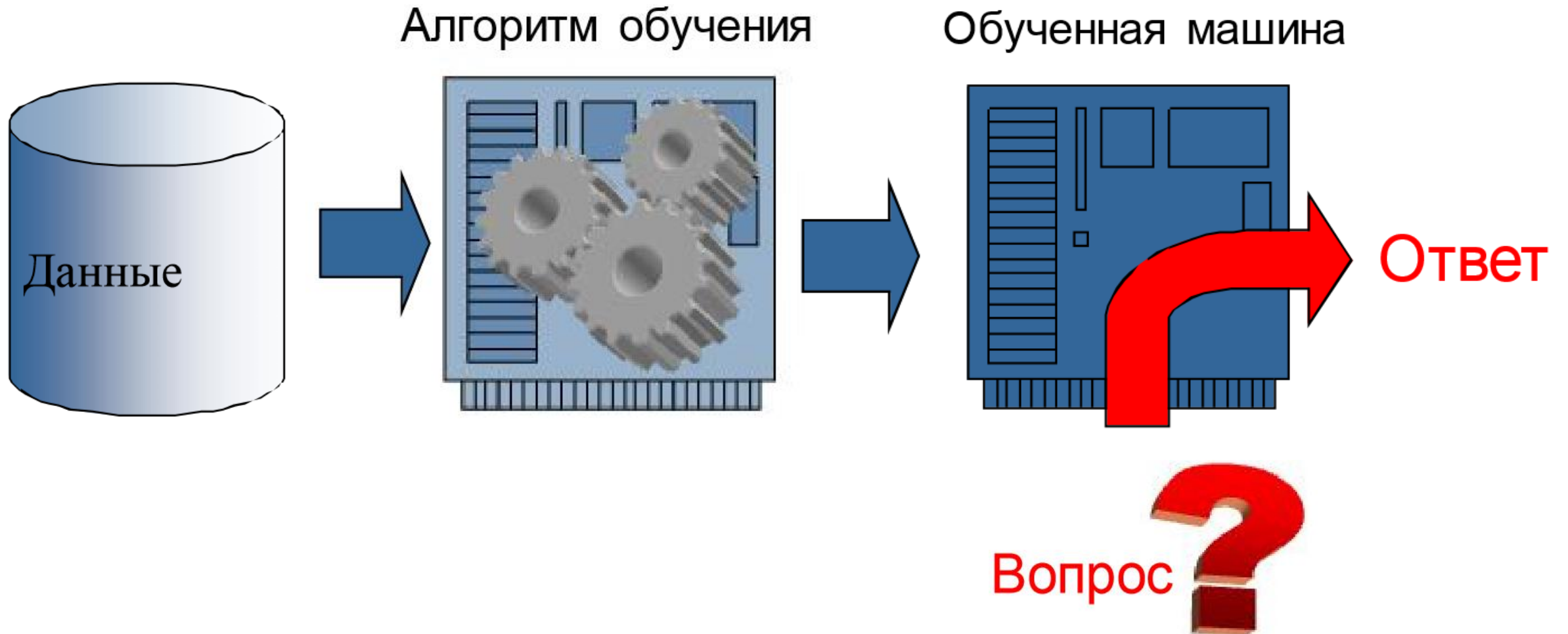
В чём причины  
ограниченности успехов  
этого подхода?



- Правила приходится подбирать вручную
- Признаки нужно использовать «осмысленные» и очень информативные
- Таких признаков мало, и сложные комбинации признаков обработать уже нельзя

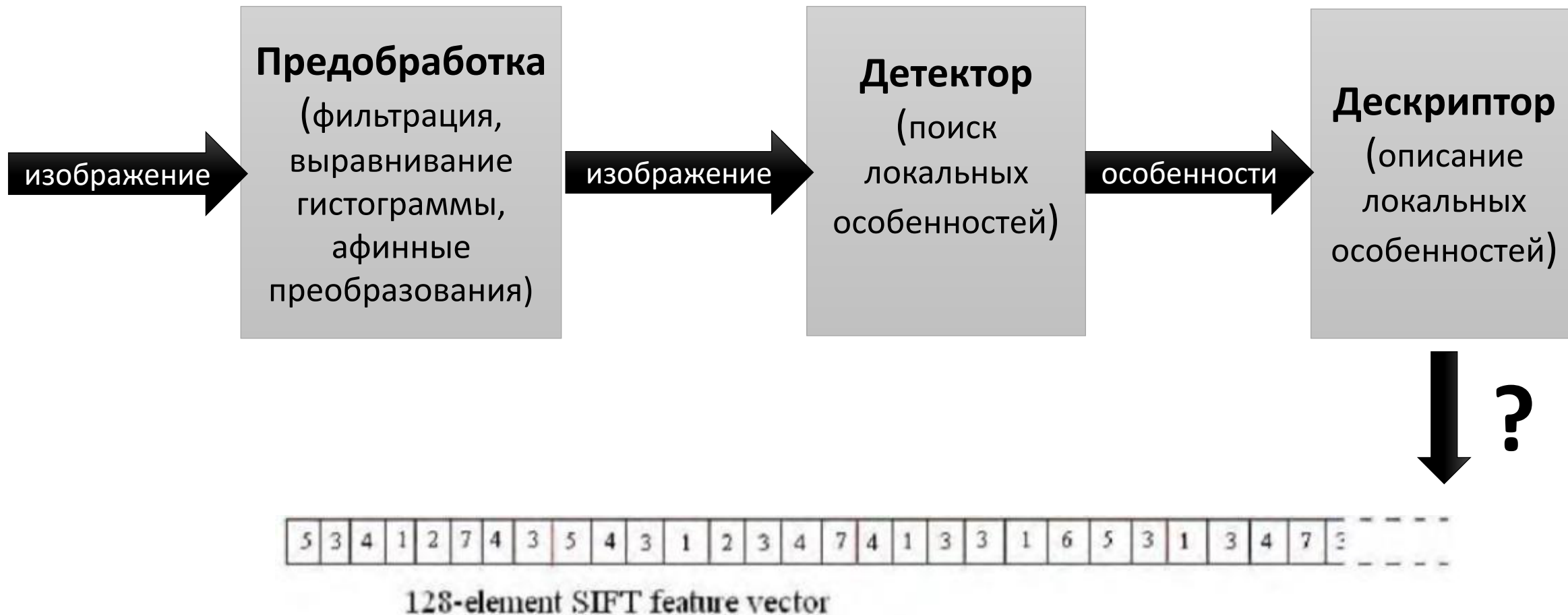
# Что хотим в идеале?

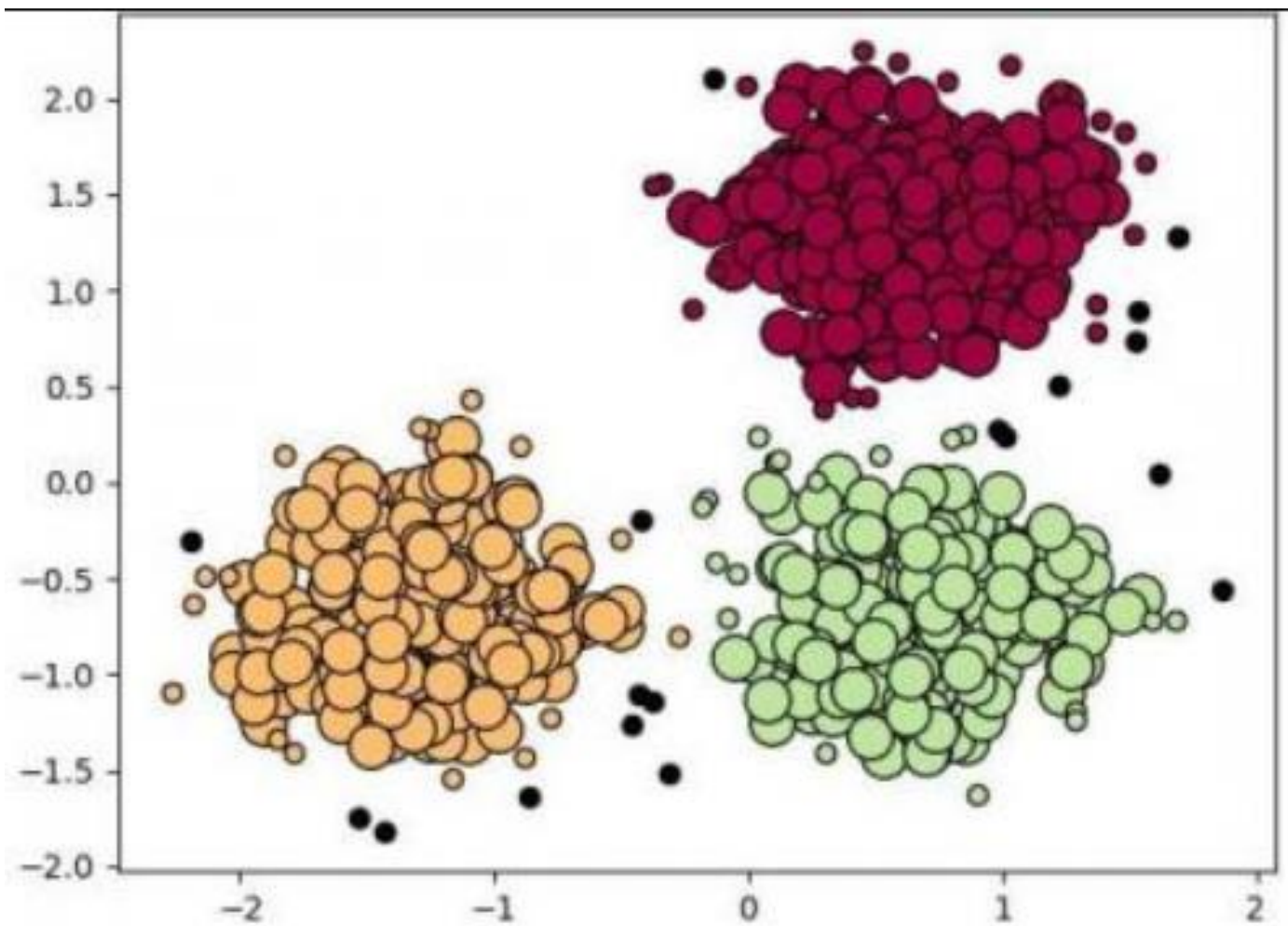
---





# Классификация (идентификация) объектов на изображении





А если пространство 128-мерное?!



# Машинное обучение

# Что такое машинное обучение?

---

- Обучение  $\neq$  «заучивание наизусть»
- Мы хотим научить машину делать выводы!
- Машина должна корректно работать на новых данных, которые мы ей раньше не давали
- По конечному набору обучающих данных машина должна научиться делать выводы на новых данных





# Машинное обучение

---

Разные методы исследовались много лет

*Нейронные сети...*

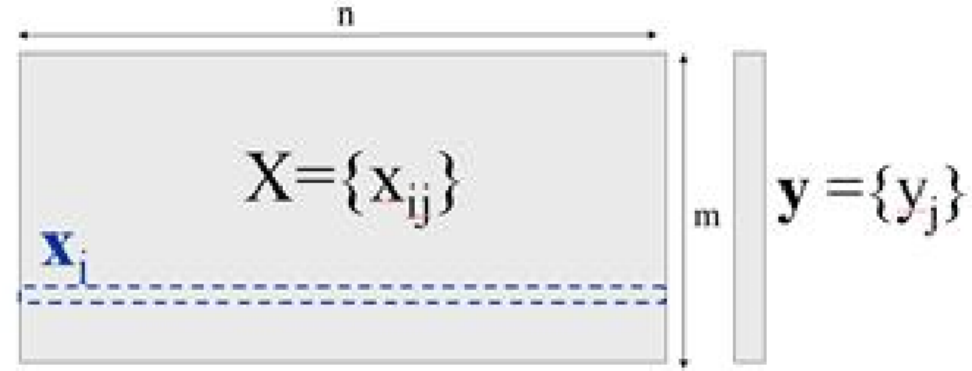
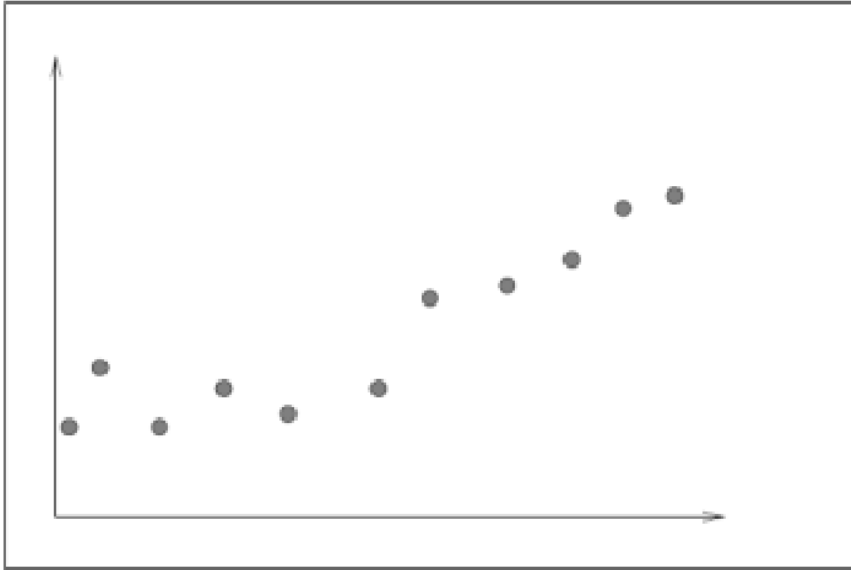
В середине 90х – начале 2000х было предложено несколько мощных методов. Их появление привело к взрывному прогрессу в области анализа данных и компьютерного зрения в частности

Говорят, что компьютерная программа обучается на основе опыта  $E$  по отношению к некоторому классу задач  $T$  и меры качества  $P$ , если качество решения задач из  $T$ , измеренное на основе  $P$ , улучшается с приобретением опыта  $E$ .

*T.M.Mitchell Machine Learning. McGraw-Hill, 1997.*

# Задача машинного обучения

---

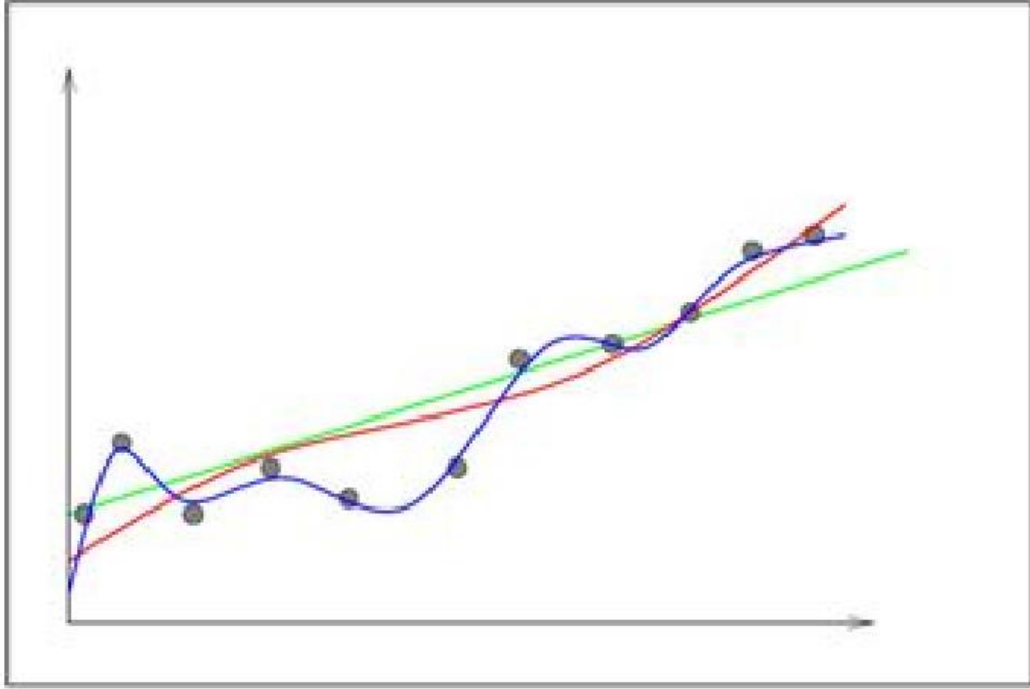


Обучающую выборку  
удобно записывать в  
виде матрицы

- В нашем распоряжении есть конечное число данных – обучающая выборка
- Каждый элемент описываем набором признаков  $x$  («**векторпризнак**»)
- Для каждого вектора параметров  $x$  известен ответ  $y$

# Задача машинного обучения

---



Требуется сконструировать функцию  $y=f(x)$  от вектора признаков  $x$ , которая выдает ответ  $y$  для любого возможного наблюдения  $x$

Требование: построенная функция должна хорошо работать на новых данных.



# Классификация

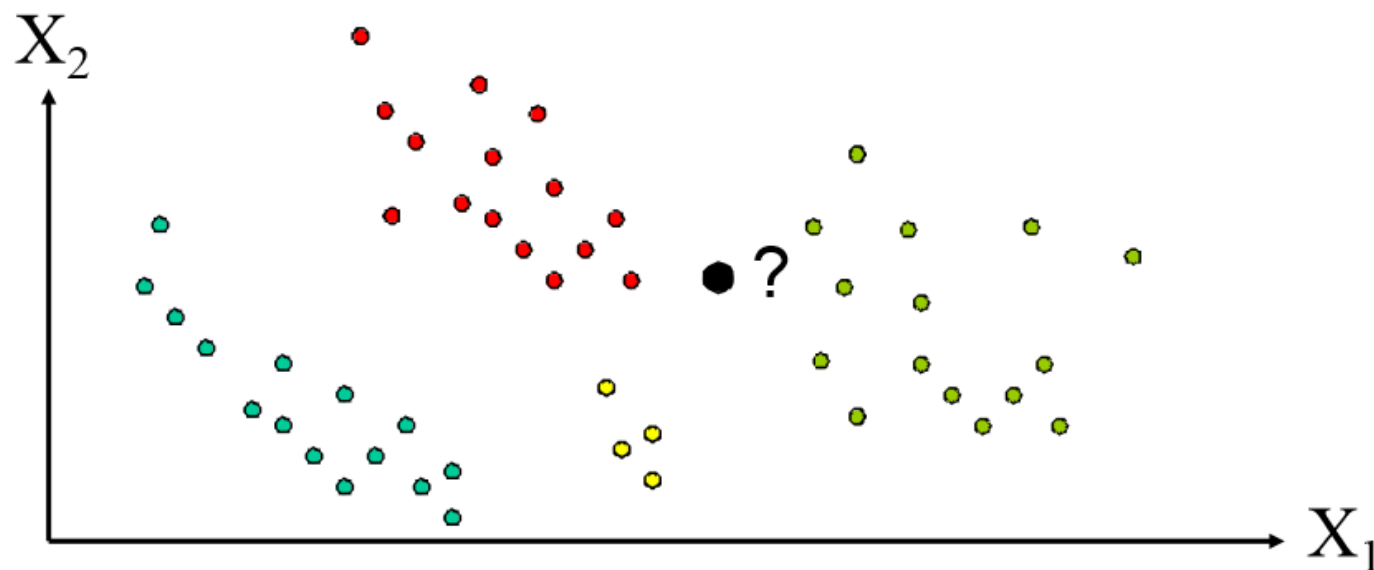
---

Дана обучающая выборка

$$X_m = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\} \quad (\mathbf{x}_i, y_i) \in \mathbf{R}^m \times Y, \quad Y = \{1, \dots, K\}$$

Объекты принадлежат одному из K классов.

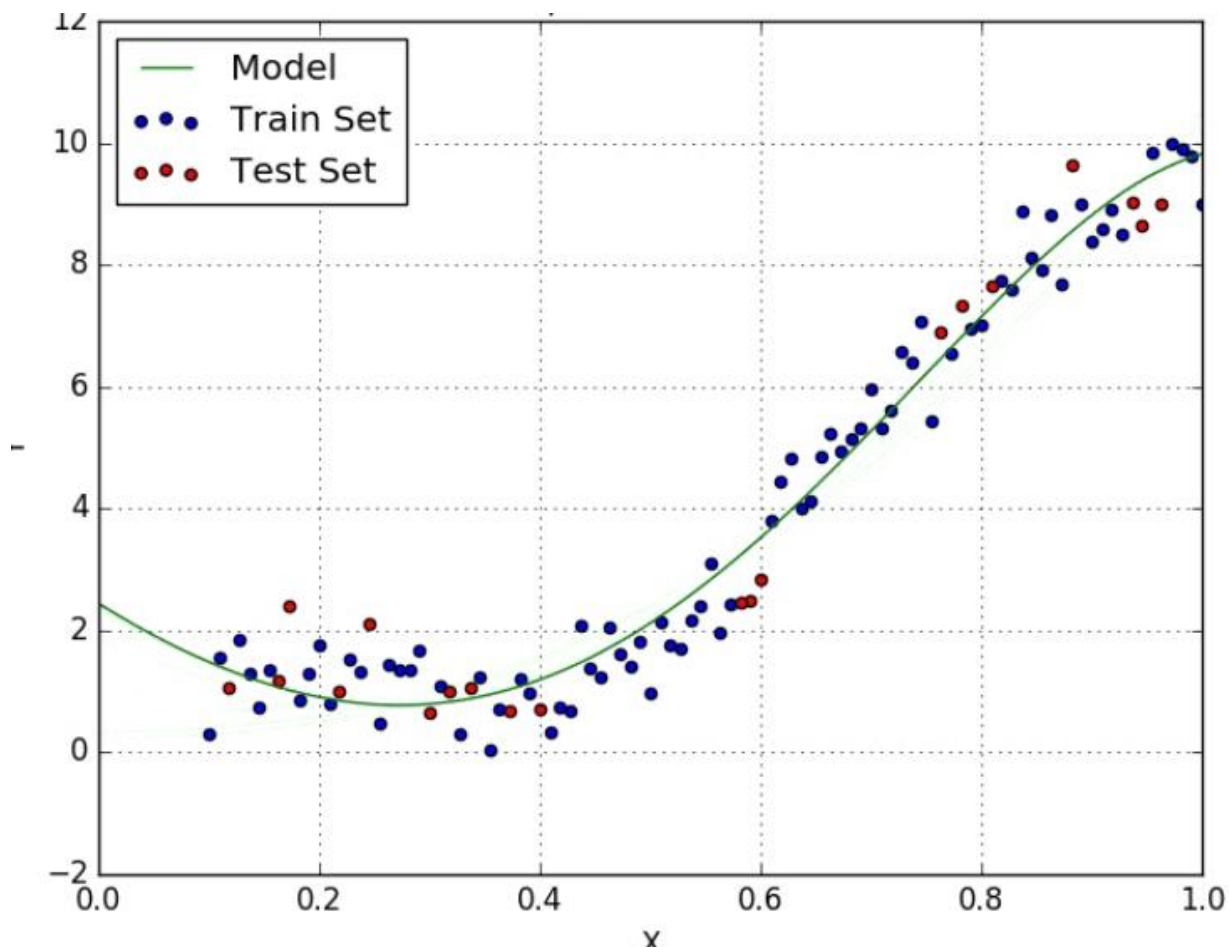
Цель: для всех новых значений  $\mathbf{x}$  определить класс и поставить метку от 1 до K



# Регрессия

- Дана обучающая выборка

$$X_m = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\} \quad (\mathbf{x}_i, y_i) \in \mathbf{R}^m \times Y, \quad Y = \mathbf{R}$$



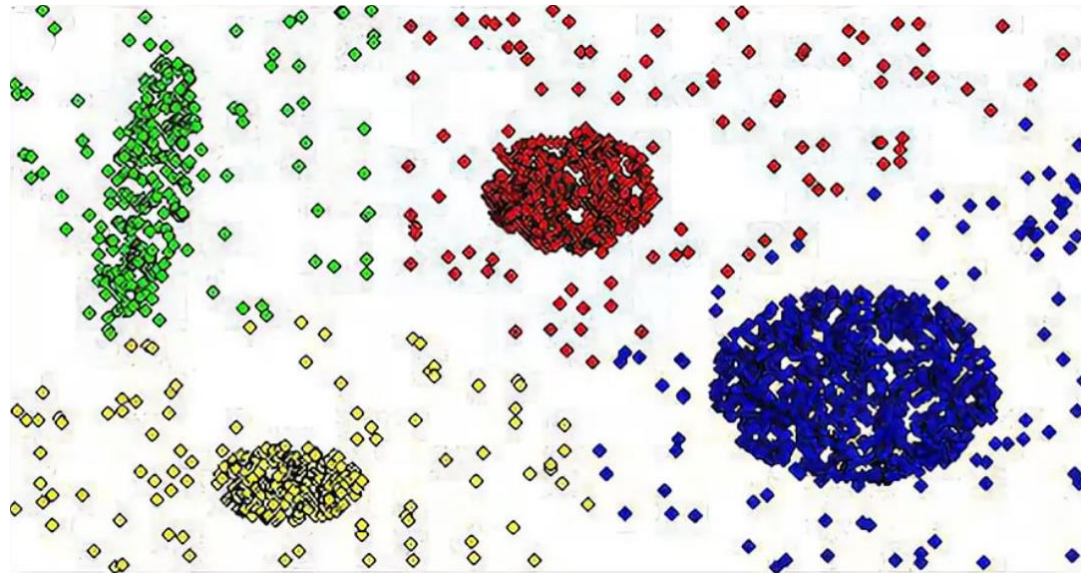
Если  $y$  – непрерывная величина. Такая задача называется задачей **регрессии** (одна из основных задач машинного обучения)

# Обучение без учителя

---

В задачах обучения без учителя (unsupervised learning) у объектов не известны выходы, и требуется найти некоторые закономерности в данных. К задачам обучения без учителя относят задачи кластеризации

**Задача кластеризации** – это задача разбиения заданного набора объектов на непересекающиеся подмножества (кластеры), т. е. группы близких по своему признаковому описанию объектов. «Похожие» друг на друга объекты должны входить в один кластер, «не похожие» объекты должны попасть в разные кластеры.

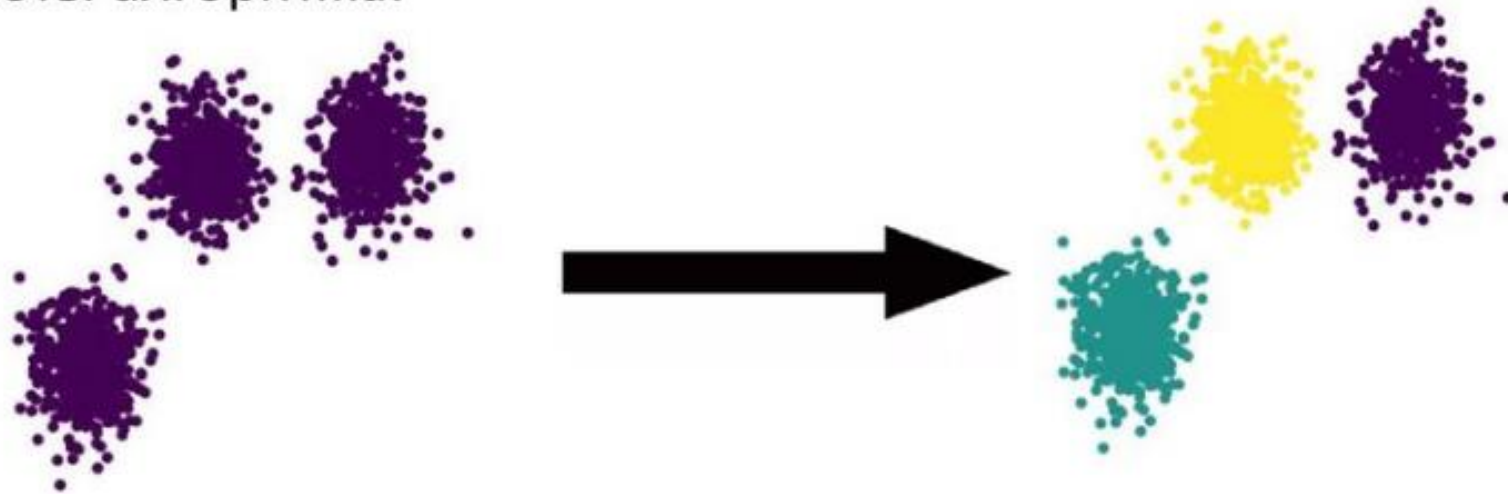




# Кластеризация

---

- Главное отличие кластеризации от классификации состоит в том, что перечень групп четко не задан и определяется в процессе работы алгоритма.

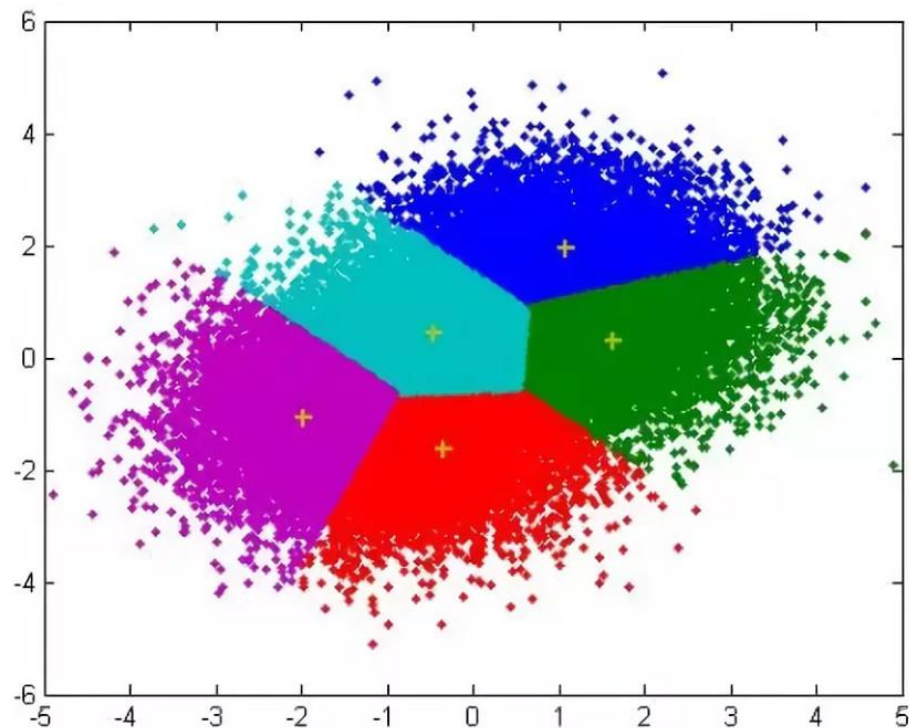


# Метод центров тяжести (k-means)

---

Для решения задачи кластеризации в библиотеки OpenCV реализован метод центров тяжести (k-means).

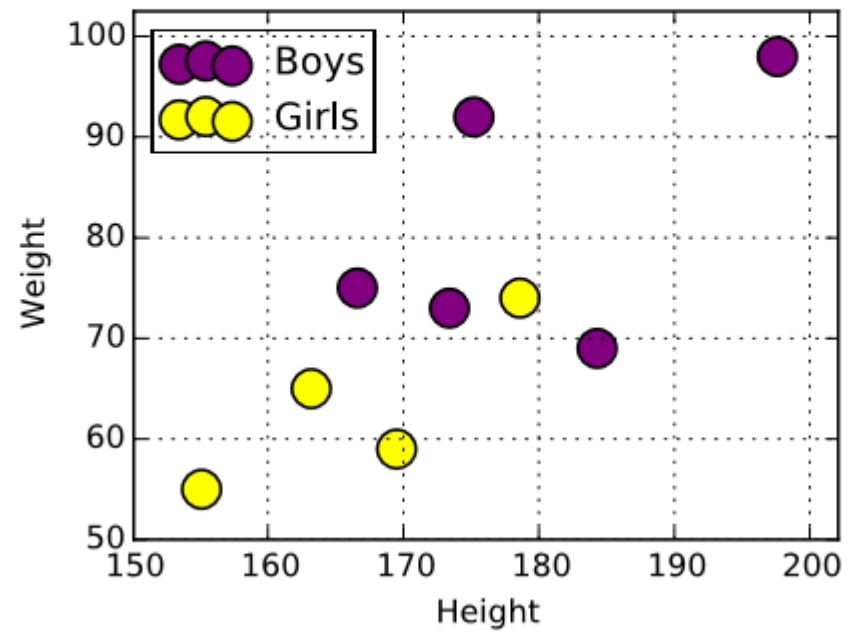
Метод центров тяжести разбивает выборку на заданное количество кластеров путем выбора их центров. Поиск центров кластеров производится из соображений минимизации суммарного расстояния от каждой точки до ближайшего центра с помощью метода локальной оптимизации.



# Задача

---

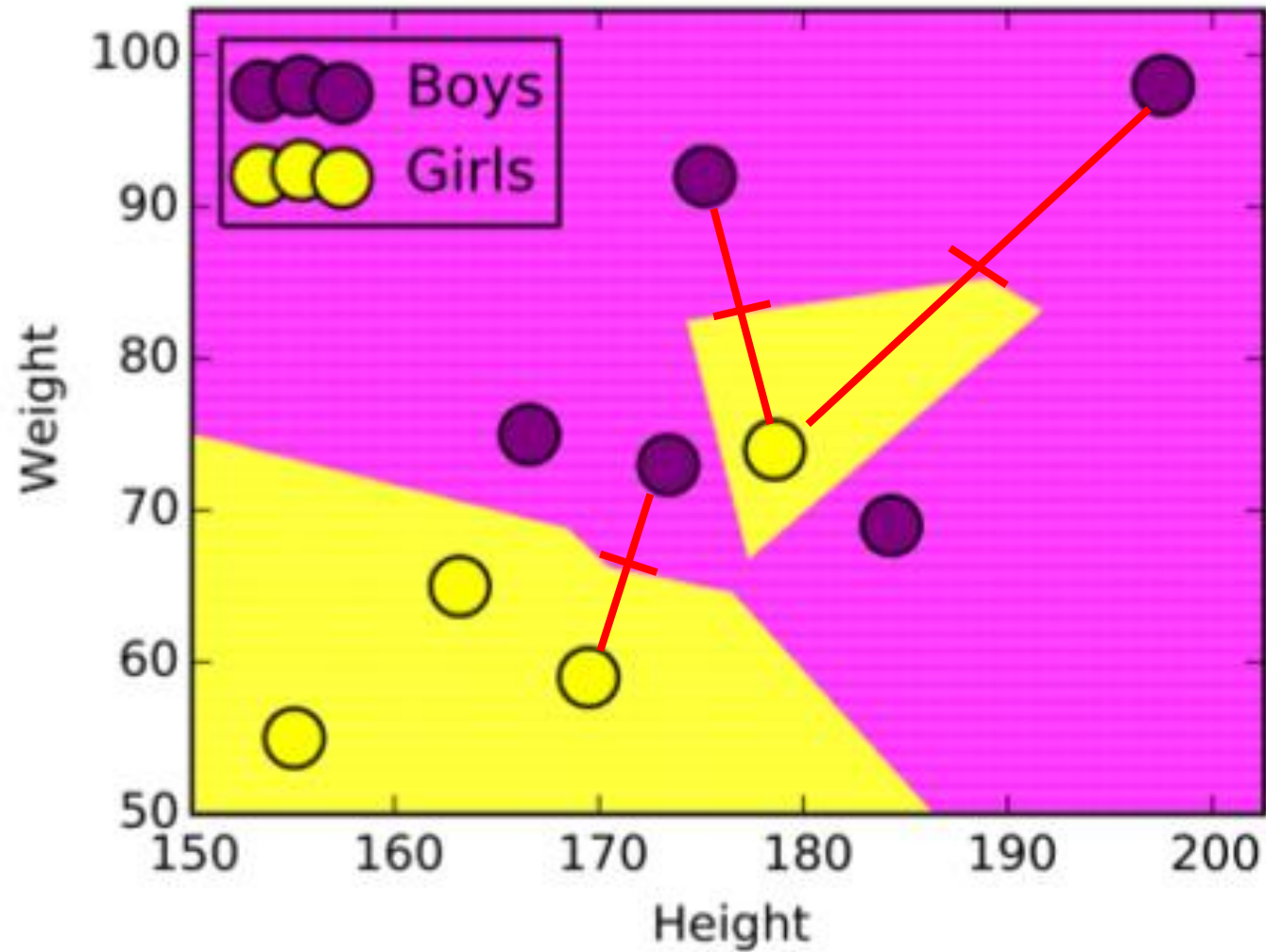
Определение пола по росту и весу





# Метод k-ближайших соседей

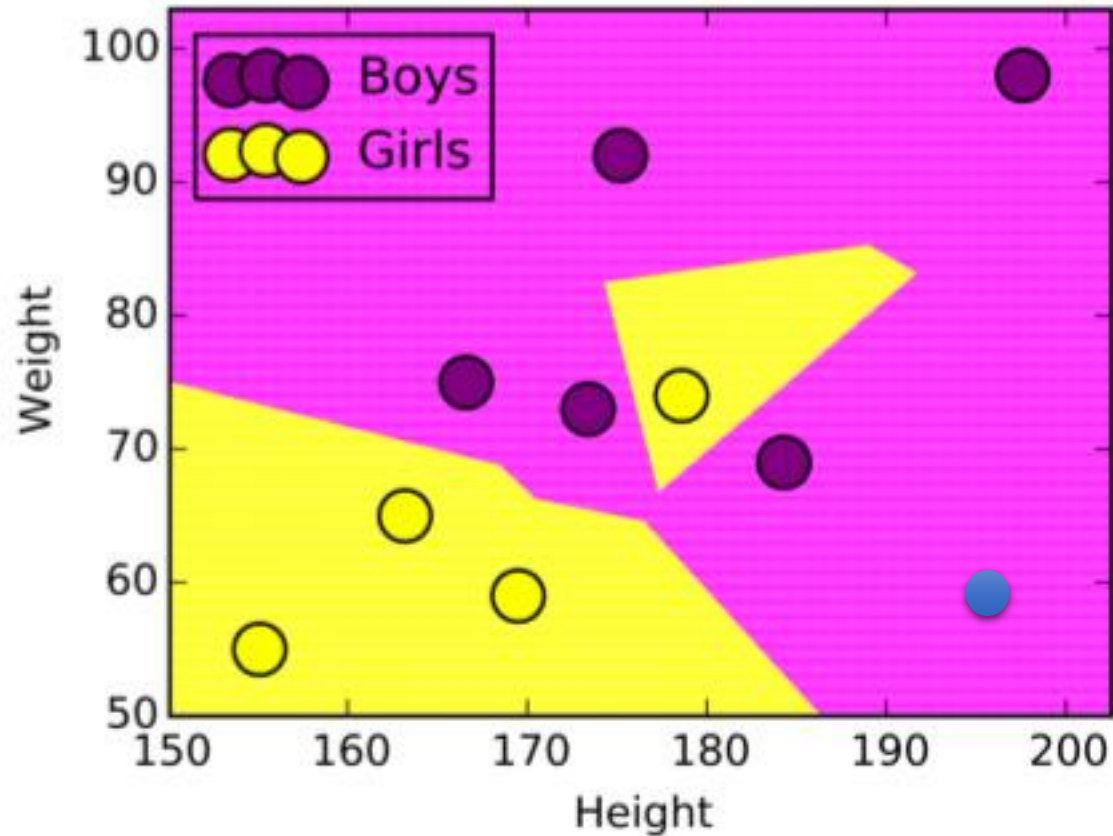
---



$k = 1$

# Метод k-ближайших соседей

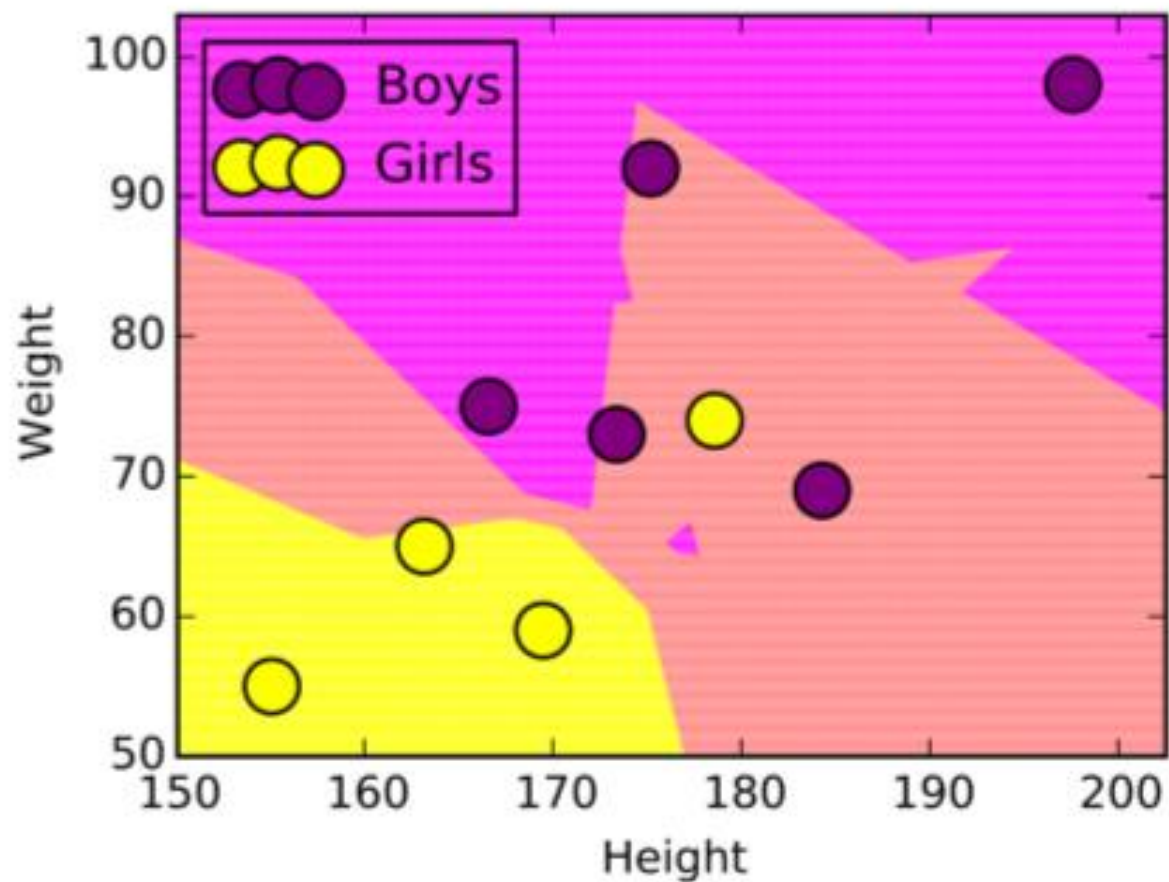
---



В случае классификации новый объект классифицируется путем отнесения его к классу, являющемуся преобладающим среди ближайших (в пространстве признаков) объектов из обучающей выборки.

# Метод k-ближайших соседей

---

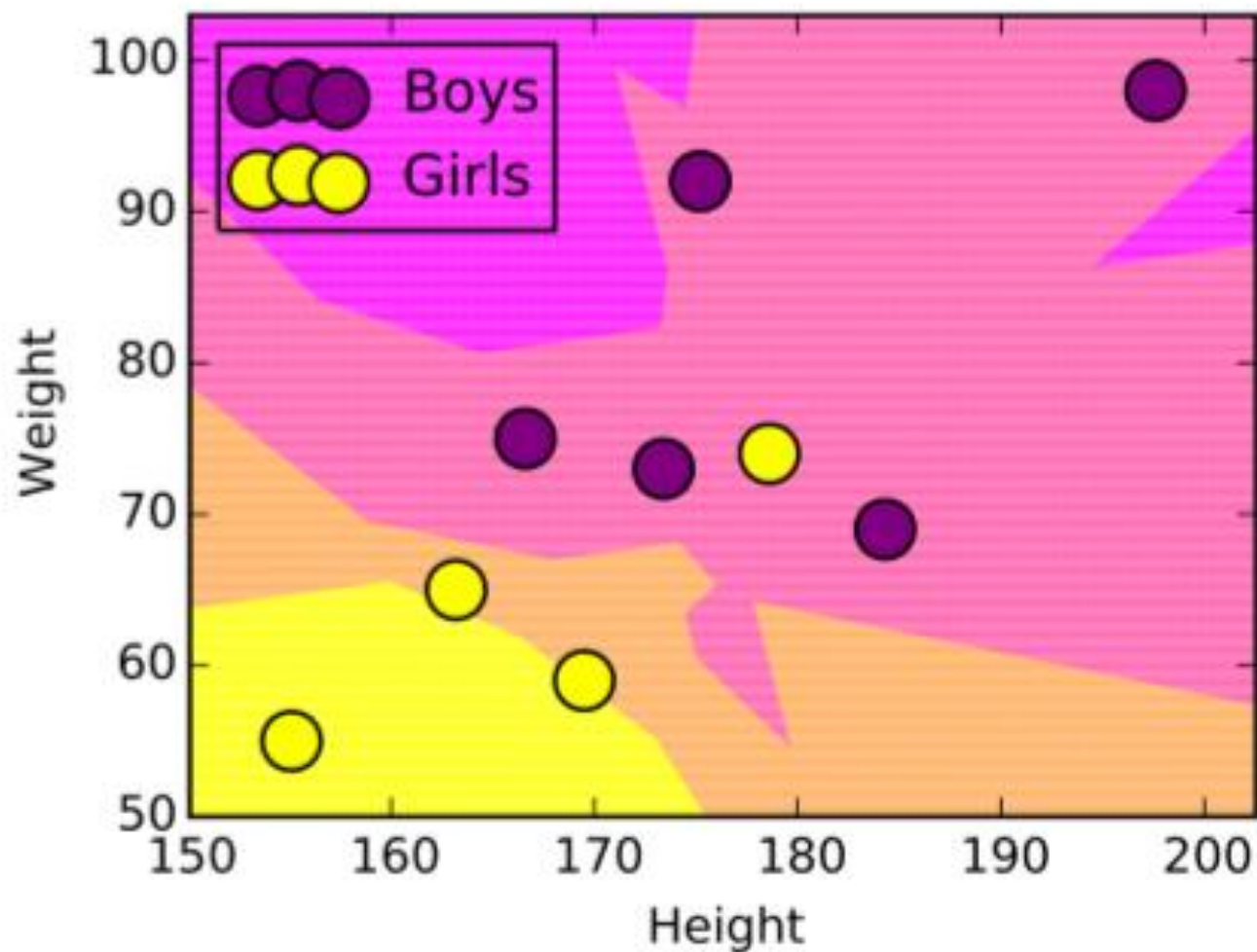


$k = 2$



# Метод k-ближайших соседей

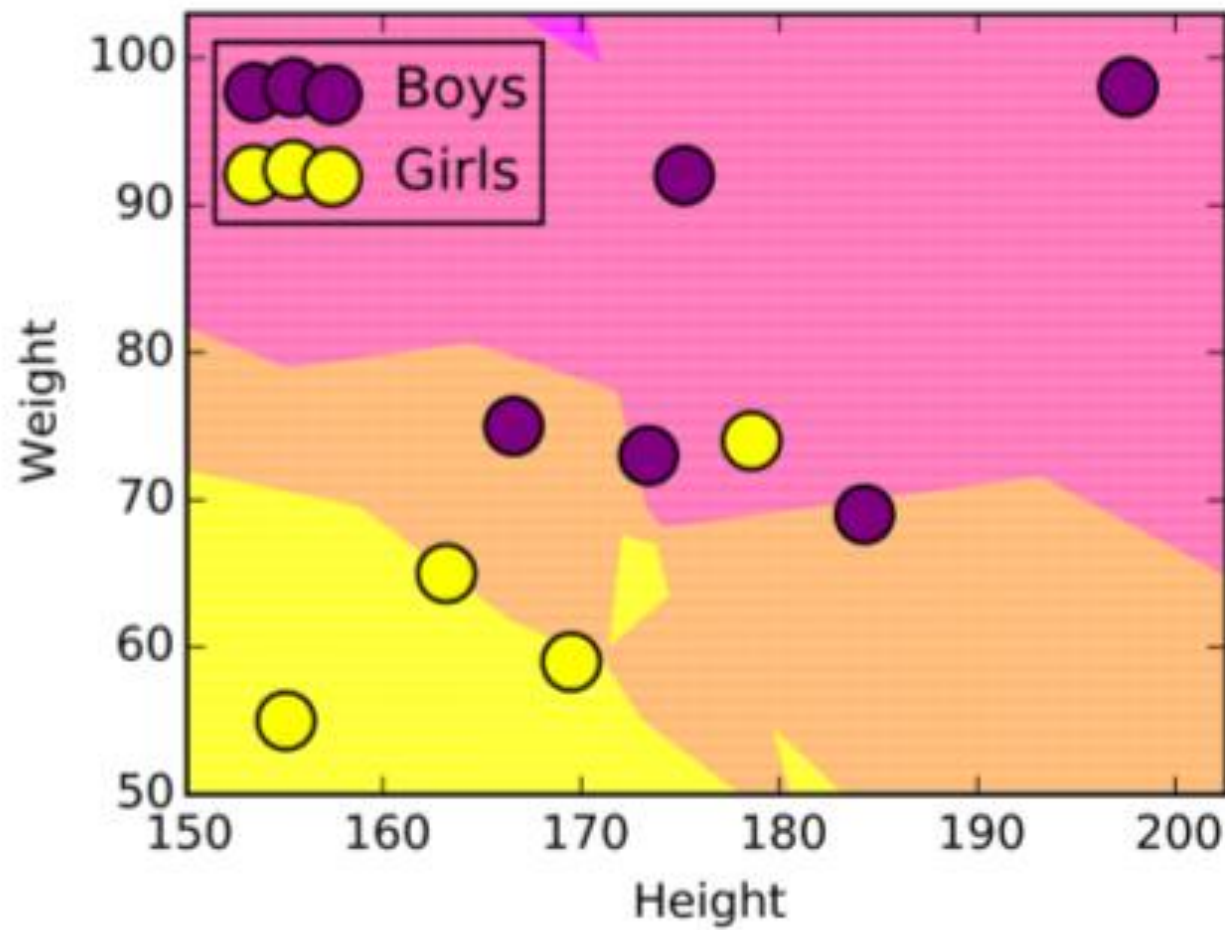
---



$k = 3$

# Метод k-ближайших соседей

---



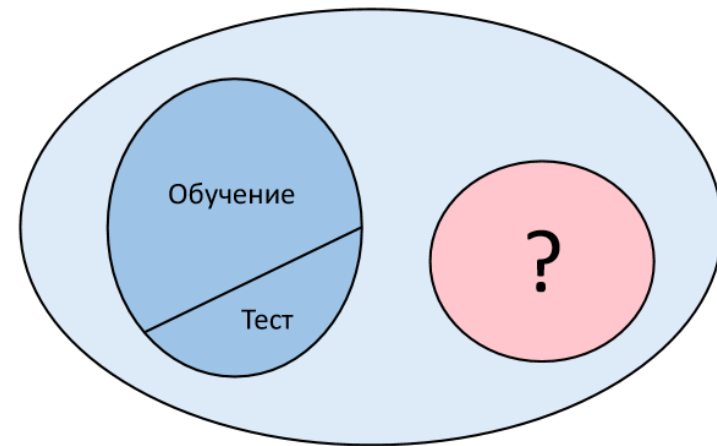
$k = 4$

Возникает вопрос «Как оценить качество алгоритма?»

Для этого, все имеющиеся данные разбивают на обучающую и тестовую выборки.

Обучение производится с использованием обучающей выборки, а оценка качества предсказания на основе данных тестовой выборки.

Все люди на Земле

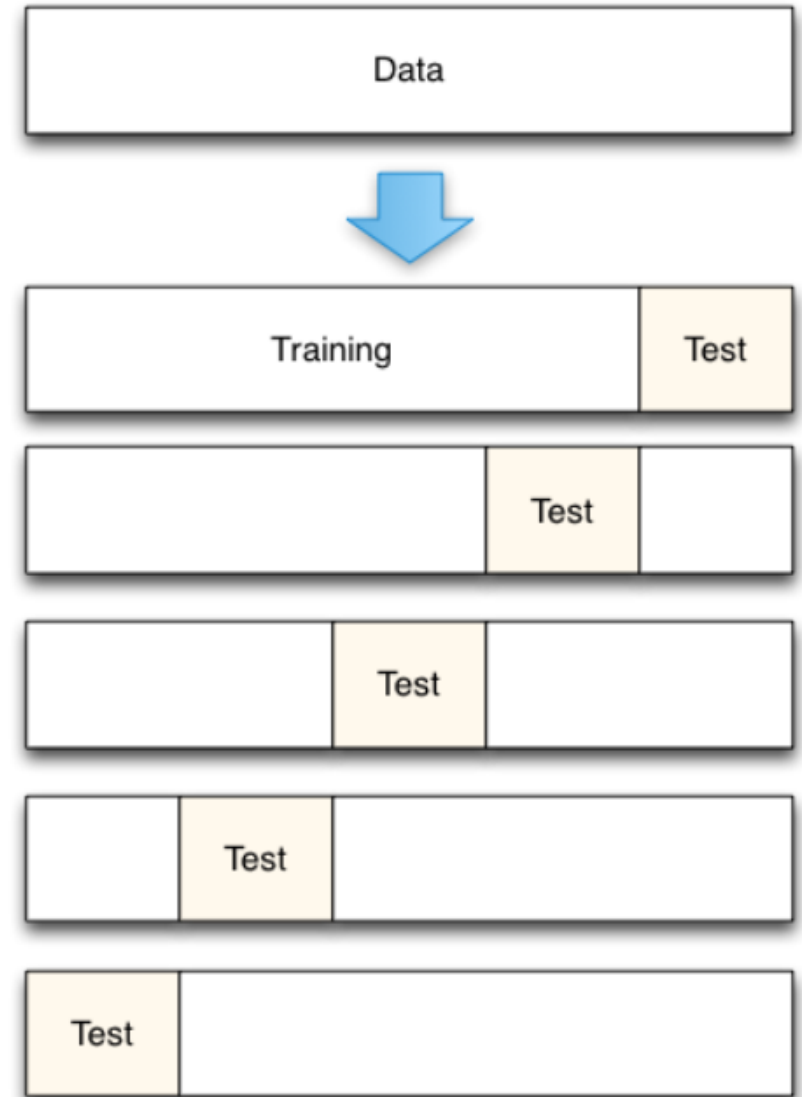


# Скользящий контроль

---

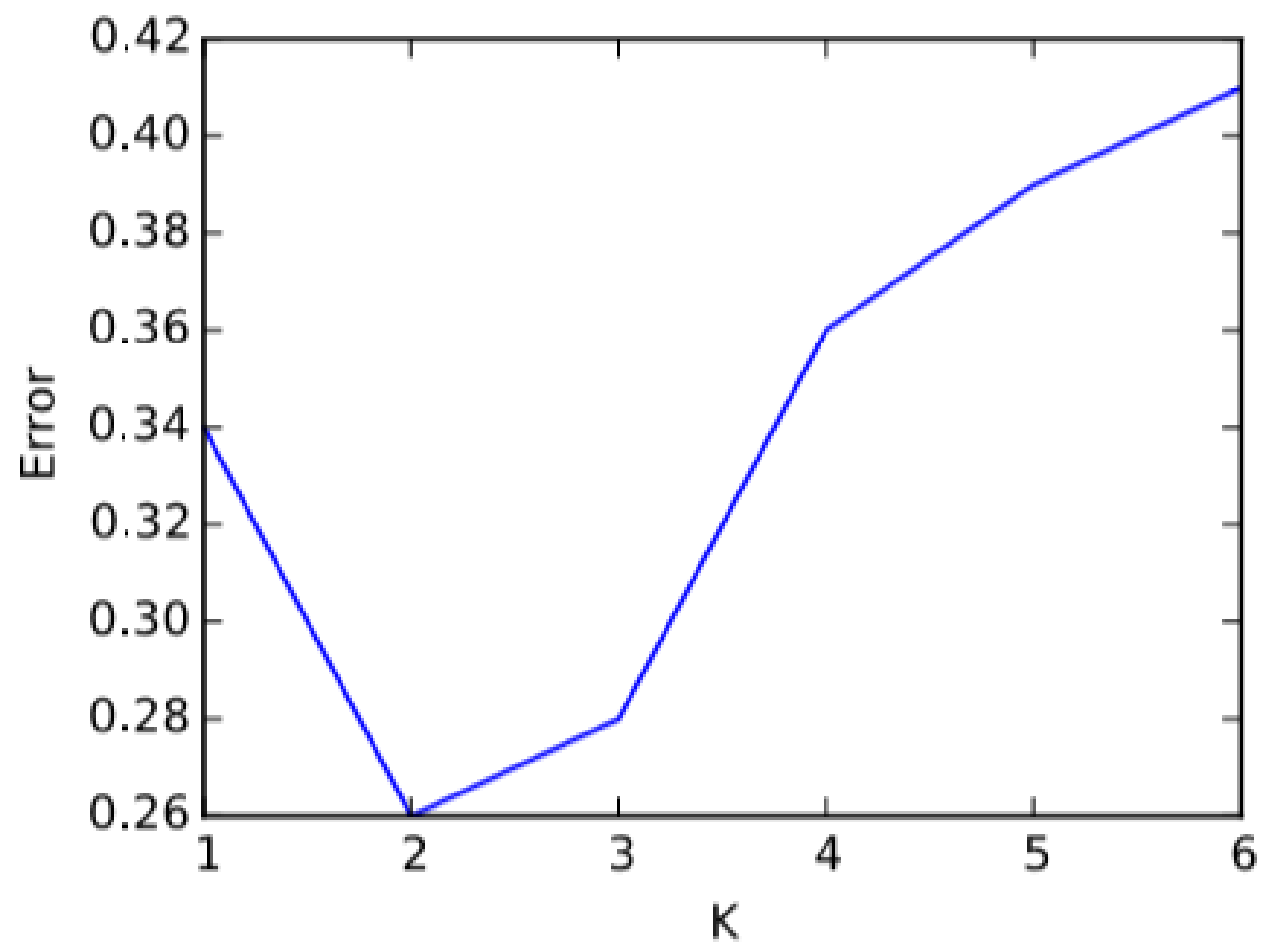
Другим практическим методом оценки обобщающей способности решающего правила является метод кросс валидации (**CV – cross validation**).

Все имеющиеся данные разбиваются на  $n$  примерно равных частей. Далее, отделяя из выборки одну за другой каждую из этих частей, используют оставшиеся данные (составленные из частей) как обучающую выборку, а отделенную часть – как тестовую. Итоговая оценка ошибки определяется как средняя по всем разбиениям.

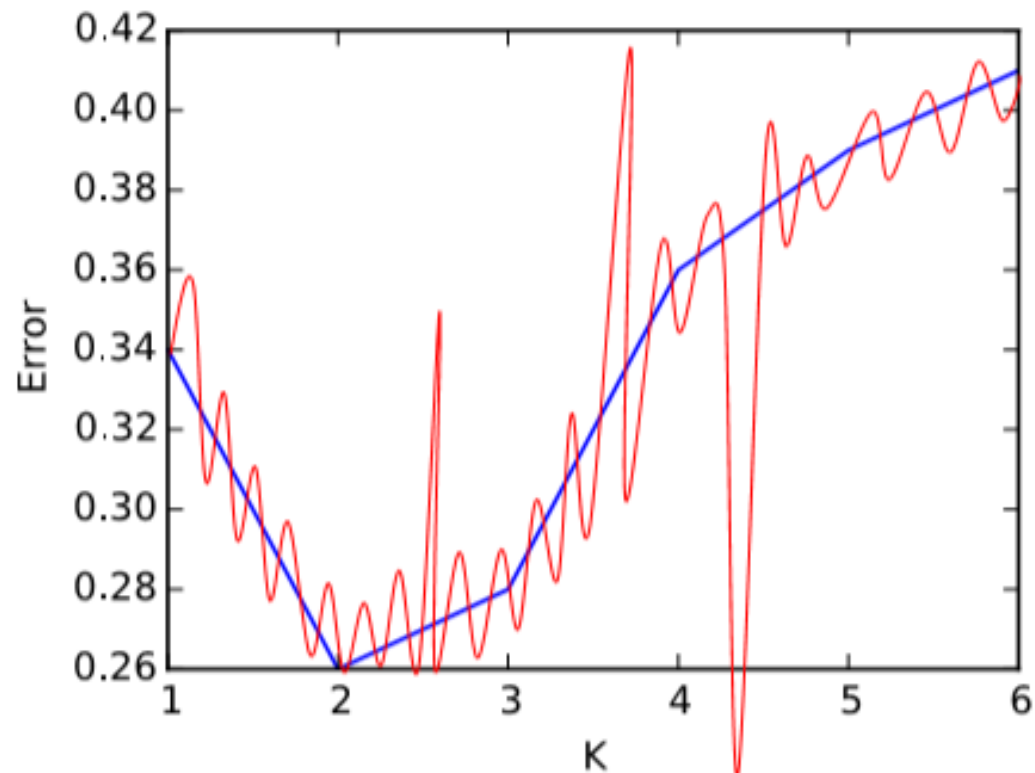




## Доля ошибок при разных $K$

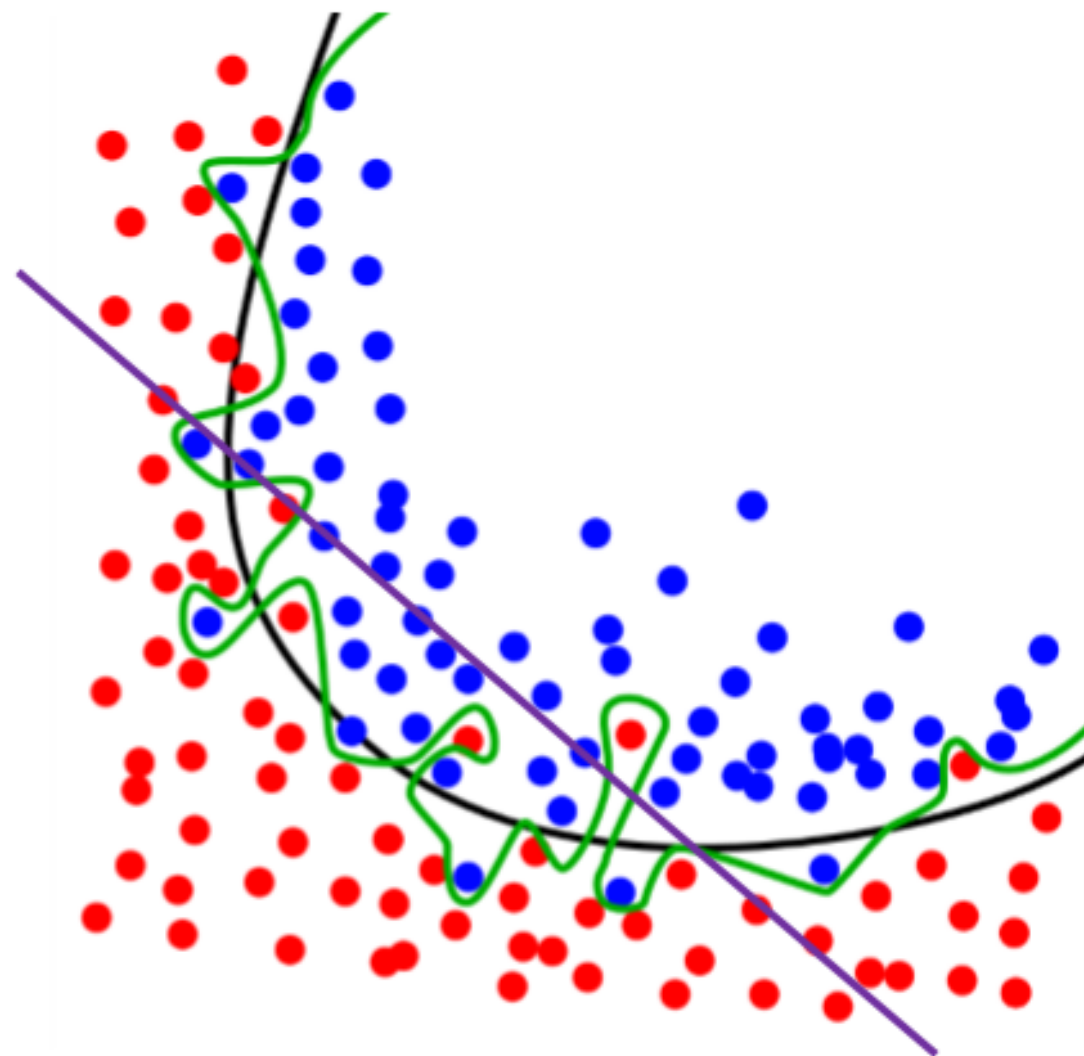


А так бывает в реальных задачах



Методы оптимизации

# Переобучение и недообучение



# Оценка качества решения

---

Для оценки качества решения в задачах обучения с учителем определяют функцию потерь, или функцию штрафа.

Например, для задачи восстановления регрессии часто используют квадратичный штраф

$$L(y, f(x)) = \frac{1}{2} (y - f(x))^2$$

или абсолютный штраф:

$$L(y, f(x)) = |y - f(x)|.$$

Для задачи классификации можно взять ошибку предсказания

$$L(y, f(x)) = I(y \neq f(x)),$$

где 
$$I(\text{условие}) = \begin{cases} 1, & \text{условие выполнено,} \\ 0, & \text{условие не выполнено.} \end{cases}$$

# Требования к качеству решения

---

Для обобщения оценки качества решения введем понятие эмпирический риск, или эмпирическая ошибка

$$\hat{R}(f) = \frac{1}{N} \sum_{i=1}^N L(y^{(i)}, f(x^{(i)})).$$

В качестве решающего правила возьмем функцию  $f$ , минимизирующую эту ошибку:

$$f = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^N L(y^{(i)}, f(x^{(i)})), \quad (1)$$

где прецеденты  $(x^{(i)}, y^{(i)}), (i = 1, 2, \dots, N)$  составляют обучающую выборку.

В итоге задача сводится к отысканию функции  $f$  из допустимого множества  $\mathcal{F}$ , удовлетворяющей условию (1), при условии, что  $f$  и  $\mathcal{F}$  фиксированы и известны. Это так называемый принцип минимизации эмпирического риска.



## Пример 1. Медицинская диагностика

Имеются данные о 114 лицах с заболеванием щитовидной железы.

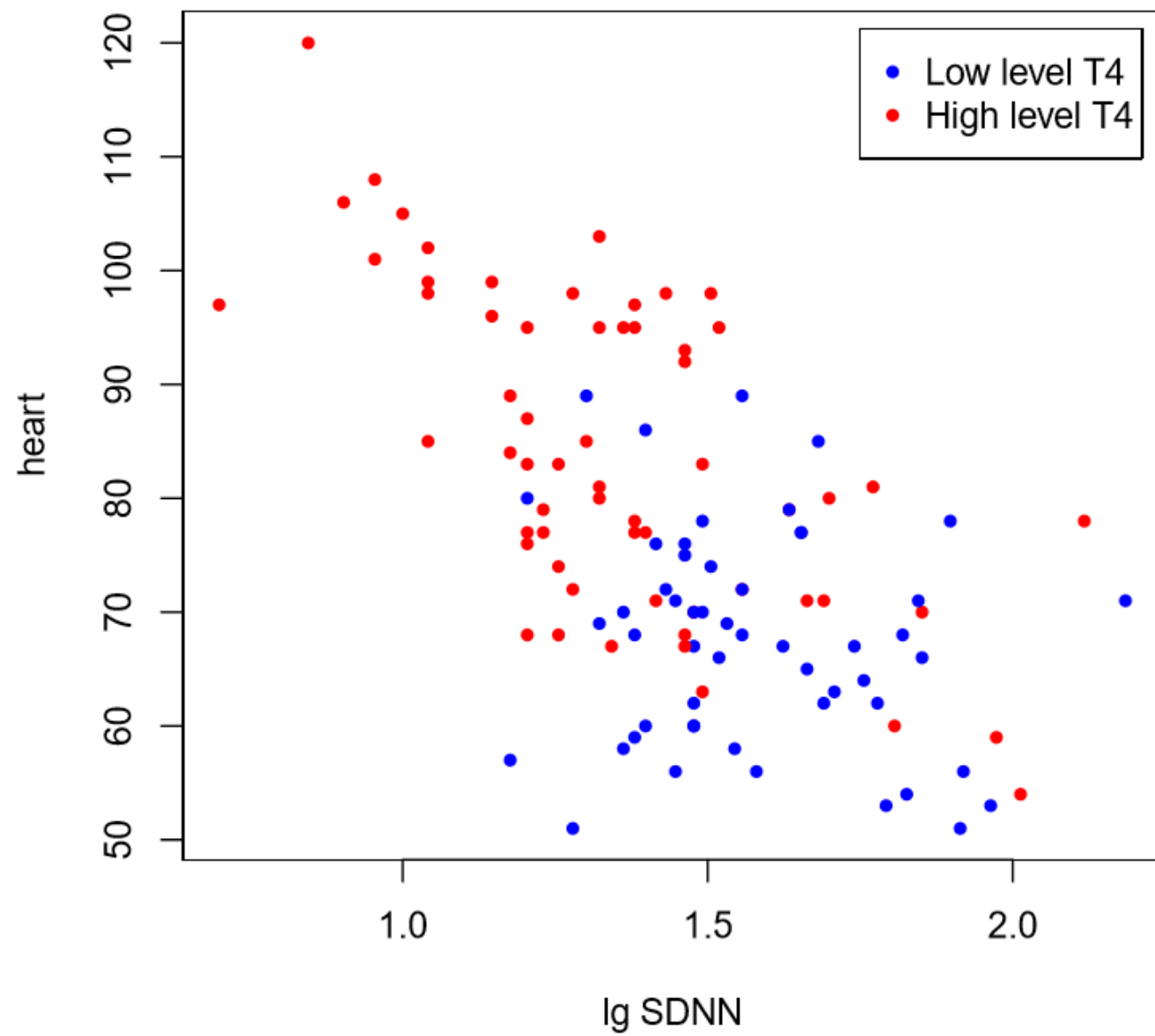
У 61 — повышенный уровень свободного гормона T4,

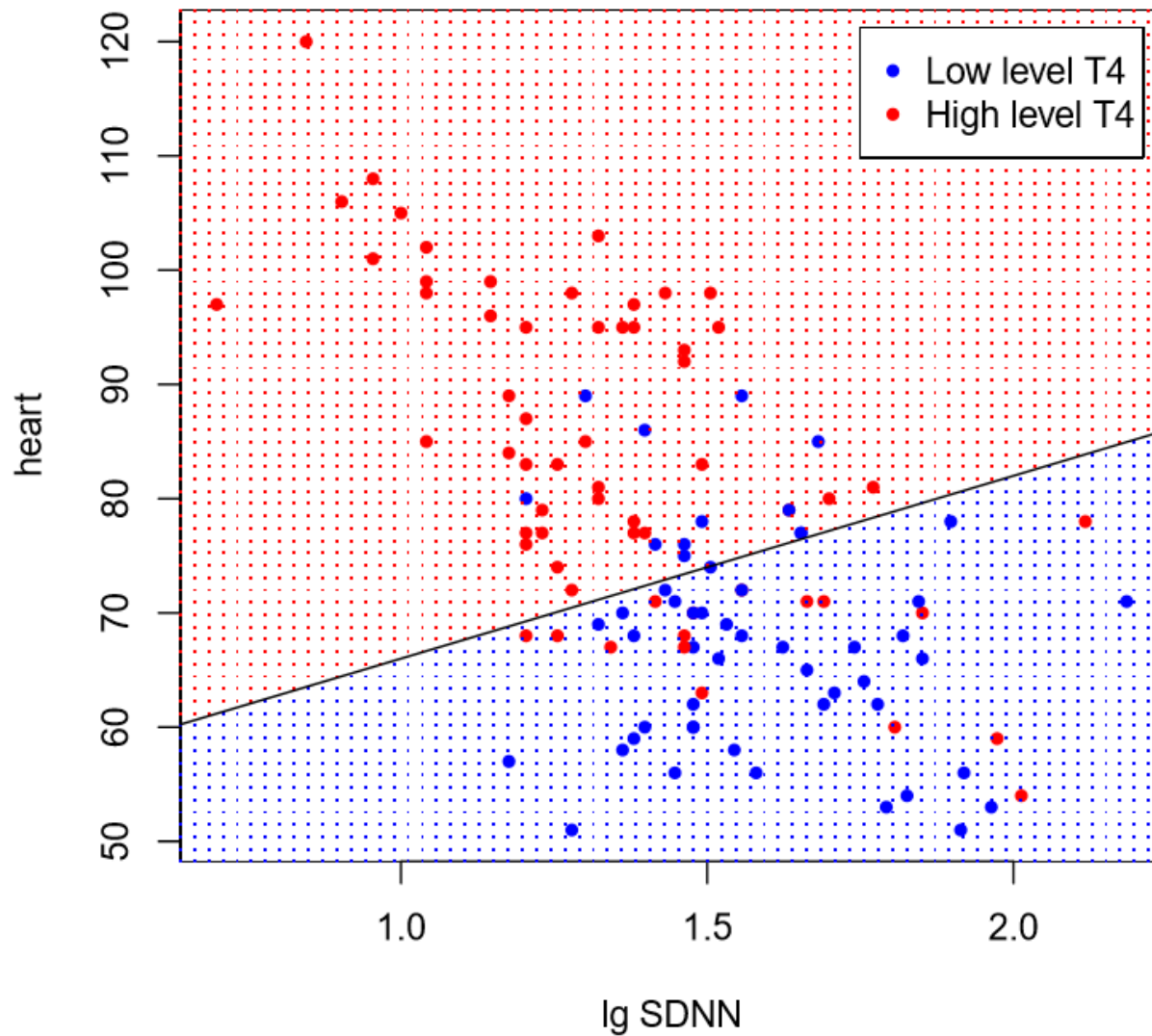
у 53 — уровень гормона в норме.

Для каждого пациента известны следующие показатели:

- $x_1 = \text{heart}$  — частота сердечных сокращений (пульс),
- $x_2 = \text{SDNN}$  — стандартное отклонение длительности интервалов между синусовыми сокращениями RR.

Можно ли научиться предсказывать (допуская небольшие ошибки) уровень свободного T4 по heart и SDNN?

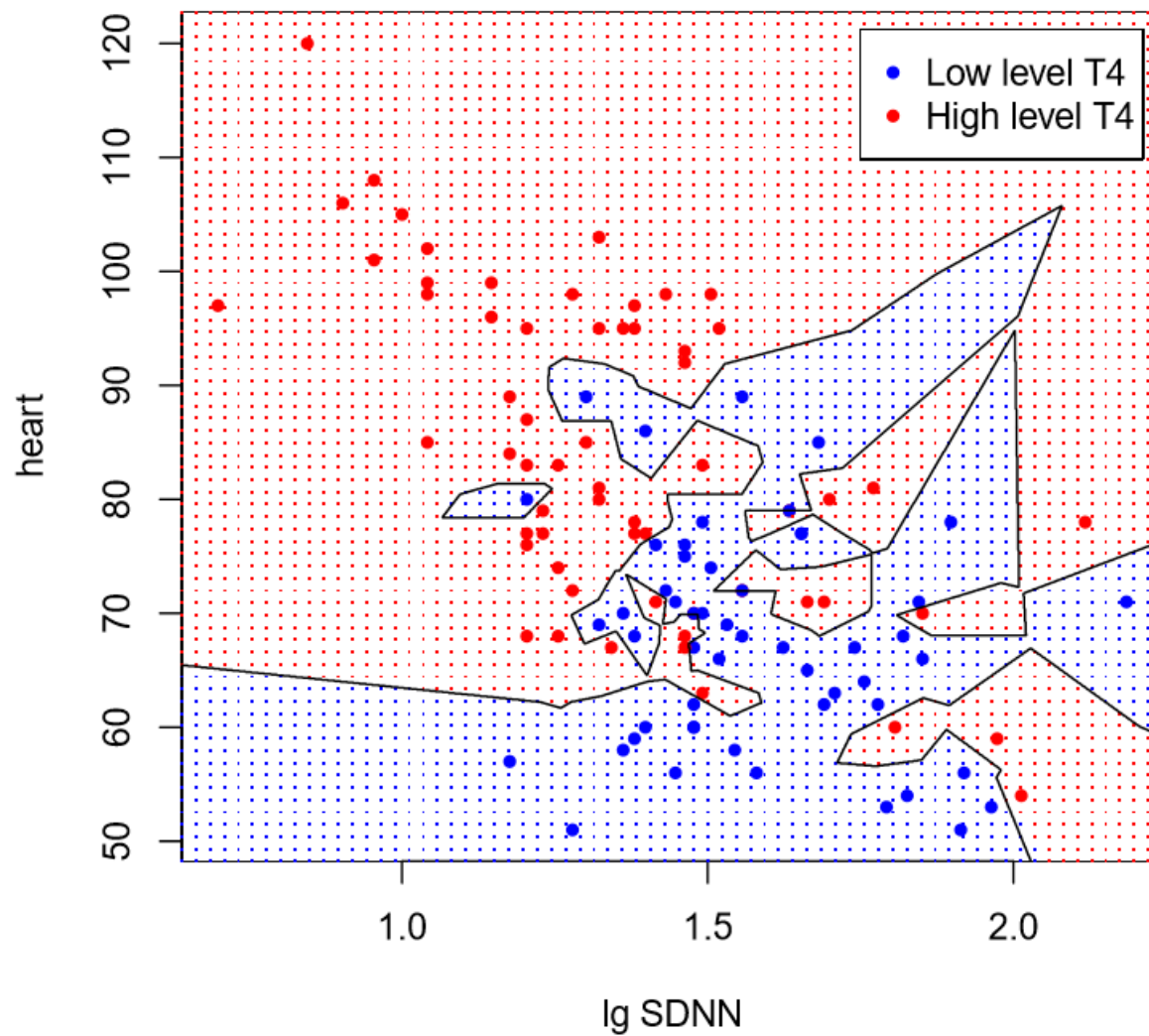




Метод линейной  
аппроксимации

$$16 \cdot \lg \text{SDNN} - \text{heart} + 50 = 0$$

Ошибка на обучающей выборке — 23 %.



Метод k ближайших  
соседей

Метод ближайшего соседа (с масштабированием)

Ошибка на обучающей выборке — 0%.

# Мешок слов (Bag of words)

---

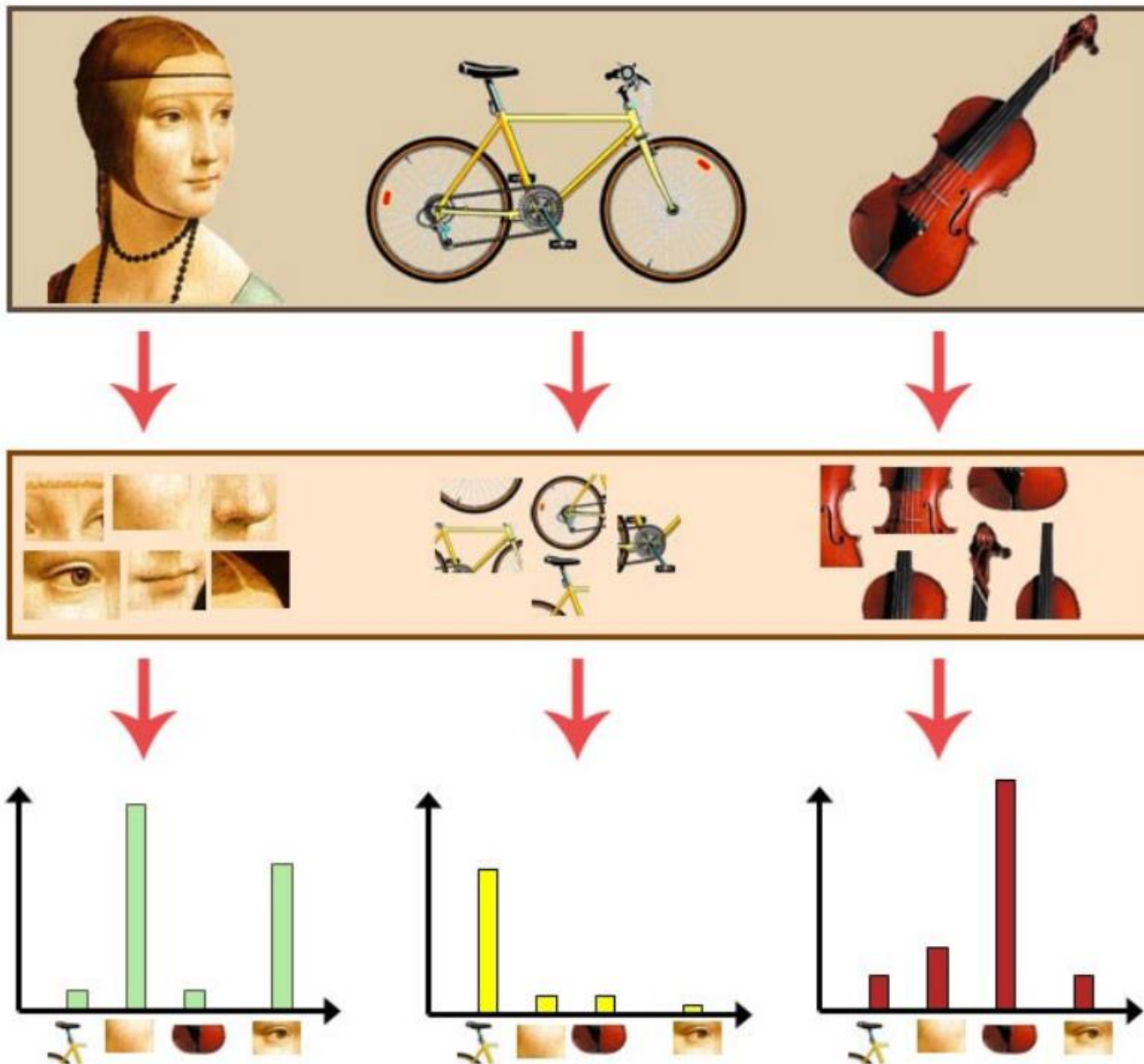
**Мешок визуальных слов (BOVW)** обычно используется в классификации изображений. Его концепция адаптирована из информационного поиска и пакета слов НЛП (BOW). В сумке слов (BOW) мы подсчитываем количество каждого слова, которое появляется в документе, используем частоту каждого слова, чтобы узнать ключевые слова документа, и делаем из него гистограмму частоты. Мы рассматриваем документ как пакет слов (BOW).

Та же концепция в наборе визуальных слов (BOVW), но вместо слов мы используем функции изображения в качестве «слов». **Особенности изображения - это уникальный шаблон, который мы можем найти в изображении.**

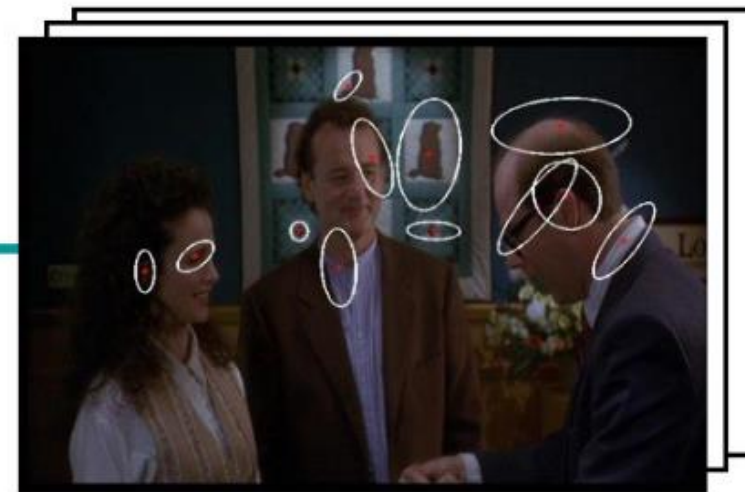
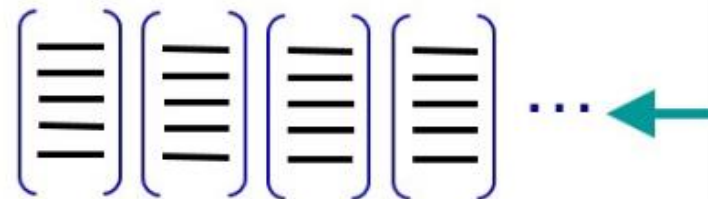


**Особенности состоят из  
ключевых точек и  
дескрипторов.**

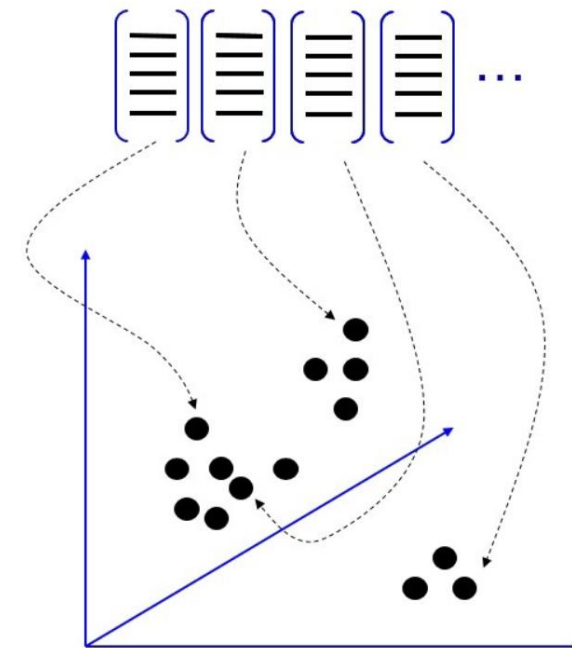
Мы используем ключевые точки  
и дескрипторы для создания  
словарей и представления  
каждого изображения в виде  
частотной гистограммы  
объектов, которые находятся в  
изображении.



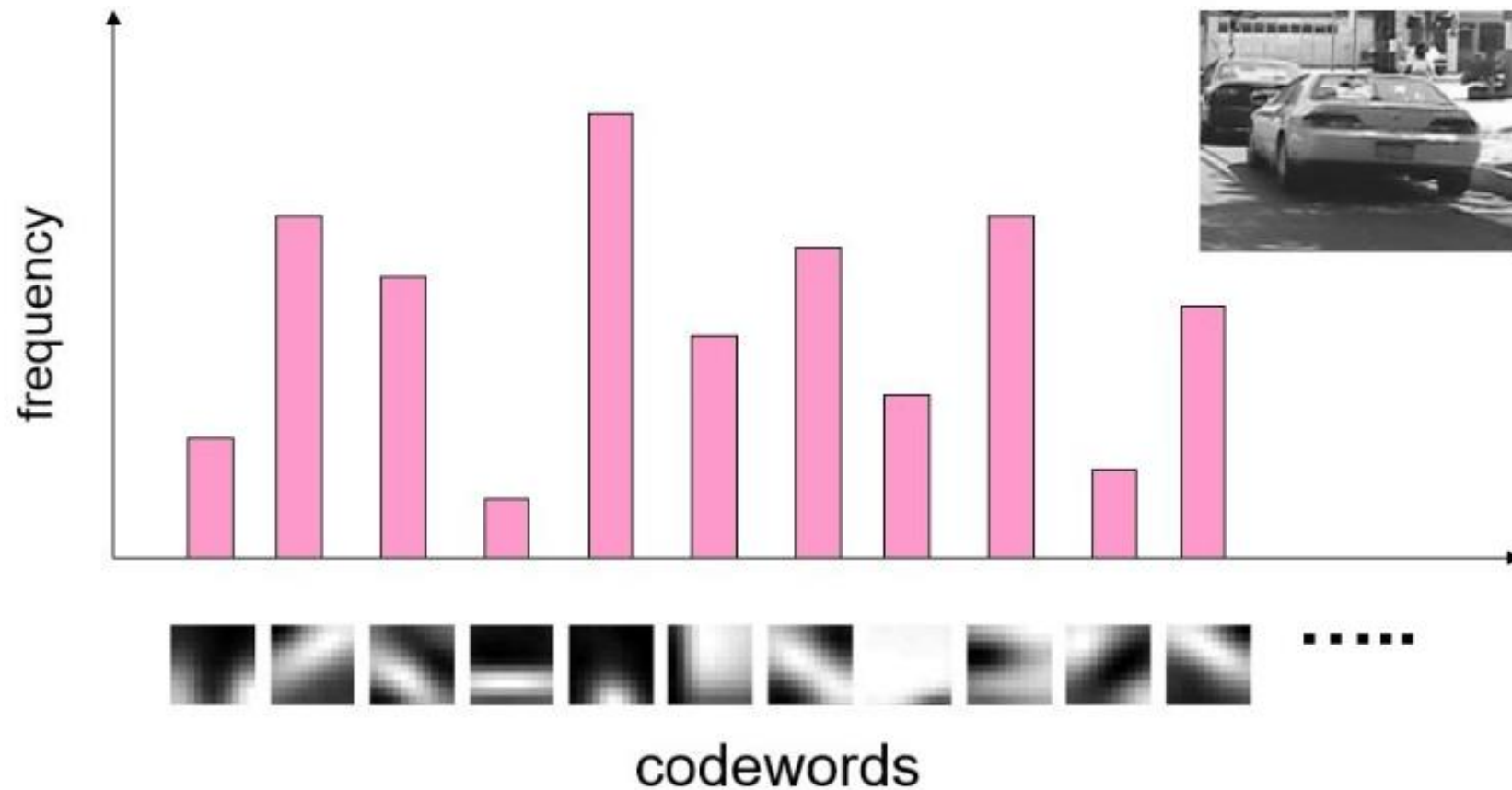
1. Обнаруживаем особенности, извлекаем дескрипторы



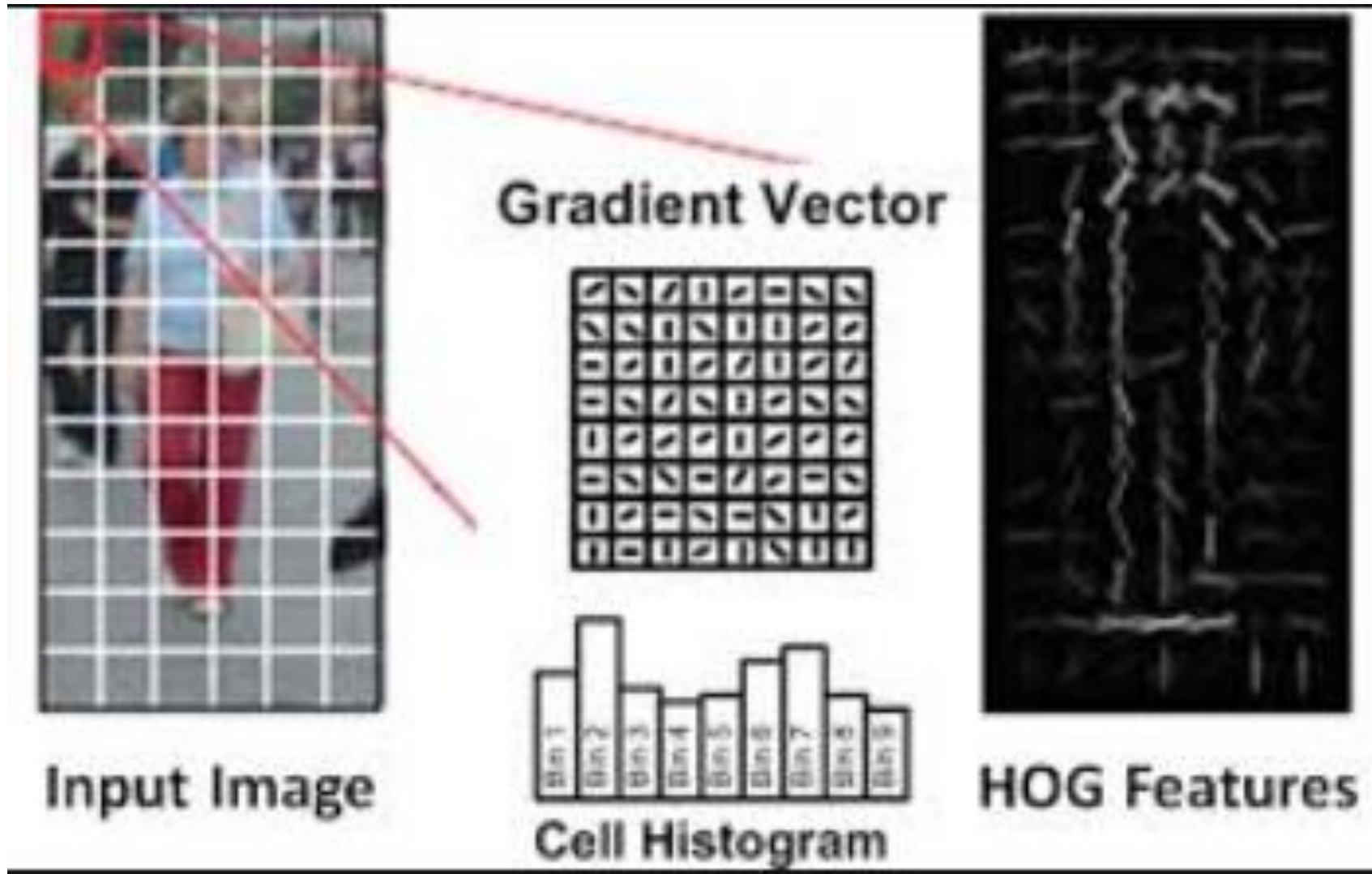
2. Делаем кластеры из дескрипторов (можно использовать любой алгоритм кластеризации, например, K-Means,).



3. Составляем частотную гистограмму из словарей и частоту словарей в изображении. Эти гистограммы - наш мешок визуальных слов (BOVW).

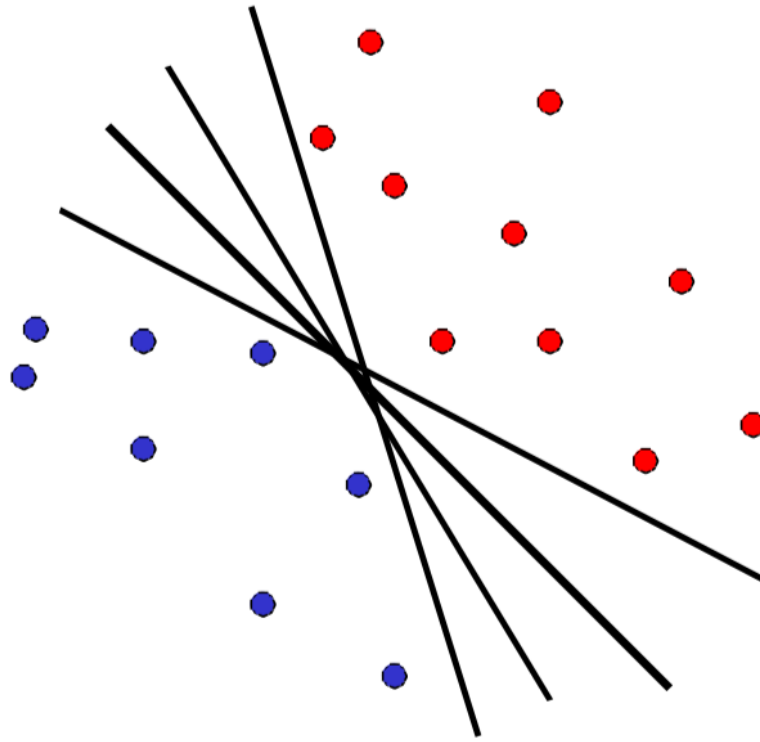


# Гистограмма направленных градиентов (Histogram of Oriented Gradients, HOG)



# Метод опорных векторов (SVM)

---



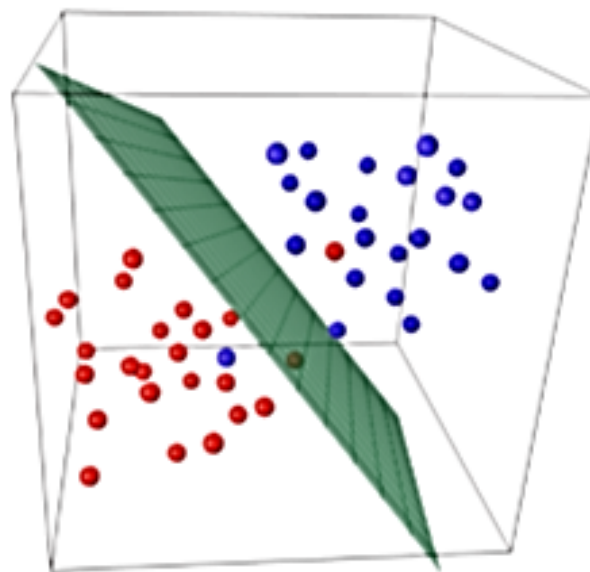
Один из самых популярных методов машинного обучения – машина опорных векторов (SVM – Support Vector Machine) – является развитием идей, предложенных в 1960–1970 гг. В. Н. Вапником и А. Я. Червоненкисом.



# Что такое гиперплоскость?

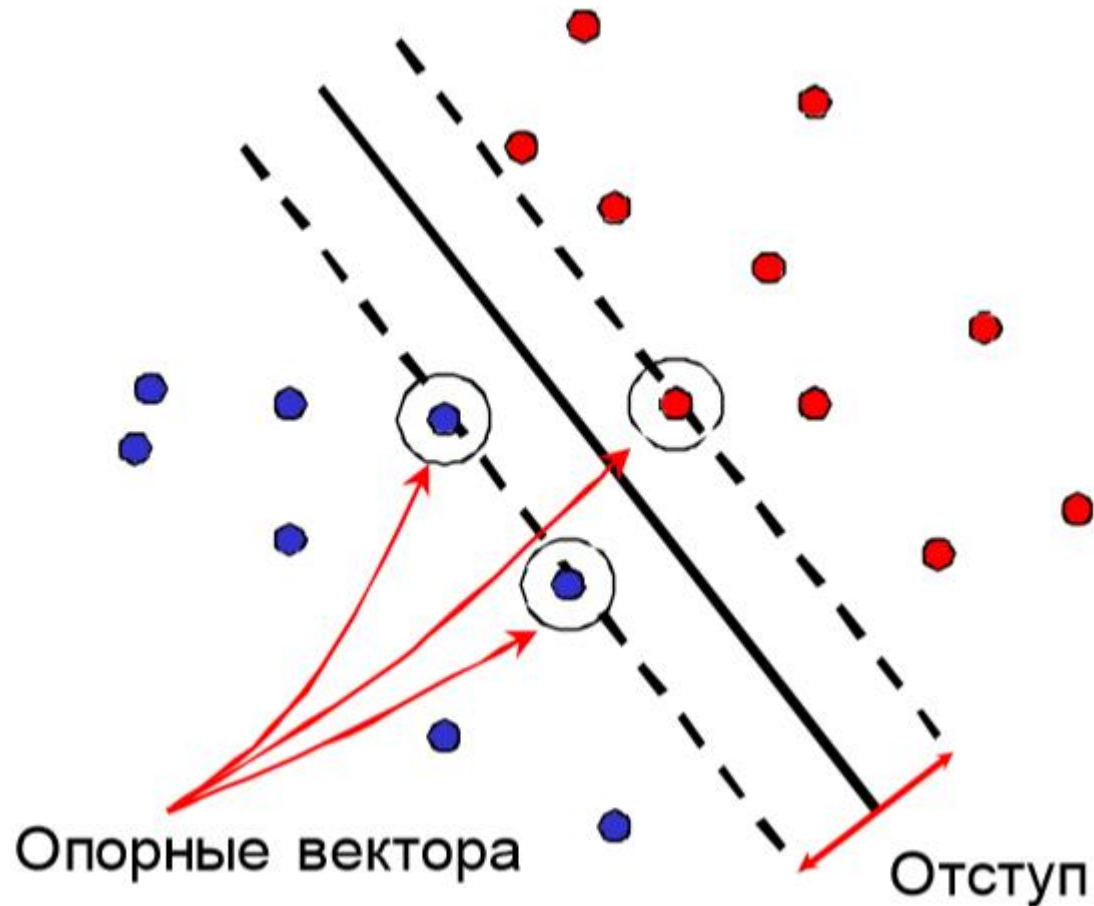
$$W_0 + W_1 X_1 + W_2 X_2 + \dots + W_n X_n = 0$$

- В двумерном случае – прямая
- В трехмерном – плоскость
- Дальше – гиперплоскость



# Метод опорных векторов

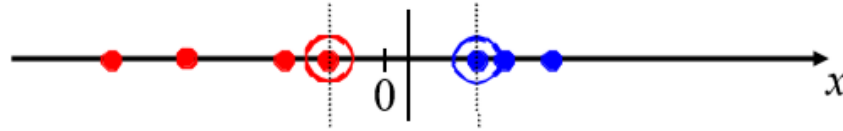
---



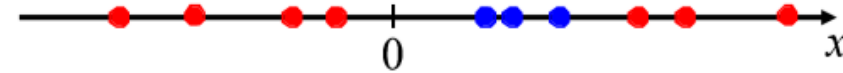
Оптимальной разделяющей гиперплоскостью называется такая гиперплоскость, расстояние от которой до ближайшей точки из обучающей выборки максимально.

# Нелинейный SVM

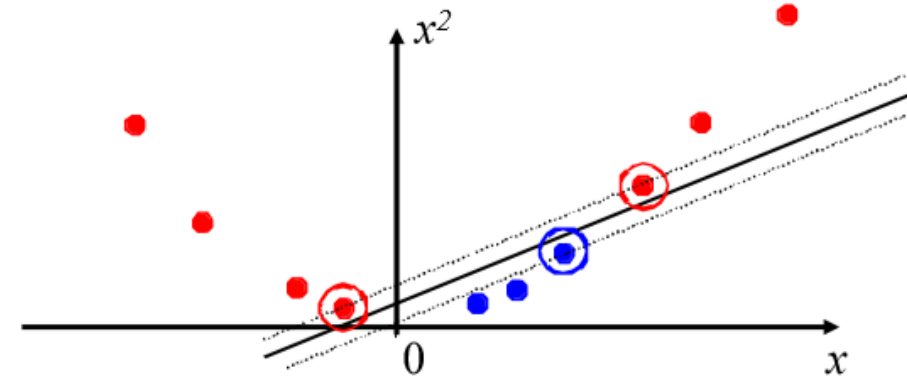
- На линейно разделимых данных МОВ работает отлично:



- Но на более сложных данных не очень:



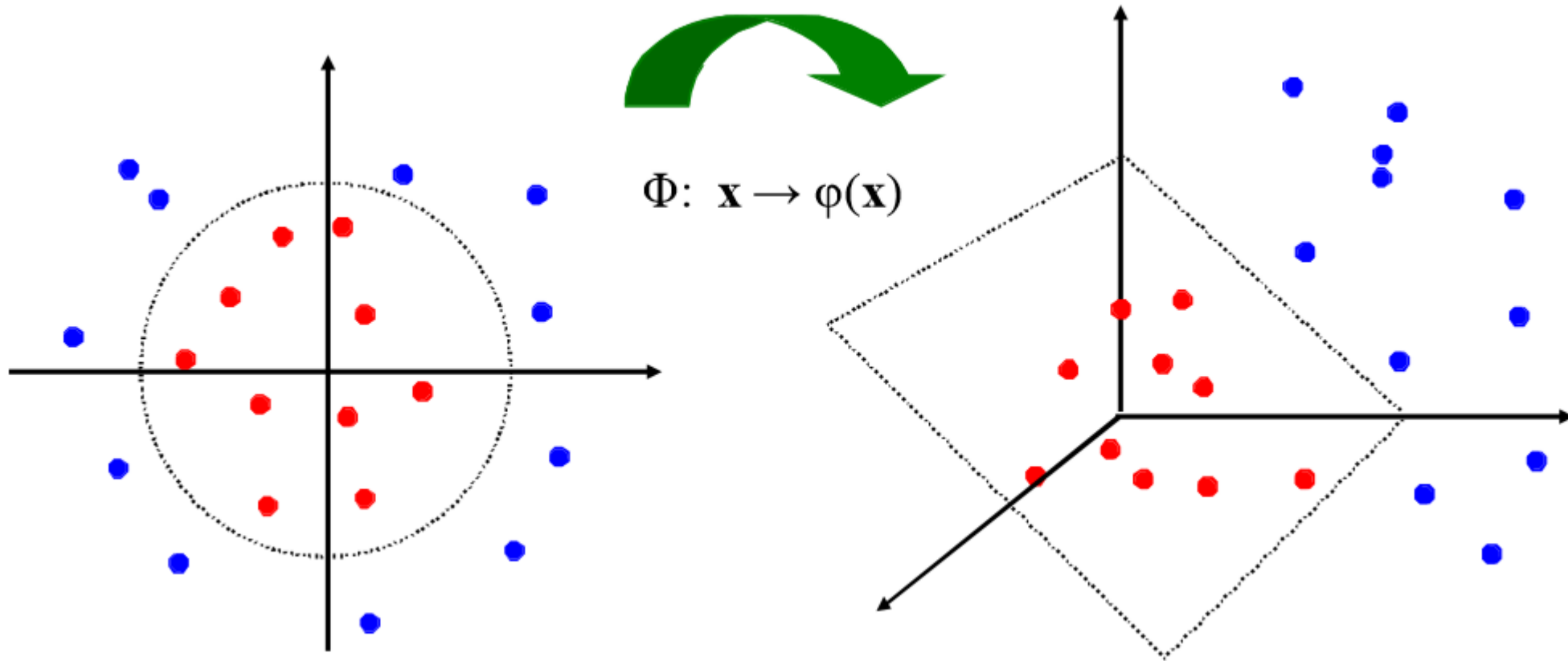
- Можно отобразить данные на пространство большей размерности и разделить их линейно там:



Slide credit: Andrew Moore

# Нелинейный SVM

Идея: отображение исходного пространства параметров на какое-то многомерное пространство признаков (feature space) где обучающая выборка линейно разделима:



# Нелинейный SVM

---

$$\mathbf{x} \rightarrow \varphi(\mathbf{x})$$

Вместо прямого вычисления преобразования  $\varphi(\mathbf{x})$ , определяют ядровую функцию  $K$ :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

На практике ядро не рассчитывают, а подбирают наиболее подходящее.

Вот, к примеру, перечень некоторых популярных функций ядер:

- линейное ядро:  $K(x, x') = xx'$ ;
- многочлен степени  $d$ :  $K(x, x') = (\gamma_0 + \gamma xx')^d$ ;
- радиальная функция:  $K(x, x') = e^{-\gamma \|x - x'\|^2}$ ;
- сигмоидальная («нейронная») функция:  $K(x, x') = \tanh(\gamma_0 + \gamma xx')$ .



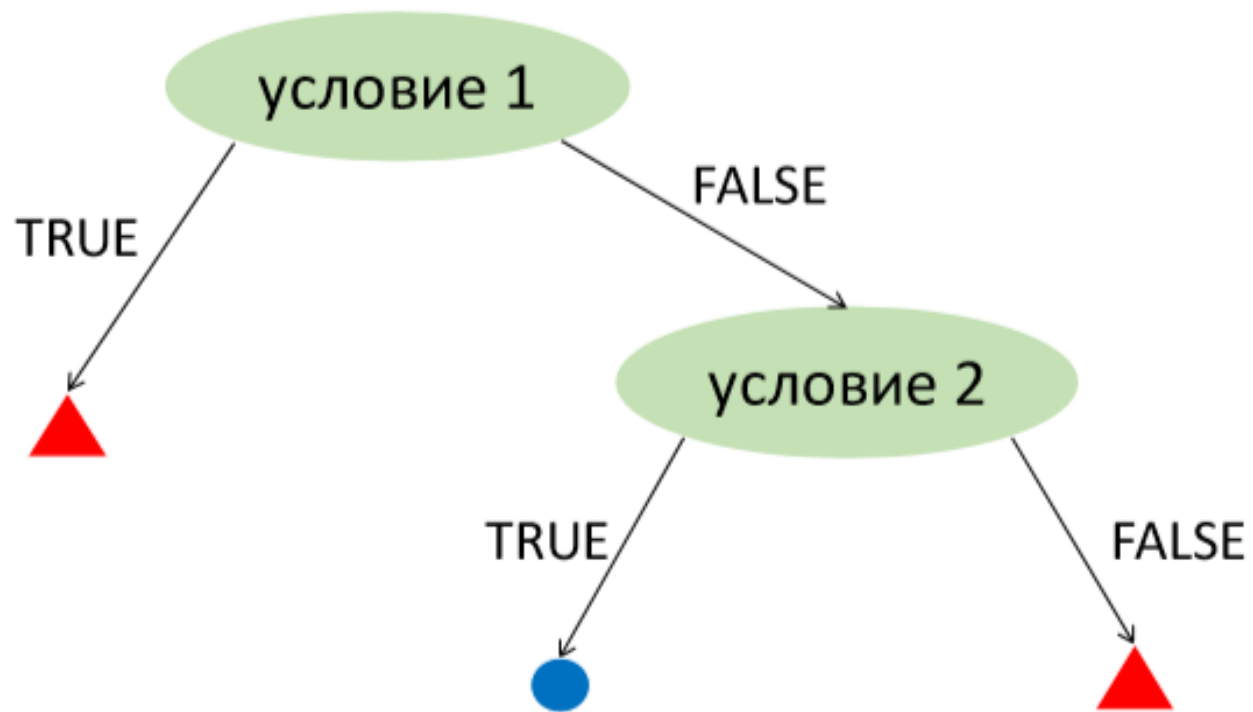
# Многоклассовый SVM

---

- Нет специальной формулировки SVM для случая многих классов
- На практике SVM для нескольких классов получается путем комбинации нескольких двухклассовых SVM
- Один против всех
  - Обучение: обучаем SVM для каждого класса против всех остальных
  - Вывод: применим все SVM к образцу и назначим класс, SVM для которого выдал наиболее достоверное решение
- Один против одного:
  - Обучение: обучим SVM для каждой пары классов
  - Вывод: каждый SVM голосует за классы, выбираем класс с наибольшим числом голосов

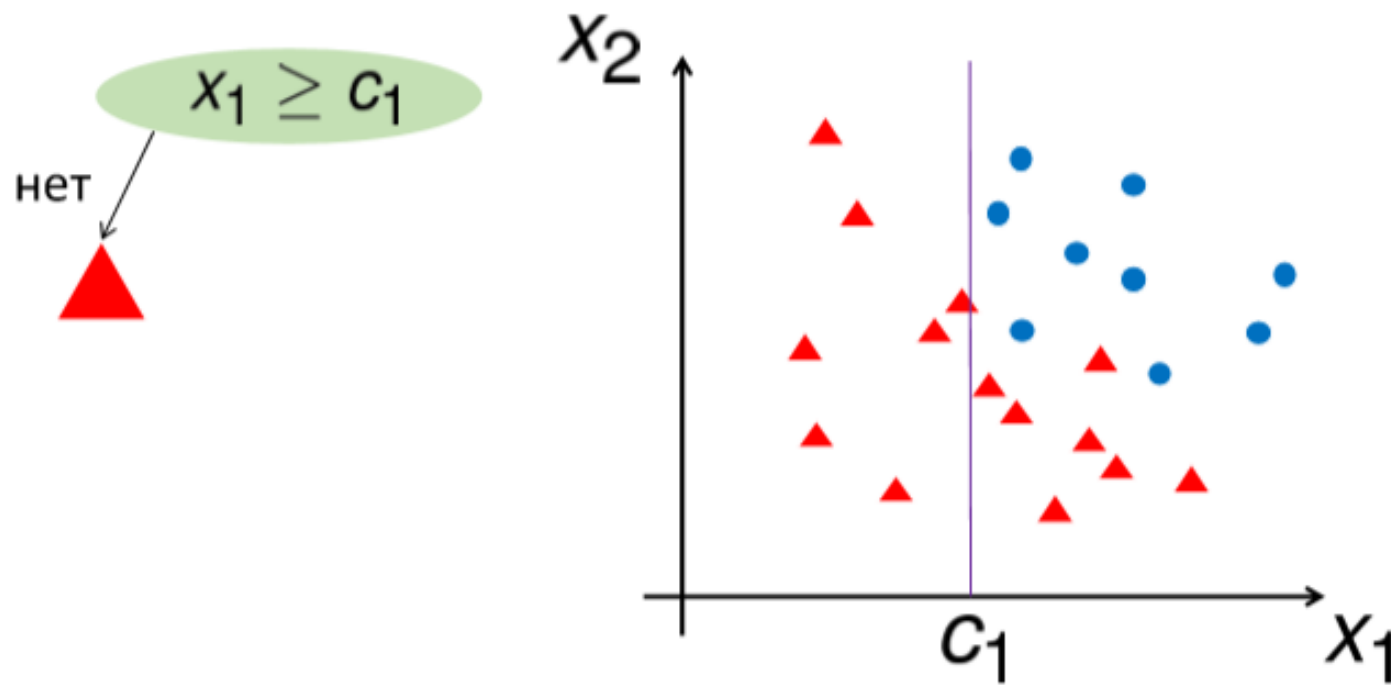
# Деревья решений

---

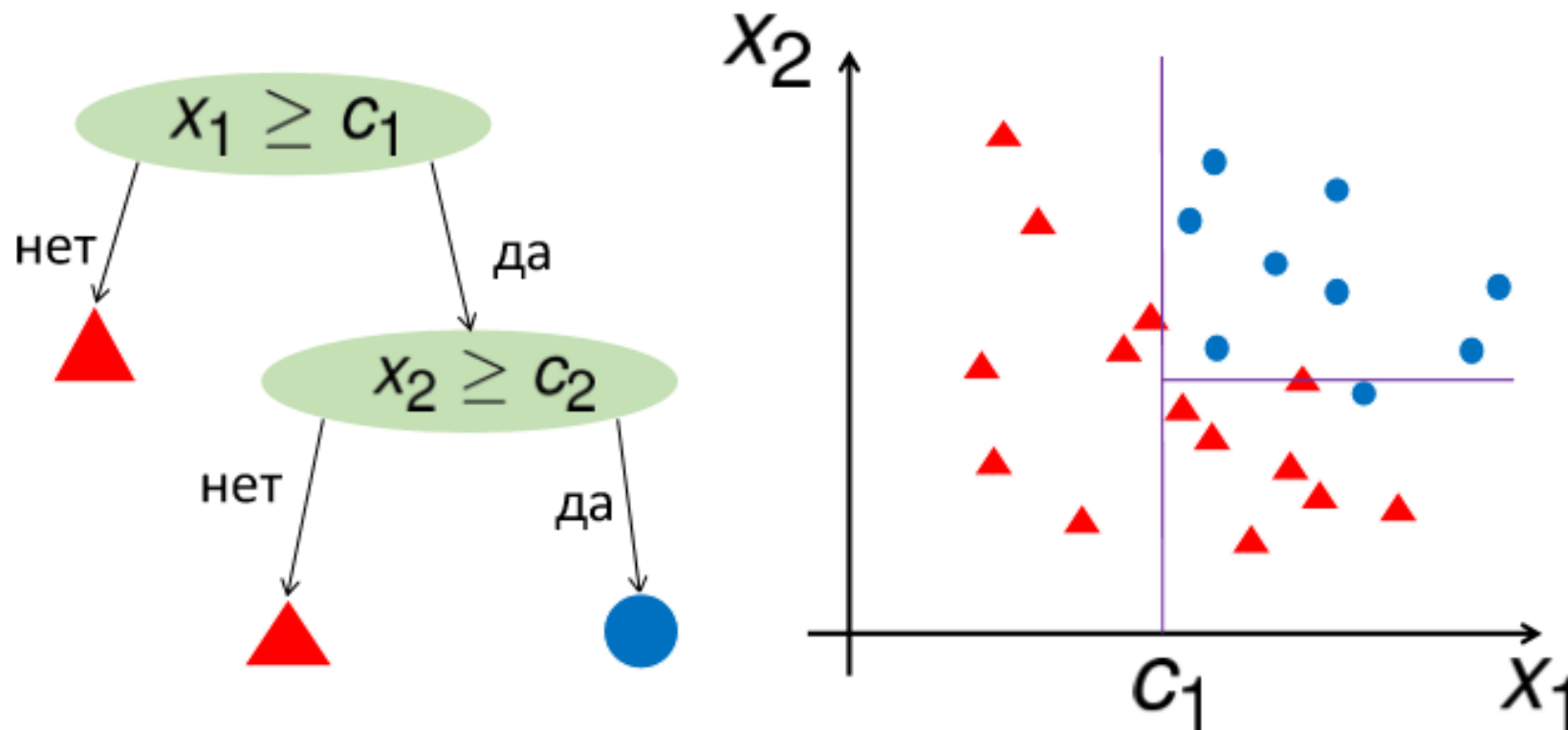


# Деревья решений

---



# Деревья решений

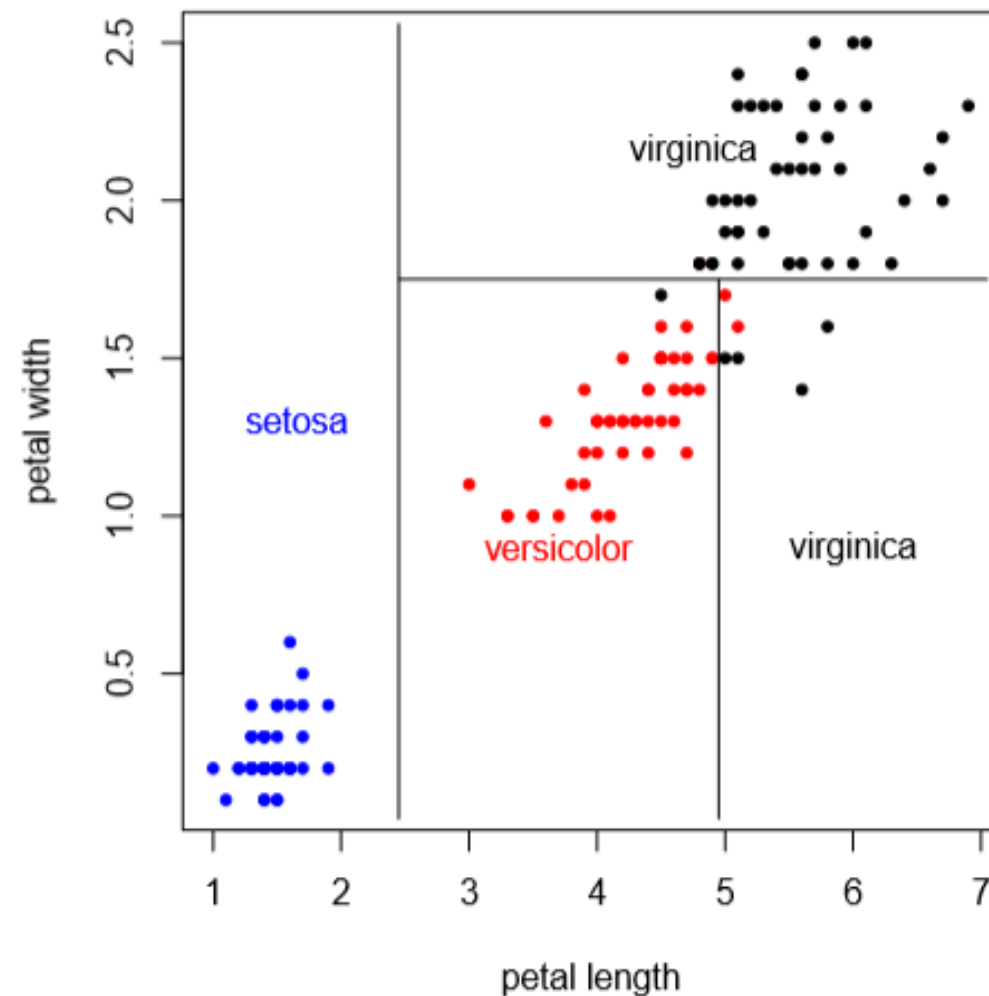
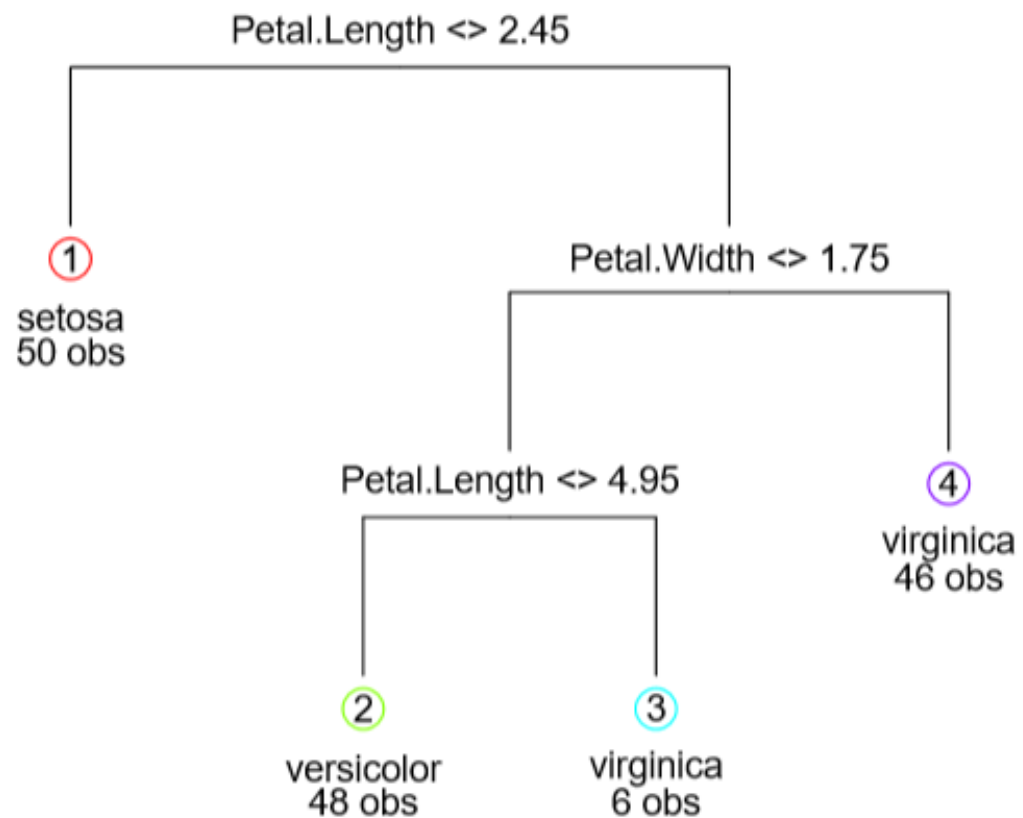


Разбиения проводятся до тех пор, пока в строящиеся вершины попадает достаточное количество точек обучающей выборки, или пока дерево не достигнет заданной глубины.

# Деревья решений

Задача классификации цветов ириса (Fisher, 1936).

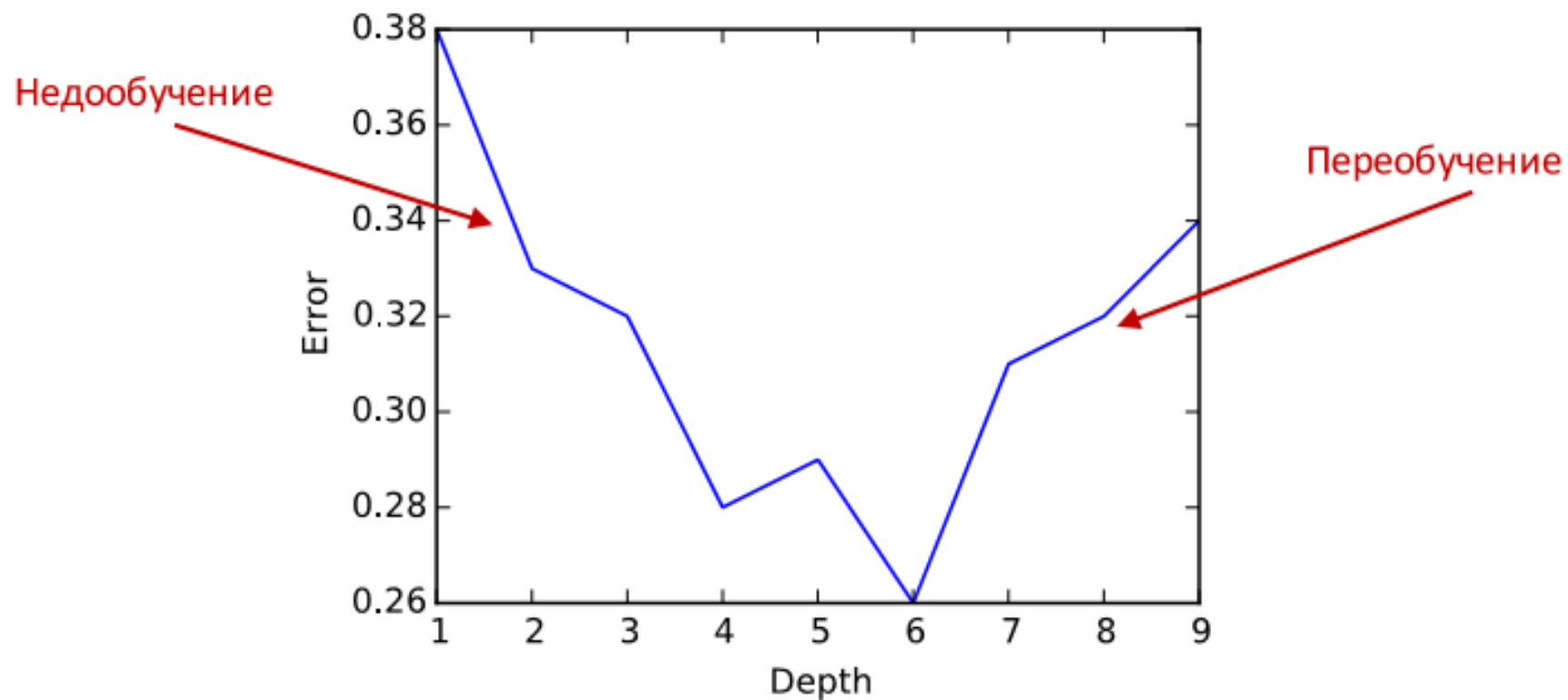
$x_1$ ,  $x_2$  — длина и ширина чашелистика.



# Деревья решений

---

Зависимость качества от глубины дерева





# Деревья решений

---

Деревья решений являются одним из наиболее наглядных и универсальных алгоритмов обучения. **К достоинствам деревьев решений следует отнести:**

- Возможность производить обучение на исходных данных без их дополнительной предобработки (нормализация и т. п.);
- Нечувствительность к монотонным преобразованиям данных;
- Устойчивость к выбросам;
- Поддержка работы с большими выборками;
- Поддержка работы с входными переменными разных типов;
- Возможность интерпретации построенного дерева решений.

## **Недостаток:**

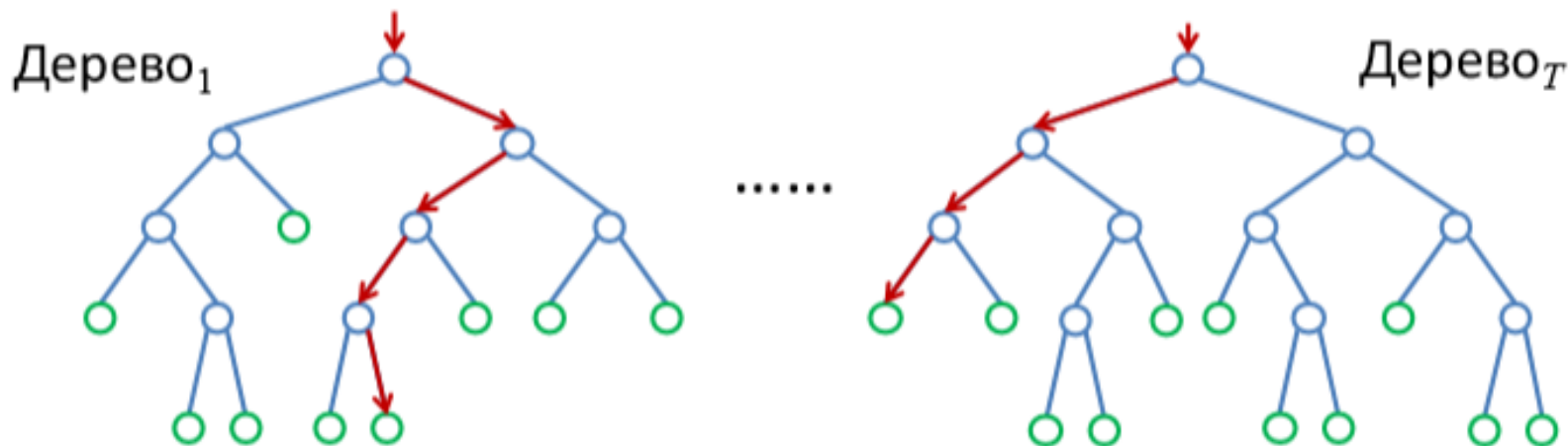
Деревья решений не устойчивы даже к небольшим изменениям.

# Случайный лес

---

Случайный лес, а точнее – случайные леса (random forests), является одним из наиболее универсальных и эффективных алгоритмов обучения с учителем, применимым как для задач классификации, так и для задач восстановления регрессии. Идея метода заключается в использовании ансамбля из деревьев решений, которые обучаются независимо друг от друга.

Баггинг - обучение базовых правил происходит на различных случайных подвыборках данных; при этом базовые правила строятся независимо друг от друга.



# Случайный лес

---

- Работает с признаками разной природы
- Не надо думать над метрикой и параметрами
- Устойчив к изменениям признаков
- Хорошо распараллеливается
- Тяжело интерпретируется человеком
- Плохо приближает линейные зависимости
- Долго строится и используется
- Не переобучается при увеличении количества деревьев

*“This ease of use also makes Random Forests an ideal tool for people without a background in statistics, allowing lay people to produce fairly strong predictions free from many common mistakes, with only a small amount of research and programming”.*

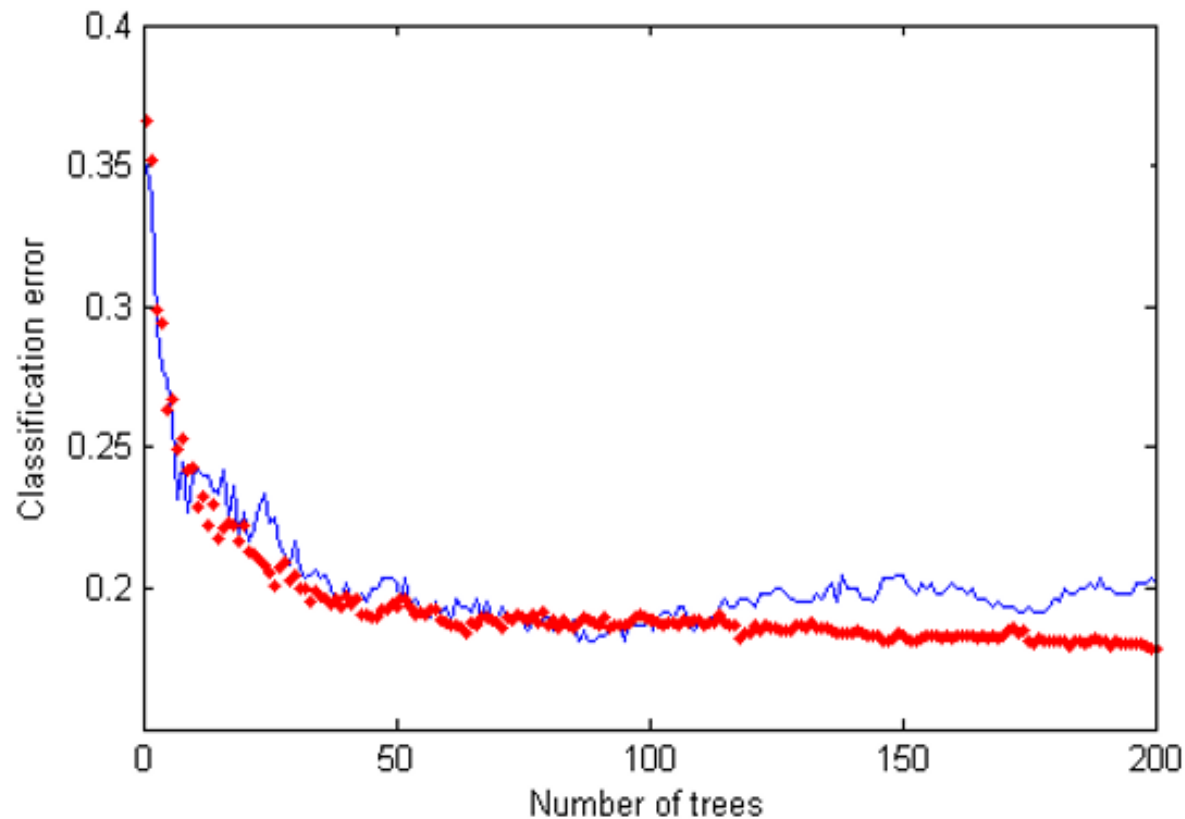
# Бустинг деревьев решений

---

Можно обучать деревья последовательно.

Каждое новое дерево можно учить на ошибку на последнем шаге.

Деревья уже могут быть невысокими.



## Плюсы

- Маленькие деревья
- Мало места
- Быстро работают

# Градиентный бустинг деревьев решений

---

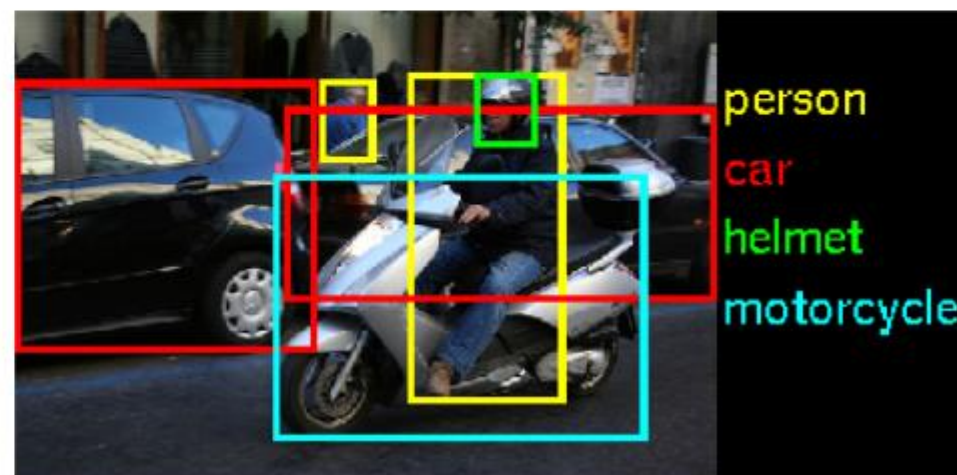
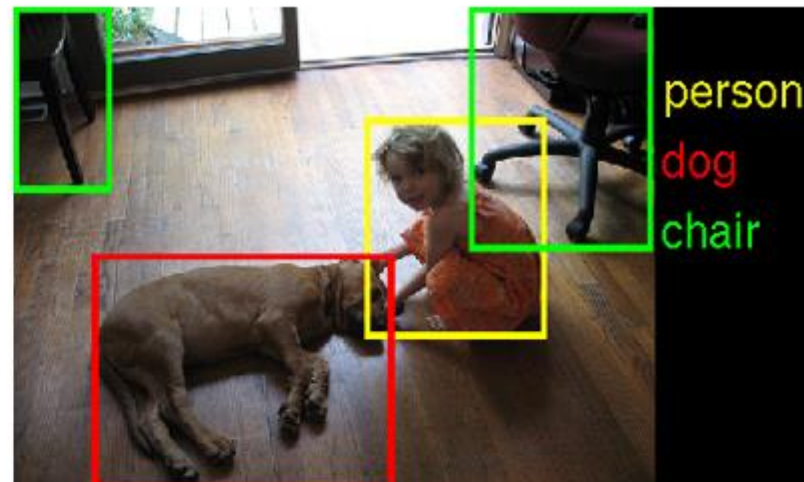
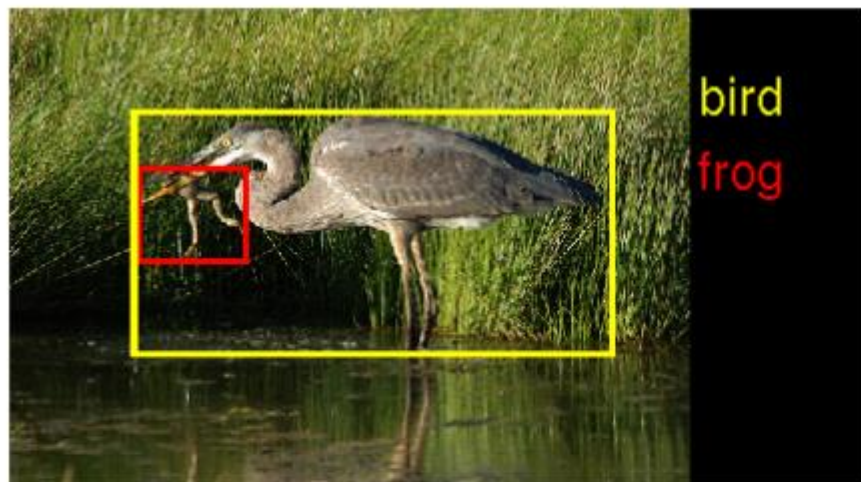
Если вместо разности рассматривается градиент вектора ответов, то получаем градиентный бустинг деревьев.

Используется

- Yandex
- Google
- Bing
- Yahoo
- CERN
- ...

С 2009 года Яндекс развивает собственный метод машинного обучения — Матрикснет. В основе Матрикснета лежит механизм градиентного бустинга. Он был разработан для ранжирования результатов поиска, но используется и в других сервисах Яндекса: например, в Почте — для фильтрации спама, в Картах и Навигаторе — для расчёта времени в пути, в Директе — для подбора релевантных рекламных объявлений. Матрикснет устойчив к переобучению и позволяет использовать множество факторов и их комбинаций.

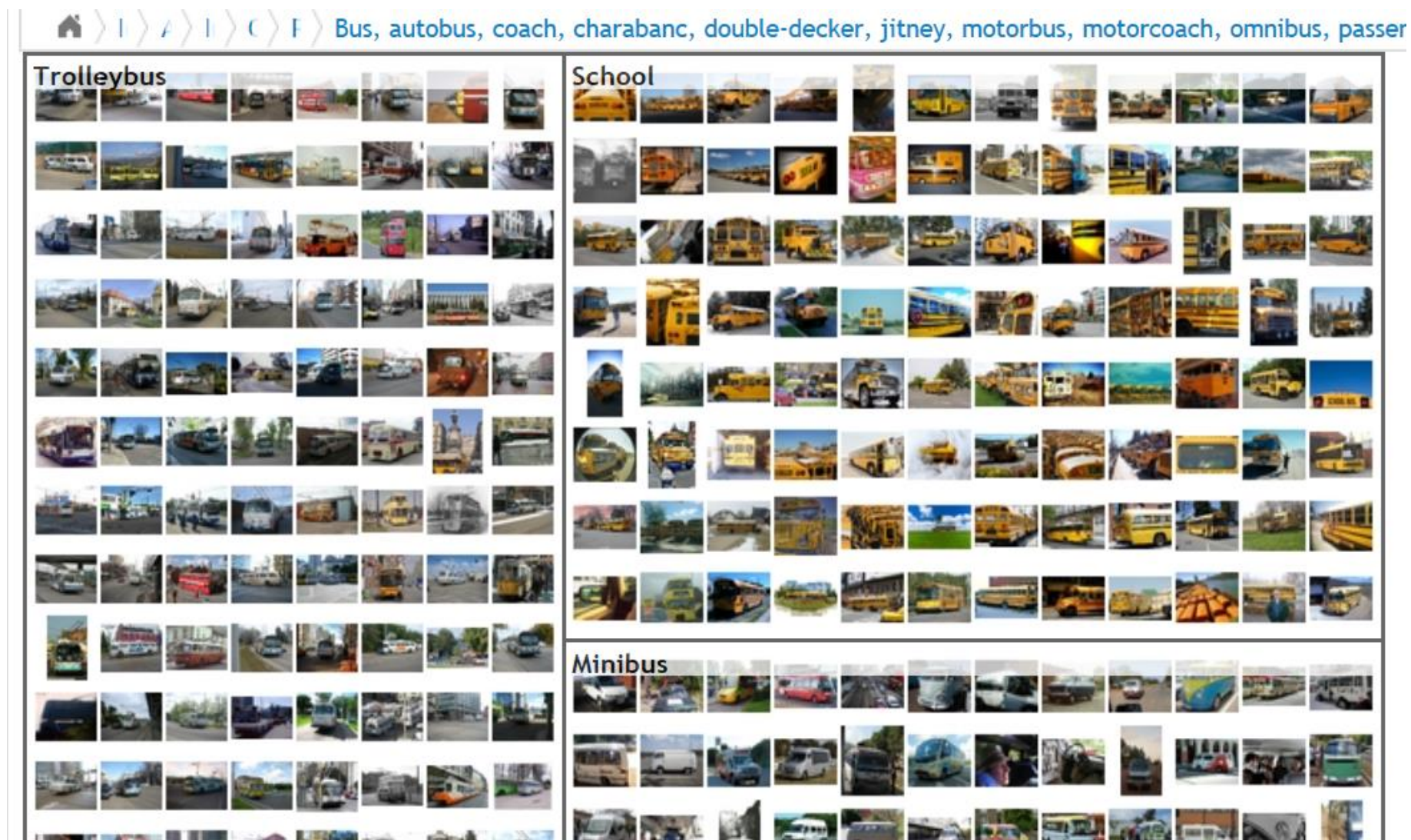
# Нейронные сети





# Нейронные сети

Определяет 200 категорий объектов

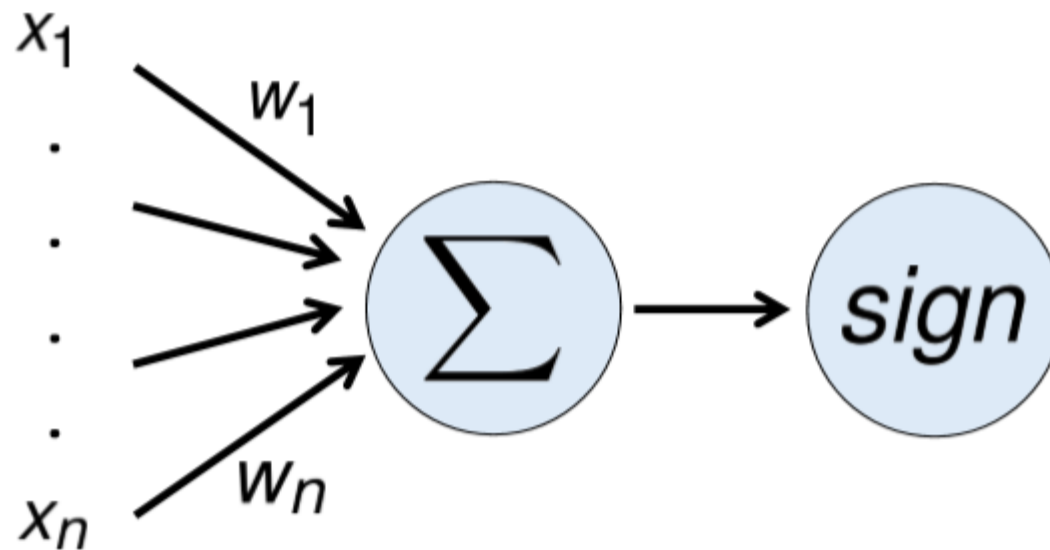




# Нейронные сети

---

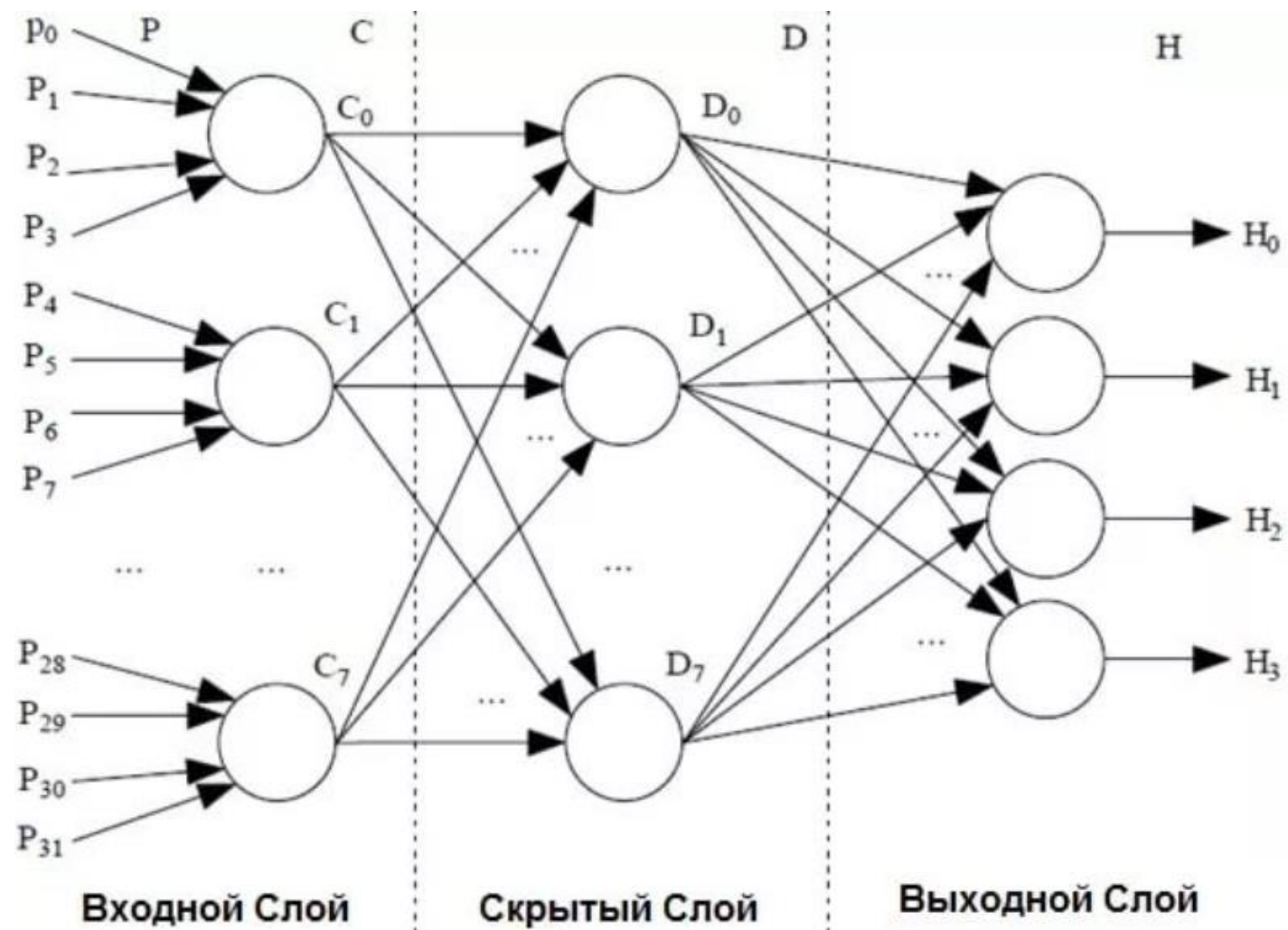
$$W_0 + W_1X_1 + W_2X_2 + \dots + W_nX_n :$$







# Нейронные сети

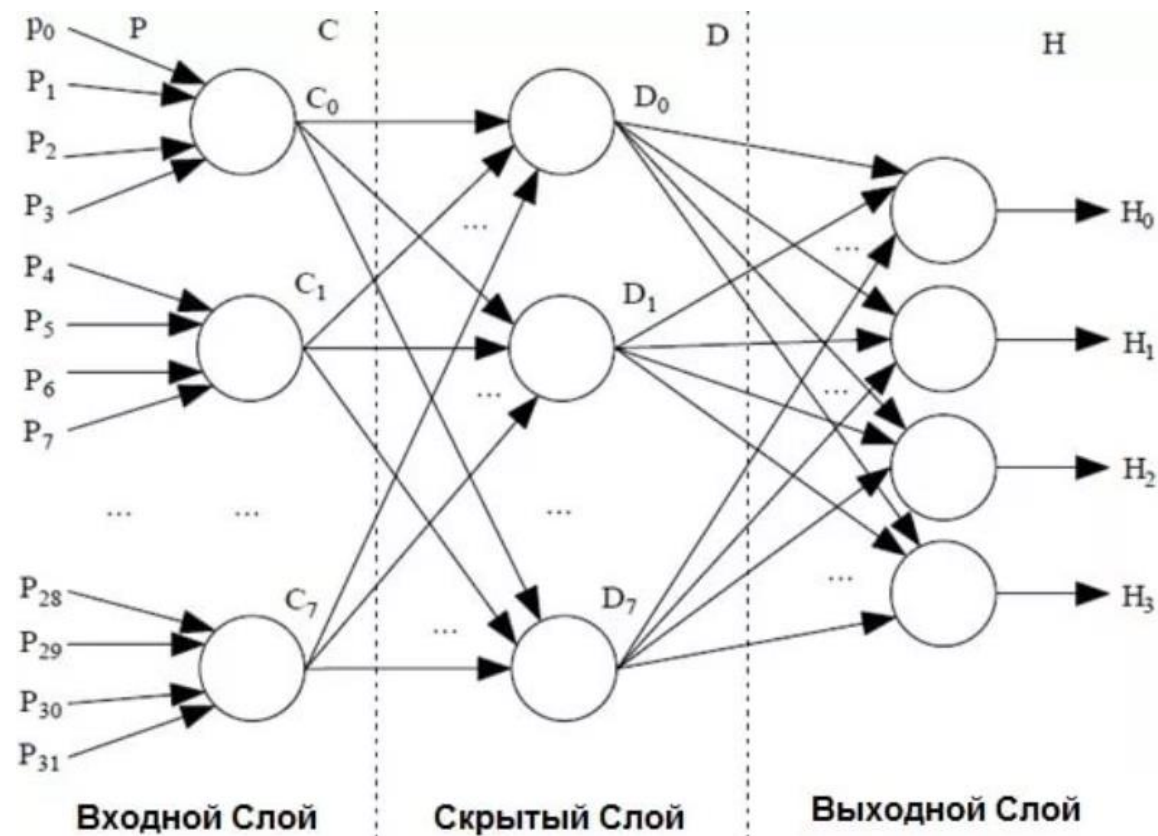


Из этих минимальных структурных единиц собирают классические искусственные нейронные сети. Принята следующая терминология:

**Входной (рецепторный) слой** — это вектор параметров (признаков). Этот слой не состоит из нейронов.

**Ассоциативный (скрытый) слой** — глубинная структура, способная к запоминанию примеров, нахождению сложных корреляций и нелинейных зависимостей, к построению абстракций и обобщений. В общем случае это даже не слой, а множество слоев между входными и выходными. Структура содержит столько нейронов и слоев, сколько душе угодно.

**Выходной слой** — это слой, каждый нейрон которого отвечает за конкретный класс. Выход этого слоя можно интерпретировать как функцию распределения вероятности принадлежности объекта разным классам. Слой содержит столько нейронов, сколько классов представлено в обучающей выборке.





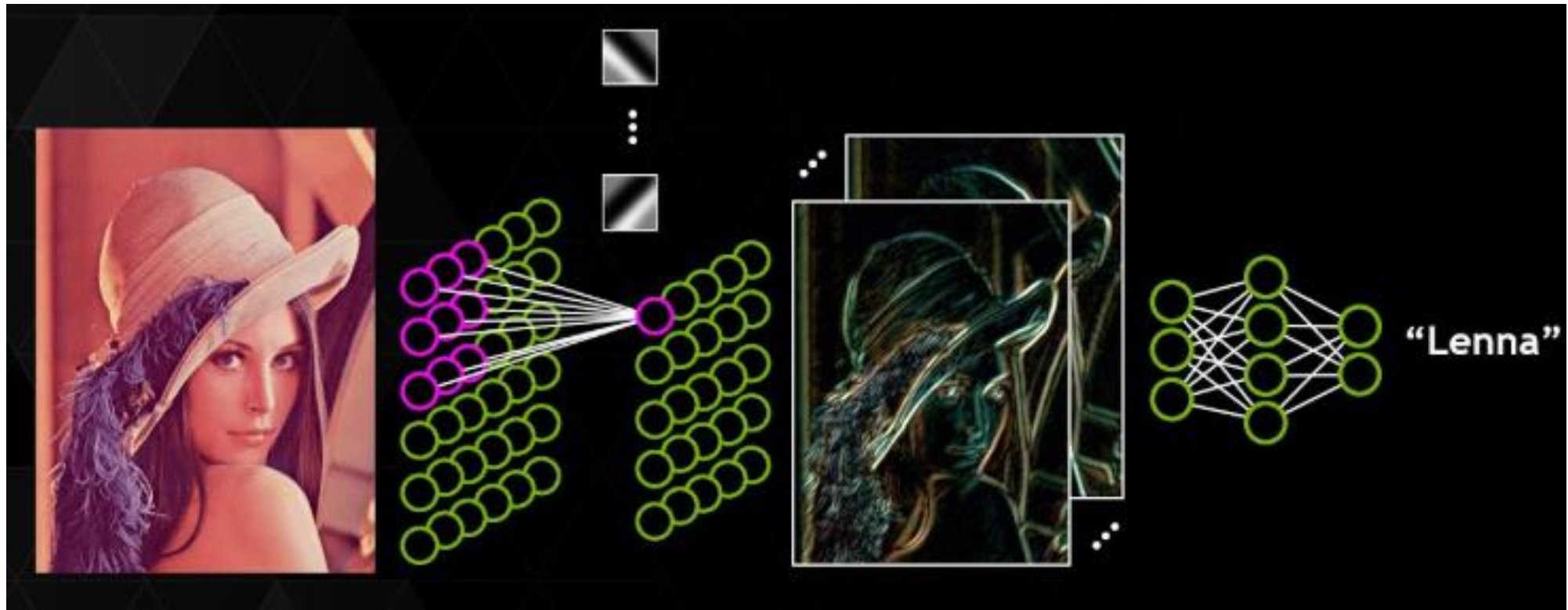
# Сверточная нейронная сеть

---

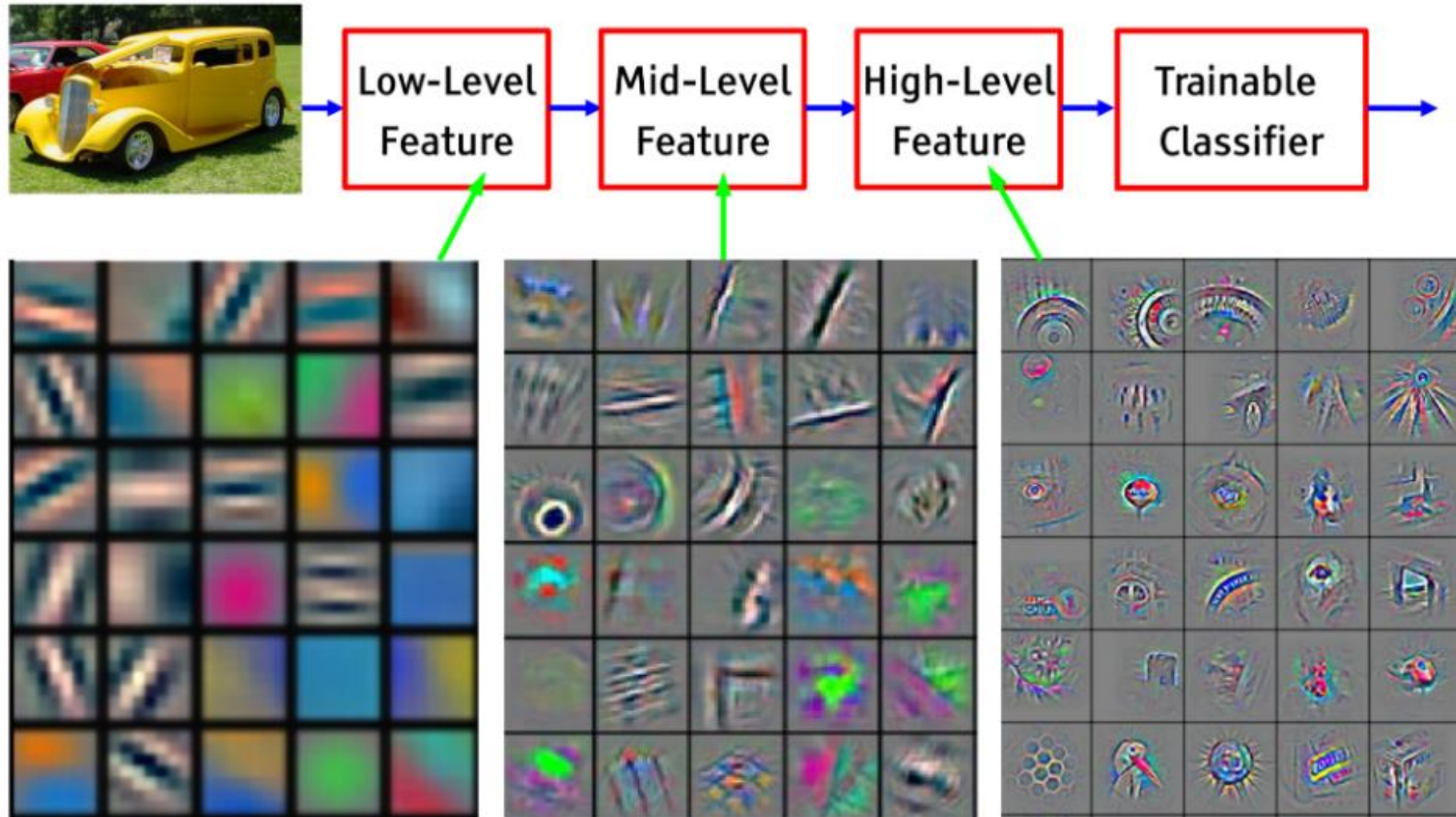
**Свёрточная нейронная сеть**— специальная архитектура искусственных нейронных сетей, предложенная Яном Лекуном в 1988 и **нацеленная на эффективное распознавание образов**, входит в состав технологий глубокого обучения (англ. deep learning). Использует некоторые особенности зрительной коры, в которой были открыты так называемые простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки, реакция которых связана с активацией определённого набора простых клеток.

Название архитектура сети получила из-за наличия операции свёртки, суть которой в том, что каждый фрагмент изображения умножается на матрицу (ядро) свёртки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения.

# Сверточная нейронная сеть



В процессе обучения в близких к «рецепторам» скрытых слоях начнут появляться (специализироваться) нейроны, возбуждающиеся от прямых линий, разного угла наклона, затем реагирующие на углы, квадраты, окружности, примитивные паттерны: чередующиеся полосы, геометрические сетчатые орнаменты. Ближе к выходным слоям — нейроны, реагирующие, например, на глаз, колесо, нос, крыло, лист, лицо и т. д.





# Как нейронные сети формируют понимание образов



**Части** ( слои mixed4b и mixed4c )

**Объекты** ( слои mixed4d и mixed4e )

# Зависимость гибкости алгоритма машинного обучения и интерпретируемости полученной модели

