Что такое машинное обучение и анализ данных?

Александр Фонарев

Любые замечания и предложения приветствуются

http://newo.su

Спецкурс ЛКШ.Август 2013

Сборка презентации от 30.07.2013

Зачем нам все это?

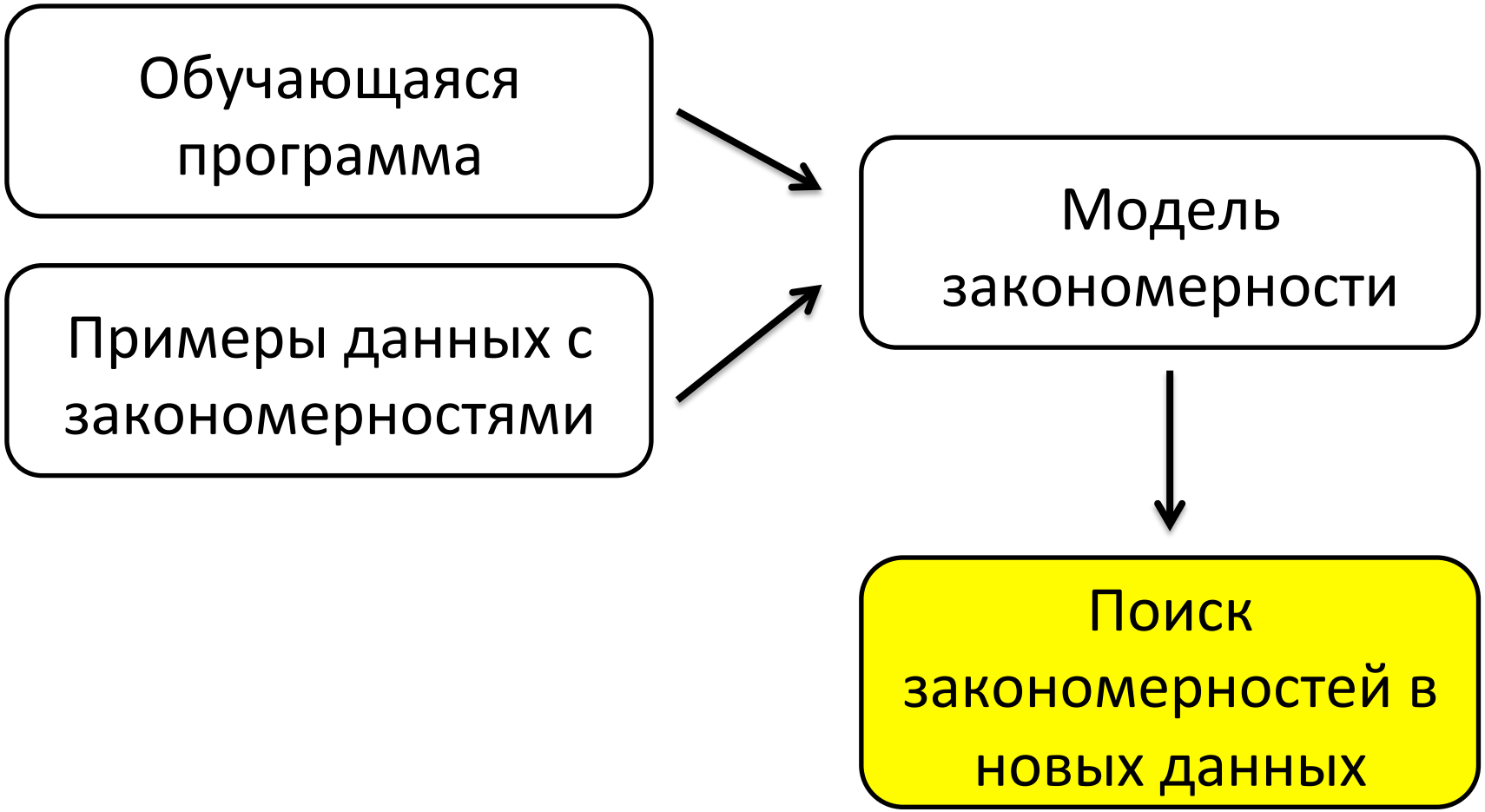
Что такое «Анализ Данных»?

- Сложно устроенные данные
- Большие объемы данных
- Надо найти или проверить закономерности в данных
- А что такое закономерность и как их искать?

Закономерности в данных

- Поиск подстроки в строке – тоже поиск закономерности
- Что делать, если мы не умеем хорошо описать закономерность?
- Почему человек понимает закономерность?

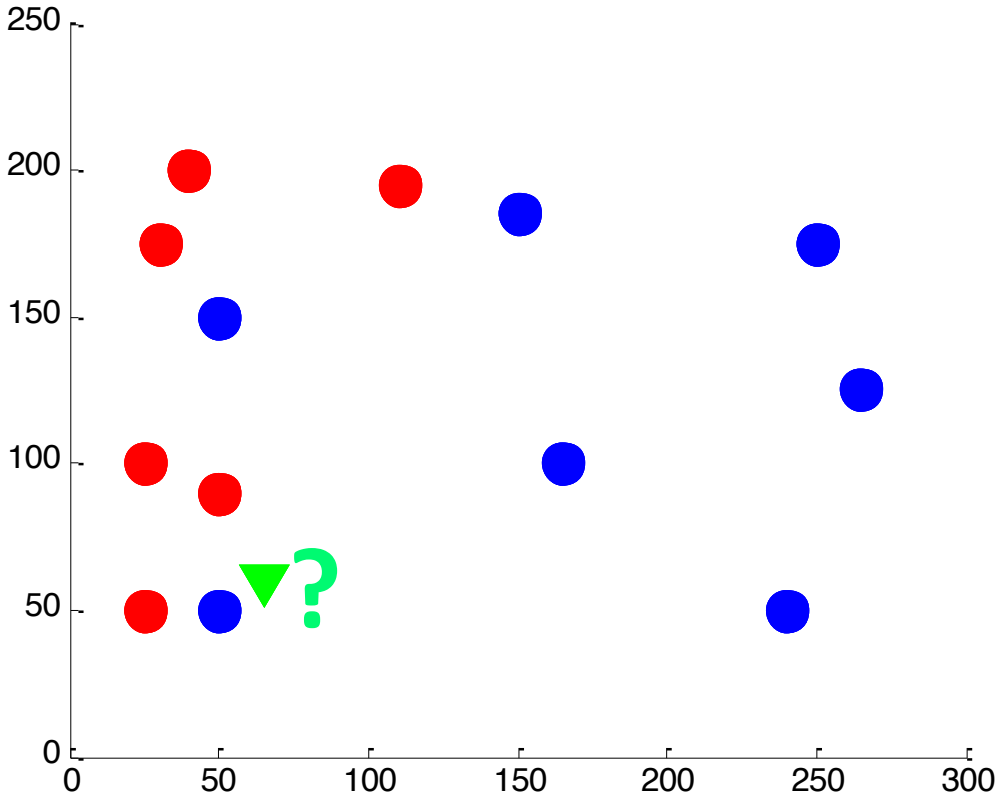
Суть машинного обучения



Простая задача и метод ближайшего соседа

Простая задача

- Синий или красный новый объект?

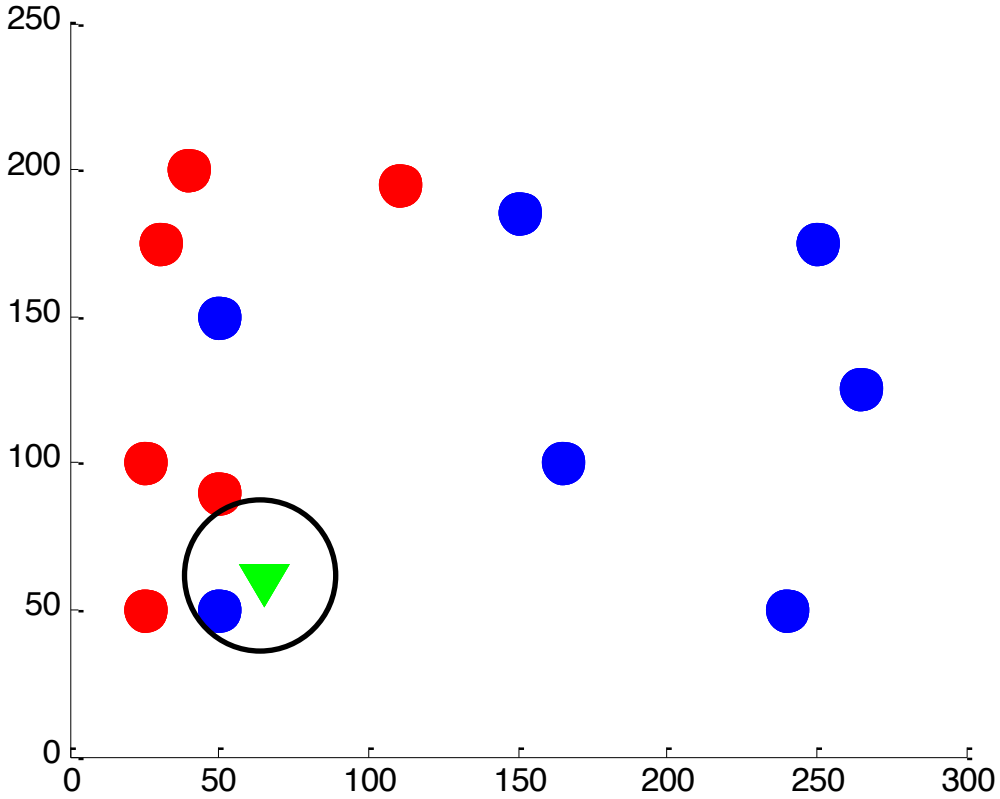


Как решать?

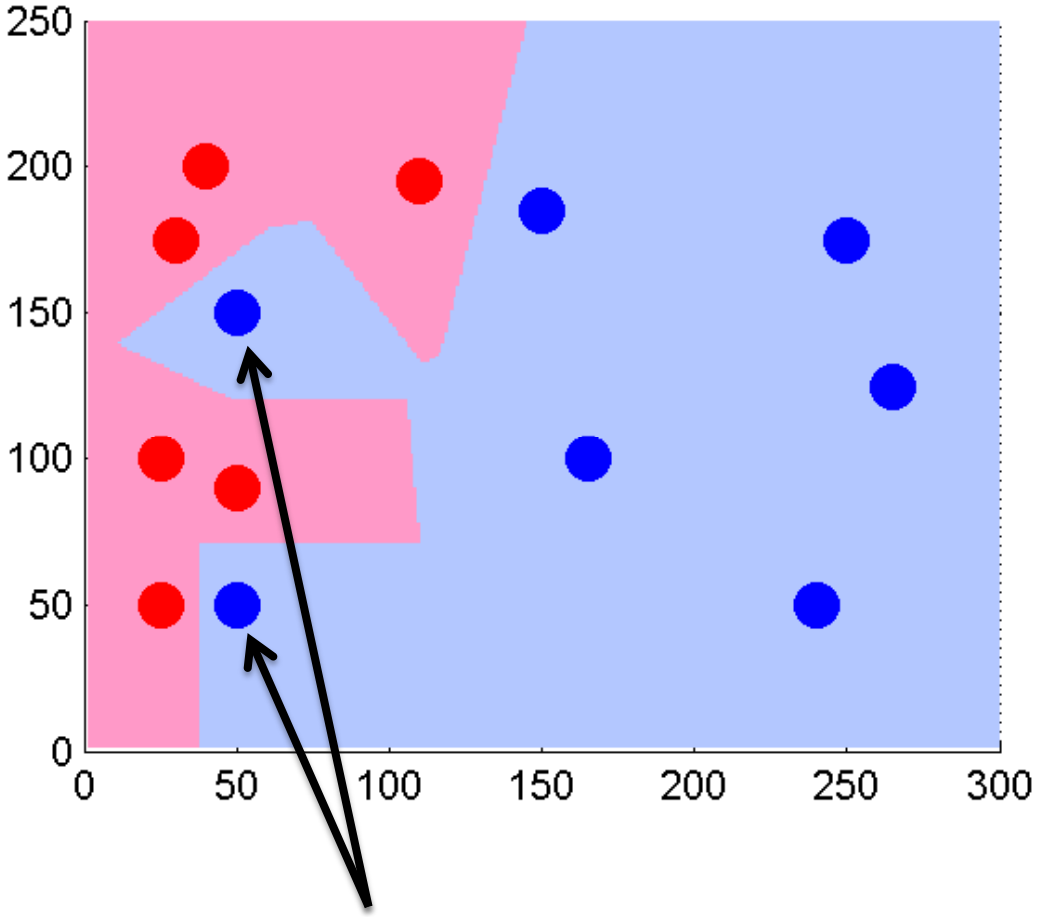
- Точного и правильного решения здесь нет
- Пытаемся решить логично с интуитивной точки зрения

Ближайший сосед

- Пусть новый объект принадлежит к тому же классу, что и его ближайший сосед



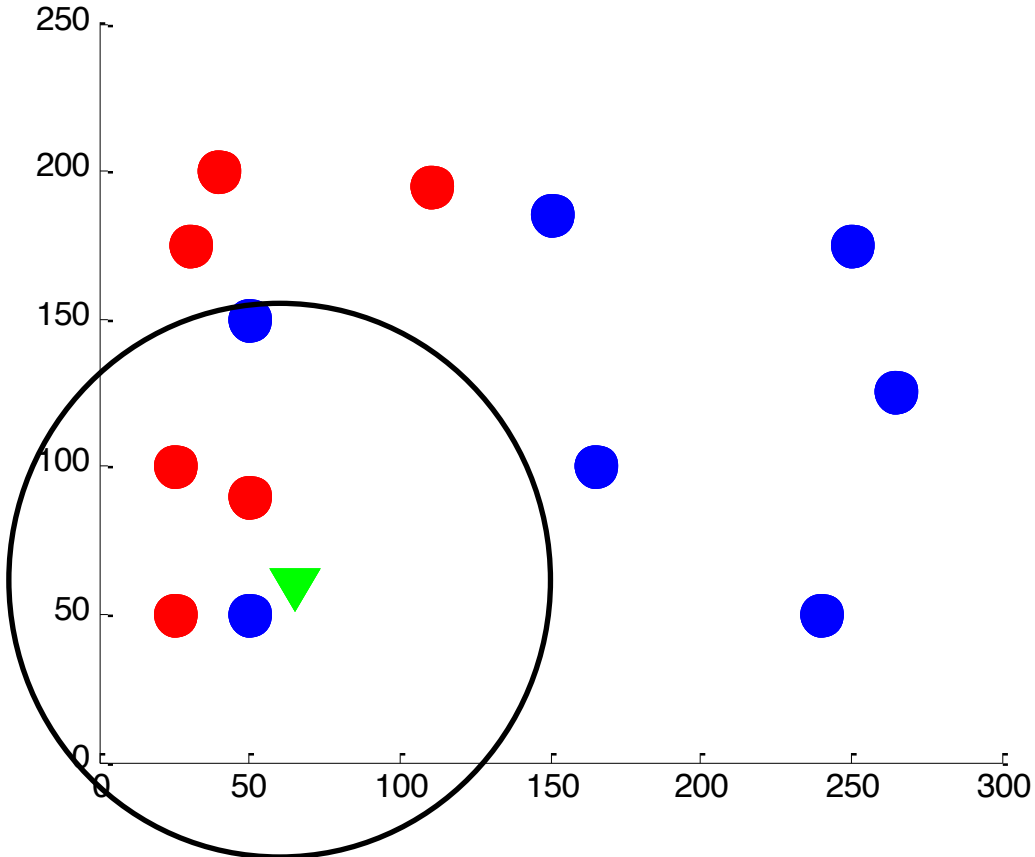
Граница разделения классов



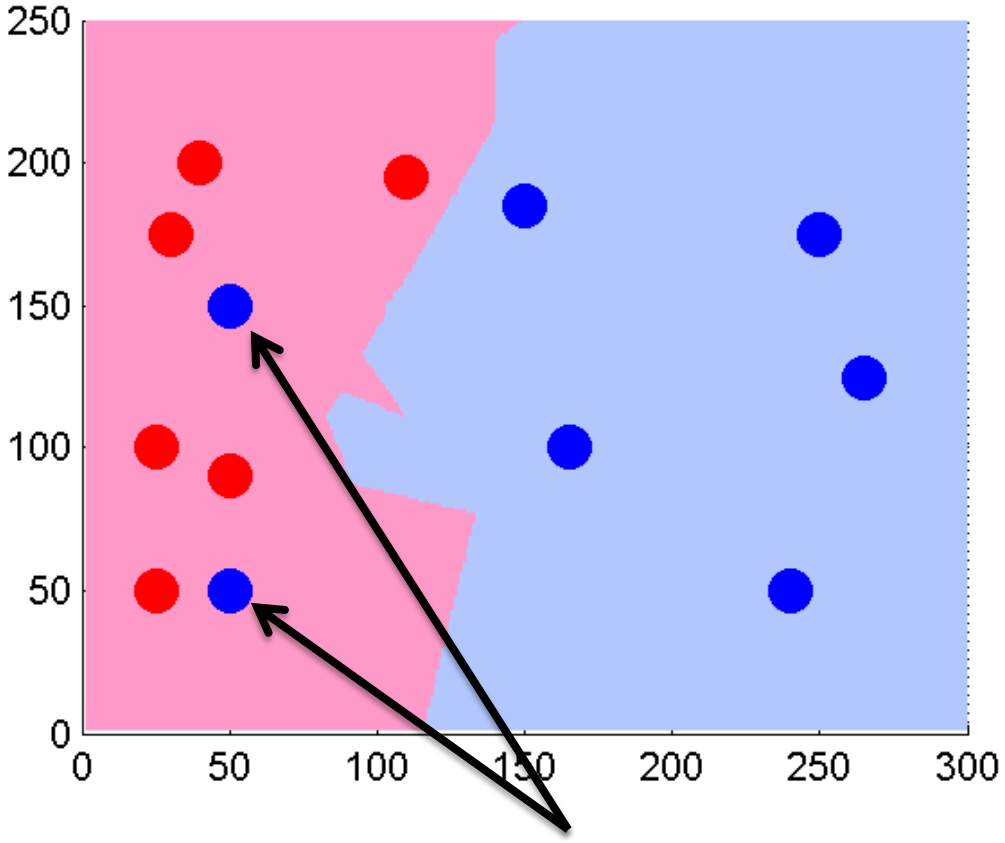
Возможно, шумовые объекты

Несколько ближайших соседей

- Новый объект принадлежит тому же классу, что и большинство из k его соседей

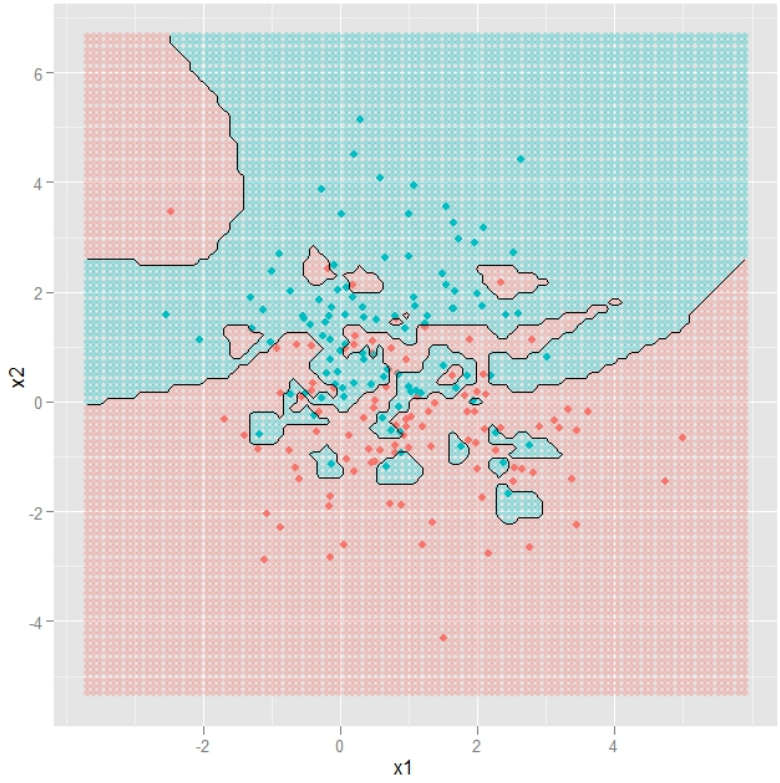


Граница разделения классов для k=5

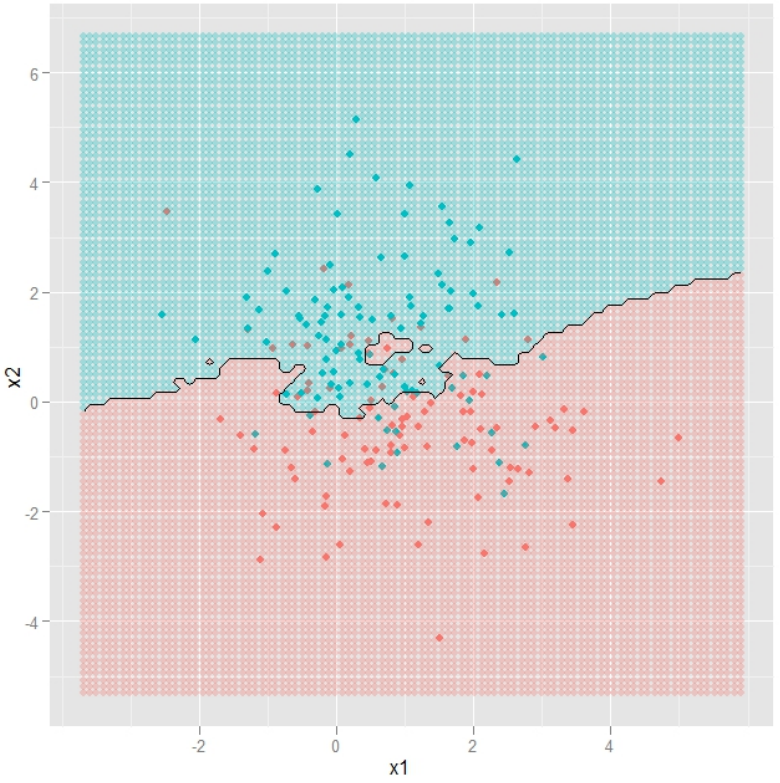


Оказывается, алгоритм дает ошибку на обучающей выборке! А это и не плохо.

А если объектов больше?



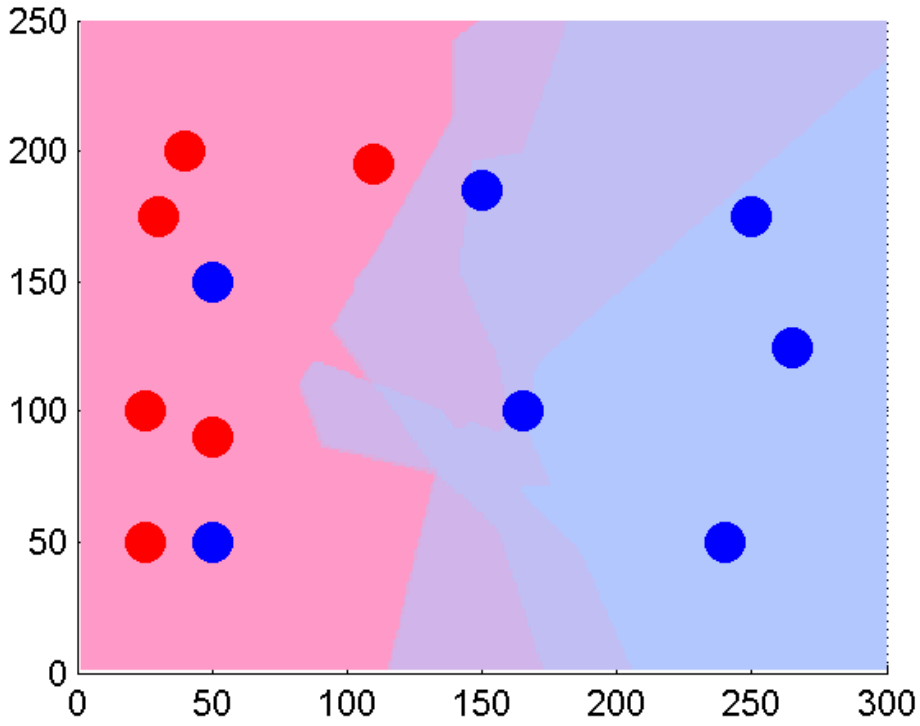
K=1



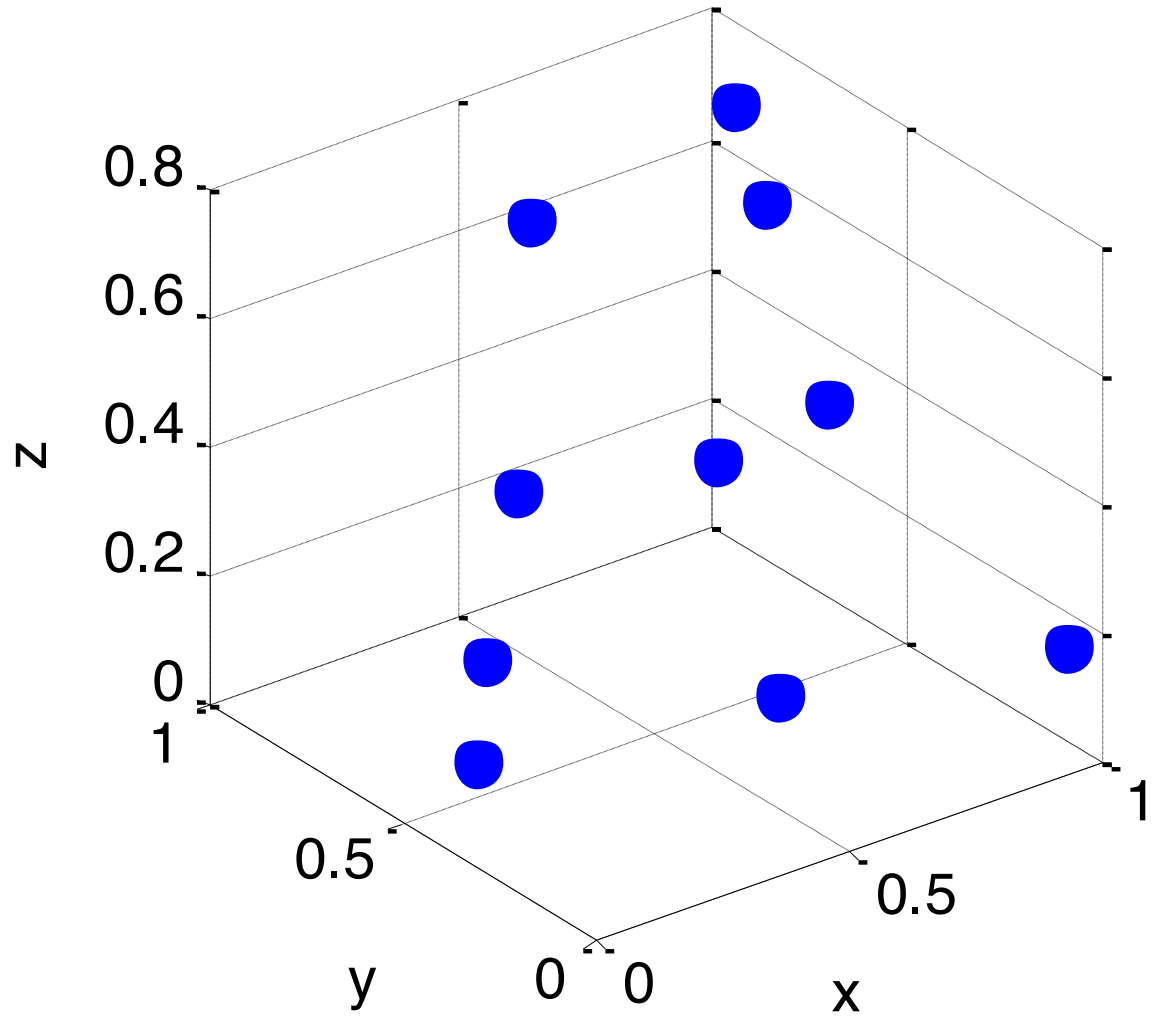
K=15

Нечеткая граница для k=5

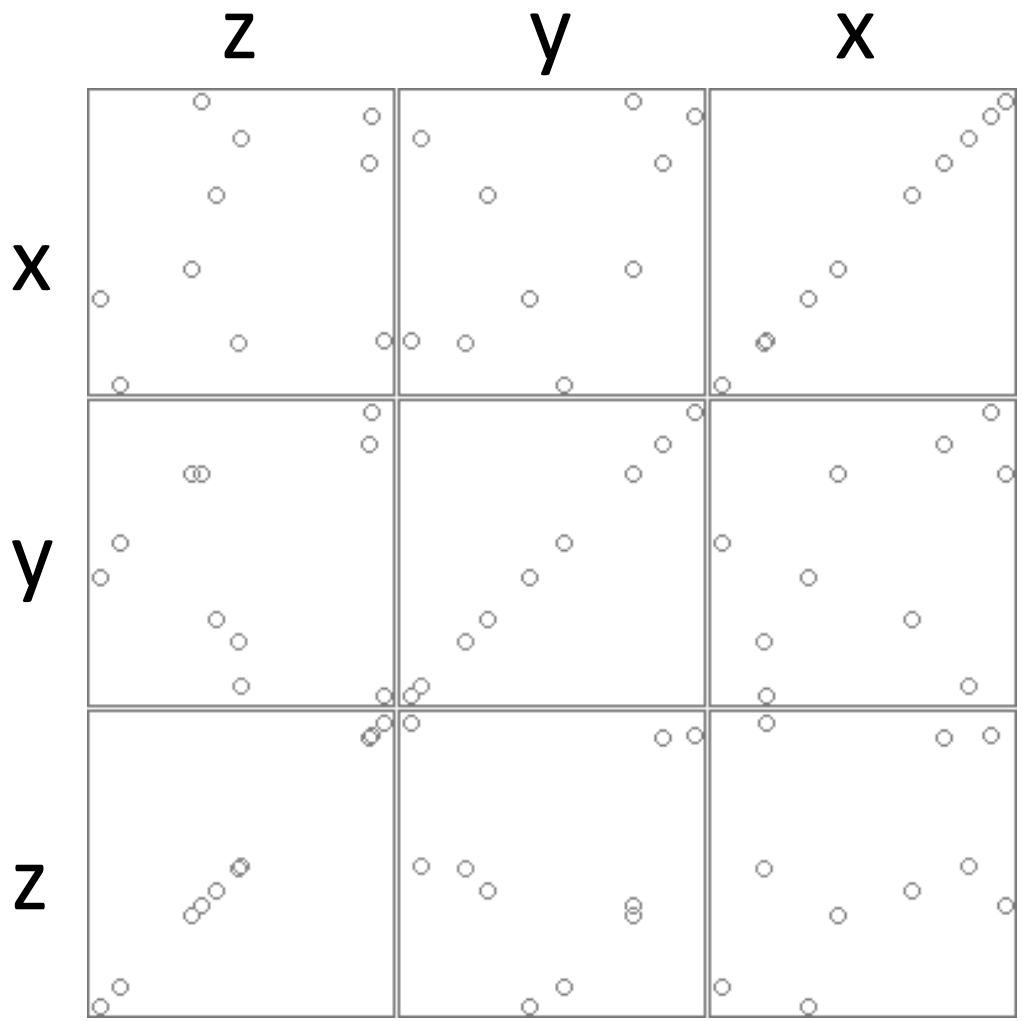
- Полутона означают, что примерно половина соседей одного класса и половина другого



Многомерное пространство

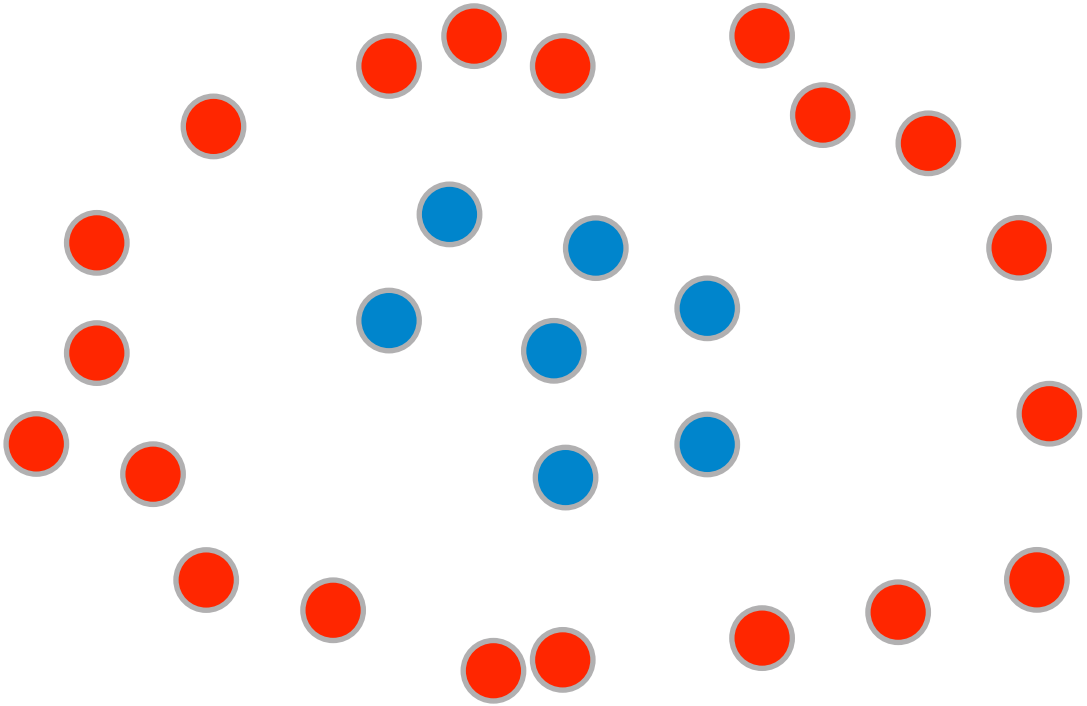


Двумерные проекции трехмерных данных

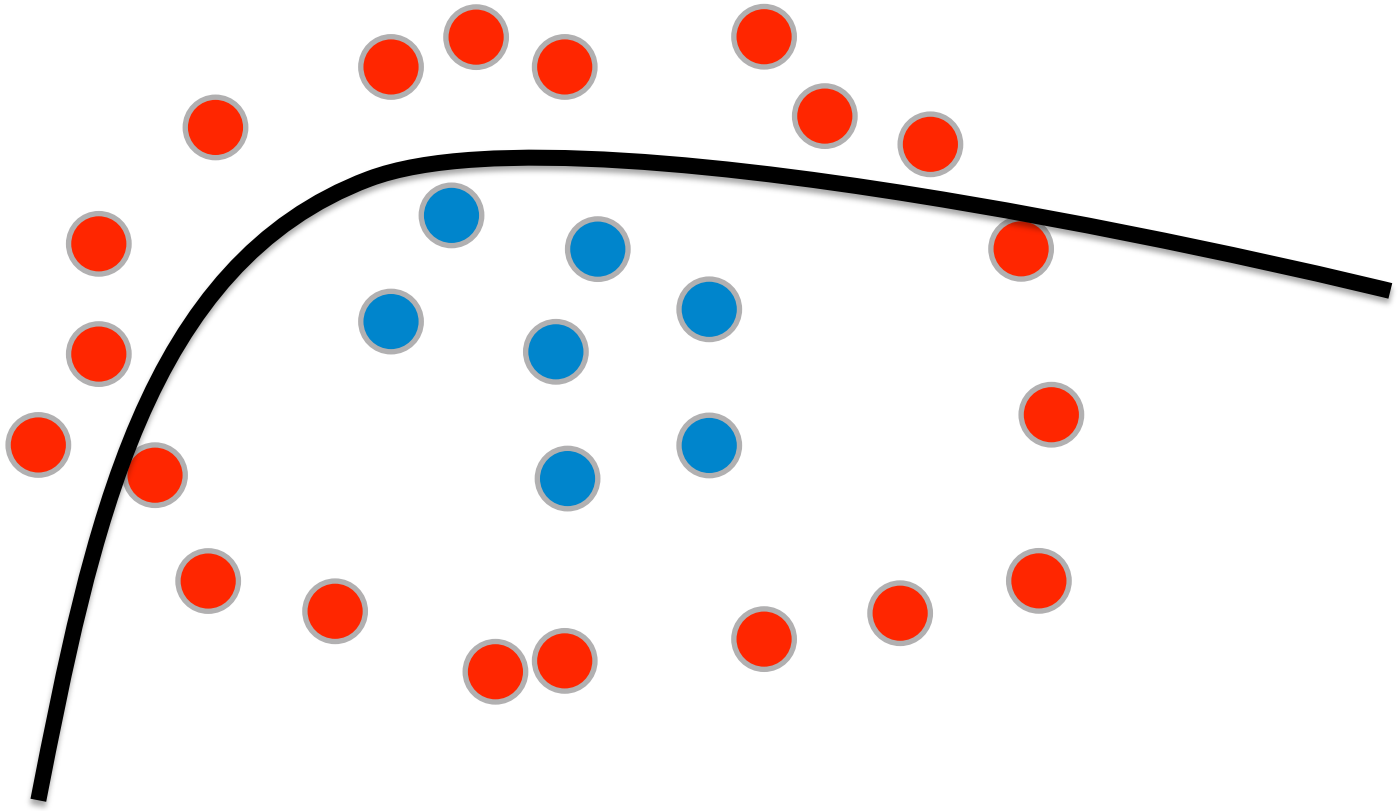


Качество и параметры алгоритмов

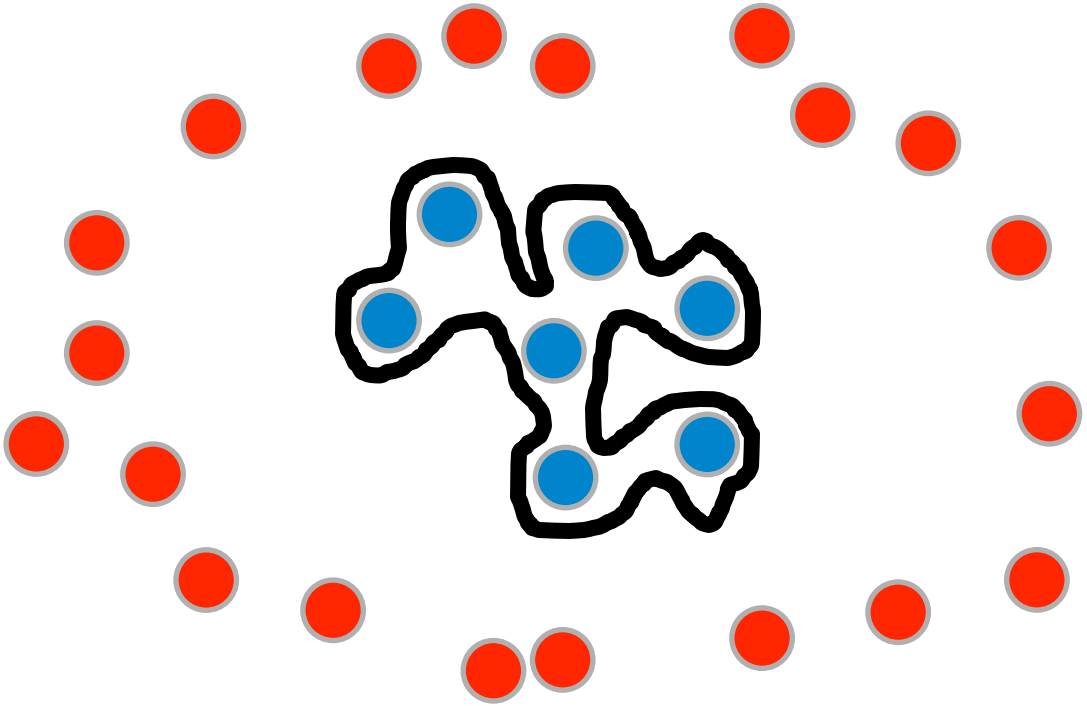
Как лучше выбрать границу?



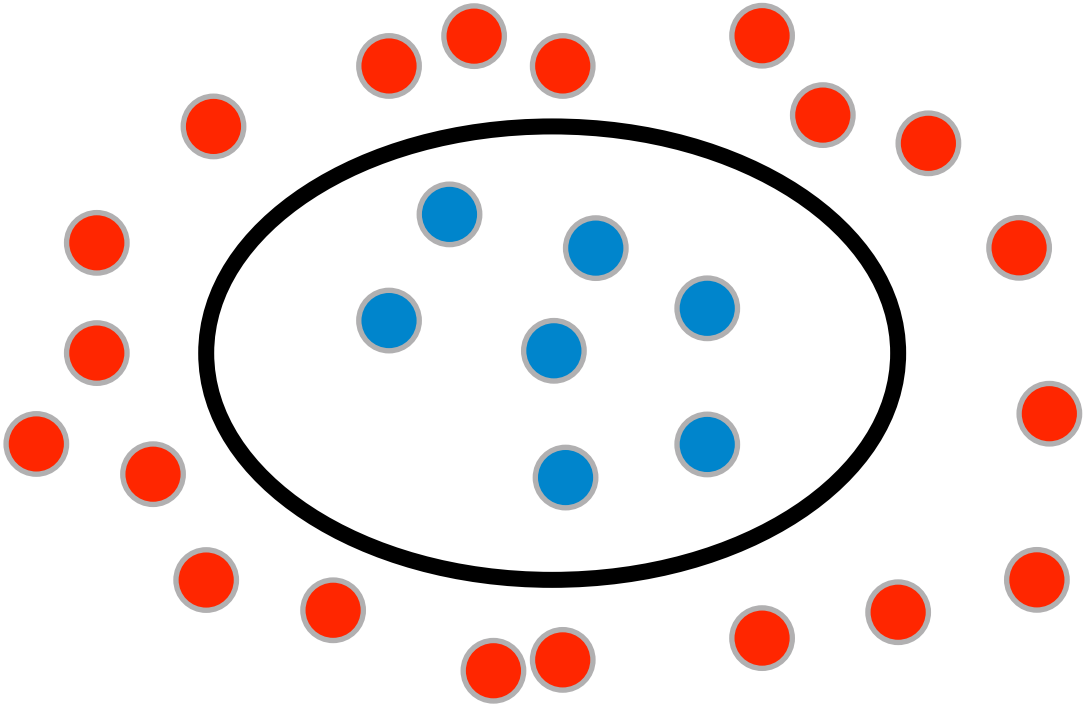
Недообученная (слабая) модель



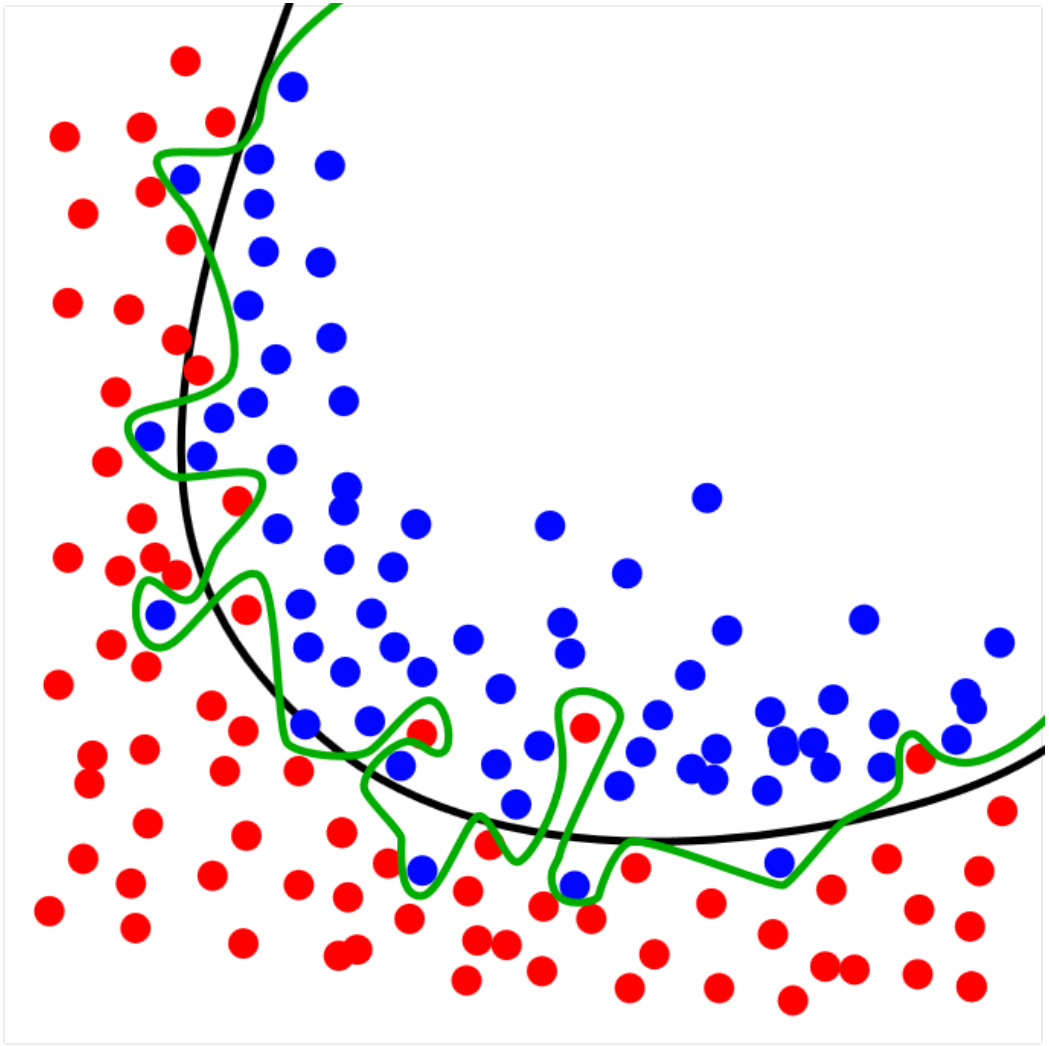
Переобученная модель



Оптимальная модель



Переобучение



Сложность модели и ее параметры

- Обычно если модель склонна переобучаться, то у нее много параметров
- Наоборот, если у модели мало параметров, то и вряд ли она переобучается

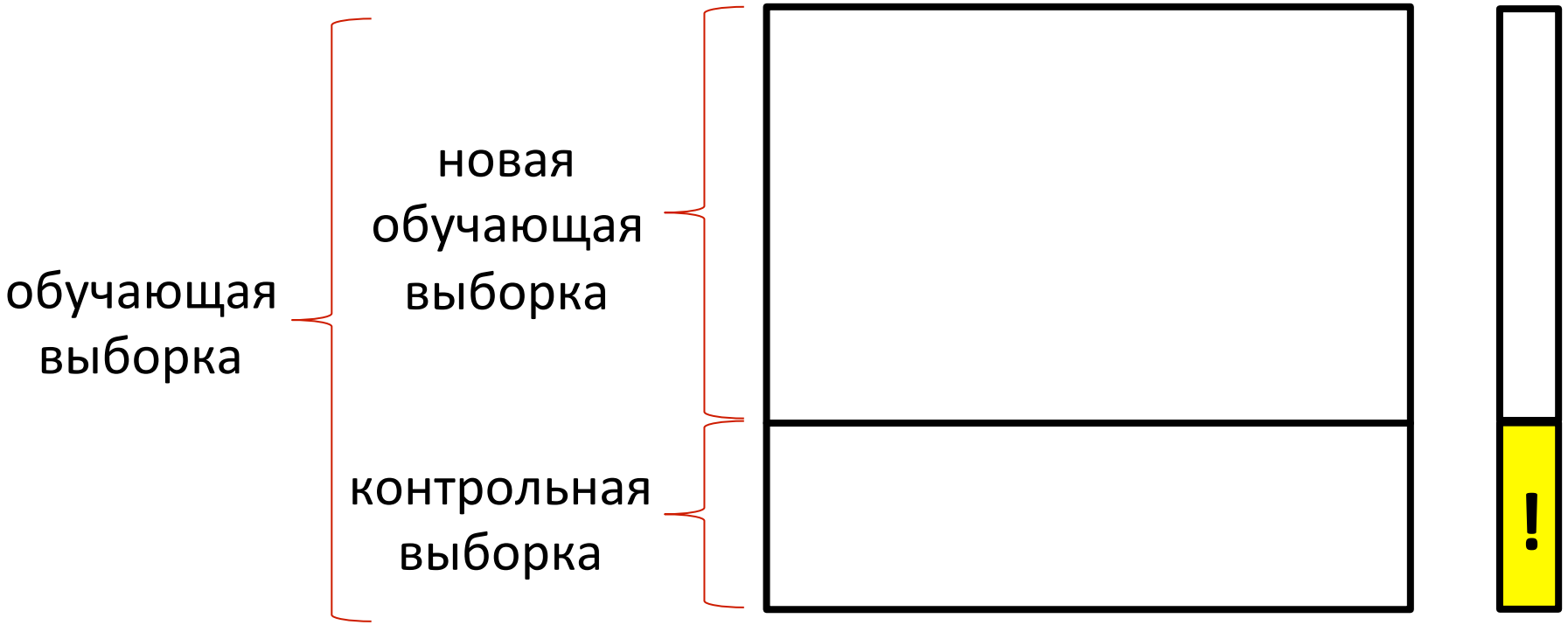
Строгая постановка задачи классификации

Какой алгоритм выбрать?

- Можно придумать много разных алгоритмов
- Качество – это доля правильных ответов на всевозможных данных
- Как понять, сильно ли алгоритм ошибается, если нам не известны правильные ответы новых объектов?

Разбиение на контроль

- Используем имеющиеся данные из обучающей выборки. Разобъем обучение на две части.
- На одной мы будем обучаться, а на второй проверять, сколько ошибок выдал алгоритм

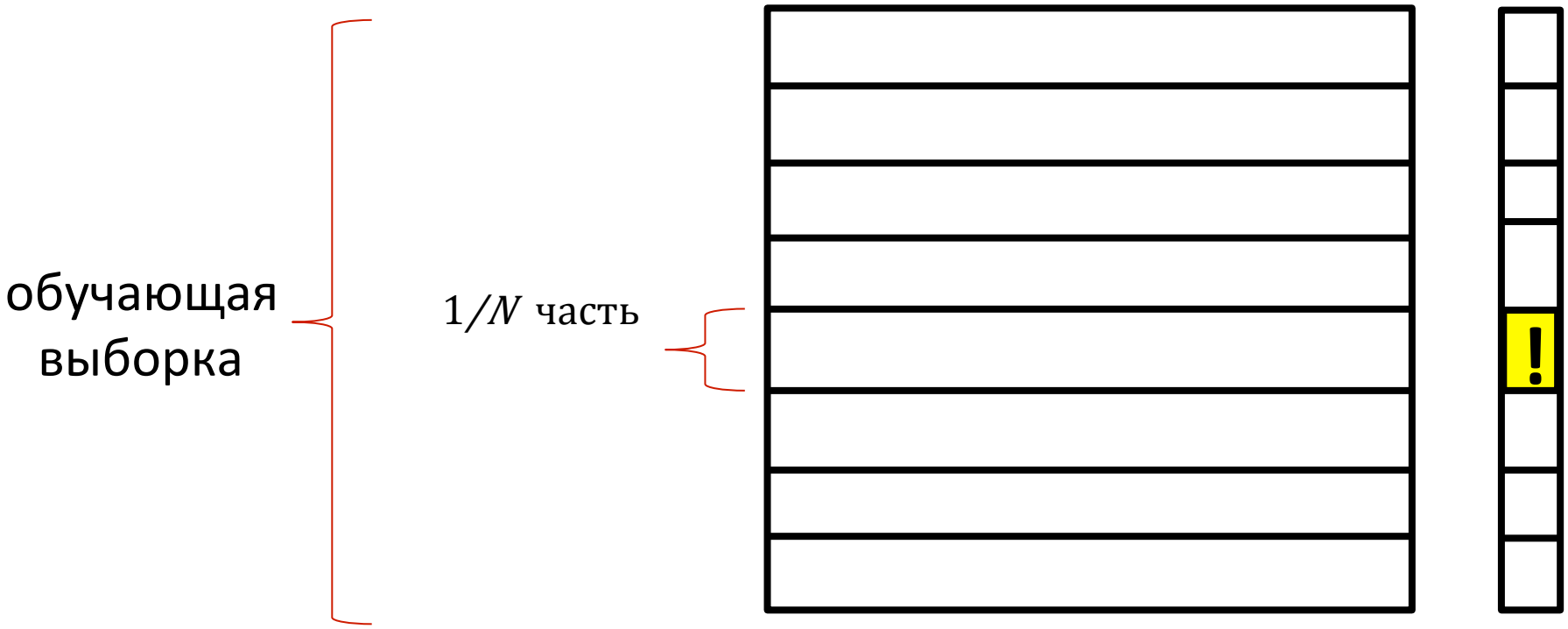


Недостатки разбиения на контроль

- Обучаемся не на всех данных, т.е. классификация получается хуже
- Проверяем качество только на малой части данных
- Как бы проверить качество на всех данных?

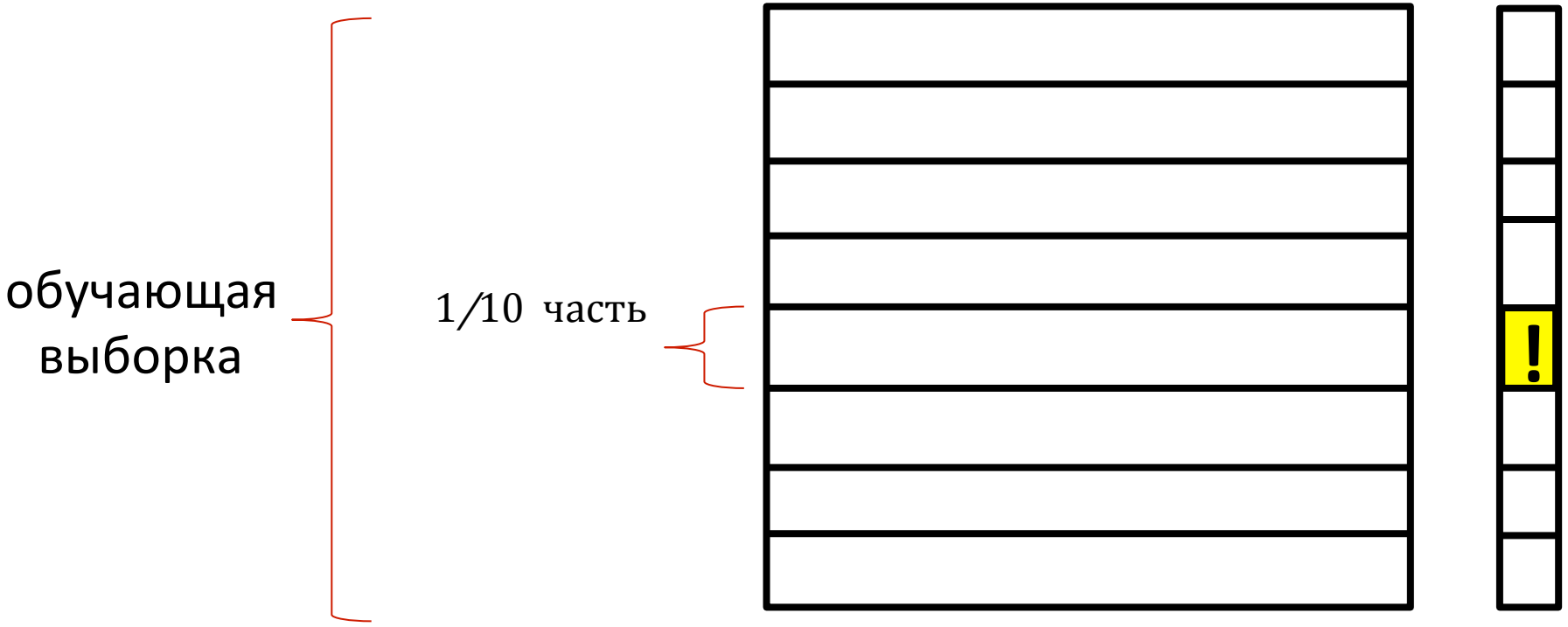
Скользящий контроль

- Разбиваем обучающую выборку на N равных частей
- Поочередной выбрасываем каждую из частей, обучаемся на остальных и оцениваем качество
- Усредняем

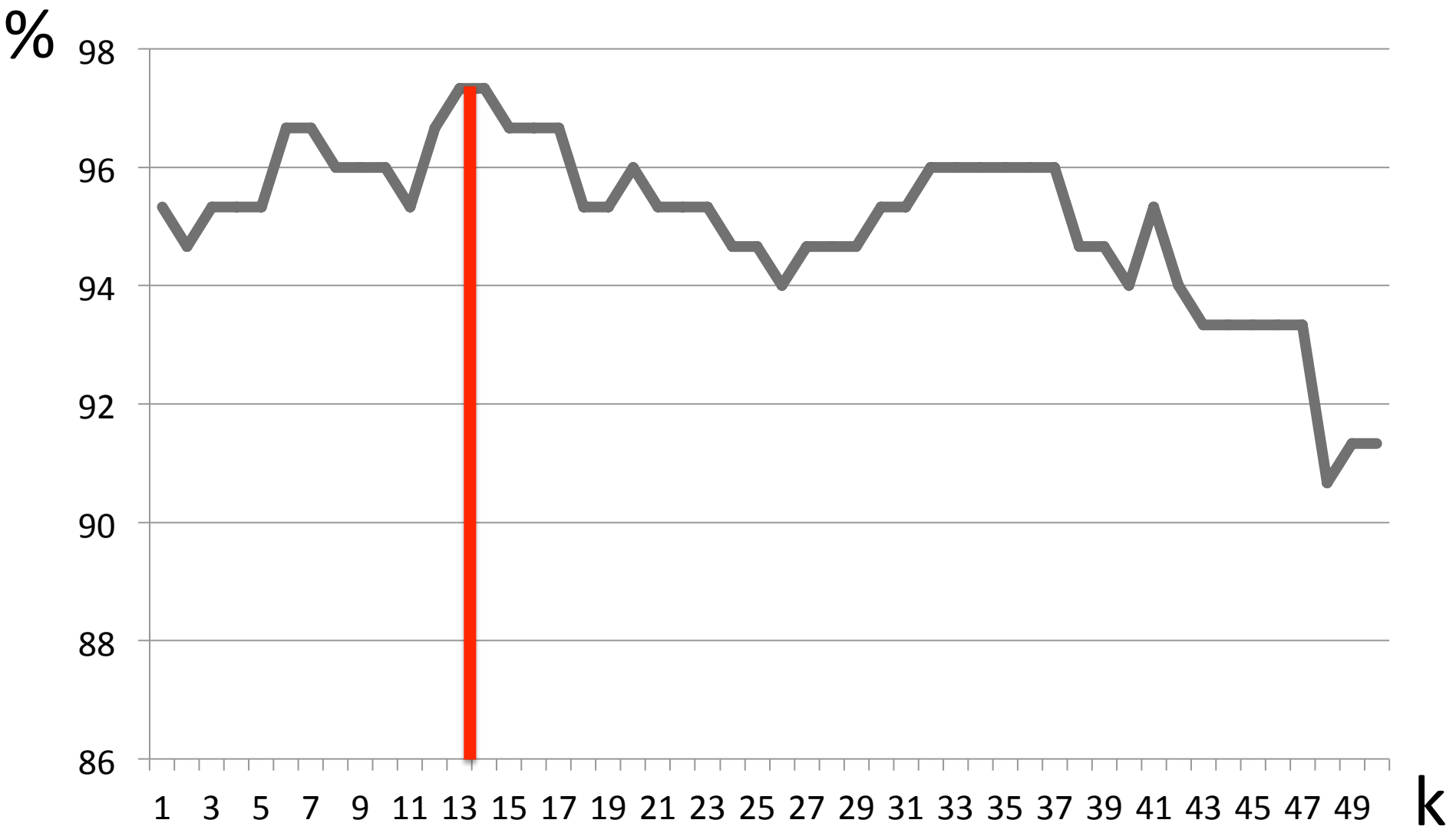


Вернемся к методу ближайшего соседа

- Какое k выбрать? Поймем, при каком k достигается наилучшее качество.
- Скользящий контроль! Произвольно разбиваем обучающую выборку на 10 равных частей, поочередно выбрасываем каждую из частей, обучаемся на остальных, оценивая качество, и все усредняем



Качество обучения в зависимости от k



Как точнее узнавать оптимальное значение параметра k?

- Видно что график скачет, почему?
- Точно узнать k тяжело
- Можно проводить кроссвалидацию много раз, а затем усреднять

Итак, что мы имеем

- Сложность: O(NM), N – количество объектов в обучении, M – количество новых объектов, O(1) – подсчет одного расстояния
- Структуры данных для ускорения: kd-tree, R-tree, Ball-tree
- Есть один оптимизируемый параметр – число соседей k (а всего параметров больше, почему?)
- Нужно знать, как считать расстояние между объектами

Про выбор расстояния

- Расстояние на плоскости между точками (a_1, a_2) и (b_1, b_2)

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

- Расстояние в многомерном случае считается аналогично

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2}$$

- Можно добавить признакам веса

$$\sqrt{10 \cdot (a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2}$$

- Можно считать вообще по-другому

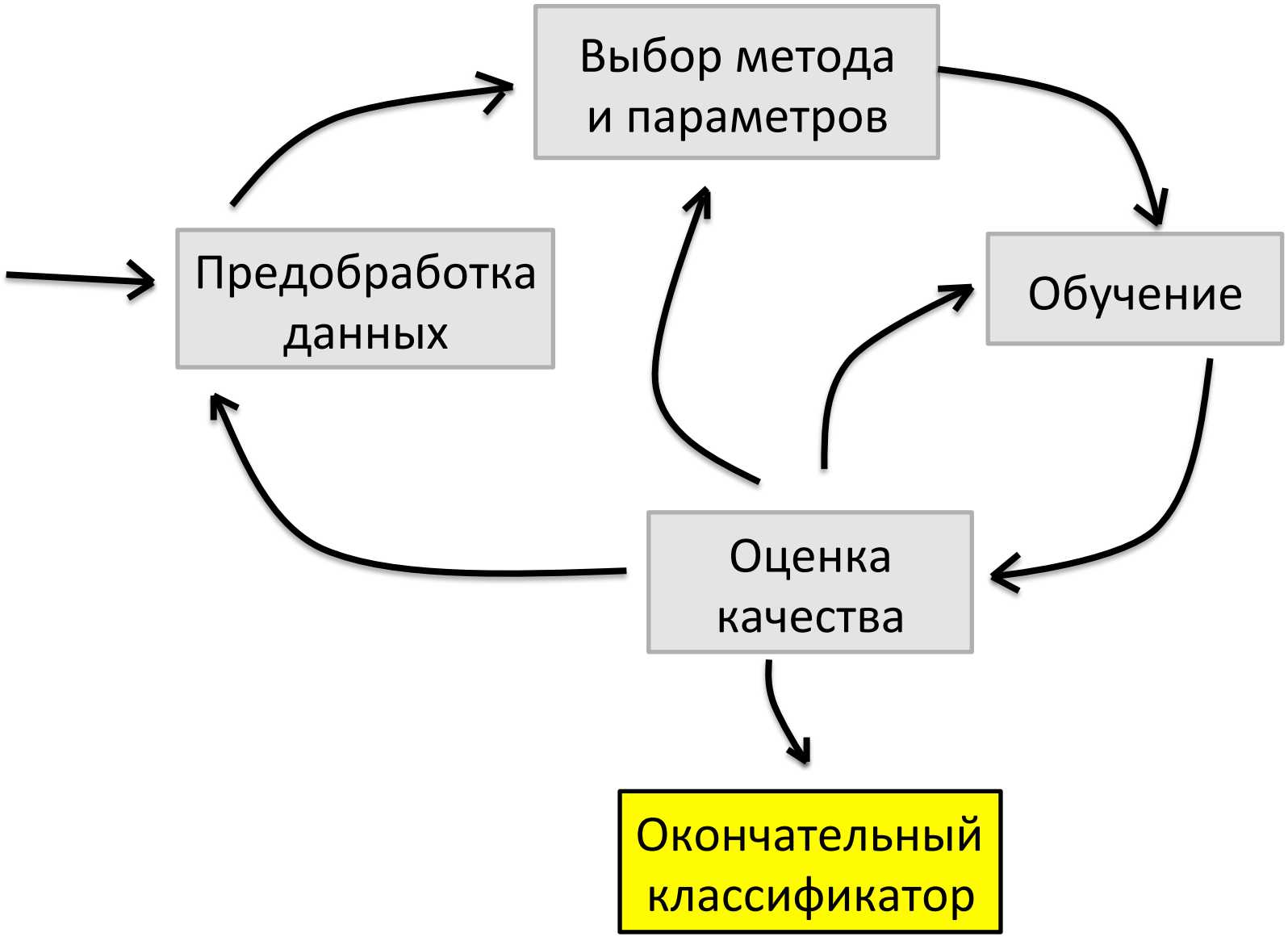
$$|a_1 - b_1| + |a_2 - b_2| + ... + |a_n - b_n|$$

- Выбор способа подсчета расстояния – основная проблема в методе ближайших соседей. Понять, как узнавать меру сходства между двумя объектами – сложная задача.

Параметры модели

- Количество настраиваемых параметров у алгоритма бывает куда больше
- Не всегда удается «тупо» перебрать все значения параметров у модели
- Придумываются разные быстрые методы нахождения параметров, близких к оптимальным (методы оптимизации)

Цикл решения задачи



Синонимы

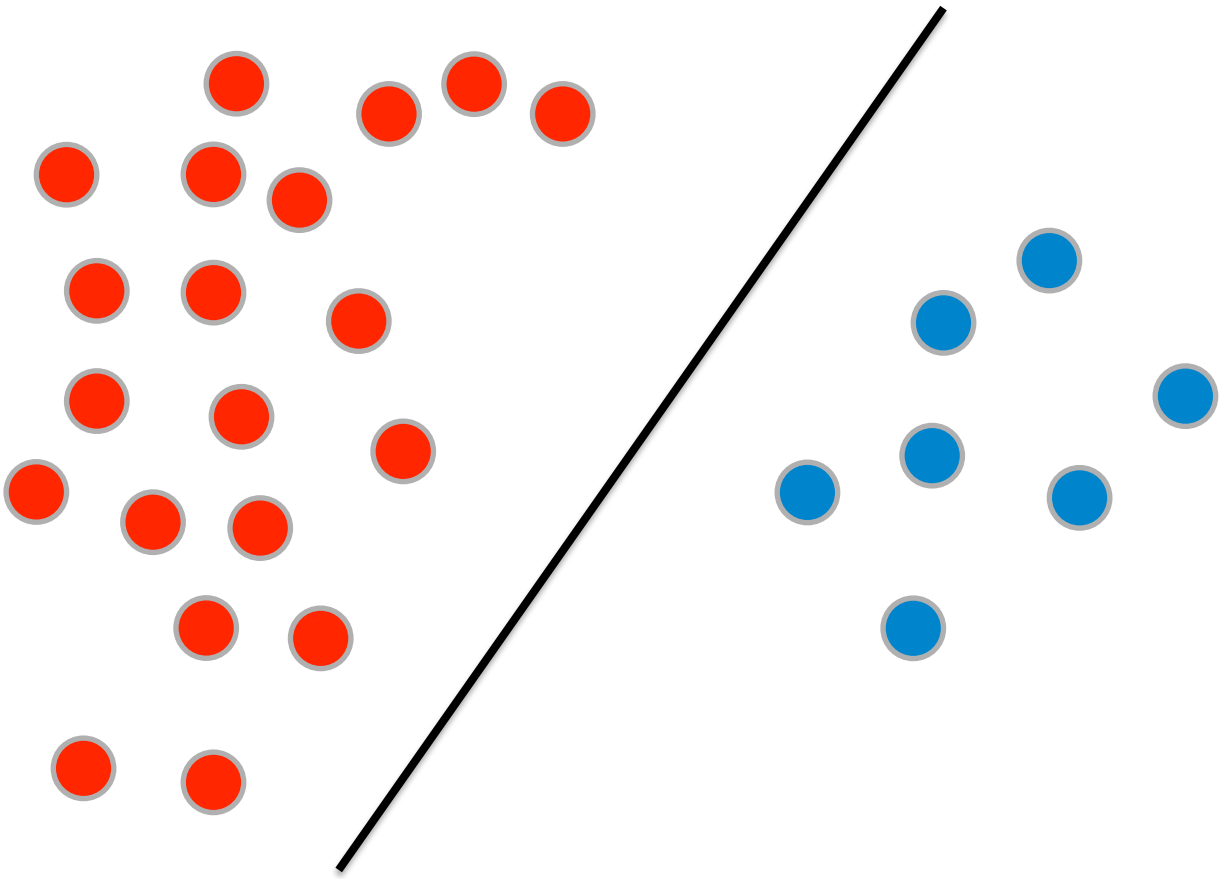
- Распознавание, предсказание, прогнозирование
- Обучающая выборка, тренировочный набор объектов, наблюдение
- Тестовая выборка, контрольная выборка, валидационная выборка, скрытая выборка
- Классы, метки классов
- Скользящий контроль, кроссвалидация

Примеры реальных задач

- Геологические данные – ищем золото
- Компьютерное зрение – распознавание чего-то на картинках
- Военная оборона – птица или ракета?
- Рекомендательные системы на сайтах
- Медицина – наличие болезни по симптомам
- Прогнозирование пробок
- Распознавание сигналов головного мозга
- Сайт научных статей - категоризация текстов
- Кредитный скоринг – надежность клиентов банка
- И еще очень много чего...

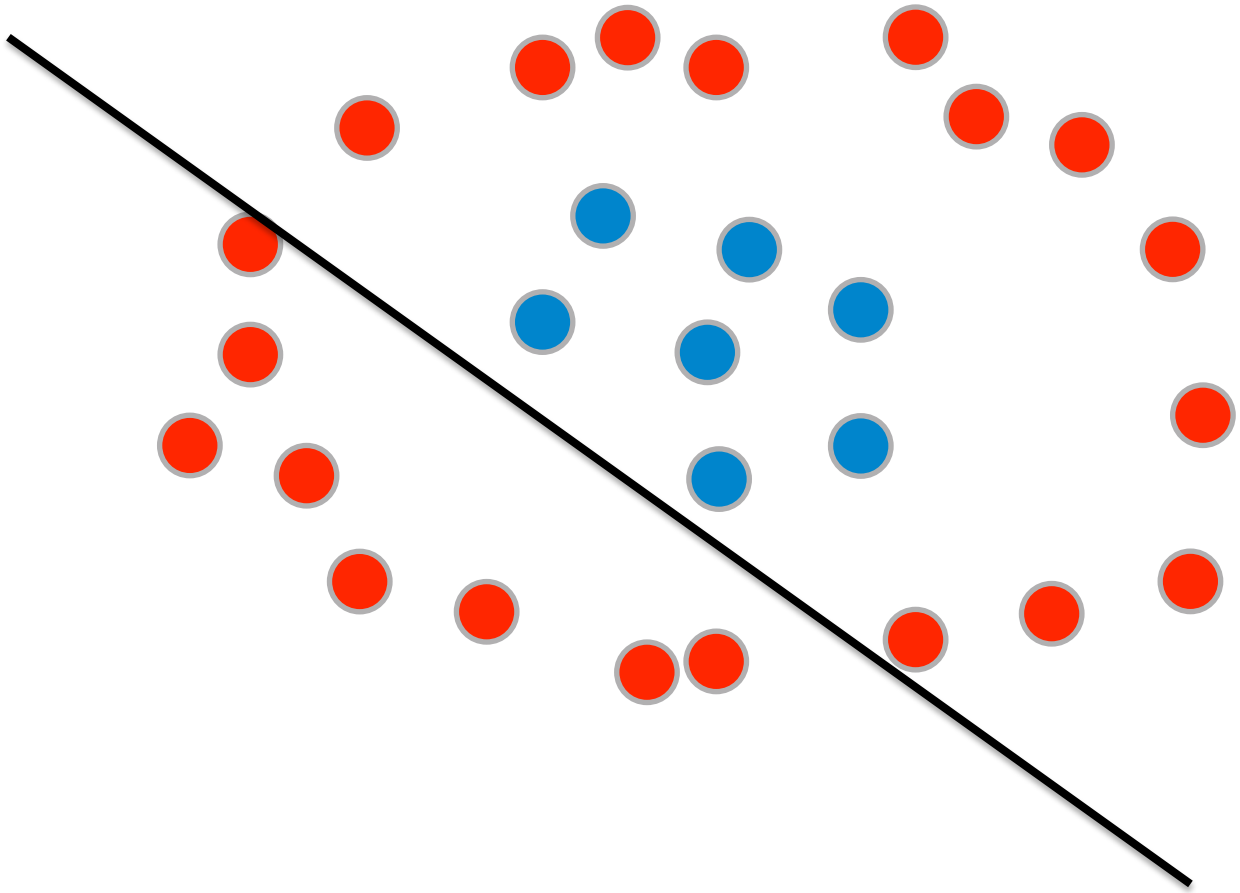
Линейная классификация и ее производные

Пусть граница – прямая



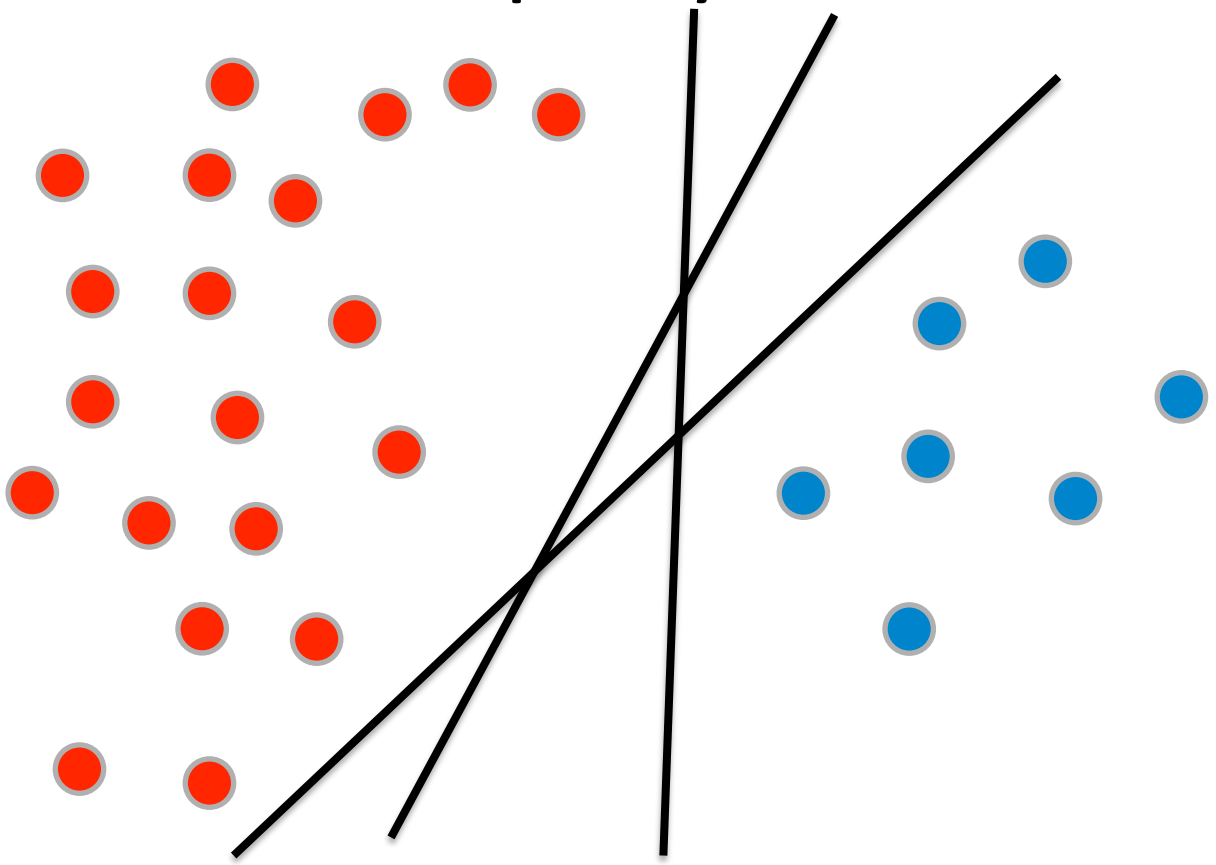
Мало параметров

Иногда прямая плохо помогает



Мало параметров – вряд ли переобучится

Есть много способов провести прямую

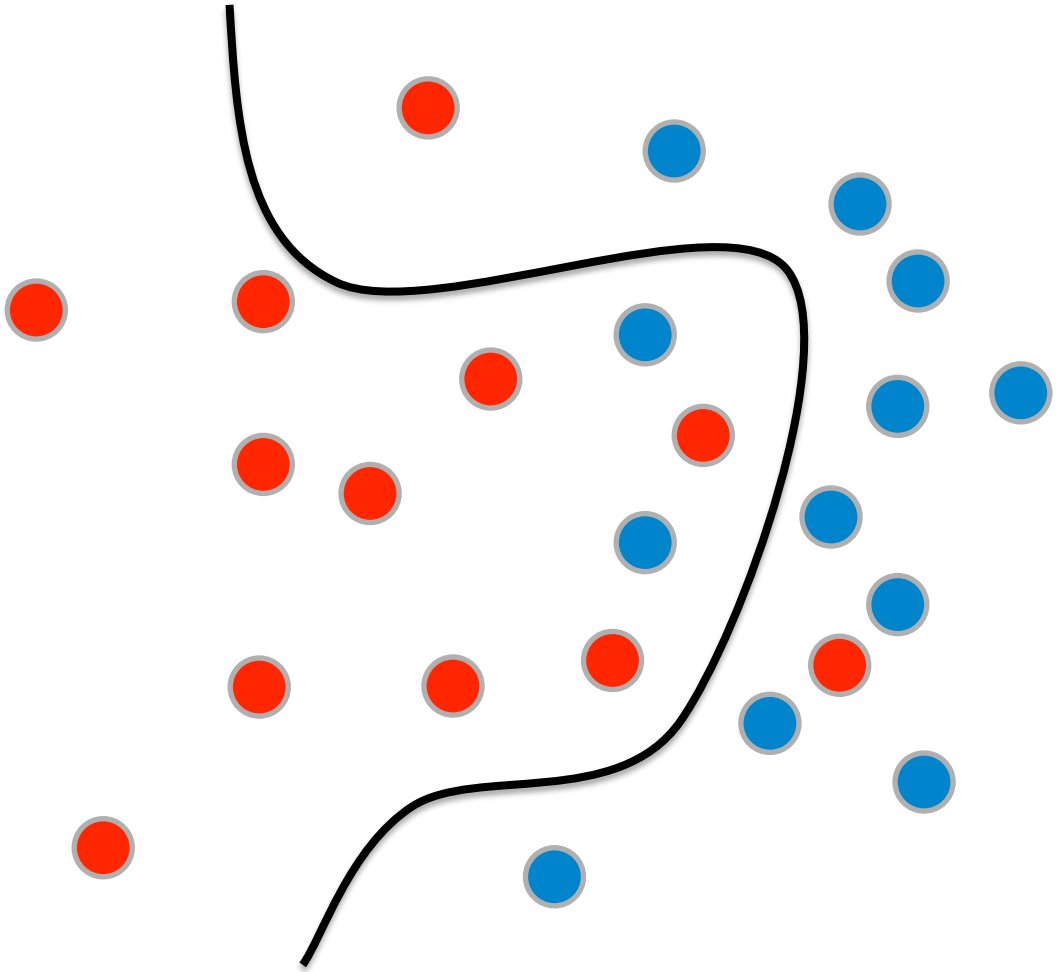


Какая прямая лучше?

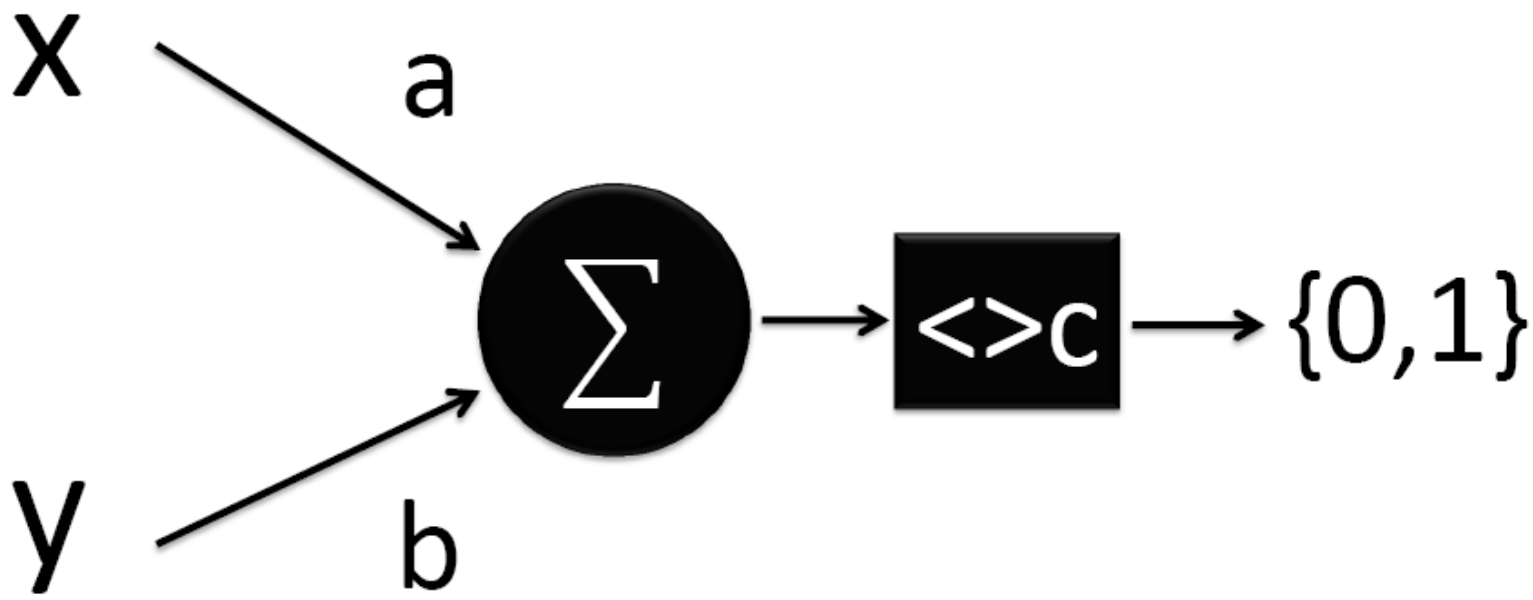
Гиперплоскости в многомерных пространствах

- В двумерном случае – прямая, в трехмерном – плоскость, дальше – гиперплоскость
- Главное, что она линейна и делит все пространство на два полупространства
- Есть много методов, умеющих строить «хорошие» гиперплоскости (например, за счет максимизации зазора между классами)

Возможны обобщения на сложные границы



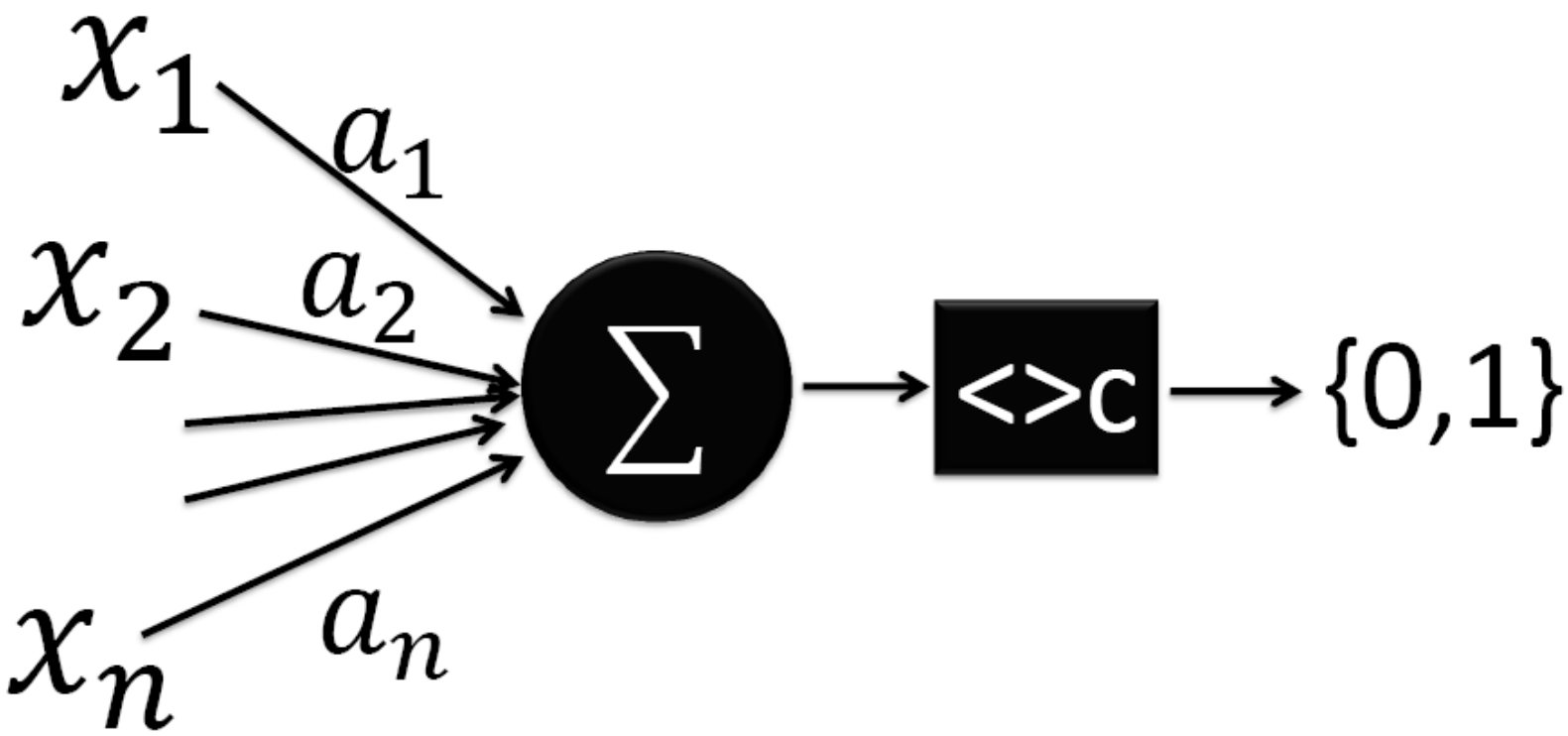
Представим в виде схемы



$$a \cdot x + b \cdot y <> c$$

Граница – прямая

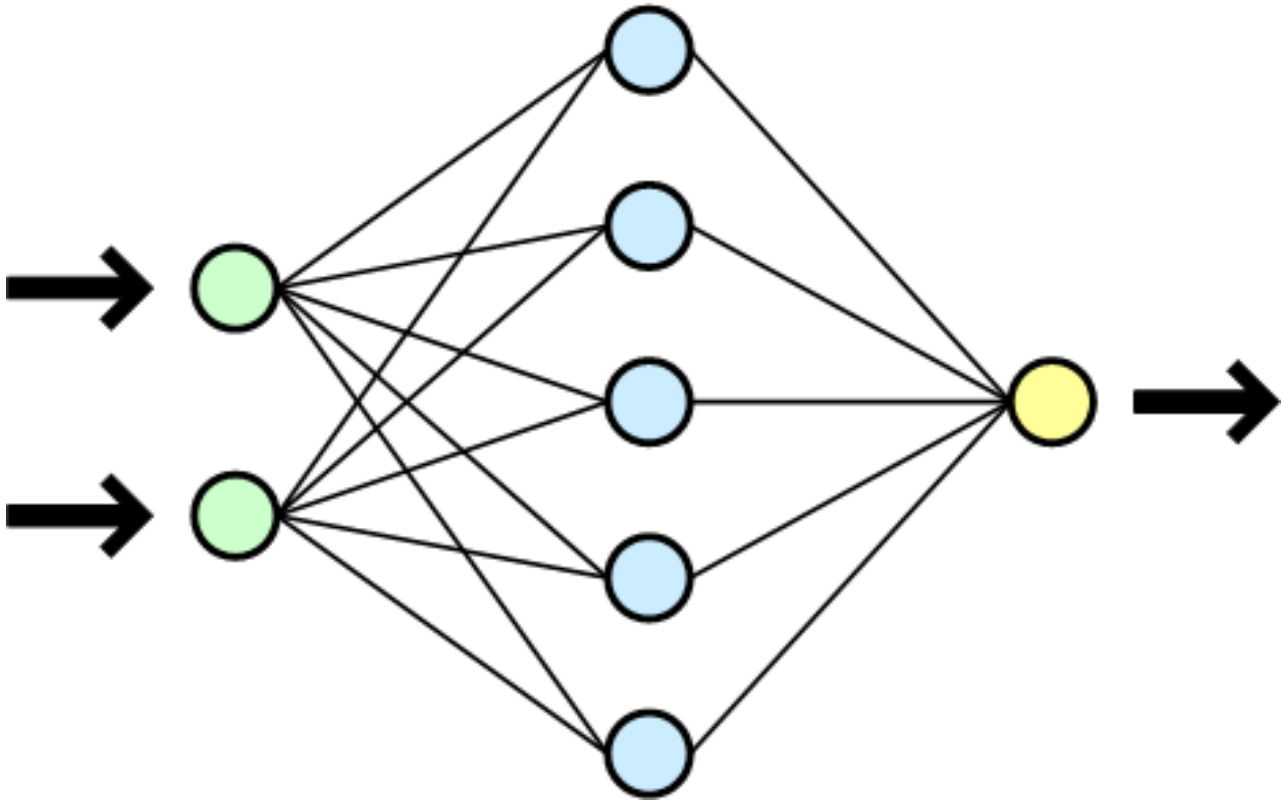
Аналогично для n входов



$$a_1 \cdot x_1 + \ldots + a_n \cdot x_n <> c$$

Граница – n-мерная гиперплоскость

Можно как угодно соединять



Получится нейронная сеть

Преимущества нейронных сетей

- Умеют подстраиваться под любые данные
- Похожи на мозг человека

Недостатки нейронных сетей

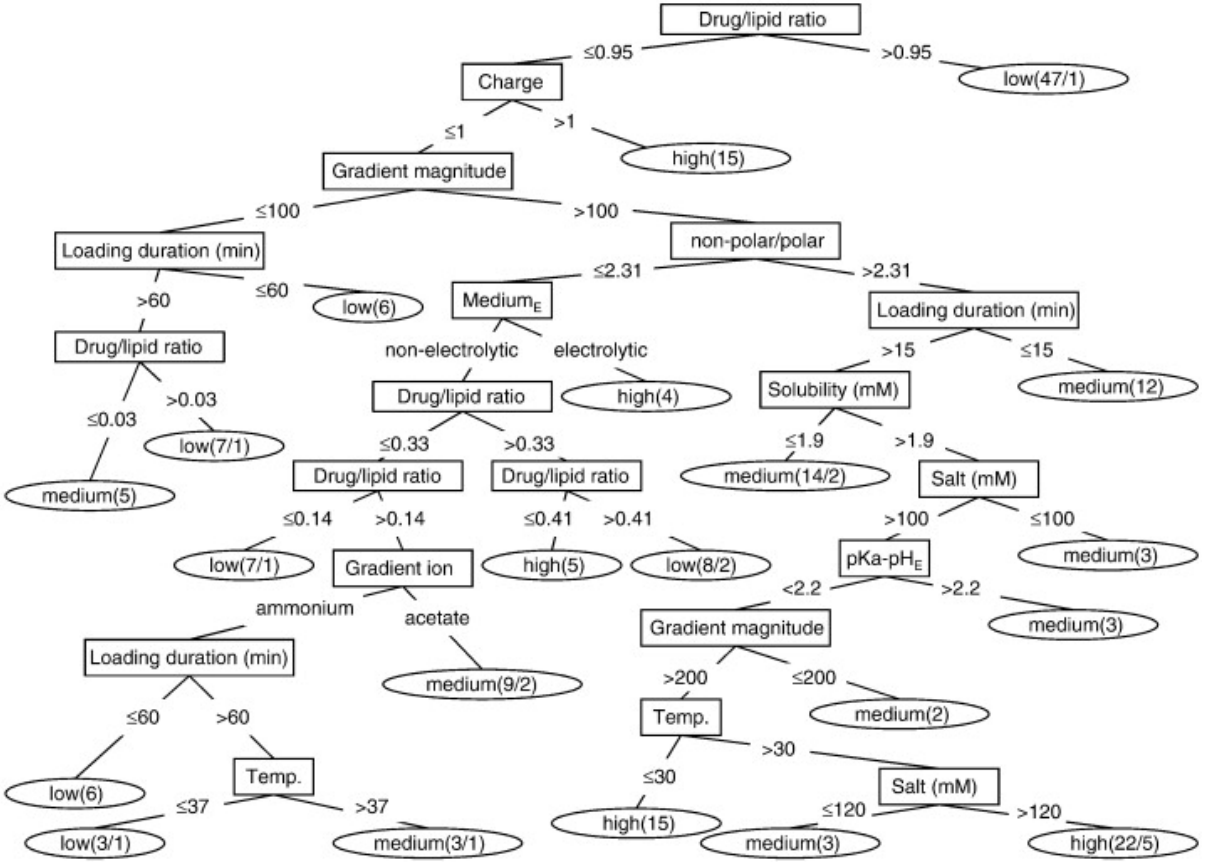
- Нейронная сеть часто очень сильно переобучается (много параметров)
- Алгоритм обучения может застопориться (так называемый «паралич» сети)
- Оказывается, имеют мало общего с реальными нейросетями в человеческом мозгу

Однако

В последний год были придуманы новые мощные алгоритмы настройки. Нейронные сети переживают новый пик популярности.

Решающие деревья и композиции над ними

Экспертные системы, составленные вручную

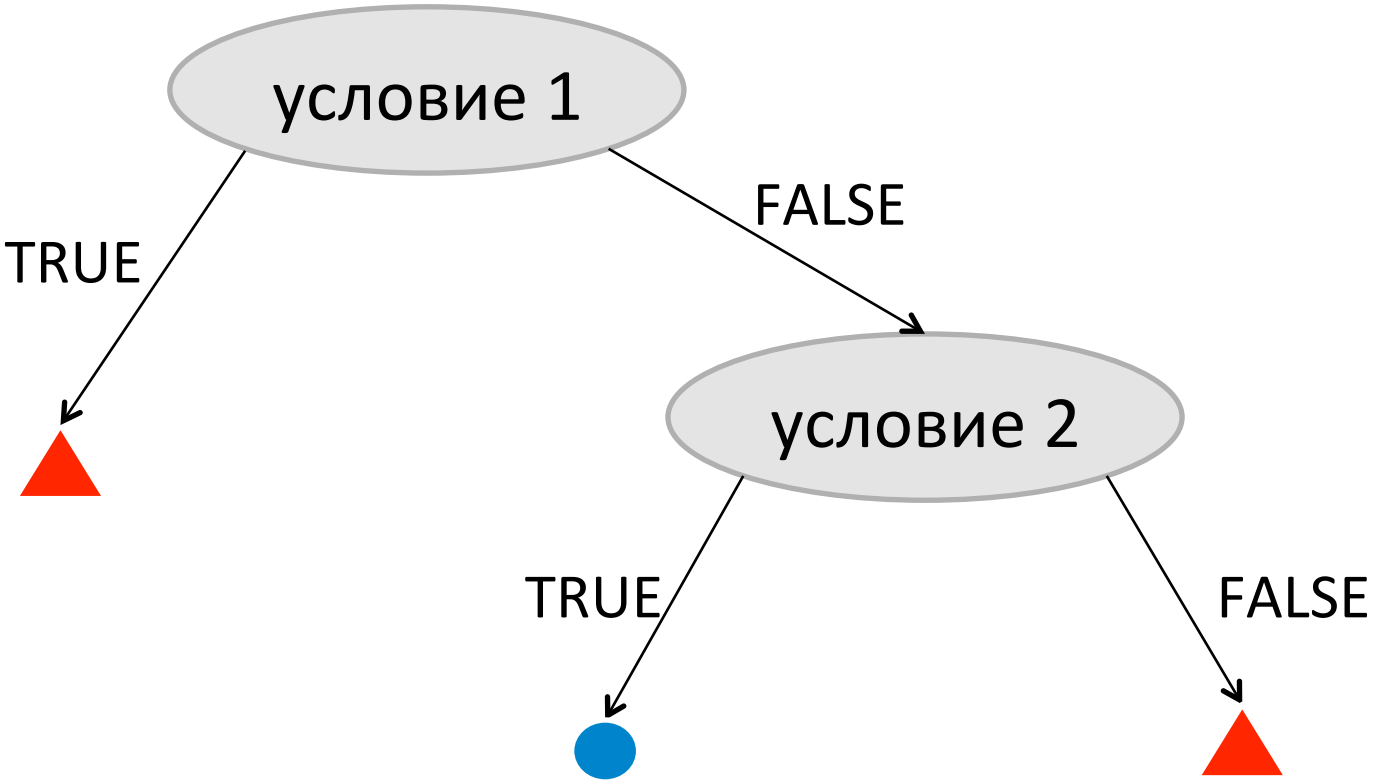


Решающее дерево для предсказания эффективности введения лекарства

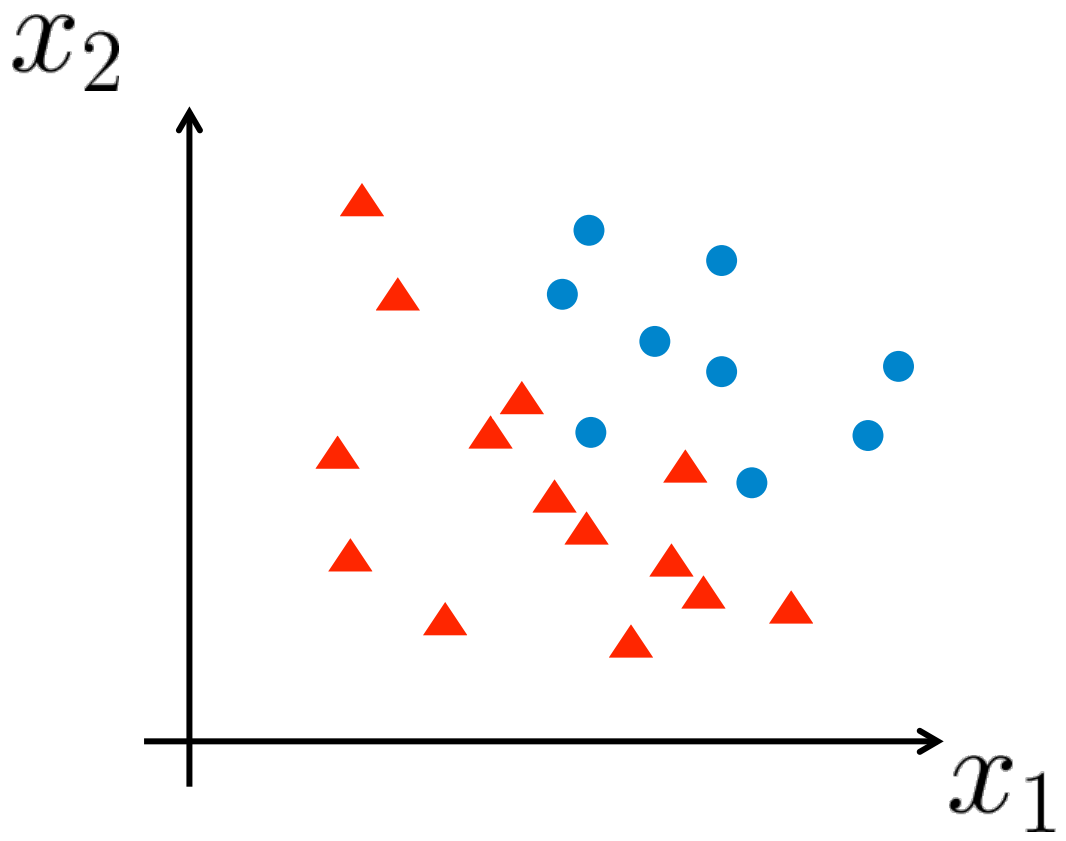
В чем проблемы?

- Правила составлялись вручную экспертами
- Мнения экспертов расходятся
- Эскперты могут ошибаться
- Эксперт не в состоянии проанализировать все данные

Построим дерево автоматически



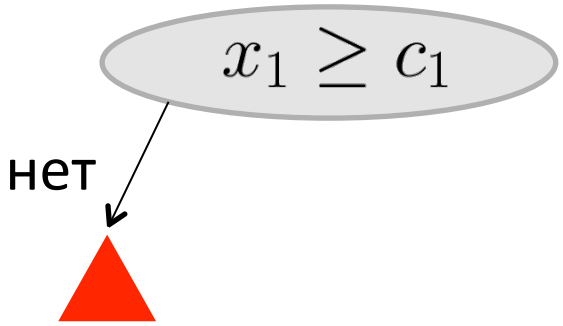
Какие условия будут в дереве?

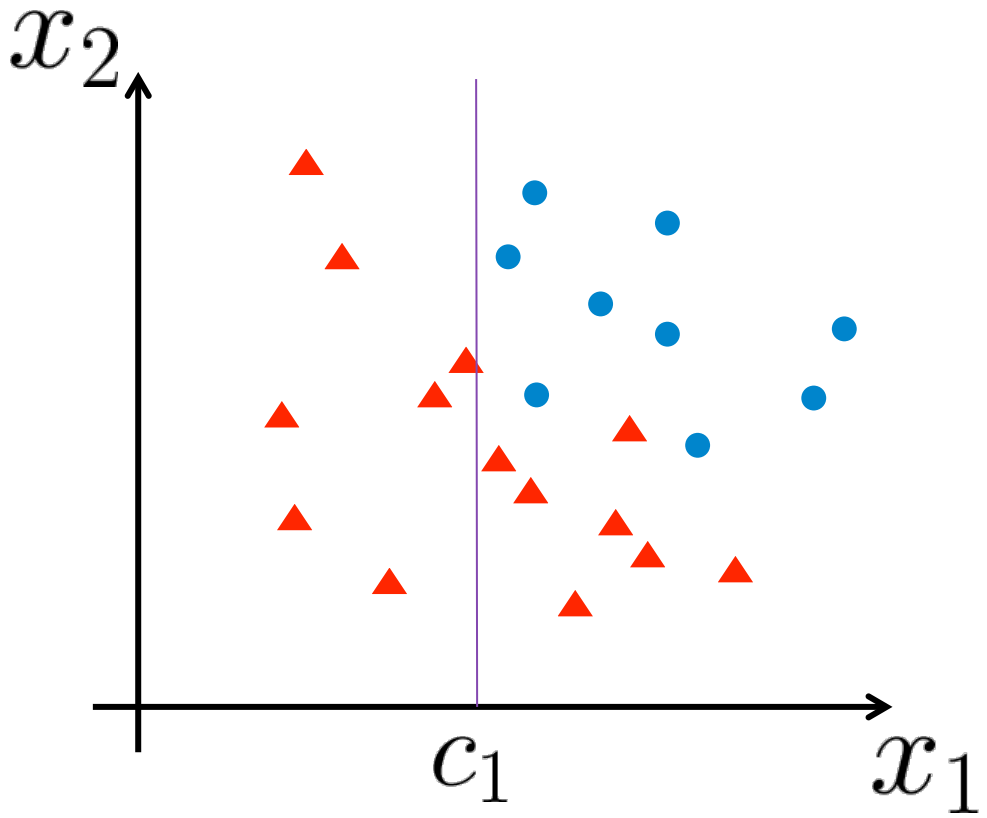


Попробуем использовать пороговые условия перехода в виде пороговых правил: x>c

Начнем строить дерево

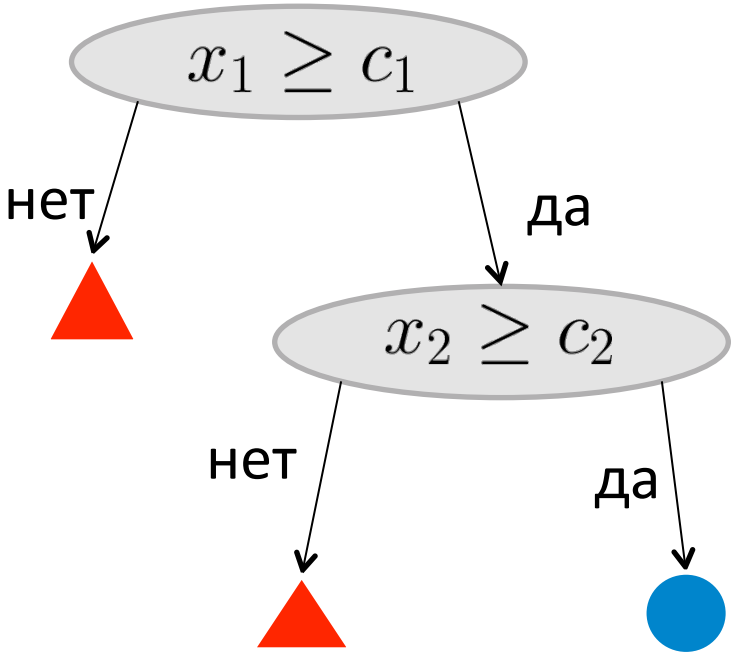
- Будем действовать жадно
- Каждый раз берем наиболее «информативное» разделение всей области

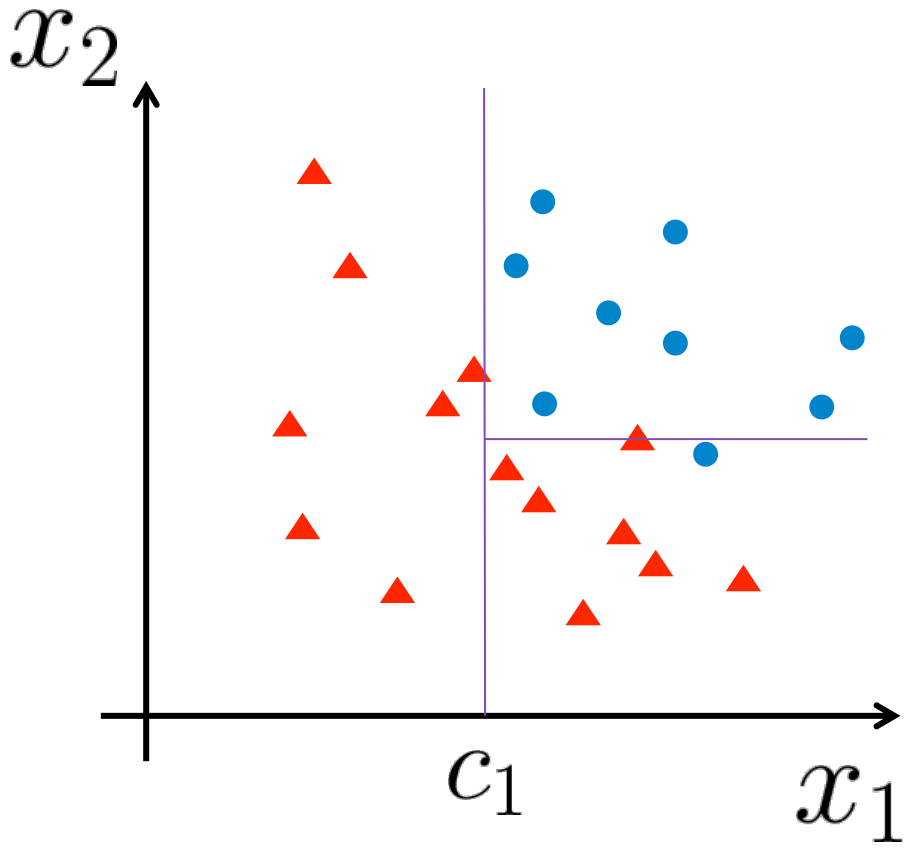




Строим дерево

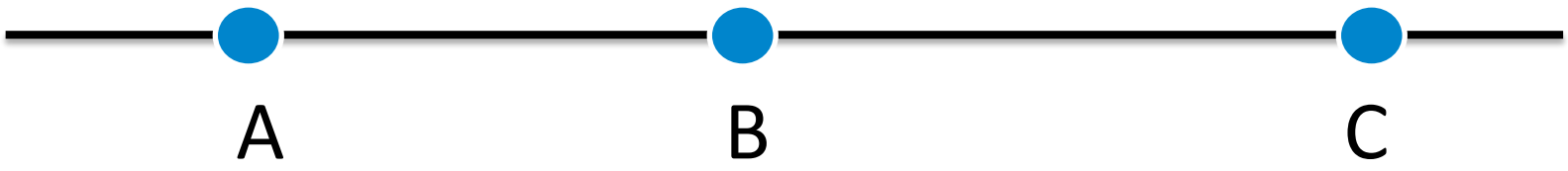
Каждый раз берем наиболее «информативное» разделение текущей области





Преимущества деревьев перед алгоритмами на метриках

- Придумать правильную меру сходства – значит почти решить задачу, это сложно. А решающие деревья не используют метрики
- Единственное что используют деревья –



точка B ближе к A, чем C по данному признаку

- Устойчивы к монотонным преобразованиям признаков

Как выбирать условия разбиения?

Перебираем признаки по очереди. Лучшее разбиение признака:

- Отделить один класс как можно сильнее
- Максимизируем число пар объектов, у которых одинаковый класс и одинаковый ответ на условие – критерий Джини
- Максимизируем число пар объектов, у которых разный класс и разные ответы на условие – критерий Донского
- Объединяем предыдущие
- Более сложные вероятностные соображения

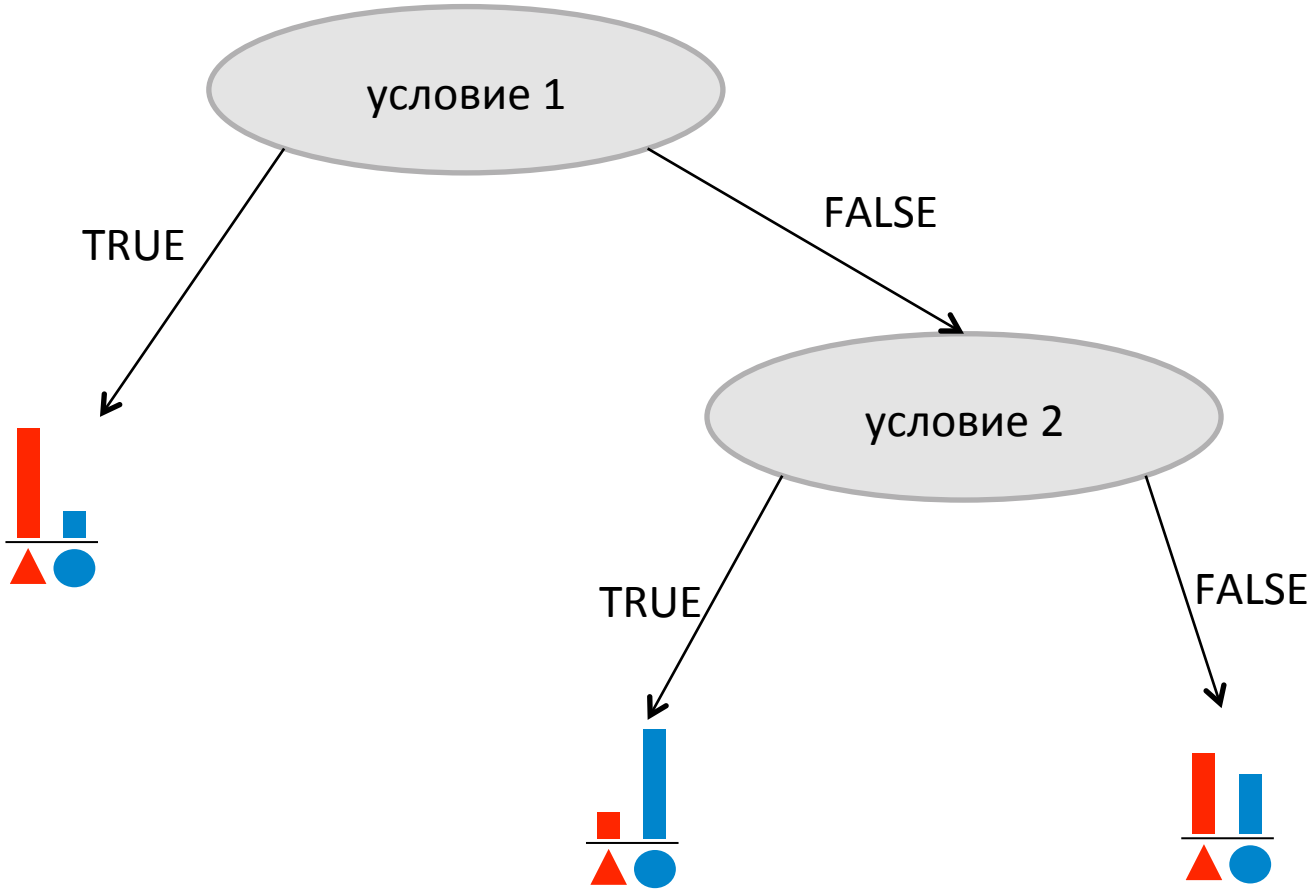
Недостатки решающего дерева

- В реальных задачах сильно переобучаются, мельчат вокруг одной области пространства (содержат в себе много параметров)
- Очень неустойчивы относительно данных

Решение: подрезания деревьев

- Если информативность условия меньше порога, то прекращаем строить дерево
- Количество объектов в листе меньше некоторого числа, то прекращаем строить
- Разбиваем обучение на две части. Пробегаем по всем поддеревьям и заменяем их левым или правым потомком, если они допускают заметно меньше ошибок

Будем возвращать вещественную степень принадлежности классу от -1 до +1



Другое решение: композиция алгоритмов

- Пусть есть какой-то набор из T алгоритмов:

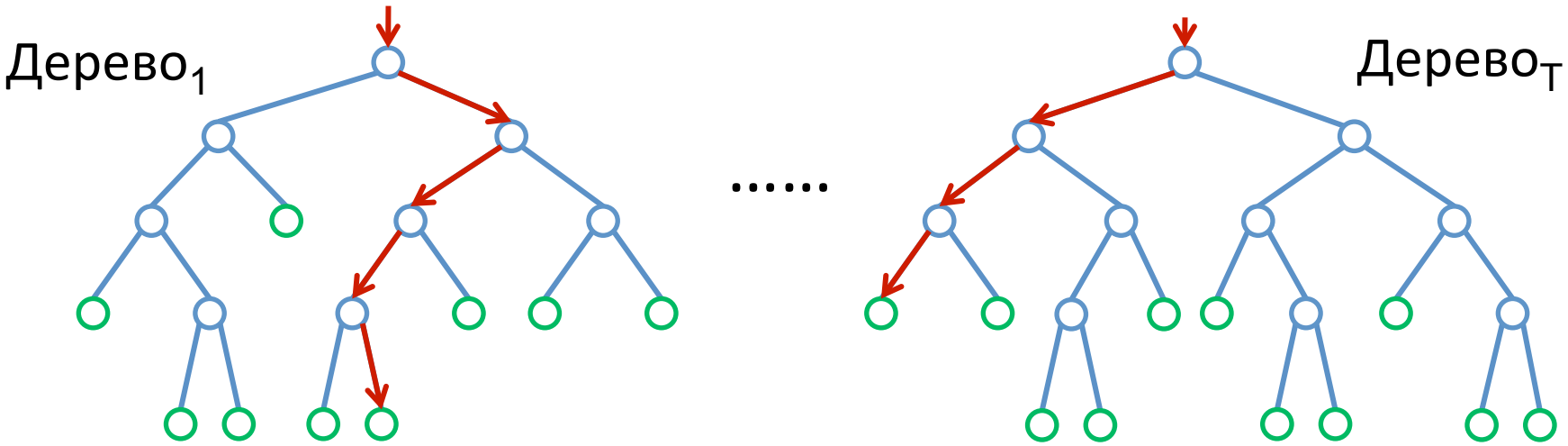
$$a_1, a_2, \ldots, a_T$$

- Финальный алгоритм:

$$result = \frac{1}{T}(a_1 + a_2 + \ldots + a_T)$$

Лес деревьев

Построим композицию из решающих деревьев

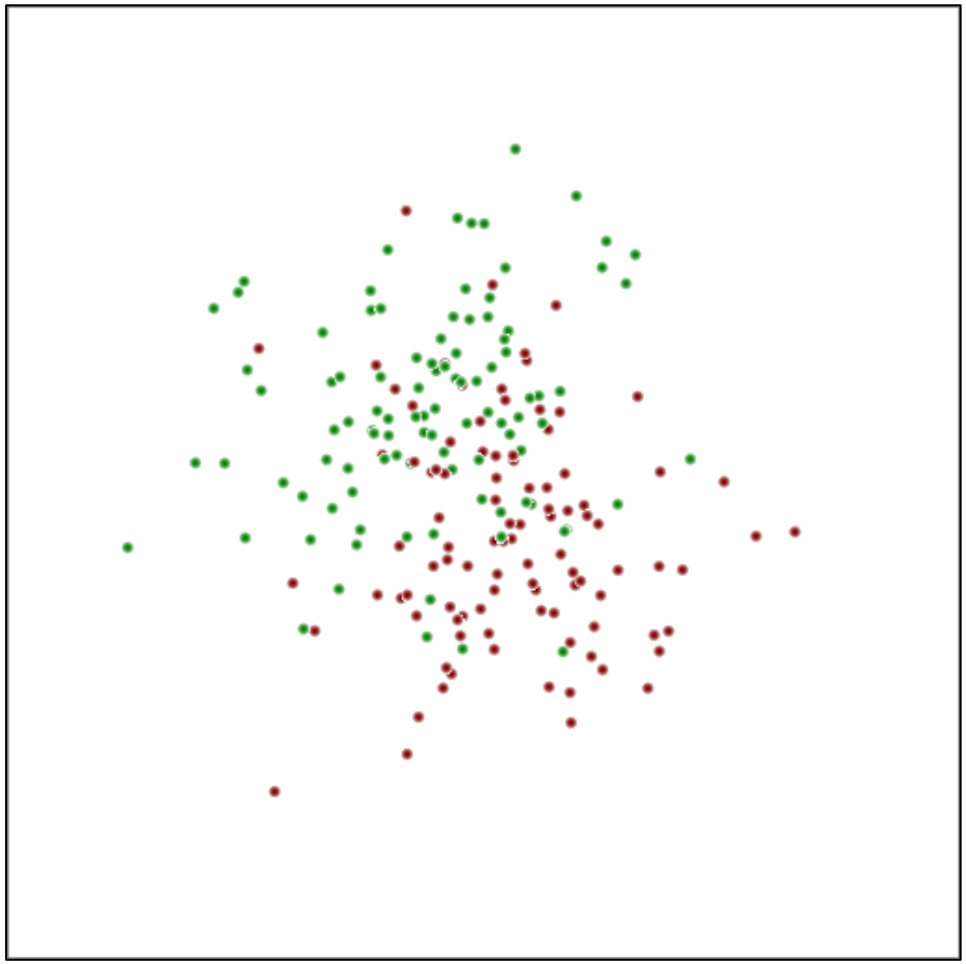


Как сделать деревья существенно разными?

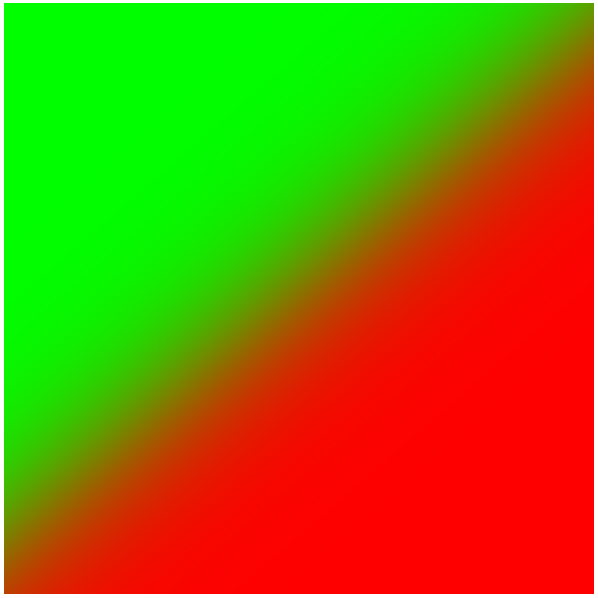
Будем обучаться на случайных подвыборках

Как работает случайный лес?

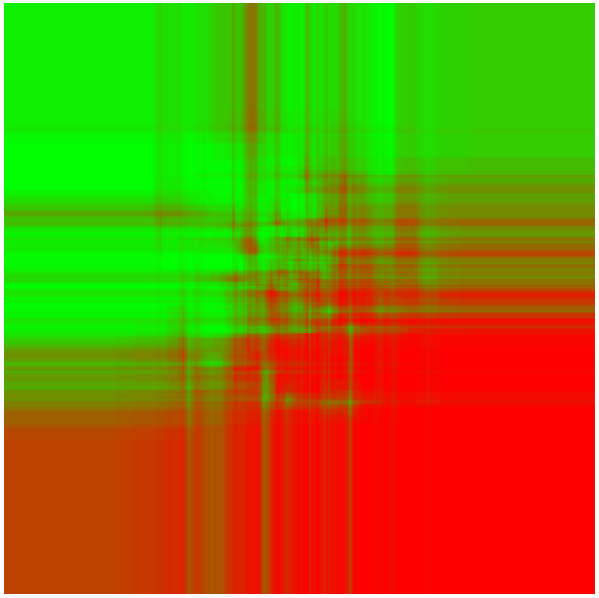
Сгенерируем данные:



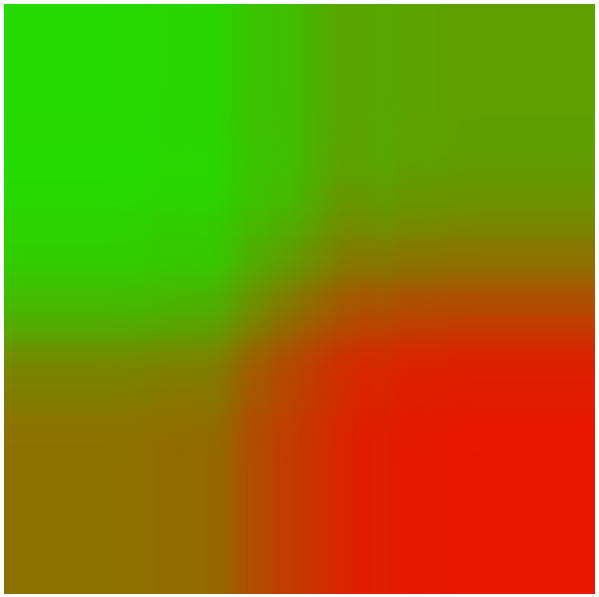
Как работает случайный лес?



Реальная оптимальная границы



Результат работы Random Forest (50 деревьев)



Результат работы Random Forest (2000 деревьев)

Особенности случайного леса

- Работает с признаками разной природы
- Не надо думать над метрикой
- Устойчив к изменениям признаков
- Хорошо распараллеливается
- Тяжело интерпретируется человеком
- Плохо приближает линейные зависимости
- Долго строится
- 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”.

Kaggle.com

Умный подбор коэффициентов

- Строчку $(a_1(x_i), \ldots, a_T(x_i))$ можно рассматривать как новое признаковое описание объекта x_i
- Что такое линейный классификатор в новом пространстве?

$$result = w_1 \cdot a_1 + w_2 \cdot a_2 + \ldots + w_T \cdot a_T$$

- Т.е. подбор коэффициентов в композиции – решение задачи линейной классификации

Последовательное наращивание композиции алгоритмов (бустинг)

- Строим композицию из слабых алгоритмов – подрезанных решающих деревьев
- Каждое новое дерево компенсирует ошибки суммы предыдущих
- Получается очень сильная композиция
- Склонна к переобучению, несмотря на многие обратные заявления :)

Основная идея бустинга

- Набор ответов, предсказанный после шага T

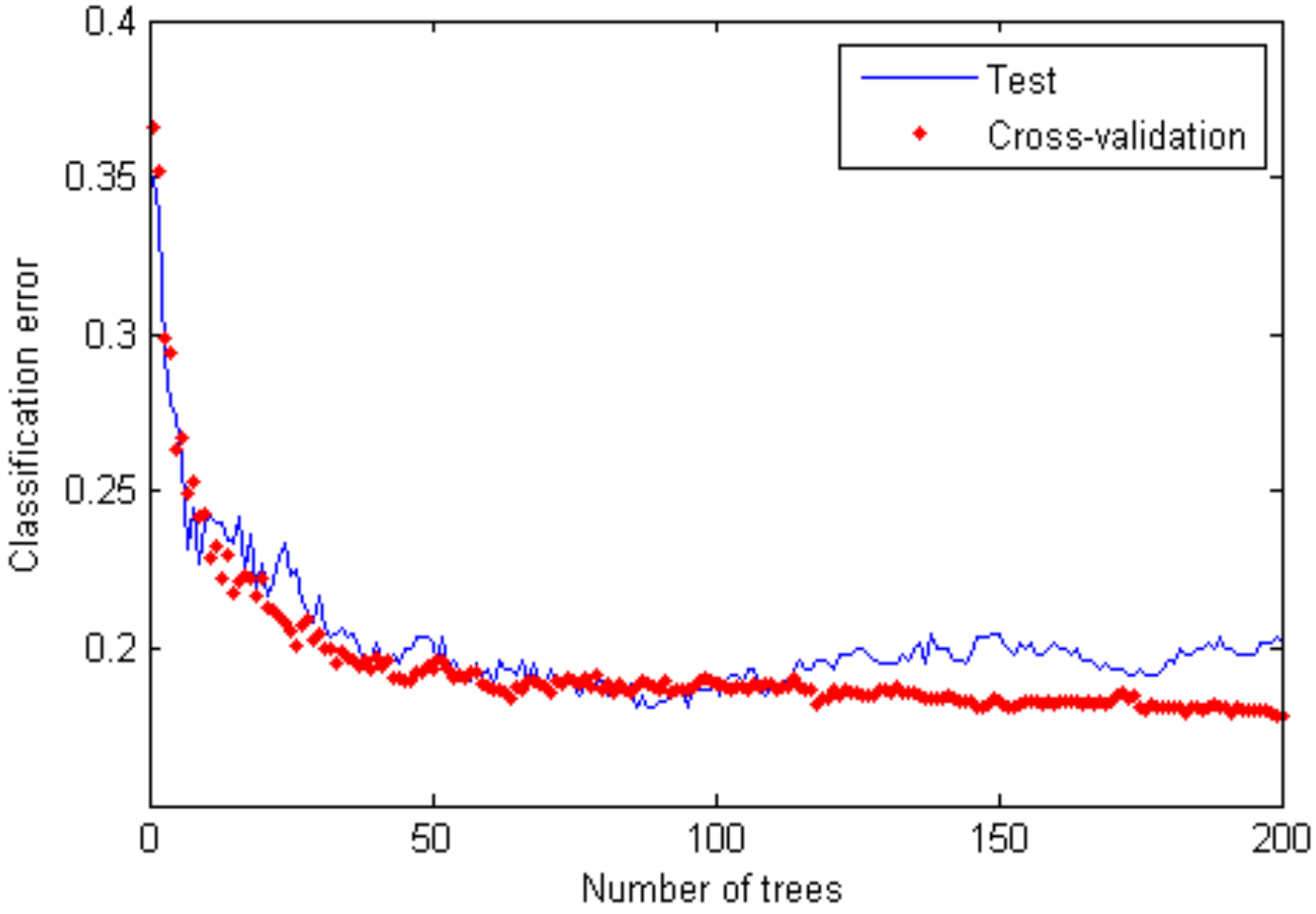
$$result_T(X) = a_1(X) + ... + a_T(X)$$

- Предсказанный ответ отличается от истинного на разность

$$Y - result_T(X)$$

- Будем обучать следующее дерево на эту разность

Качество при разном числе деревьев



Что такое бустинг на самом деле?

- Рассматривается не разность, а градиент ответов, на который настраивается очередное дерево
- Перед деревьями ставятся маленькие коэффициенты (порядка 0.02) для избежания переобучения
- На каждом шаге используется произвольная часть объектов (стохастичность)

Yandex MatrixNet – стохастический градиентный бустинг над решающими деревьями (а еще его используют Yahoo, CERN и другие)

Заключение

Про терминологию

- Интеллектуальный **анализ данных** (Data Mining)
- **Машинное обучение** (Machine Learning, Statistical Learning)
- **Прикладная статистика** (Applied Statistics)
- **Факторный анализ** (Factor Analysis)
- Теория **оптимизации** (Optimization Theory)
- **Искусственный Интеллект** (Artiffitial Intellengence)

Соревнования по анализу данных

- Сайты
  - Kaggle.com
  - Яндекс Интернет-Математика
  - и другие
- Кем проводятся
  - Компаниями
  - Работодателями
  - Университетами

Отличия от олимпиадного программирования

- Дается одна задача, а не несколько
- Решаются значительно дольше (недели, месяцы, годы)
- Не существует точного и правильного решения, проводится много экспериметнтов, чтобы понять, какое решение выбрать
- Идет борьба за сущие проценты качества
- Не важен язык, скорость работы и ресурсы; важен только результат
- В одиночку или командами

На чем пишут алгоритмы обучения?

- Готовые наборы методов машинного обучения (для общего понимания, какой метод лучше)
  - Weka
  - RapidMiner
  - Orange
- Интерпретируемые языки (для экспериментов и выбора алгоритма)
  - Matlab (Octave – бесплатная версия)
  - Python (+ scipy based библиотеки)
  - R
- Более низкоуровневые языки (для скорости работы, когда уже ясно, какой алгоритм будет использоваться)
  - C
  - C++

Примеры реальных задач

- Геологические данные – ищем золото
- Компьютерное зрение – распознавание чего-то на картинках
- Военная оборона – птица или ракета?
- Рекомендательные системы на сайтах
- Медицина – наличие болезни по симптомам
- Прогнозирование пробок
- Распознавание сигналов головного мозга
- Сайт научных статей - категоризация текстов
- Кредитный скоринг – надежность клиентов банка
- И еще очень много чего...

Что еще есть в машинном обучении?

- Многоклассовая классификация
- Регрессия (предсказываемые метки непрерывные, а не дискретные)
- Кластеризация (обучение без учителя)
- Частичное обучение
- Ассоциативные правила
- Пропуски в данных
- Структурное обучение
- Онлайн обучение
- Отбор признаков
- Ранжирование

Решения задач

- Отбор признаков
- Перебор параметров
- Задачи имеют нестандартную постановку
- Методы часто приходится придумывать самому

Литература

- Курс лекций К.В. Воронцова по машинному обучению
- Вики-ресурс, посвященный машинному обучению
- Лекции Юрия Лифшица по структурам и алгоритмам для поиска ближайших соседей (на английском, хотя лектор из Питера)
- Классические лекции по машинному обучению (на английском)
- Методическое пособие по Matlab и алгоритмам машинного обучения
- Презентация к спецкурсу "Введение в машинное обучение и анализ данных" ЛКШ 2012

Очень интересные и несложные статьи о решении реальных задач:

- Дьяконов А.Г. «Введение в анализ данных»
- Дьяконов А.Г. «Шаманство в анализе данных»
- Дьяконов А.Г. «Чему не учат в анализе данных»