Введение в анализ данных

Лекция 9

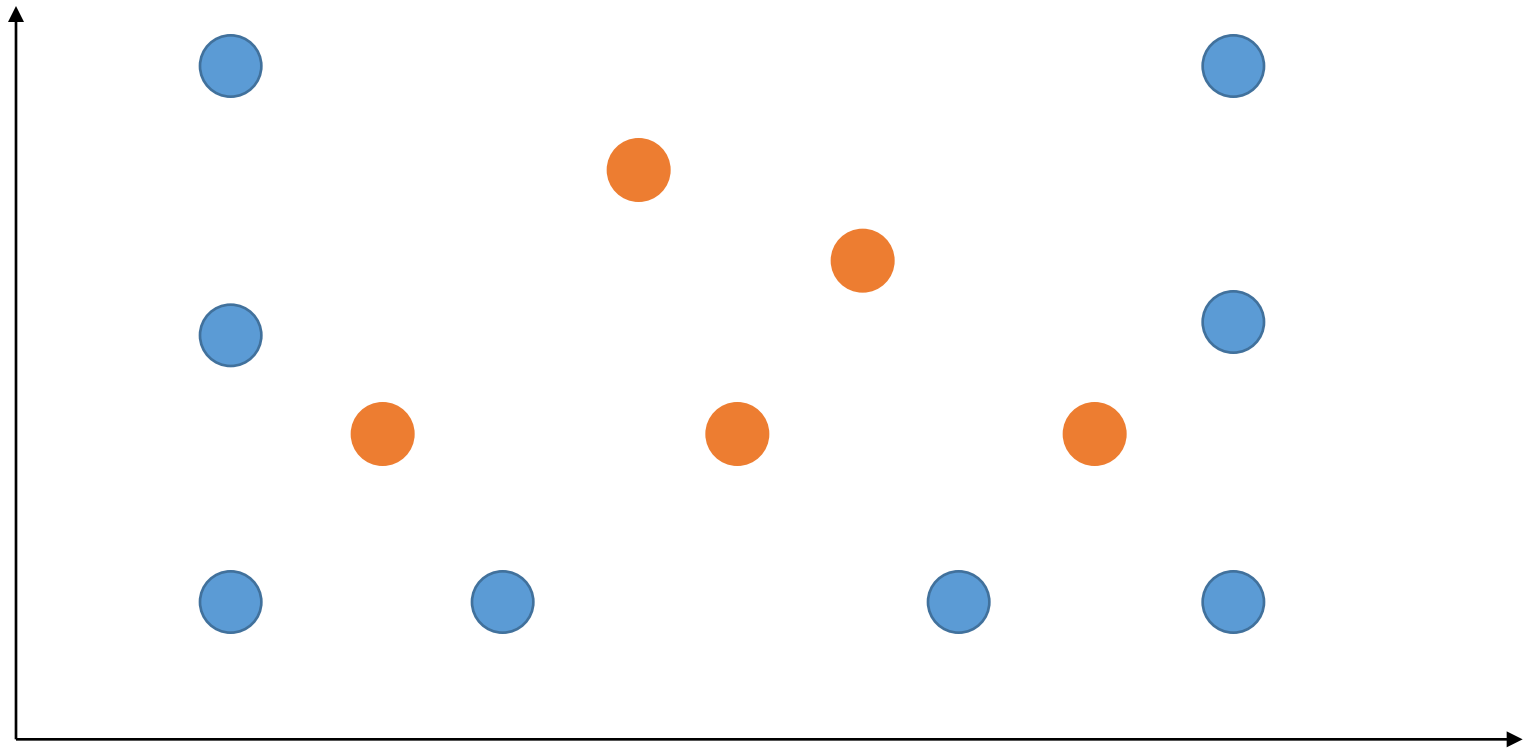
Решающие деревья и случайные леса

Евгений Соколов

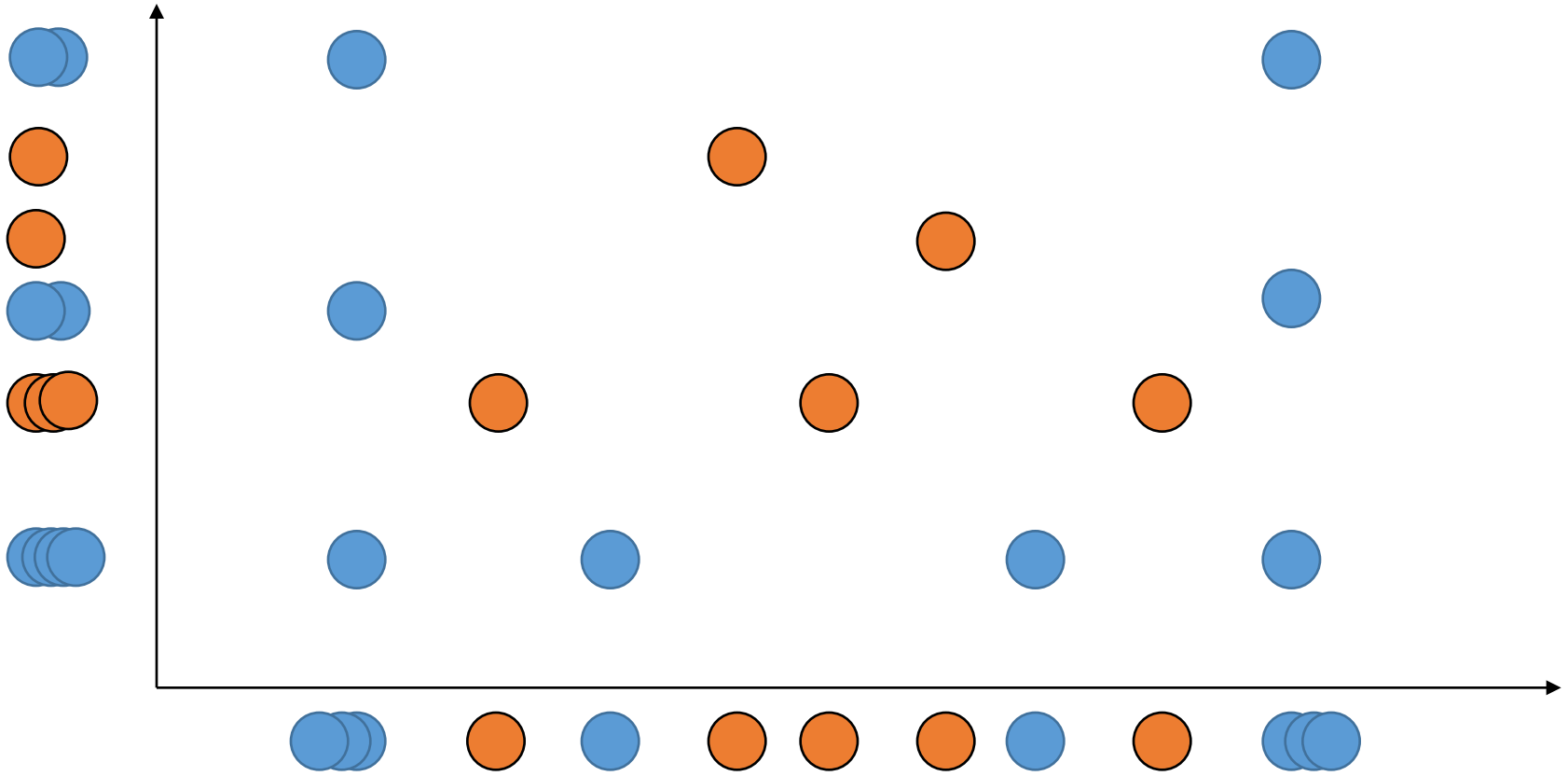
esokolov@hse.ru

НИУ ВШЭ, 2017

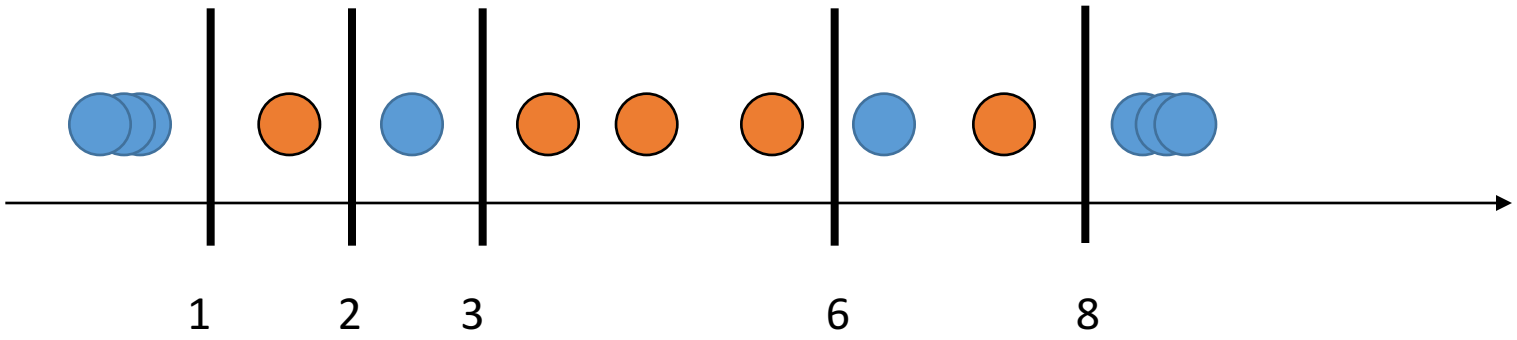
Обучение деревьев



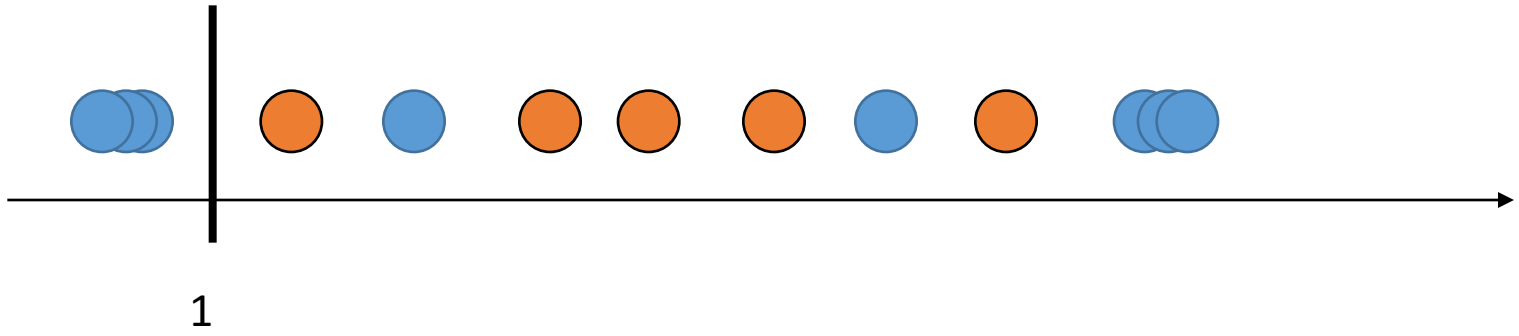
Признаки



Разбиения по признаку 1



Разбиения по признаку 1

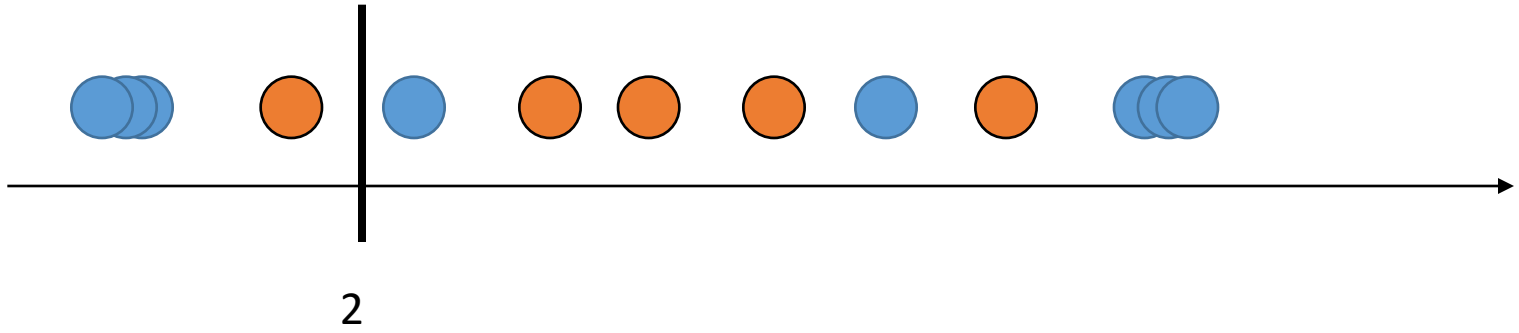


(1, 0)
H(p) = 0

(1/2, 1/2)
H(p) = 0.69

$$\frac{3}{13}H(p_l) + \frac{10}{13}H(p_r) = 0.53$$

Разбиения по признаку 1

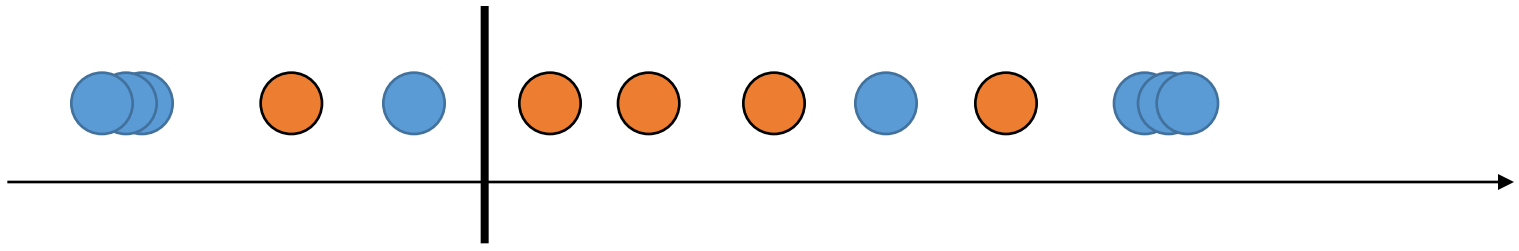


(3/4, 1/4)
H(p) = 0.56

(5/9, 4/9)
H(p) = 0.69

$$\frac{4}{13}H(p_l) + \frac{9}{13}H(p_r) = 0.65$$

Разбиения по признаку 1

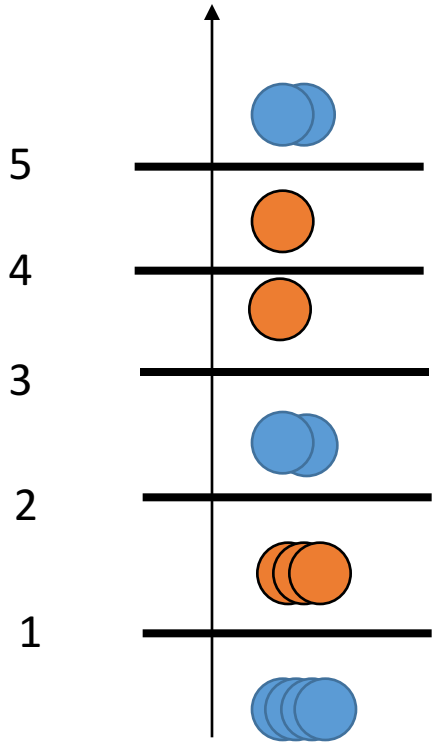


(4/5, 1/5)
H(p) = 0.5

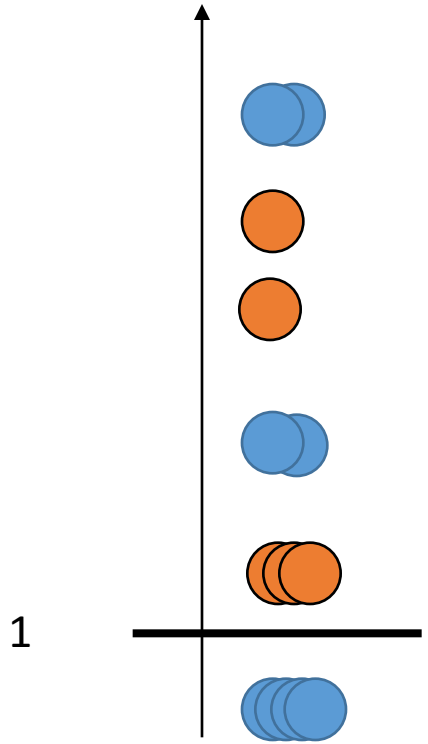
(1/2, 1/2)
H(p) = 0.69

$$\frac{5}{13}H(p_l) + \frac{8}{13}H(p_r) = 0.62$$

Разбиения по признаку 2



Разбиения по признаку 2

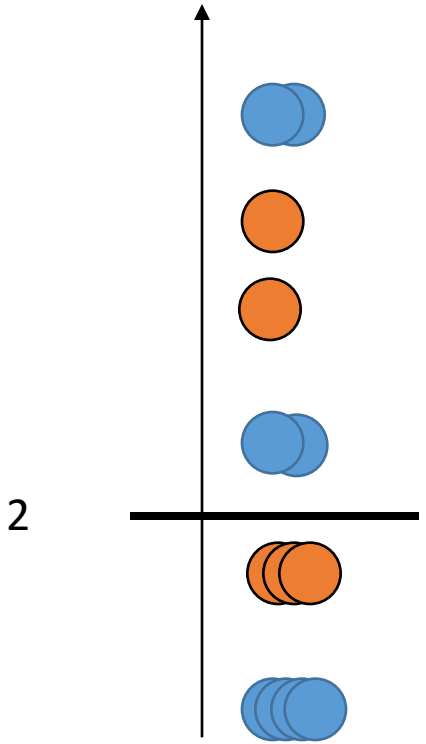


(4/9, 5/9)
H(p) = 0.69

(1, 0)
H(p) = 0

$$\frac{4}{13}H(p_l) + \frac{9}{13}H(p_r) = 0.47$$

Разбиения по признаку 2

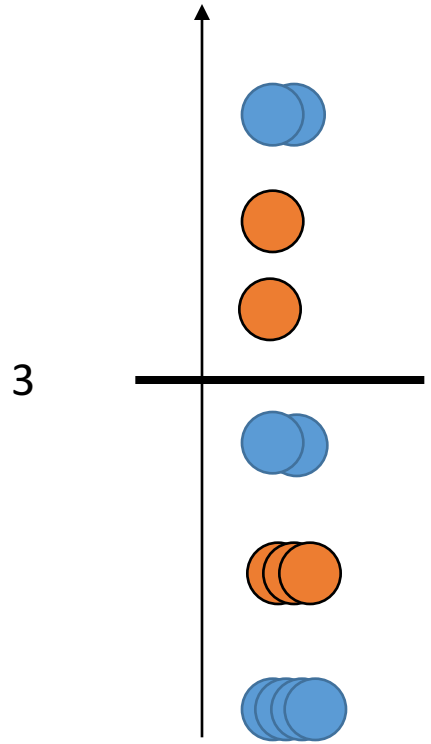


(4/6, 2/6)
H(p) = 0.64

(4/7, 3/7)
H(p) = 0.68

$$\frac{7}{13}H(p_l) + \frac{6}{13}H(p_r) = 0.66$$

Разбиения по признаку 2

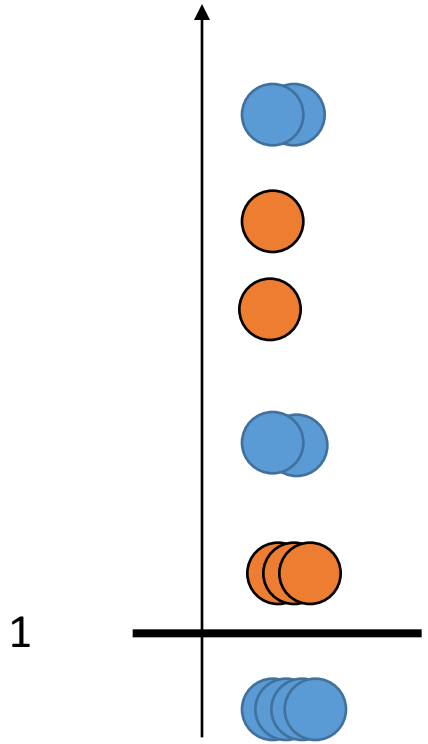


(1/2, 1/2)
H(p) = 0.69

(6/9, 3/9)
H(p) = 0.46

$$\frac{9}{13}H(p_l) + \frac{4}{13}H(p_r) = 0.53$$

Разбиения по признаку 2



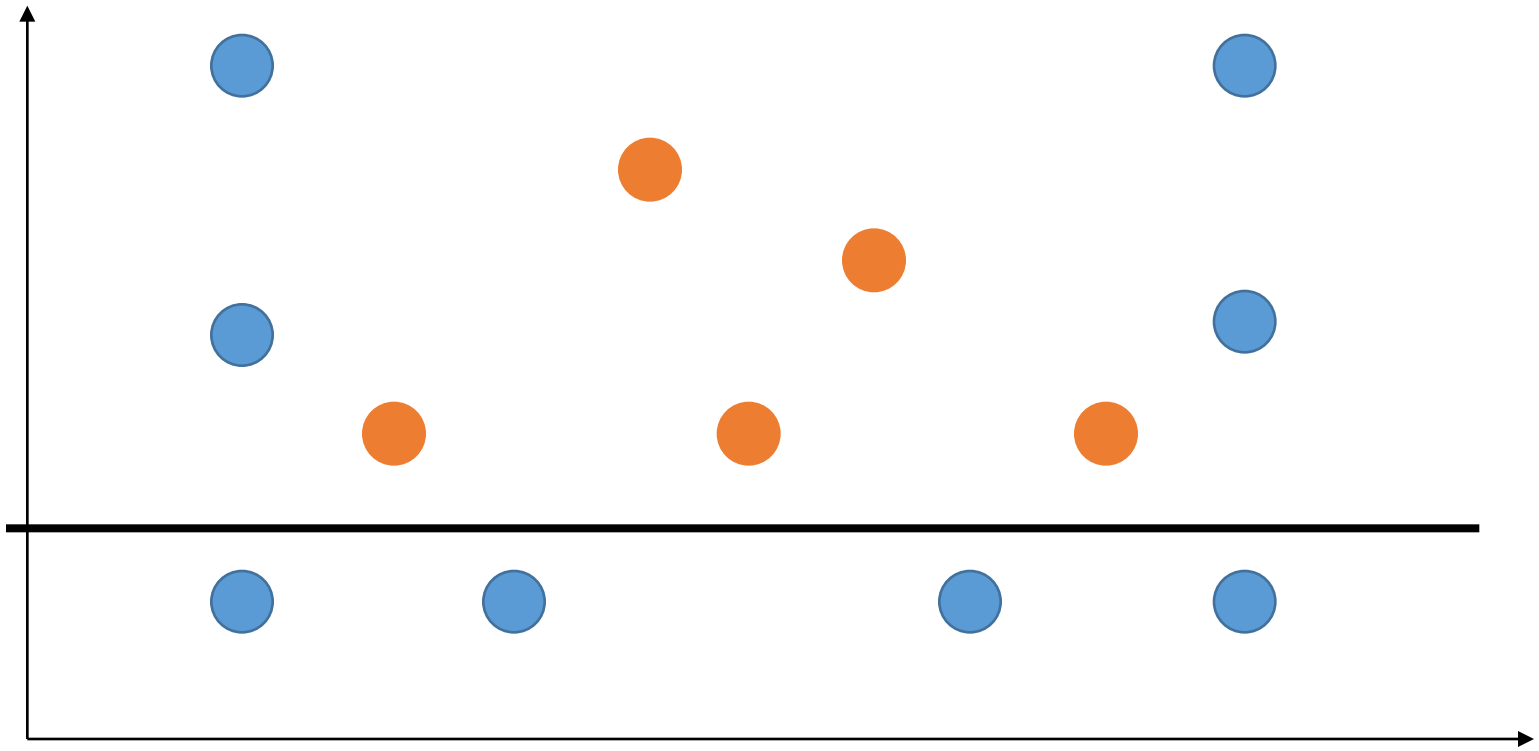
(4/9, 5/9)
H(p) = 0.69

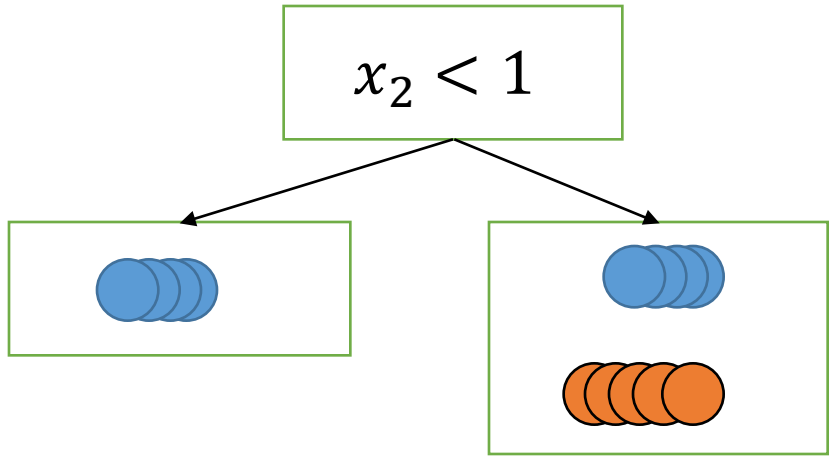
(1, 0)
H(p) = 0

$$\frac{4}{13}H(p_l) + \frac{9}{13}H(p_r) = 0.47$$

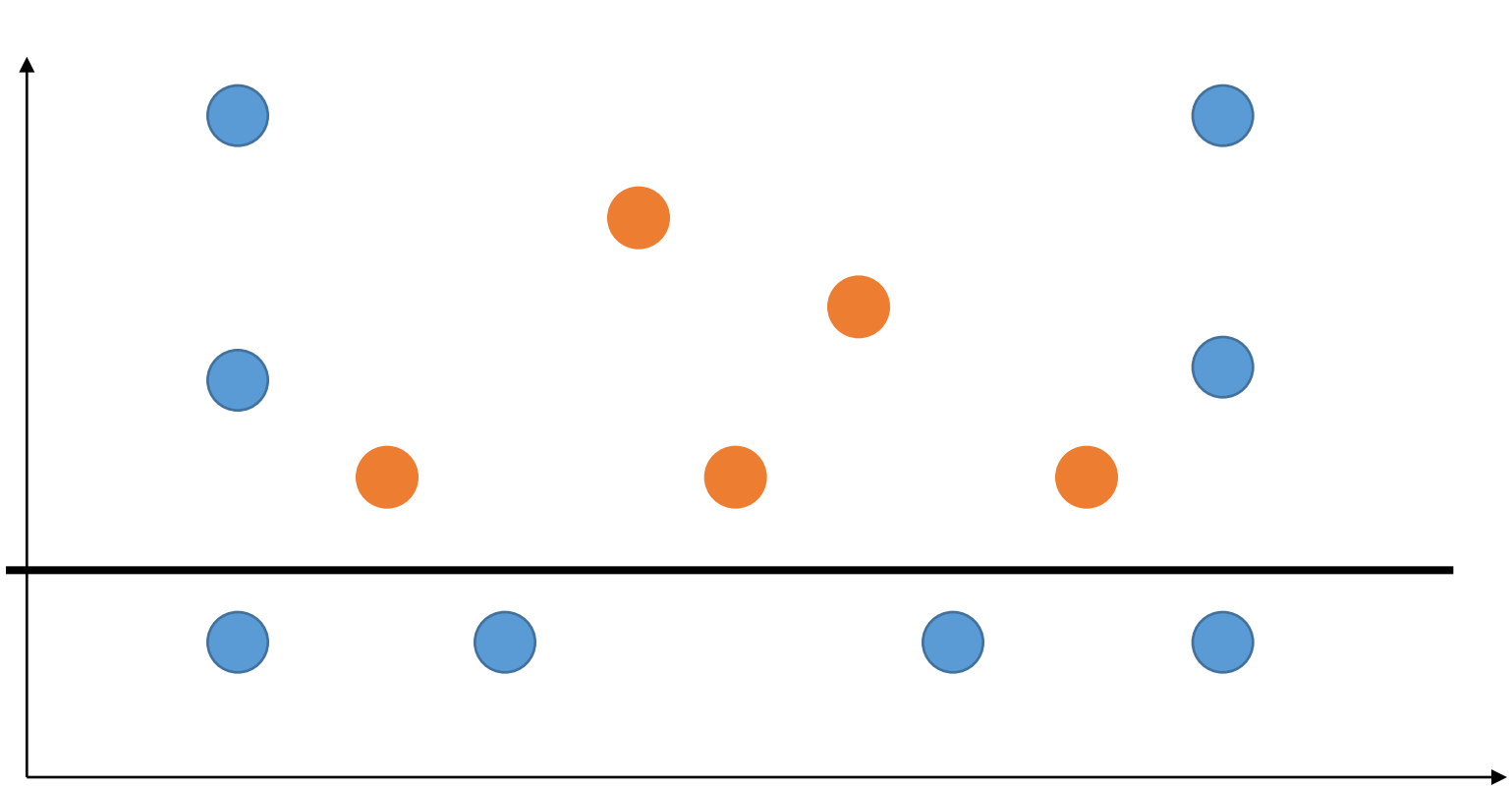
Лучшее разбиение!

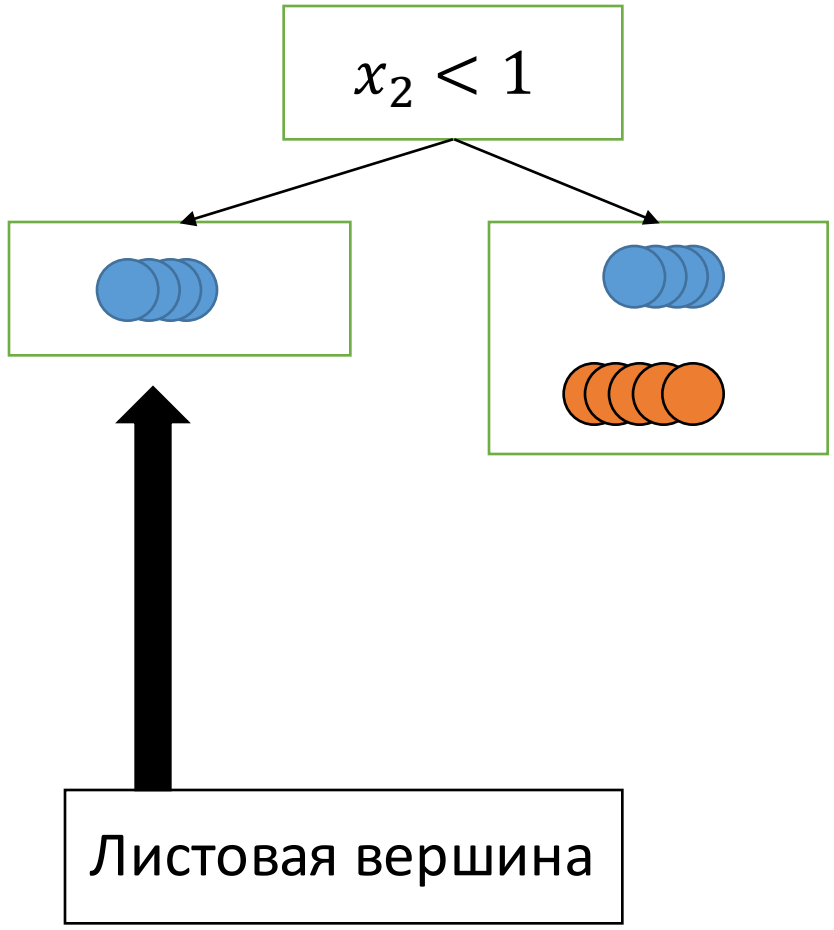
Обучение деревьев



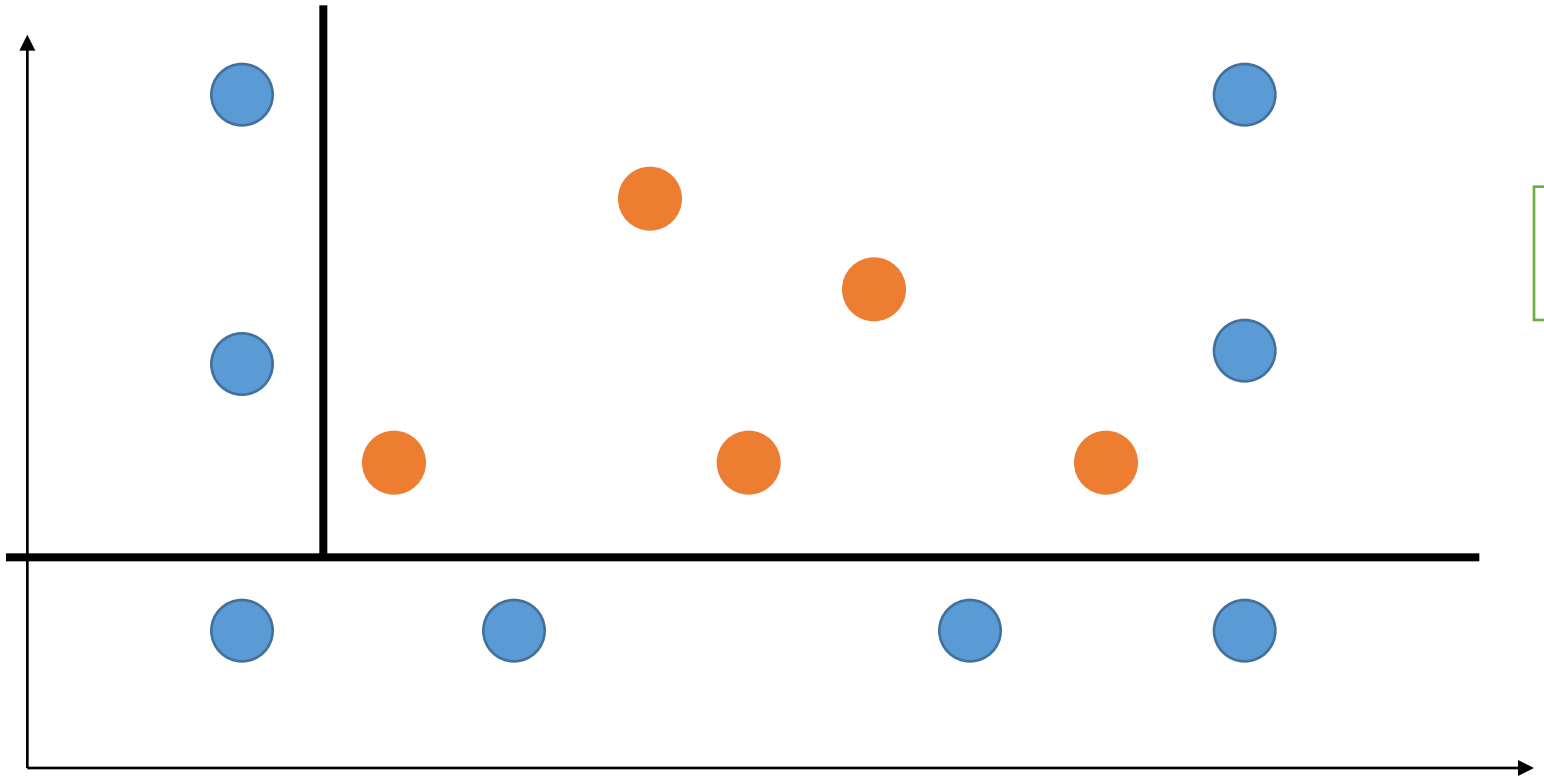


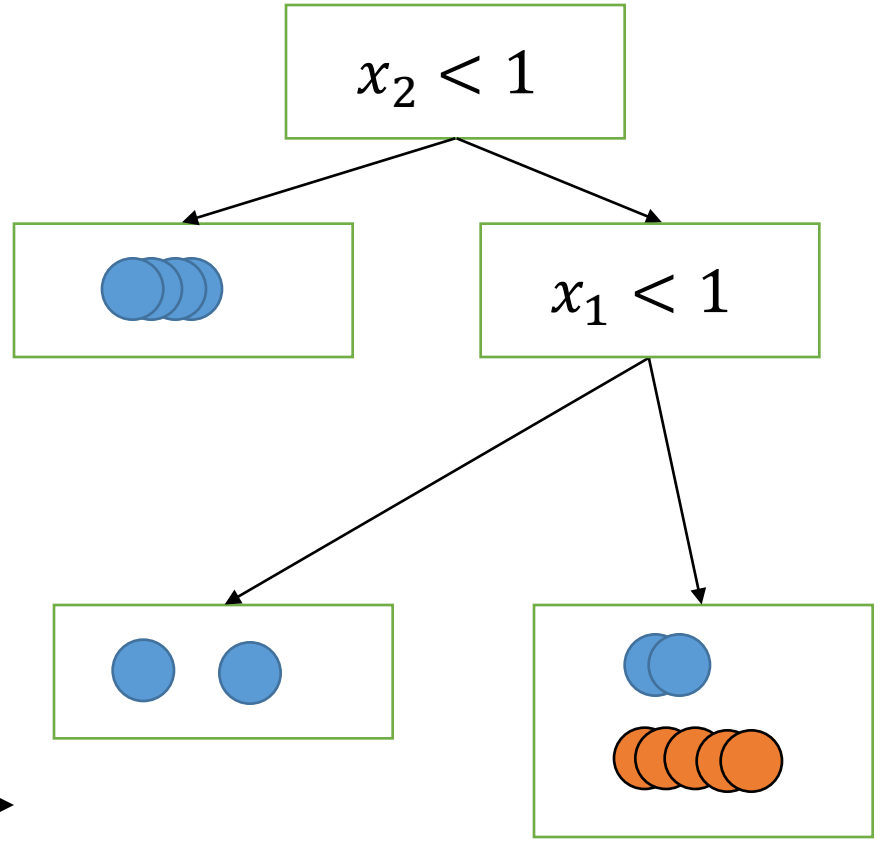
Обучение деревьев



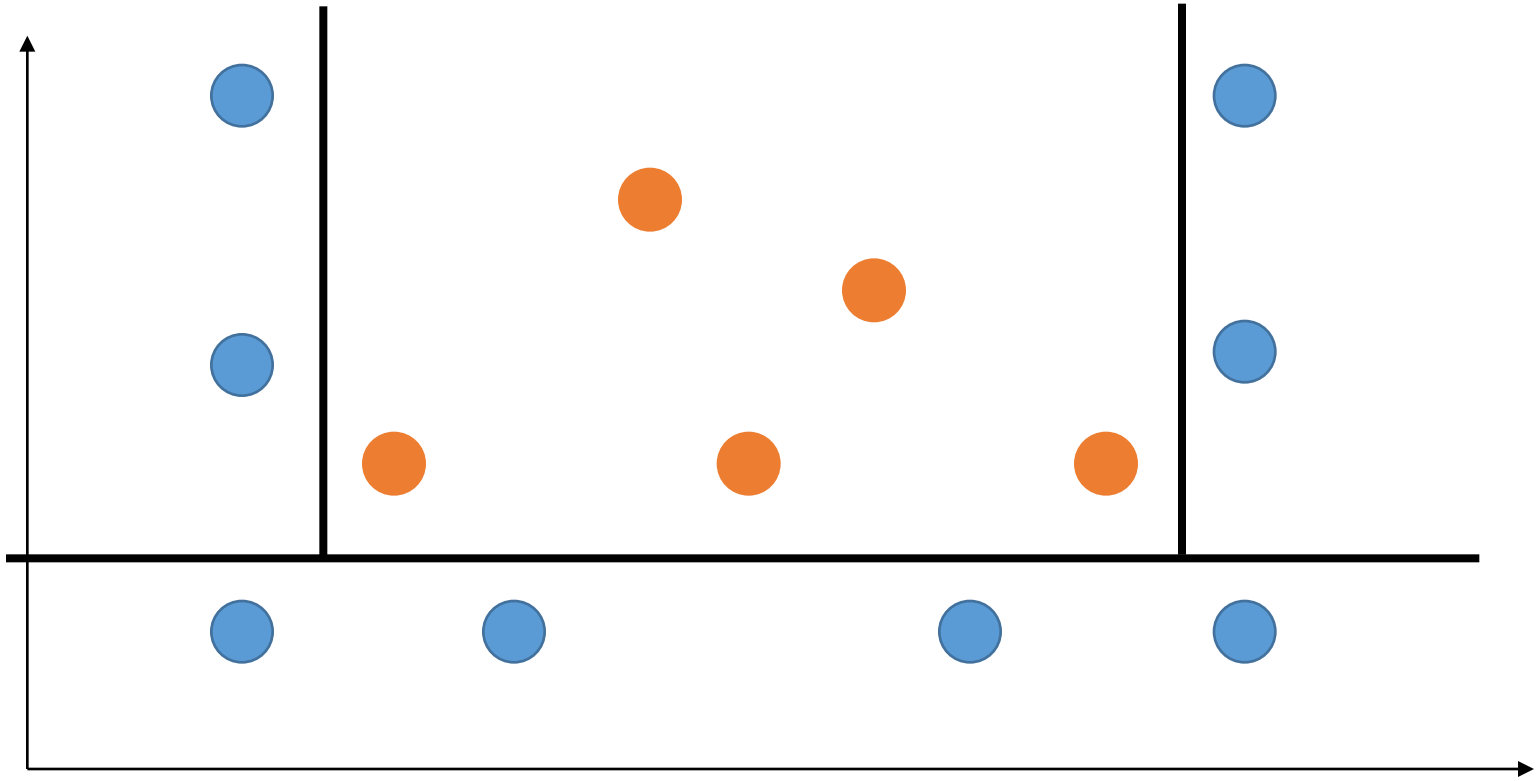


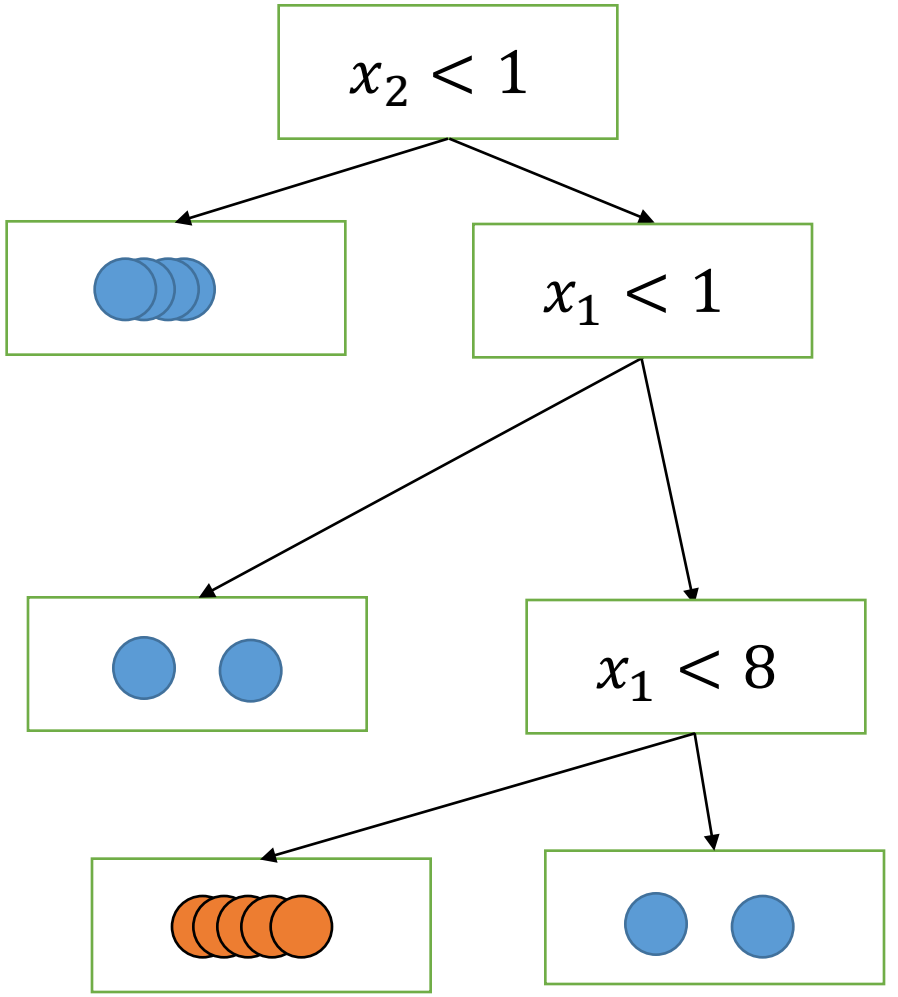
Обучение деревьев





Обучение деревьев

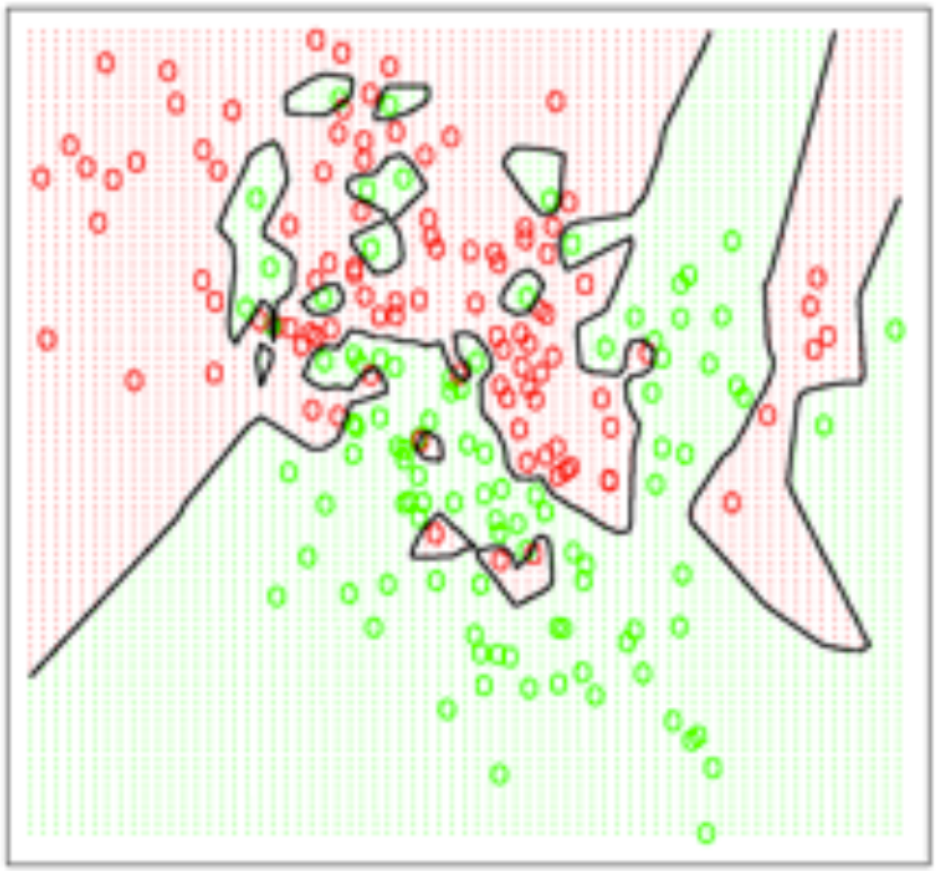


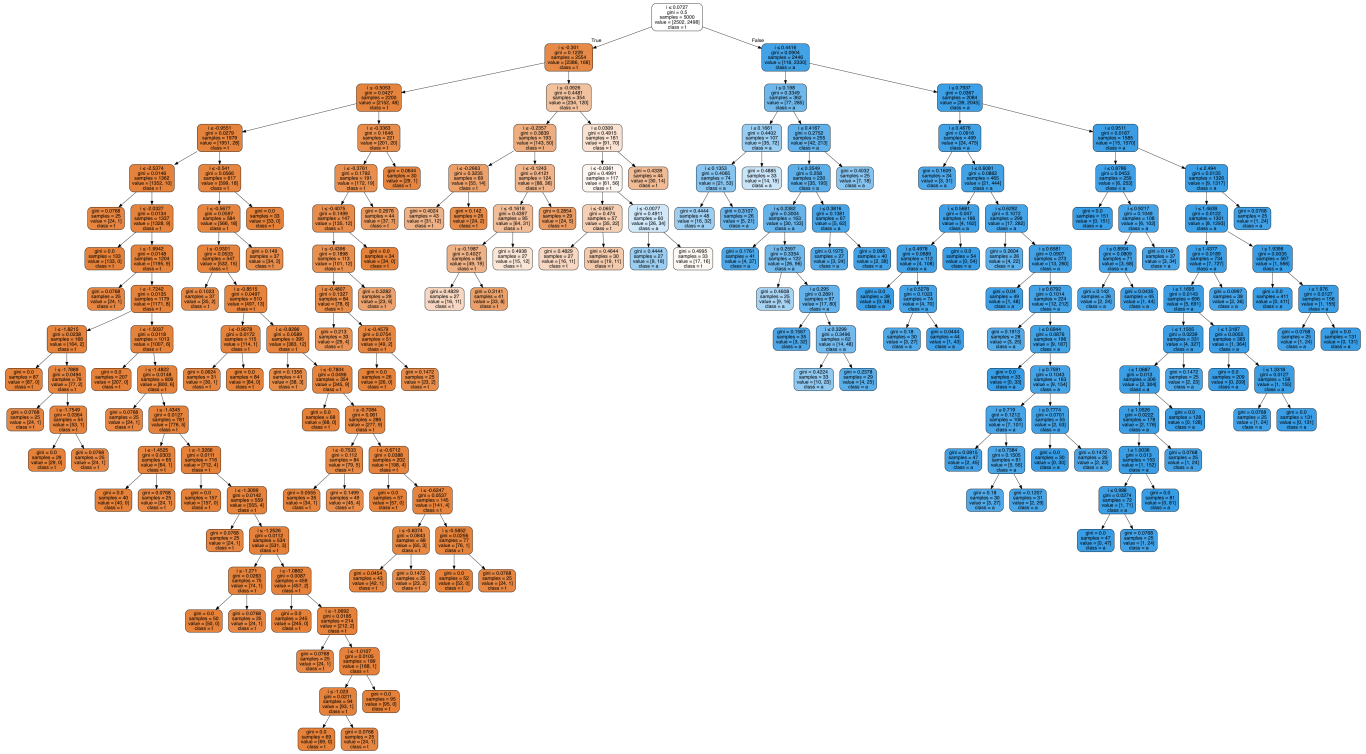


Жадный алгоритм построения дерева

1. Поместить в корень всю выборку: $X_1 = X$
2. Начать построение с корня: $m = 1$
3. Если выполнен критерий останова для вершины m, то выход
4. Найти лучшее разбиение $[x^j \leq t]$ для вершины m
5. Разбить вершину m на дочерние вершины l и r
6. Повторить шаги 3-6 для дочерних вершин l и r

Переобучение деревьев





Переобучение деревьев

- Дерево может достичь нулевой ошибки на любой выборке
- Как правило, такое дерево окажется переобученным
- Выход — ограничивать глубину или число объектов в листе

Композиции деревьев

Majority vote

- Дано: N базовых алгоритмов $b_1(x), \ldots, b_N(x)$
- Каждый хотя бы немного лучше случайного угадывания
- Композиция: класс, за который проголосовало больше всего базовых алгоритмов

$$a(x) = \arg\max_{y \in \mathbb{Y}} \sum_{n=1}^{N} [b_n(x) = y]$$

Усреднение наблюдений

- Дано: N базовых алгоритмов $b_1(x), \ldots, b_N(x)$
- Каждый хотя бы немного лучше случайного угадывания
- Композиция:

$$a(x) = \frac{1}{N} \sum_{n=1}^{N} b_n(x)$$

Композиции алгоритмов

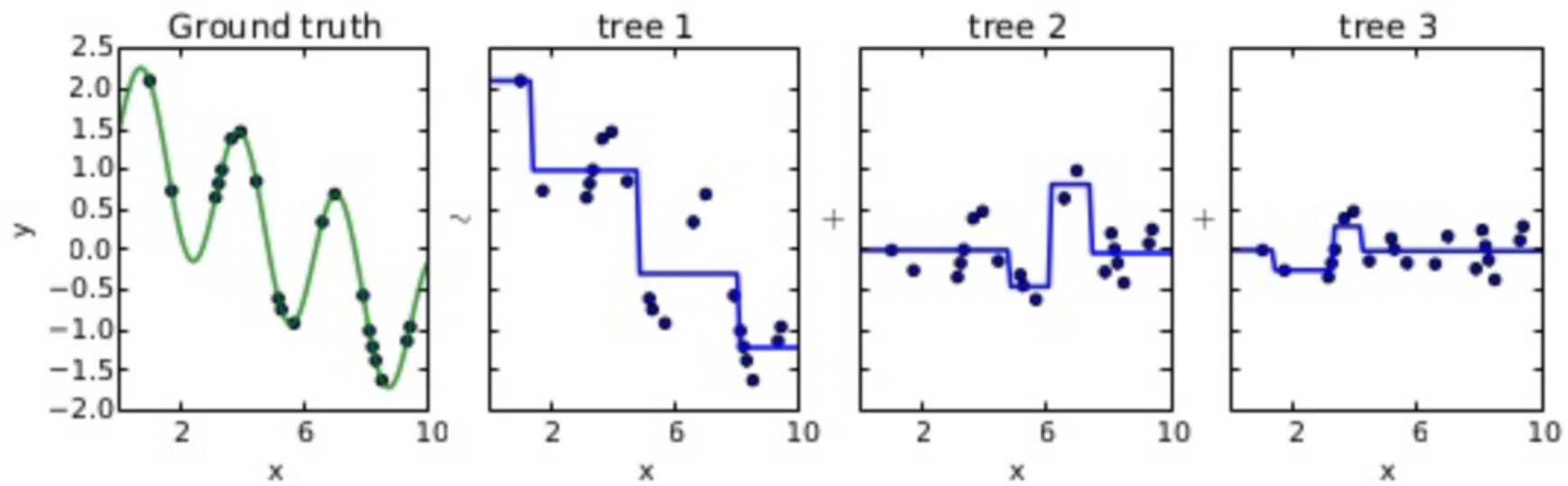
- Базовые алгоритмы: $b_1(x), \ldots, b_N(x)$
- Композиция: $a(x)$

- Как по одной и той же выборке обучить N различных моделей?

Бустинг

- Каждый следующий алгоритм исправляет ошибки предыдущих

- Яркий пример: градиентный бустинг над решающими деревьями
- В следующем курсе



Бэггинг

- Bagging (Bootstrap Aggregation)
- Базовые алгоритмы обучаются независимо
- Каждый обучается на подмножестве данных
- Усреднение ответов или выбор по большинству

- Яркий пример: случайный лес (random forest)

Бэггинг

Идея:

- Обучим много деревьев $b_1(x), \ldots, b_N(x)$
- Выберем ответ по большинству:

$$a(x) = \arg\max_{y \in \mathbb{Y}} \sum_{n=1}^{N} [b_n(x) = y]$$

Пример

- Прогнозы деревьев: $-1, -1, 1, -1, 1, -1$

$$a(x) = -1$$

Рандомизация

- Как сделать деревья разными?
- Обучать по подвыборкам!

Рандомизация

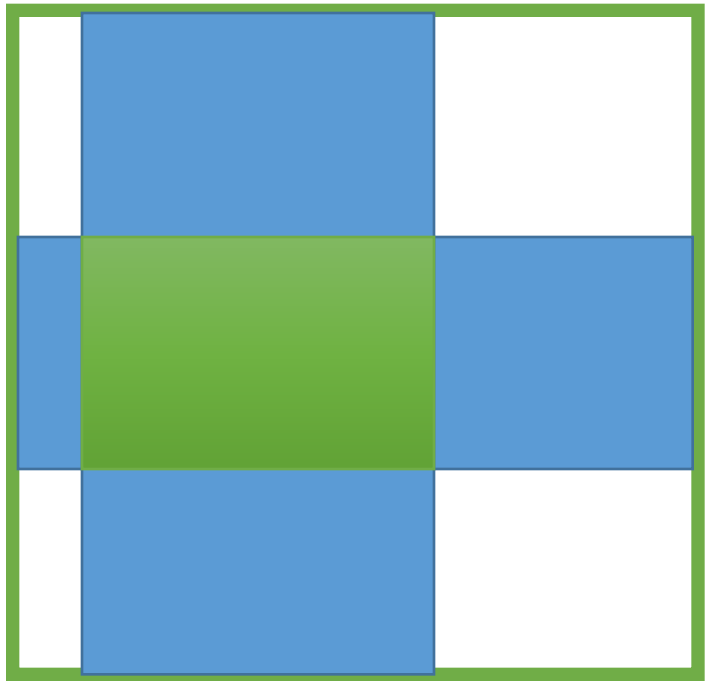
- Популярный подход: бутстрап
- Выбираем из обучающей выборки ℓ объектов с возвращением
- Пример: $\{x_1, x_2, x_3, x_4\} \to \{x_1, x_2, x_2, x_4\}$
- Примерно $0.632 * \ell$ различных объектов

Рандомизация

- Другой подход: выбор случайного подмножества объектов
- Гиперпараметр: размер подмножества

Виды рандомизации

- Бэггинг: обучаем на случайной подвыборке
- Метод случайных подпространств: обучаем на случайном подмножестве признаков
- Размер подвыборки/подмножества — гиперпараметр



Рандомизация

- Этого недостаточно
- Как можно рандомизировать сам процесс построения дерева?

Поиск разбиения

- Пусть в вершине m оказалась выборка X_m
- $Q(X_m, j, t)$ — критерий ошибки условия $[x^j \leq t]$
- Ищем лучшие параметры j и t перебором:

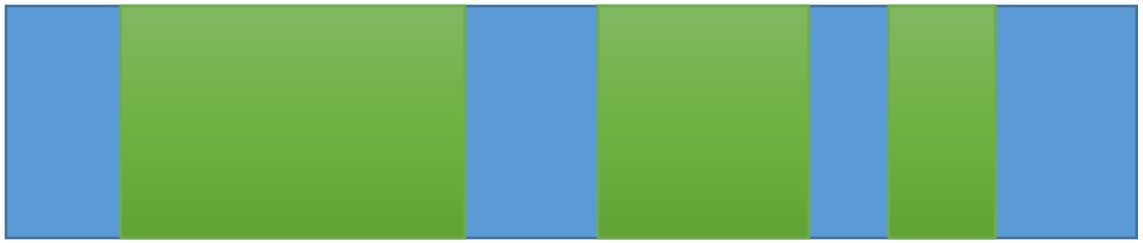
$$Q(X_m, j, t) \to \min_{j,t}$$

Поиск разбиения

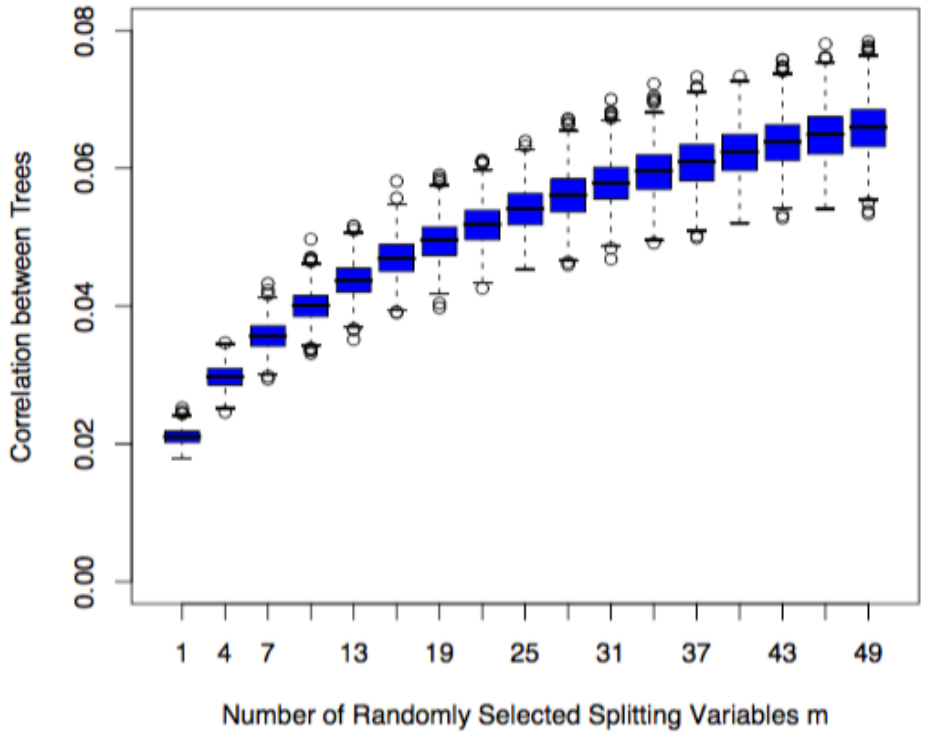
- Пусть в вершине m оказалась выборка X_m
- $Q(X_m, j, t)$ — критерий ошибки условия $[x^j \leq t]$
- Ищем лучшие параметры j и t перебором:

$$Q(X_m, j, t) \to \min_{j,t}$$

- Случайный лес: выбираем j из случайного подмножества признаков размера q



Корреляция между деревьями



Корреляция между деревьями

Рекомендации для q:

- Регрессия: $q = \frac{d}{3}$
- Классификация: $q = \sqrt{d}$

Случайный лес (Random forest)

1. Для $n = 1, \dots, N$:
2. Сгенерировать выборку $\tilde{X}$ с помощью бутстрапа
3. Построить решающее дерево $b_n(x)$ по выборке $\tilde{X}$
4. Дерево строится, пока в каждом листе не окажется не более n_{min} объектов
5. Оптимальное разбиение ищется среди q случайных признаков

Случайный лес (Random forest)

1. Для $n = 1, \dots, N$:
2. Сгенерировать выборку $\tilde{X}$ с помощью бутстрапа
3. Построить решающее дерево $b_n(x)$ по выборке $\tilde{X}$
4. Дерево строится, пока в каждом листе не окажется не более n_{min} объектов
5. Оптимальное разбиение ищется среди q случайных признаков

Выбираются заново при каждом разбиении!

Случайный лес

- Регрессия:

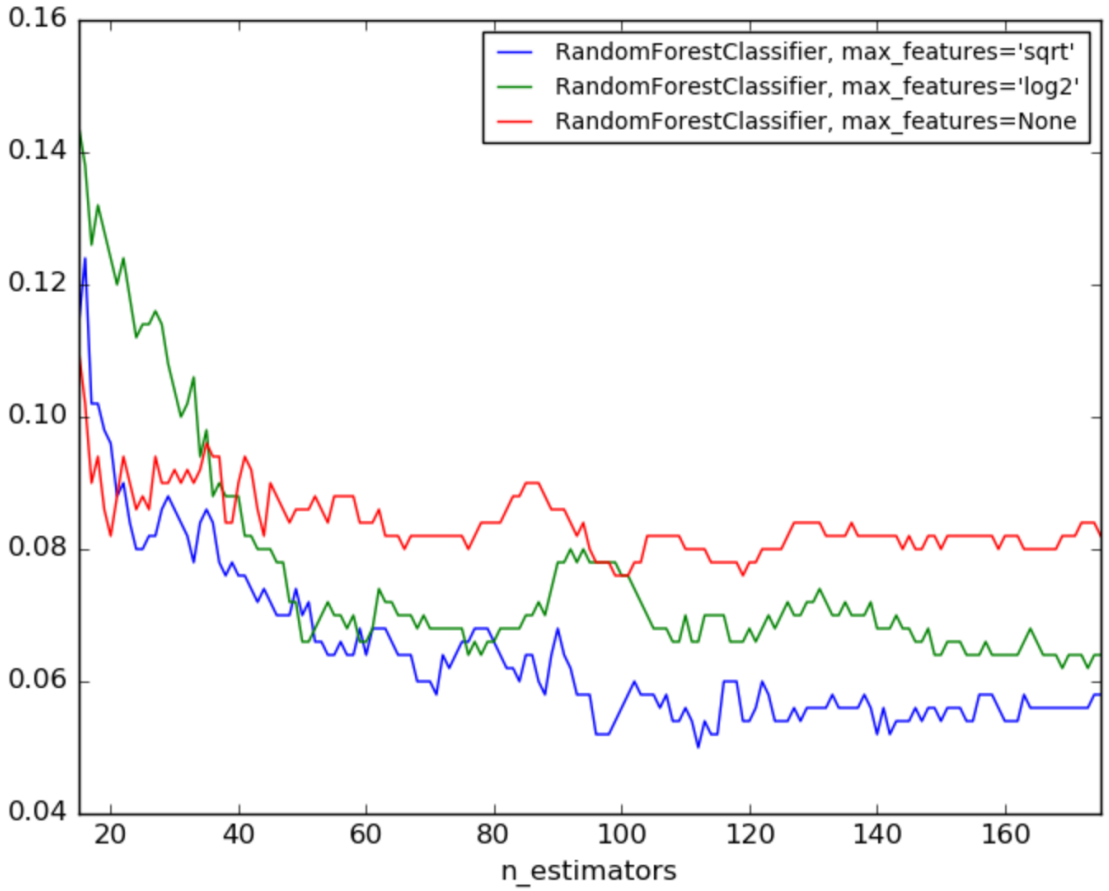
$$a(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x)$$

- Классификация:

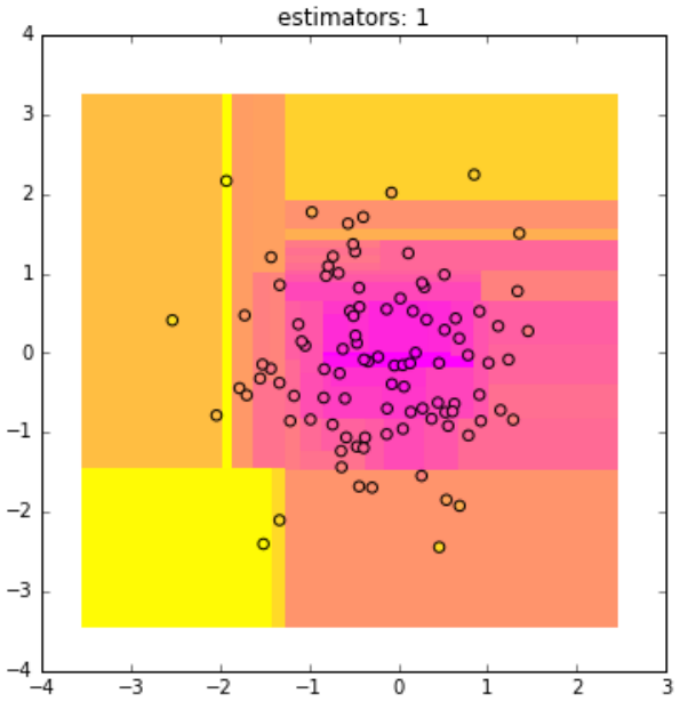
$$a(x) = \arg\max_{y \in \mathbb{Y}} \sum_{n=1}^{N} [b_n(x) = y]$$

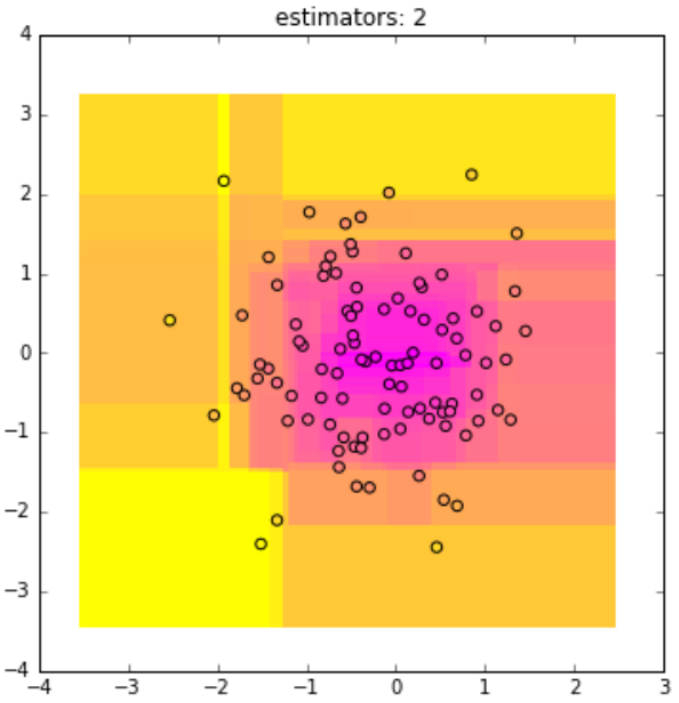
Ошибка на тесте

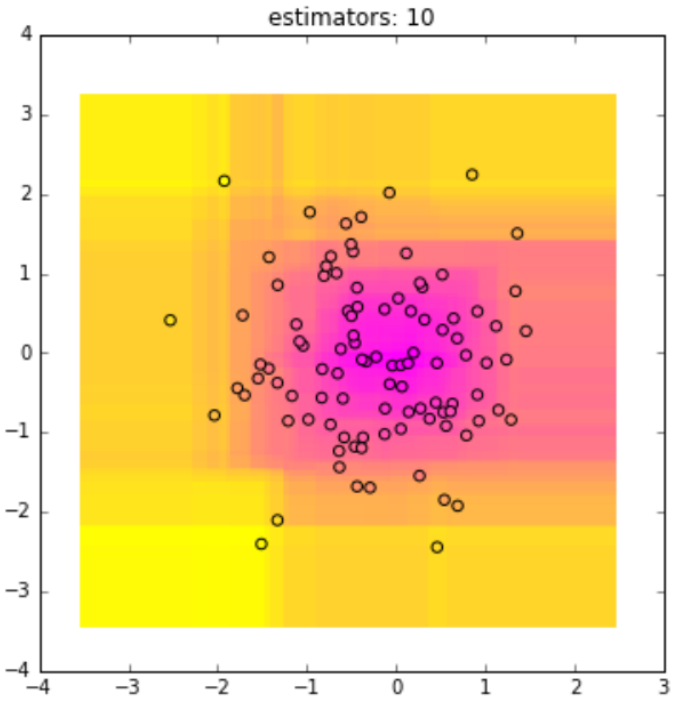
- Ошибка сначала убывает, а затем остаётся примерно на одном уровне
- Случайный лес не переобучается при росте N



Случайный лес







Резюме

- Деревья обучаются жадно
- Разбиения выбираются так, чтобы как можно сильнее уменьшить критерий информативности
- Борьба с переобучением: ограничение глубины или числа объектов в листьях
- Композиции алгоритмов
- Случайные леса

Обучение с учителем (заключение)

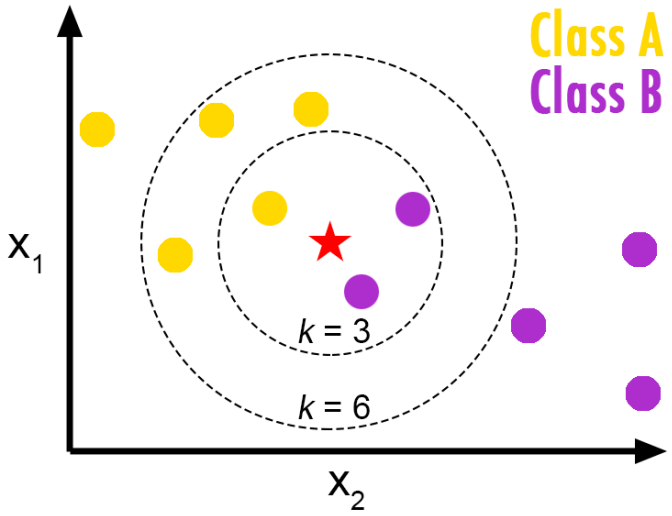
Обучение с учителем

- В нашем курсе: классификация или регрессия
- $X = (x_i, y_i)_{i=1}^{\ell}$ — обучающая выборка
- $a(x)$ — алгоритм, модель

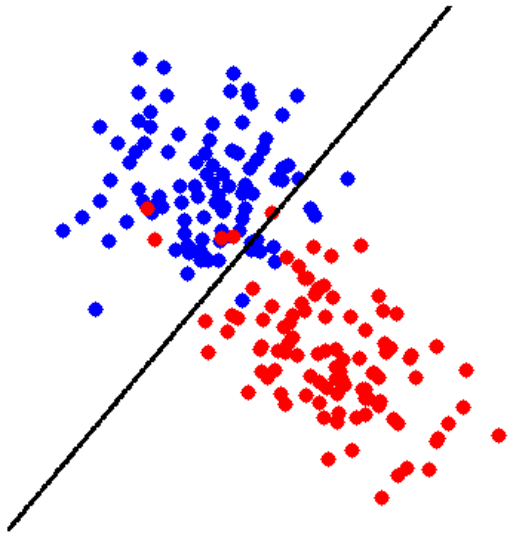
- «С учителем» — т.е. на обучающей выборке известны ответы y_i

Обучение с учителем

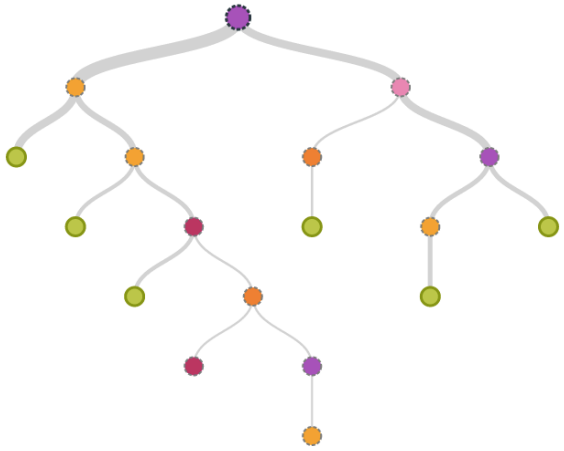
Метод k ближайших соседей



Линейные модели

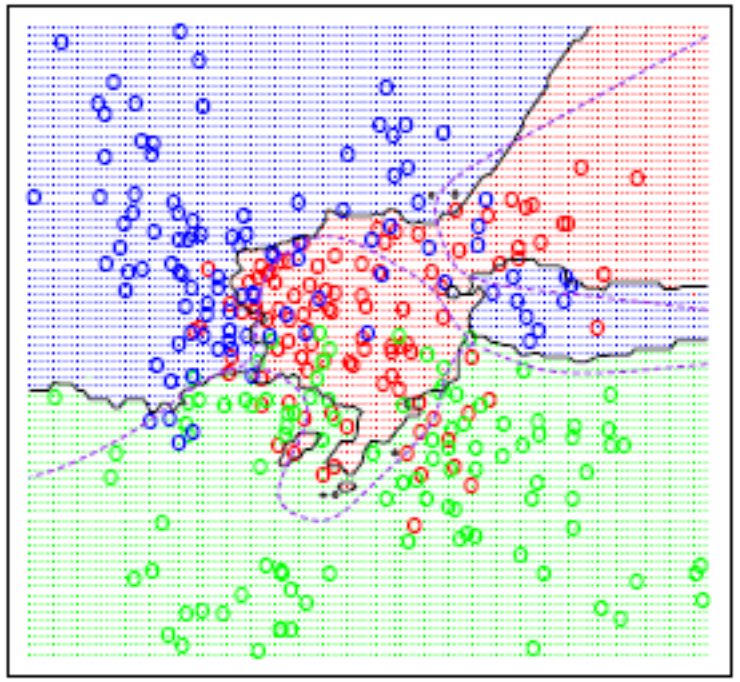


Решающие деревья



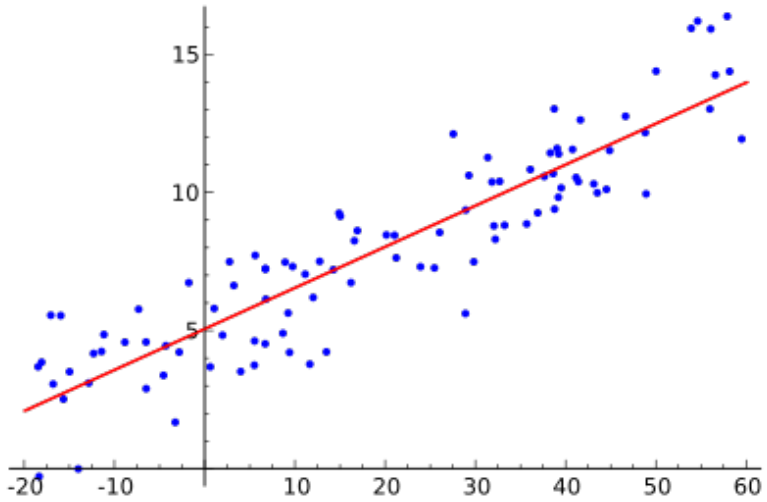
Метод k ближайших соседей

- (+) Очень мало параметров
- (+) Может восстанавливать сложные закономерности
- (-) Нередко показывает плохое качество
- (-) Приличных результатов можно добиться при подборе метрики



Линейные модели

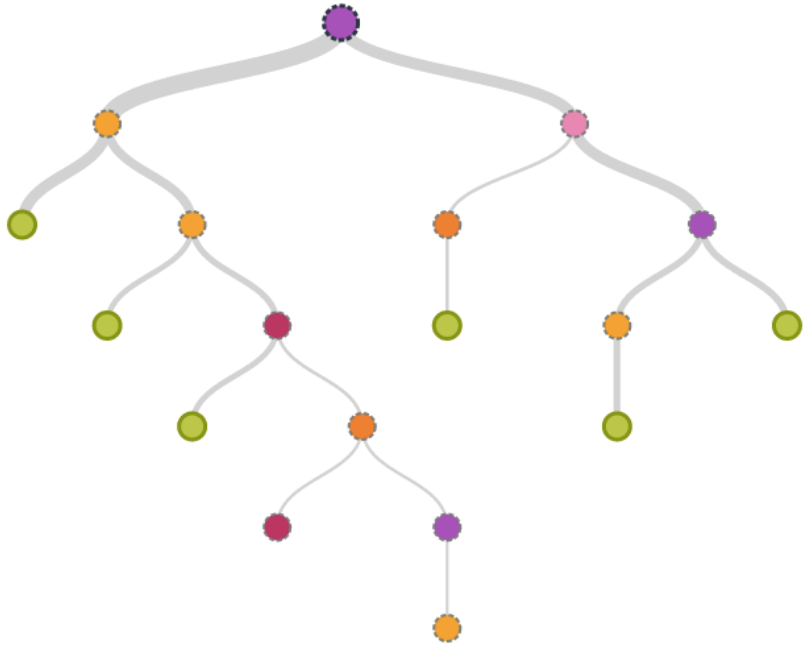
- (+) Легко контролировать переобучение (регуляризация)
- (+) Быстро обучаются даже на огромных объёмах данных
- (+) Хорошо работают при большом числе признаков (например, на категориальных признаках)
- (-) Восстанавливают всего лишь линейные закономерности



Решающие деревья

- (+) Могут дать нулевую ошибку на любой обучающей выборке
- (+) Можно интерпретировать
- (-) Очень легко переобучаются

- (+) Хорошо объединяются в композиции



Обучение с учителем

- Мы изучили основные типы моделей
- Измерять качество в регрессии и классификации тоже научились

- Важные этапы — подготовка данных (откуда их взять?) и разработка признаков
- Пример задачи: автоответ на письма
  - по мотивам статьи «Smart Reply: Automated Response Suggestion for Email», KDD 2016

Автоответ на письма

Сейчас 25% писем-ответов содержат меньше 20 токенов

Требования к автоответу:

- Высокое качество с точки зрения языка и смысла
- Разнообразие
  - Показывать несколько разных вариантов
  - «Yes, I will be there» и «I’ll be there»
- Сохранение приватности переписки пользователей

Задачи машинного обучения

Триггеринг:

- Понять, нужен ли для данного письма автоответ
- Письма со сложным вопросом
  - «Where do you want to go today?»
- Письма, на которые ответ не нужен вообще

Задачи машинного обучения

Выбор наиболее подходящих ответов:

- Классификация на K классов
- K — число допустимых ответов

Задачи не про машинное обучение

- Как собрать обучающую выборку для задачи триггеринга?
- Какие автоответы являются допустимыми?
- На каких признаках обучать классификаторы?
- Как добиться разнообразия ответов?

Триггеринг

Данные:

- Положительные примеры — письма, на которые ответили с мобильного устройства
- Отрицательные примеры — письма, не которые не ответили вообще
- 238 миллионов объектов

Допустимые ответы

Как не надо:

- Your the best!
- Thanks hon
- Yup
- Got it thx
- Leave me alone

Допустимые ответы

Как не надо (все предлагаемые ответы — одинаковые по смыслу):

Yes, I'll be there.
Yes, I will be there.
I'll be there.
Yes, I can.
What time?
I'll be there!
I will be there.
Sure, I'll be there.
Yes, I can be there.
Yes!

Допустимые ответы

- Взяли несколько миллионов наиболее частых ответов пользователей
- Кластеризовали их
- Выбрали из каждого кластера пять представителей и проверили допустимость силами асессоров

Разнообразие ответов

- Из каждого кластера выбирается представитель с максимальной оценкой вероятности от классификатора

- Есть смещение в сторону позитивных ответов
- Если топ-3 кандидатов позитивные, то третий заменяется на наиболее вероятный негативный ответ

Задачи не про машинное обучение

- Как собрать обучающую выборку для задачи триггеринга?
- Какие автоответы являются допустимыми?
- На каких признаках обучать классификаторы?
- Как добиться разнообразия ответов?