Введение в анализ данных

Лекция 10

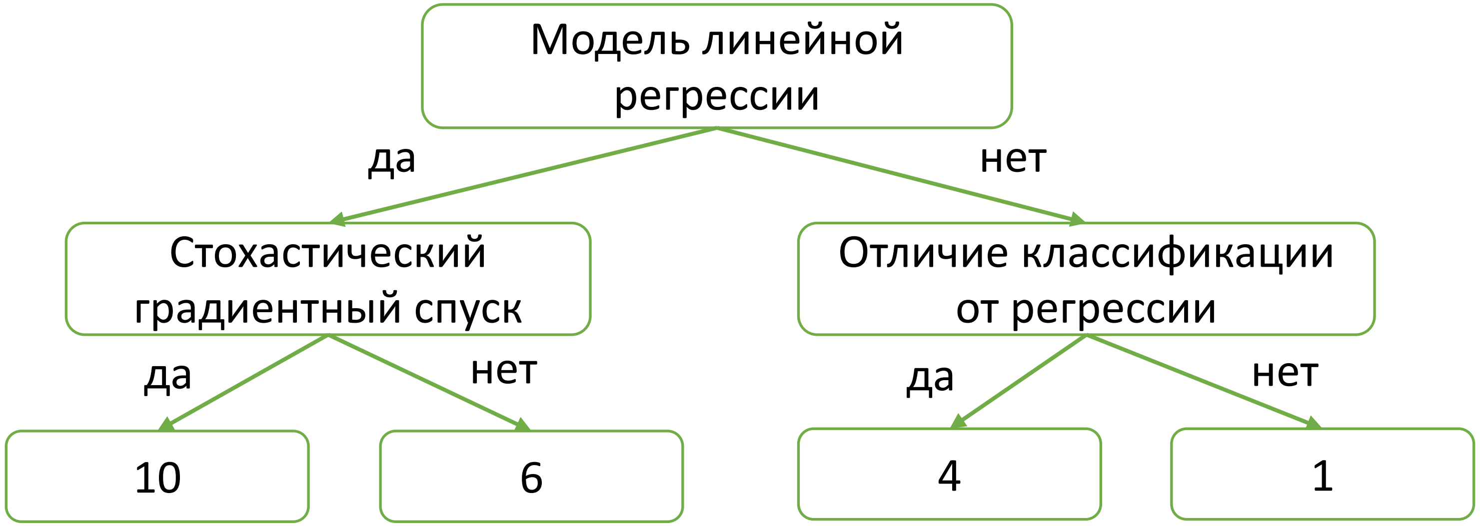
Решающие деревья и композиции моделей

Евгений Соколов

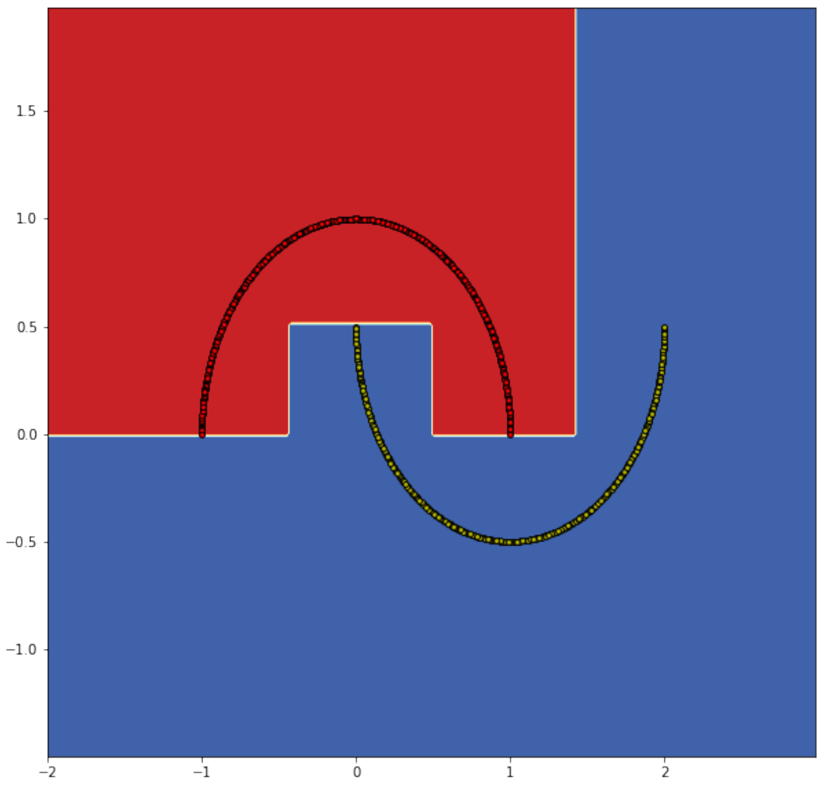
esokolov@hse.ru

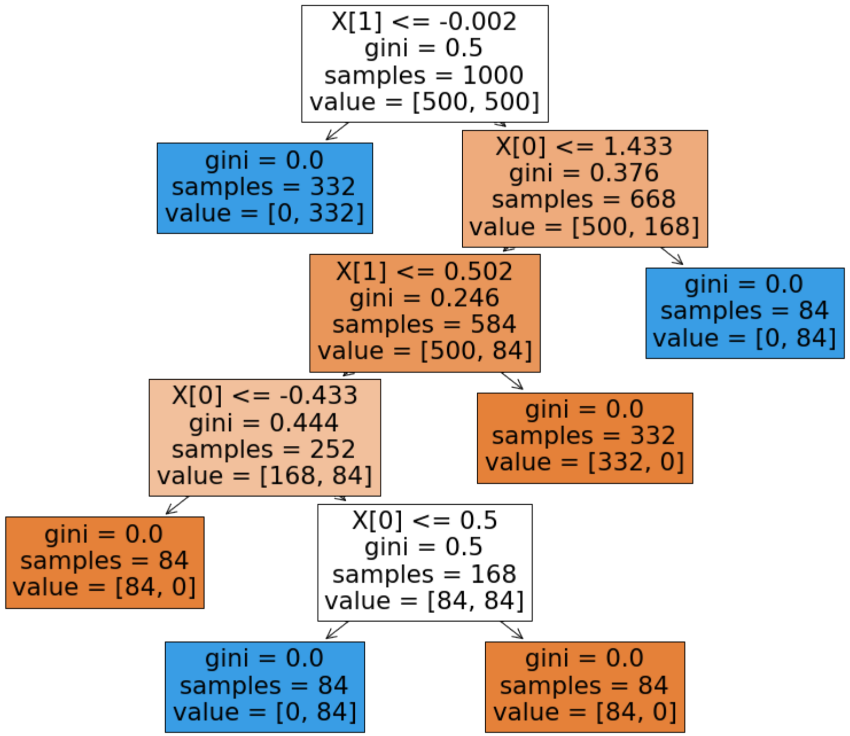
НИУ ВШЭ, 2020

Решающее дерево



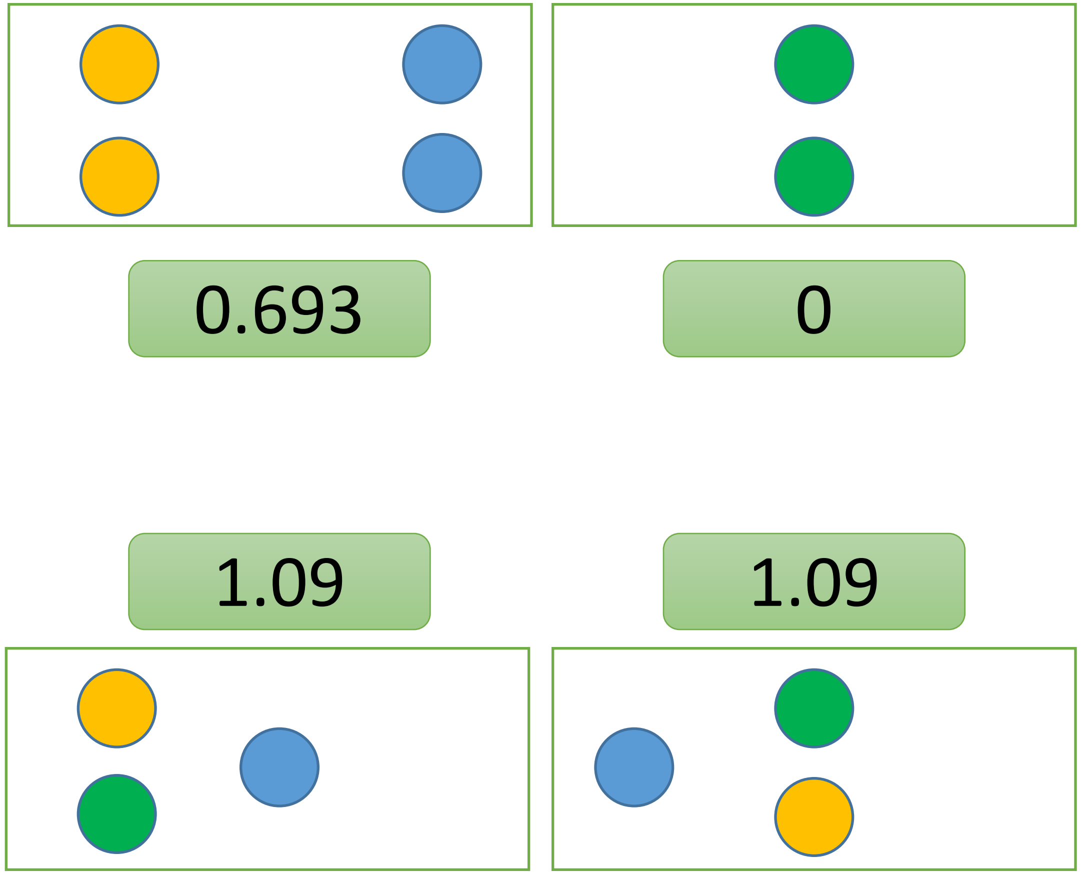
Решающее дерево





Как выбирать предикаты

Как сравнить разбиения?



- $(0.5, 0.5, 0)$ и $(0, 0, 1)$
- $H = 0.693 + 0 = 0.693$

- $(0.33, 0.33, 0.33)$ и $(0.33, 0.33, 0.33)$
- $H = 1.09 + 1.09 = 2.18$

Энтропия

$$H(p_1, \dots, p_K) = -\sum_{i=1}^{K} p_i \log_2 p_i$$

- Характеристика «хаотичности» вершины
- Impurity

Критерий информативности

- Как понять, какой предикат лучше?
- Сравнить хаотичность в исходной вершине и в двух дочерних!

Критерий информативности

- Как понять, какой предикат лучше?
- Сравнить хаотичность в исходной вершине и в двух дочерних!

Критерий информативности

- Как понять, какой предикат лучше?
- Сравнить хаотичность в исходной вершине и в двух дочерних!

$$Q(R, j, t) = H(R) - H(R_\ell) - H(R_r) \to \max_{j,t}$$

Критерий информативности

$$Q(R, j, t) = H(R) - \frac{|R_\ell|}{|R|} H(R_\ell) - \frac{|R_r|}{|R|} H(R_r) \to \max_{j,t}$$

- Или так:

$$Q(R, j, t) = \frac{|R_\ell|}{|R|} H(R_\ell) + \frac{|R_r|}{|R|} H(R_r) \to \min_{j,t}$$

Задача регрессии

$$H(R) = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} (y_i - y_R)^2$$

$$y_R = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} y_i$$

- То есть «хаотичность» вершины можно измерять дисперсией ответов в ней

Жадное построение дерева

Как строить дерево?

- Оптимальный вариант: перебрать все возможные деревья, выбрать самое маленькое среди безошибочных

- Слишком долго

Как строить дерево?

- Мы уже умеем выбрать лучший предикат для разбиения вершины

- Будем строить жадно

- Начнём с корня дерева, будем разбивать последовательно, пока не выполнится некоторый критерий останова

Критерий останова

- Ограничить глубину
- Ограничить количество листьев
- Задать минимальное число объектов в вершине
- Задать минимальное уменьшение хаотичности при разбиении
- И так далее

Жадный алгоритм

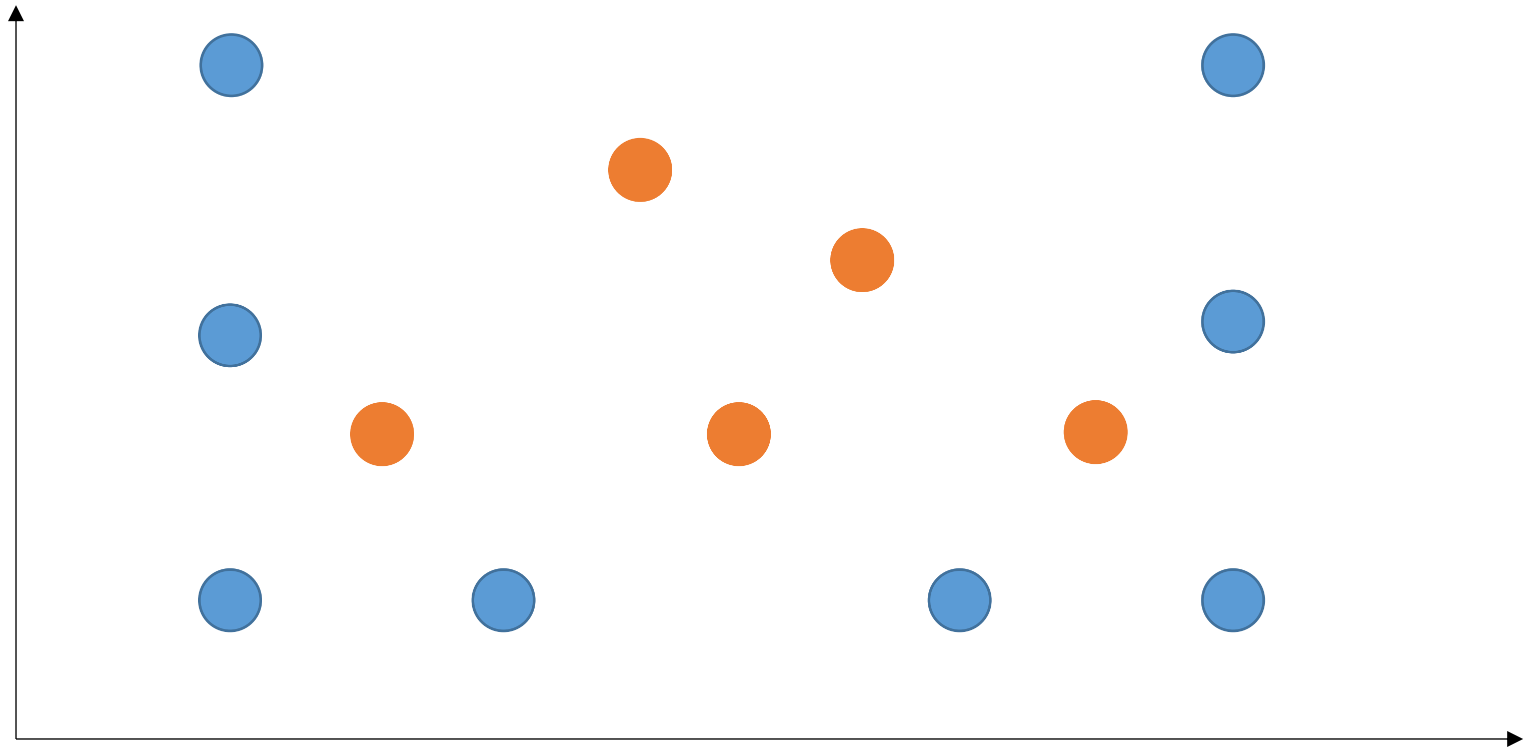
1. Поместить в корень всю выборку: $R_1 = X$
2. Запустить построение из корня: $\mathrm{SplitNode}(1, R_1)$

Жадный алгоритм

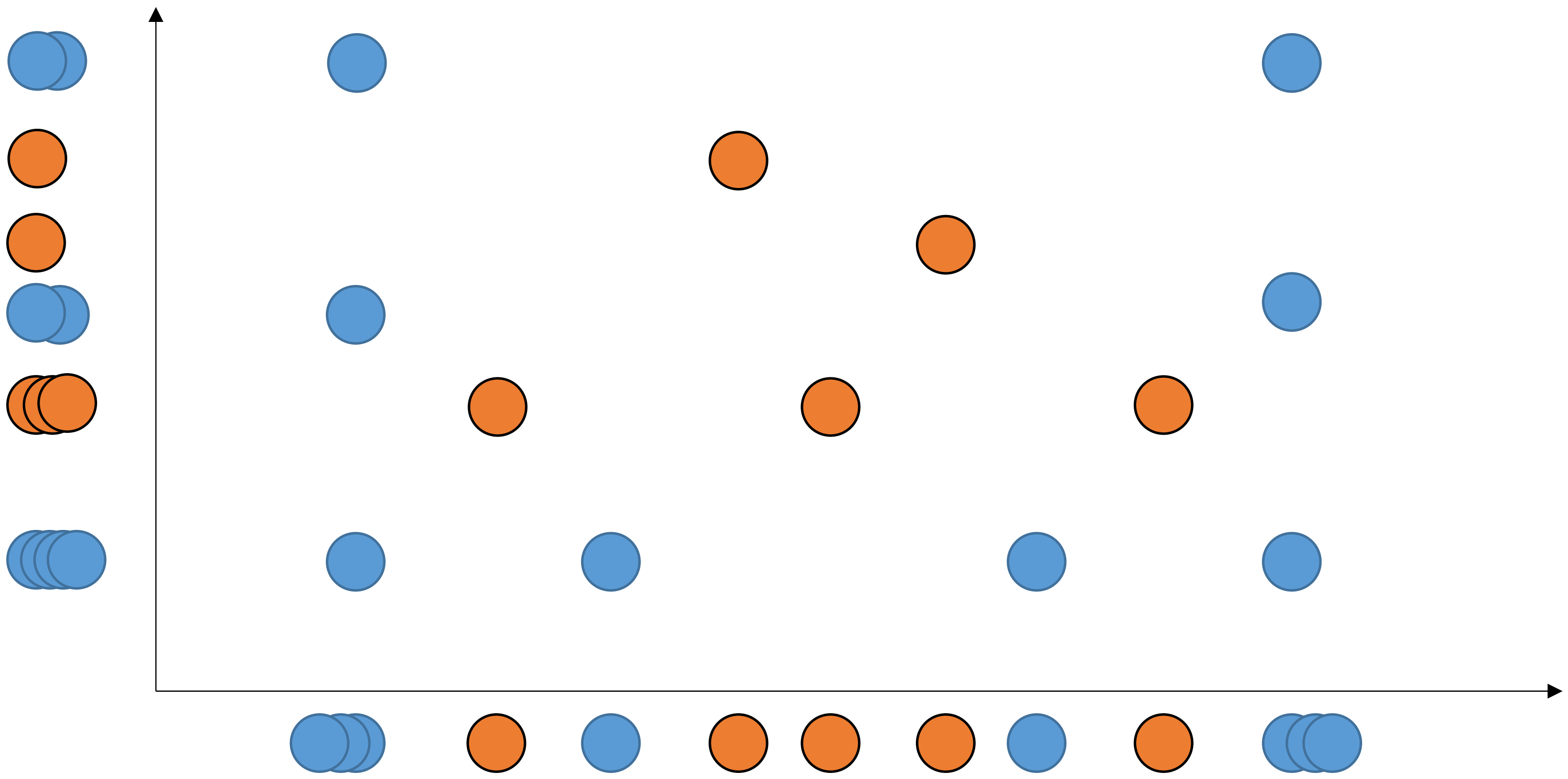
- $\mathrm{SplitNode}(m, R_m)$

1. Если выполнен критерий останова, то выход
2. Ищем лучший предикат: $j, t = \arg\min_{\mathrm{j,t}} Q(R_m, j, t)$
3. Разбиваем с его помощью объекты: $R_\ell = \left\{\{(x, y) \in R_m | [x_j < t]\right\}$, $R_r = \left\{\{(x, y) \in R_m | [x_j \geq t]\right\}$
4. Повторяем для дочерних вершин: $\mathrm{SplitNode}(\ell, R_\ell)$ и $\mathrm{SplitNode}(r, R_r)$

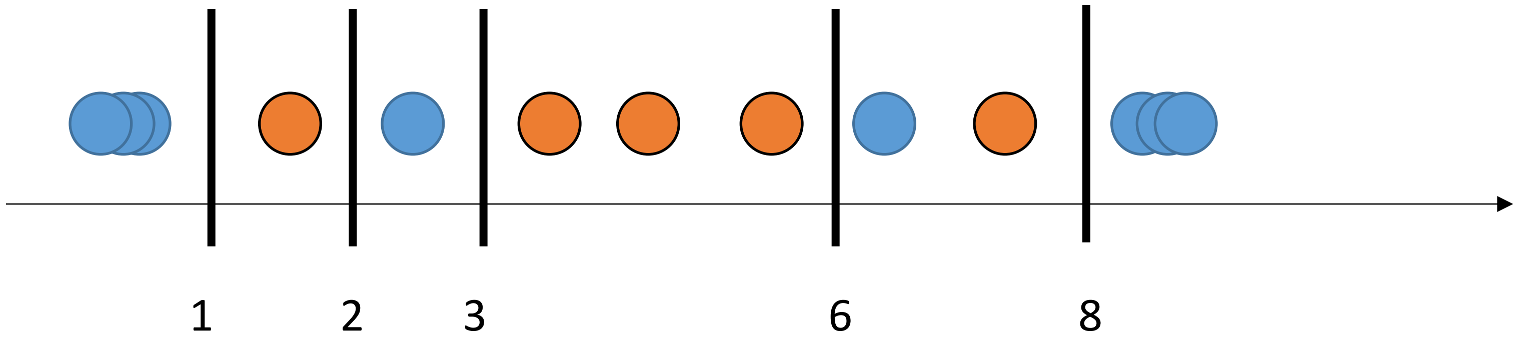
Обучение деревьев



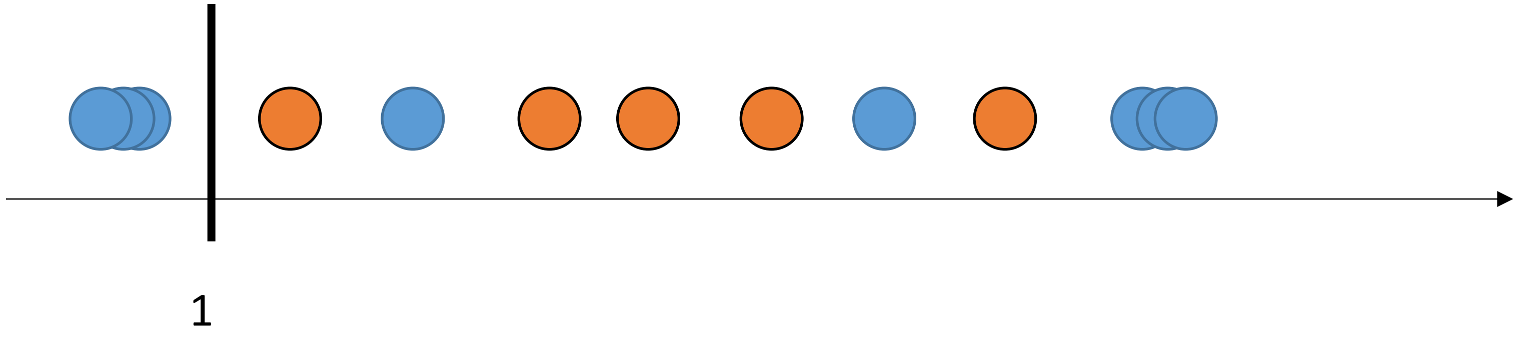
Признаки



Разбиения по признаку 1



Разбиения по признаку 1

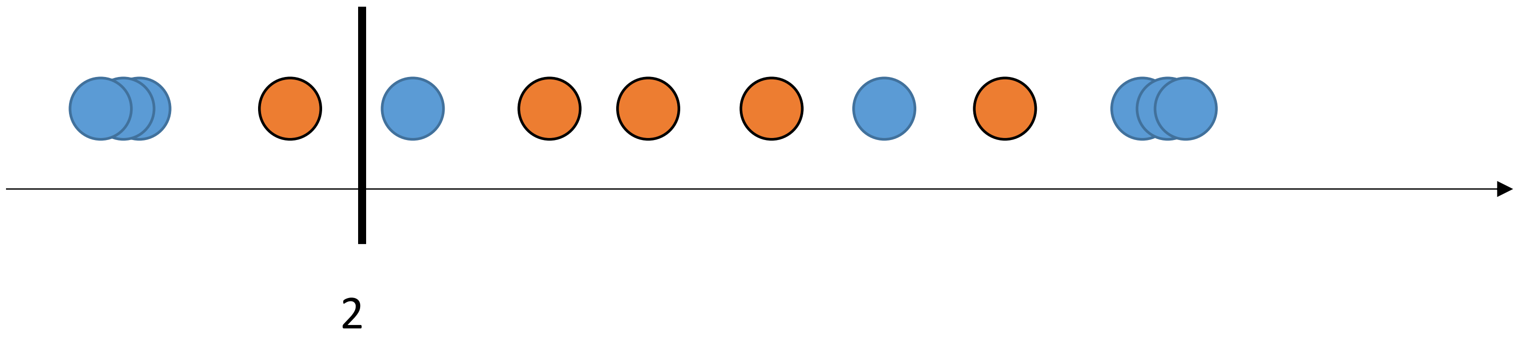


(1, 0)
H(p) = 0

(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

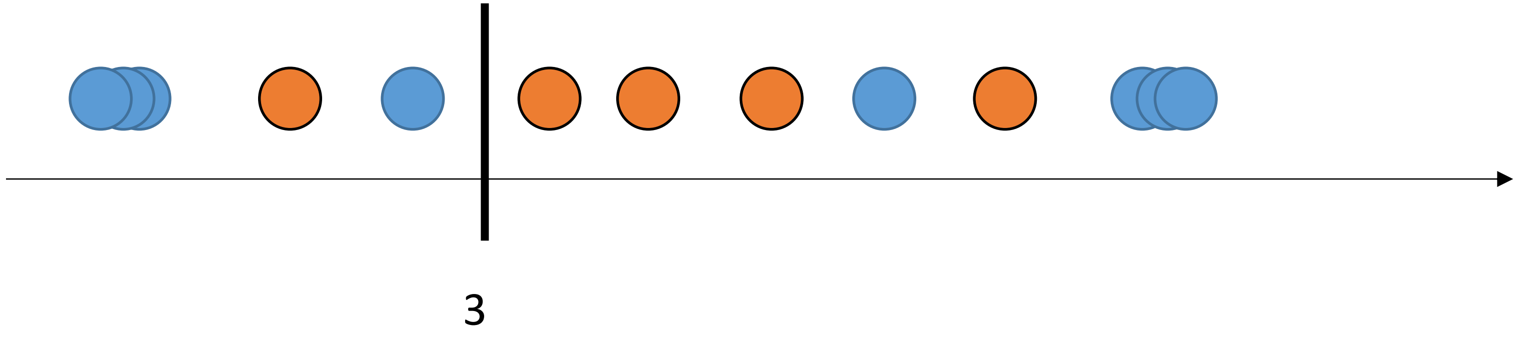


(3/4, 1/4)
H(p) = 0.56

(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

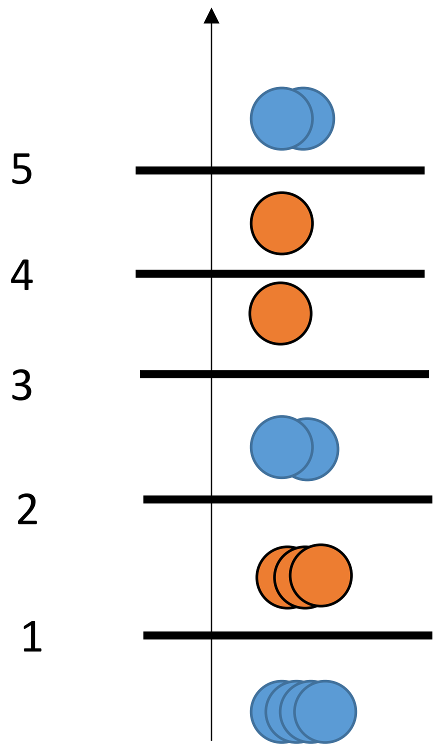


(4/5, 1/5)
H(p) = 0.5

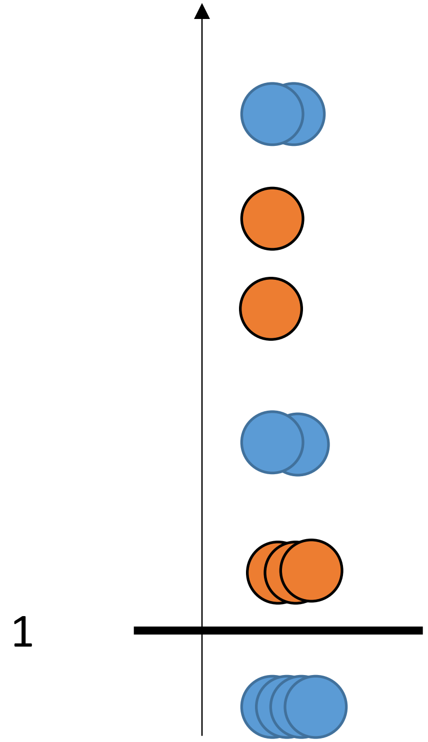
(1/2, 1/2)
H(p) = 0.69

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

Разбиения по признаку 2



Разбиения по признаку 2

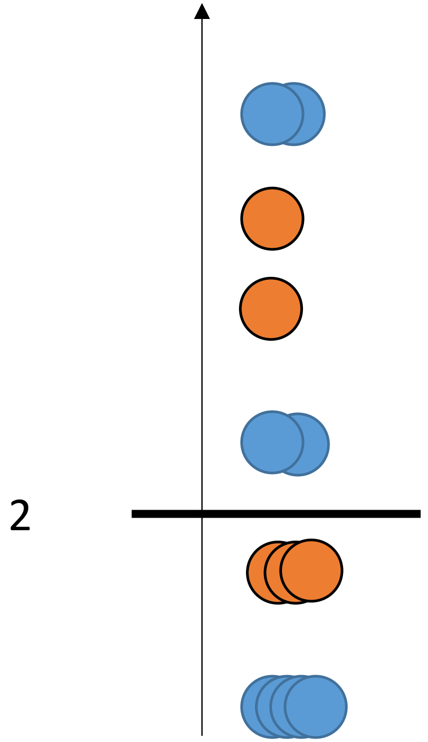


(4/9, 5/9)
H(p) = 0.69

(1, 0)
H(p) = 0

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

Разбиения по признаку 2

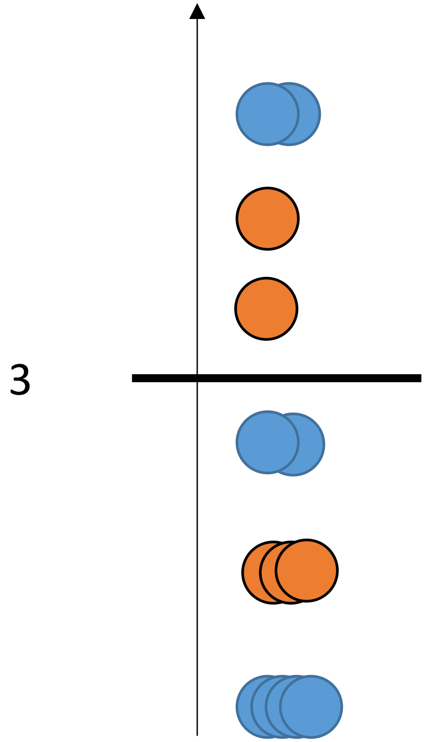


(4/6, 2/6)
H(p) = 0.64

(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

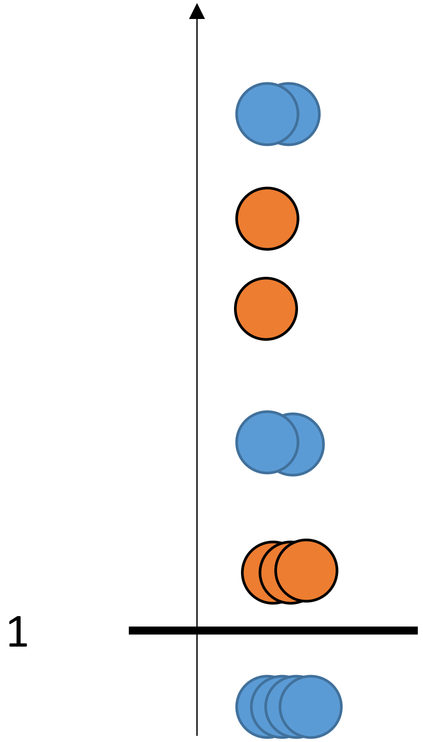


(1/2, 1/2)
H(p) = 0.69

(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



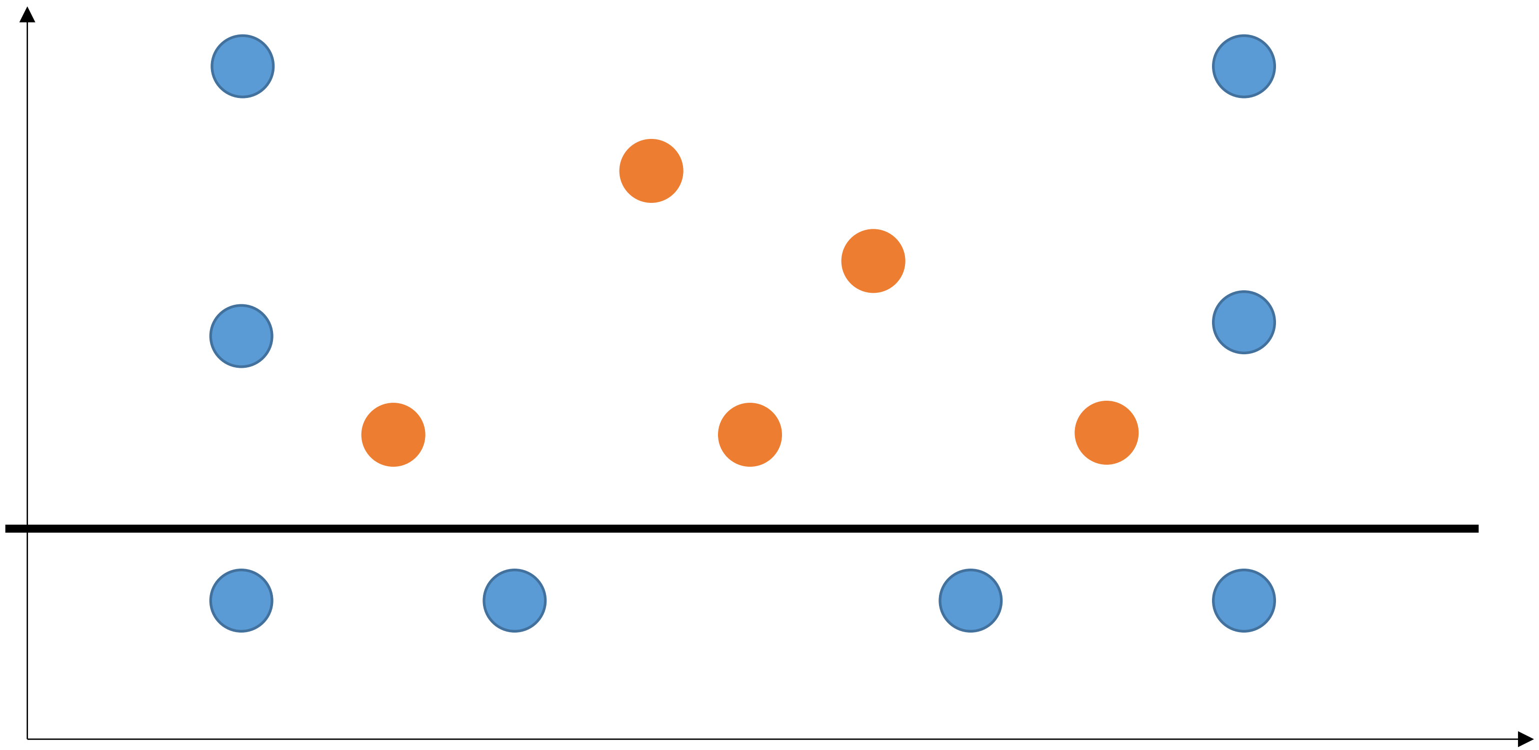
(4/9, 5/9)
H(p) = 0.69

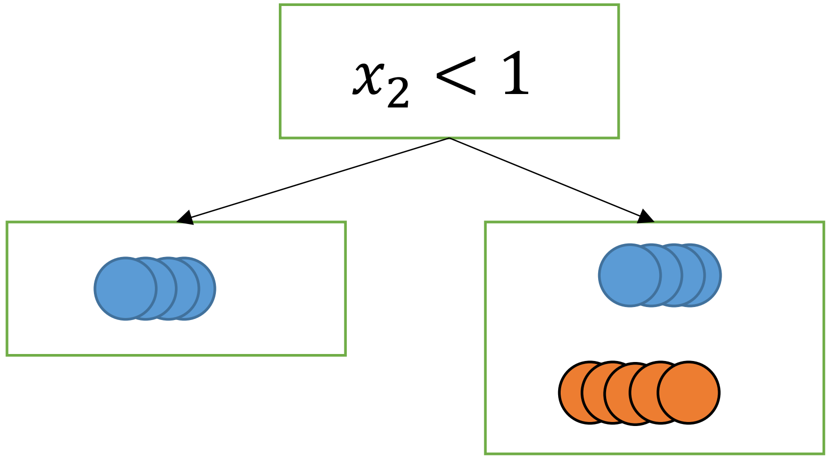
(1, 0)
H(p) = 0

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

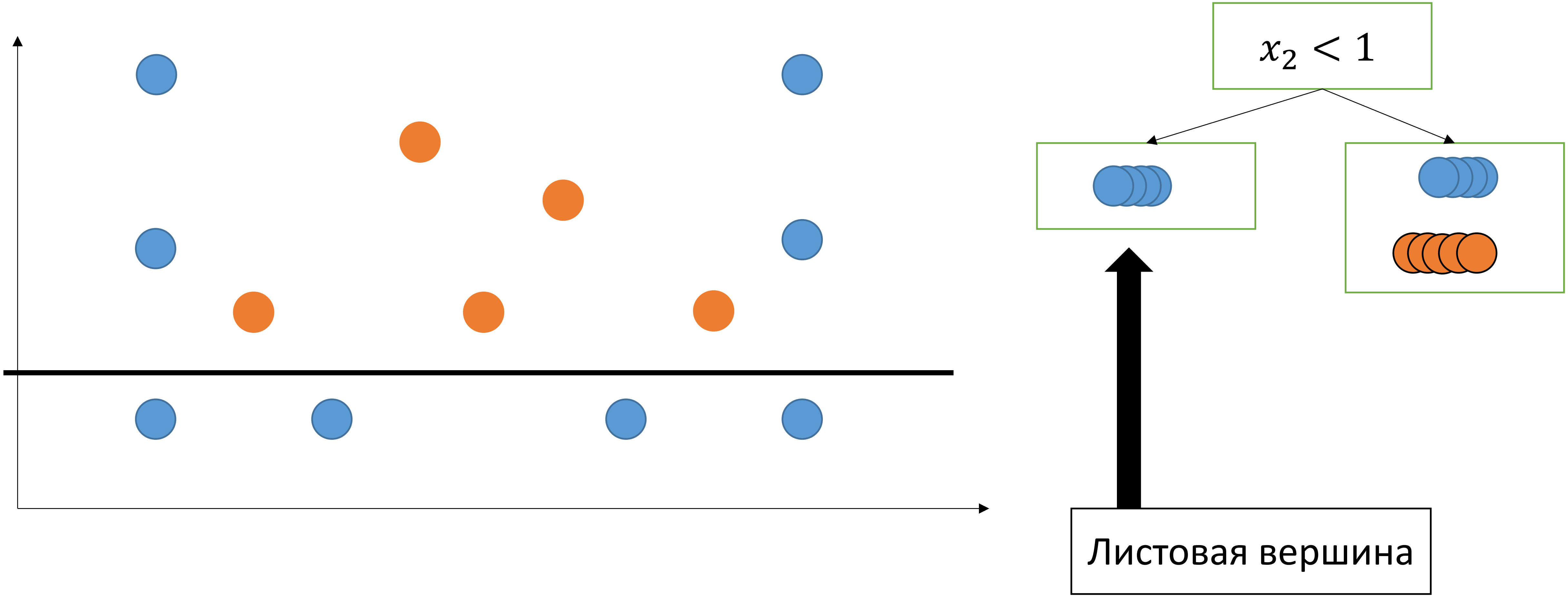
Лучшее разбиение!

Обучение деревьев

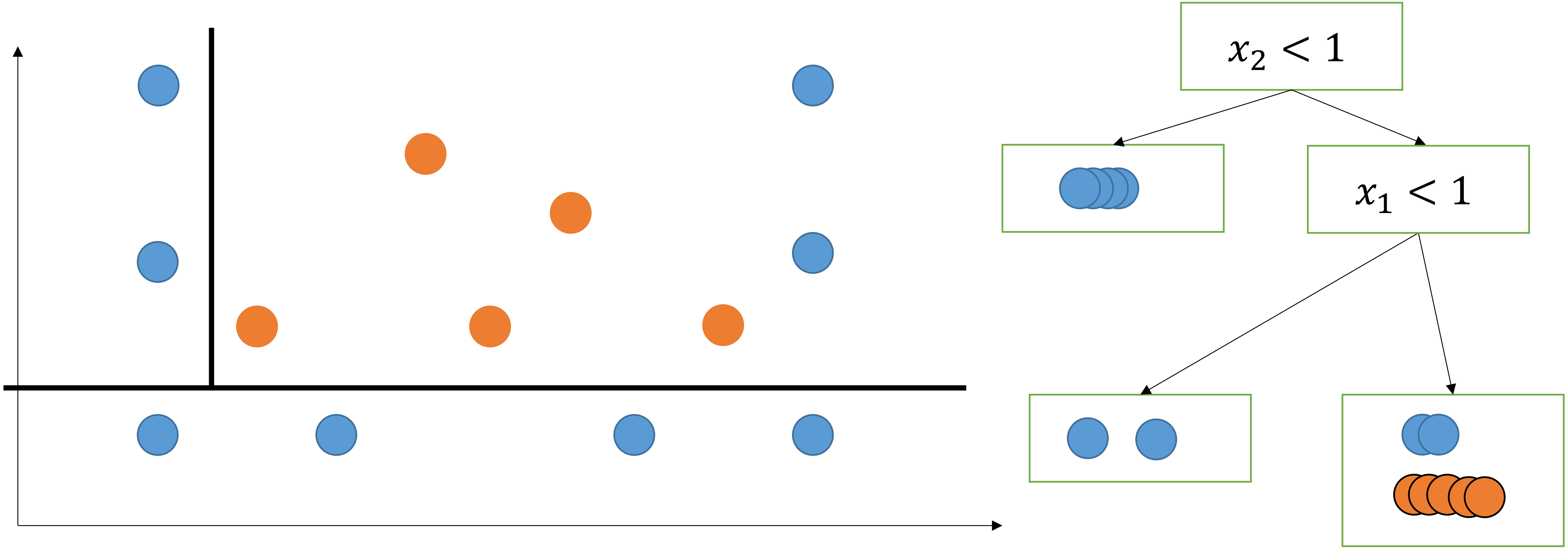




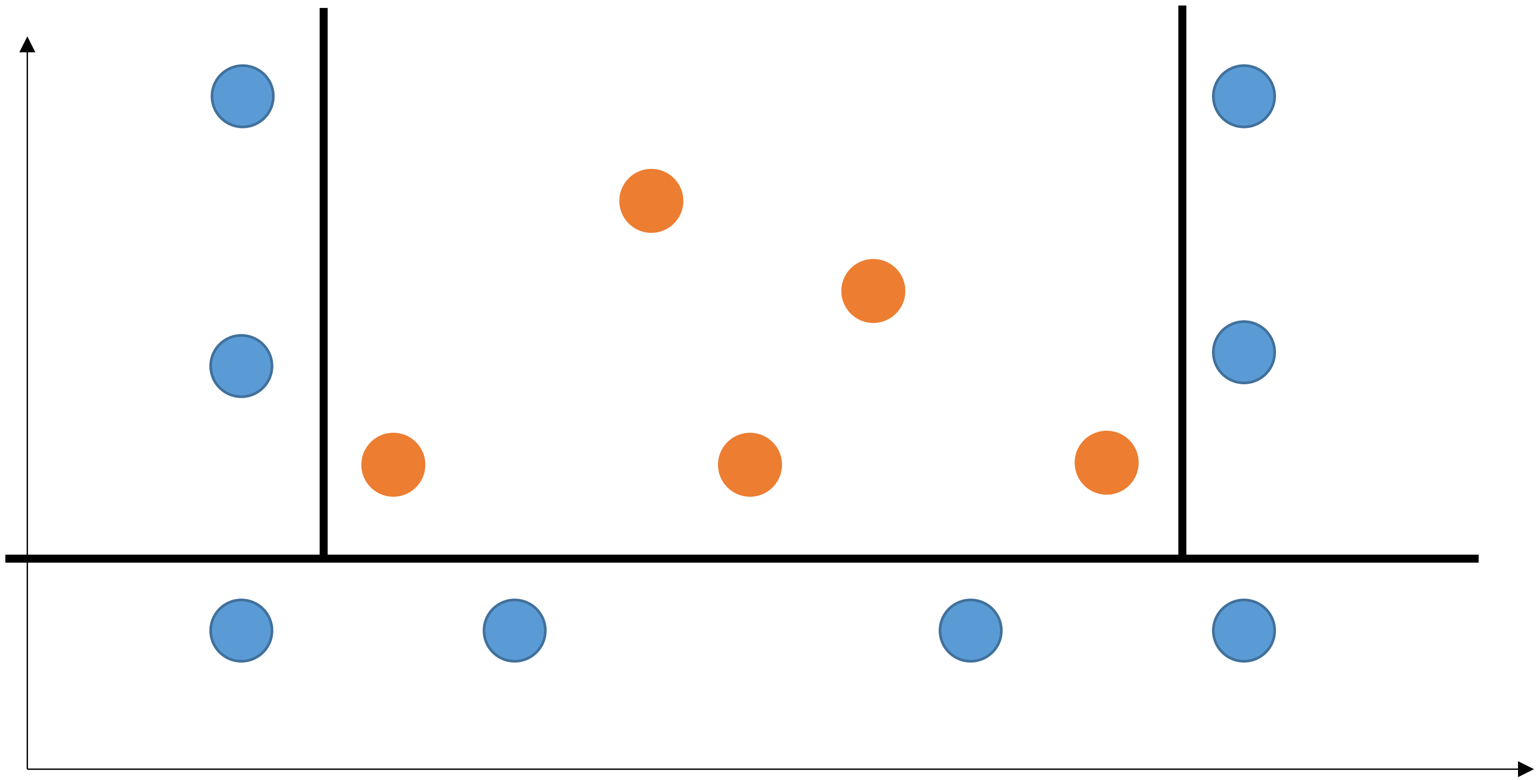
Обучение деревьев

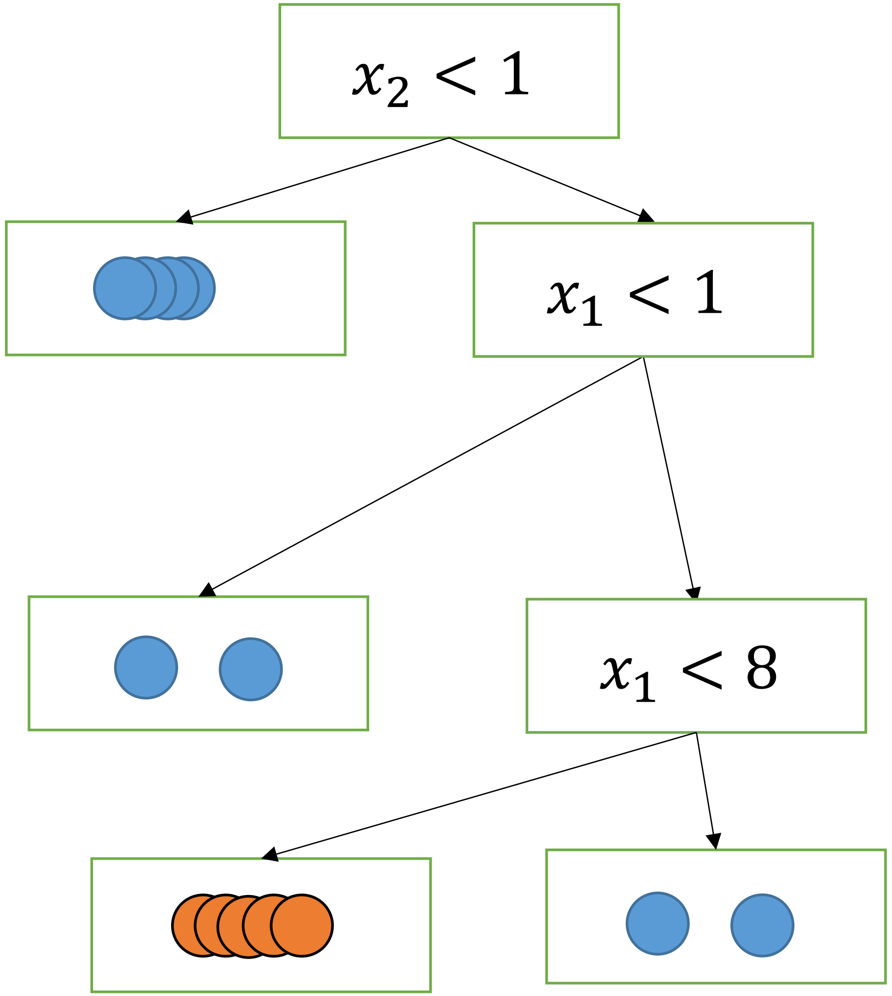


Обучение деревьев



Обучение деревьев





Резюме

- Решающие деревья позволяют строить сложные модели, но есть риск переобучения
- Деревья строятся жадно, на каждом шаге вершина разбивается на две с помощью лучшего из предиктов
- Алгоритм довольно сложный и требует перебора всех предикатов на каждом шаге

Работа с категориальными признаками

Кодирование категориальных признаков

- Значения признака «район»: $U = \{u_1, \dots, u_m\}$
- Новые признаки вместо x_j: $[x_j = u_1], \dots, [x_j = u_m]$
- One-hot кодирование

Кодирование категориальных признаков

Район
ЦАО
ЮАО
ЦАО
САО
ЮАО

→

ЦАО	ЮАО	САО
1	0	0
0	1	0
1	0	0
0	0	1
0	1	0

Кодирование категориальных признаков

Район	**Цена**
ЦАО	10.000.000
ЮАО	4.000.000
ЦАО	9.000.000
САО	7.000.000
ЮАО	5.000.000

Счётчики

- Значения признака x_j: $U_j = \{u_1, \dots, u_m\}$
- Посчитаем все категории в обучающей выборке:

$$\text{count}(j, u_p) = \sum_{i=1}^{\ell} [x_{ij} = u_p]$$

Счётчики

- Значения признака x_j: $U_j = \{u_1, \dots, u_m\}$
- Для регрессии посчитаем суммарный ответ в категории:

$$\text{target}(j, u_p) = \sum_{i=1}^{\ell} [x_{ij} = u_p]\, y_i$$

Счётчики

- Значения признака x_j: $U_j = \{u_1, \dots, u_m\}$
- Для классификации посчитаем классы в категории:

$$\text{target}_k(j, u_p) = \sum_{i=1}^{\ell} [x_{ij} = u_p]\,[y_i = k]$$

Счётчики

- Mean-target encoding
- Задача регрессии
- Заменим категориальный признак на числовой:

$$\widetilde{x_{ij}} = \frac{\operatorname{target}(j, x_{ij})}{\operatorname{count}(j, x_{ij})}$$

Счётчики

- Mean-target encoding
- Задача классификации
- Заменим категориальный признак на K числовых:

$$\widetilde{x_{ij}} = \left(\frac{\mathrm{target}_1(j, x_{ij})}{\mathrm{count}(j, x_{ij})}, \dots, \frac{\mathrm{target}_K(j, x_{ij})}{\mathrm{count}(j, x_{ij})} \right)$$

Кодирование категориальных признаков

Район	Счётчик	Цена
ЦАО	9.500.000	10.000.000
ЮАО	4.500.000	4.000.000
ЦАО	9.500.000	9.000.000
САО	7.000.000	7.000.000
ЮАО	4.500.000	5.000.000

Борьба с переобучением в счётчиках

- Проблема в основном с редкими категориями
- Решение 1: добавление шума

Район	Счётчик	Цена
ЦАО	9.130.000	10.000.000
ЮАО	4.023.000	4.000.000
ЦАО	10.124.000	9.000.000
САО	7.942.000	7.000.000
ЮАО	4.728.000	5.000.000

Борьба с переобучением в счётчиках

- Проблема в основном с редкими категориями
- Решение 2: добавление априорных величин в счётчики

$$\widetilde{x_{ij}} = \frac{\mathrm{target}(j, x_{ij}) + a}{\mathrm{count}(j, x_{ij}) + b}$$

Борьба с переобучением в счётчиках

- Решение 3: кросс-валидация счётчиков

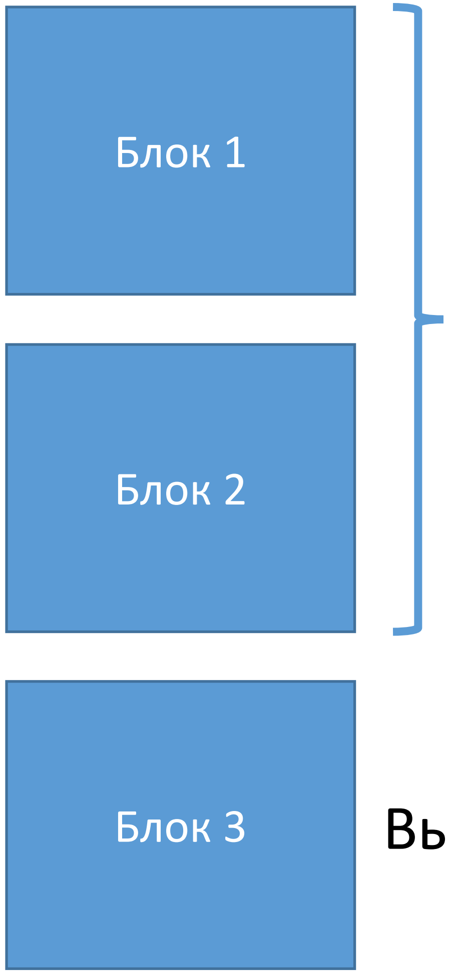
Борьба с переобучением в счётчиках

- Решение 3: кросс-валидация счётчиков

Считаем $\text{count}(j, u_p)$ и $\text{target}(j, u_p)$

Борьба с переобучением в счётчиках

- Решение 3: кросс-валидация счётчиков

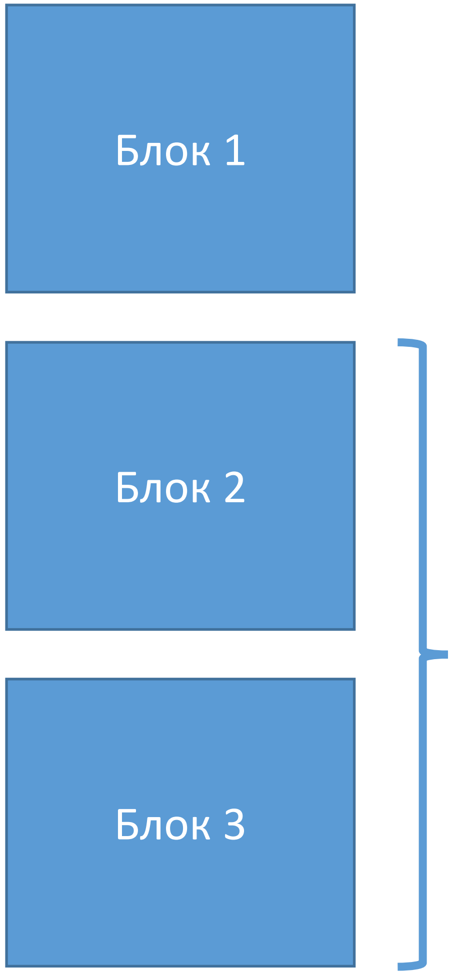


Считаем $\text{count}(j, u_p)$ и $\text{target}(j, u_p)$

Вычисляем признаки: $\widetilde{x_{ij}} = \dfrac{\text{target}(j, x_{ij})}{\text{count}(j, x_{ij})}$

Борьба с переобучением в счётчиках

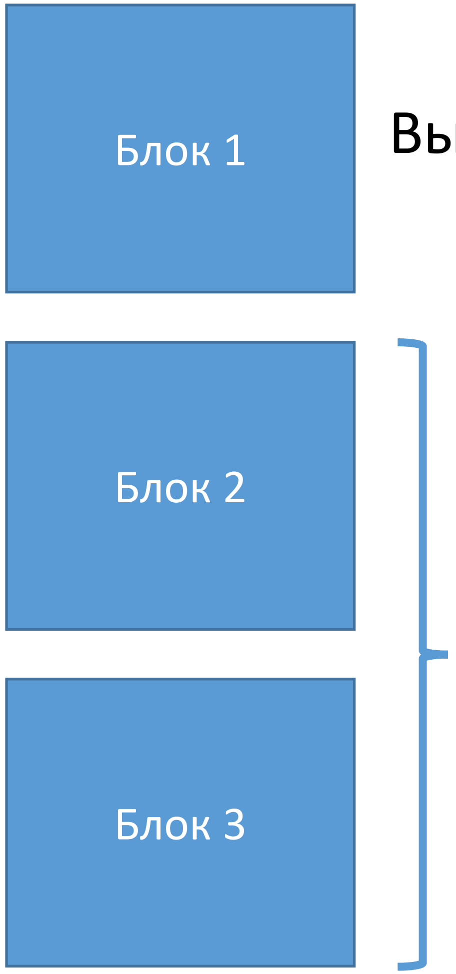
- Решение 3: кросс-валидация счётчиков



Считаем $\mathrm{count}(j, u_p)$ и $\mathrm{target}(j, u_p)$

Борьба с переобучением в счётчиках

- Решение 3: кросс-валидация счётчиков



Вычисляем признаки: $\widetilde{x_{ij}} = \frac{\text{target}(j,x_{ij})}{\text{count}(j,x_{ij})}$

Считаем $\text{count}(j, u_p)$ и $\text{target}(j, u_p)$

Резюме

- Счётчики позволяют заменить категориальный признак на один числовой
- Могут привести к переобучению
- Можно бороться с ним через добавление шума, априорных значений или кросс-валидацию

Неустойчивость деревьев

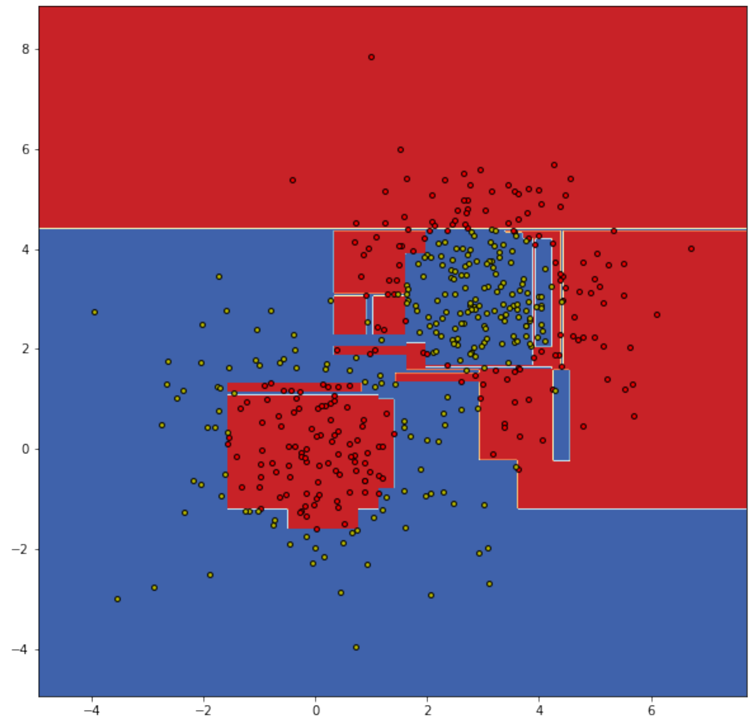
Устойчивость моделей

- $X = (x_i, y_i)_{i=1}^{\ell}$ — обучающая выборка
- Обучаем модель $a(x)$
- Ожидаем, что модель устойчивая
- То есть не сильно меняется при небольших изменениях в X
- $\tilde{X}$ — случайная подвыборка, примерно 90% исходной

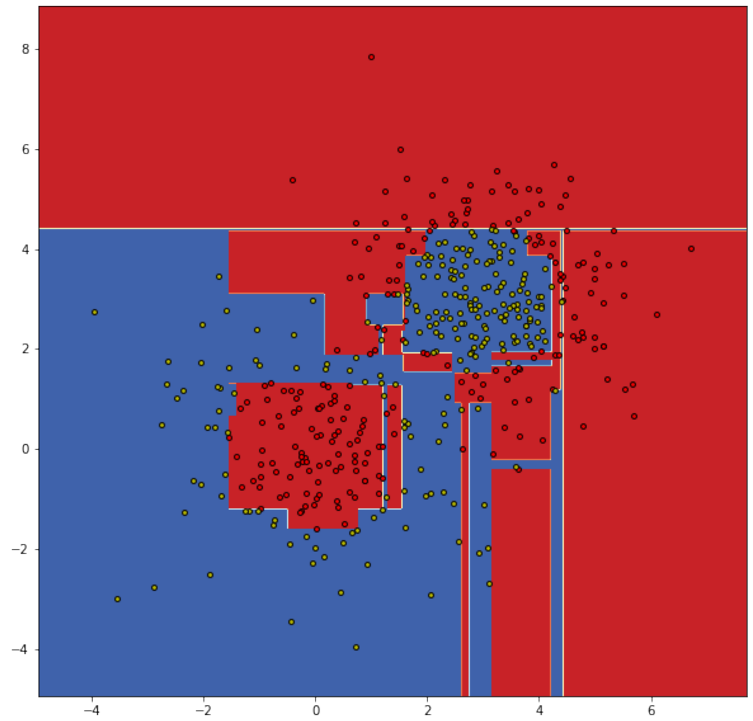
Устойчивость моделей

- $\tilde{X}$ — случайная подвыборка, примерно 90% исходной
- Что будет происходить с деревьями на разных подвыборках?

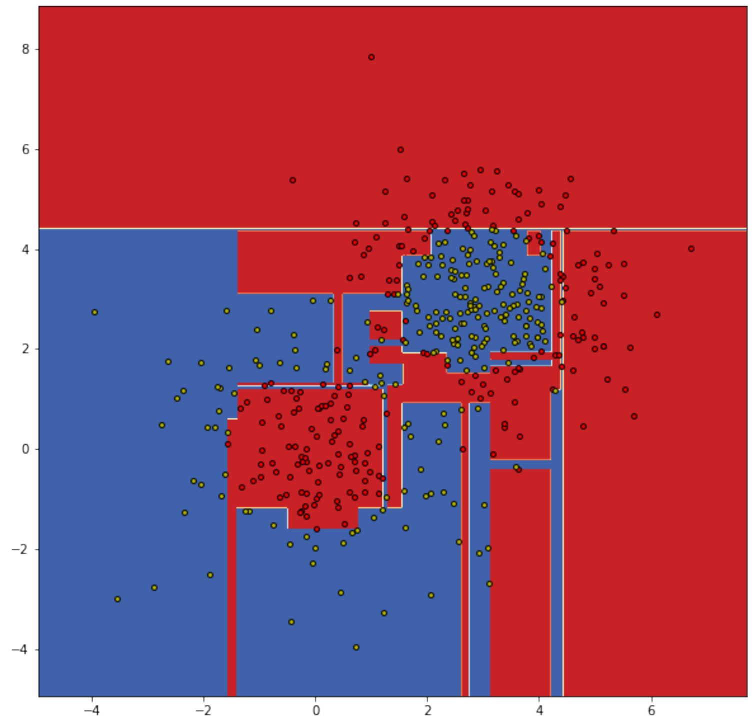
Обучение на подвыборках



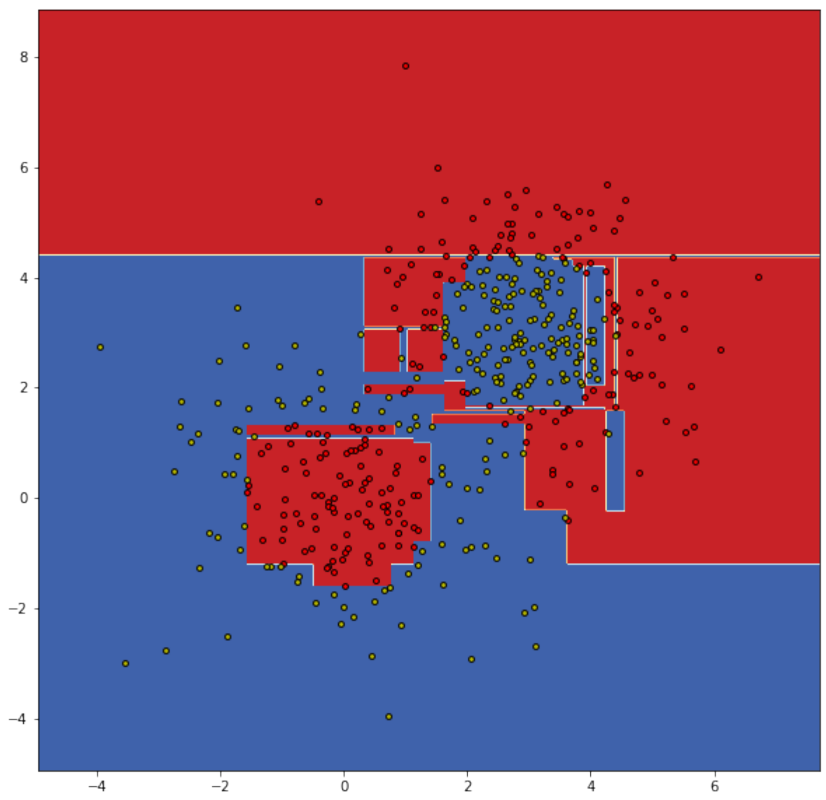
Обучение на подвыборках

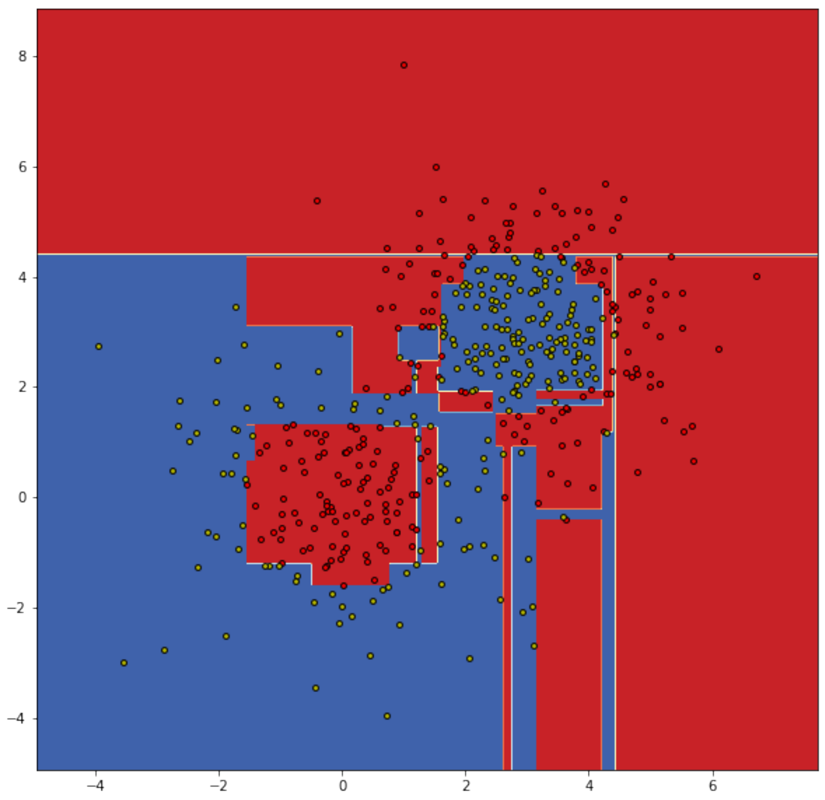


Обучение на подвыборках



Обучение на подвыборках



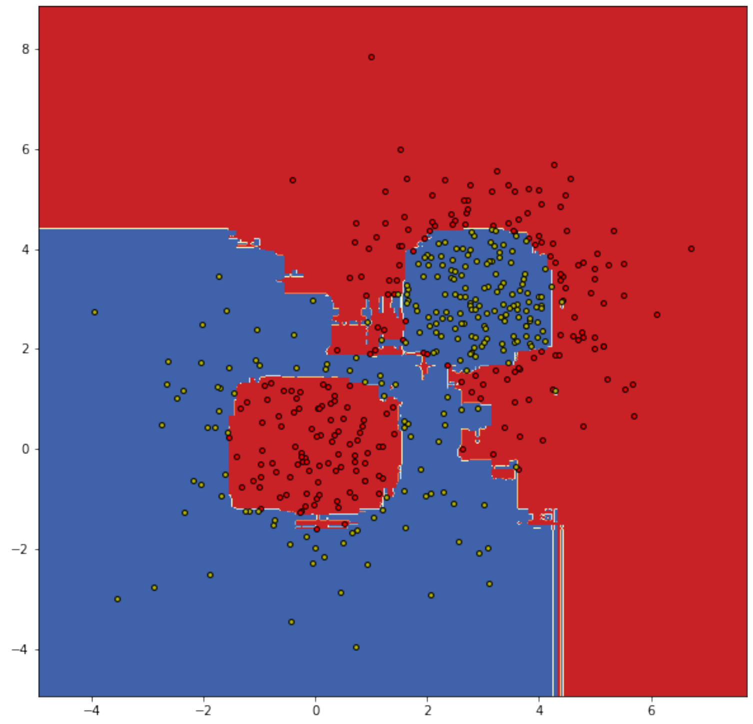


Композиция моделей

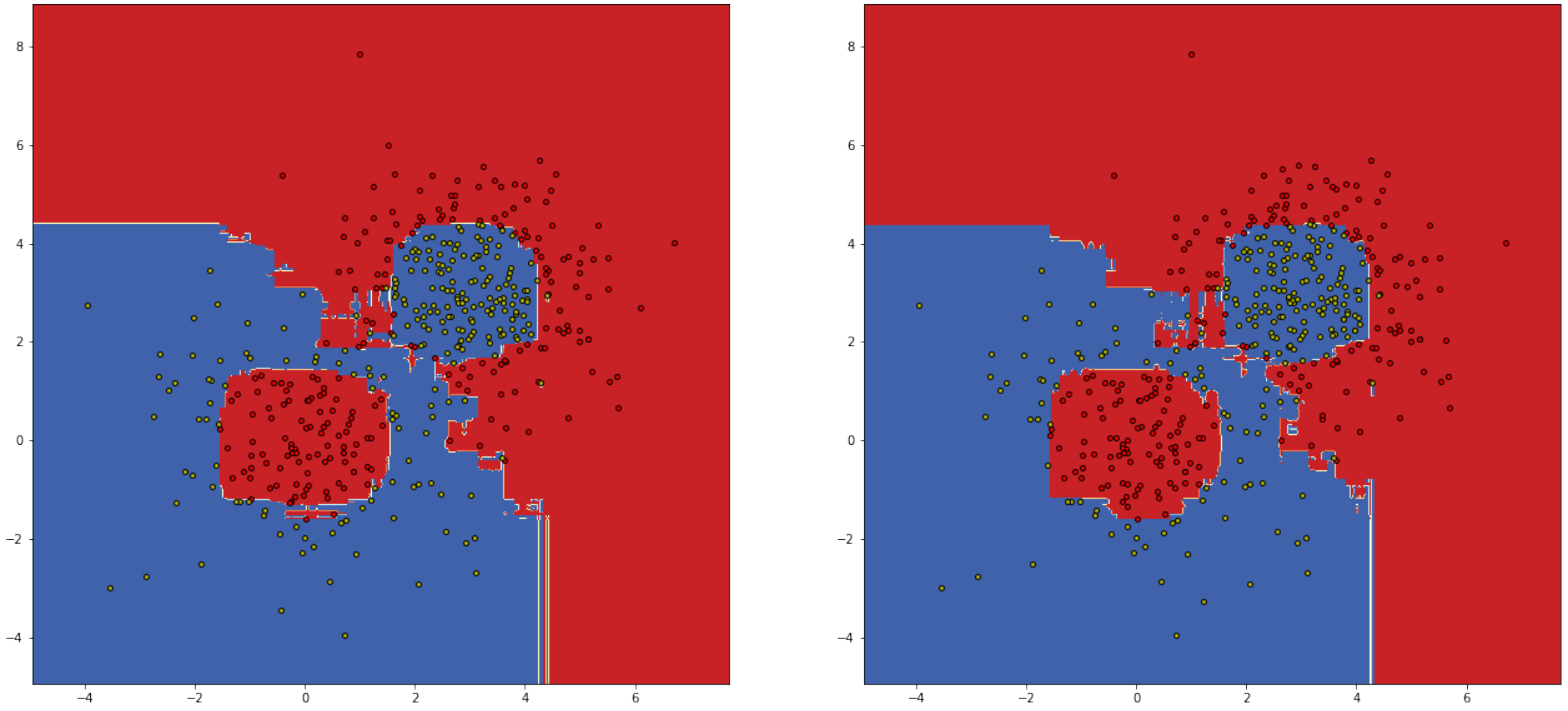
- У нас получилось N деревьев: $b_1(x), \dots, b_N(x)$
- Объединим их через голосование большинством (majority vote):

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

Композиция моделей



Композиция моделей



Голосование по большинству и усреднение

Majority vote

- Какой из двух логотипов более старый?

Majority vote

- Как выглядит корпус Вышки в Перми?

Majority vote

- Покоординатный спуск — это метод оптимизации 1-го или 2-го порядка?

Majority vote

- Дано: N базовых алгоритмов $b_1(x), \dots, b_N(x)$
- 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

- Наблюдение: усреднение результатов повышает их точность
- Измерение артериального давления
- Измерение скорости света
- Усреднение соседних пикселей изображения

Усреднение наблюдений

- Сколько лет факультету компьютерных наук?

Усреднение наблюдений

- Сколько метров в 1 сажени?

Усреднение наблюдений

- Сколько лет лектору?

Усреднение наблюдений

- Сколько всего стран в мире?

Композиции моделей

Общий вид: классификация

- $b_1(x), \dots, b_N(x)$ — базовые модели
- Каждая хотя бы немного лучше случайного угадывания
- Композиция: голосование по большинству (majority vote)

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

Общий вид: регрессия

- $b_1(x), \dots, b_N(x)$ — базовые модели
- Каждая хотя бы немного лучше случайного угадывания
- Композиция: усреднение

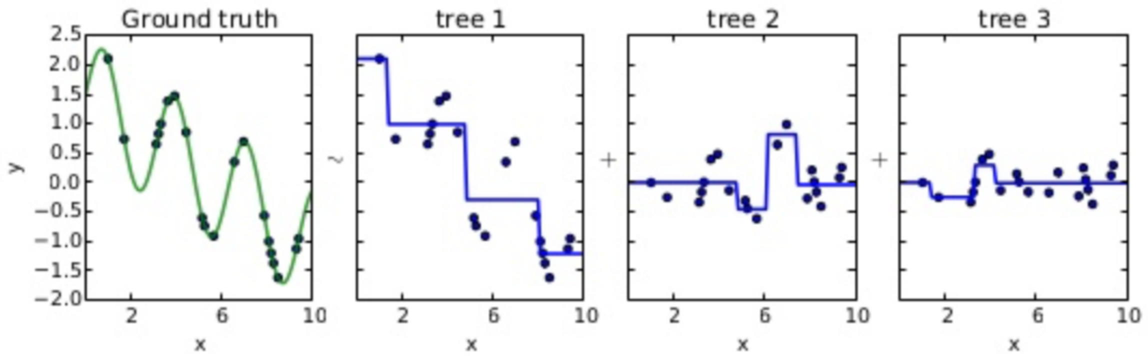
$$a_N(x) = \frac{1}{N} \sum_{n=1}^{N} b_n(x)$$

Базовые модели

- $b_1(x), \dots, b_N(x)$ — базовые модели
- Как на одной выборке построить N различных моделей?
- Вариант 1: обучить их независимо на разных подвыборках
- Вариант 2: обучать последовательно для корректировки ошибок

Бустинг

- Каждая следующая модель исправляет ошибки предыдущих
- Например, градиентный бустинг



Бэггинг

- Bagging (<u>b</u>ootstrap <u>agg</u>regat<u>ing</u>)
- Базовые модели обучаются независимо
- Каждый обучается на подмножестве обучающей выборки
- Подмножество выбирается с помощью бутстрапа

Бутстрап

- Выборка с возвращением
- Берём ℓ элементов из X
- Пример: $\{x_1, x_2, x_3, x_4\} \to \{x_1, x_2, x_2, x_4\}$
- В подвыборке будет ℓ объектов, из них около 63.2% уникальных
- Если объект входит в выборку несколько раз, то мы как бы повышаем его вес

Случайные подпространства

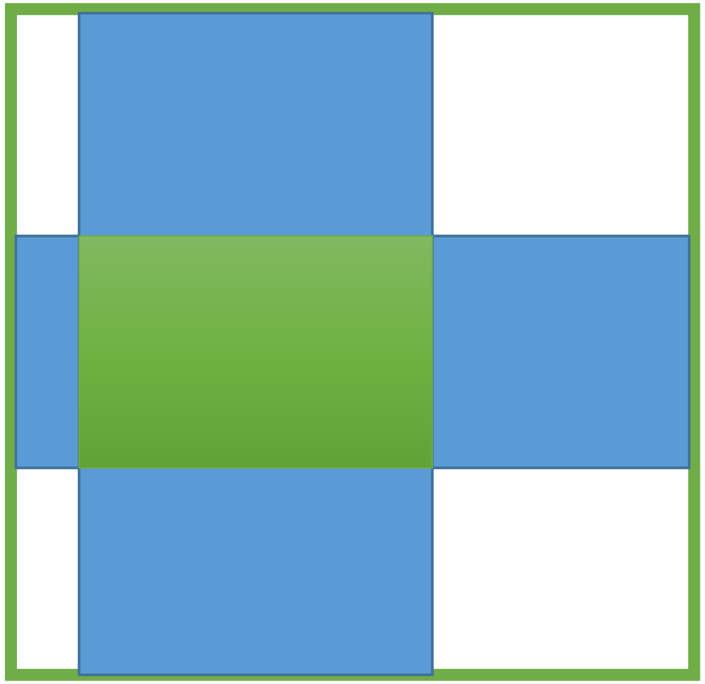
- Выбираем случайное подмножество признаков
- Обучаем модель только на них

Случайные подпространства

- Выбираем случайное подмножество признаков
- Обучаем модель только на них
- Может быть плохо, если имеются важные признаки, без которых невозможно построить разумную модель

Виды рандомизации

- Бэггинг: случайная подвыборка
- Случайные подпространства: случайное подмножество признаков



Резюме

- Будем объединять модели в композиции через усреднение или голосование большинством
- Бэггинг — композиция моделей, обученных независимо на случайных подмножествах объектов
- Можно ещё рандомизировать по признакам
- Как лучше всего?