

- $\left[\begin{matrix} \underline{\omega}, \underline{\phi(x)} \end{matrix} \right] \in \mathbb{R}$
веса
признак
 $\in \{0, 1\}$
 - $\text{sign} \left[\begin{matrix} \underline{\omega}, \underline{\phi(x)} \end{matrix} \right] \in \underbrace{\{-1, 1\}}_{\text{типа класса?}} : g \left(\begin{matrix} \underline{\omega}, \underline{\phi(x)} \end{matrix} \right)$
- [Решающие деревья]**

Лекция 7

линейная по ϕ



$$\langle w, \phi(x) \rangle$$

$$\langle w, (x, x^2, x_i \cdot x_j, \dots) \rangle$$

Как делать нелинейные модели?

Предсказание стоимости квартиры

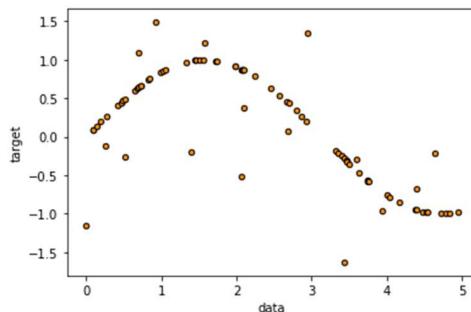
- Признаки: площадь, этаж, расстояние до метро и т.д.
- Целевая переменная: рыночная стоимость квартиры

Предсказание стоимости квартиры

- Линейная модель:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж}) \\ + w_3 * (\text{расстояние до метро}) + \dots$$

- Вряд ли признаки линейно связаны с целевой переменной



Предсказание стоимости квартиры

- Линейная модель:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж}) \\ + w_3 * (\text{расстояние до метро}) + \dots$$

- Вряд ли признаки не связаны между собой

$$x \mapsto x_i \cdot x_j \quad i=j \quad x_i^2$$

Предсказание стоимости квартиры

- Линейная модель с полиномиальными признаками:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж})$$

$$+ w_3 * (\text{расстояние до метро}) + w_4 * (\underbrace{\text{площадь}}_D)^2$$

$$+ w_5 * (\underbrace{\text{этаж}}_D)^2 + w_6 * (\text{расстояние до метро})^2$$

$$+ w_7 * (\underbrace{\text{площадь}}_C * \underbrace{(\text{этаж})}_D) + \dots$$

L_2

$$(X^T X + \lambda I)$$

⑤ 1. Возможные параллельные иерархии между

$$D \rightarrow C^2 + D^2$$

2. Рост пространства признаков

Предсказание стоимости квартиры

- Линейная модель с полиномиальными признаками:

$$\begin{aligned}a(x) = & w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж}) \\& + w_3 * (\text{расстояние до метро}) + w_4 * (\text{площадь})^2 \\& + w_5 * (\text{этаж})^2 + w_6 * (\text{расстояние до метро})^2 \\& + w_7 * (\text{площадь}) * (\text{этаж}) + \dots\end{aligned}$$

- Может быть сложно интерпретировать модель
- Что такое $(\text{расстояние до метро}) * (\text{этаж})^2$?

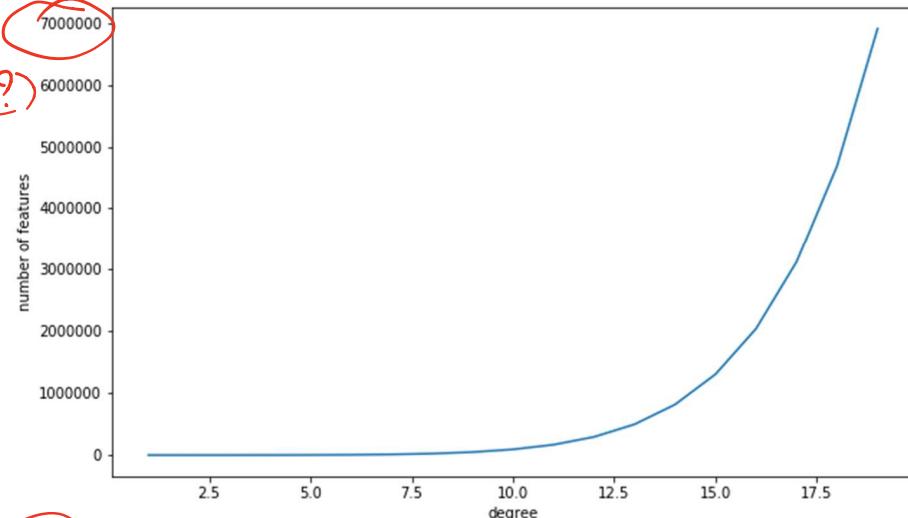
Предсказание стоимости квартиры

- Допустим, изначально имеем 10 признаков
- Полиномиальных степени 2: 55
- Полиномиальных степени 3: 220
- Полиномиальных степени 4: 715

Предсказание стоимости квартиры

- Линейная модель с полиномиальными признаками:

• $D \gg N$
• ✓
всегда
число?





$$\cdot S = 50 \dots 100$$

• этаж

Предсказание стоимости квартиры

① $\times \rightarrow$ правило

- Линейная модель с логическими правилами:

$$a(x) = w_0 + w_1 * [30 < \text{площадь} < 50]$$

$$+ w_2 * [50 < \text{площадь} < 80] + \dots$$

$$+ w_{20} * [2 < \text{этаж} < 5] + \dots$$

$$+ w_{100} * [30 < \text{площадь} < 50][2 < \text{этаж} < 5] + \dots$$

} one-hot
encoding.

- Признаки интерпретируются куда лучше: $[30 < \text{площадь} < 50][2 < \text{этаж} < 5][100 < \text{расстояние до метро} < 500]$
- Но их станет еще больше!

✓ ② признаки как логические правила

• алгоритм.

② не все возможные такие признаки, а "хорошие"?

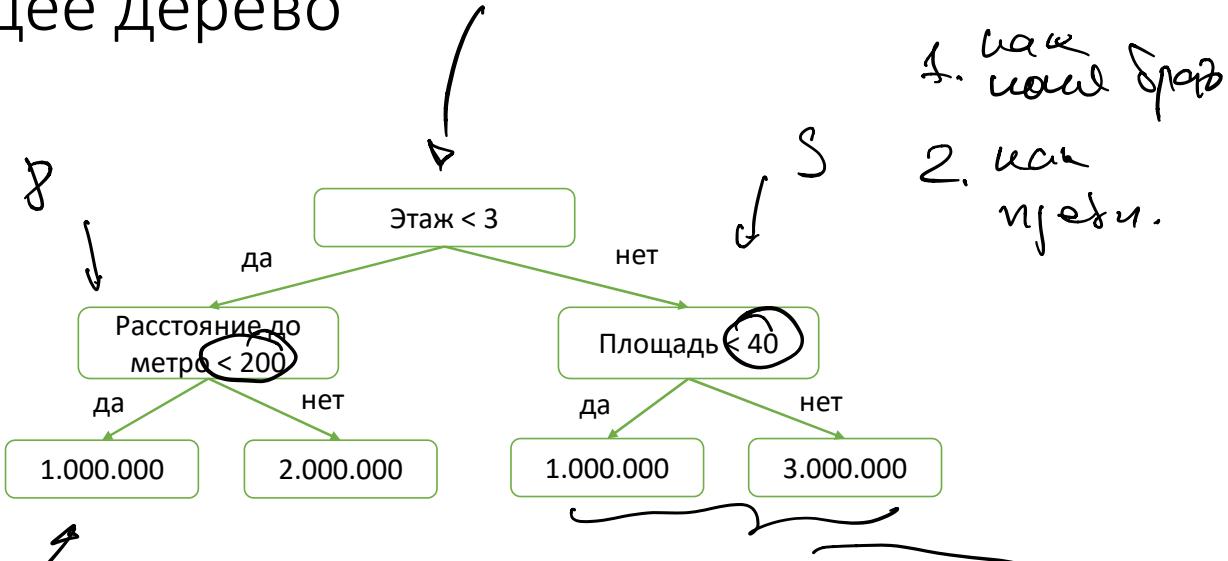
Решающие деревья

Логические правила

- $[30 < \text{площадь} < 50][2 < \text{этаж} < 5][500 < \text{расстояние до метро} < 1000]$
- Легко объяснить, как работают
- Находят нелинейные закономерности
- Нужно как-то искать хорошие логические правила
- Нужно уметь составлять модели из логических правил

чего? $\leftarrow [\text{этаж}, \beta, S]$

Решающее дерево



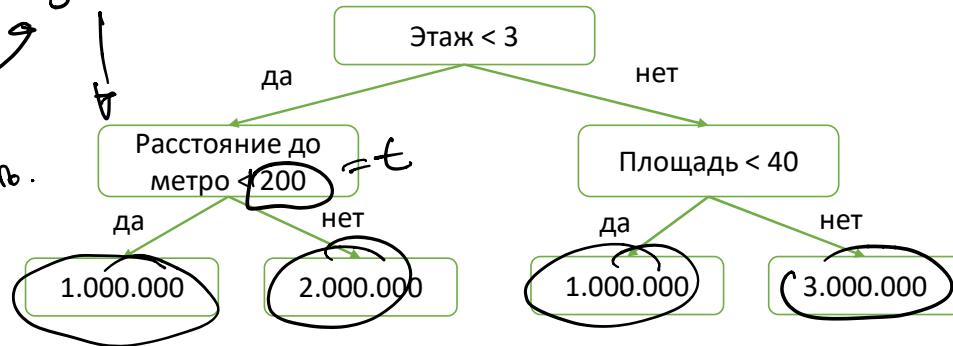
$$y = f(x)$$

$$[\text{этаж} < 3] \cdot [\beta < 200] \cdot 80^6$$

$$+ [\text{этаж} < 3] \cdot [\beta > 200] \cdot 2 \cdot 10^6 + \dots$$



из
нашего
учебника
старшего
уровня



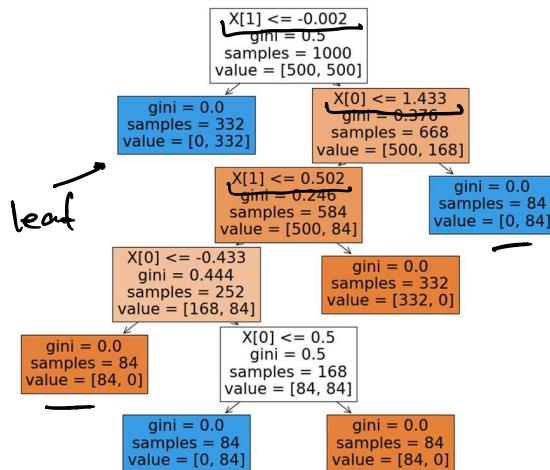
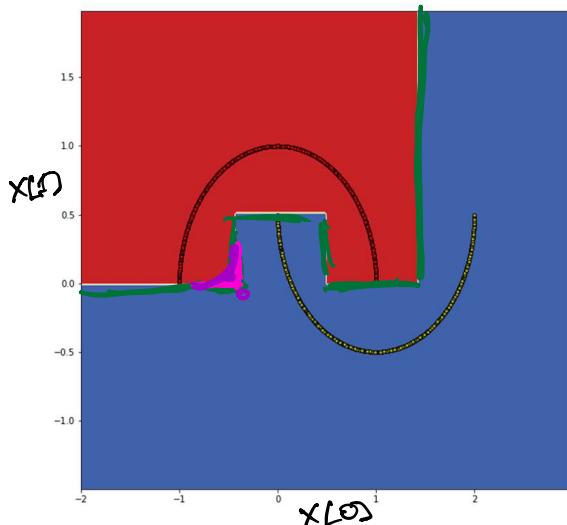
- Внутренние вершины: предикаты $[x_j < t]$
- Листья: прогнозы $c \in \mathbb{Y}$

$$c \in \mathbb{R}$$

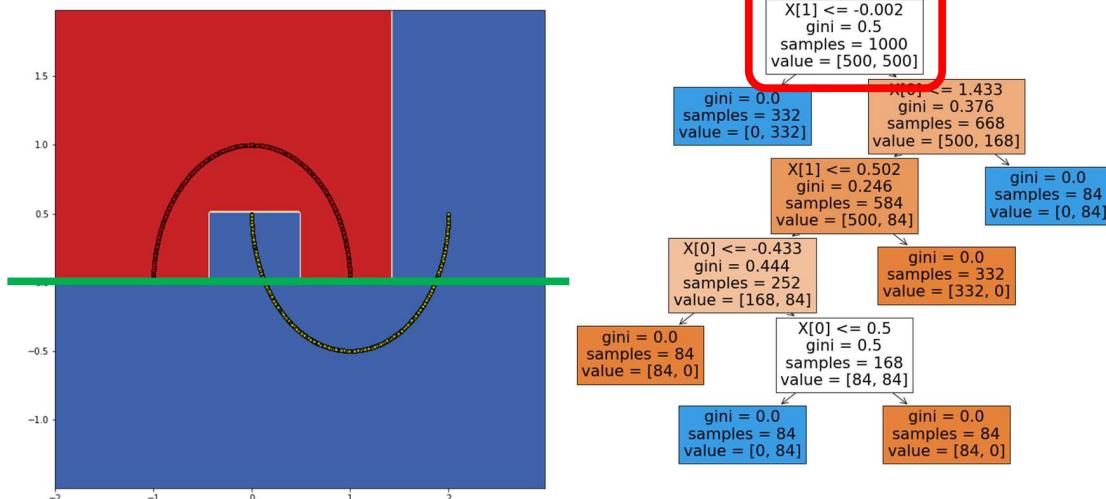
$$c \in [-l, l] \quad \dots$$

Решающее дерево
⇒ $\sum' \text{деревья}_i$

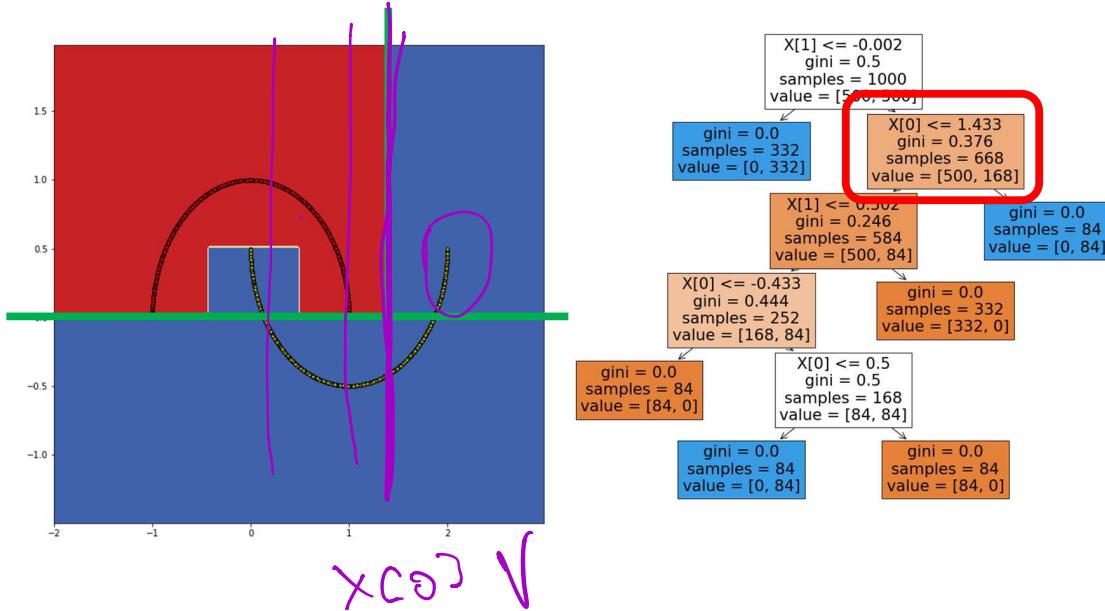
R_{root}



Решающее дерево



Решающее дерево

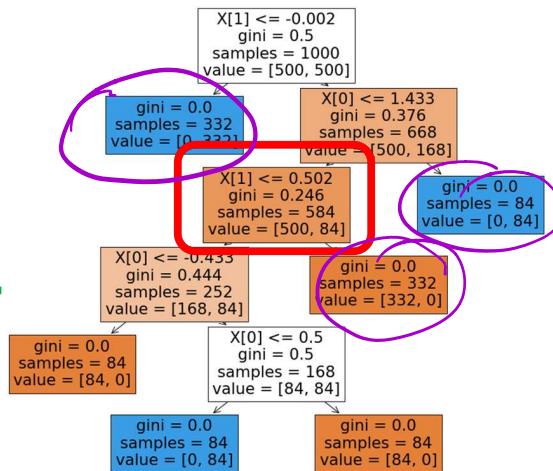
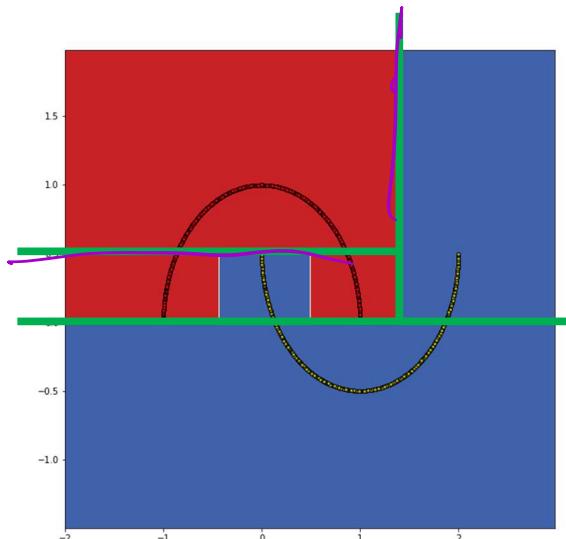


① как делают модели?

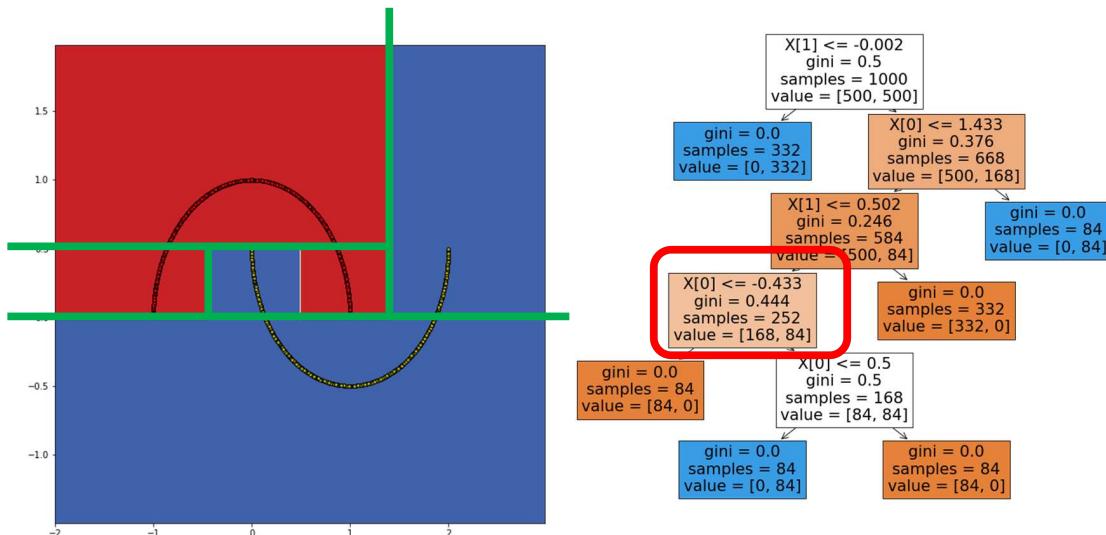
• Так, модель оценивает
объекты
одного класса

Решающее дерево

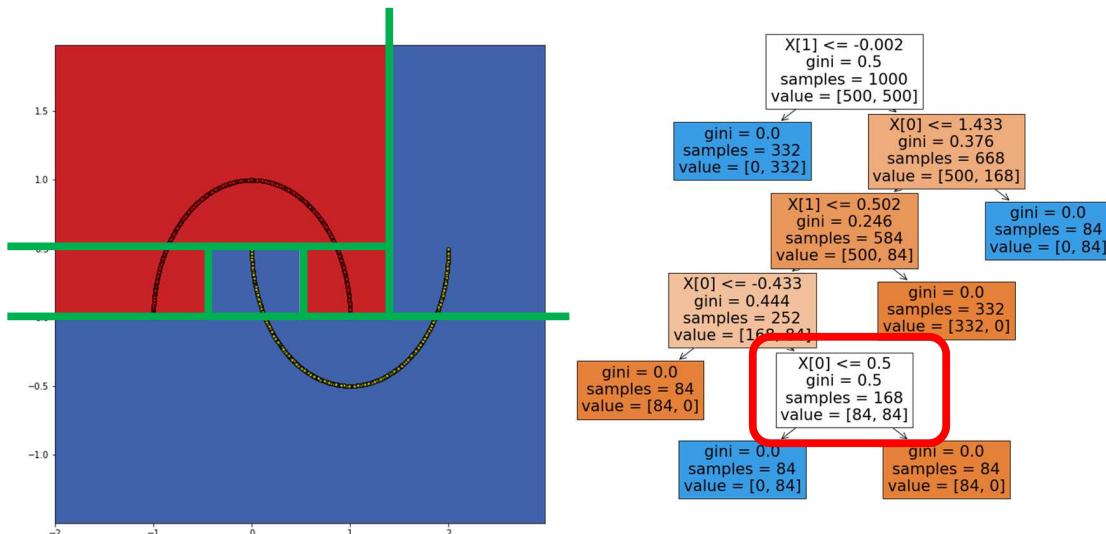
• $[X_j < t]$



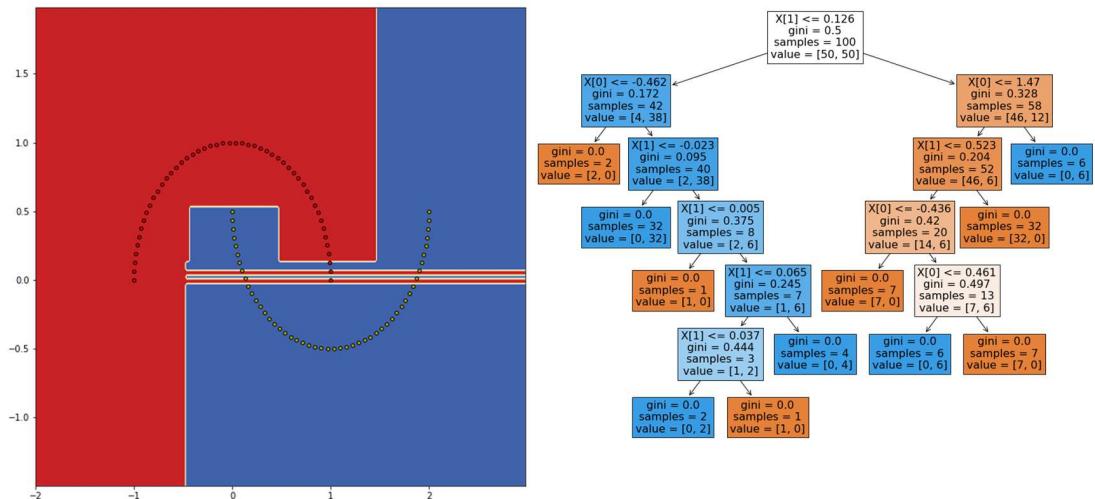
Решающее дерево



Решающее дерево



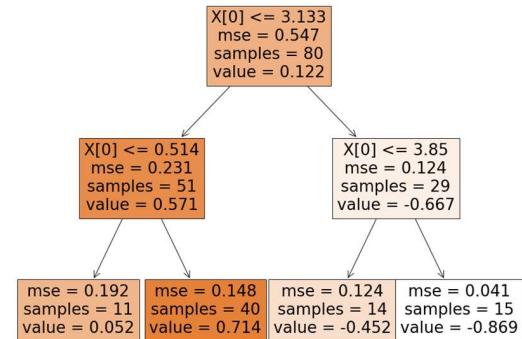
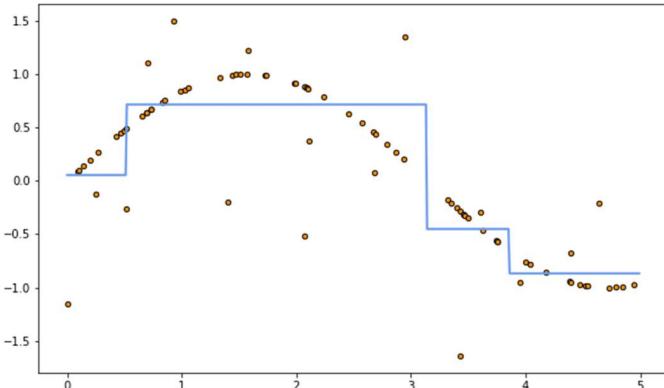
Решающее дерево



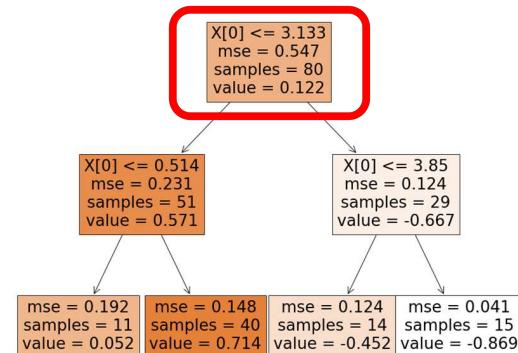
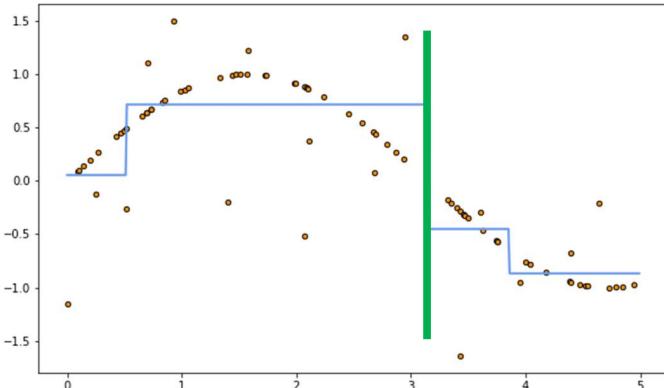
Сложность дерева

- Решающее дерево можно строить до тех пор, пока каждый лист не будет соответствовать ровно одному объекту
- Деревом можно идеально разделить любую выборку!
- Если только нет объектов с одинаковыми признаками, но разными ответами

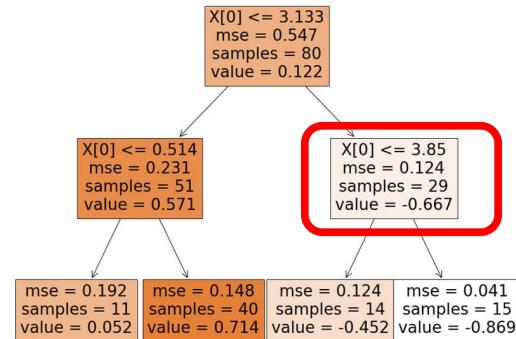
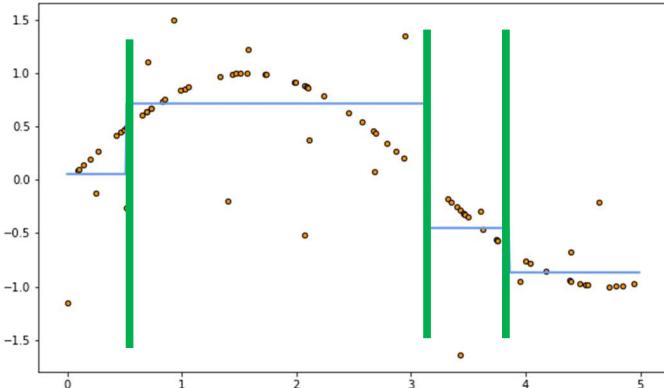
Решающее дерево для регрессии



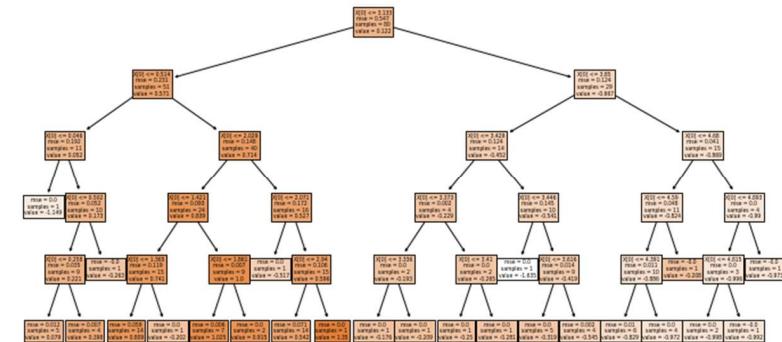
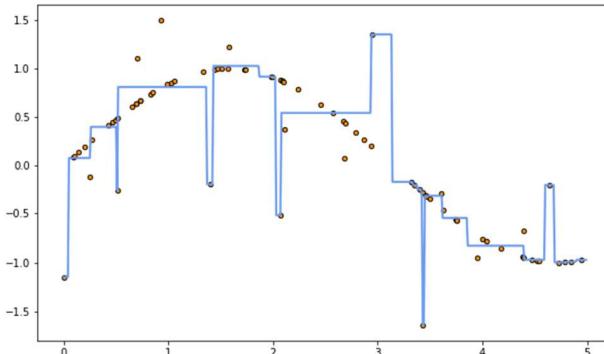
Решающее дерево для регрессии



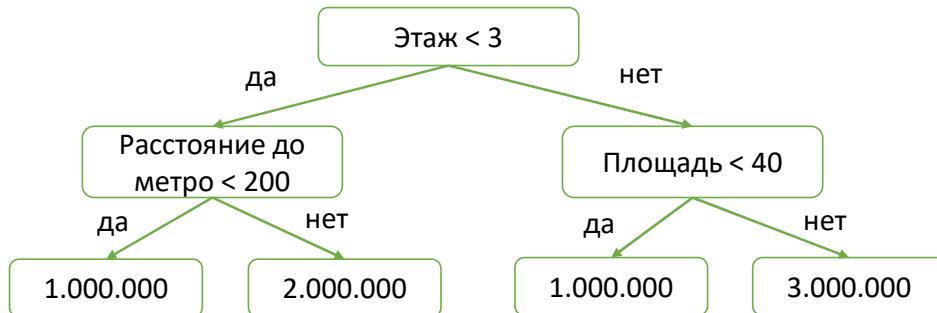
Решающее дерево для регрессии



Решающее дерево для регрессии



Решающее дерево



- Внутренние вершины: предикаты $[x_j < t]$
- Листья: прогнозы $c \in \mathbb{Y}$

Предикаты

- Порог на признак $[x_j < t]$ — не единственный вариант
- Предикат с линейной моделью: $[\langle w, x \rangle < t]$
- Предикат с метрикой: $[\rho(x, x_0) < t]$
- И много других вариантов
- Но даже с простейшим предикатом можно строить очень сложные модели

Прогнозы в листьях

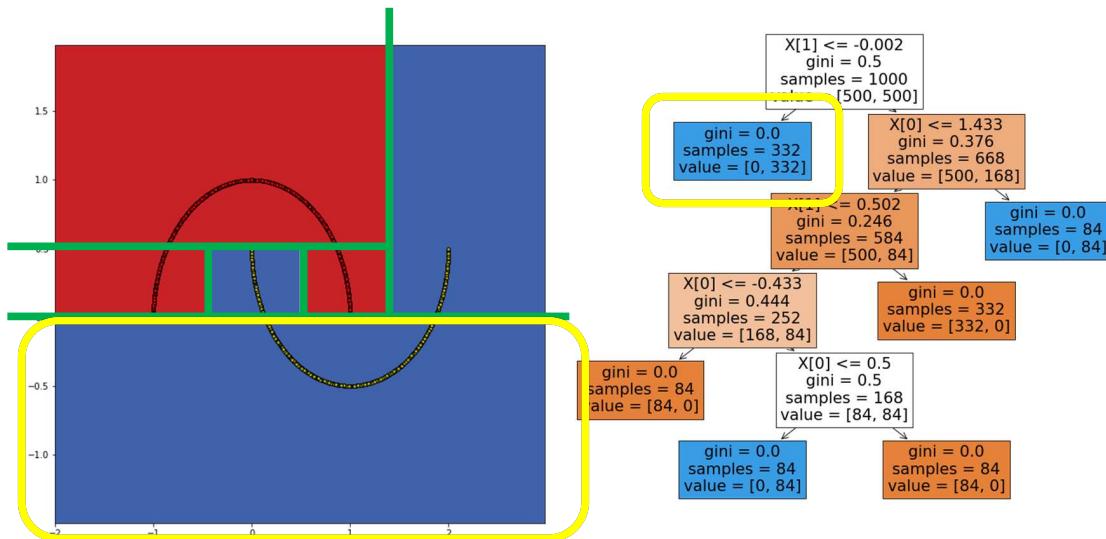
- Наш выбор: константные прогнозы $c_v \in \mathbb{Y}$
- Регрессия:

$$c_v = \frac{1}{|R_v|} \sum_{(x_i, y_i) \in R_v} y_i$$

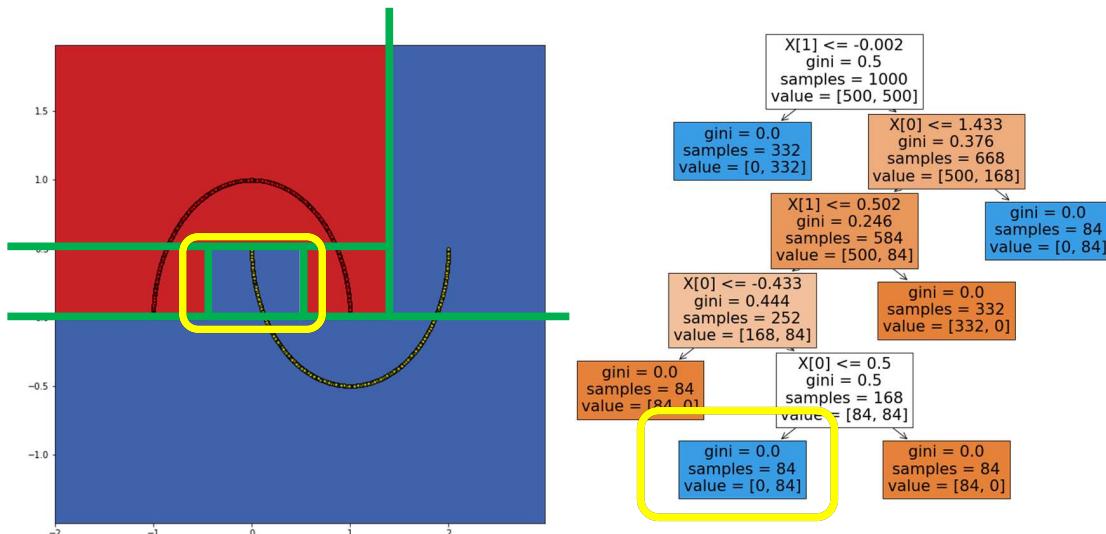
- Классификация:

$$c_v = \arg \max_{k \in \mathbb{Y}} \sum_{(x_i, y_i) \in R_v} [y_i = k]$$

Решающее дерево



Решающее дерево



Формула для дерева

- Дерево разбивает признаковое пространство на области R_1, \dots, R_J
- Каждая область R_j соответствует листу
- В области R_j прогноз c_j константный

$$a(x) = \sum_{j=1}^J c_j [x \in R_j]$$

Формула для дерева

$$a(x) = \sum_{j=1}^J c_j [x \in R_j]$$

- Решающее дерево находит хорошие новые признаки
- Над этими признаками подбирает линейную модель

Как выбирать предикаты

"Шаг за шагом"

не оптимальное решение

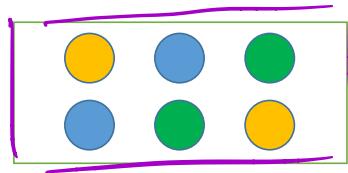
- "Жадное построение

- Разберёмся на примере
- Начнём с задачи классификации

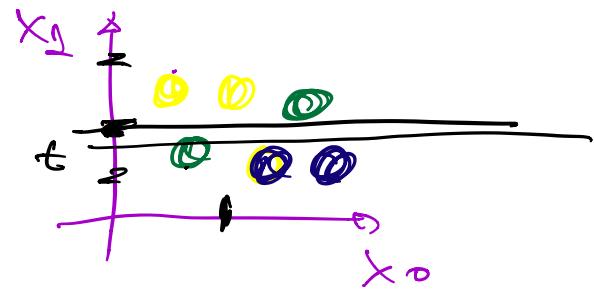
1) Быстро

2) не так плохо.

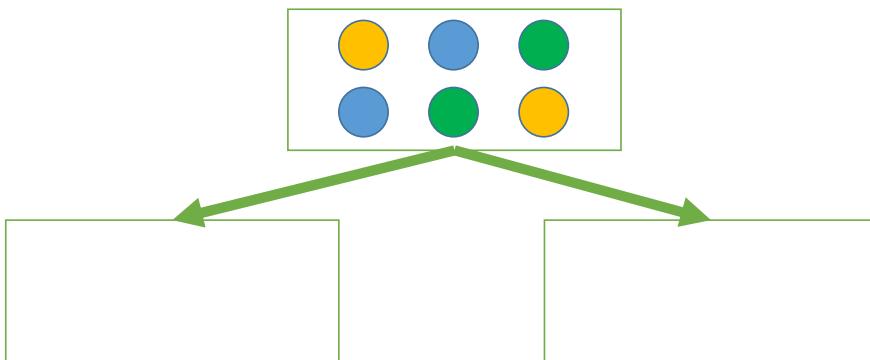
Жадное построение



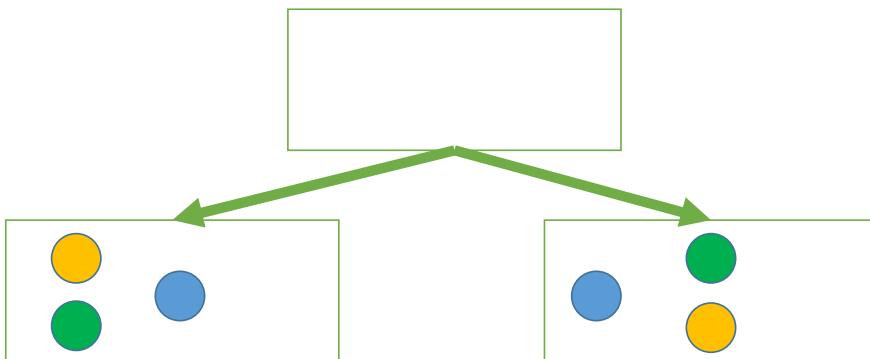
- Как разбить вершину?



Жадное построение

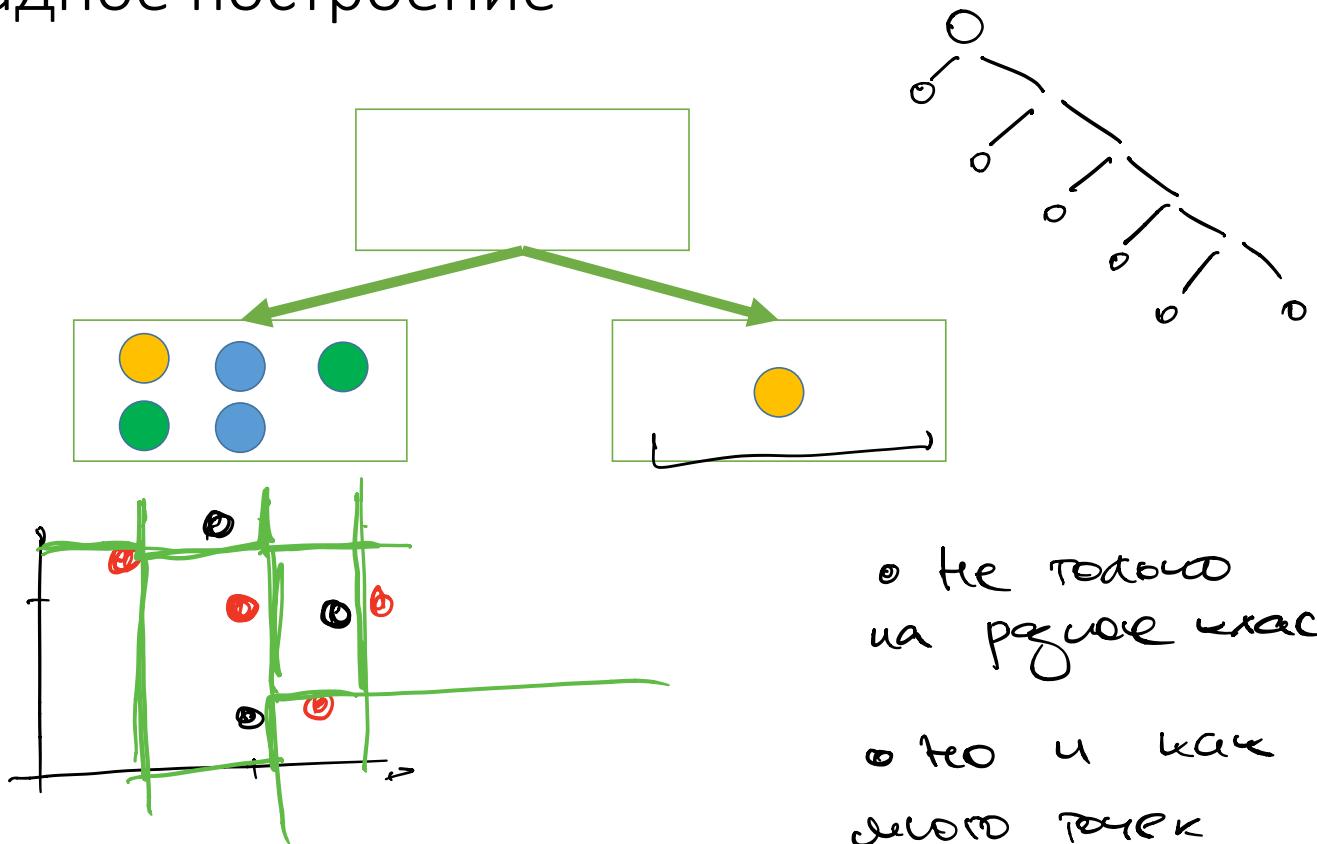


Жадное построение

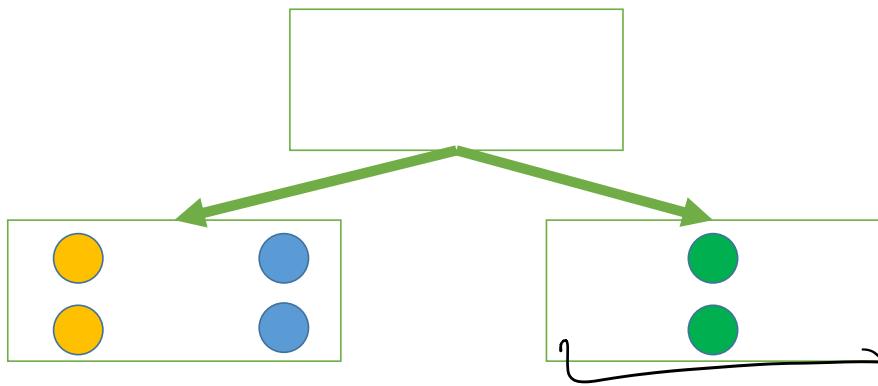


Почему это
не очень?

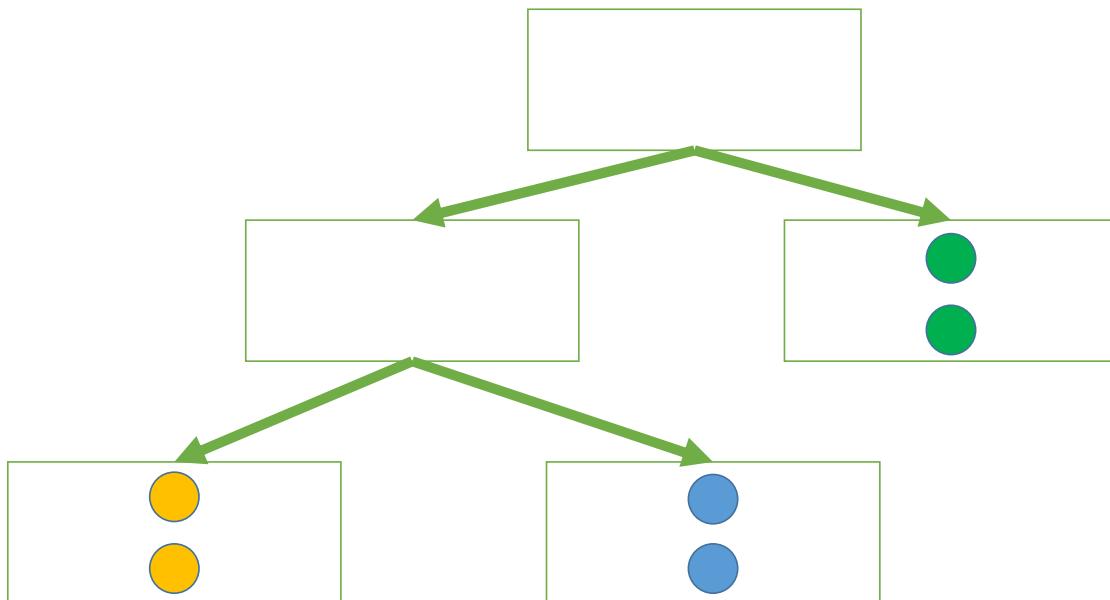
Жадное построение



Жадное построение

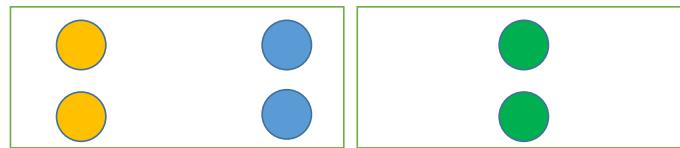


Жадное построение



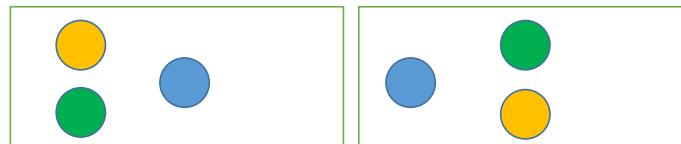
Как сравнить разбиения?

•



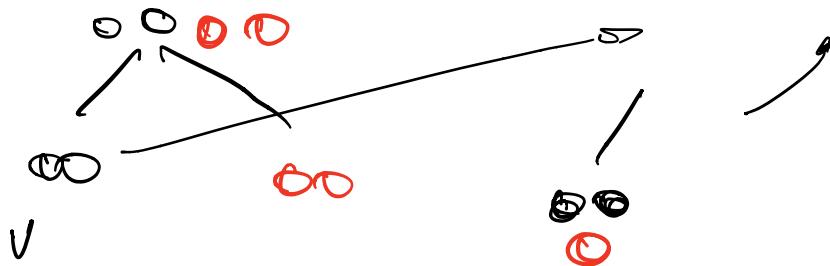
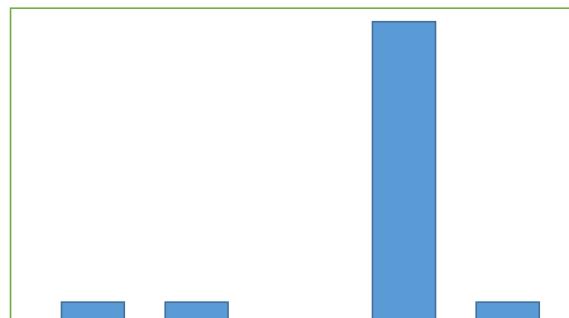
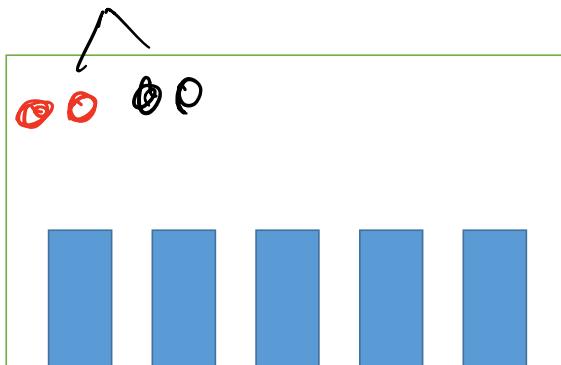
или

•



{ Энтропия }

- Мера неопределённости распределения



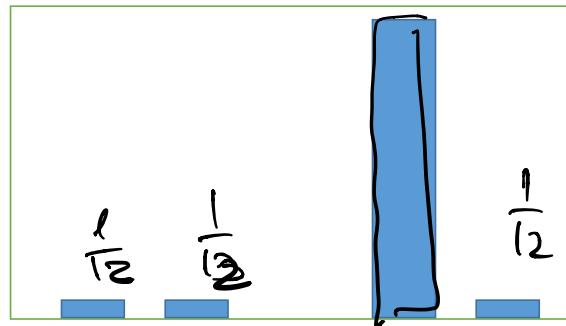
Энтропия

- Мера неопределённости распределения

$$\frac{3}{4}$$



Высокая энтропия



Низкая энтропия

Энтропия

- Дискретное распределение
- Принимает n значений с вероятностями p_1, \dots, p_n
- Энтропия:

$$\underbrace{H(p_1, \dots, p_n)} = - \sum_{i=1}^n p_i \log p_i$$

исходов у исхода ε $= p_\varepsilon$
ВСТО
исходов

Энтропия

- $\underbrace{(0.2, 0.2, 0.2, 0.2, 0.2)}_{H = 1.60944 \dots}$

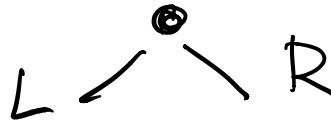
$$- 6 \times 0.2 \cdot \log_2 0.2$$

$$- 1.2 \cdot \underbrace{\log_2 0.2}_{\approx 0}$$

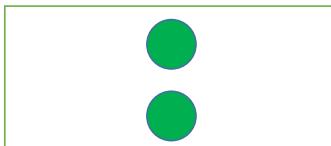
- $(\overbrace{0.9, 0.05, 0.05}^V, 0, 0)$
- $H = \underbrace{0.394398 \dots}_{0.1}$

$$+ f \cdot \log_2 f = 0.$$

- $(0, 0, 0, 1, 0)$
- $H = 0 \quad \checkmark$



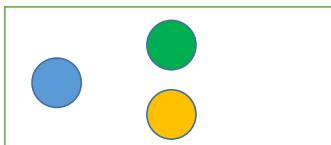
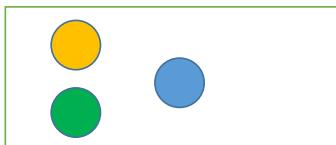
Как сравнить разбиения?



0.693

0

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



1.09

1.09

- $(0.33, 0.33, 0.33)$ 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

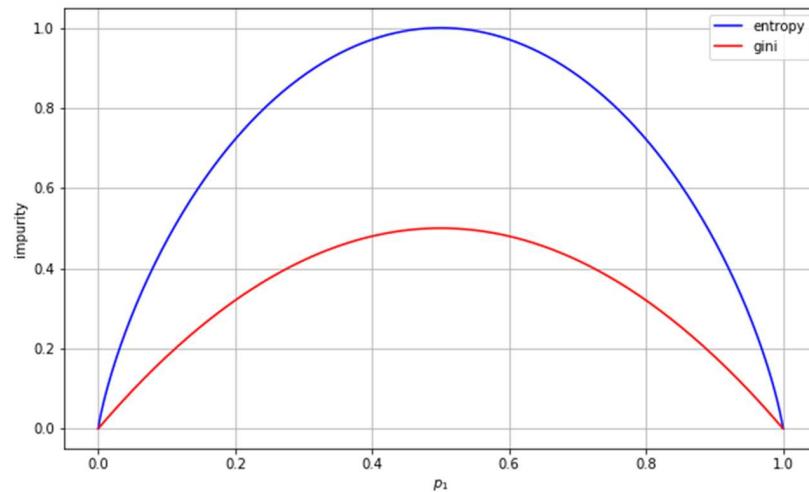
Энтропия — $\sum p \log p$

Критерий Джини | . гини
| . entropy

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

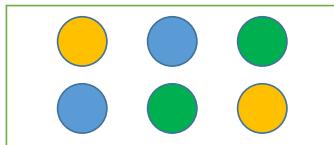
- Вероятность ошибки случайного классификатора, который выдаёт класс k с вероятностью p_k

Критерии качества вершины

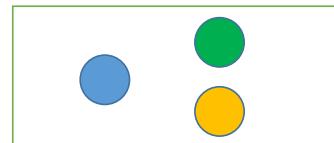
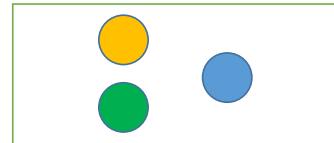


Критерий информативности

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

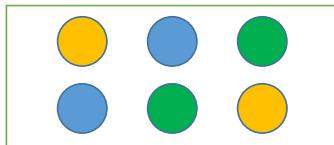


против

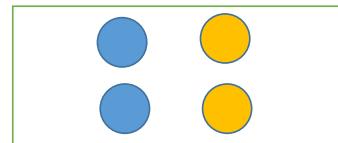
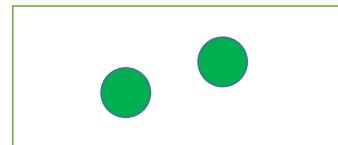


Критерий информативности

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



против



Критерий информативности

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

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

Критерий информативности

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

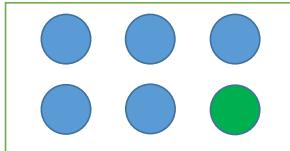
$$Q(R, j, t) = H(R) - H(R_\ell) - H(R_r) \rightarrow \max_{j,t}$$

- Или так:

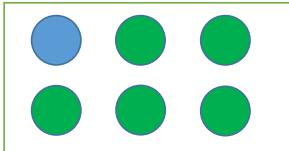
$$Q(R, j, t) = H(R_\ell) + H(R_r) \rightarrow \min_{j,t}$$

- (у этих формул есть проблемы!)

Как сравнить разбиения?

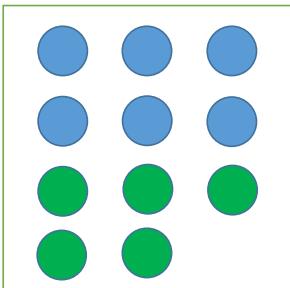


0.65



0.65

- $(5/6, 1/6)$ и $(1/6, 5/6)$
- $0.65 + 0.65 = 1.3$



0.994



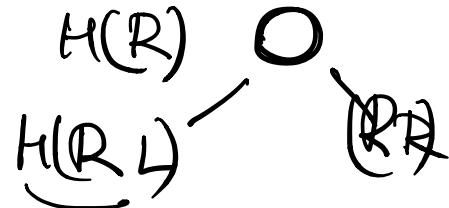
0

- $(6/11, 5/11)$ и $(0, 1)$
- $0.994 + 0 = 0.994$

1. Перебор по j

2. Перебор по $\frac{H(R)}{R}$

$Q(j,t)$ оцнк.



Критерий информативности

$$Q(R, j, t) = H(R) \left[\underbrace{\frac{|R_\ell|}{|R|} H(R_\ell) + \frac{|R_r|}{|R|} H(R_r)}_{\# \text{число правл}} \right] \rightarrow \max_{j,t}$$

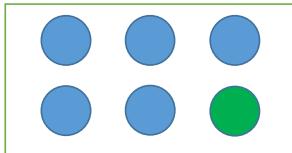
- Или так:

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

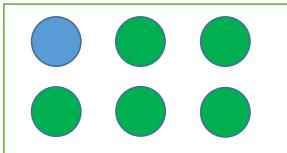
$\overset{j}{\circ}$ обратно t $x_1^j \quad x_2^j \quad x_3^j$ \bullet гистограмма
отсортируй \rightarrow $x_2^j < x_3^j < x_1^j$

12 разные

Как сравнить разбиения?



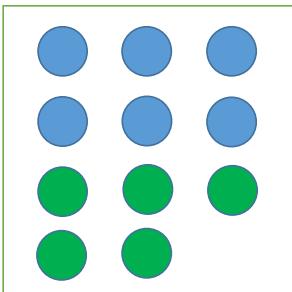
0.65



0.65

с одинаковыми
значениями

- $(5/6, 1/6)$ и $(1/6, 5/6)$
- $0.5 * 0.65 + 0.5 * 0.65 = \underline{0.65}$



0.994

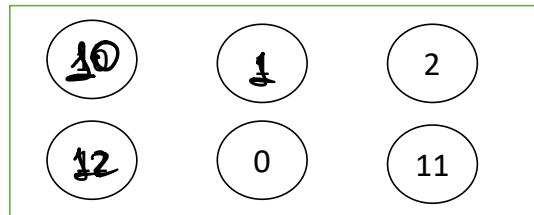


0

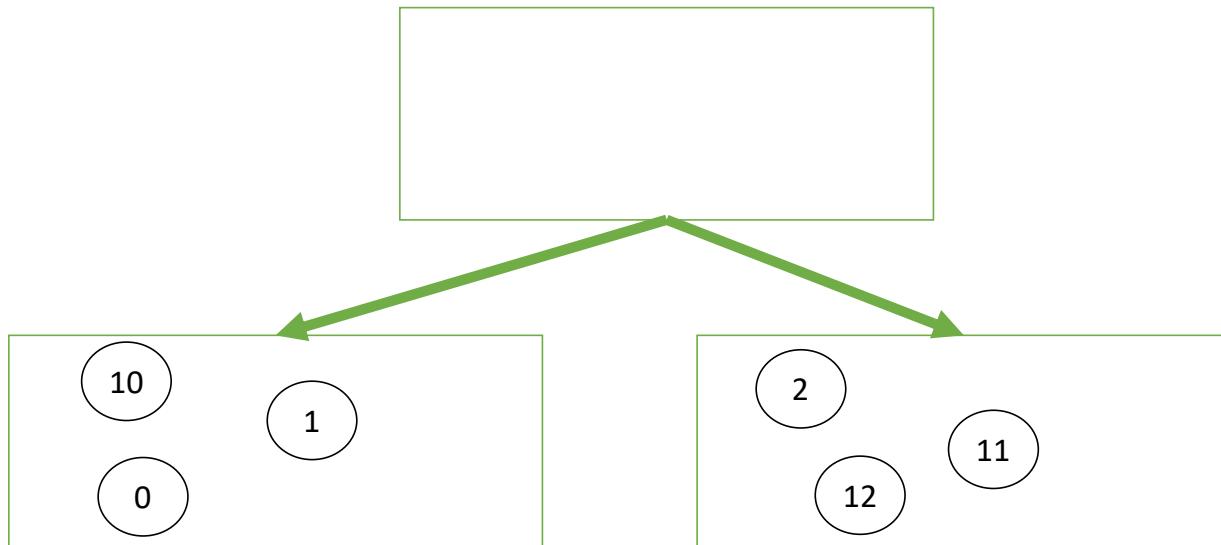
- $(6/11, 5/11)$ и $(0, 1)$
- $\frac{11}{12} * 0.994 + \frac{1}{12} * 0 = 0.911$

\bar{y}

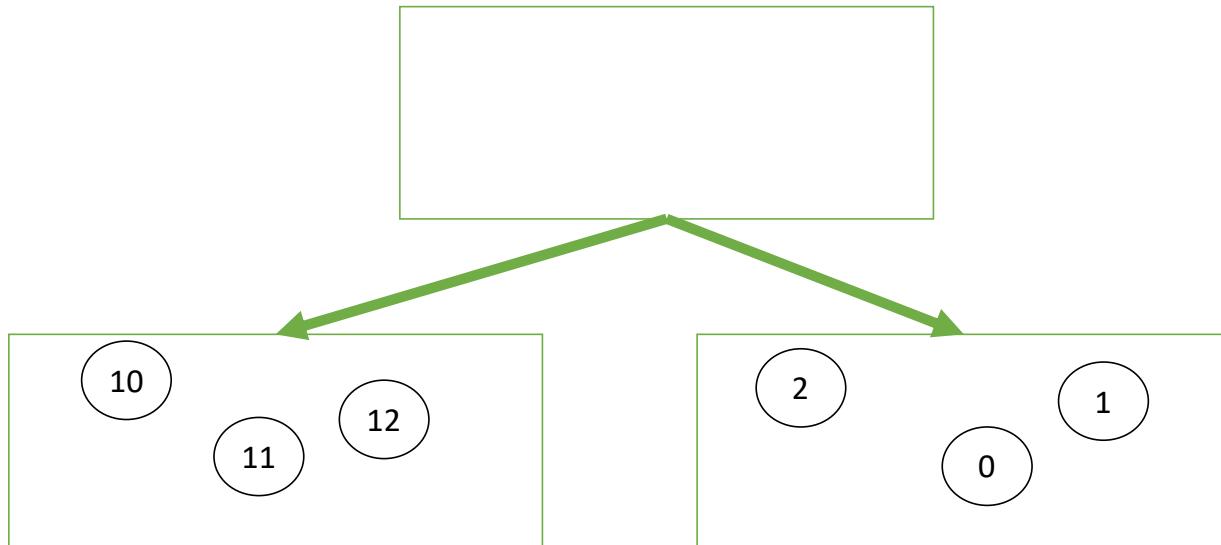
А для регрессии?



А для регрессии?



А для регрессии?



$$D(R) = \frac{1}{N} \sum_{i=1}^n (y_i - \bar{y})^2$$

Задача регрессии

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

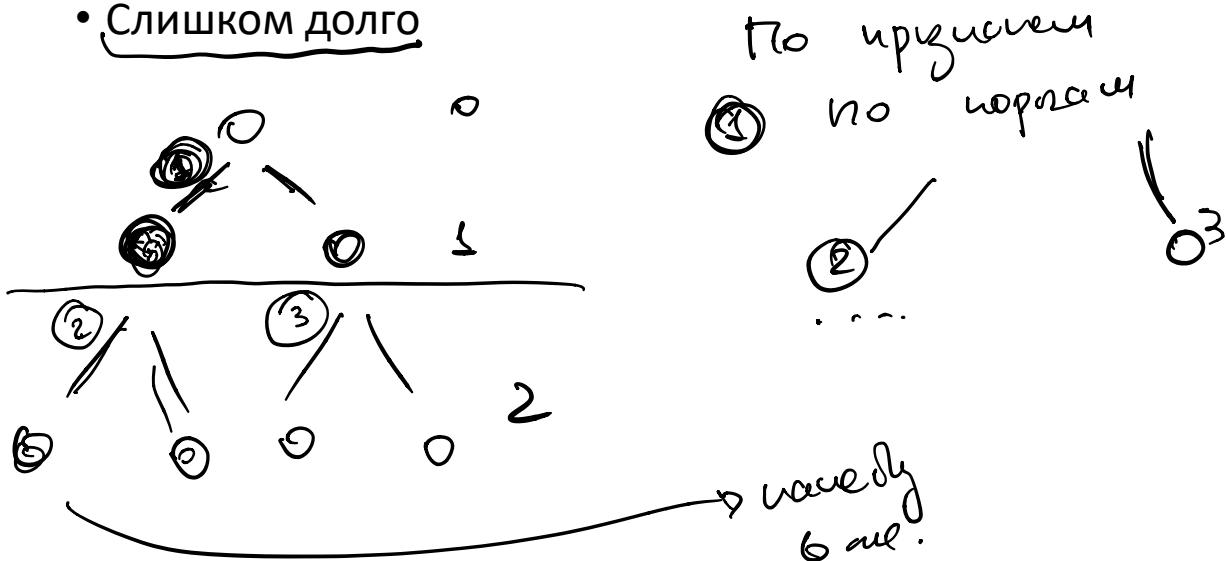
$$\bar{y}_R = \frac{1}{|R|} \underbrace{\sum_{(x_i, y_i) \in R} y_i}_{\text{---}}$$

- 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

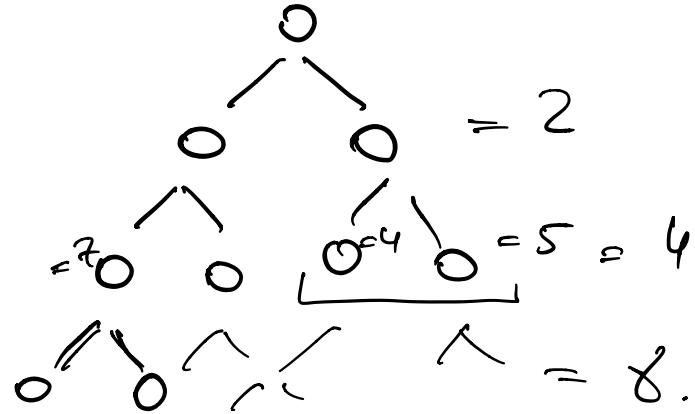


Как строить дерево?

- 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

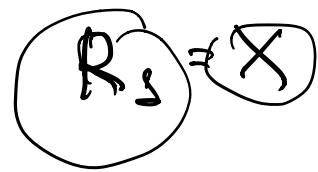
Критерий останова

- Ограничить глубину
- Ограничить количество листьев
- Задать минимальное число объектов в вершине = 5
- Задать минимальное уменьшение хаотичности при разбиении
- И так далее



листов < 8 \Rightarrow либо 3 групп
= 0.5

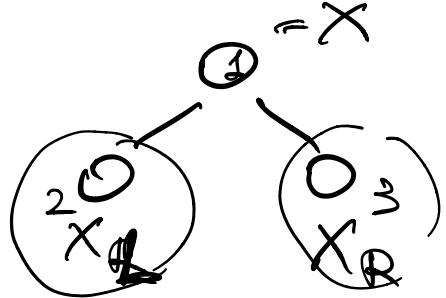
$$H(R) + H(R_e) + H(R_c) \leq \epsilon.$$



Жадный алгоритм

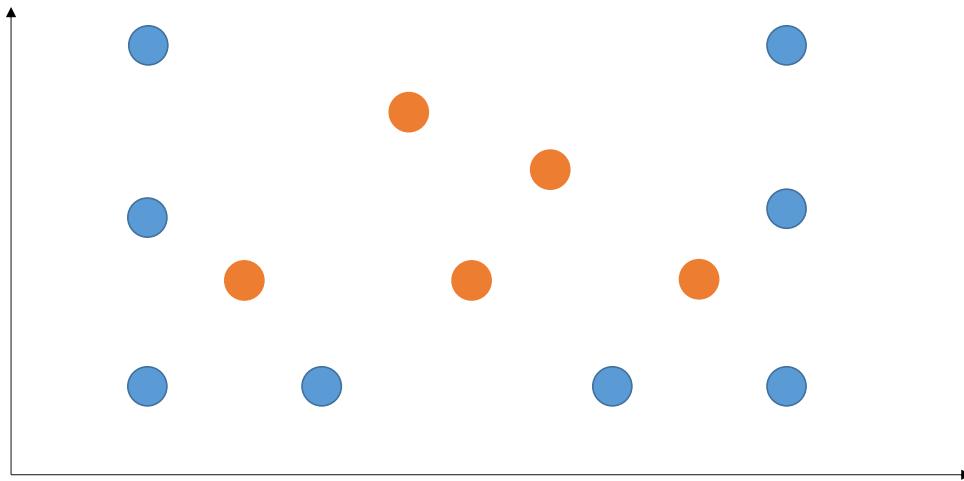
- Поместить в корень всю выборку: $R_1 = X$
- Запустить построение из корня: $\text{SplitNode}(1, R_1)$

Жадный алгоритм

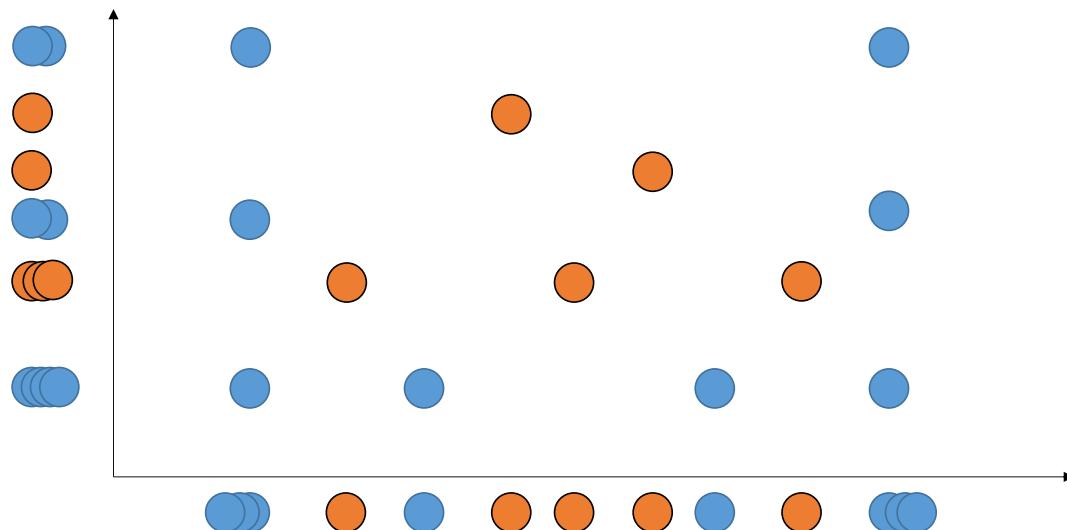


- $\text{SplitNode}(m, R_m)$
- Если выполнен критерий останова, то выход
- Ищем лучший предикат: $j, t = \arg \min_{j,t} Q(R_m, j, t)$
- Разбиваем с его помощью объекты: $R_\ell = \left\{ (x, y) \in R_m \mid [x_j < t] \right\}$,
 $R_r = \left\{ (x, y) \in R_m \mid [x_j \geq t] \right\}$
- Повторяем для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

Обучение деревьев

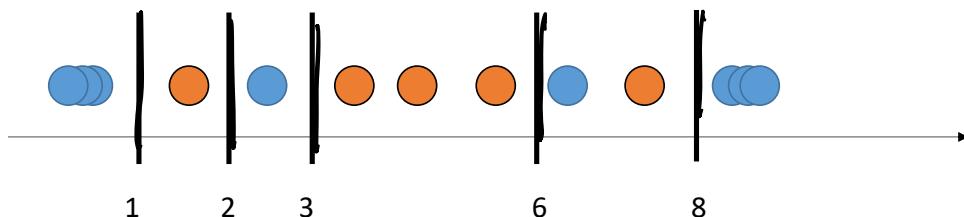


Признаки

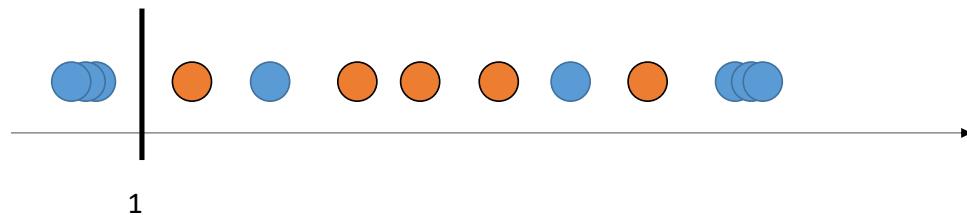


не беे тeR

Разбиения по признаку 1



Разбиения по признаку 1

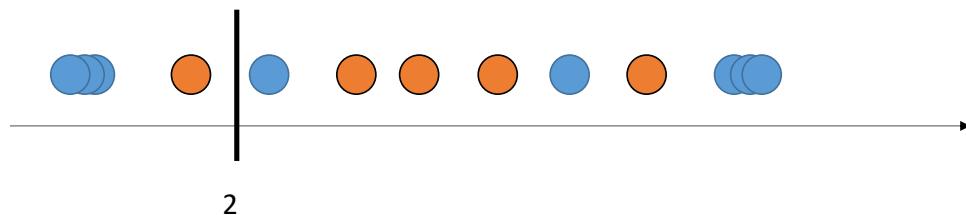


(1, 0)
 $H(p) = 0$

(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

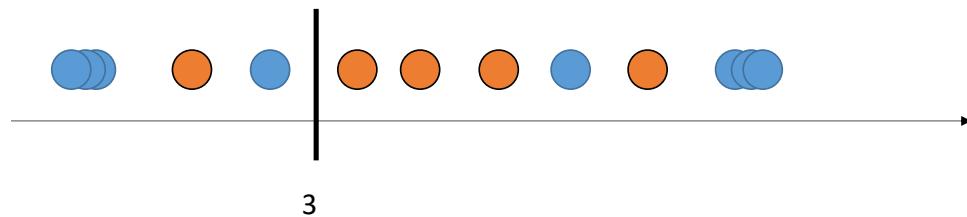


$$(3/4, 1/4)$$
$$H(p) = 0.56$$

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

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

Разбиения по признаку 1

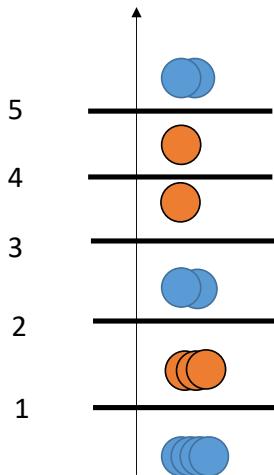


$(4/5, 1/5)$
 $H(p) = 0.5$

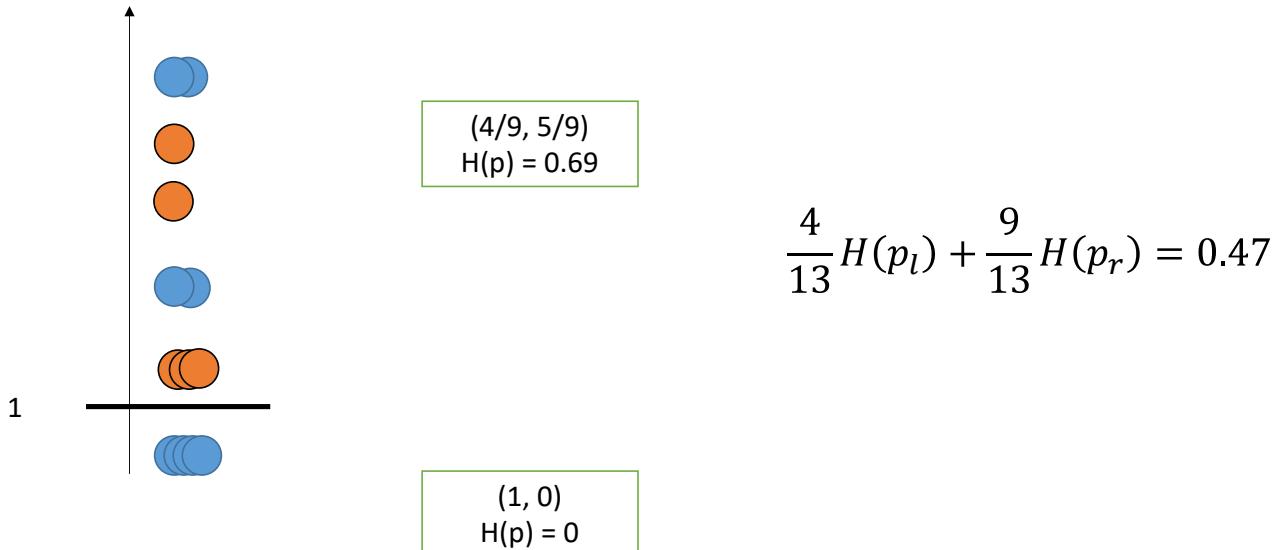
$(1/2, 1/2)$
 $H(p) = 0.69$

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

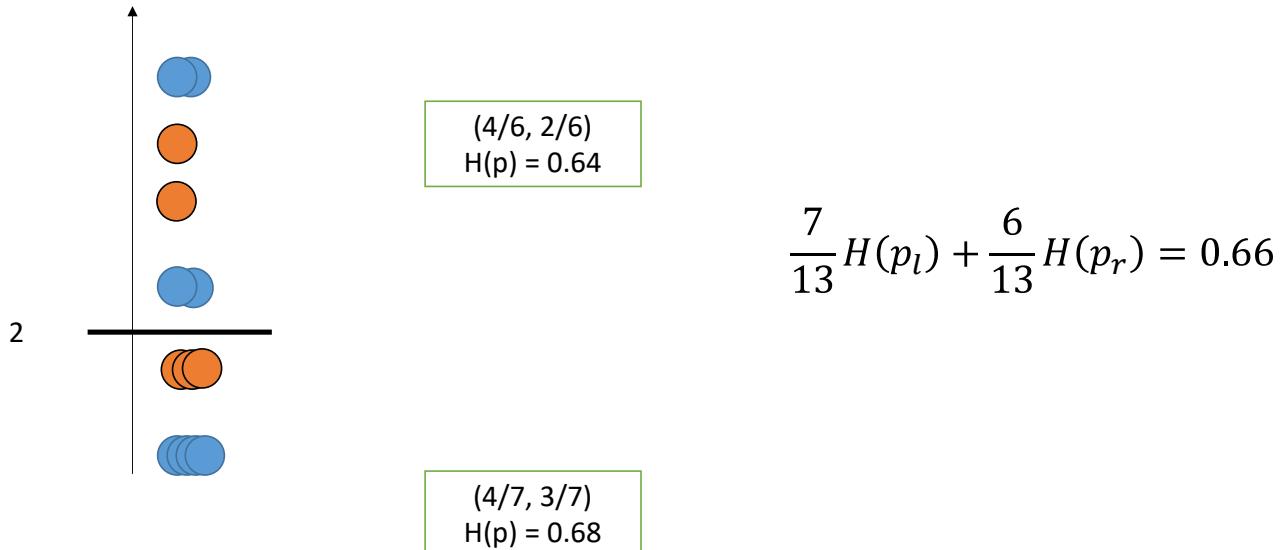
Разбиения по признаку 2



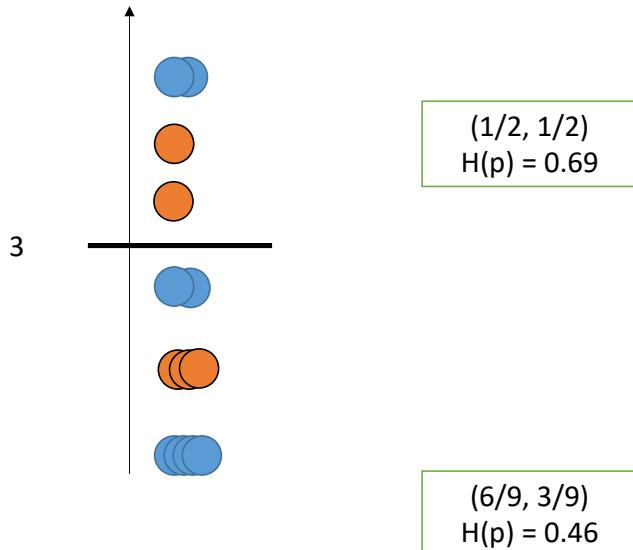
Разбиения по признаку 2



Разбиения по признаку 2

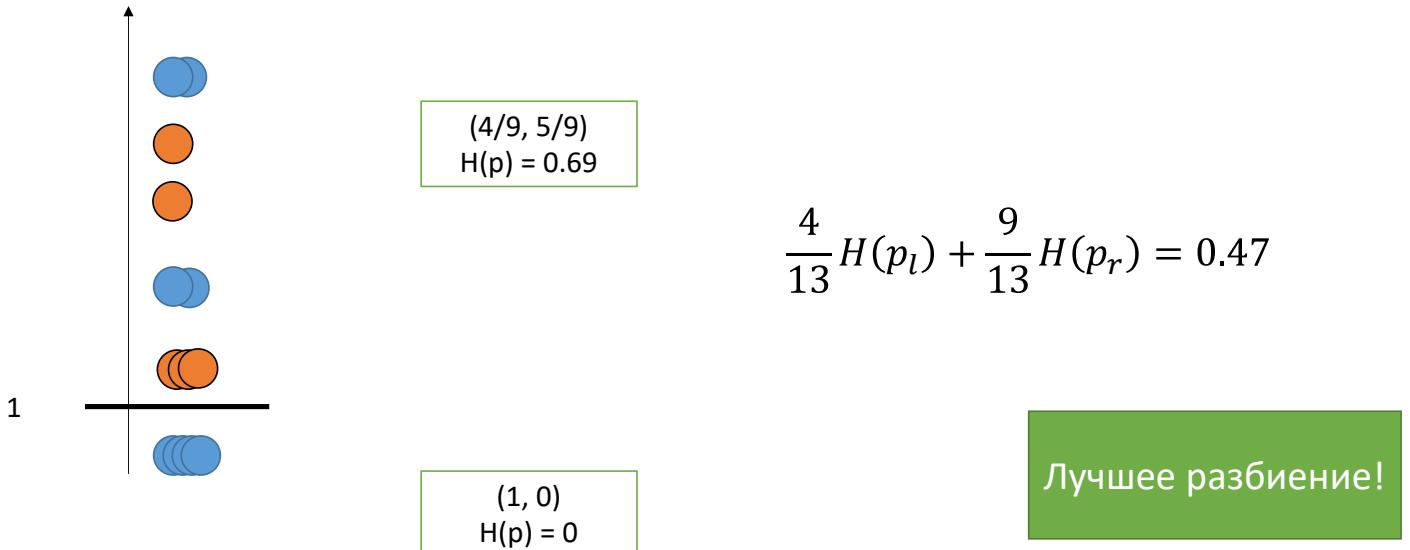


Разбиения по признаку 2

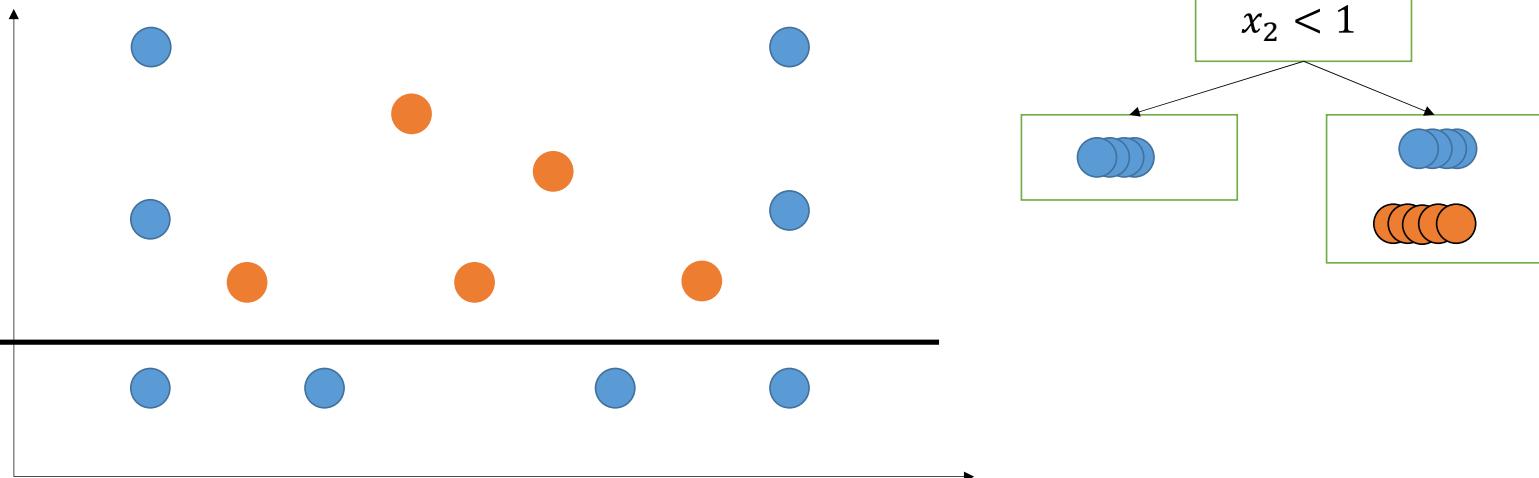


$$\frac{9}{13}H(p_l) + \frac{4}{13}H(p_r) = 0.53$$

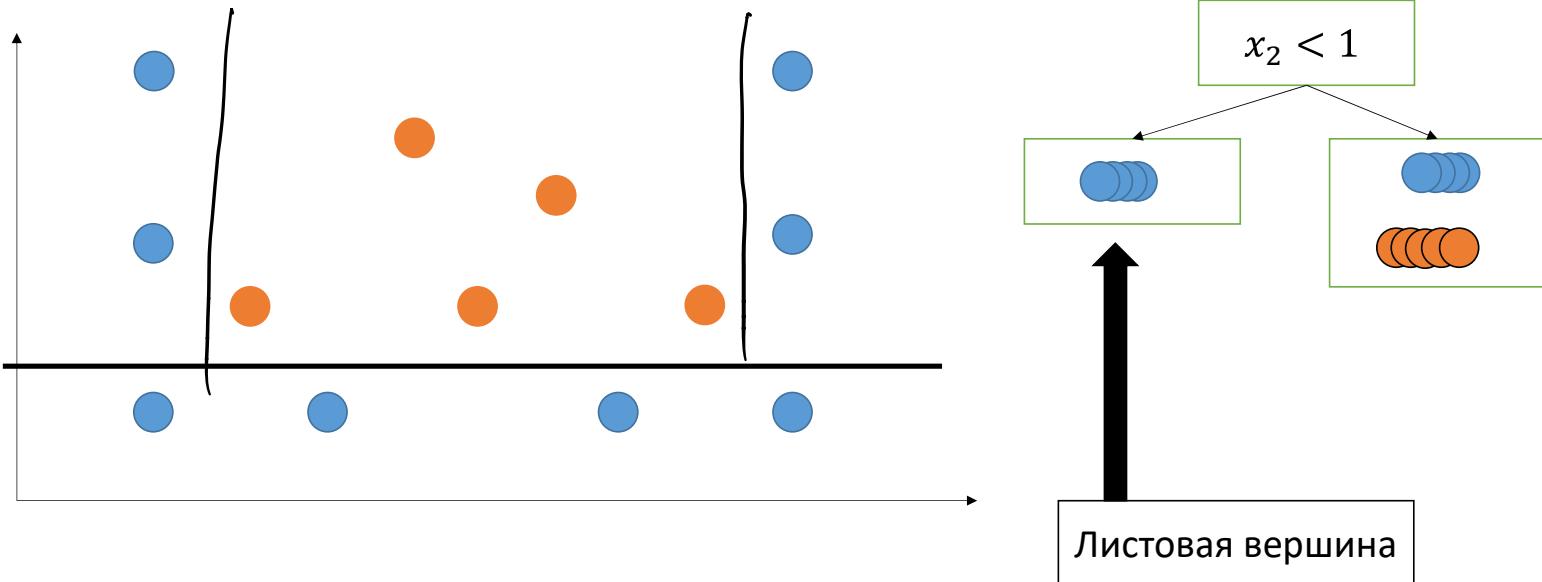
Разбиения по признаку 2



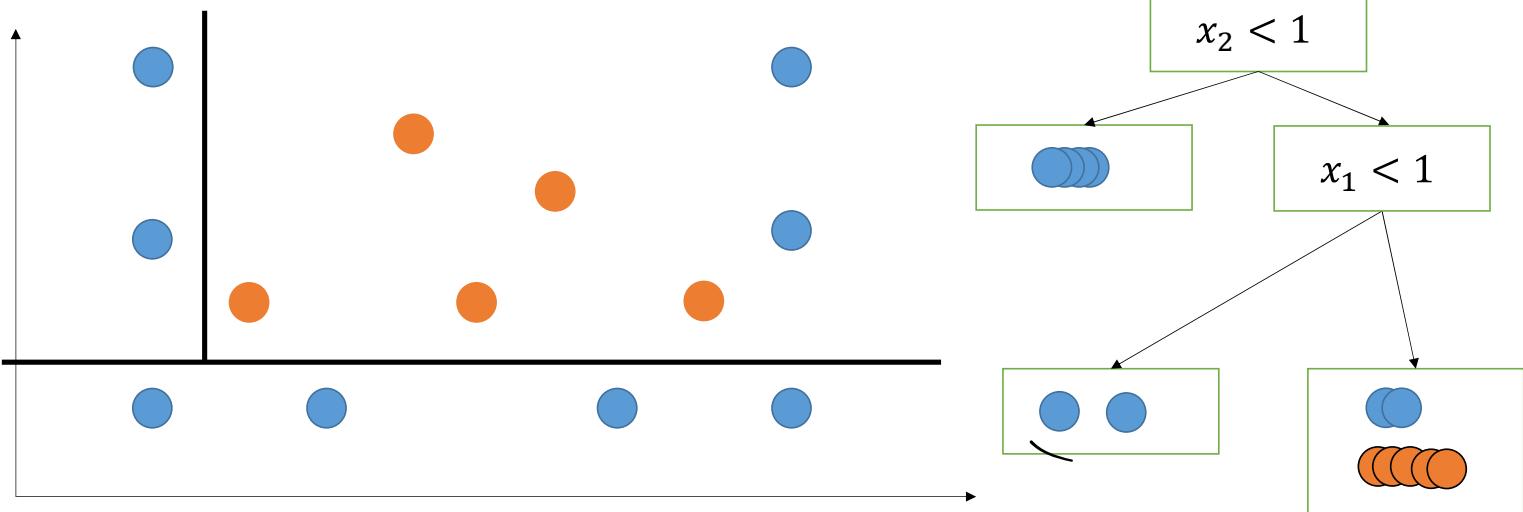
Обучение деревьев



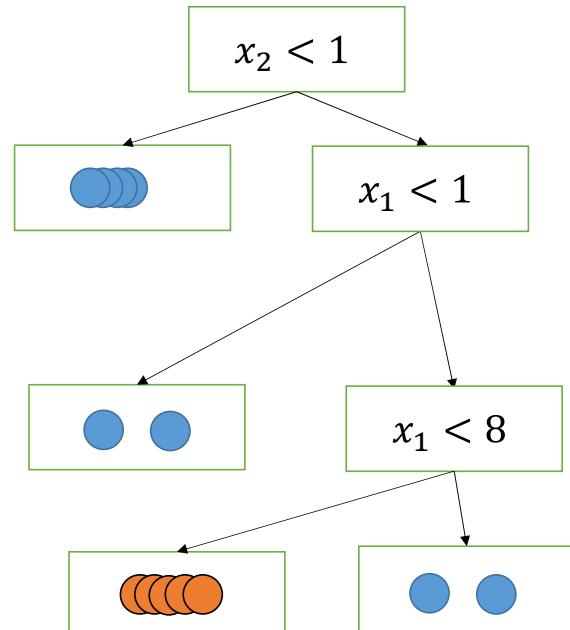
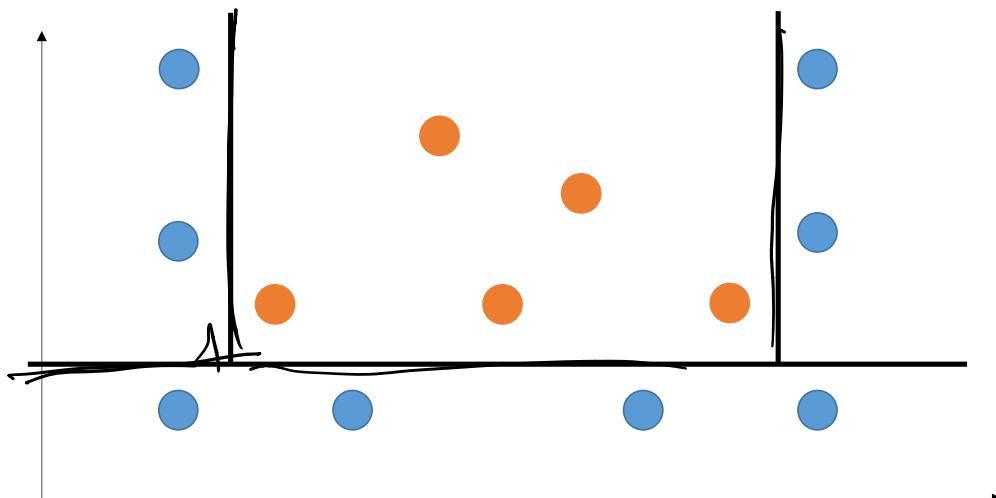
Обучение деревьев



Обучение деревьев

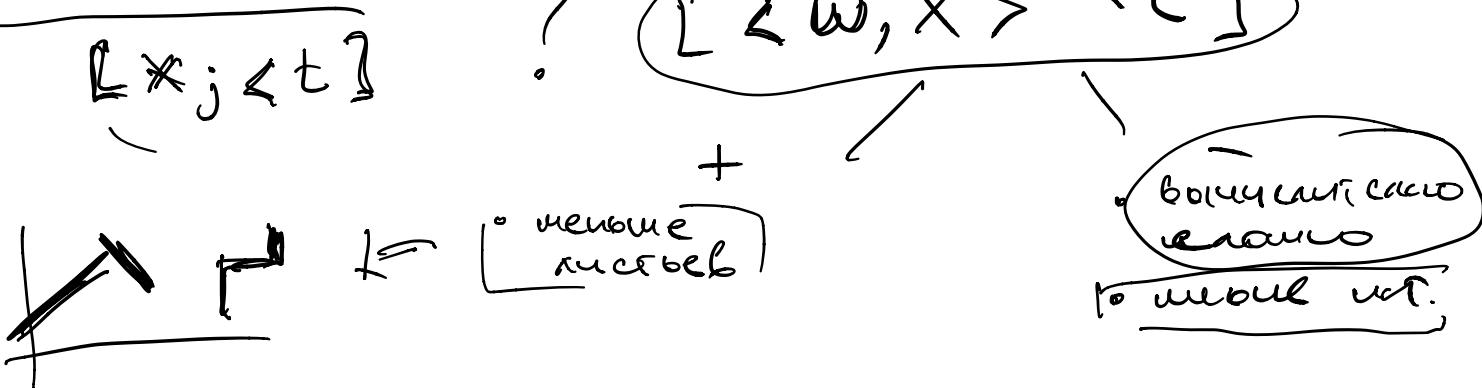


Обучение деревьев



Резюме

- Решающие деревья позволяют строить сложные модели, но есть приск переобучения
- Деревья строятся жадно, на каждом шаге вершина разбивается на две с помощью лучшего из предиктов $\{x_j < t\}$
- Алгоритм довольно сложный и требует перебора всех предиктов на каждом шаге |



Алгоритм «~~сокращений~~⁴» $\{X_j < t\}$

- переход по признакам (j)
- переход по возрастанию (t)



② запуск
значе

① + ②

- база подразумевает

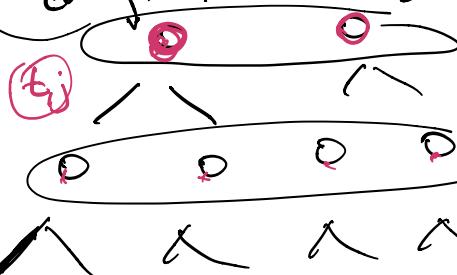
$$N = 10^5$$

• переход по j

переход по признакам (j)

① \Rightarrow связанные подразумеваются

② \Rightarrow связанные подразумеваются



def split(X^l, y^l):
for i in range(X^l .shape[1]):
for j in range(X^l .shape[0]):

$$\bar{t}_j = \text{mean}(X^l[j < t])$$

$$\bar{t}_{j+} = \text{mean}(X^l[j \geq t])$$

$$\text{не так } z = (\text{Var}(y^l[X^l[j < t]]) - \text{Var}(y^l[X^l[j \geq t]]))$$

if $z < z_{-P}$:

$z_{-P} = \dots$
else
return j

\Rightarrow • требуется очень многое \Leftrightarrow неэффективно.

Decision Tree



Oblivious Tree

Extreme Randomized

... .



① Нес, не делают

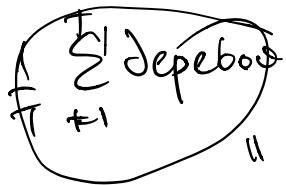


① Делают
но фиг., хорол.



1. Делают неподгружен

2. Аксессуары



"weak learners"

