Деревья решений
Критерии информативности

Паточенко Евгений

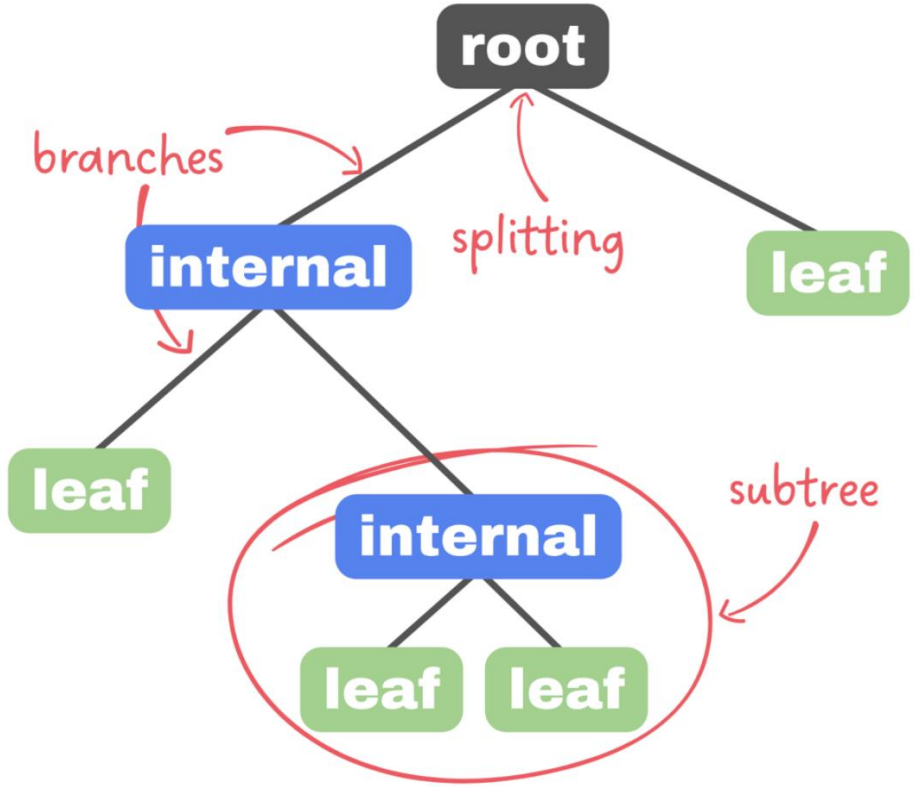
НИУ ВШЭ

План занятия

- Деревья решений
- Критерии информативности
- Критерии останова
- Стрижка дерева

Деревья решений

- Семейство моделей, которые позволяют восстанавливать нелинейные зависимости произвольной сложности
- Могут использоваться как для классификации, так и для регрессии
- Строит прогноз, следуя иерархии простых условий (предикатов) от корня к листьям
- Легко интерпретируются, поскольку процесс структурно схож с человеческой логикой принятия решений



Источник: https://mljar.com/glossary/decision-trees/

Деревья решений

Каждой вершине v приписана функция (предикат) $\beta_v: X \to \{0,1\}$

Каждой листовой вершине v приписан прогноз $c_v \in Y$ (для классификации – класс или вероятность класса, для регрессии – действительное значение целевой переменной)

Деревья решений

Жадный алгоритм построения

1 шаг

- найдем наилучшее разбиение всей выборки X на две части: $R_1(j,t) = \{x \mid x_j < t\}$ и $R_2(j,t) = \{x \mid x_j \geq t\}$ с точки зрения некоторого функционала $Q(X,j,t)$:
- найдем наилучшие j и t
- создадим корень дерева, поставив в него предикат $[x_j < t]$.

Деревья решений

Жадный алгоритм построения

1 шаг

- найдем наилучшее разбиение всей выборки X на две части: $R_1(j,t) = \{x \mid x_j < t\}$ и $R_2(j,t) = \{x \mid x_j \geq t\}$ с точки зрения некоторого функционала $Q(X,j,t)$:
- найдем наилучшие j и t
- создадим корень дерева, поставив в него предикат $[x_j < t]$.

2 шаг

Для каждой из полученных подвыборок R_1 и R_2 рекурсивно применим шаг 1.

В каждой вершине на каждом шаге проверяем, не выполнилось ли условие останова.

Деревья решений

Жадный алгоритм построения

1 шаг

- найдем наилучшее разбиение всей выборки X на две части: $R_1(j,t) = \{x \mid x_j < t\}$ и $R_2(j,t) = \{x \mid x_j \geq t\}$ с точки зрения некоторого функционала $Q(X,j,t)$:
- найдем наилучшие j и t
- создадим корень дерева, поставив в него предикат $[x_j < t]$.

2 шаг

Для каждой из полученных подвыборок R_1 и R_2 рекурсивно применим шаг 1.

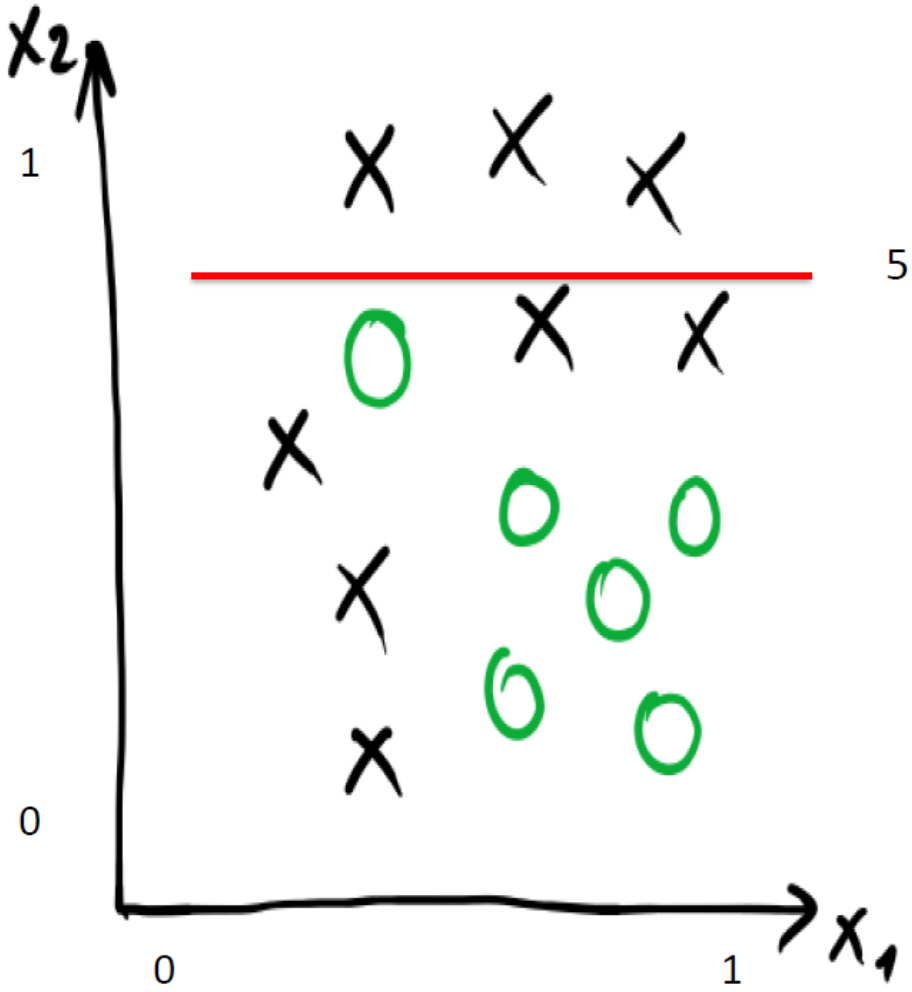
В каждой вершине на каждом шаге проверяем, не выполнилось ли условие останова.

Если выполнилось, то объявляем вершину листом и записываем в него предсказание.

Деревья решений

Пример

Найдем лучший предикат



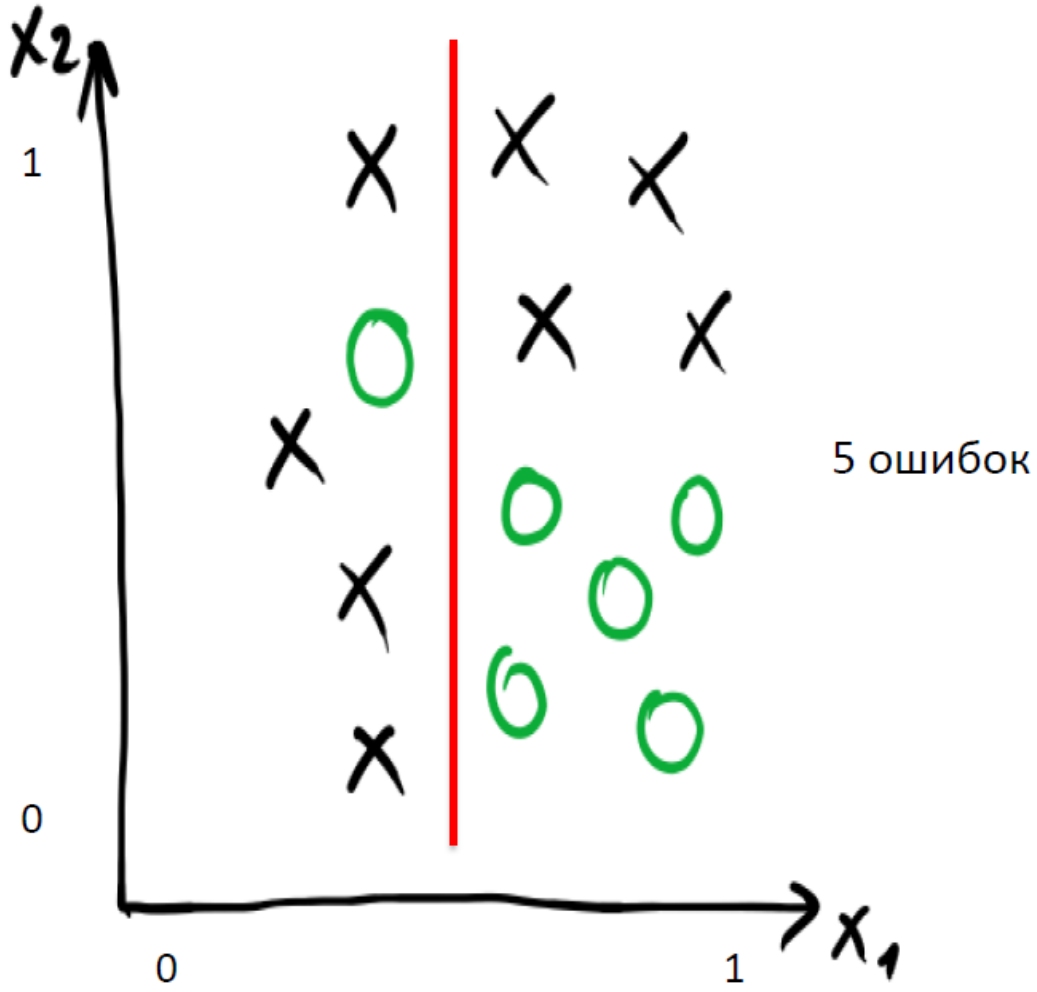
X2 > 0.8

5 ошибок

Деревья решений

Пример

Найдем лучший предикат



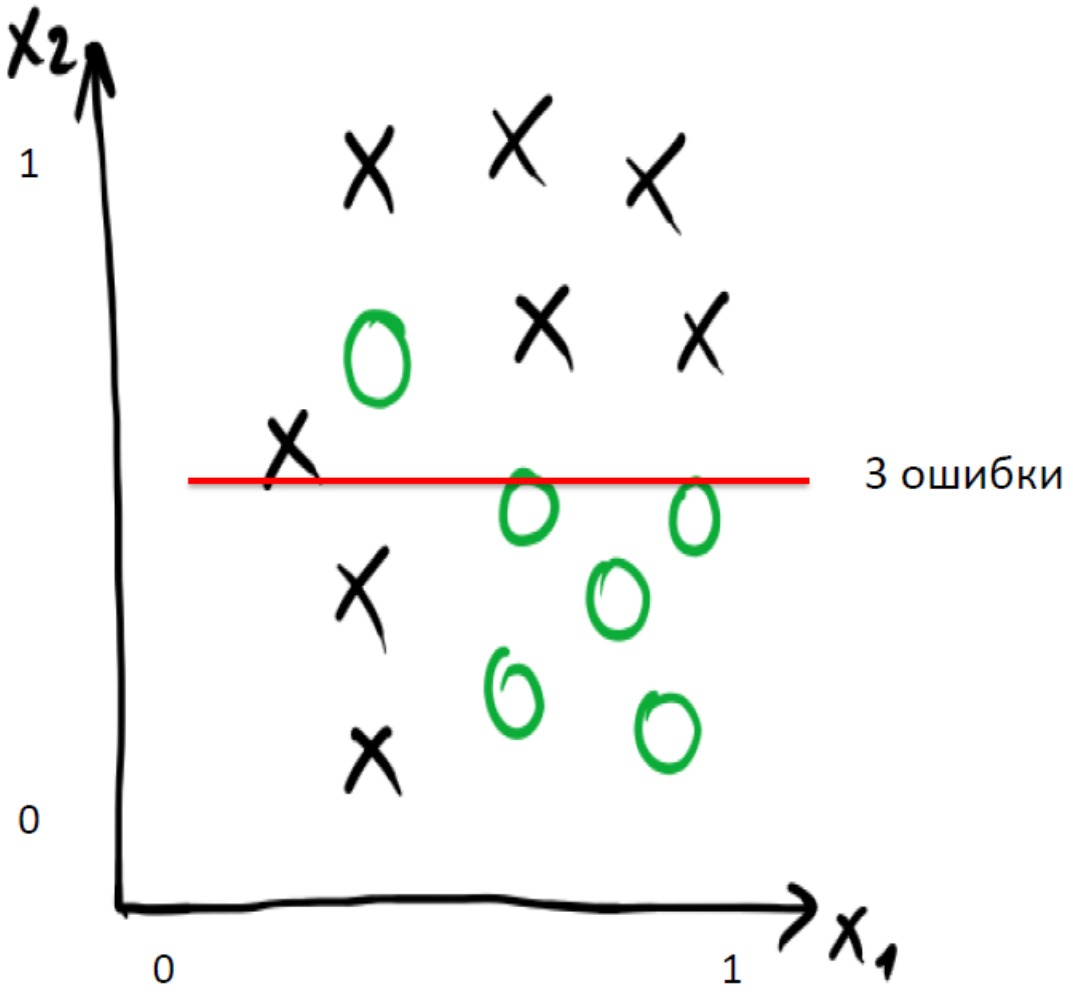
X1 > 0.5

5 ошибок

Деревья решений

Пример

Найдем лучший предикат

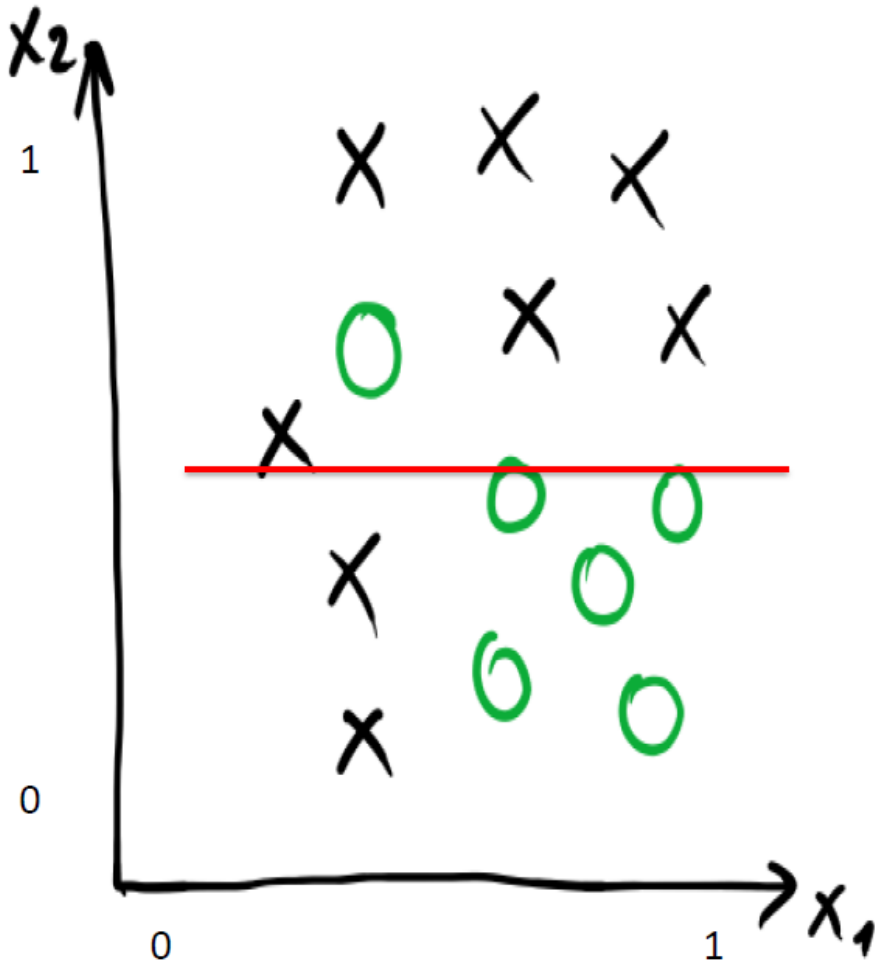


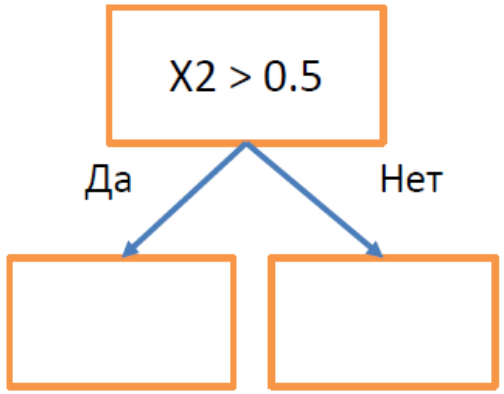
X2 > 0.5

Деревья решений

Пример

Нашли лучшее первое ветвление

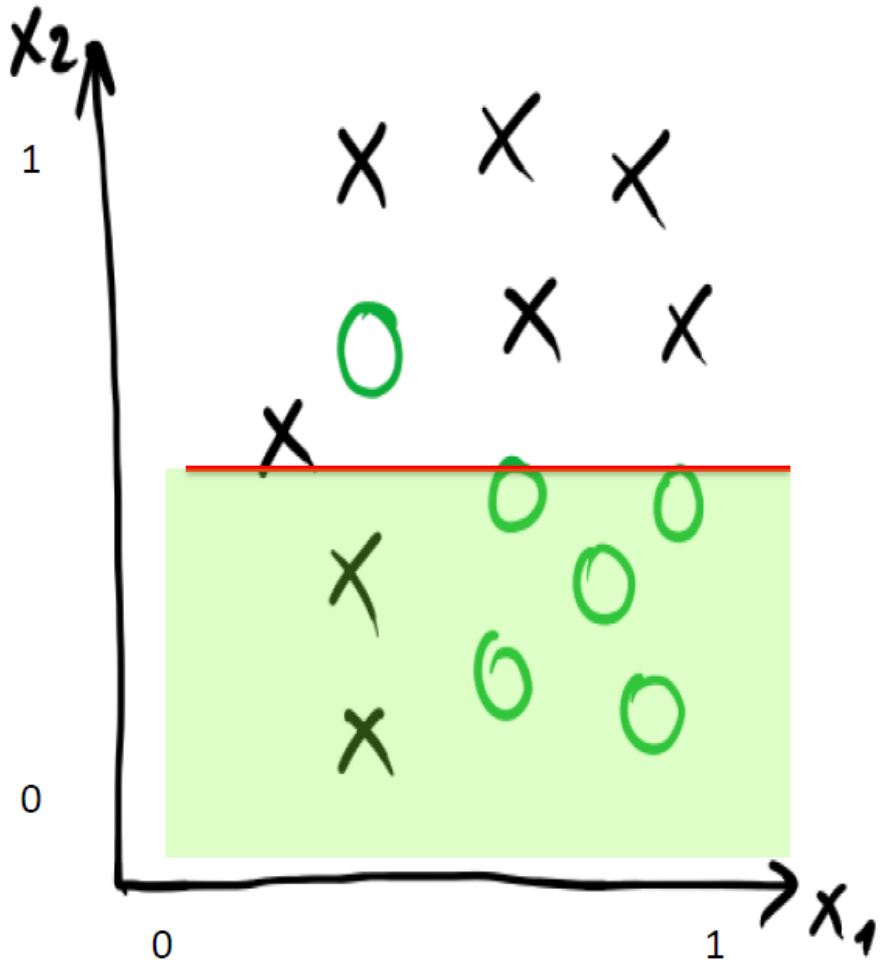


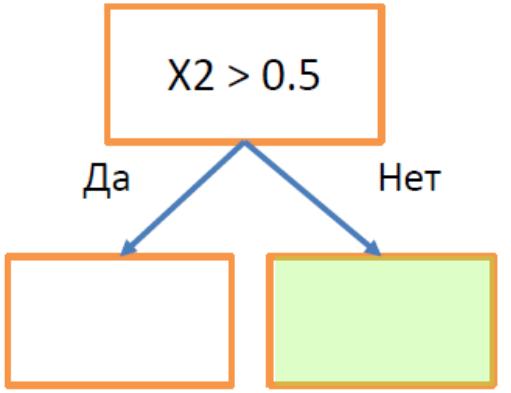


Деревья решений

Пример

Нашли лучшее первое ветвление



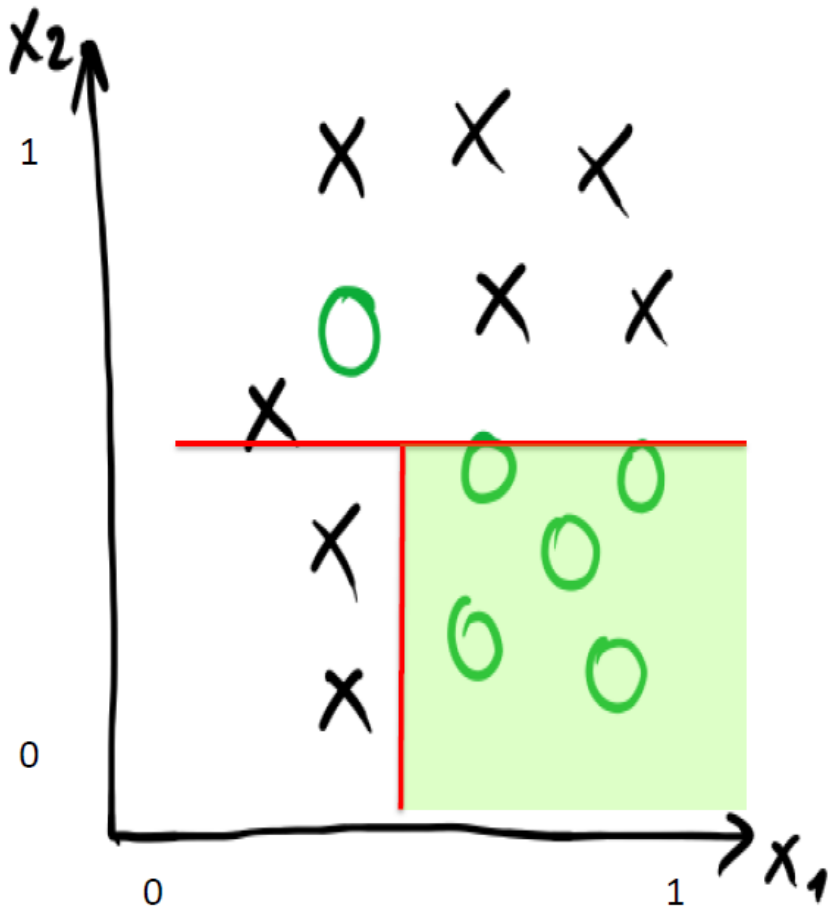


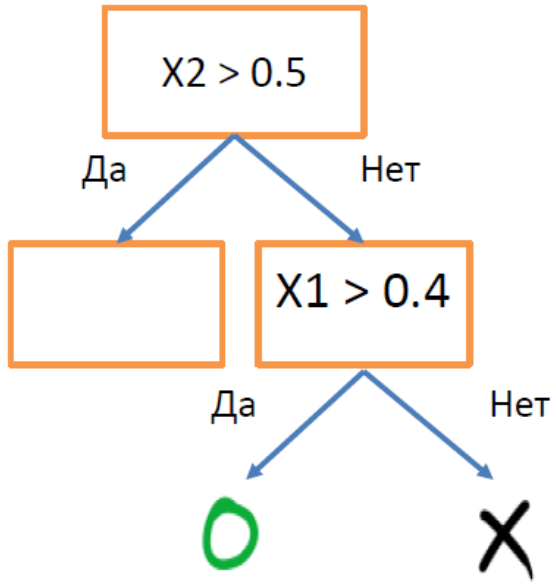
Продолжим эту ветку

Деревья решений

Пример

Нашли лучшее первое ветвление

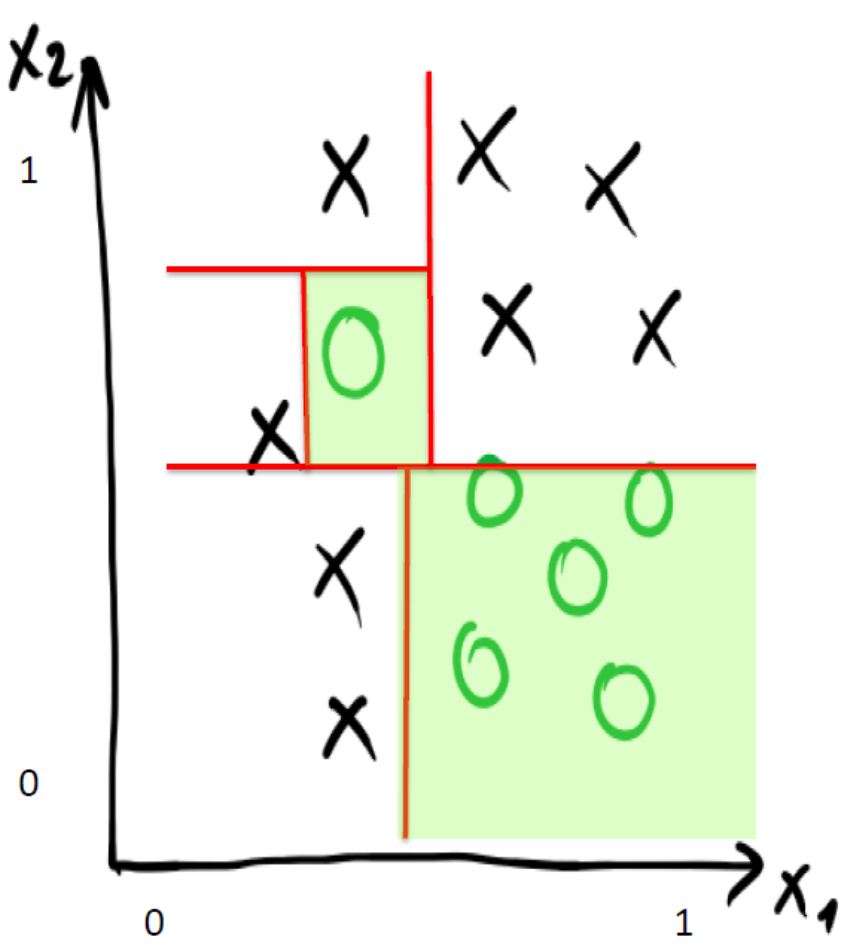


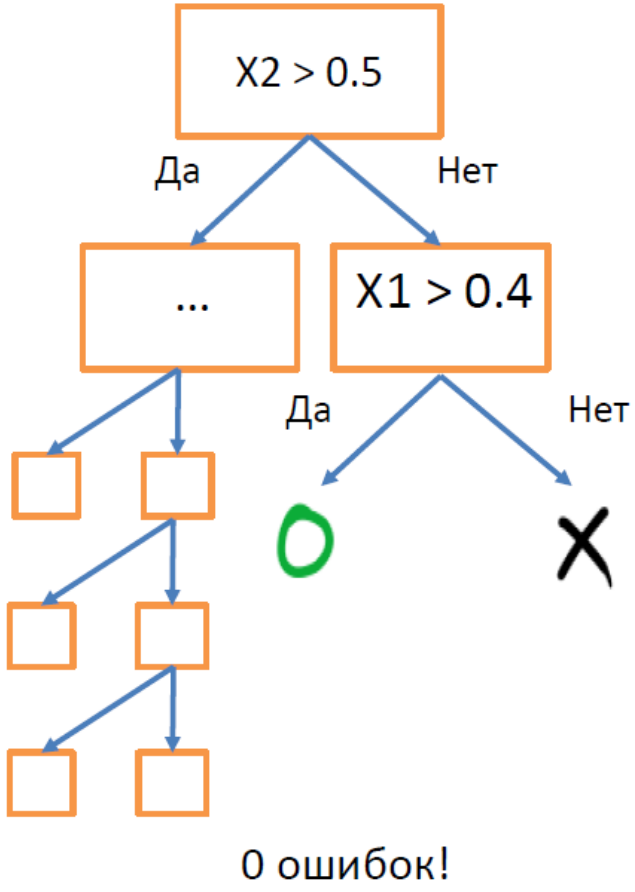


Деревья решений

Пример

Построили все дерево



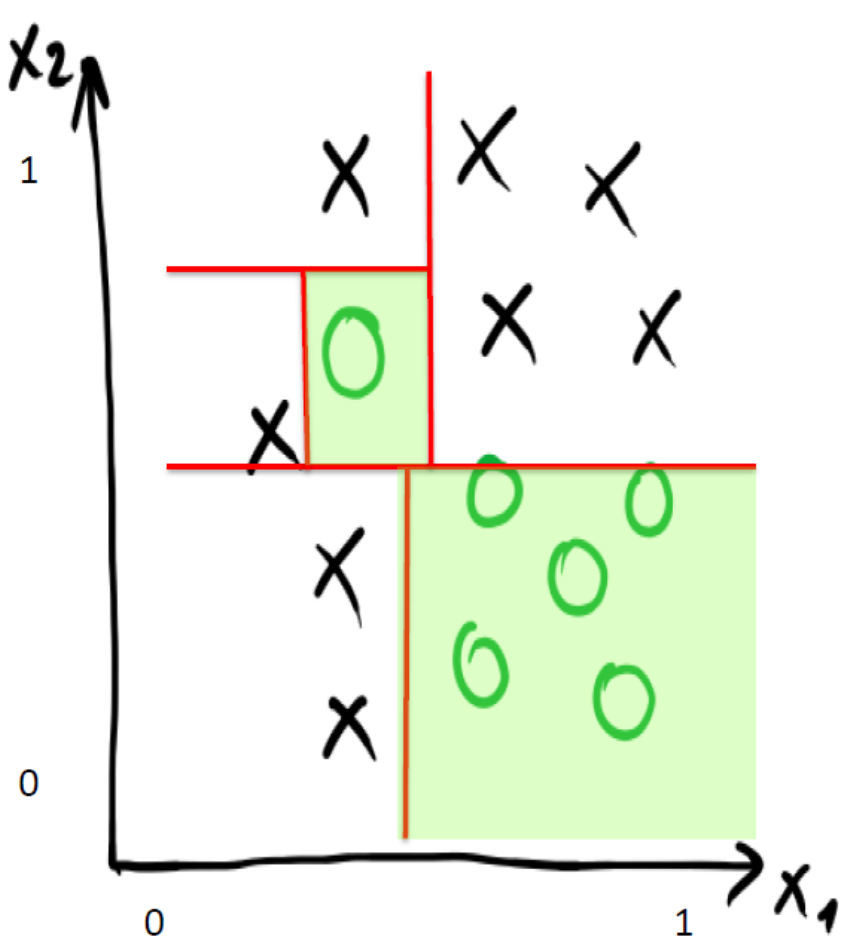


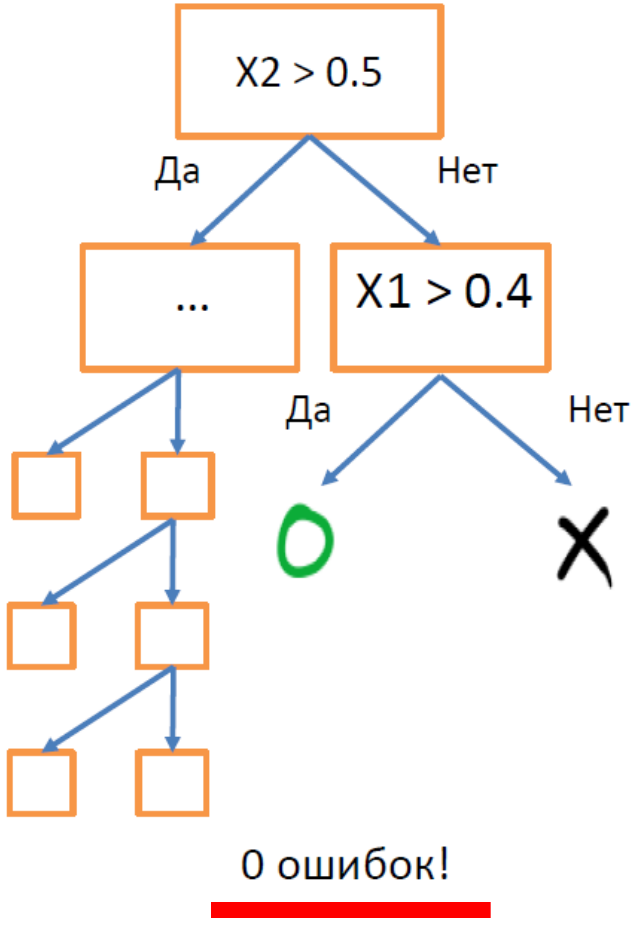
0 ошибок!

Деревья решений

Пример

Построили все дерево





Деревья решений

Переобучение

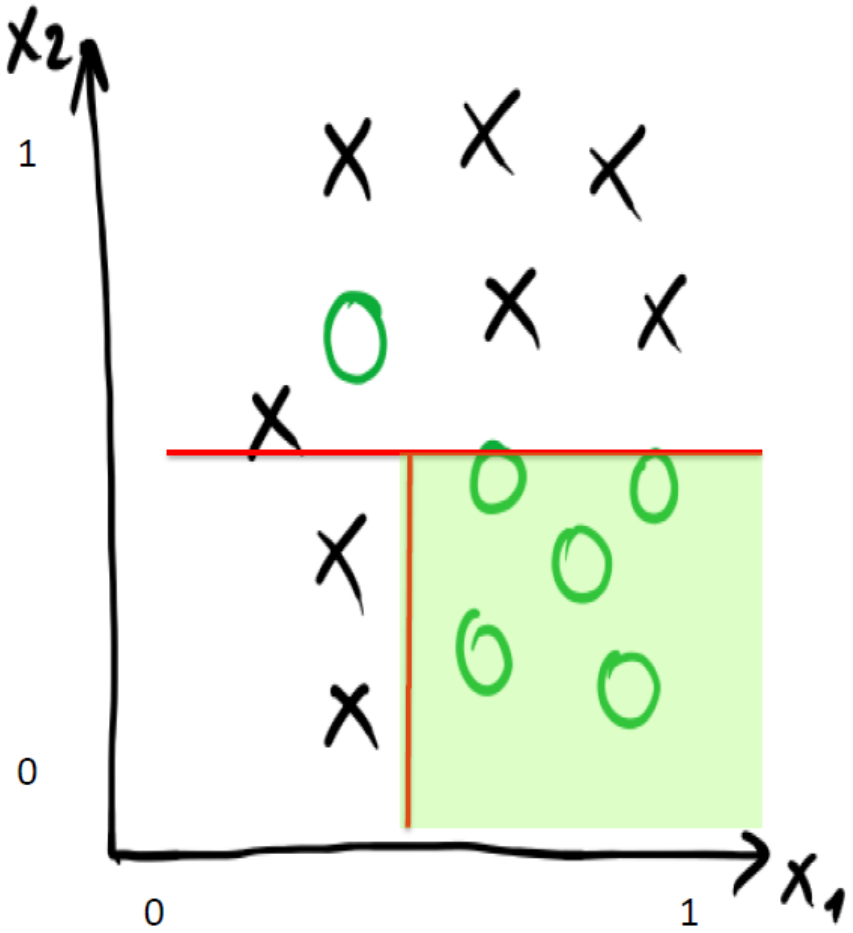
Для любой выборки можно построить решающее дерево, не допускающее на ней ни одной ошибки: в каждом листе разместить по одному объекту выборки

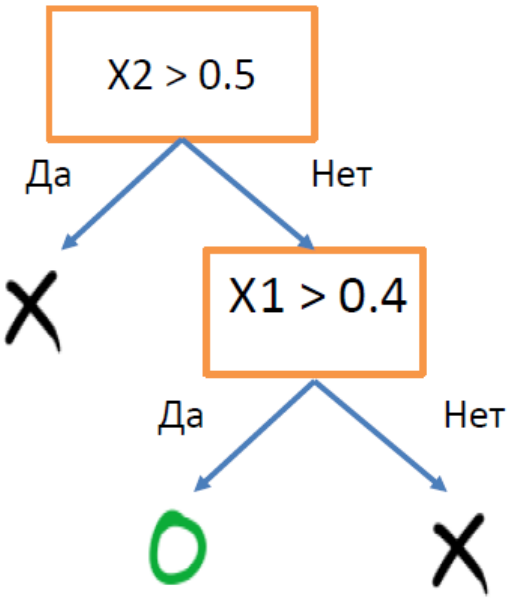
Такое дерево скорее всего будет переобученным и не сможет показать хороший результат на новых данных

Деревья решений

Пример

Остановимся раньше!





1 ошибка, но скорее всего будет лучше на тесте!

Деревья решений

Что влияет на построение дерева

- вид предикатов в вершинах
- функционал качества $Q(X, j, t)$
- критерий останова

Критерии информативности

Пусть R – множество объектов, попадающих в вершину на данном шаге, а R_l и R_r - объекты, попадающие в левую и правую ветки после разбиения.

Критерий информативности (impurity criterion) оценивает качество распределения целевой переменной среди объектов множества R (т. е. это мера неоднородности или разнообразие целевых переменных внутри группы R)

Чем меньше разнообразие целевой переменной, тем меньше должно быть значение критерия информативности — соответственно, мы минимизируем его значение:

$$\boldsymbol{H(R_l) \to min, H(R_r) \to min}$$

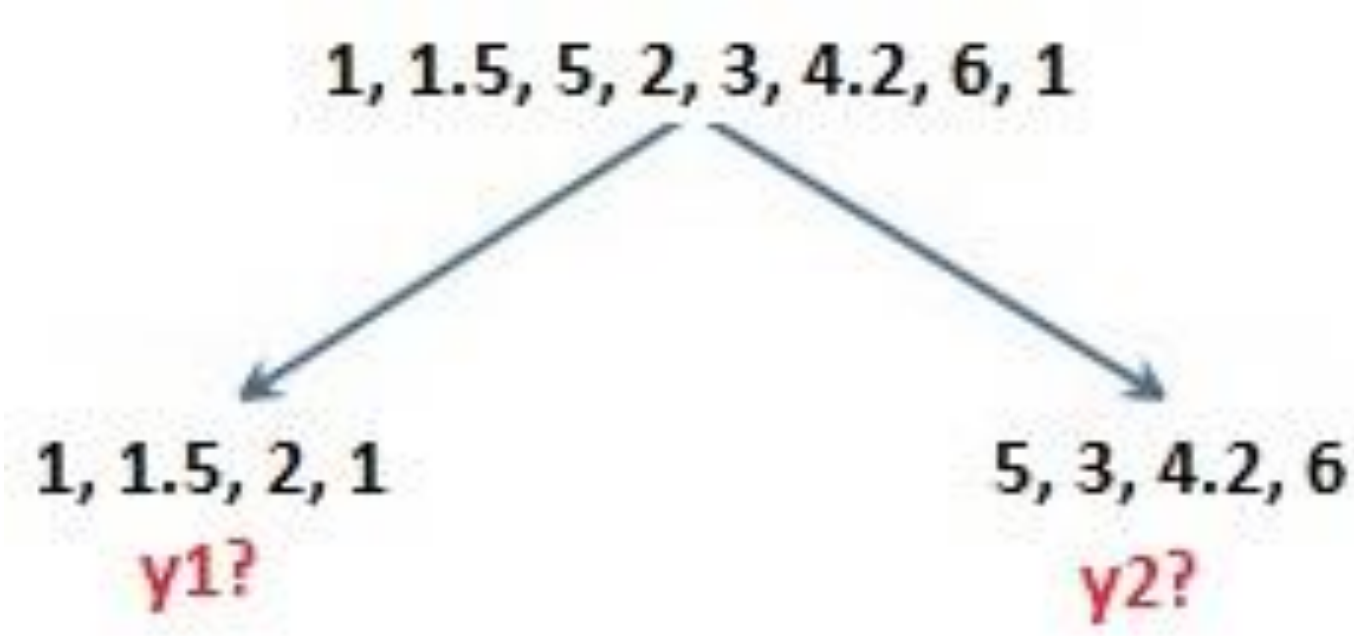
Функционал качества Q мы при этом будем максимизировать

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

Критерии информативности

Задача регрессии

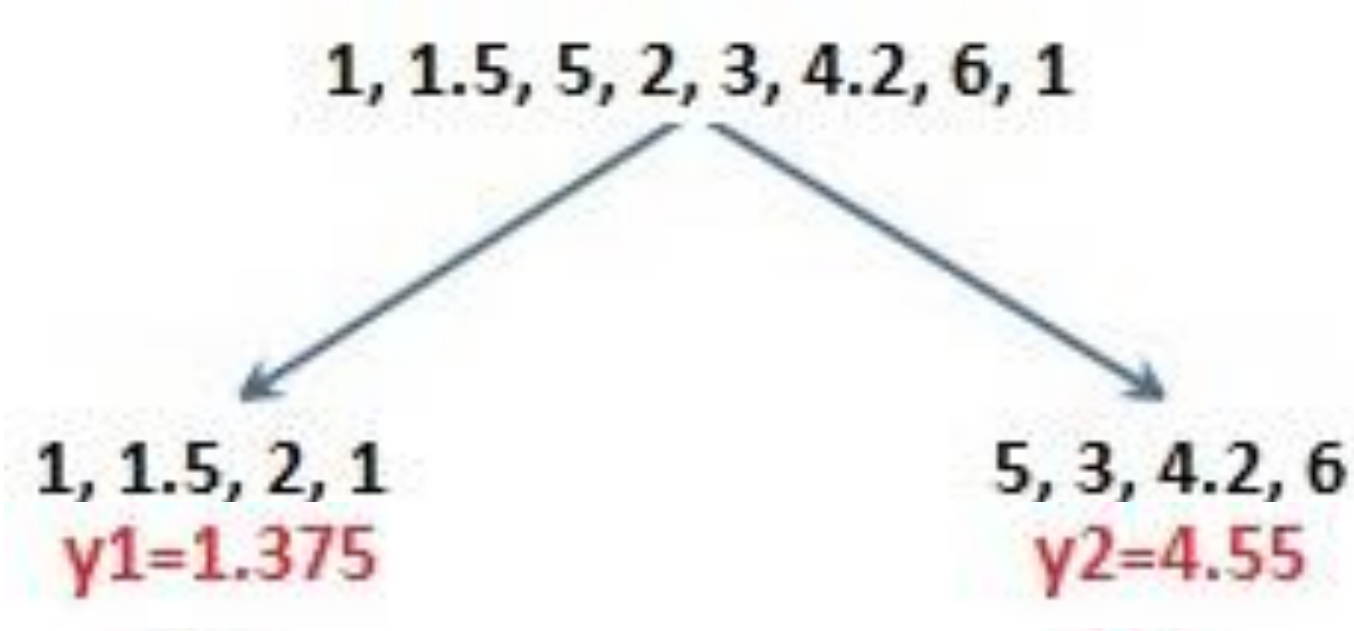
Предположим, что в лист дерева попало несколько объектов. В каждом листе дерево предсказывает константу. Какую константу выгоднее всего выдать в качестве ответа?



Критерии информативности

Задача регрессии

Если в качестве функционала ошибки в листе использовать среднеквадратичную ошибку, то в качестве ответа надо выдавать среднее значение целевых переменных всех объектов, попавших в лист.



Критерии информативности

Вид критерия информативности

- В каждом листе дерево выдает константу c (вещественное число – в регрессии, класс или вероятность класса – в классификации).
- Чем лучше объекты в листе предсказываются этой константой, тем меньше средняя ошибка на объектах:

$$\boldsymbol{H(R) = \min_{c \in \mathbb{R}} \frac{1}{|R|} \sum_{(x_i, y_i) \in R} L(y_i, c)},$$

где $L(y, c)$ – некоторая функция потерь.

Критерии информативности

H(R) в задаче регрессии

Если в качестве функции потерь взять квадратичную ошибку, то

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

Критерии информативности

H(R) в задаче регрессии

Если в качестве функции потерь взять квадратичную ошибку, то

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

Ее минимум достигается при

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

то есть в листе предсказывается среднее значение целевой переменной на объектах, попавших в лист, поэтому

Критерии информативности

H(R) в задаче регрессии

Если в качестве функции потерь взять квадратичную ошибку, то

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

Ее минимум достигается при

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

то есть в листе предсказывается среднее значение целевой переменной на объектах, попавших в лист, поэтому

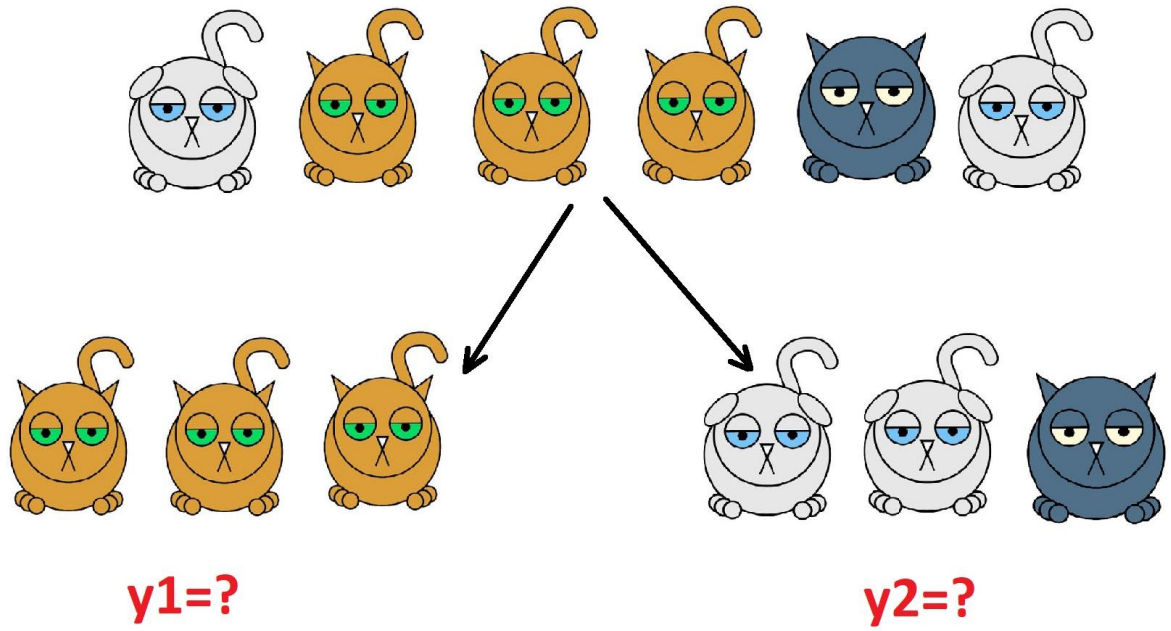
Информативность $\mathbf{H(R)}$ в вершине дерева – это дисперсия целевой переменной (для объектов, попавших в вершину).

Чем меньше дисперсия, тем меньше разброс целевой переменной объектов, попавших в лист

Критерии информативности

Задача классификации

Предположим, что в лист дерева попало несколько объектов. В каждом листе дерево предсказывает класс объекта. Какой класс выгоднее всего выдать в качестве ответа?

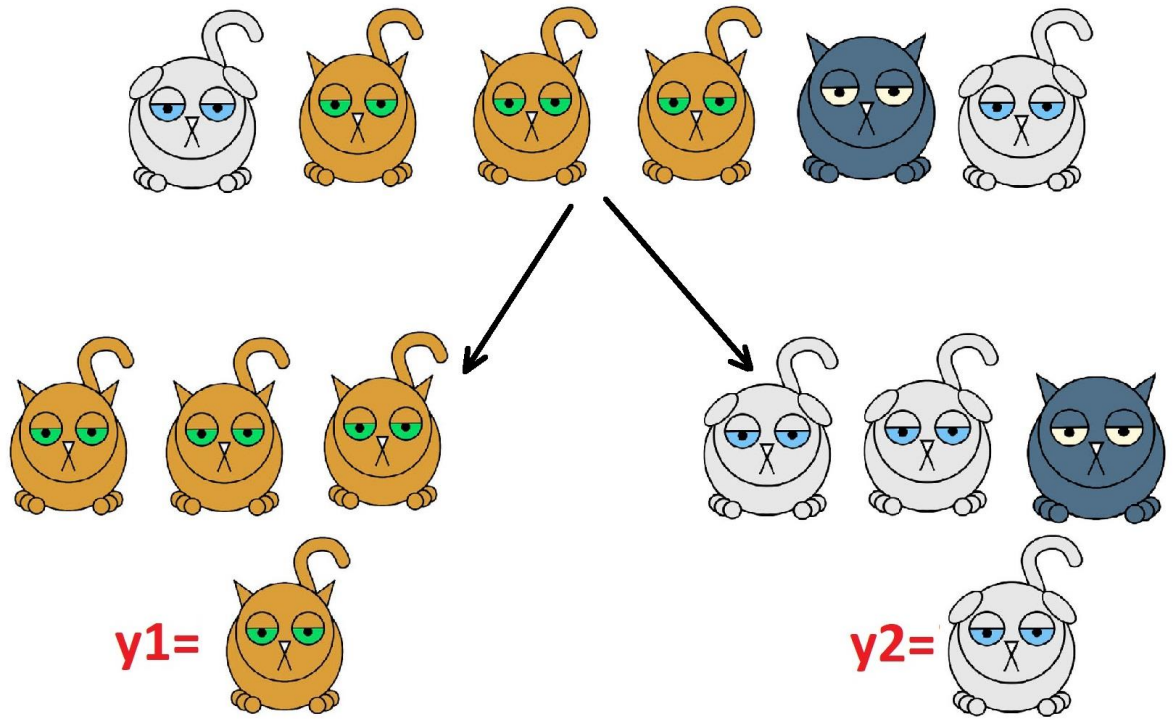


Источник: «Статистика и котики»: В. Савельвев (https://books.yandex.ru/reader/iReNl3e7?resource=book)

Критерии информативности

Задача классификации

Предположим, что в лист дерева попало несколько объектов. В каждом листе дерево предсказывает класс объекта. Какой класс выгоднее всего выдать в качестве ответа?



Источник: «Статистика и котики»: В. Савельвев (https://books.yandex.ru/reader/iReNl3e7?resource=book)

Критерии информативности

H(R) в задаче классификации

Решаем задачу классификации с K классами: $1,2,\dots,K$.

Пусть p_k доля объектов класса k, попавших в вершину:

$$p_k = \frac{1}{|R|} \sum_{(x_i,y_i)\in R} [y_i = k]$$

Пусть k_* - самый представительный класс в данной вершине:

$$k_* = \underset{k}{argmax}\, p_k$$

Информативность H(R) в вершине дерева – это ошибка классификации:

$$H(R) = \min_{c\in Y} \frac{1}{|R|} \sum_{(x_i,y_i)\in R} [\boldsymbol{y_i \neq c}]$$

Данный критерий является достаточно грубым, поскольку учитывает частоту p_{k_*} лишь одного класса

Критерии информативности

Критерий Джинни

В каждой вершине в качестве ответа будем выдавать не класс, а распределение вероятностей классов: $c = (c_1, \dots, c_K), \sum_i c_i = 1$.

Качество распределения можно измерить с помощью **критерия Бриера**, измеряющего точность вероятностных прогнозов:

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x_i, y_i) \in R} \sum_{k=1}^{K} (c_k - [y_i = k])^2$$

Можно показать, что оптимальный вектор вероятностей состоит из долей классов: $c_* = (p_1, \dots, p_K)$

Если подставить эти вероятности в исходный критерий информативности и провести ряд преобразований, то мы получим **критерий Джини:** $\boldsymbol{H(R) = \sum_{k=1}^{K} p_k (1 - p_k)}$

Критерии информативности

Энтропийный критерий

Запишем логарифм правдоподобия:

$$H(R) = \min_c \left(-\frac{1}{|R|} \sum_{(x_i, y_i) \in R} \sum_{k=1}^{K} [y_i = k] \log c_k \right) \quad (*)$$

На векторе $c_* = (p_1, \ldots, p_K)$ функционал (*) записывается в виде

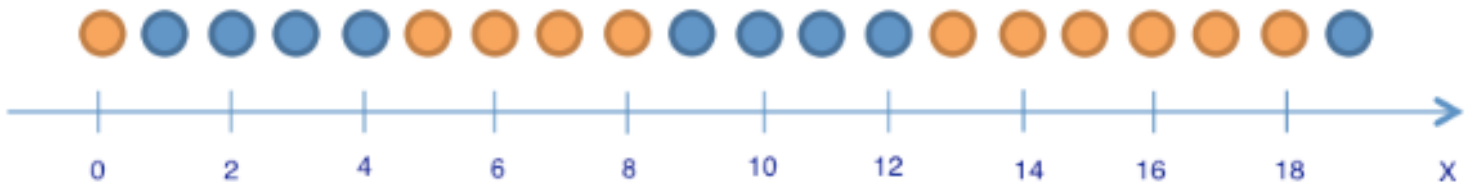
$$H(R) = -\sum_{k=1}^{K} p_k \log p_k \text{ (энтропия)}$$

Энтропия $H(R) \geq 0$ (минимум на распределении $p_i = 1, p_j = 0, j \neq i$)

$max\ H(R)$ достигается на равномерном распределении $p_1 = \cdots = p_K = \frac{1}{K}$.

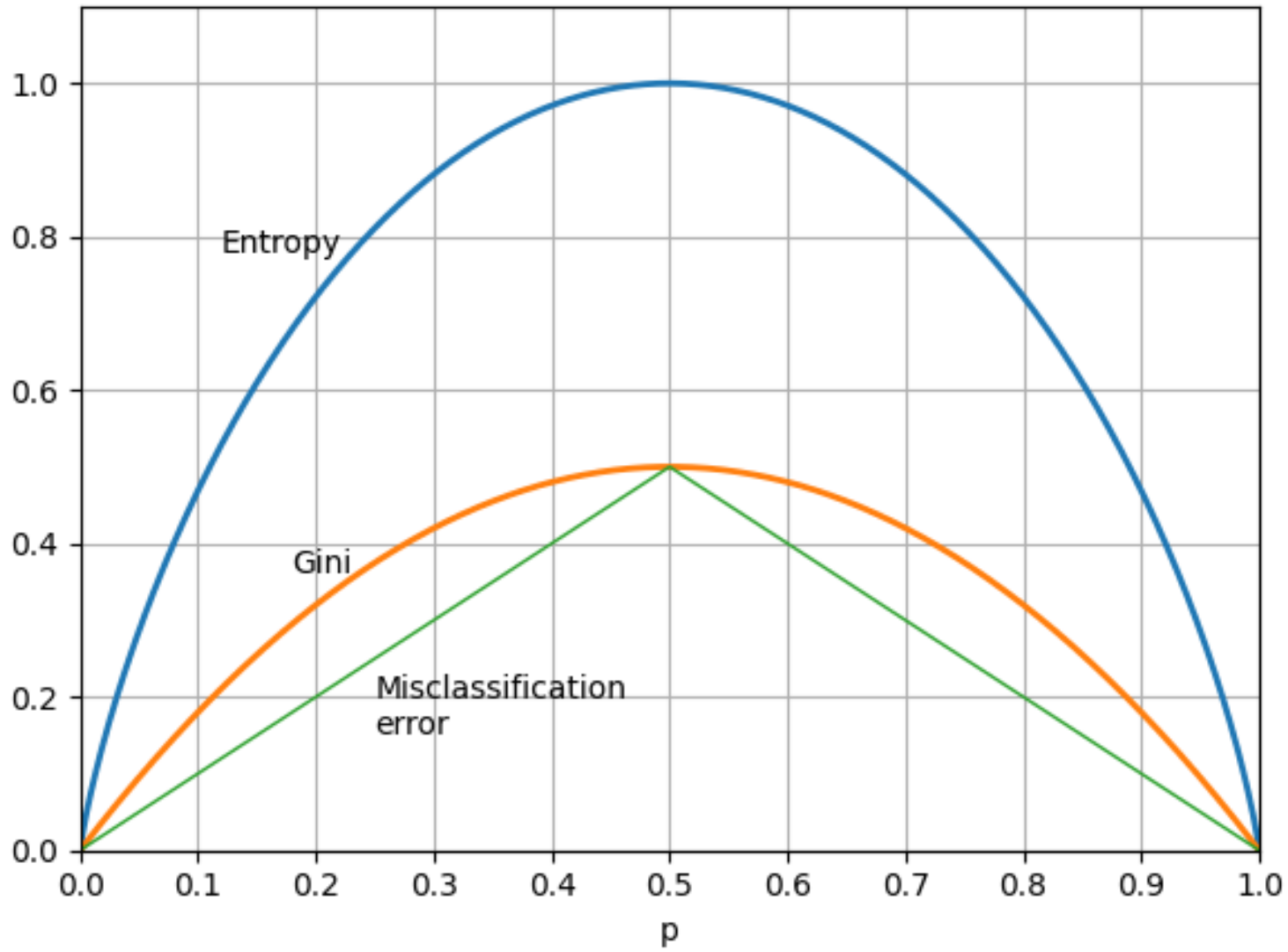
Критерии информативности

Энтропийный критерий (пример использования)



$$p_1 = \frac{9}{20}, p_2 = \frac{11}{20} \Rightarrow \text{энтропия } H_0 = -\frac{9}{20}\log\frac{9}{20} - \frac{11}{20}\log\frac{11}{20} \approx 1$$

Критерии информативности



Критерии останова

- Ограничение максимальной глубины дерева (max_depth)
- Ограничение минимального числа объектов в листьях (min_samples_leaf)
- Ограничение максимального числа листьев в дереве
- Останов в случае если все объекты в листе относятся к одному классу
- Требование, чтобы функционал качества при дроблении увеличивался как минимум на s %.

Правильный подбор критерия существенно повышает качество, но затратен и требует кросс-валидации.

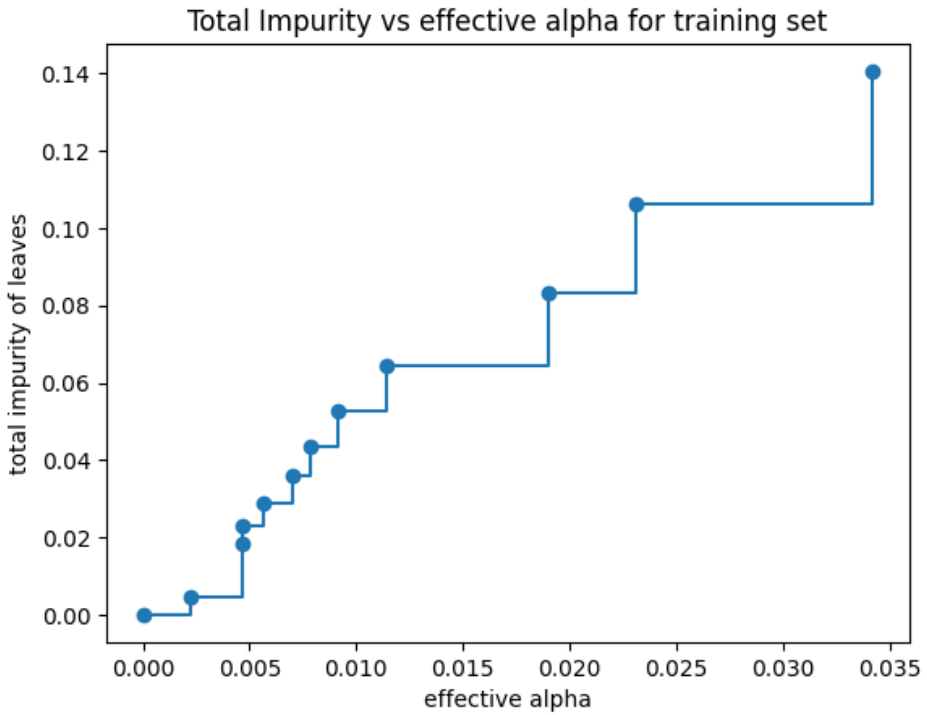
Стрижка дерева (прунинг)

Альтернатива критериям останова: строится переобученное дерево (например, пока в каждом листе не окажется по одному объекту) и затем производится оптимизация структуры.

Источник: https://www.geeksforgeeks.org/machine-learning/how-to-choose-a-in-cost-complexity-pruning/

Стрижка дерева (прунинг)

Параметр прунинга — α — подбирается на кросс-валидации



Источник: https://www.geeksforgeeks.org/machine-learning/how-to-choose-a-in-cost-complexity-pruning/

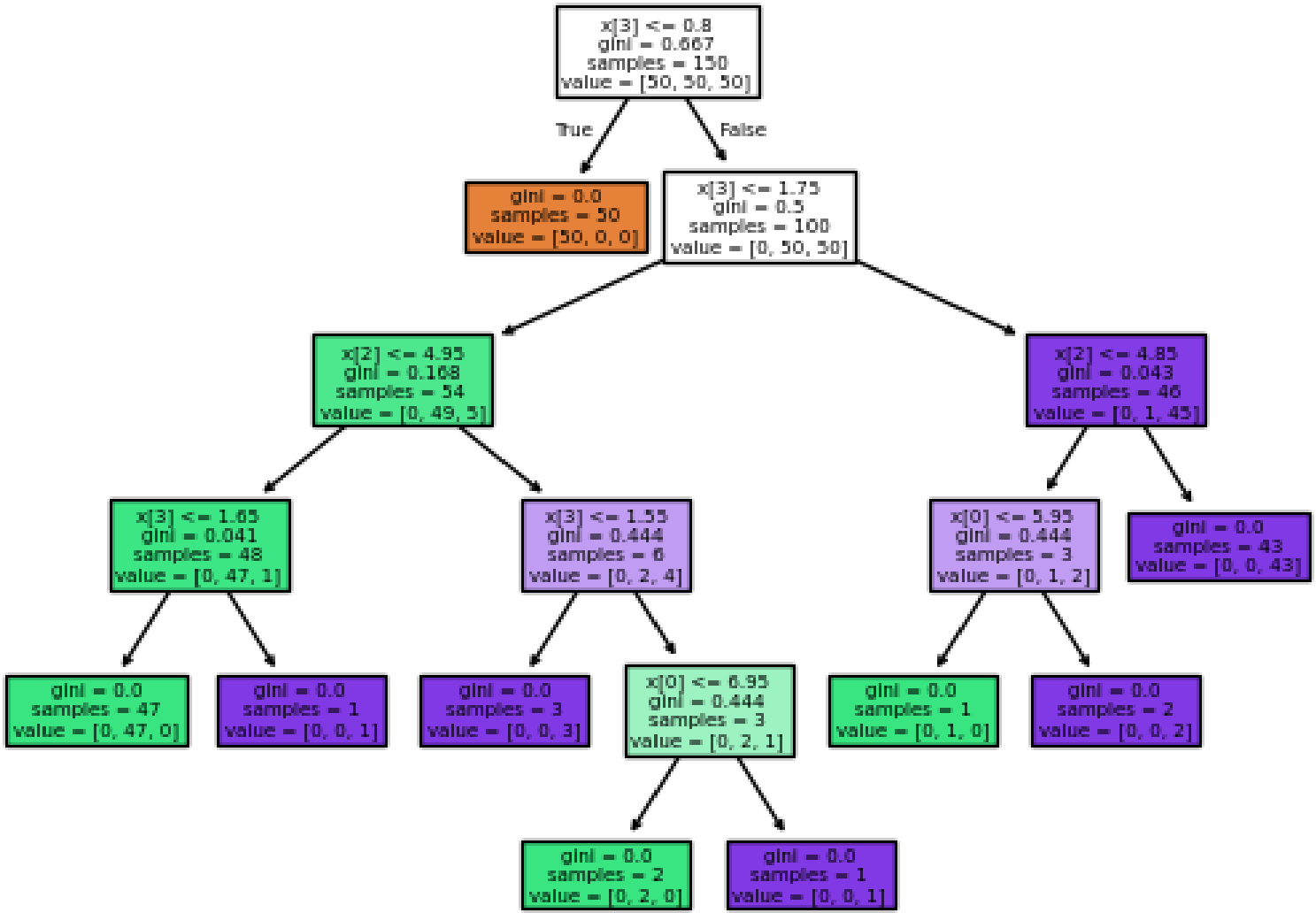
Стрижка дерева (прунинг)

Стрижка позволяет достичь лучшего качества по сравнению с ранним остановом, но фактически используется редко и почти не реализована в большинстве библиотек, так как сами по себе деревья — это слабые алгоритмы и используются в основном только в более сложных моделях (случайных лесах, бустинге).

В первом случае они должны быть переобучены, во втором — должны иметь маленькую глубину и не требуют стрижки

Пример дерева

Ирисы Фишера



Плюсы деревьев

- Высокая степень интерпретируемости: хорошо визуализируются и имеют четкие понятные предикаты (например, «возраст > 25»)
- Быстро обучаются и выдают прогноз
- Имеют малое число параметров

Минусы деревьев

- Очень чувствительны к шумам в данных, модель сильно меняется при небольшом изменении обучающей выборки
- Разделяющая граница имеет свои ограничения (состоит из гиперплоскостей)
- Проблема поиска оптимального дерева (NP-полная задача, поэтому на практике используется жадное построение дерева)
- Необходимость борьбы с переобучением (стрижка или какой-либо из критериев останова)