Занятие 8. Решающие деревья

Колмагоров Евгений
ml.hse.dpo@yandex.ru

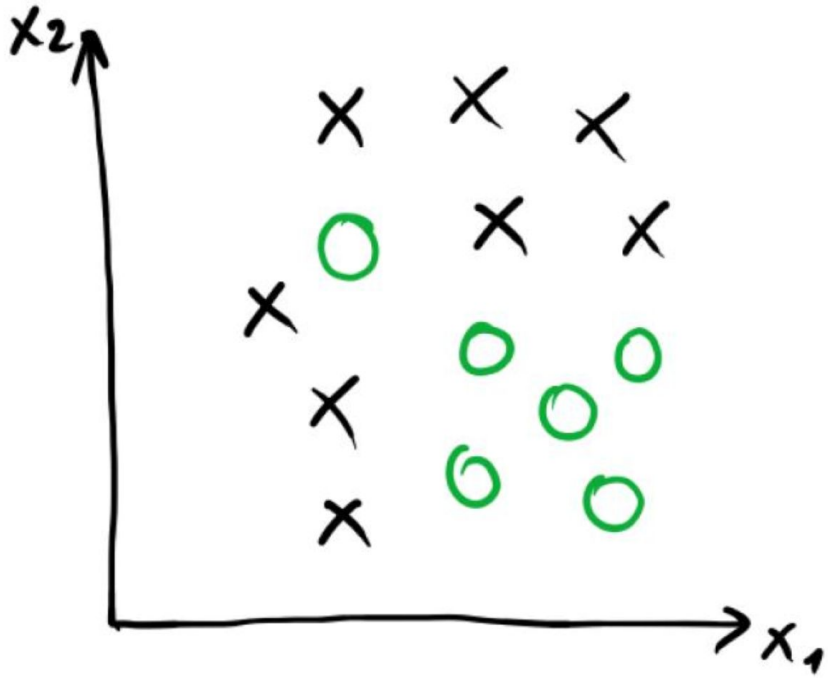
2 декабря 2024

План лекции

1. Определение решающего дерева
2. Алгоритм построения
3. Критерии информативности для классификации и регрессии
4. Обрезка дерева и работа с пропусками
5. Кодирование категориальных признаков

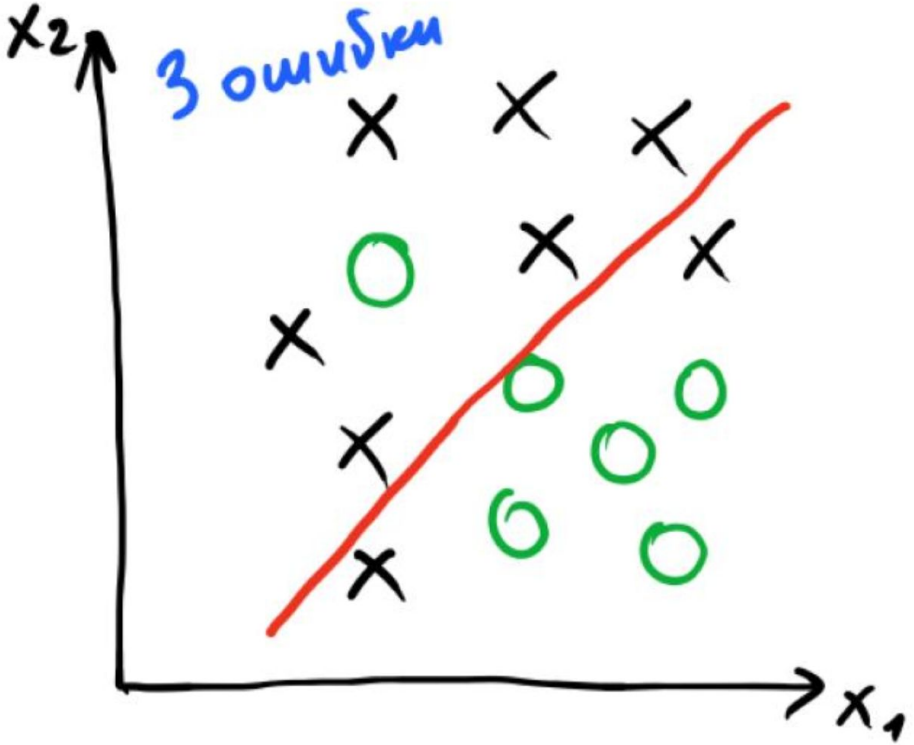
Пример классификации

Представим, что необходимо провести классификацию на данной выборке



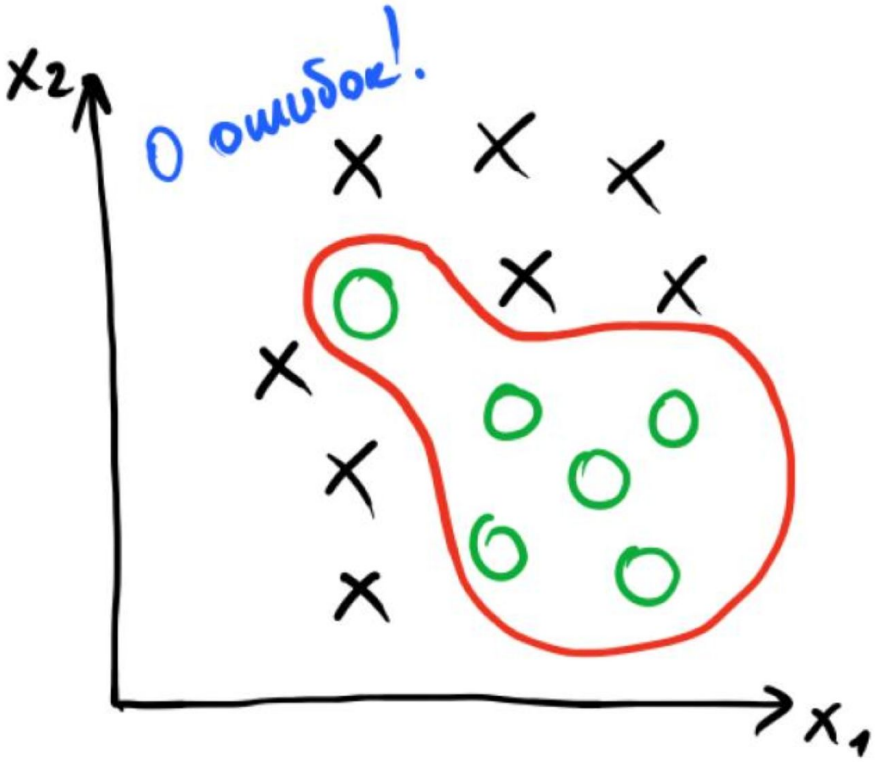
Пример классификации

Так как выборка линейно неразделима линейные методы будут допускать как минимум 3 ошибки



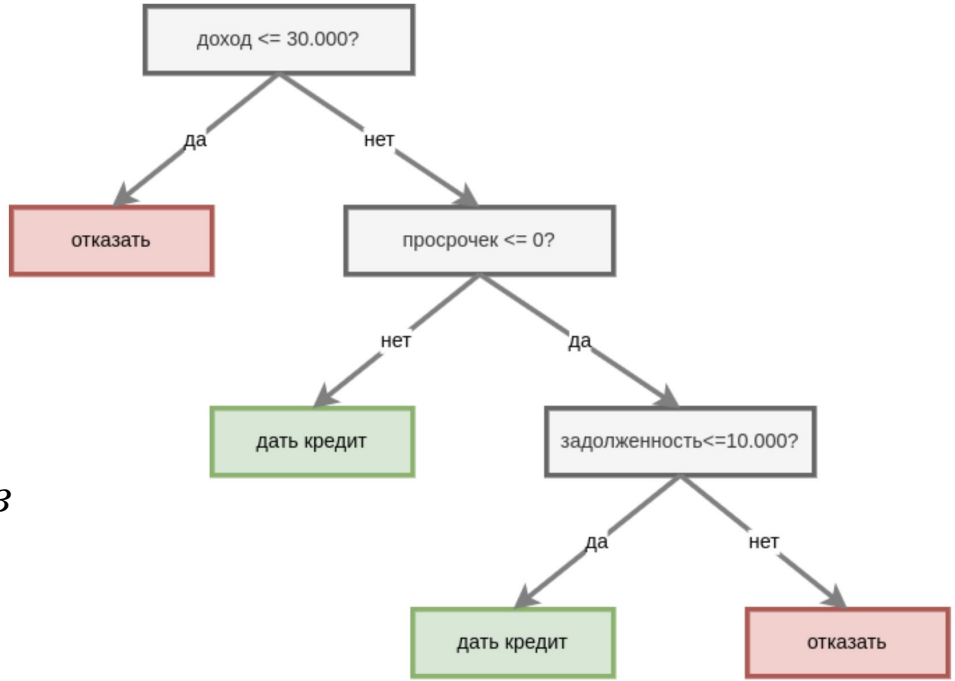
Пример классификации

Чтобы получить 0 ошибок необходимо использовать алгоритмы более сложной нелинейной природы



Деревья решений

На самом деле в повседневной жизни, люди довольно часто не замечают, что при принятии того или иного решения используется древовидный подход

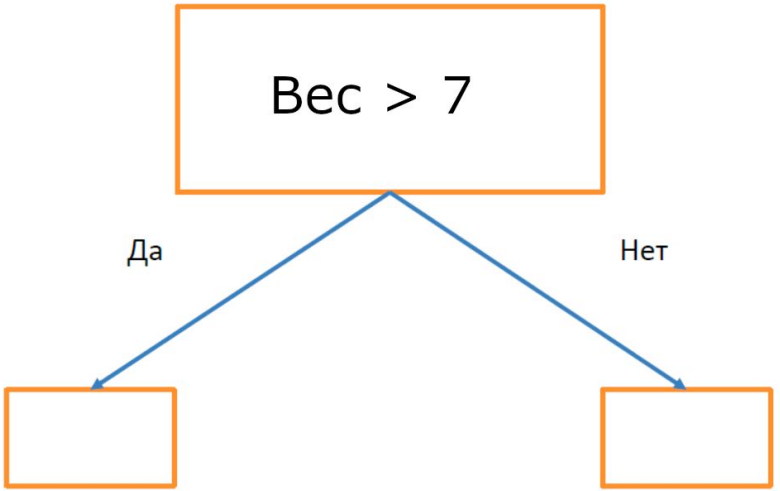


Основная идея: попробовать исходя из имеющихся данных для обучения автоматически построить подобное дерево

Определение решающего дерева

Решающее дерево – это бинарное дерево, в котором:

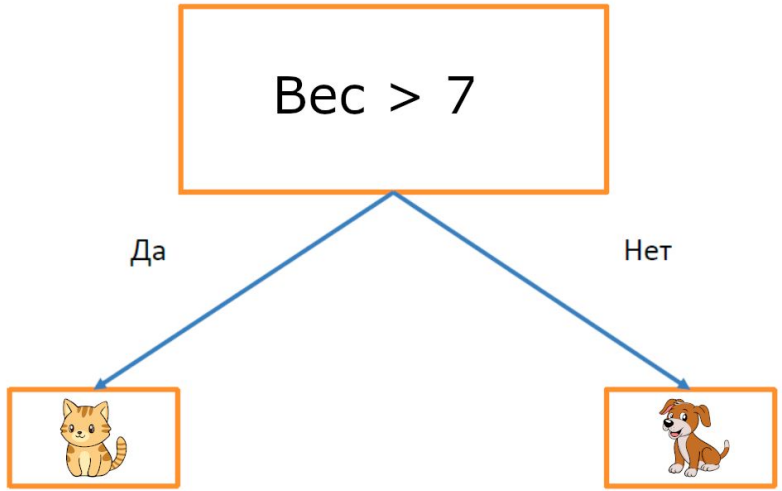
1) каждой внутренней вершине v приписана функция β_v: $R \rightarrow \{0, 1\}$



Определение решающего дерева

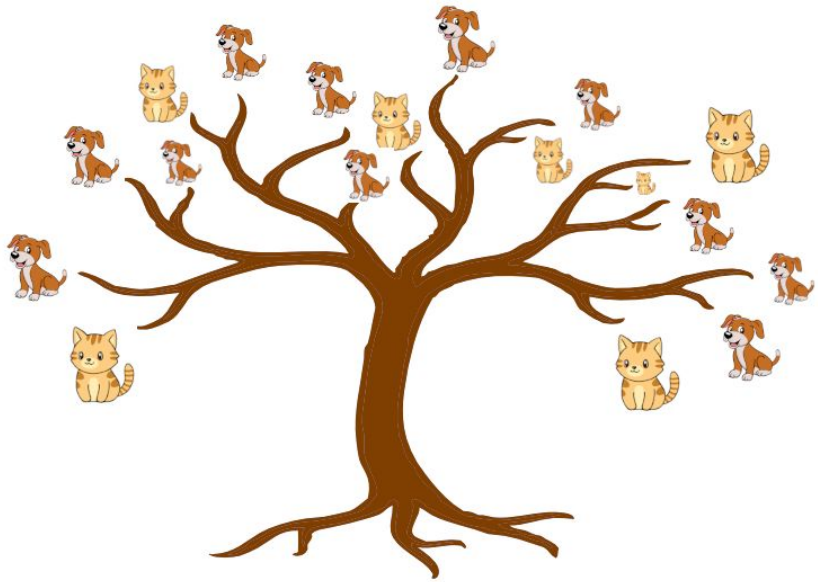
Решающее дерево – это бинарное дерево, в котором:

1) каждой внутренней вершине v приписана функция β_v: $R \to \{0, 1\}$
2) каждой листовой вершине v приписан прогноз $c_v \in Y$. Для классификации это класс, для регрессии вещественное значение целевой переменной



Построение дерева

Можно построить решающее дерево, не допускающее ни одной ошибки на обучающем множестве, поместив в каждый его узел один единственный объект.



Но скорее всего, оно будет слишком глубоким и переобученым

Проблема построения оптимального дерева

Построение оптимального дерева с минимальным количеством листовых вершин и не допускающим на обучающем множестве ни одной ошибки относится к NP-полным задачам, которые могут быть решены только полным перебором

Жадный подход построения дерева

Будем строить дерево жадным алгоритмом, выбирая оптимальное разбиение по некоторому некоторому признаку x_j

Напоминание. **Жадный алгоритм** – это алгоритм, который на каждом шагу делает локально наилучший выбор в надежде, что итоговое решение будет оптимальным.

Жадный подход построения дерева

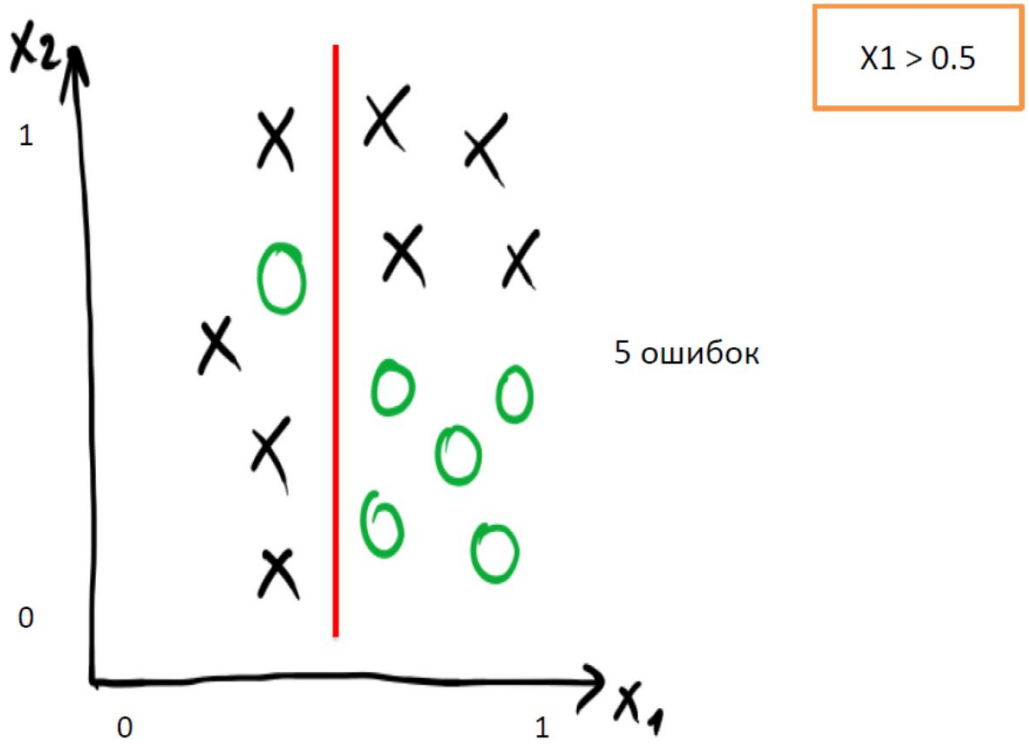
1 шаг: найдём наилучшее разбиение всей выборки X на две части $R_l(j, t) = \{x \mid x_j < t\}$ и $R_r(j, t) = \{x \mid x_j \geq t\}$ с точки зрения некоторого функционала $Q(X, j, t)$:

- найдём наилучшие j и t (j – отвечает за номер признака, по которому производится разбиение, а t – за величину порога)
- создадим внутреннюю вершину v, поставив в неё предикат $[x_j < t]$

2 шаг: для каждой из полученных подвыборок R_l и R_r рекурсивно применим шаг 1. И тд.

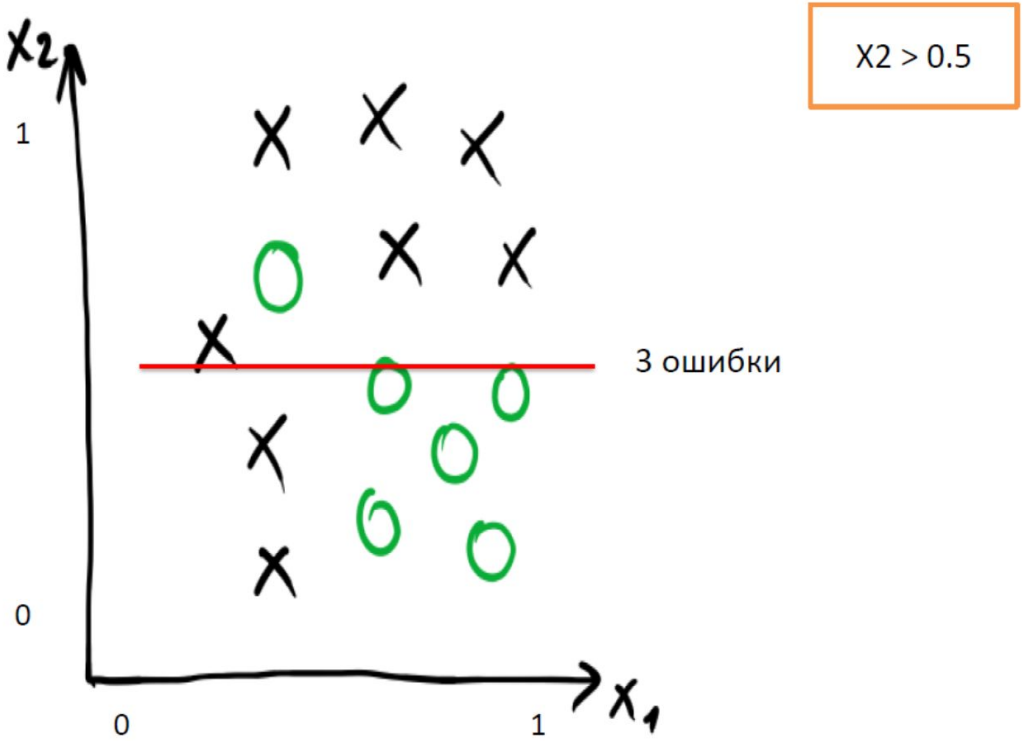
Пример

- Попробуем найти наилучшее разбиение по первому признаку



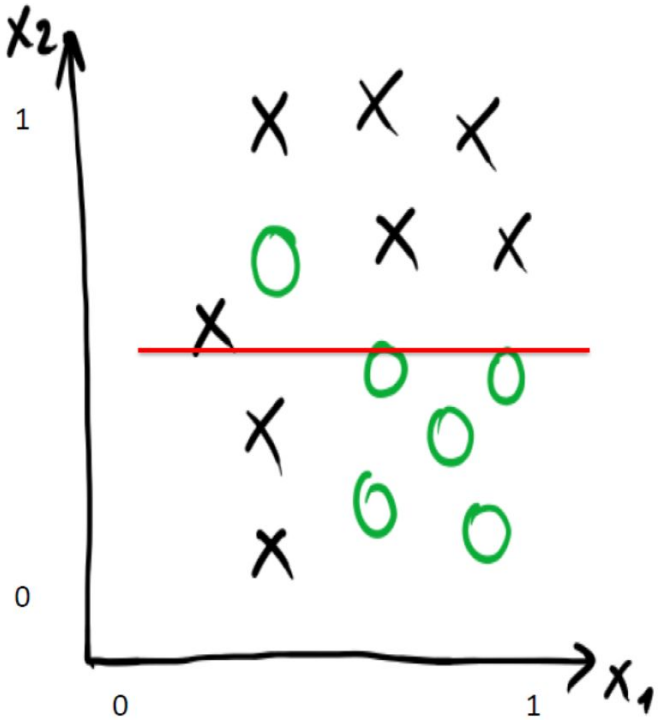
Пример

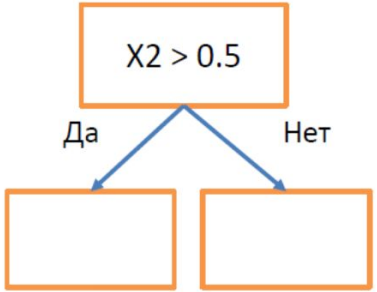
- Попробуем найти наилучшее разбиение по второму признаку



Пример

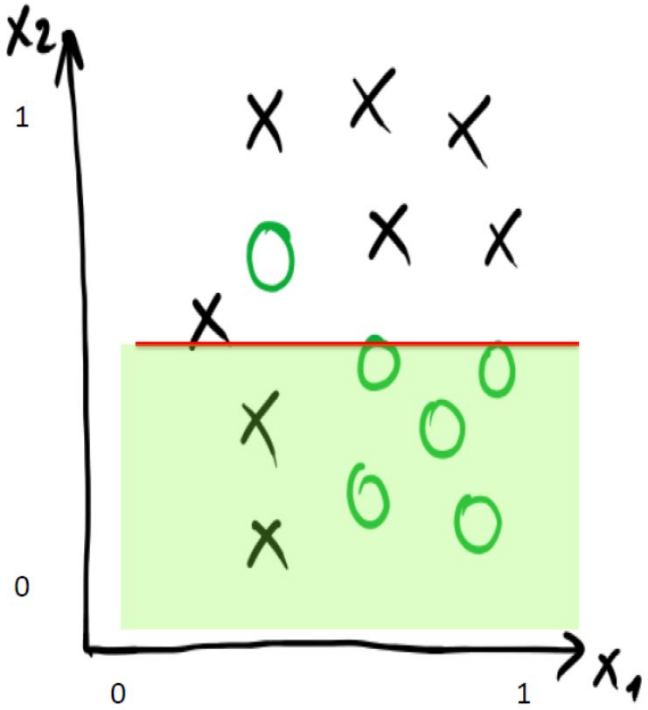
- разбиение по x_2 даёт меньшую ошибку, поэтому формируем решающую функцию во внутреннем узле на основе него

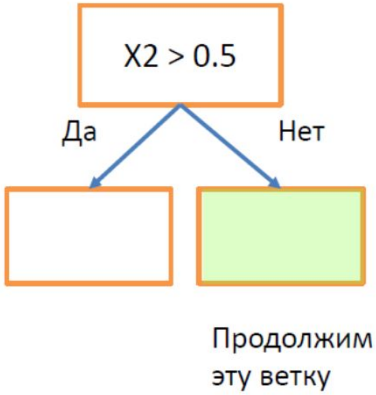




Пример

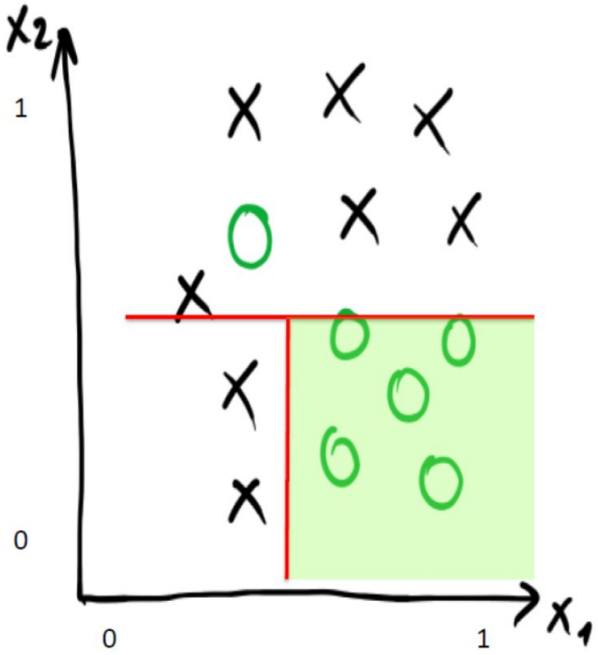
- Ищем оптимальное разбиение для объектов из правого поддерева

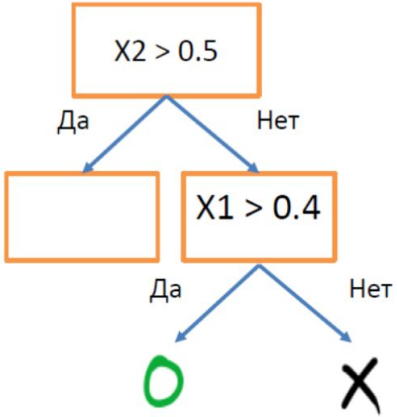




Пример

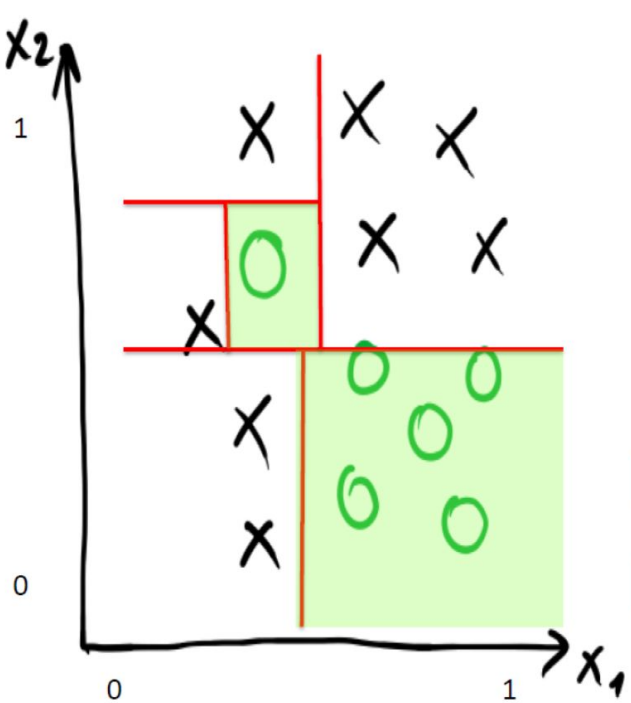
- Оптимальное разбиение для правого поддерева теперь по x_1

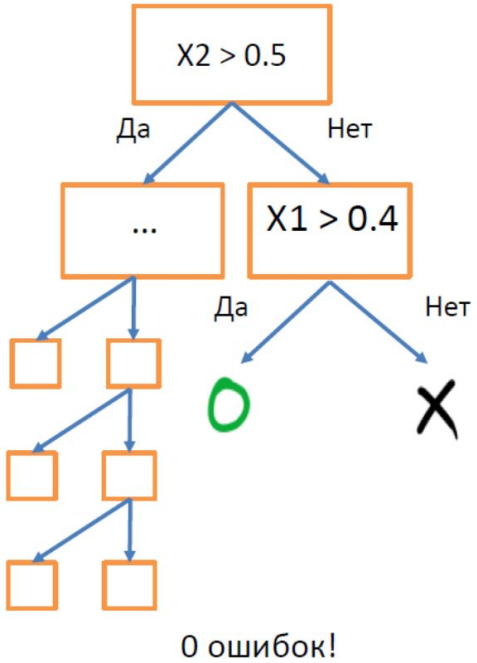




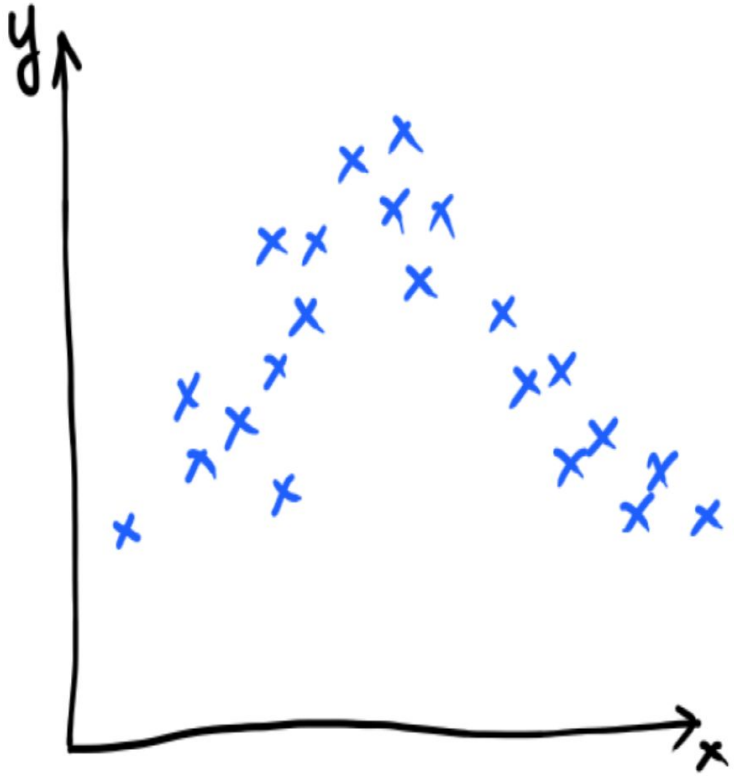
Пример

- Строим до некоторого момента остановки

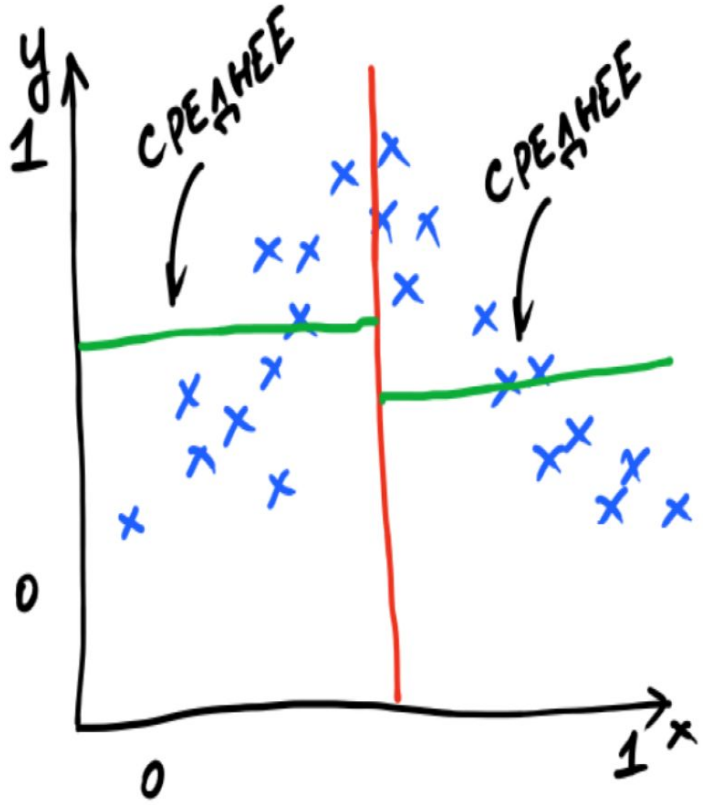


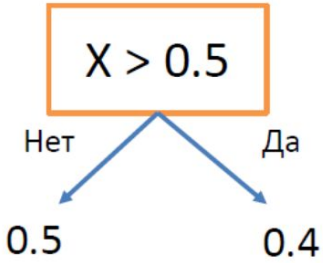


Пример для задачи регрессии



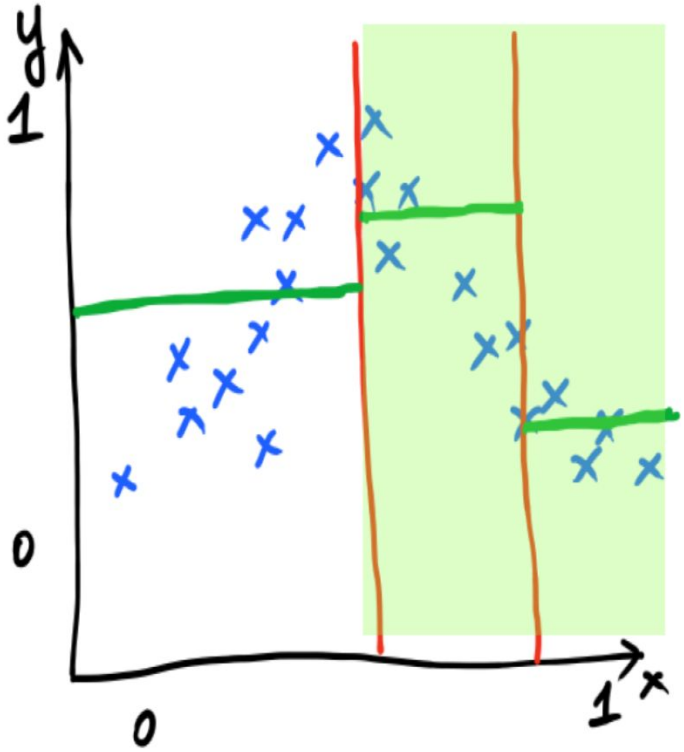
Пример для задачи регрессии

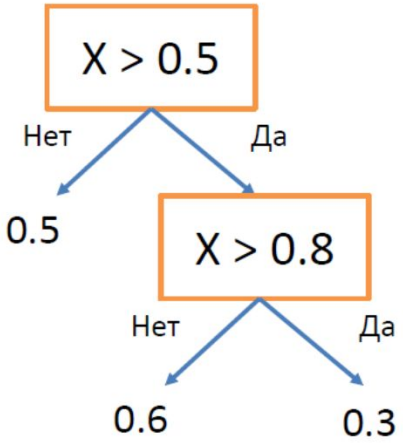




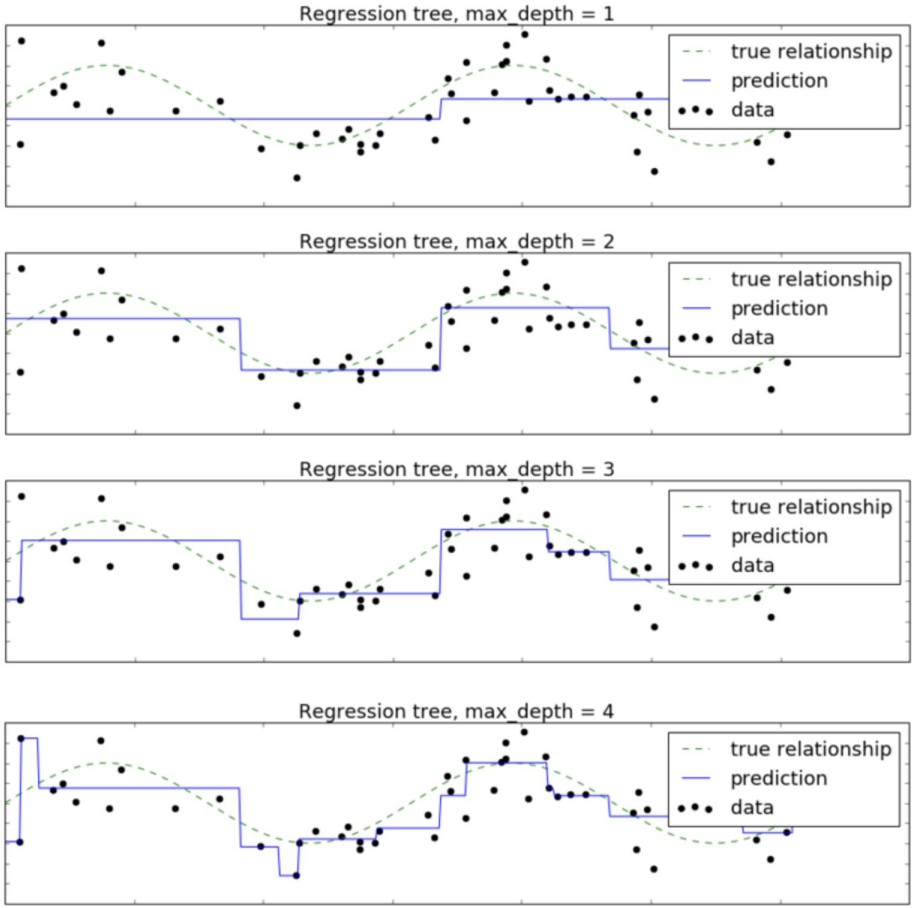
Продолжим эту ветку

Пример для задачи регрессии





Влияние глубины дерева на сложность решения



Когда останавливаемся

При построении дерева могут быть использованы следующие критерии остановки:

- Достигнута максимальная глубина **H**
- Число объектов в узле меньше **K**
- Достигнуто максимальное количество узлов **L**
- Функционал качества после разбиения не улучшился на минимально допустимую величину **E**

Величины H, K, L и E играют роль гиперпараметров в построении решающего дерева

Функция неопределённости

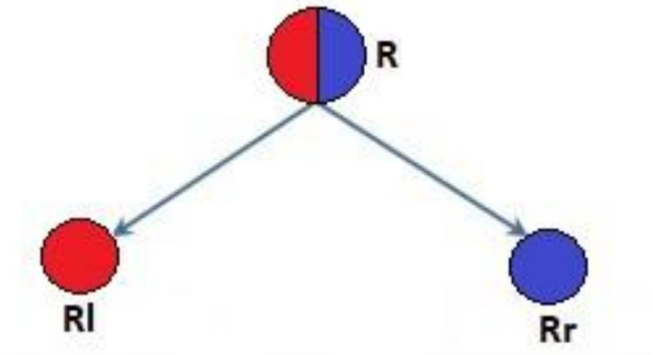
При поиске оптимальных j и t использовался функционал $Q(X, j, t)$, который называется **функцией неопределённости**.

Выбираются такие функции, чтобы:

- Была равна 0, если все объекты попадающие в лист имеют одинаковое значение
- Функция тем больше, чем сильнее неопределённость в объектах, попадающих в один лист

Критерий информативности

Цель: хотим минимзировать неопределённость, чтобы после разбиения объектов на две группы внутри каждой группы как можно больше объектов было одного класса или значения



Критерий информативности

Введём H(R) – критерий информативности – мера неоднородности целевых переменных внутри группы R.

Тогда высокая степень однородности в узлах R_l, R_r дерева соответствует:

- $H(R_l) \rightarrow \min$
- $H(R_r) \rightarrow \min$

Минимизация функции неопределённости

Будем использовать функционалы следующего вида:

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

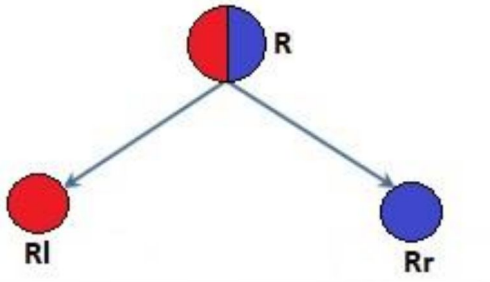
В задаче построения решающего дерева данный функционал в отличии от других алгоритмов максимизируется:

$$Q(X, j, t) \to max_{j,t}$$

Интерпретация функционала

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

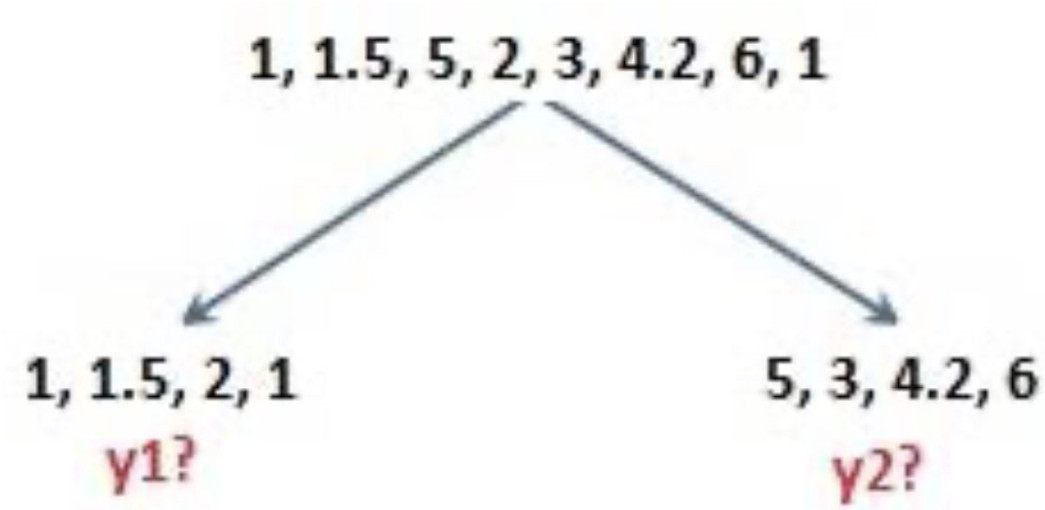
Максимизация данного функционала обусловлена тем, что после расщепления объектов хорошо иметь в узлах более однородные множества, чем было до этого во внутренней вершине



Пример для задачи регрессии

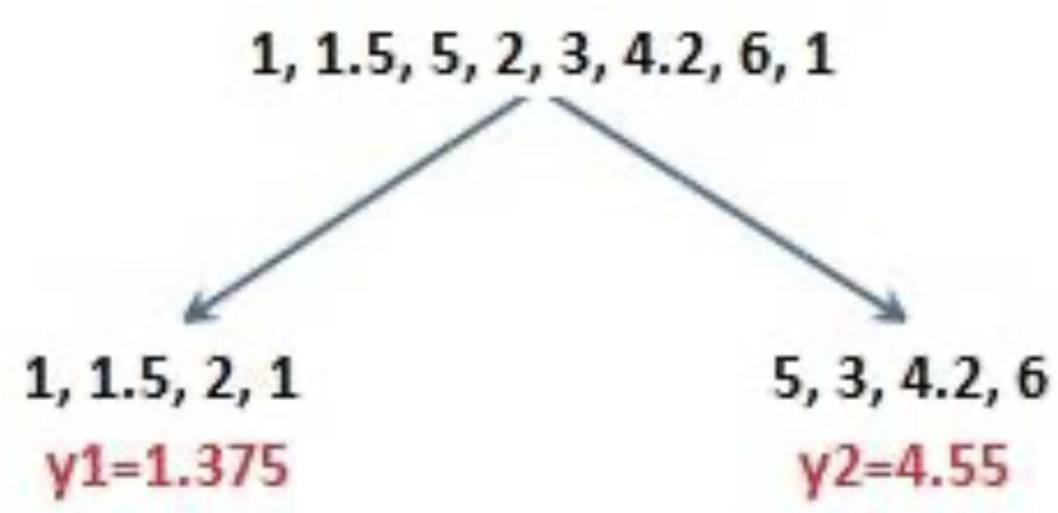
Предположим, что в лист дерева попало несколько объектов. В каждом листе дерево может предсказывать константу.

Какую константу выгоднее всего выдавать в качестве ответа?



Пример для задачи регрессии

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



Критерий информативности для регрессии

- В каждом листе дерево выдаёт константу C
- Чем лучше объекты в листе предсказываются этой константой, тем меньше средняя ошибка на объектах:

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

$L(y_i, c)$ – некоторая функция потерь для задачи регрессии

Критерий информативности для регрессии

Если в качестве L взять среднеквадратичную ошибку, то

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

Минимум среднеквадратичной ошибки

Если в качестве L взять среднеквадратичную ошибку, то

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

Минимум константного решения в MSE достигает на среднем значении таргетов (см. лекцию 3):

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

Критерий информативности для регрессии

Тогда подставив оптимальный c в исходную формулу:

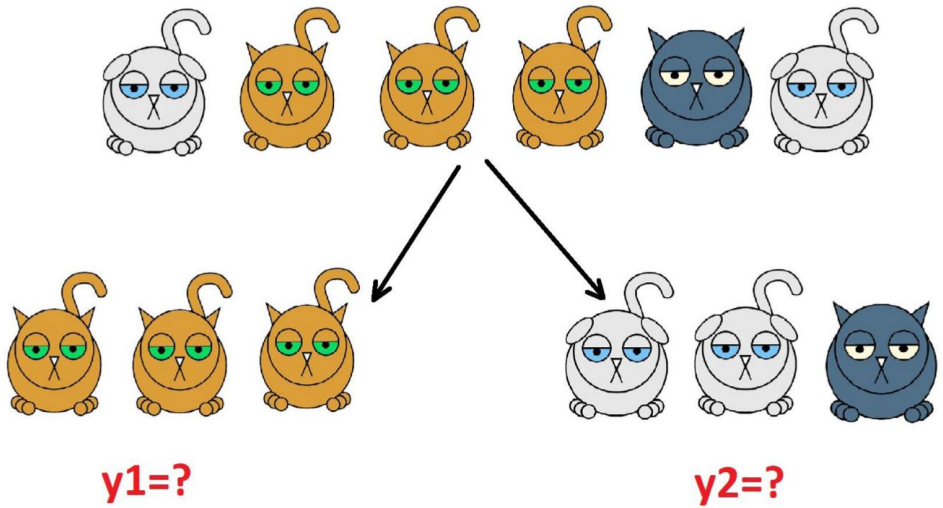
$$H^*(R) = \frac{1}{|R|}\sum_{(x_i,y_i)\in R}(y_i - c)^2 = \frac{1}{|R|}\sum_{(x_i,y_i)\in R}(y_i - \bar{y})^2 = D_{y\in R}[Y]$$

Таким образом дисперсия целевых переменных в листе есть мера неопределённости для задачи регрессии с функционалом ошибк MSE

Пример для задачи классификации

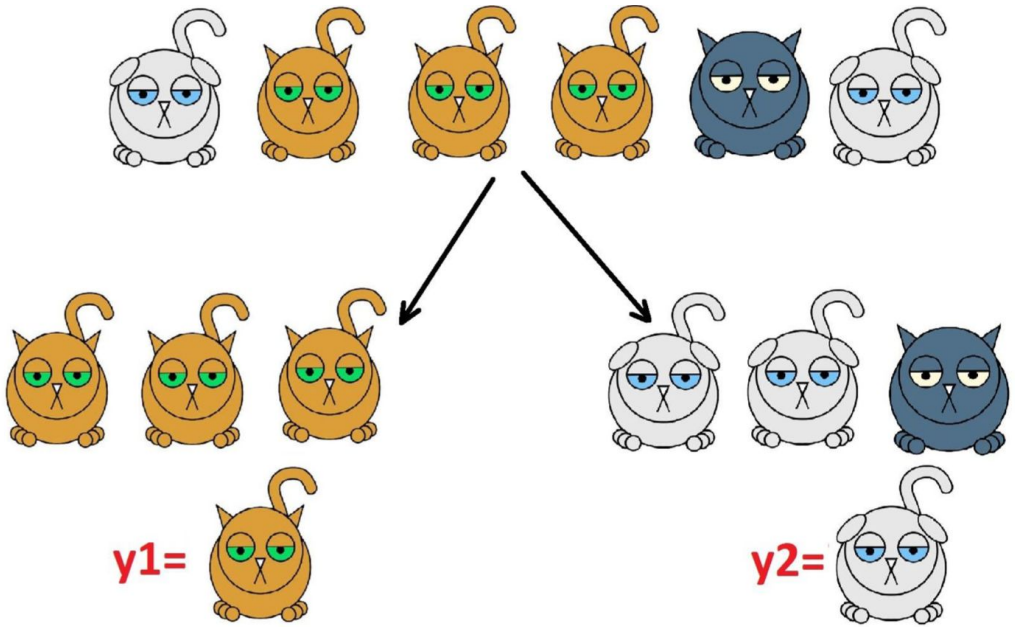
Предположим, что в лист дерева попало несколько объектов. В каждом листе дерево предсказывает класс объекта.

Какой класс выгоднее всего выдавать в качестве ответа?



Пример для задачи классификации

Разумнее всего в качестве константного ответа в листе выдавать самый представительный класс



Критерий информативности для классификации

Решаем задачу классификации с K классами: 1, 2, …, 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 частотный класс в данной вершине:

$$k^* = argmax_k p_k$$

Тогда вероятность ошибки классификации:

$$p_{err} = 1 - p_{k^*}$$

Критерий Джини

Критерий Джини измеряет вероятность ошибки при случайном угадывании класса по правилу:

$$\hat{y} = \begin{cases} 1, \ c \text{ вероятностью } p_1 \\ 2, \ c \text{ вероятностью } p_2 \\ \dots \\ K, \ \text{с вероятностью } p_K \end{cases}$$

Тогда вероятность ошибки:

$$p(\hat{y} \neq y) = \sum_{i=1}^{K} p(\hat{y} \neq i | y = i) p(y = i) = \sum_{i=1}^{K} (1 - p_i) \cdot p_i$$

Критерий Джини

$$H(R) = \sum_{i=1}^{K} (1 - p_i) \cdot p_i - \text{ критерий Джини}$$

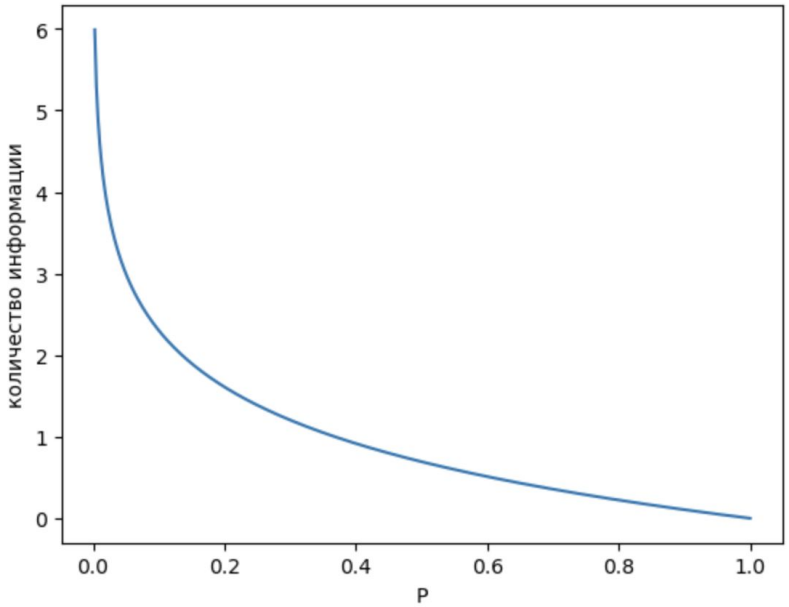
- H(R) будет максимален, когда все классы равновероятны
- H(R) будет равен нулю, когда все объекты принадлежат одному классу

Энтропийный критерий

Ещё одним способом измерить уровень неопределённости переменной целевой переменной y в вершине дерева может быть её энтропия .

Определим количество информации, которую мы получаем при случайном событии с вероятностью p

$$Info\{\text{событие с вероятностью } p\} = -ln\, p$$



Энтропийный критерий

Тогда энтропия случайной величины y будет равна ожидаемому количеству информации, которую мы получим, узнав реализацию этой случайной величины:

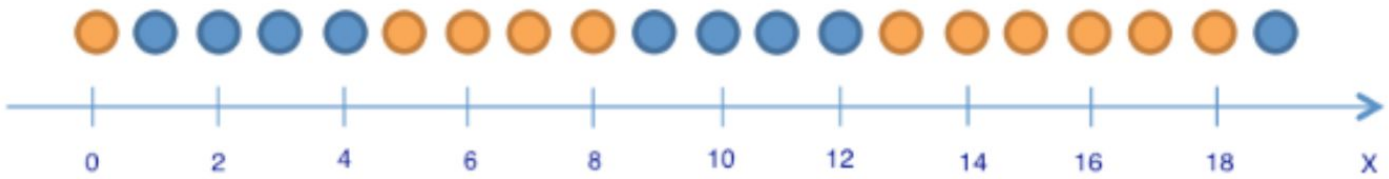
$$Entropy(y) = \mathbb{E}[Info(y)] = -\sum_{i=1}^{k} p_i \cdot ln\, p_i$$

Можно доказать, что энтропия:

- Достигает своего максимума, когда все классы равновероятны
- Достигает своего минимума, когда реализуется только один из классов

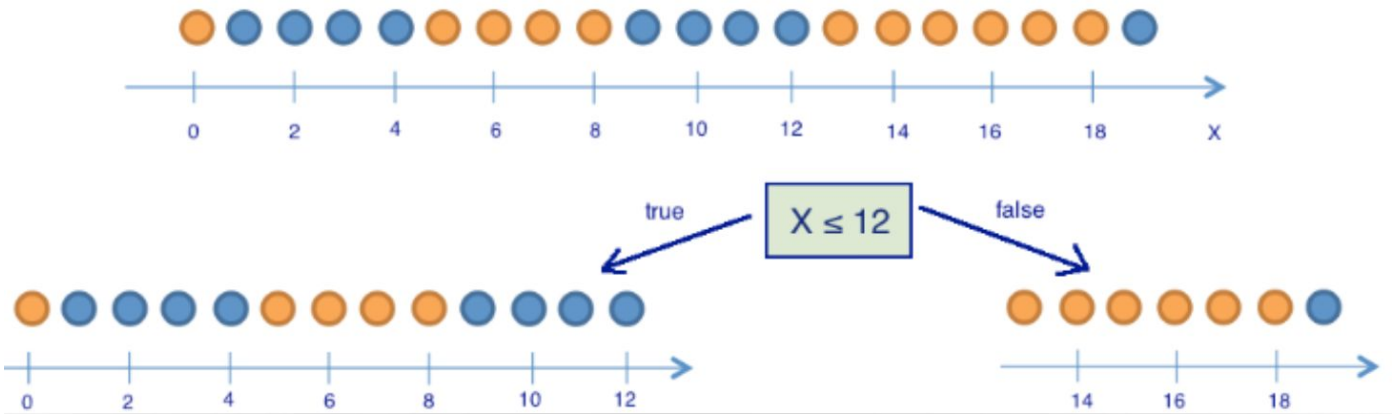
Пример использования энтропийного критерия

Пусть дана обучающая выборка с одним одномерным признаком x:



$$\begin{cases} p_1 = \frac{9}{20} \\ p_2 = \frac{11}{20} \end{cases} \Rightarrow H(R) = -\frac{9}{20} ln\ \frac{9}{20} - \frac{11}{20} ln \frac{11}{20} \approx 1$$

Пример использования энтропийного критерия



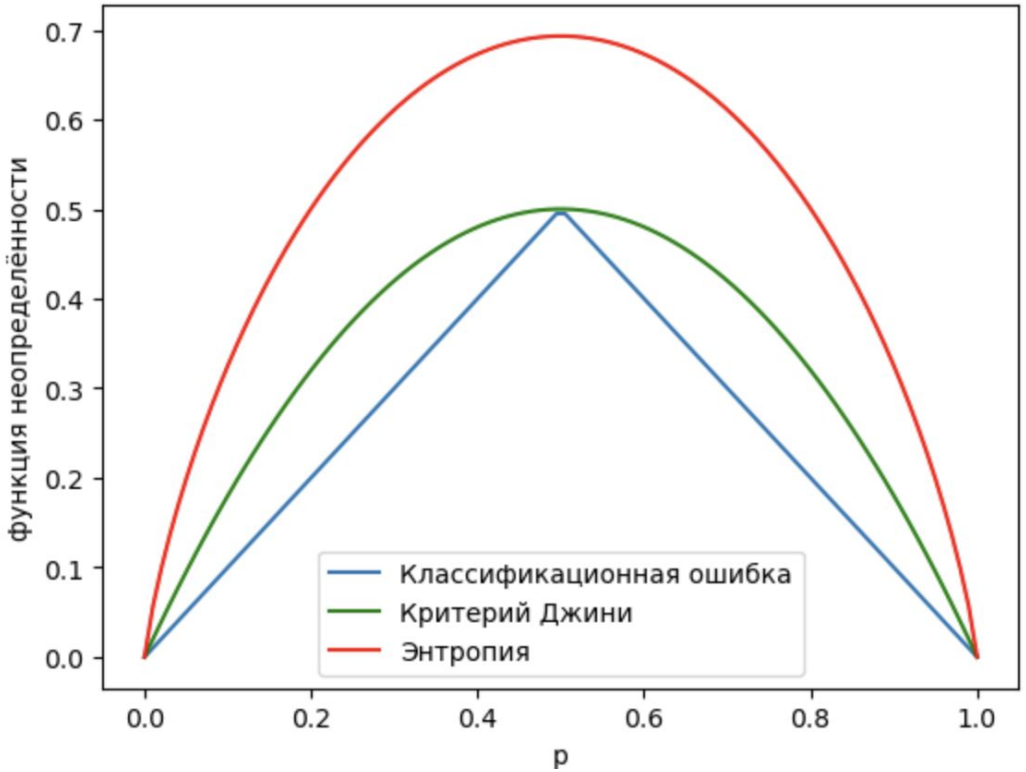
$$H(R_l) = -\frac{5}{13}\, ln\frac{5}{13} - \frac{8}{13}\, ln\frac{8}{13} \approx 0.96$$

$$H(R_r) = -\frac{1}{7}\, ln\frac{1}{7} - \frac{6}{7}\, ln\frac{6}{7} \approx 0.6$$

$$Q(X, j, t) = H(R) - \frac{|R_l|}{|R|} \cdot H(R_l) - \frac{|R_r|}{|R|} \cdot H(R_r) = 1 - \frac{13}{20} \cdot 0.96 - \frac{7}{20} \cdot 0.6 \approx 0.16$$

Графики критериев информативности

Для двух классов можно построить графики информативности



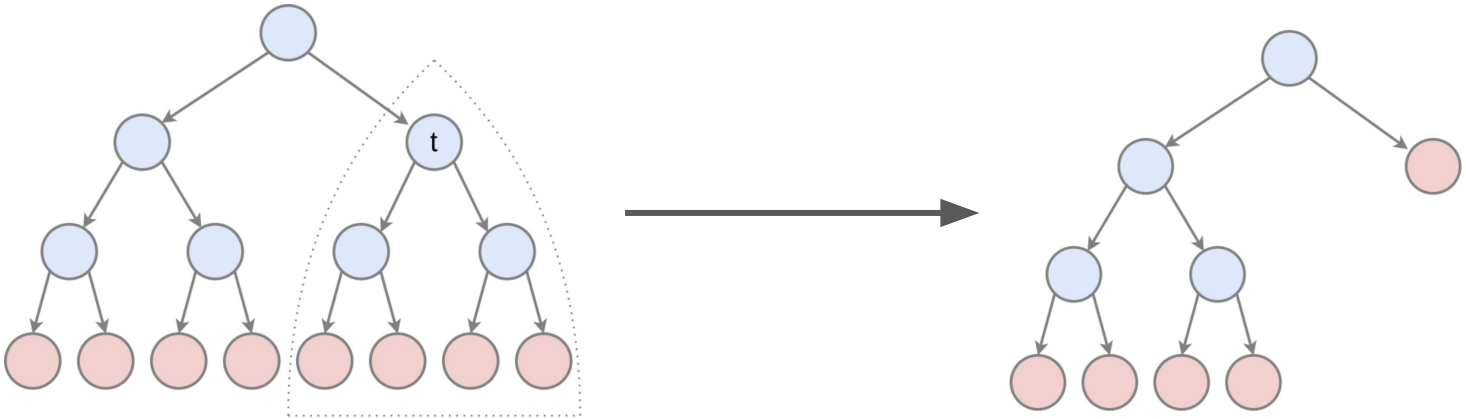
Стрижка дерева

Для борьбы с переобучением рассмотренных ранее критериев остановки применяют ещё и метод о**брезки дерева** (tree pruning)

Идея метода заключается в том, чтобы

1. Построить переобученное дерево (в каждом листе один объект)
2. Произвести оптимизацию его структуры

Обрезка по минимальной цене



Вводится новый функционал $H_\alpha(T)$ на дереве T, как

$$H_{alpha}(T) = H(T) + \alpha \cdot M(T)$$

где M(T) – число листьев в дереве, α – параметр регуляризации

Обрезка по минимальной цене

Можно показать, что оптимальное значение α при замене некоторого поддерева T на одиночный лист t определяется по формуле (см. вывод):

$$\alpha^* = \frac{H(t) - H(T)}{M(T) - 1}$$

- Чем меньше α, тем целесообразней заменить некоторое поддерево T на лист t, поскольку тем ближе H(t) к H(T), и качество сильно не меняется

Обрезка по минимальной цене

В алгоритме обрезки по минимальной цене для каждого поддерева вычисляется своё значение α и ищется минимальное, то поддерево на котором оно было достигнуто обрезается и заменяется на лист

В итоге обрезки получается убывающая последовательность деревьев T_1, T_2, T_3, ….

$$T_1 \supset T_2 \supset \ldots \supset T_K = \text{корень первоначального дерева}$$

На валидации среди множества {T_1, T_2, …} выбирается, то на котором ошибка минимальна

Обработка пропущенных значений

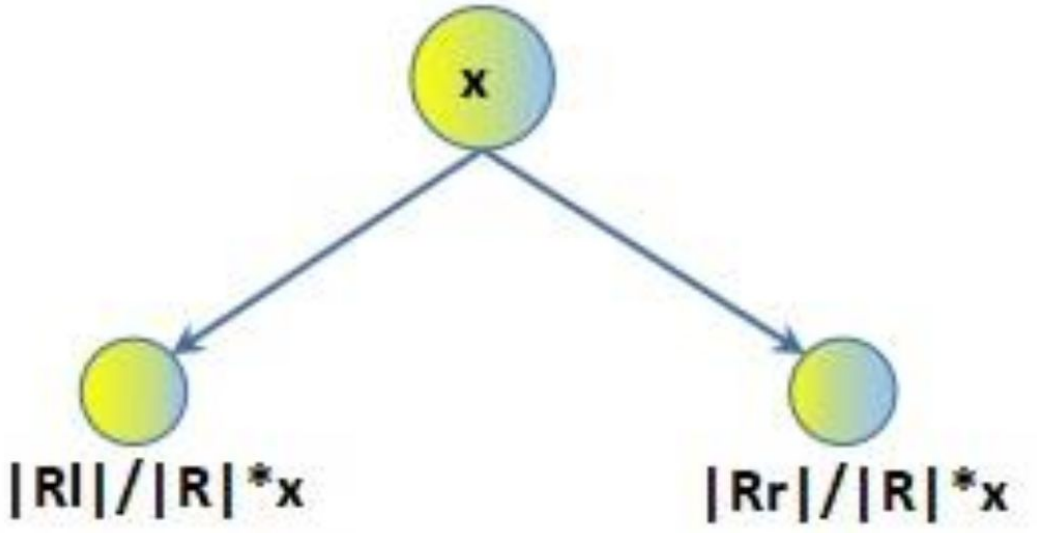
Одним из преимуществ решающих деревьев является их способность работать с пропущенными значениями, как во время **обучения** так и во время **прогноза**.

Пусть в выборке X есть подмножество V_j объектов, у которых пропущено значение j-го признака. Тогда при построении дерева будем вычислять функционал j-го признака игнорируя объекты из V_j:

$$Q(R, j, t) \approx \frac{|R \setminus V_j|}{|R|} Q(R \setminus V_j, j, t)$$

Обработка пропущенных значений

- При расщеплении по j-му признаку, поместим объекты из V_j как в левое, так и в правое поддеревья. Присвоим этим поддеревьям веса $|R_l|/|R|$ и $|R_r|/|R|$ соответственно



Получение предсказаний для пропущенных значений

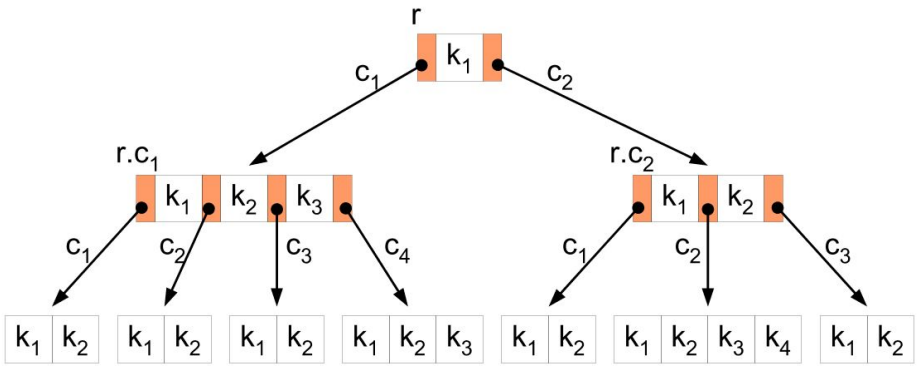
- Если на стадии прогноза в вершину v с расщеплением по j-ому признаку попал объект x, у которого пропущен данный признак, то прогноз a(x) строится по следующему правилу:

$$a(x) = \frac{|R_l|}{|R|} \cdot a_l(x) + \frac{|R_r|}{|R|} \cdot a_r(x)$$

где $a_l(x)$ и $a_r(x)$ – прогнозы в левом и правом поддереве

Обработка категориальных признаков

- Multi-way splits: можно разбивать вершину на столько поддеревьев, сколько различных значений имеет категориальный признак



В результате будут получаться деревья с большим числом листьев!

Разбиение на два подмножества

- Поделим множество возможных значений $U=\{u_1, \ldots, u_m\}$ признака x_j на два подмножества $U = U_1 \cup U_2$. В качестве функции расщепления будем использовать $\beta(x_j) = [x_j \in U_1]$.

При таком подходе существует $2^{m-1} - 1$ возможных вариантов построить два таких множества, и для поиска оптимального правила нужно их все перебрать

Обработка в бинарной классификации

- Рассмотрим вершину дерева. Пусть $N(u_1)$ – количество объектов в этой вершине со значением категориального признака $x_j = u_1$.

- Тогда $\frac{\sum_{x_{ij}=u_1} I[y_i=+1]}{N(u_1)}$ – доля положительных объектов, у которых $x_j=u_1$

- Упорядочим значения категориального признака x_j по возрастанию долей

$$\frac{\sum_{x_{ij}=u_{(1)}} I[y_i=+1]}{N(u_{(1)})} \leq \frac{\sum_{x_{ij}=u_{(2)}} I[y_i=+1]}{N(u_{(2)})} \leq \ldots \leq \frac{\sum_{x_{ij}=u_{(m)}} I[y_i=+1]}{N(u_{(m)})}$$

- Заменим категорию $u_{(i)}$ на число i и будем работать с ним как с вещественным признаком

Обработка в бинарной классификации

Утверждение.

Данный подход с использованием критерия Джини или Энтропии даёт тот же результат, как если бы происходил поиск оптимального разбиения $U = U_1 \cup U_2$

Плюсы деревьев решений

- Чёткие правила классификации (Например, правило “возраст” > 25)
- Деревья решений легко визуализируются и тем самым хорошо интерпретируются
- Быстро обучаются и быстро делают предсказание
- Малое число параметров

Минусы деревьев решений

- Очень чувствительны к шумам в данных модель сильно меняется при небольшом изменении обучающей выборки
- Склонны к переобучению, поэтому необходимо проводить тонкую настройку критериев останова или проводить стрижку
- Поиск оптимального дерева представляет собой вычислительно сложную NP-полную задачу