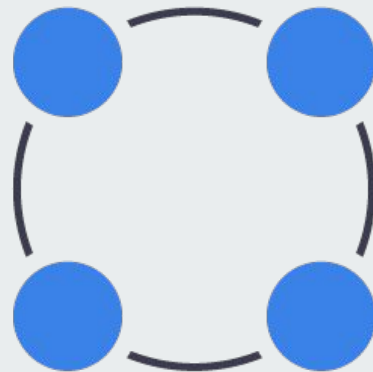




# Деревья решений

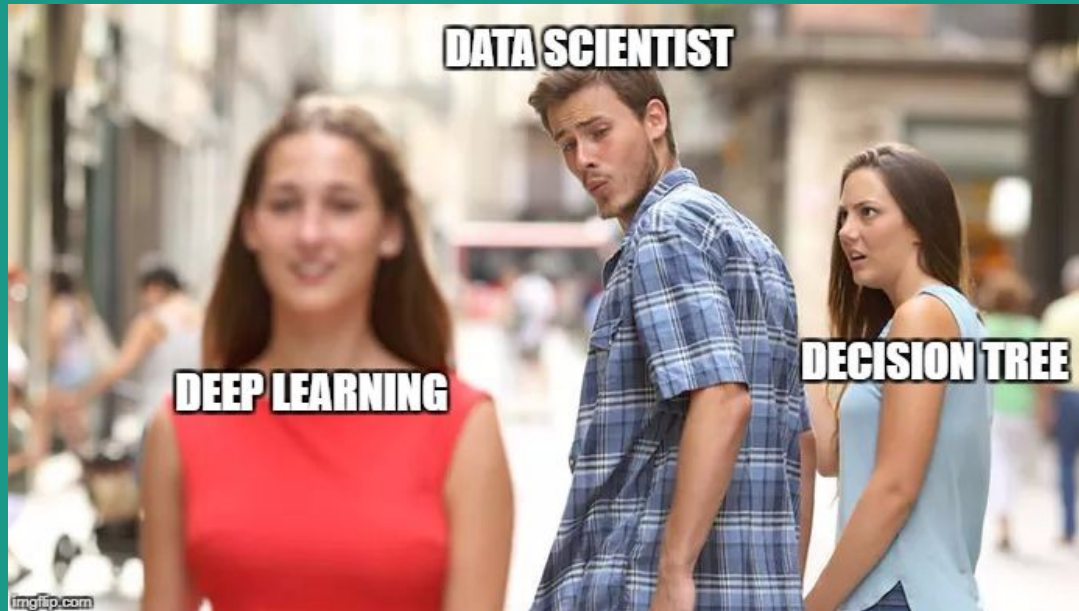
Лекция 4

(26.02.2022)



---

# Деревья решений

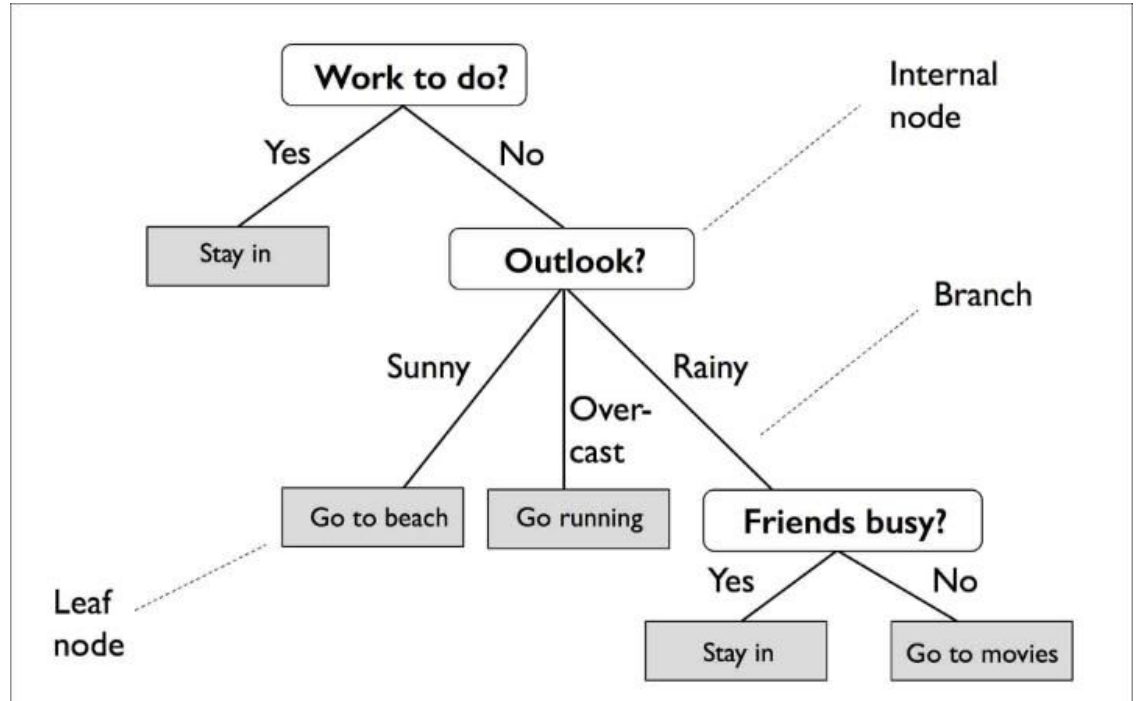


# Как выглядит дерево решений

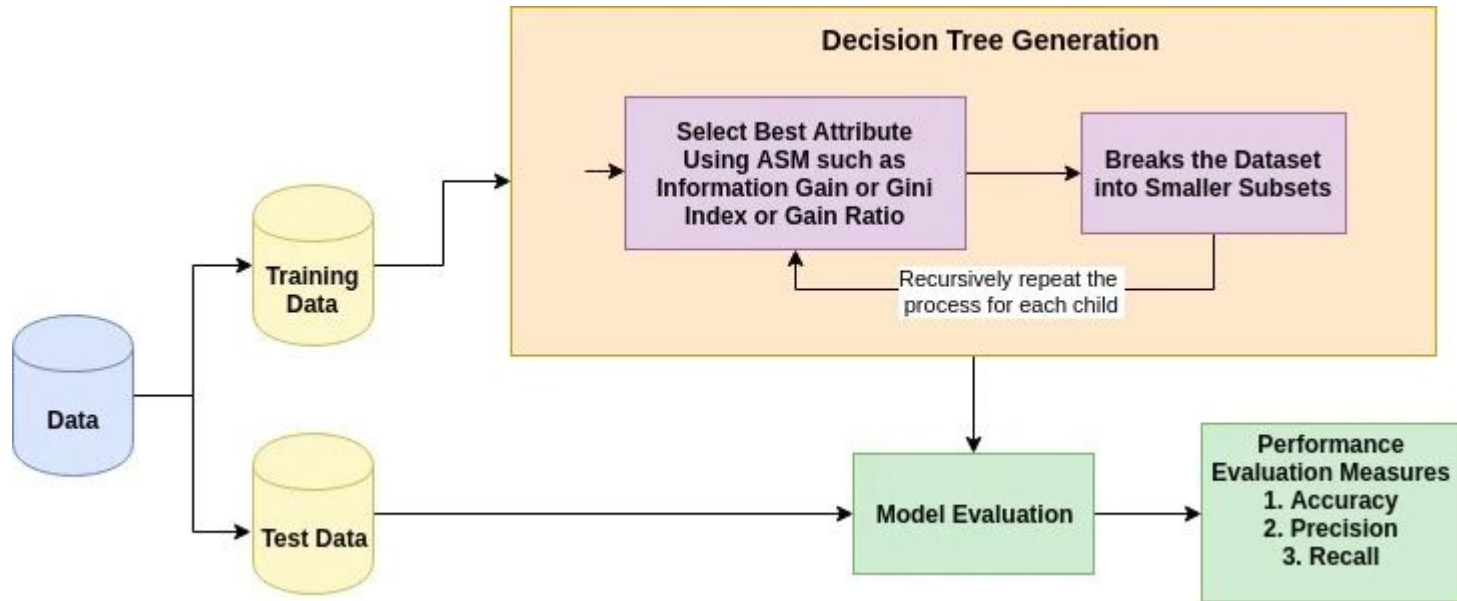


# Примерный алгоритм построения

Объединение логических правил вида “Значение признака А меньше Х и значение признака В меньше признака Y



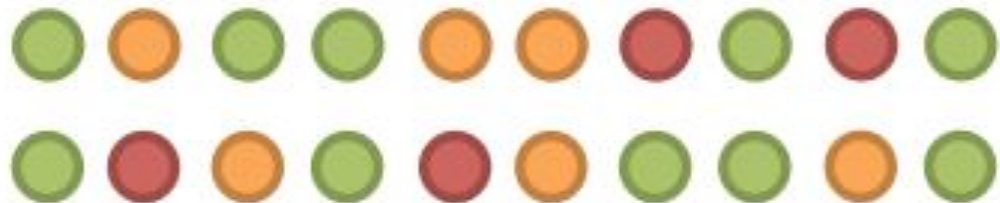
# Примерный алгоритм построения



---

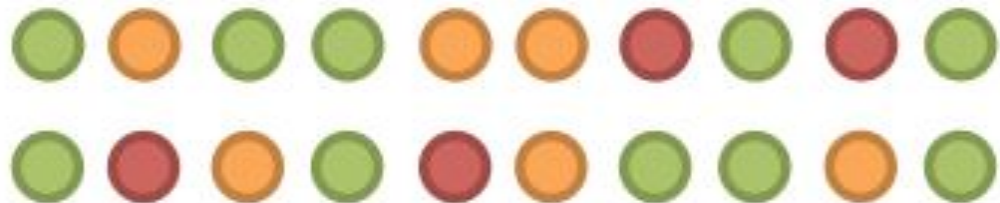
**С какого признака начать**

# Комбинаторная энтропия



Давайте посчитаем количество различных перестановок, учитывая что шарики одного цвета — неразличимы.

# Комбинаторная энтропия



Давайте посчитаем количество различных перестановок, учитывая что шарики одного цвета — неразличимы.

$$W = 10! / 5! * 2! * 3!$$



# Комбинаторная энтропия

Все перестановки можно пронумеровать числами от 0 до  $(W - 1)$ . Следовательно, строка из  $\log_2(W)$  бит однозначно кодирует каждую из перестановок.

Поскольку перестановка состоит из  $N$  шариков, то среднее количество бит, приходящихся на один элемент перестановки можно выразить как

$$S_{comb} = \frac{\log_2(W)}{N} = \frac{1}{N} \cdot \log_2 \left( \frac{N!}{\prod N_i!} \right) = \frac{1}{N} \cdot \log_2 \left( \frac{N!}{N_1! \cdot N_2! \cdot N_3! \cdot \dots} \right)$$

# Энтропия Шеннона



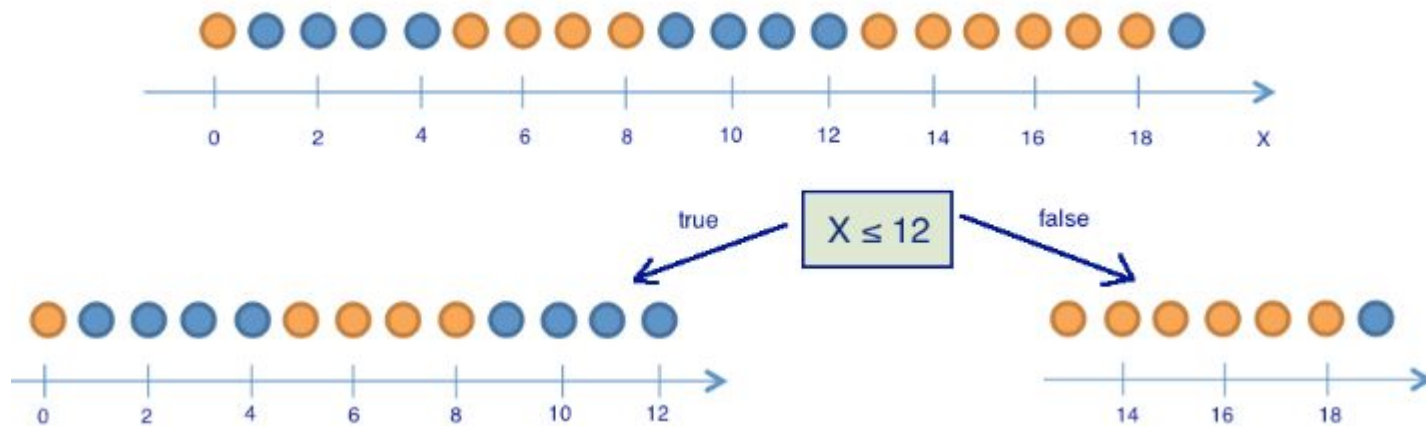
$N$  - количество возможных состояний системы

$p_i$  - вероятность нахождения системы в  $i$ -ом состоянии

По сути - оценка упорядоченности системы (1 - система полностью не упорядочена)

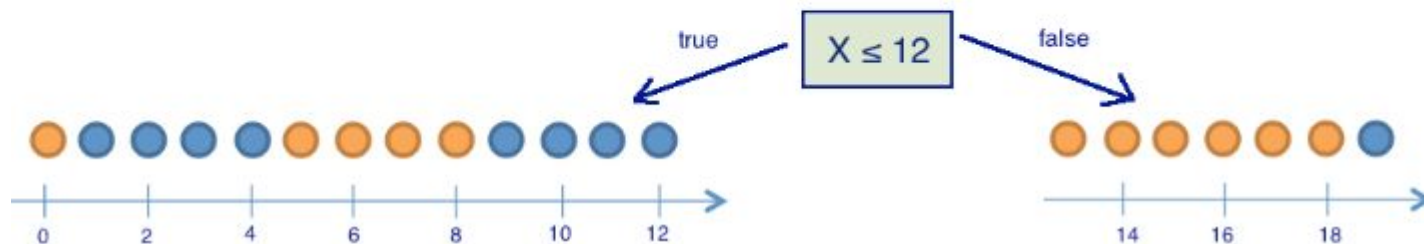
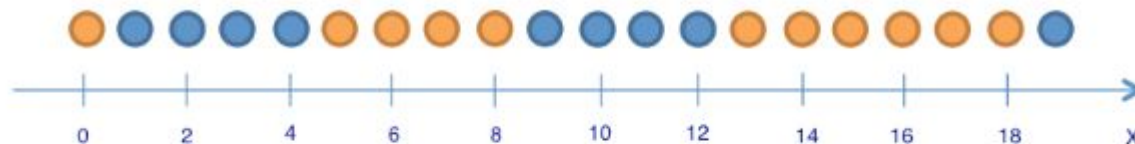
$$S = - \sum_{i=1}^N p_i \log_2 p_i$$

# Энтропия Шеннона - пример



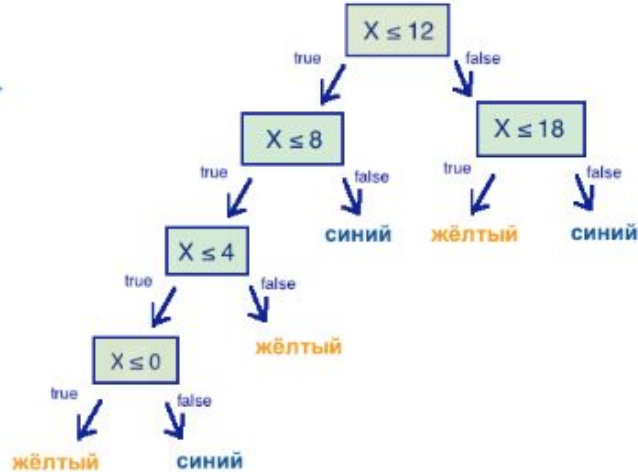
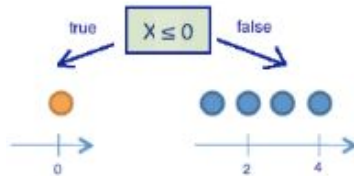
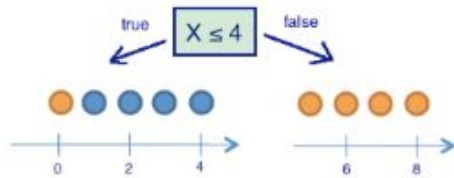
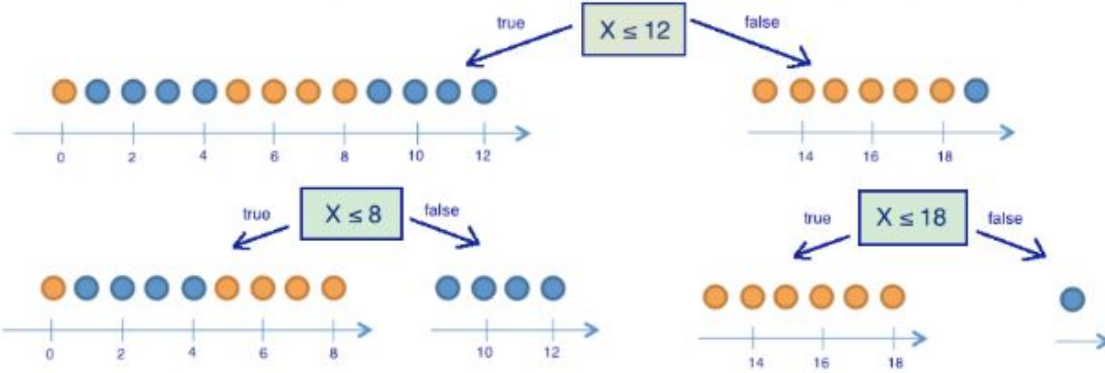
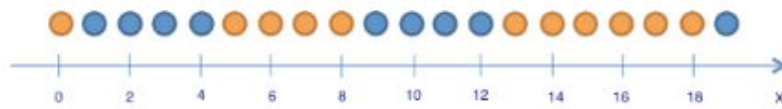
# Энтропия Шеннона - пример

$$S_0 = -\frac{9}{20}\log_2 \frac{9}{20} - \frac{11}{20}\log_2 \frac{11}{20} \approx 1.$$



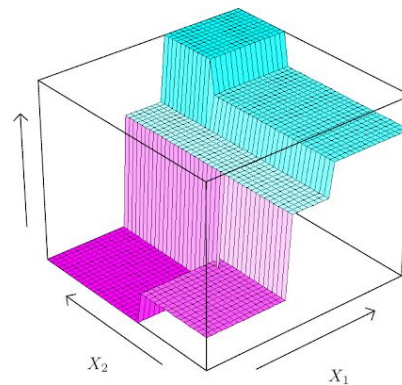
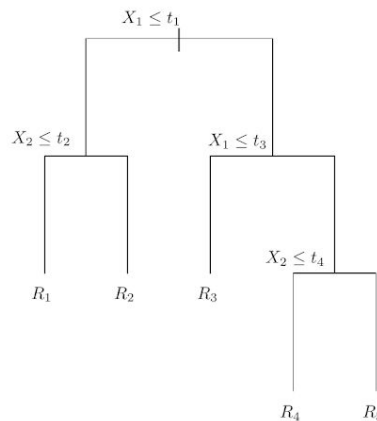
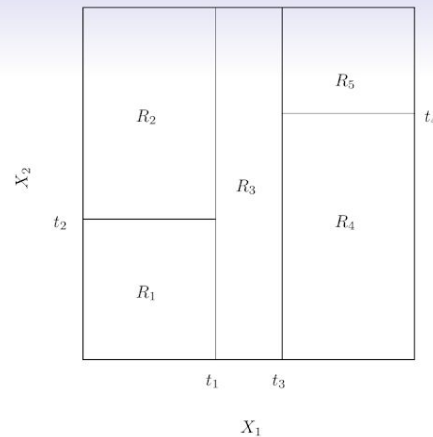
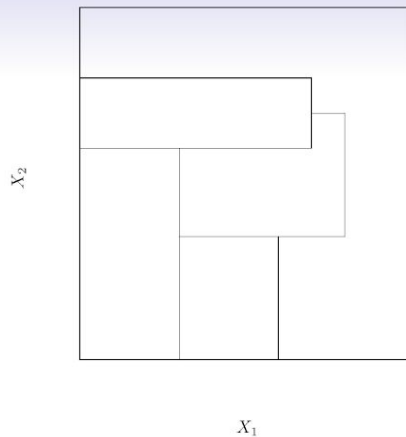
$$S_1 = -\frac{5}{13}\log_2 \frac{5}{13} - \frac{8}{13}\log_2 \frac{8}{13} \approx 0.96$$

$$S_2 = -\frac{1}{7}\log_2 \frac{1}{7} - \frac{6}{7}\log_2 \frac{6}{7} \approx 0.6$$



# Как это выглядит в пространстве

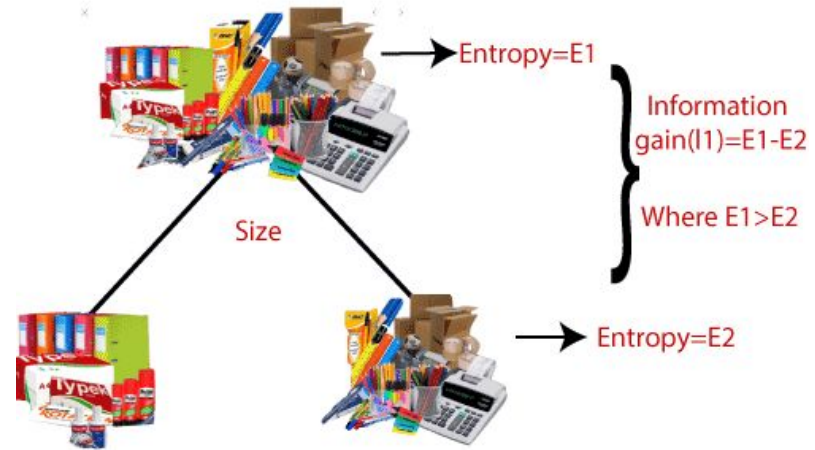
??? Можно ли придумать  
ситуацию, когда дерево не  
сможет достичь энтропии  
равной нулю



# Information gain

$$IG(Q) = S_O - \sum_{i=1}^q \frac{N_i}{N} S_i$$

Прирост информации = уменьшение энтропии



## Information gain для случая больше 12



$$IG(x \leq 12) = S_0 - \frac{13}{20}S_1 - \frac{7}{20}S_2 \approx 0.16$$

??? Чему равна энтропия с группами шарика одного цвета



# Другие критерии качества



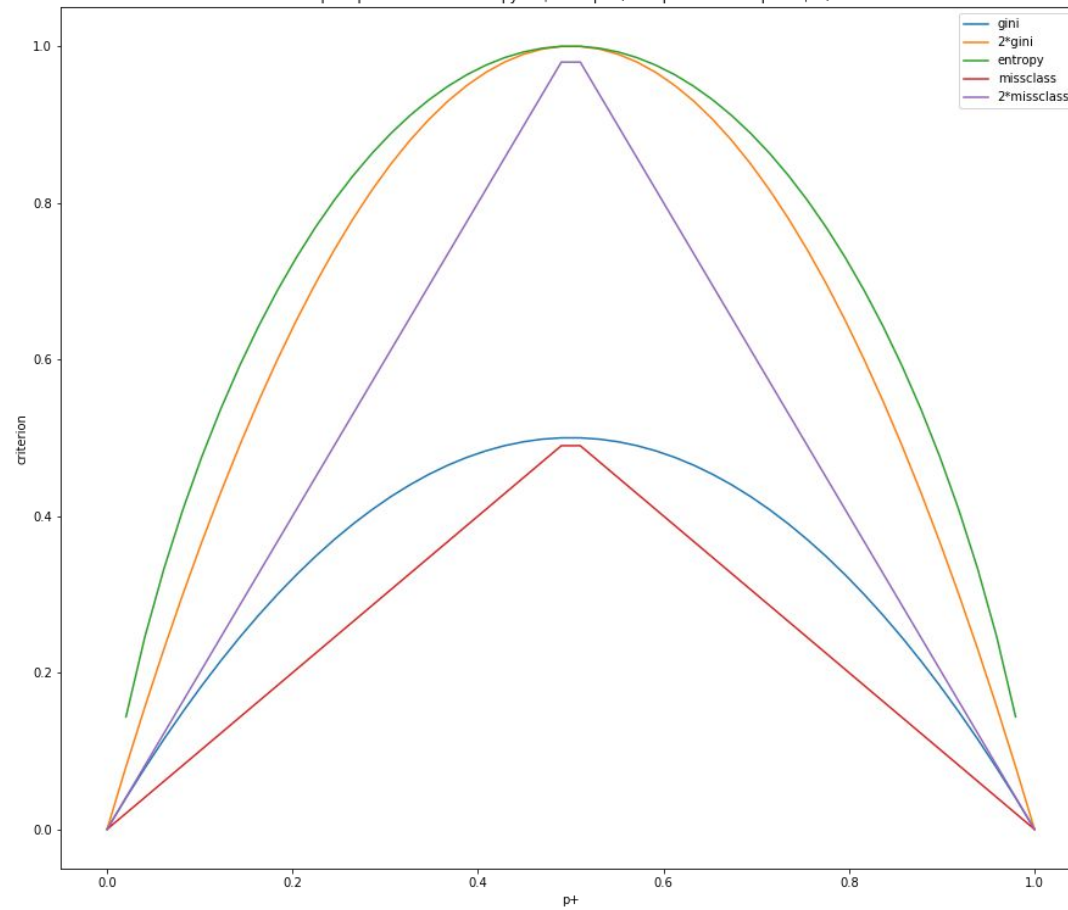
Неопределенность Джини - максимизацию этого критерия можно интерпретировать как максимизацию числа пар объектов одного класса, оказавшихся в одном поддереве.

$$G = 1 - \sum_k (p_k)^2$$


Ошибка классификации:

$$E = 1 - \max_k p_k$$

Критерии качества как функции от  $p_+$  (бинарная классификация)



# Суммарный алгоритм построения дерева

1.   $s_0$  = вычисляем энтропию исходного множества
2. Если  $s_0 == 0$  значит:
  - Все объекты исходного набора, принадлежат к одному классу
  - Сохраняем этот класс в качестве листа дерева
3. Если  $s_0 \neq 0$  значит:
  - Перебираем все элементы исходного множества:
    - Для каждого элемента перебираем все его атрибуты:
      - На основе каждого атрибута генерируем предикат, который разбивает исходное множество на два подмножества
      - Рассчитываем среднее значение энтропии
      - Вычисляем  $\Delta S$
      - Нас интересует предикат, с наибольшим значением  $\Delta S$
      - Найденный предикат является частью дерева принятия решений, сохраняем его
  - Разбиваем исходное множество на подмножества, согласно предикату
  - Повторяем данную процедуру рекурсивно для каждого подмножества

# Когда остановиться

- Ограничение максимальной глубины дерева.
- Ограничение минимального числа объектов в листе.
- Ограничение максимального количества листьев в дереве.
- Остановка в случае, если все объекты в листе относятся к одному классу.
- Требование, что функционал качества при дроблении улучшился как минимум на  $s$  процентов

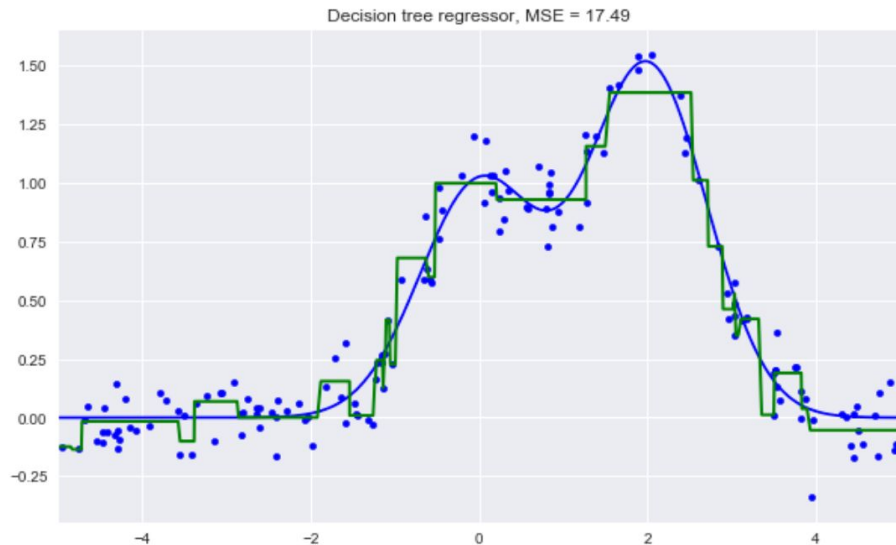


# Методы построения деревьев\*



- ID3: использует энтропийный критерий. Строит дерево до тех пор, пока в каждом листе не окажутся объекты одного класса, либо пока разбиение вершины дает уменьшение энтропийного критерия.
- C4.5: использует критерий Gain Ratio (нормированный энтропийный критерий). Критерий останова — ограничение на число объектов в листе. Обработка пропущенных значений - игнорирование их, а затем перенос таких объектов в оба поддерева с определенными весами.
- CART: использует критерий Джини. Стрижка осуществляется с помощью CostComplexity Pruning. Для обработки пропусков используется метод суррогатных предикатов.

# Задачи регрессии



$$D = \frac{1}{\ell} \sum_{i=1}^{\ell} (y_i - \frac{1}{\ell} \sum_{i=1}^{\ell} y_i)^2$$

минимизируя дисперсию вокруг среднего, мы ищем признаки, разбивающие выборку таким образом, что значения целевого признака в каждом листе примерно равны.

# Плюсы



- + Деревья очень легко объяснить людям. На самом деле их легче объяснить, чем линейную регрессию!
- + Некоторые люди считают, что деревья решений более точно отражают принятие решений человеком, чем регрессия или классификационные подходы,
- + Деревья могут быть отображены графически, и легко интерпретируется даже не экспертом (особенно если они маленькие).
- + Деревья могут легко обрабатывать качественные предикторы без нужды в создании фиктивных переменных.

# Минусы



- У порождения четких правил классификации есть и другая сторона: деревья очень чувствительны к шумам во входных данных -- можно изменить построенное дерево (решение - ансамбли)
- Разделяющая граница, построенная деревом решений, имеет свои ограничения (состоит из гиперплоскостей, перпендикулярных какой-то из координатной оси), и на практике дерево решений по качеству классификации уступает некоторым другим методам;
- Проблема поиска оптимального дерева решений (минимального по размеру и способного без ошибок классифицировать выборку) NP-полна, поэтому на практике используются эвристики типа жадного поиска признака с максимальным приростом информации, которые не гарантируют нахождения глобально оптимального дерева;
- Сложно поддерживаются пропуски в данных (на поддержку пропусков в данных ушло около 50% кода CART )
- Модель умеет только интерполировать, но не экстраполировать (это же верно и для леса и бустинга на деревьях).



# Материалы



1. <https://habr.com/en/post/116385/>
2. [https://www.youtube.com/watch?v=gV2cBLxQ\\_EQ&list=PLEqoHzpnmTfDwuwrFHWVHdr1-qJsfgCUX&index=5](https://www.youtube.com/watch?v=gV2cBLxQ_EQ&list=PLEqoHzpnmTfDwuwrFHWVHdr1-qJsfgCUX&index=5)
3. <https://habr.com/en/post/171759/>