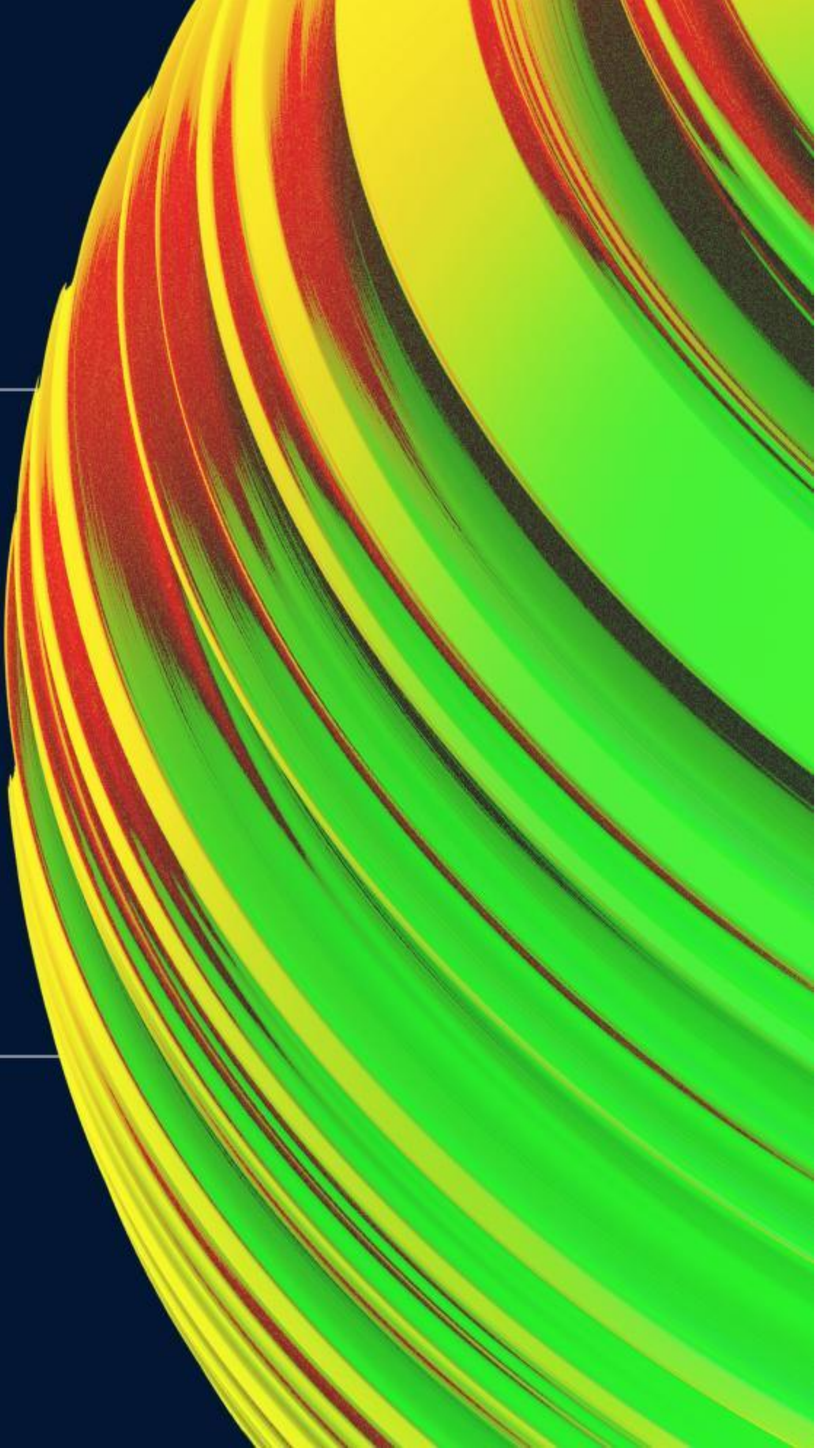
УНИВЕРСИТЕТ ИННОПОЛИС

Деревья решений

Воробьёва Мария

- maria.vorobyova.ser@gmail.com
- @SparrowMaria



План лекции

INNOPOLIS UNIVERSITY

1) Что такое дерево решений
2) Как строится дерево решений
3) Демо

Решающее правило

1) Если у клиента зарплата больше 10 тыс и у клиента есть в собственности недвижимость и авто, то выдаем кредит

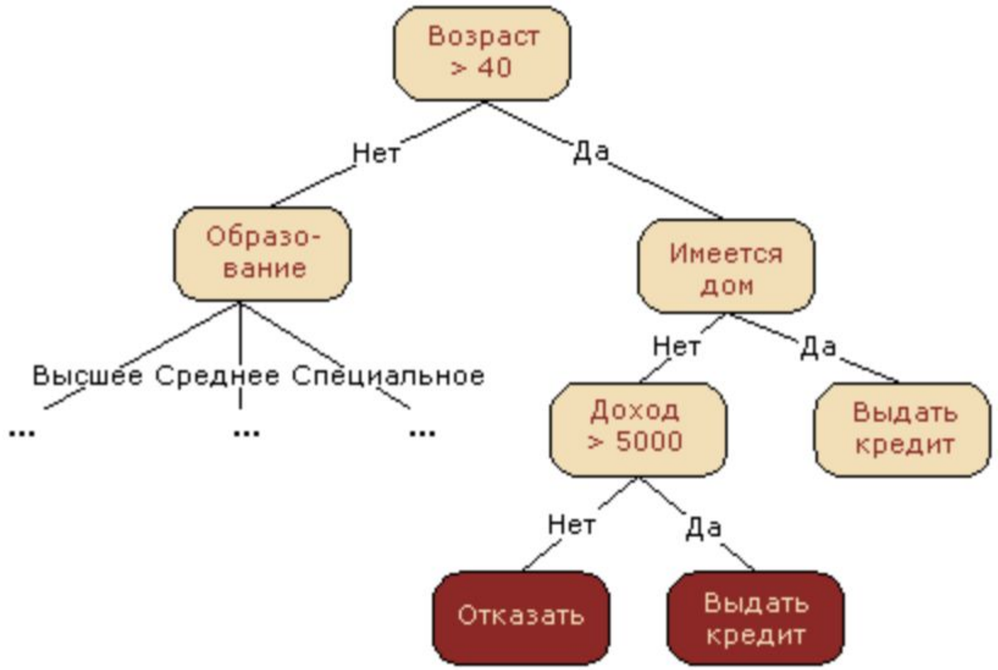
1) Если стоимость продукта снизилась на 10% и срок годности в норме, то покупаем продукт

1) Если на складе сегодня бесплатная приемка и стоимость хранения менее 10 коп за 1 литр, то везем поставку на склад

1) Если в ДМС есть клиника “Скандинавия” и средний возраст страхователя больше 60 лет, то стоимость страховки 100 тыс рублей

Что такое дерево решений

Дерево решений - это алгоритм машинного обучения, представляющий собой совокупность последовательных правил. Дерево решений можно отнести к **логическим алгоритмам**

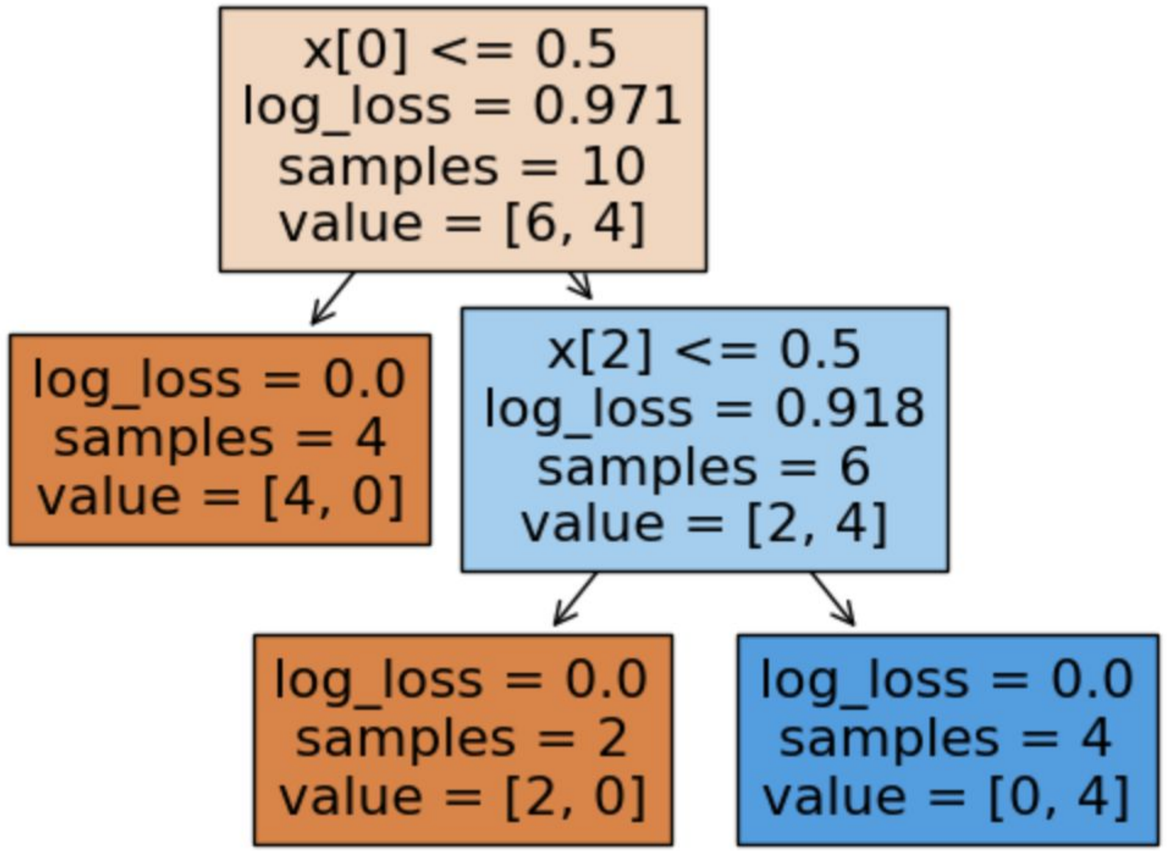


Дерево решений - это алгоритм, который объединяет логические правил вида "Значение признака меньше И Значение признака меньше … => Класс 1" в структуру данных "Дерево".

https://habr.com/ru/companies/ods/articles/322534/

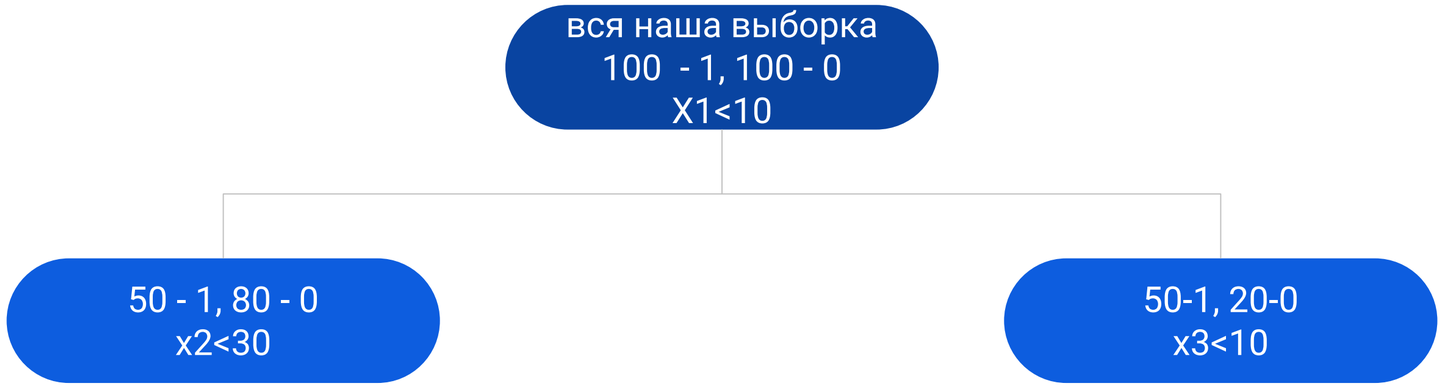
Как устроено дерево

1. Дерево состоит из **корня**, это стартовый узел
2. Далее в каждом **узле** задается вопрос: признак больше/меньше определенного значения
3. Узел, после которого нет разделения, называется **терминальным узлом или листом**



Как обучается дерево

На каждом следующем этапе мы разбиваем выборку так, чтобы концентрация того или иного класса была больше. **1 правило:** x1<10 → разбиваем выборку на 2 подвыборки и для каждой считаем количество 1 и 0

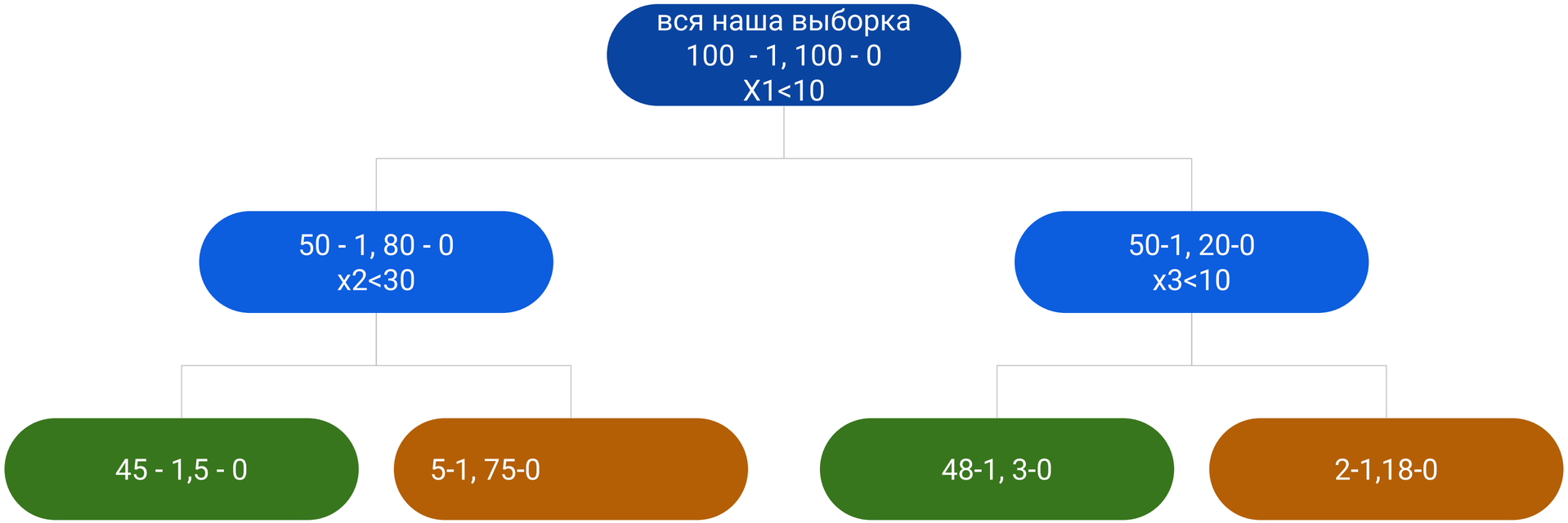


Как обучается дерево

Далее идем влево и применяем уже другое правило: x2<30 и снова считаем количество 1 и 0

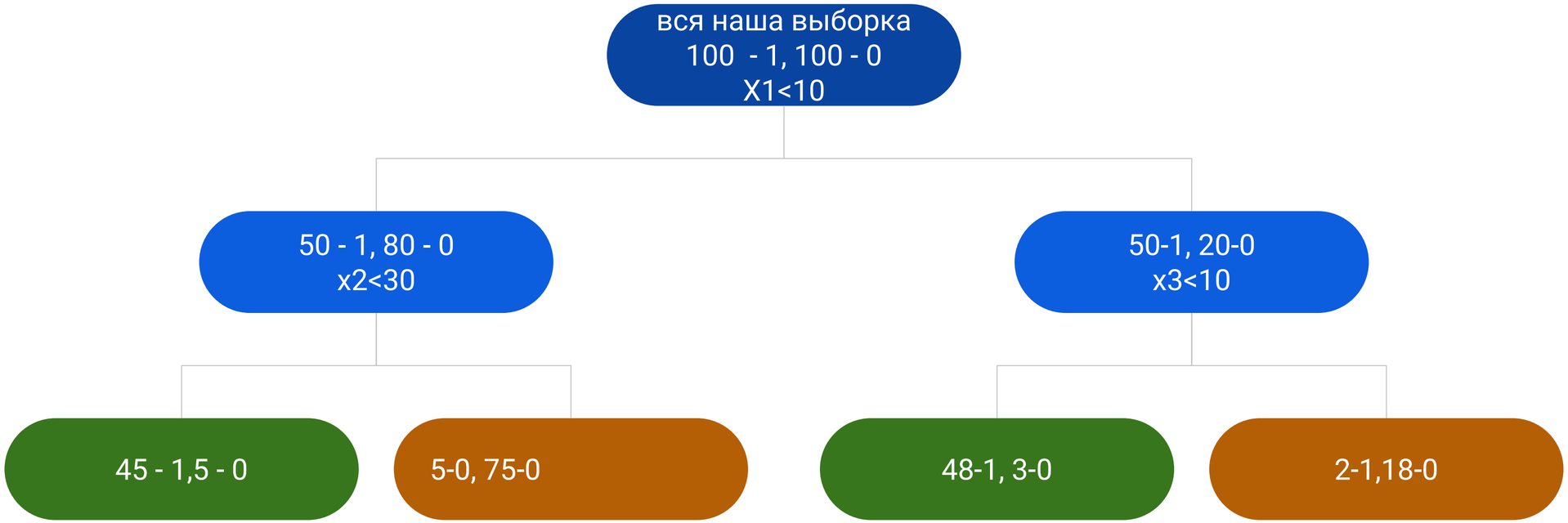
Как обучается дерево

Далее идем вправо и применяем уже другое правило: x3<10 и снова считаем количество 1 и 0. Дальше разбиение не имеет уже смысла, так как в терминальных узлах (листах) выделились классы-лидеры



Как обучается дерево

Мы экспертно разбили наши данные достаточно точно. Интересно, а есть ли возможность оценить насколько правило(предикат) хорошее? Может, мы могли еще точнее построить дерево?



Энтропия Шеннона

Энтропия Шеннона - это мера информационной неопределенности в системе или случайной величине. Она основывается на теории информации и используется в различных областях, включая информатику, статистику, машинное обучение и теорию вероятности.

Чем выше значение энтропии Шеннона, тем больше неопределенности содержится в системе или данных. Когда все возможные значения **равновероятны**, энтропия достигает своего **максимального значения,** и система имеет наибольший уровень разброса или неопределенности.

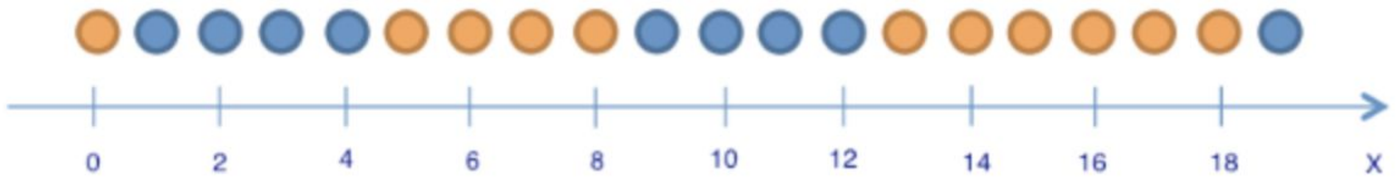
$$S = -\sum_{i=1}^{N} p_i \log_2 p_i$$

Как обучается дерево

Рассмотрим простой пример 9 синих шариков и 11 желтых.

Если мы наудачу возьмем шарик, то он с вероятностью будет синим 9/20 и с вероятностью 11/20 – желтым

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



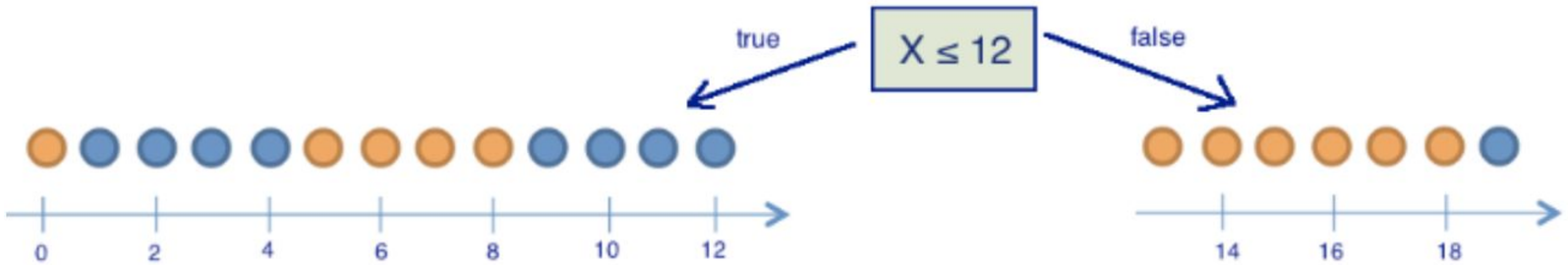
https://habr.com/ru/companies/ods/articles/322534/

Как обучается дерево

Теперь посмотрим, как изменится энтропия, если разбить шарики на две группы – с координатой меньше либо равной 12 и больше 12.

В левой группе оказалось 13 шаров, из которых 8 синих и 5 желтых. Энтропия этой группы равна $S_1 = -\frac{5}{13}\log_2\frac{5}{13} - \frac{8}{13}\log_2\frac{8}{13} \approx 0.96$

В правой группе оказалось 7 шаров, из которых 1 синий и 6 желтых. Энтропия правой группы равна $S_2 = -\frac{1}{7}\log_2\frac{1}{7} - \frac{6}{7}\log_2\frac{6}{7} \approx 0.6$



https://habr.com/ru/companies/ods/articles/322534/

Information gain

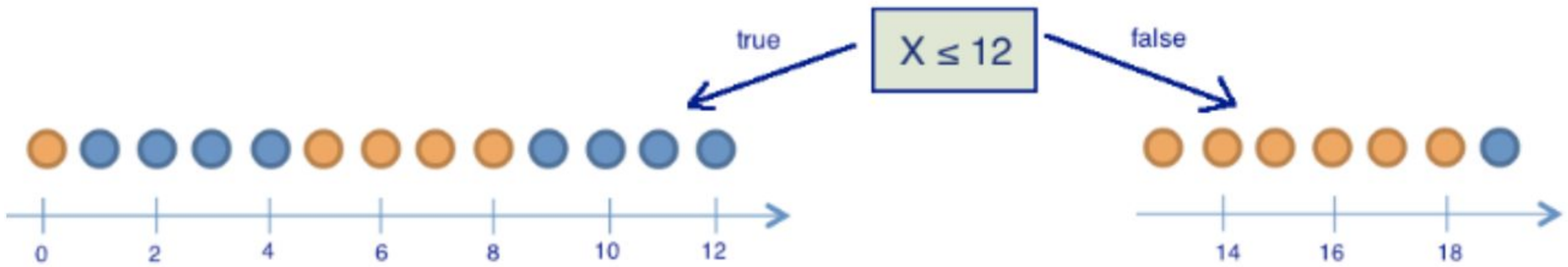
В нашем примере рассчитаем IG (Information gain)

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

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

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

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



Перебираем все возможные правила и выбираем такое правило, у которого IG наибольший

Information gain

Энтропия – степень хаоса (или неопределенности) в системе, уменьшение энтропии называют приростом информации. Прирост информации - information gain (IG)

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

где

q - число разбиений,

N_i - число элементов

S_0 - энтропия Шеннона до разбиения

S_i - энтропия Шеннона после разбиения в каждой группе i

Чем больше Information gain, тем лучше правило, то есть после разбиения это правило сильнее уменьшает информационную энтропию Шеннона

Как обучается дерево

Во время обучения мы можем протестировать все возможные правила:

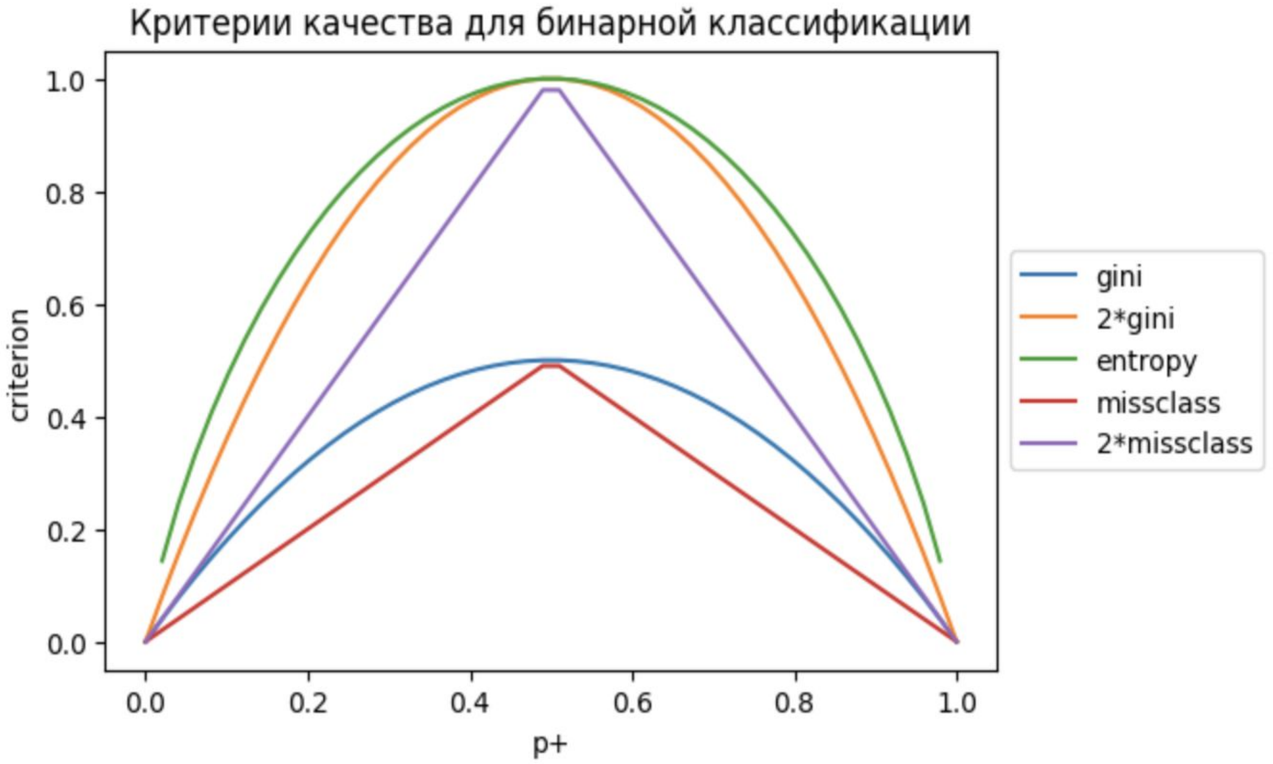
x<1, x<2 … x<19 и далее на основании прироста информации (IG) выбрать правило, у которого наибольший IG

Тестирование правил проходит сначала внутри одной переменной, а потом тестирование происходит между всеми переменными и выбирается наилучшее правило по IG

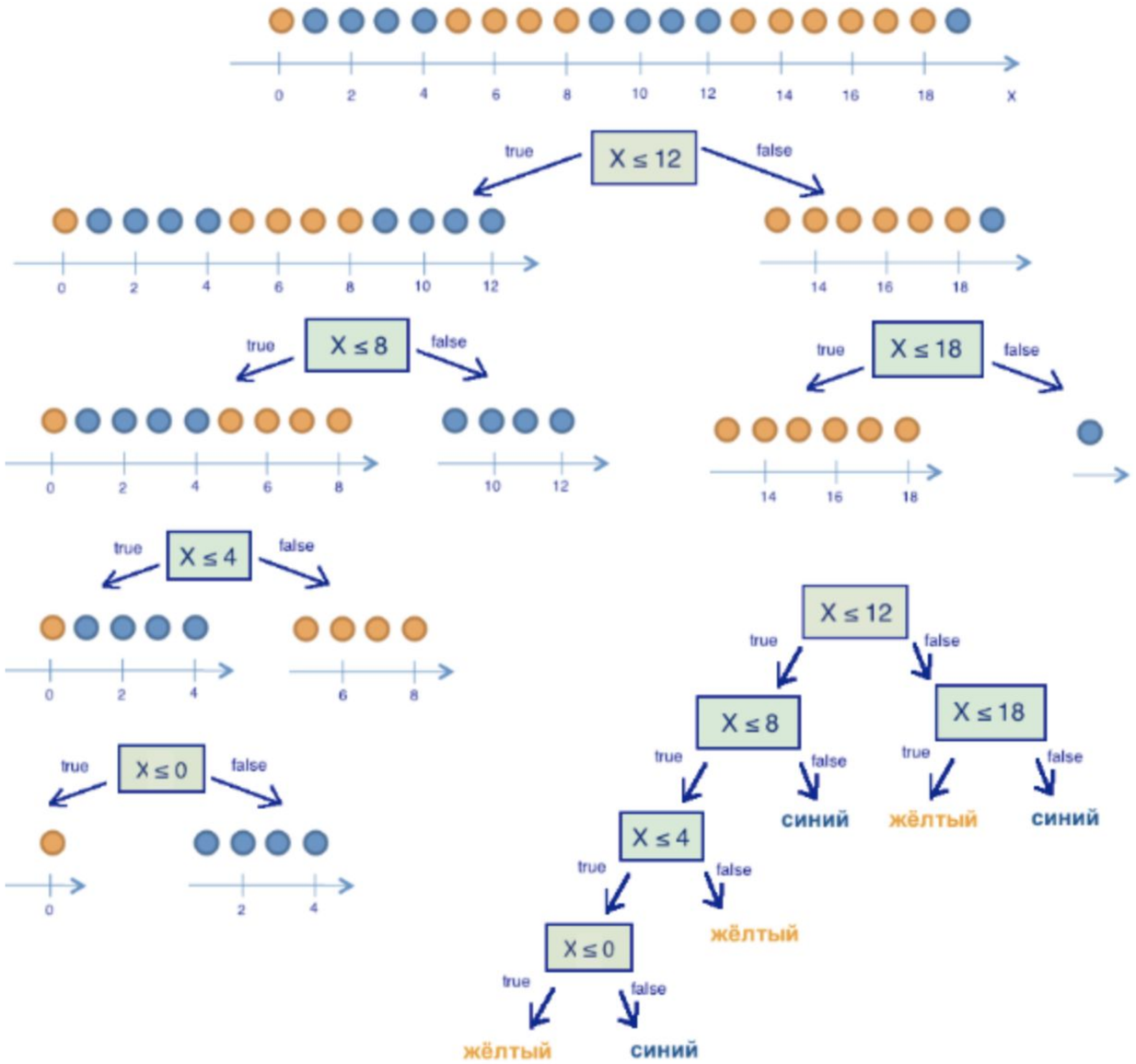
Как обучается дерево

Для оценки насколько правило (предикат) разумно используются обычно следующие метрики:

1) Information gain $IG(Q) = S_O - \sum_{i=1}^{q} \frac{N_i}{N} S_i$
2) Неопределенность Джини $G = 1 - \sum_k (p_k)^2$
3) Ошибка классификации (misclassification error) $E = 1 - \max_k p_k$



Как обучается дерево



Процесс обучения происходит итеративно до тех пор, пока не достигнуты критерии останова

Критерии качества. Обобщение

Запишем задачу формально: мы находимся в узле R_m и наша задача оптимально разбить на два подмножества:

$$H(R_m) - \frac{|R_l|}{|R_m|} H(R_l) - \frac{|R_r|}{|R_m|} H(R_r) \to \max$$

Функция $H(x)$ зависит от задачи, которую мы решаем:

Если **регрессия**:

$$\bar{y}_m = \frac{1}{n_m} \sum_{y \in Q_m} y$$

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} (y - \bar{y}_m)^2$$

$$median(y)_m = \underset{y \in Q_m}{\text{median}}(y)$$

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} |y - median(y)_m|$$

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} (y \log \frac{y}{\bar{y}_m} - y + \bar{y}_m)$$

Критерии качества. Обобщение

Запишем задачу формально: мы находимся в узле R_m и наша задача оптимально разбить на два подмножества:

$$H(R_m) - \frac{|R_l|}{|R_m|} H(R_l) - \frac{|R_r|}{|R_m|} H(R_r) \to \max$$

Функция $H(x)$ зависит от задачи, которую мы решаем

Если классификация:

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

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

Принцип жадной максимизации прироста информации

Принцип **жадной максимизации** прироста информации (greedy information gain maximization) заключается в том, что **оптимальность оценивается только на момент разбиения узла**, а не глобально для всего дерева. Это значит, что на каждом шаге **алгоритм выбирает локально наилучшее разделение** данных на основе текущего узла, без учета последствий выбора на более глобальном уровне

Этот принцип называется "жадным", потому что алгоритм принимает локально оптимальное решение на каждом шаге, стремясь получить максимальный прирост информации или уменьшение неопределенности на текущем уровне дерева. Он **не гарантирует**, что выбранные разделения на каждом уровне приведут к **глобально оптимальному дереву**. В некоторых случаях, жадный подход может привести к переобучению или недообучению.

УНИВЕРСИТЕТ
ИННОПОЛИС

ВОПРОСЫ И ОТВЕТЫ