Передовые инженерные школы

МИНОБРНАУКИ РОССИИ

УНИВЕРСИТЕТ ИННОПОЛИС

онлайн университет

Программная инженерия. Разработка ПО (Python для продвинутых специалистов. Машинное обучение)

Модуль: Предобработка данных и машинное обучение

Лекция 8: Деревья решений

Дата: 09.06.2025

План

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА
УНИВЕРСИТЕТА ИННОПОЛИС

Что мы уже изучили:

- как готовить данные для моделей машинного обучения
- линейная регрессия (для задачи регрессии) и логистическая регрессия (для задачи классификации)

Что планируем сегодня изучить:

- алгоритм машинного обучения: дерево решений

Решающее правило

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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

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

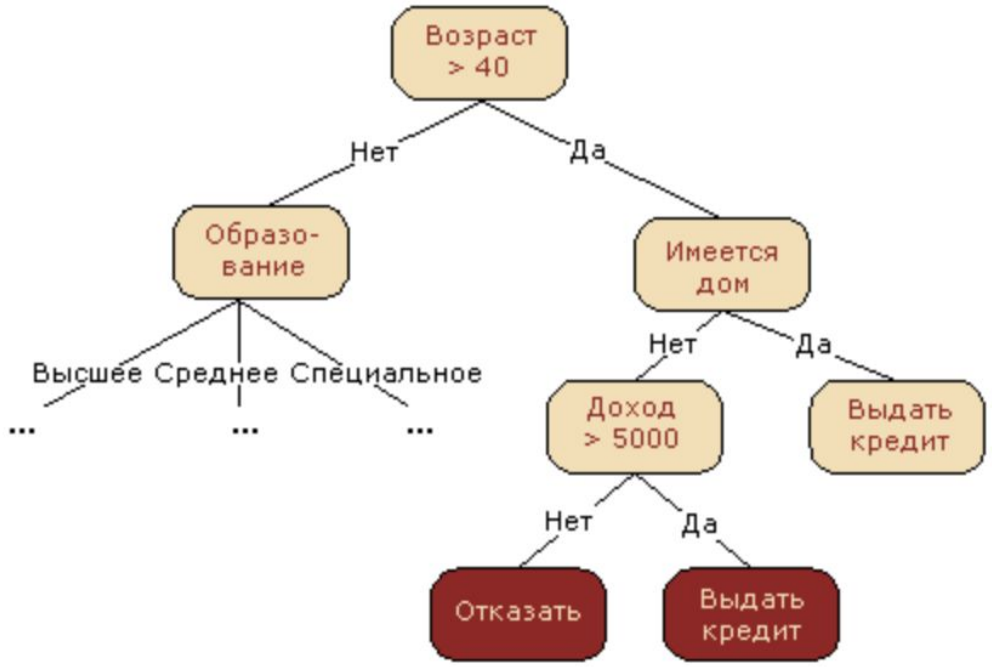
3) Если на складе сегодня бесплатная приемка и стоимость хранения менее 10 коп за 1 литр, то везем поставку на склад

4) Если в ДМС есть клиника “Скандинавия” и средний возраст страхователя больше 60 лет, то стоимость страховки 300 тыс рублей

Что такое дерево решений

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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



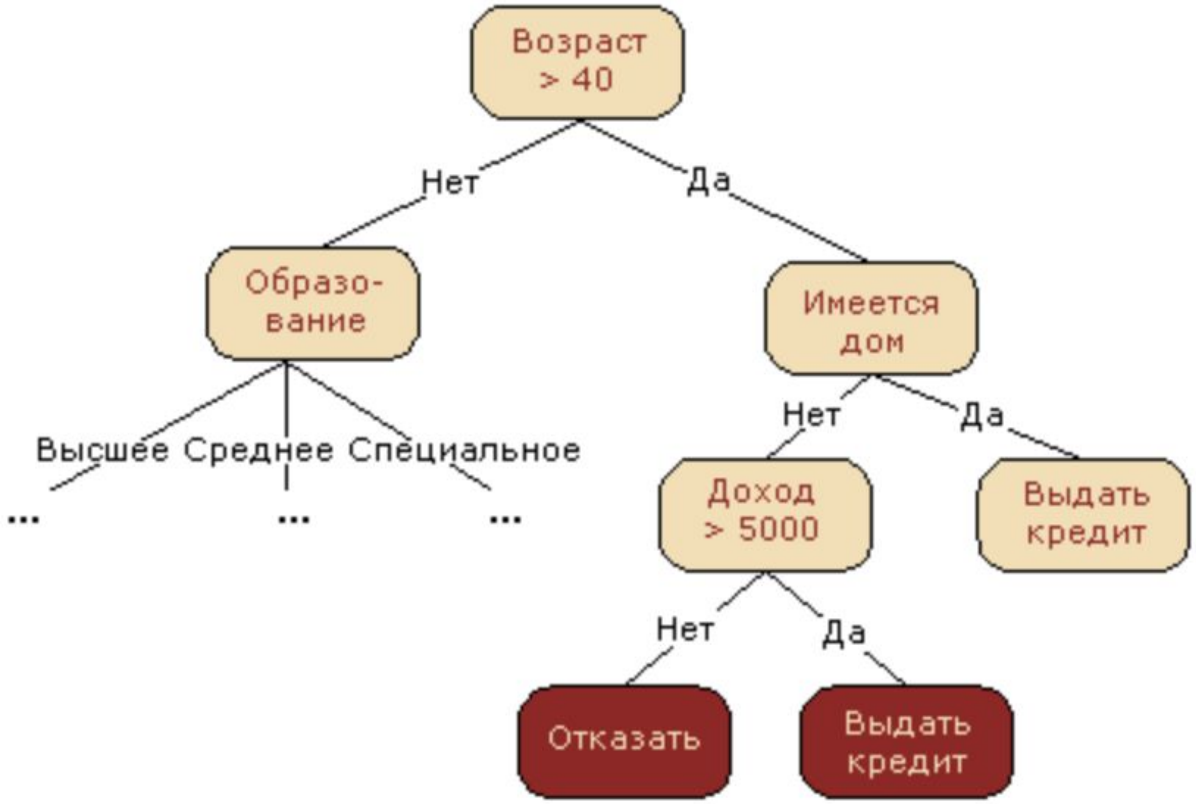
Дерево решений - это алгоритм, который объединяет логические правил вида "Значение признака меньше И Значение признака меньше … => Класс 1" в структуру данных "Дерево".

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

Как устроено дерево

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

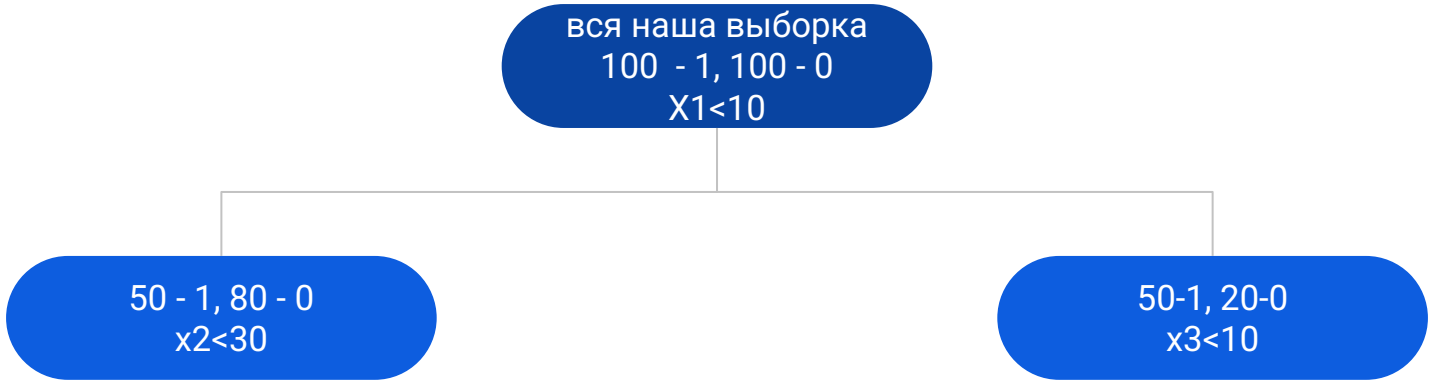
1. Дерево состоит из **корня**, это стартовый узел
2. Далее в каждом **узле** задается вопрос: признак больше/меньше определенного значения
3. Узел, после которого нет разделения, называется **терминальным узлом или листом**



Как обучается дерево

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

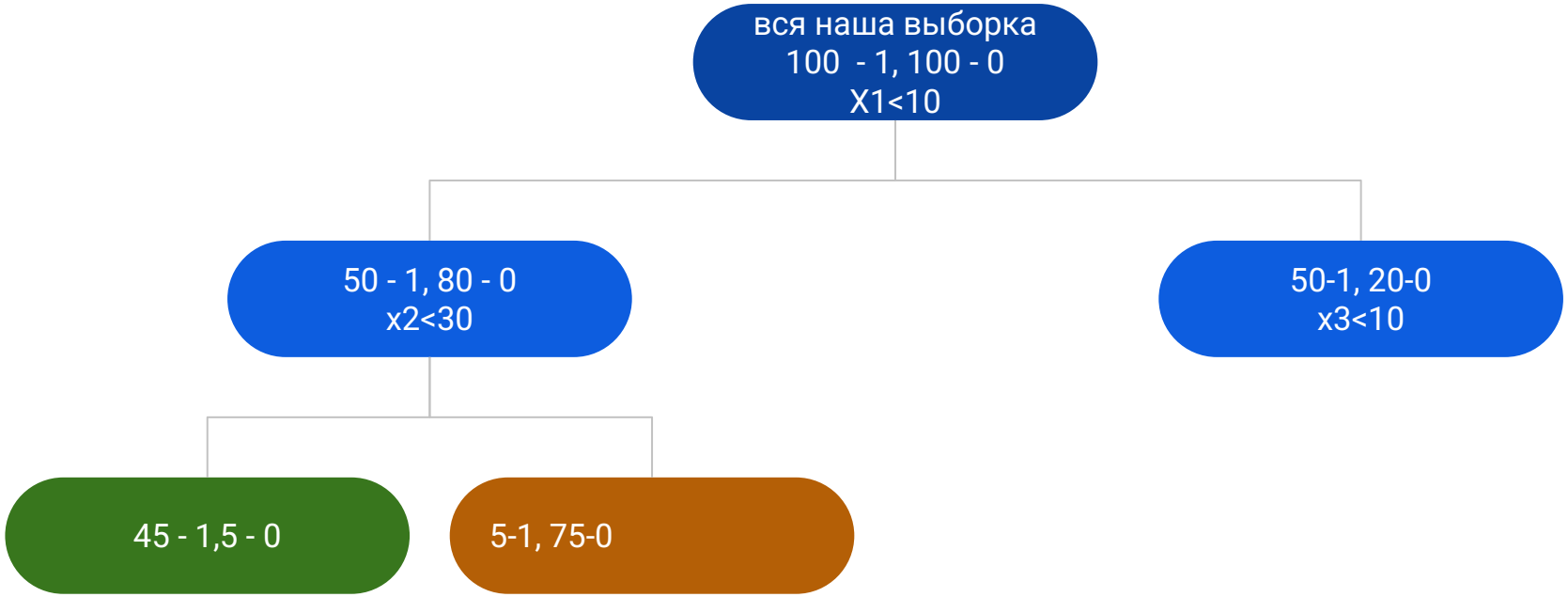
На каждом следующем этапе мы разбиваем выборку так, чтобы концентрация того или иного класса была больше. **1 правило:** x1<10 → разбиваем выборку на 2 подвыборки и для каждой считаем количество 1 и 0



Как обучается дерево

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

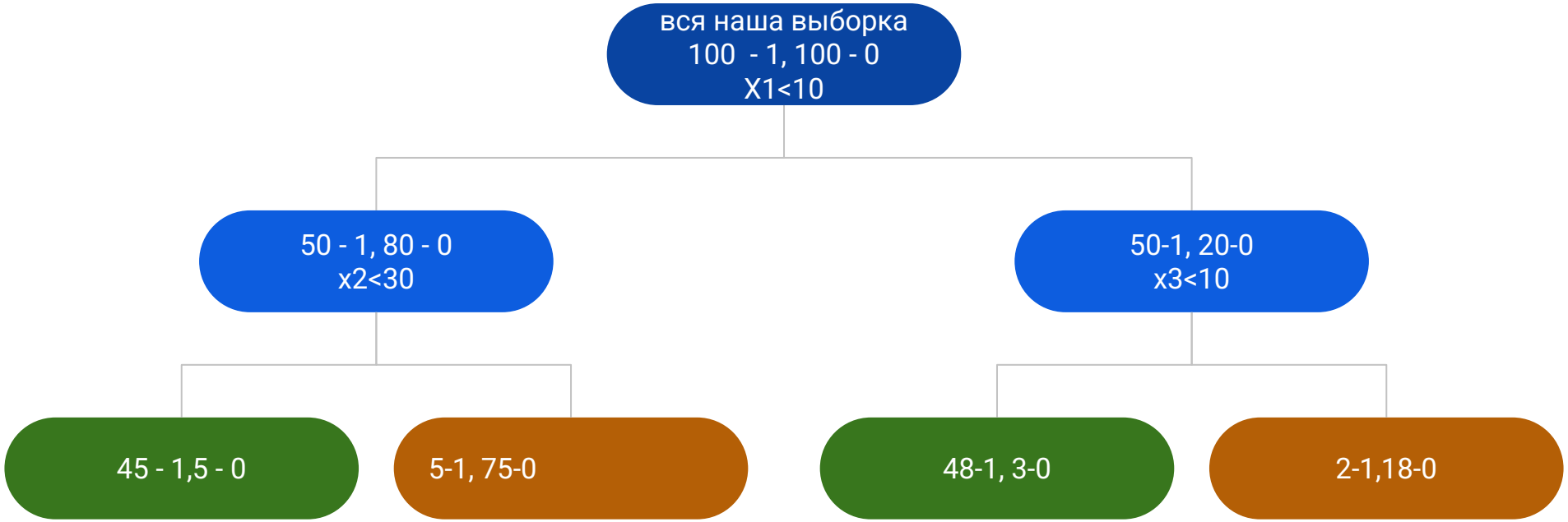
Далее идем влево и применяем уже другое правило: x2<30 и снова считаем количество 1 и 0



Как обучается дерево

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

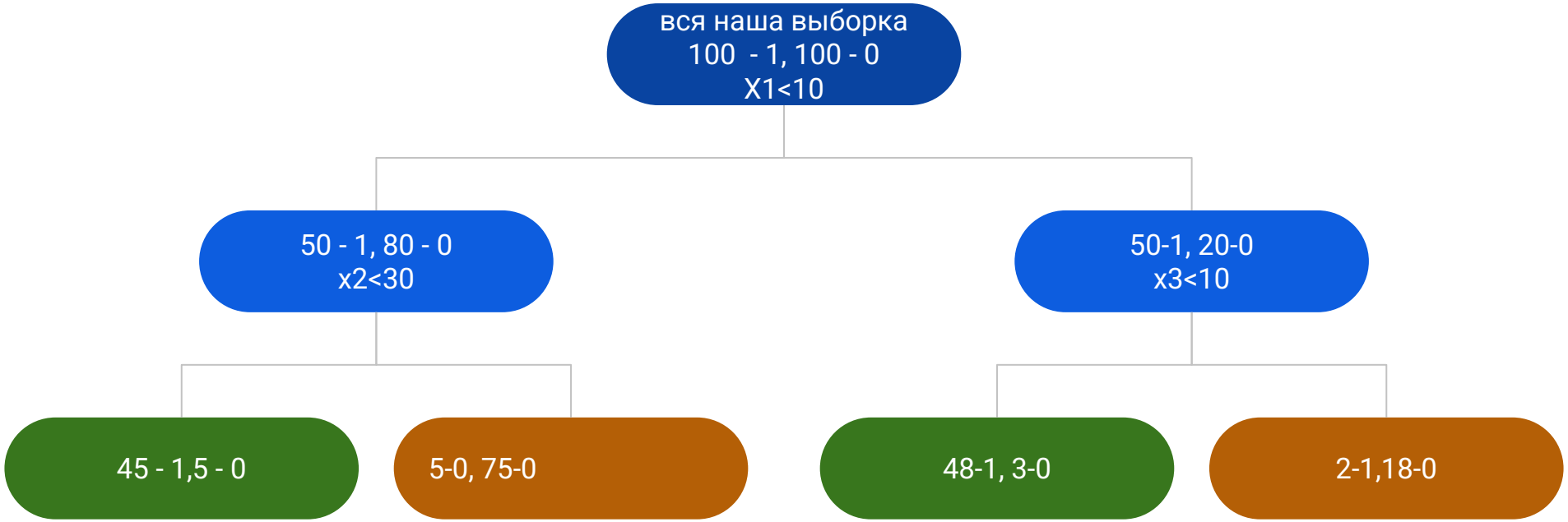
Далее идем вправо и применяем уже другое правило: x3<10 и снова считаем количество 1 и 0. Дальше разбиение не имеет уже смысла, так как в терминальных узлах (листах) выделились классы-лидеры



Как обучается дерево

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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.

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

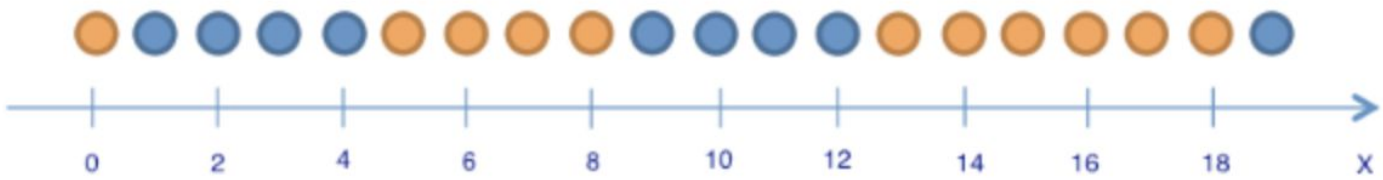
Как обучается дерево

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Рассмотрим простой пример 9 синих шариков и 11 желтых.

Если мы наудачу возьмем шарик, то он будет синим с вероятностью 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/

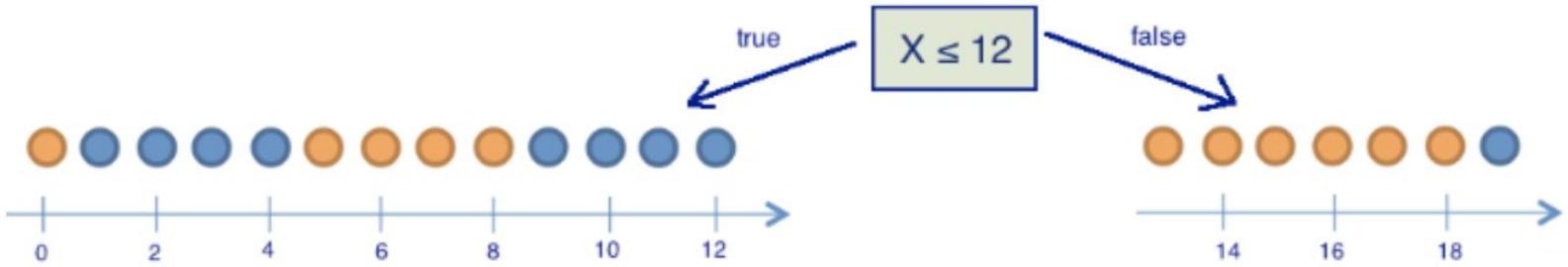
Как обучается дерево

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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

Information gain

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

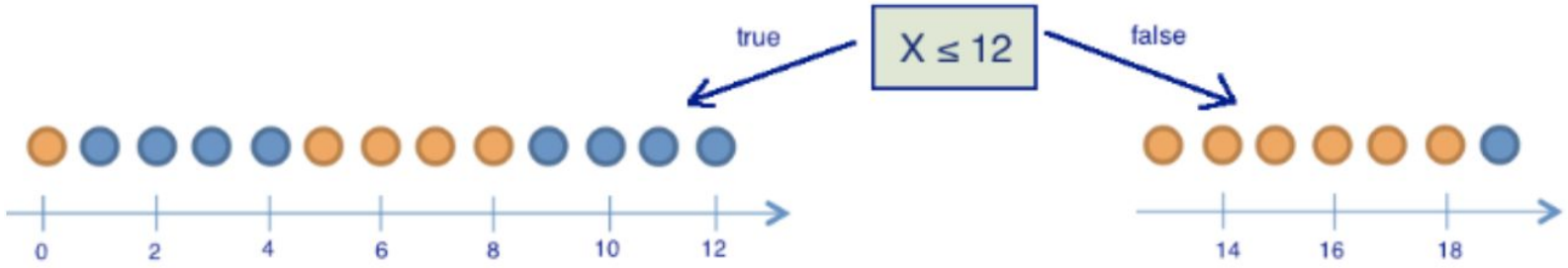
В нашем примере рассчитаем IG (Information gain)

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

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

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

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



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

Как обучается дерево

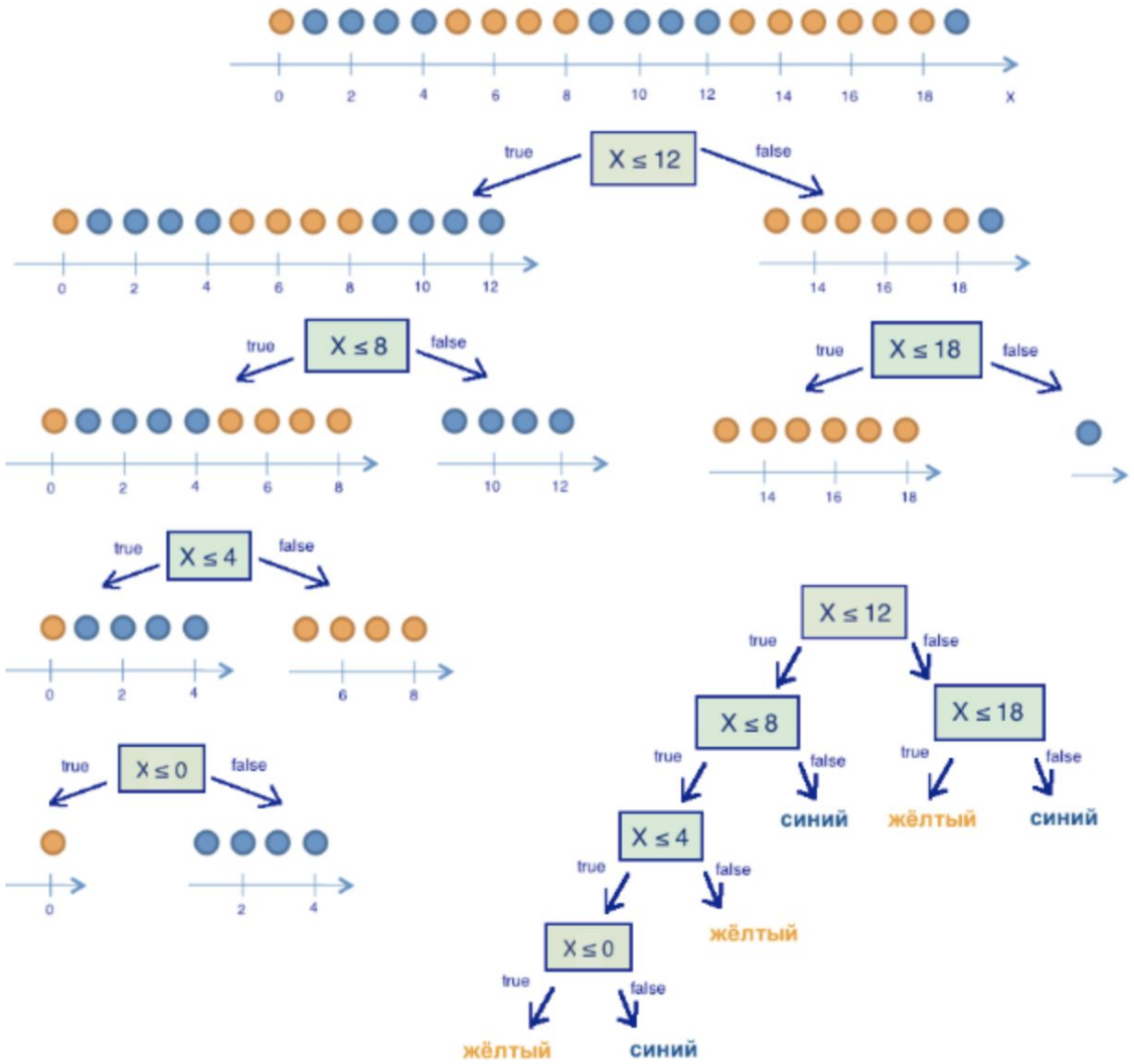
ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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



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

Критерии

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

- ограничение глубины дерева (max_depth)
- ограничение минимального количества наблюдений во внутреннем узле (min_samples_split)
- ограничение минимального количества наблюдений в листе (min_samples_leaf)
- ограничение на количество листиков - max_leaf_nodes
- минимальный прирост :

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

Критерии качества. Обобщение

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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**:

MAE

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

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

MSE

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

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

Poisson deviance (или log-likelihood)

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

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**:

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

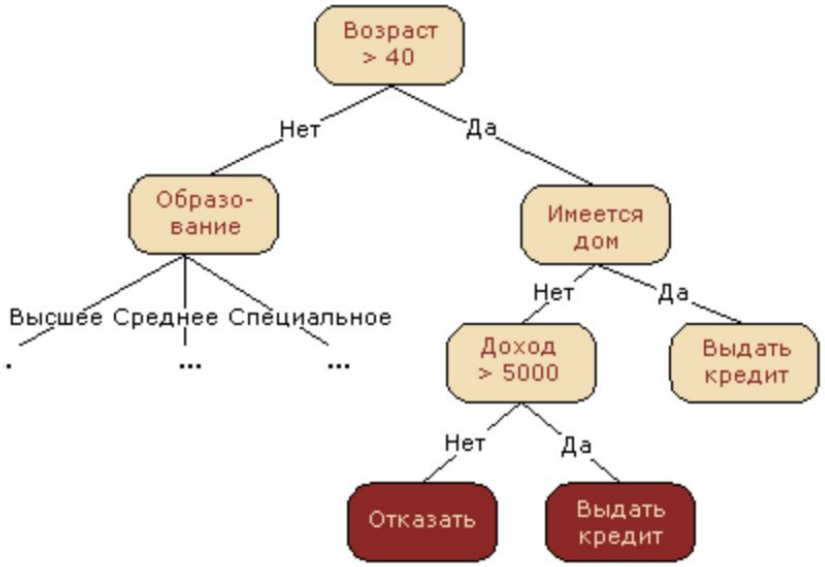
$$H(R) = \sum_{k=1}^{K} p_k \cdot (1 - p_k)$$

Принцип жадной максимизации прироста информации

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

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.



Алгоритмы построения дерева решений. История

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

ID3 (Iterative Dichotomiser 3) (Разработан Джоном Р. Квинланом):

- Работает только с дискретной целевой переменной.
- Построенные деревья являются квалифицирующими, то есть каждый лист соответствует конкретному значению целевой переменной.
- Число потомков в узле не ограничено, и дерево может иметь различную структуру.
- Алгоритм не поддерживает обработку пропущенных данных, что требует предварительной обработки данных перед использованием ID3.

C4.5:

- Является продвинутой версией алгоритма ID3.
- Работает с дискретными и непрерывными значениями атрибутов.
- Позволяет обрабатывать пропущенные значения атрибутов.
- Деревья решений, построенные с помощью C4.5, могут быть квалифицирующими, но также и квантифицирующими (допускающими промежуточные значения на листьях).

CART (Classification and Regression Tree) (Разработан Leo Breiman):

- Поддерживает как задачи классификации, так и регрессии.
- Может работать как с дискретными, так и с непрерывными целевыми переменными.
- Деревья, построенные на основе CART, имеют только два потомка для каждого узла, что делает их бинарными.

Как бороться с переобучением

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Ранняя остановка:

- ограничение глубины дерева (max_depth)
- ограничение минимального количества наблюдений во внутреннем узле (min_samples_split)
- ограничение минимального количества наблюдений в листе (min_samples_leaf)
- ограничение на количество листиков - max_leaf_nodes
- минимальный прирост : min_impurity_decrease

Как бороться с переобучением. Стрижка дерева или Pruning

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

Pruning — это процесс **уменьшения размера дерева решений**, чтобы избежать **переобучения (overfitting)**. Вместо того чтобы расти до идеального соответствия обучающим данным, дерево **обрезается**, чтобы лучше обобщать на новых данных

- Вводим штраф за сложность ccp_alpha
- R(T) - ошибки дерева, T - количество узлов $R_\alpha(T) = R(T) + \alpha|\widetilde{T}|$
- наша задача минимизировать $R_\alpha(T)$

Деревья решений. Плюсы

ПЕРЕДОВАЯ
ИНЖЕНЕРНАЯ ШКОЛА
УНИВЕРСИТЕТА ИННОПОЛИС

- Интерпретация
- Универсальность (классификация и регрессия)
- Быстрая обучаемость и предсказания
- Устойчивость к выбросам
- Не требуют масштабирования данных
- Эффективны для больших наборов данных
- Выявление важности признаков
- Легкая обработка пропущенных значений
- Возможность комбинирования в ансамблях

Деревья решений. Минусы

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

- Склонность к переобучению на сложных данных
- Неустойчивость к небольшим изменениям данных
- Сложность построения оптимальных деревьев с большим количеством признаков
- Не всегда гарантировано нахождение глобально оптимального решения
- Могут создавать слишком сложные модели, которые трудно интерпретировать
- Подвержены проблемам с несбалансированными данными в задачах классификации

Ссылки

ПЕРЕДОВАЯ ИНЖЕНЕРНАЯ ШКОЛА УНИВЕРСИТЕТА ИННОПОЛИС

- Воронцов К.В. Логические методы классификации. URL: https://web.archive.org/web/20191126130242/http://www.machinelearning.ru/wiki/images/3/3e/Voron-ML-Logic.pdf
- Деревья решений (Decision trees) . URL: https://levutkin.github.io/files/Machine_Learning_LTU_4.pdf
- Дерево классификации и регрессии (CART) . URL: https://fin-accounting.ru/cfa/l2/quantitative/cfa-classification-and-regression-tree
- Деревья решений — CART математический аппарат. Часть 1. URL: https://basegroup.ru/community/articles/math-cart-part1
- Дерево решений (CART). От теоретических основ до продвинутых техник и реализации с нуля на Python. URL: https://habr.com/ru/articles/801515/
- Деревья решений. URL: http://www.uic.unn.ru/~zny/ml/Course/13.Decision%20Trees.pdf
- Decision Trees. URL: https://scikit-learn.org/stable/modules/tree.html#tree-mathematical-formulation
- ID3, C4.5, CART and Pruning. URL: https://bitmask93.github.io/ml-blog/ID3-C4-5-CART-and-Pruning/
- Деревья решений в pySpark: от семечка до параметрической оптимизации случайного леса. URL: https://habr.com/ru/articles/760444/
- Решающие деревья. URL: https://education.yandex.ru/handbook/ml/article/reshayushchiye-derevya

Передовые инженерные школы

МИНОБРНАУКИ РОССИИ

УНИВЕРСИТЕТ ИННОПОЛИС

онлайн университет

Спасибо за внимание