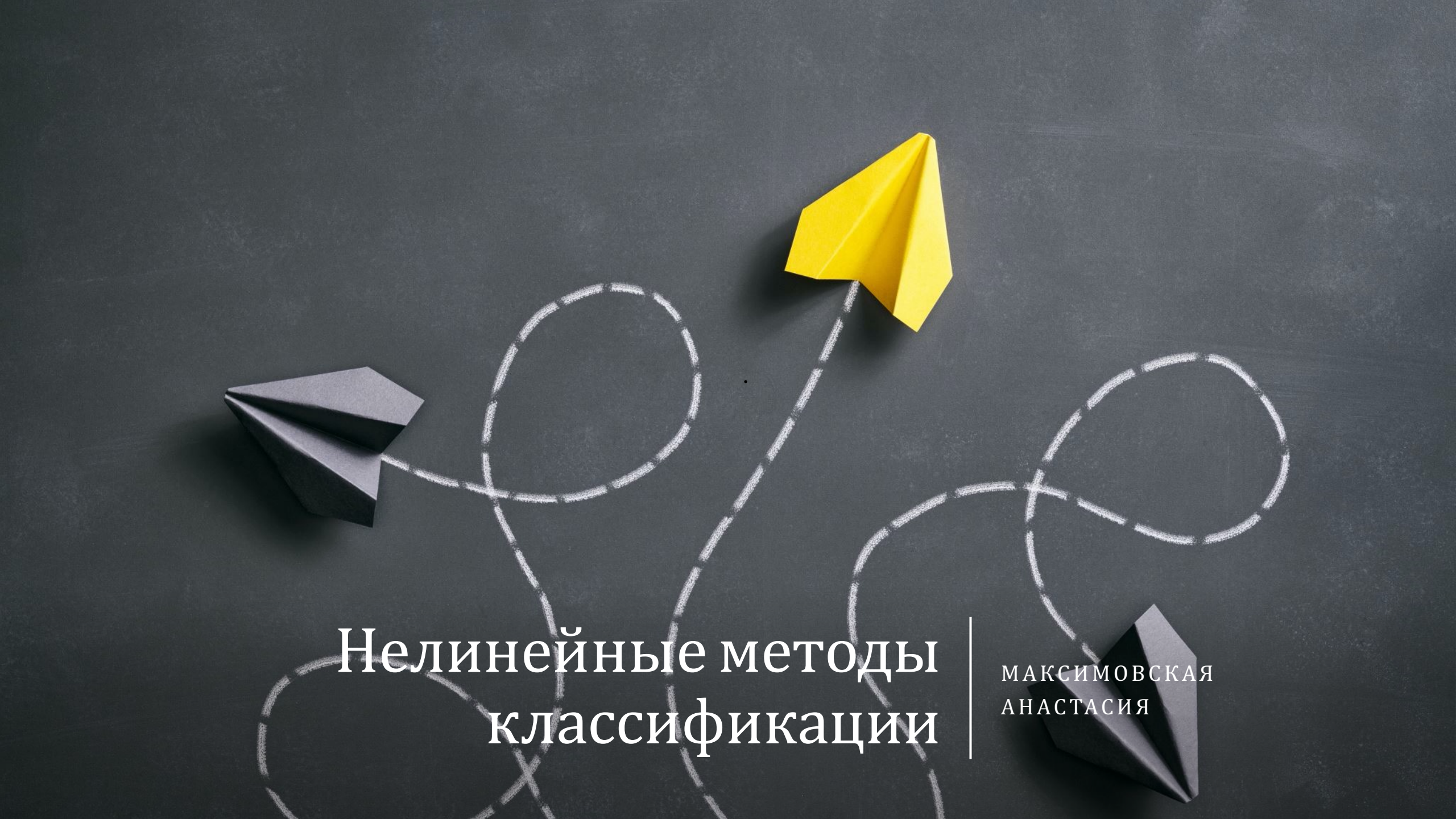


Нелинейные методы классификации

МАКСИМОВСКАЯ
АНАСТАСИЯ

Наивный байесовский классификатор

НБК

- Алгоритм классификации, основанный на теореме Байеса с допущением о независимости признаков.

НБК

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)}$$

- $P(c|x)$ – вероятность того, что объект со значением признака x принадлежит классу c
- $P(c)$ – априорная вероятность класса c
- $P(x|c)$ – вероятность того, что значение признака равно x при условии, что объект принадлежит классу c
- $P(x)$ – априорная вероятность значения признака x

НБК

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

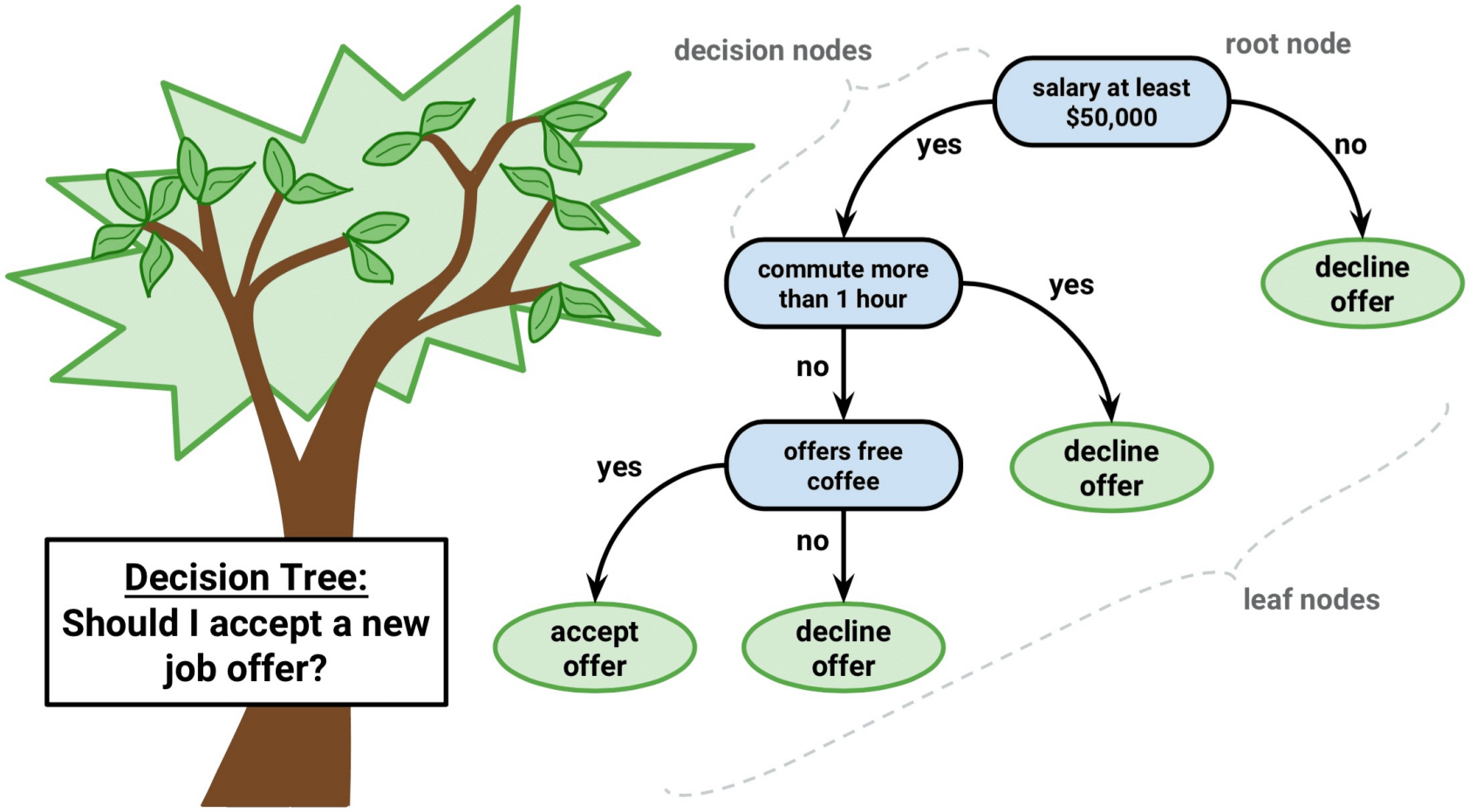
Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

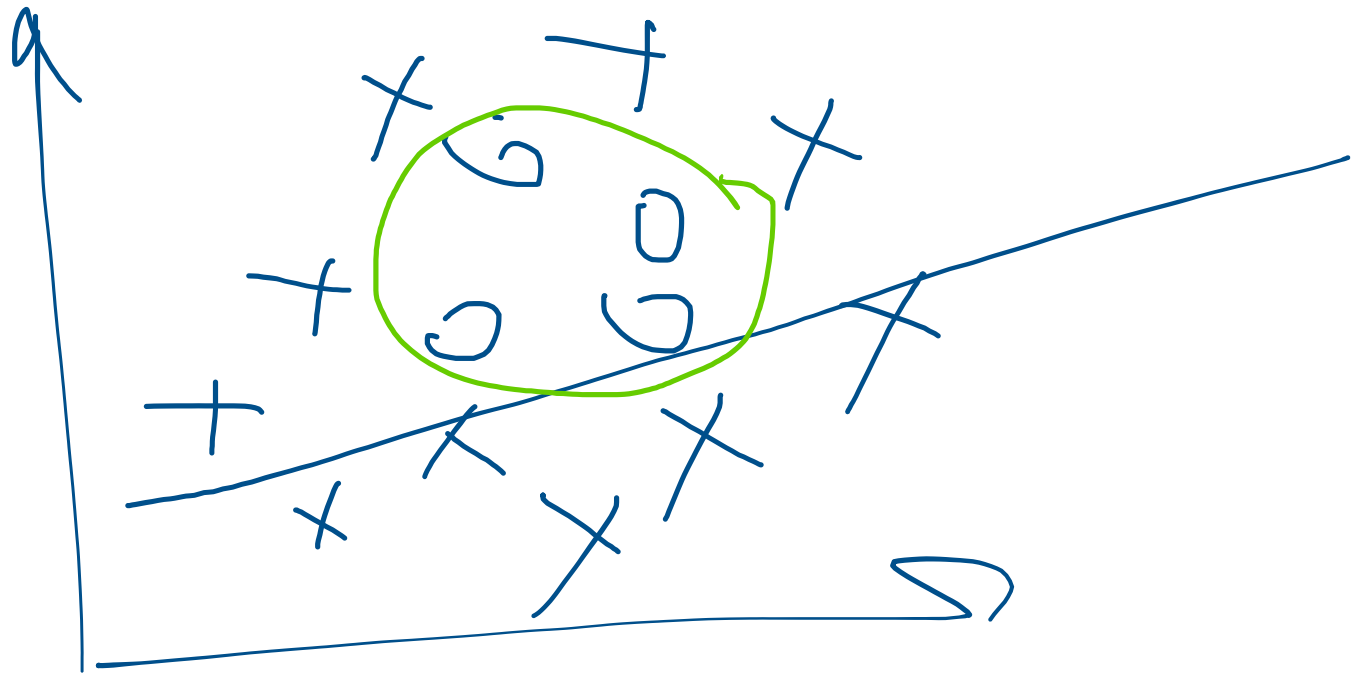
НБК

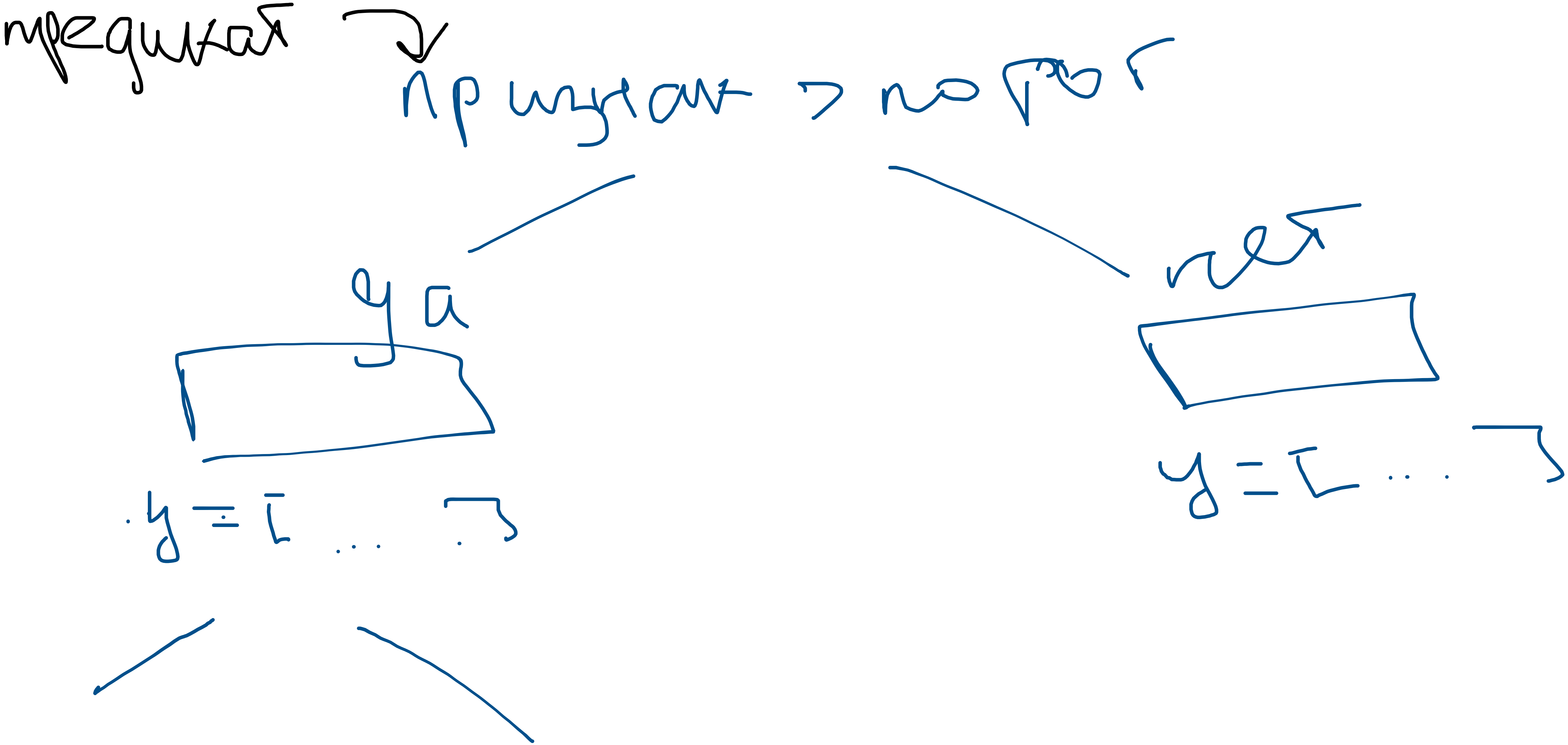
- Игра состоится, если погода солнечная. Верно ли это утверждение?
- $P(Yes \mid Sunny) = P(Sunny \mid Yes) * P(Yes) / P(Sunny)$
- $P(Sunny \mid Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes) = 9/14 = 0.64$
- $P(Yes \mid Sunny) = 0.33 * 0.64 / 0.36 = 0.60$

Деревья и леса

Решающие деревья






предикат
признак > порог
да
нет
y = [...]
y = [...]

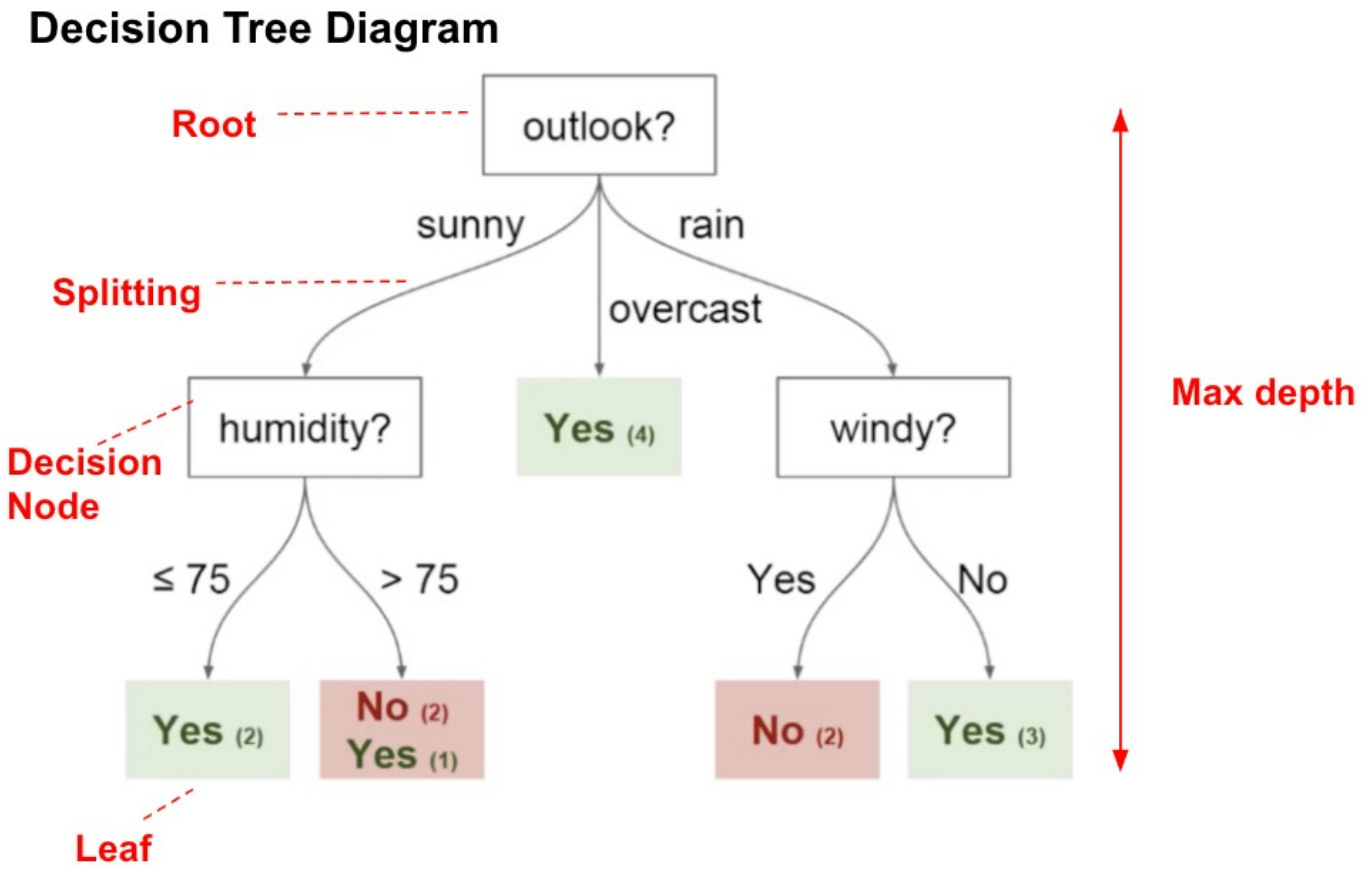
Жадный алгоритм построения

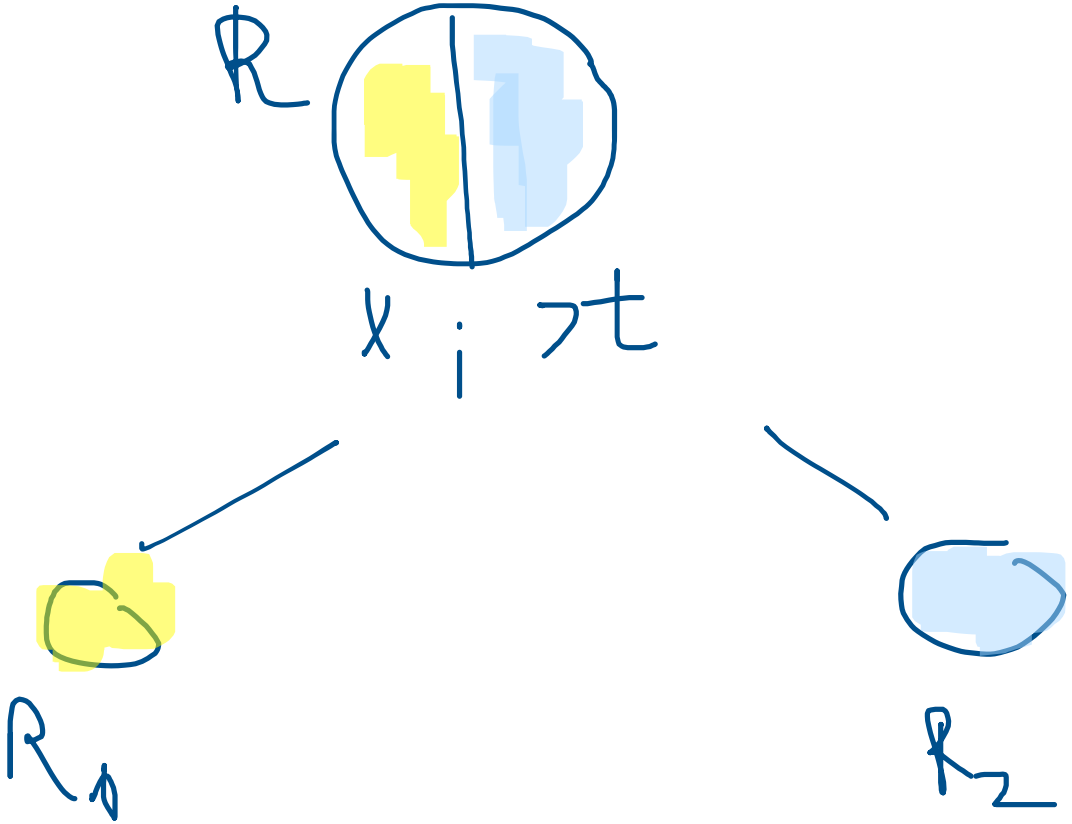
1. Найдем наилучшее разбиение выборки X на 2 части $R_1(j,t) = \{x \mid x_j < t\}$ и $R_2(j,t) = \{x \mid x_j < t\}$ с точки зрения некого функционала $Q(X,j,t)$: найдем наилучшие j и t и создадим корень дерева, поставив в него предикат $[x_j < t]$
2. Для полученных подвыборок R_1 и R_2 рекурсивно повторяем процедуру: строим дочерние вершины, в каждой проверяем выполнился ли некий критерий останова (если выполнилось – завершаем рекурсию, объявляем вершину листом)

После построения дерева каждому листу ставится в соответствие ответ:

- для классификации – класс, к которому относится большинство ответов
- для регрессии – среднее всех объектов в листе (или медиана, или другая функция от целевых переменных объектов в листе)

Решающие деревья – классификация




R
X i ㄱt
R1
R2

Критерии информативности

- При построении дерева необходимо задать функционал качества, на основе которого осуществляется разбиение выборки на каждом шаге:

$$Q(R_m, j, s) = H(R_m) - \frac{|R_\ell|}{|R_m|} H(R_\ell) - \frac{|R_r|}{|R_m|} H(R_r) \rightarrow \max_{j,t}$$

- R_m – множество объектов, попавших в вершину, разбиваемую на данном шаге
- R_r и R_l – объекты, попавшие в правое и левое поддеревья
- $H(R)$ – критерий информативности (impurity criterion) $\rightarrow \min$

Критерии информативности

Информативность в листе – это дисперсия целевой переменной (для объектов, попавших в этот лист). Чем меньше дисперсия, тем меньше разброс целевой переменной объектов, попавших в лист.

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

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

Регрессия

- Зададим как квадрат отклонения:

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

- Минимум будет достигаться на среднем значении целевой переменной:

$$H(R) = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} \left(y_i - \frac{1}{|R|} \sum_{(x_j, y_j) \in R} y_j \right)^2$$

Регрессия

- Мы получили, что информативность вершины измеряется её дисперсией— чем ниже разброс целевой переменной, тем лучше вершина.
- Разумеется, можно использовать и другие функции ошибки L — например, при выборе абсолютного отклонения мы получим в качестве критерия среднее абсолютное отклонение от медианы.

Классификация

- Доля объектов класса, попавших в вершину:

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

- Через k_* обозначим класс, чьих представителей оказалось больше всего среди объектов, попавших в данную вершину:

$$k_* = \arg\max_k p_k$$

Ошибка классификации

- Рассмотрим индикатор ошибки как функцию потерь:

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

- Легко видеть, что оптимальным предсказанием тут будет наиболее популярный класс k_* — значит, критерий будет равен следующей доле ошибок:

$$H(R) = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} [y_i \neq k_*] = 1 - p_{k_*}$$

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

Критерий Джини

- Рассмотрим ситуацию, в которой мы выдаём в вершине не один класс, а распределение на всех классах c $= (c_1, \dots, c_K)$, $\sum_{k=1}^{K} c_k = 1$. Качество такого распределения можно измерять, например, с помощью критерия Бриера (Brier score):

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

- Оптимальный вектор вероятностей состоит из долей классов: $c_* = (p_1, \dots, p_K)$

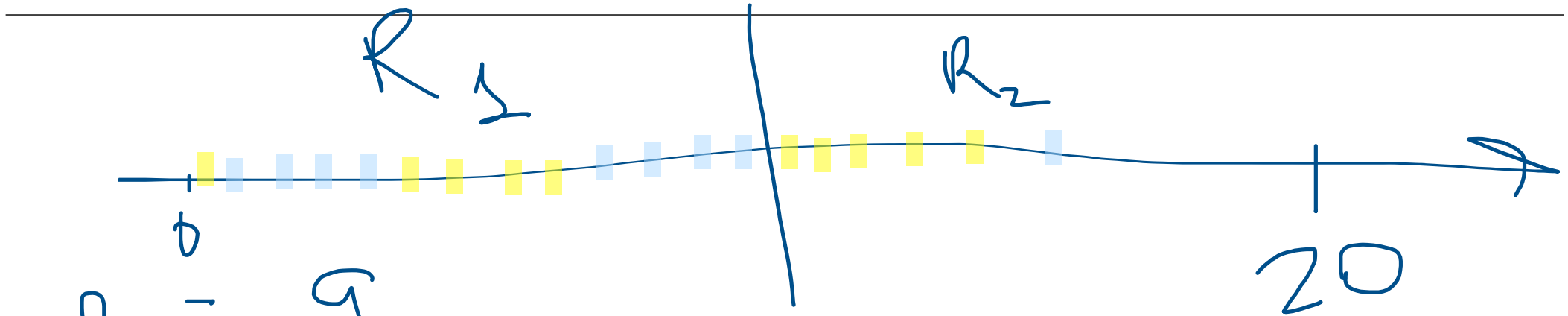
- Если подставить эти вероятности в исходный критерий информативности и провести ряд преобразований, то мы получим критерий Джини:

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

Энтропийный критерий

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

- Из теории вероятностей известно, что энтропия ограничена снизу нулем, причем минимум достигается на вырожденных распределениях ($p_i = 1, p_j = 0$ для $i \neq j$).
- Максимальное значение энтропия принимает для равномерного распределения.
- Отсюда видно, что энтропийный критерий отдает предпочтение более «вырожденным» распределениям классов в вершине.



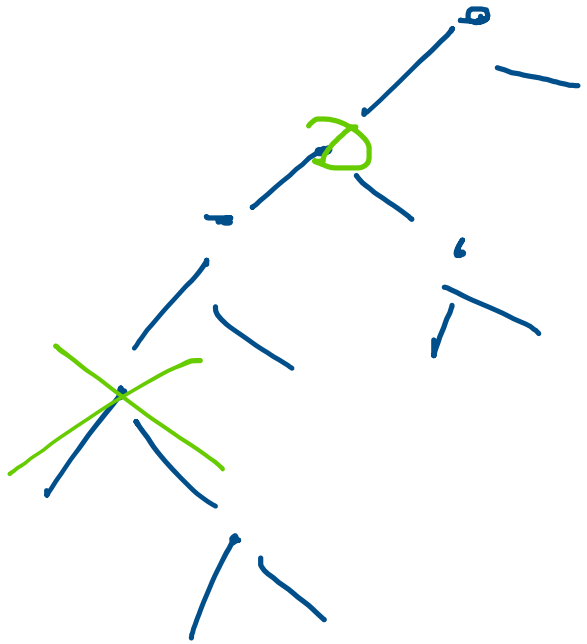
$p_{nc} = \frac{9}{20}$

$p_c = \frac{11}{20}$

$H(p) = -\frac{9}{20}\ln\frac{9}{20} - \frac{11}{20}\ln\frac{11}{20}$

Критерии останова

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



Стрижка деревьев

- После того, как дерево построено, можно провести его стрижку (pruning) — удаление некоторых вершин с целью понижения сложности и повышения обобщающей способности.
- На данный момент методы стрижки редко используются и не реализованы в большинстве библиотек для анализа данных. Причина заключается в том, что деревья сами по себе являются слабыми алгоритмами и не представляют большого интереса, а при использовании в композициях они либо должны быть переобучены (в случайных лесах), либо должны иметь очень небольшую глубину (в бустинге), из-за чего необходимость в стрижке отпадает.

Плюсы деревьев

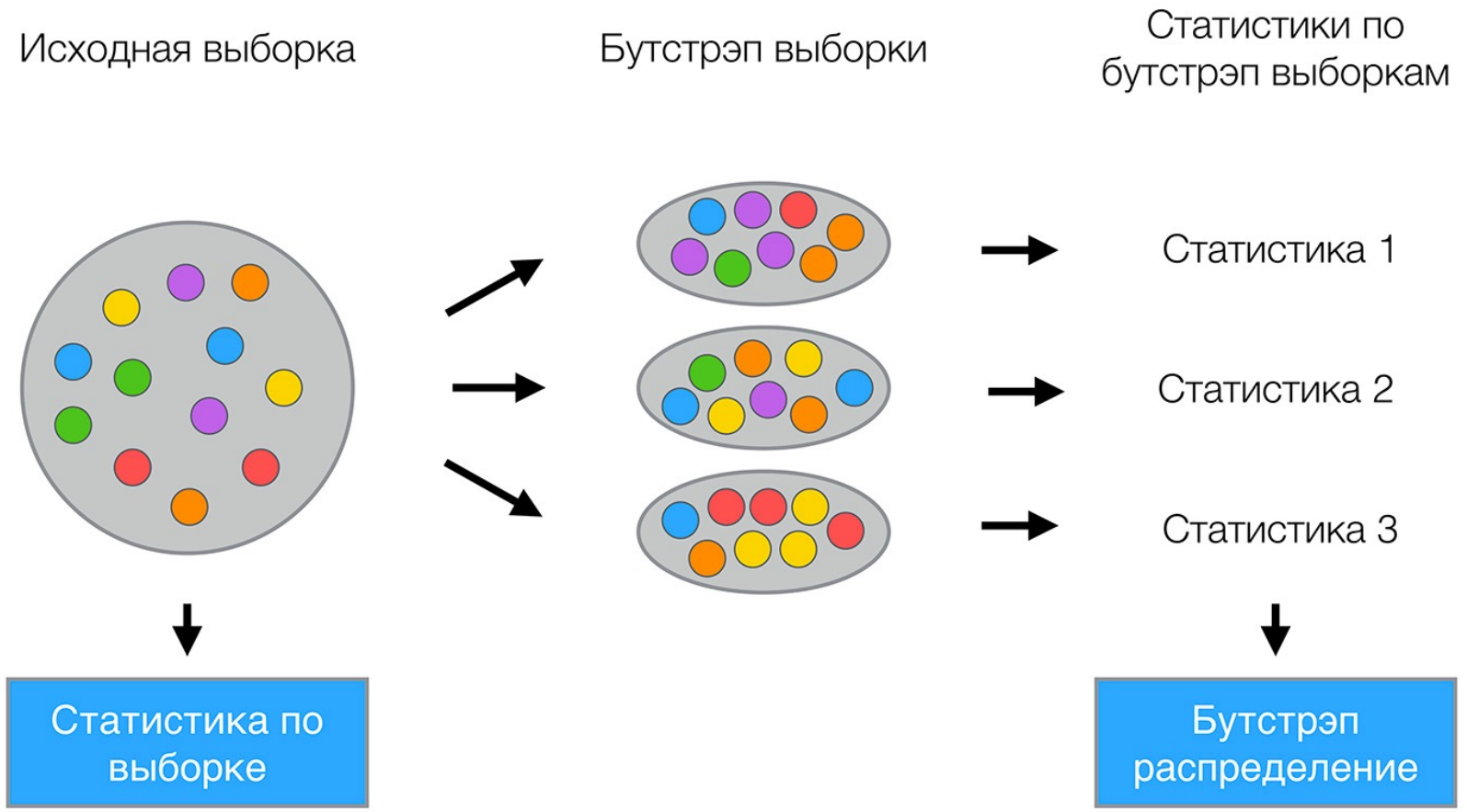
- Четкие правила классификации
- Хорошо интерпретируются
- Быстро обучаются и выдают прогноз
- Малое число параметров

Минусы деревьев

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

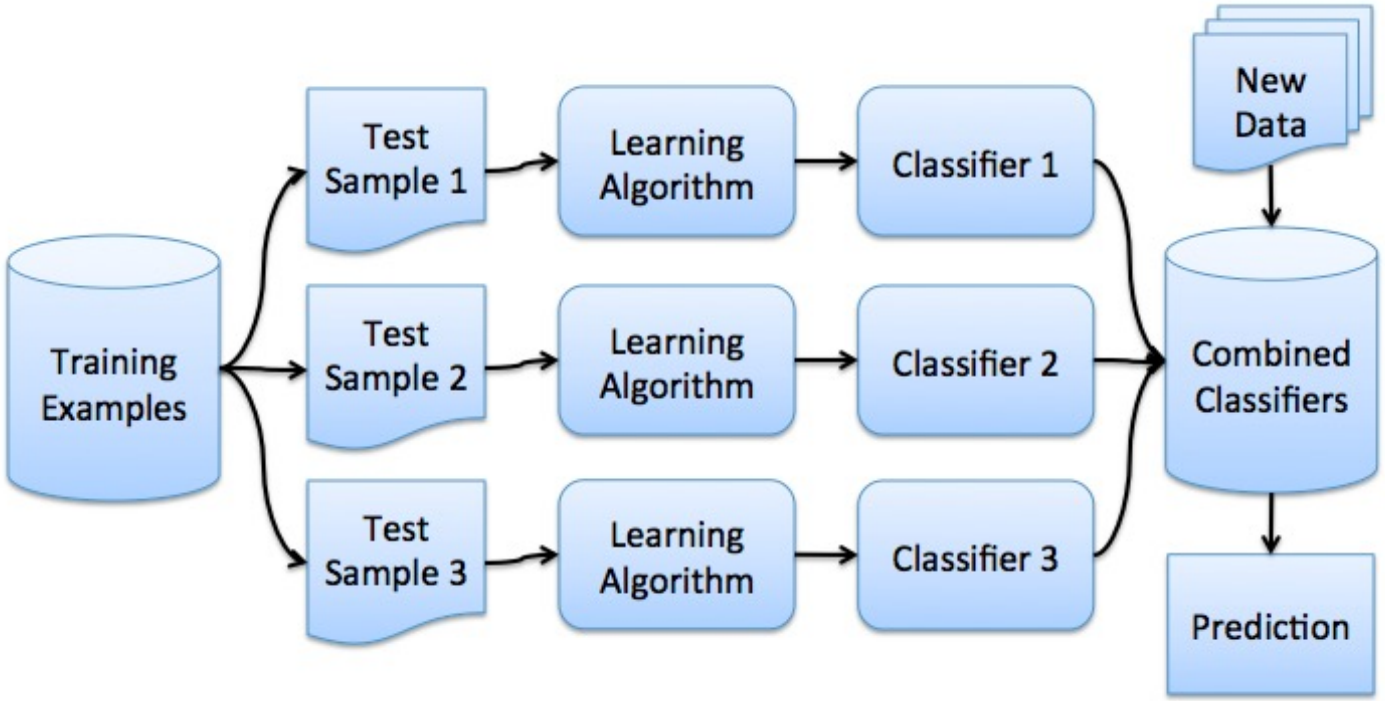
Случайный лес

Бутстрэп



Источник: URL

Бэггинг



Источник: URL

Алгоритм случайного леса

- Алгоритм построения случайного леса, состоящего из N деревьев, выглядит следующим образом:

- Для каждого $n = 1, \dots, N$:
  - Сгенерировать выборку X_n с помощью бутстрэпа;
  - Построить решающее дерево b_n по выборке X_n :
    — по заданному критерию мы выбираем лучший признак, делаем разбиение в дереве по нему и так до исчерпания выборки
    — дерево строится, пока в каждом листе не более n_{min} объектов или пока не достигнем определенной высоты дерева
    — при каждом разбиении сначала выбирается m случайных признаков из n исходных, и оптимальное разделение выборки ищется только среди них.