Градиентный бустинг: имплементации

МАКСИМОВСКАЯ АНАСТАСИЯ

Extreme Gradient Boosting

XGBoost

В градиентном бустинге на каждой итерации вычисляется вектор сдвигов s:

$$s = \left(- \frac{\partial L}{\partial z} \Big|_{z=a_{N-1}(x_i)} \right)_{i=1}^{\ell} = -\nabla_z \sum_{i=1}^{\ell} L(y_i, z_i) \Big|_{z_i=a_{N-1}(x_i)}$$

После этого обучается новый базовый алгоритм:

$$b_N(x) = \arg\min_{b \in \mathcal{A}} \sum_{i=1}^{\ell} (b(x_i) - s_i)^2$$

$a_N = \sum b_i(x)$

y - таргет

1) $b_1(x)$ $\quad \frac{1}{\ell}\sum \left(b(x_i) - y_i\right)^2 \to \min\limits_{b_i}$

2) $s_i = \cancel{(y_i - b_i(x))}$ $\quad s_i = -\frac{\partial L}{\partial z} b(x_i)$

$b_2(x)$ $\quad \frac{1}{\ell}\sum \left(b(x_i) - s_i\right)^2 \to \min\limits_{b_i}$

$a(x) = b_1(x) + b_2(x)$

3) $s_i = y_i - b_1(x_i) - b_2(x_i)$

XGBoost

В XGBoost на каждом шаге решается задача:

$$\sum_{i=1}^{l}\left(-s_i b(x_i) + \frac{1}{2} h_i b^2(x_i)\right) + \gamma J + \frac{\lambda}{2} \sum_{j=1}^{J} b_j^2 \to \min_b$$

$$h_i = \frac{\partial^2 L}{\partial z^2}\Big|_{a_{N-1}(x_i)}$$

XGBoost

Мы хотим найти алгоритм $b(x)$, решающий следующую задачу:

$$\sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b(x_i)) \to \min_b$$

XGBoost

Разложим функцию L в каждом слагаемом в ряд Тейлора до второго члена с центром

в ответе композиции $a_{N-1}(x_i)$:

$$\sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b(x_i)) \approx$$

$$\approx \sum_{i=1}^{\ell} \left(L(y_i, a_{N-1}(x_i)) - s_i b(x_i) + \frac{1}{2} h_i b^2(x_i) \right),$$

XGBoost

Где h_i – производные по сдвигам:

$$h_i = \frac{\partial^2}{\partial z^2} L(y_i, z) \Big|_{a_{N-1}(x_i)}$$

XGBoost

Выкидываем первое слагаемое, т.к. оно не зависит от нового базового алгоритма:

$$\sum_{i=1}^{\ell} \left(-s_i b(x_i) + \frac{1}{2} h_i b^2(x_i) \right) \to \min_b$$

XGBoost

$$
\begin{aligned}
&\sum_{i=1}^{\ell} (b(x_i) - s_i)^2 \\
&\qquad = \sum_{i=1}^{\ell} \left(b^2(x_i) - 2s_i b(x_i) + s_i^2\right) = \{\text{последнее слагаемое не зависит от } b\} \\
&\qquad = \sum_{i=1}^{\ell} \left(b^2(x_i) - 2s_i b(x_i)\right) \\
&\qquad = 2\sum_{i=1}^{\ell} \left(-s_i b(x_i) + \frac{1}{2} b^2(x_i)\right) \to \min_b
\end{aligned}
$$

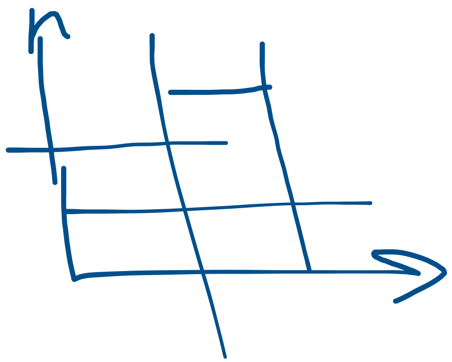
$b(x) = \sum b_i [x \in R_i]$

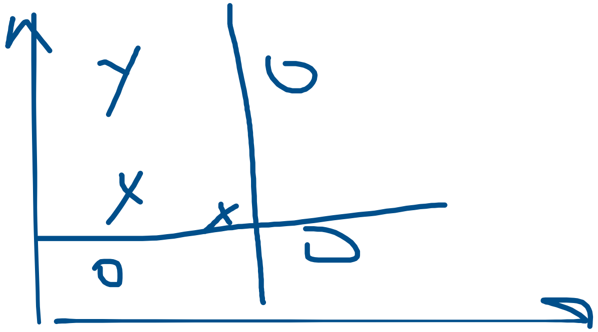
XGBoost

В XGBoost на каждом шаге решается задача:

$$\sum_{i=1}^{l} \left(-s_i b(x_i) + \frac{1}{2} h_i b^2(x_i)\right) + \gamma J + \frac{\lambda}{2} \sum_{j=1}^{J} b_j^2 \to \min_b$$

$$h_i = \frac{\partial^2 L}{\partial z^2}\bigg|_{a_{N-1}(x_i)}$$





XGBoost

Сложность дерева $b(x)$ зависит от:

- Число листьев J. Чем больше листьев имеет дерево, тем сложнее его разделяющая поверхность, тем больше у него параметров и тем выше риск переобучения

- Норма коэффициентов в листьях . Чем сильнее коэффициенты отличаются от нуля, тем сильнее данный базовый алгоритм будет влиять на итоговый ответ композиции.

$$\|b\|_2^2 = \sum_{j=1}^{J} b_j^2$$

$x_i > t$

XGBoost

y 1, 2, 3 — 2 $\qquad$ y 4, 5, 6 — 5

Т.к. дерево дает одинаковый ответ на объектах, попадающих в один лист, можно упростить функционал:

$$\sum_{j=1}^{J}\left\{\underbrace{\left(-\sum_{i\in R_j} s_i\right)}_{=-S_j} b_j + \frac{1}{2}\left(\lambda + \underbrace{\sum_{i\in R_j} h_i}_{=H_j}\right) b_j^2 + \gamma\right\} \to \min_b$$

$$b = \frac{S_j}{H_j + \lambda}$$

XGBoost

Отдельное слагаемое представляет собой параболу относительно b_j, благодаря чему можно аналитически найти оптимальные коэффициенты в листьях:

$$b_j = \frac{S_j}{H_j + \lambda}$$

XGBoost

Ошибка дерева с оптимальными коэффициентами:

$$H(b) = -\frac{1}{2}\sum_{j=1}^{J}\frac{S_j^2}{H_j+\lambda} + \gamma J$$

XGBoost

Этот функционал подходит в качестве критерия информативности:

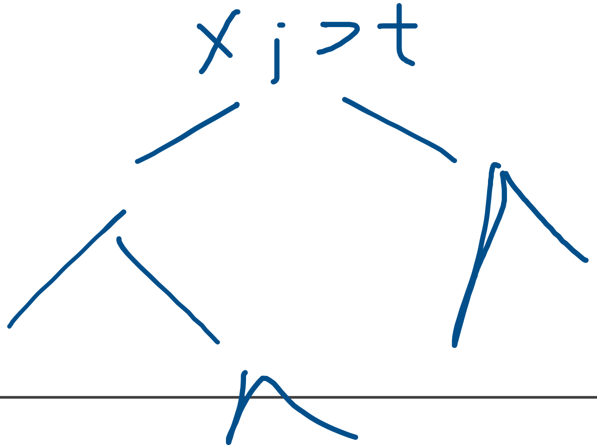
$$Q = H(R) - H(R_\ell) - H(R_r) \to \max$$

$$H(R) = -\frac{1}{2} \left(\sum_{(h_i, s_i) \in R} s_j \right)^2 \Bigg/ \left(\sum_{(h_i, s_i) \in R} h_j + \lambda \right) + \gamma.$$

XGBoost

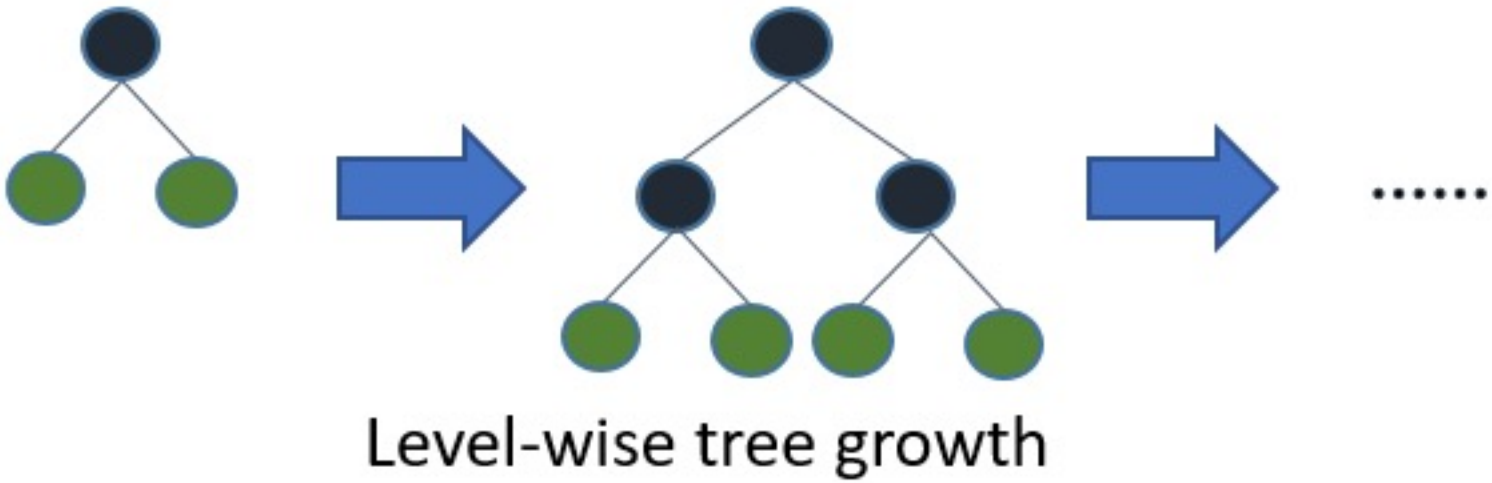
- Базовый алгоритм приближает направление, посчитанное с учетом вторых производных функции потерь;
- Функционал регуляризуется — добавляются штрафы за количество листьев и за норму коэффициентов;
- При построении дерева используется критерий информативности, зависящий от оптимального вектора сдвига;
- Критерий останова при обучении дерева также зависит от оптимального сдвига.

lightGBM



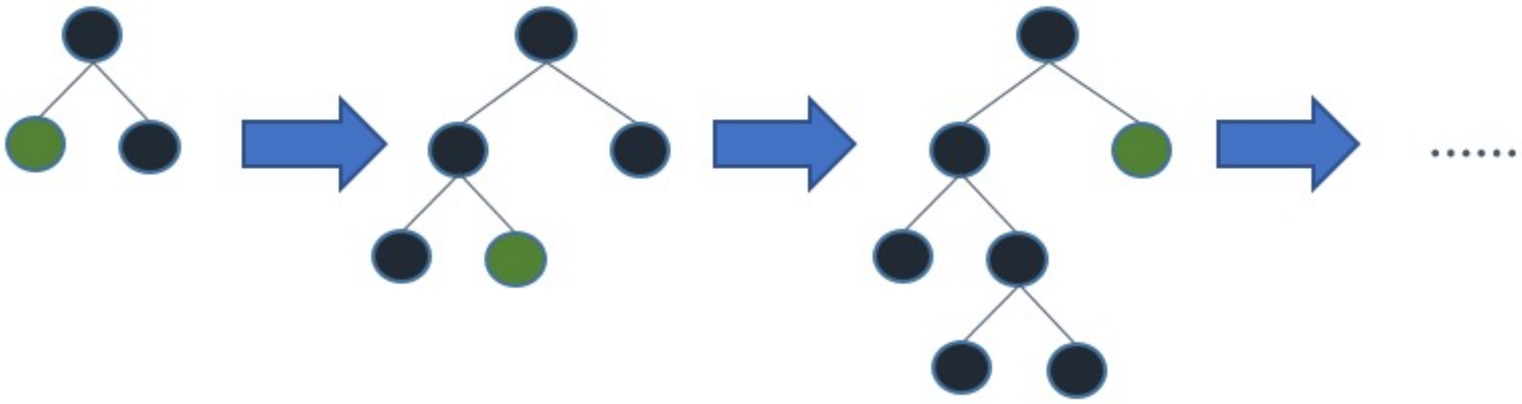
lightGBM

В других реализациях градиентного бустинга деревья строятся по уровням:



lightGBM

LightGBM строит деревья, добавляя на каждом шаге один лист. Это позволяет добиться более высокой точности решения задачи оптимизации



Leaf-wise tree growth

lightGBM

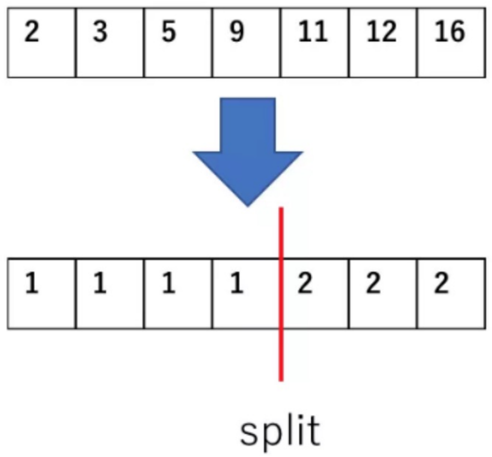
Кодирование категориальных признаков.

- LightGBM разбивает значения категориального признака на два подмножества в каждой вершине дерева, находя при этом наилучшее разбиение

- Если категориальный признак имеет k различных значений, то возможных разбиений $2^{k-1} - 1$. В LightGBM реализован способ поиска оптимального разбиения за $O(klogk)$ операций.

lightGBM

Ускорение построения деревьев за счёт бинаризации признаков:



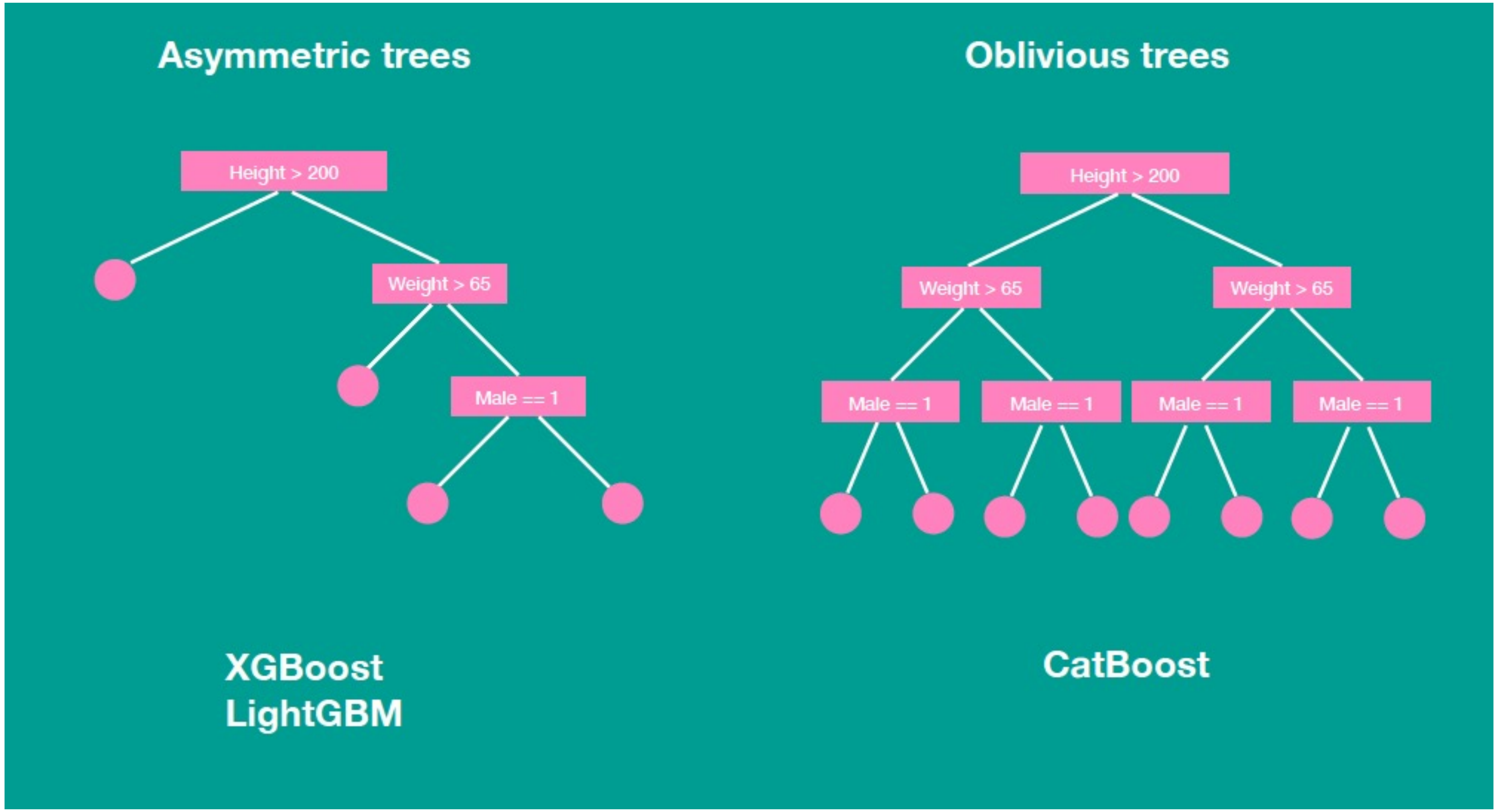
An example of how binning can reduce the number of splits to explore. The features must be sorted in advance for this method to be effective.

CatBoost

CatBoost

- Алгоритм Яндекса, является оптимизацией Xgboost и в отличие от Xgboost умеет обрабатывать категориальные признаки.

Симметричные деревья



Выбор лучшего сплита

$$\text{score(split)} = \frac{\sum_{doc} leafValue(doc) * gradient(doc) \; * \; w(doc)}{\sqrt{\sum_{doc} w(doc) * \; leafValue(doc)^2}}$$

$$leafValue(doc) = \frac{sumWeightedDer}{sumWeights}$$

Бутстрап

Бернулли:

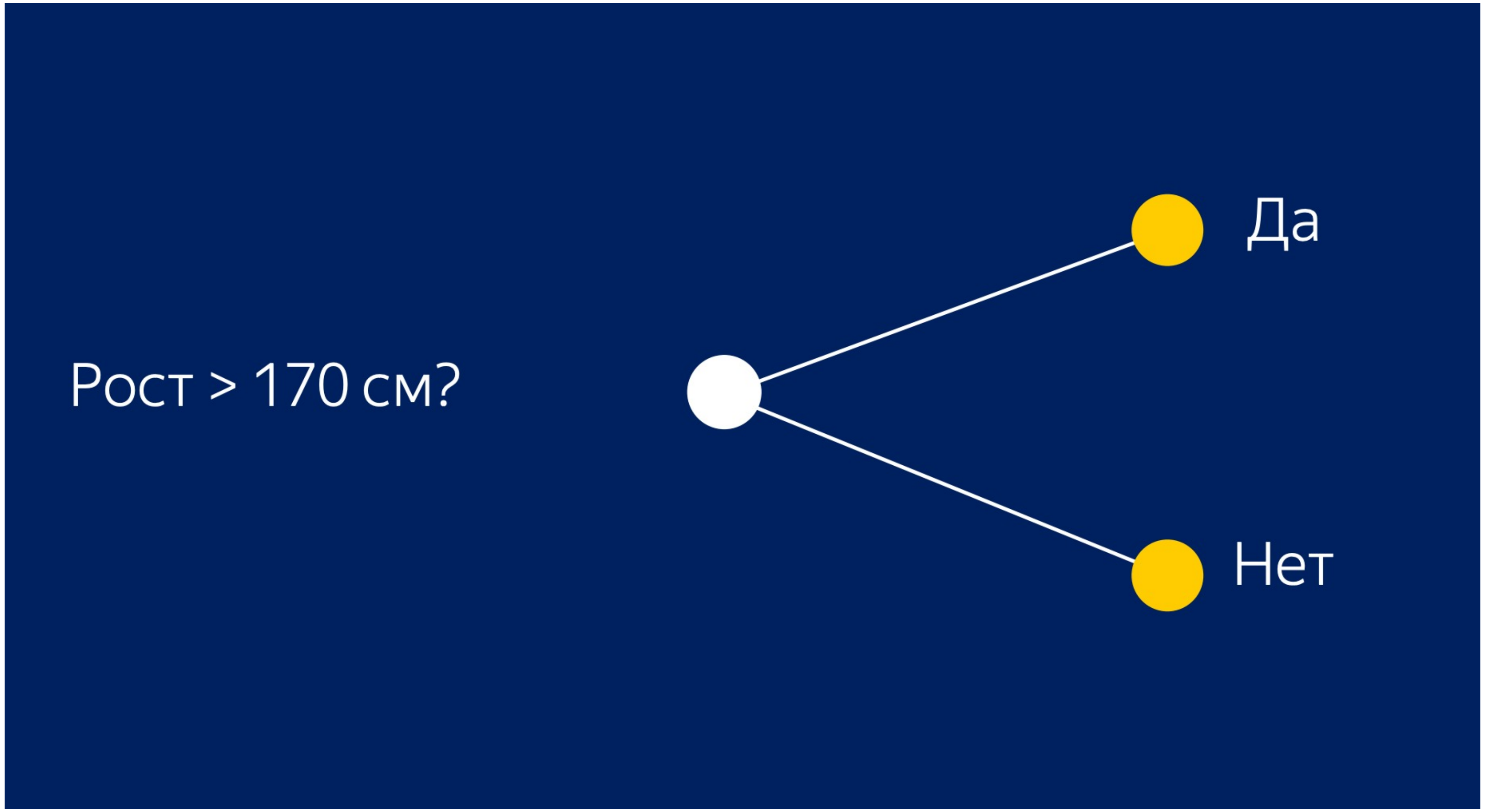
$$w(doc) = 0 \text{ or } 1 \; (P(1) = sample\_rate)$$

Байесовский:

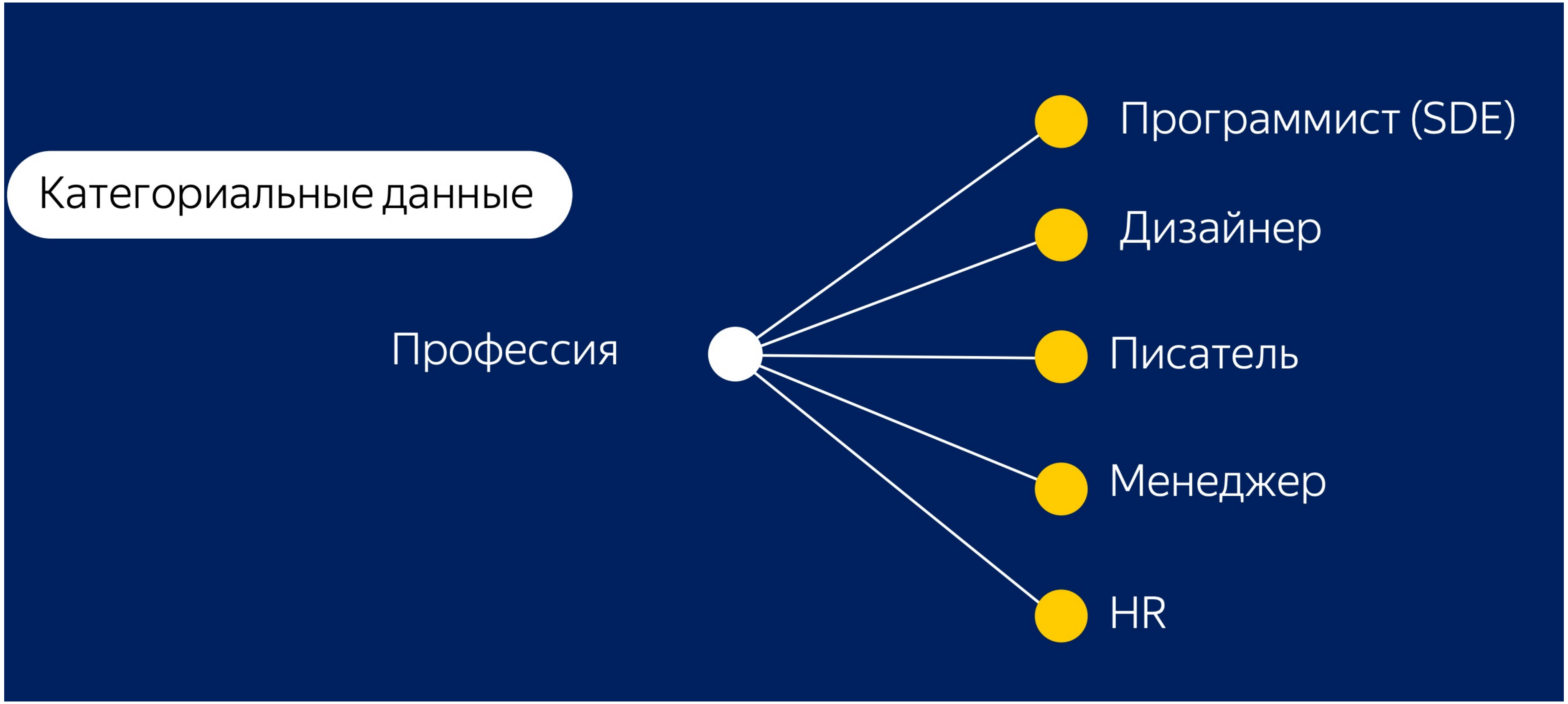
$$w(doc) \mathrel{*}= (-\log(rand(0,1)))^{bagging\_temperature}$$

Только на этапе выбора структуры дерева

Числовые факторы

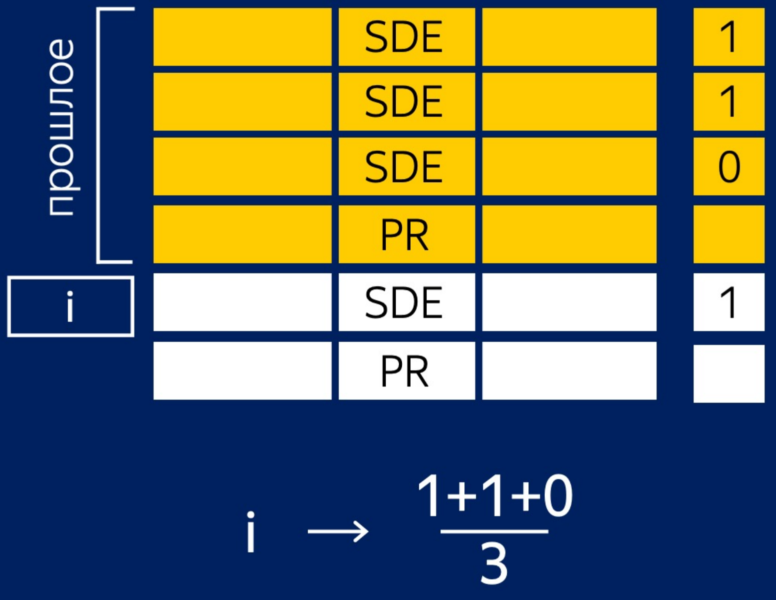


Категориальные факторы

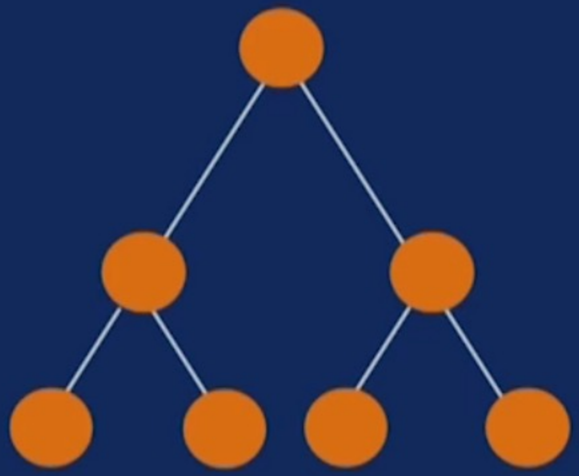


Статистики по категориальным факторам

- One-hot кодирование
- Статистики без использования таргета
- Статистики по случайным перестановкам
- Комбинации факторов

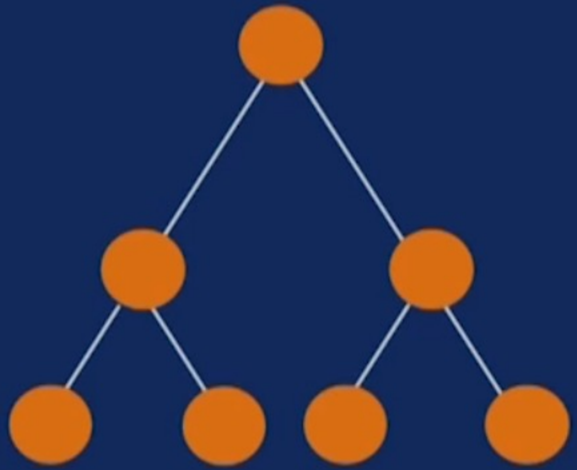


Обычный бустинг



$$\text{leafValue} = \sum_{i=1}^{n} \frac{g(\text{approx}(i), \text{target}(i))}{n}$$

Динамический бустинг



$$\text{leafValue(doc)} = \sum_{i=1}^{doc} \frac{\text{g(approx(i), target(i))}}{\textit{docs in the past}}$$

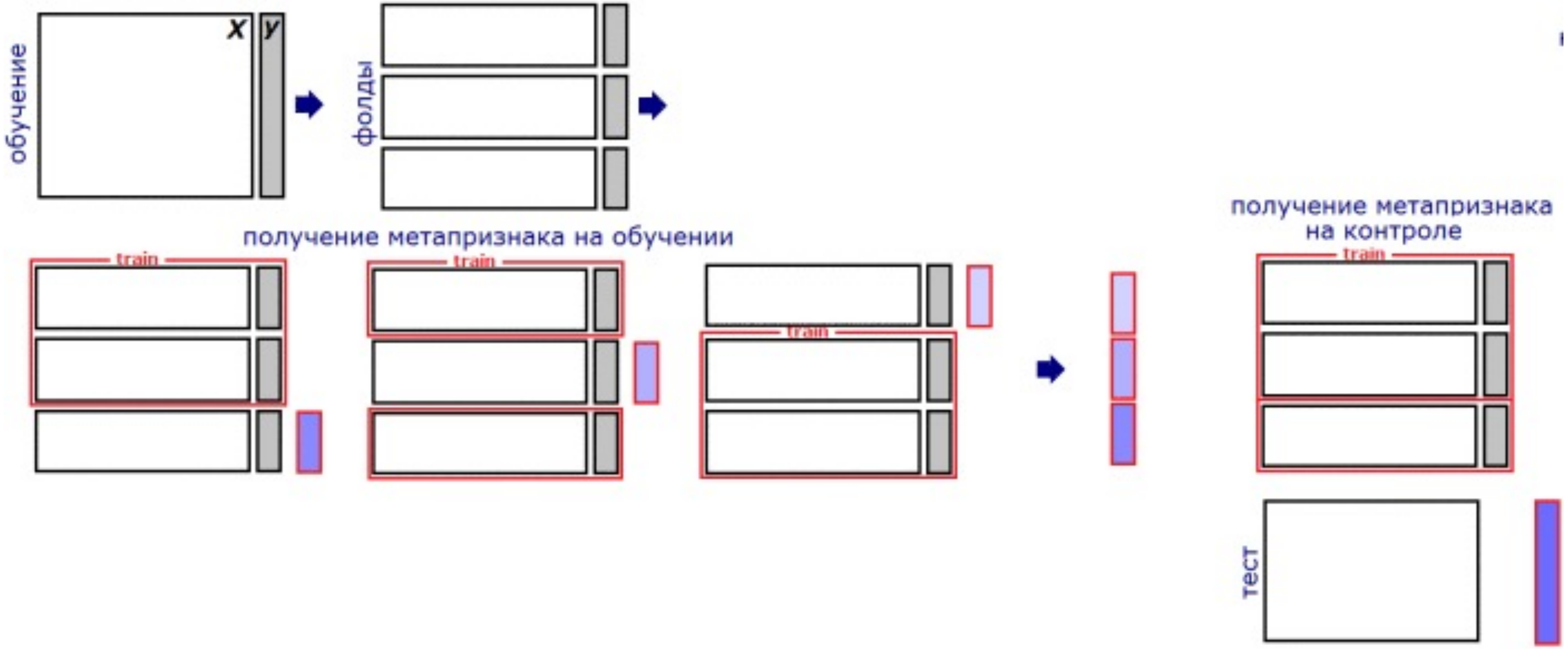
Как еще можно строить композици?

$N \quad b_1(x) \ldots b_N(x) \qquad a(x)$

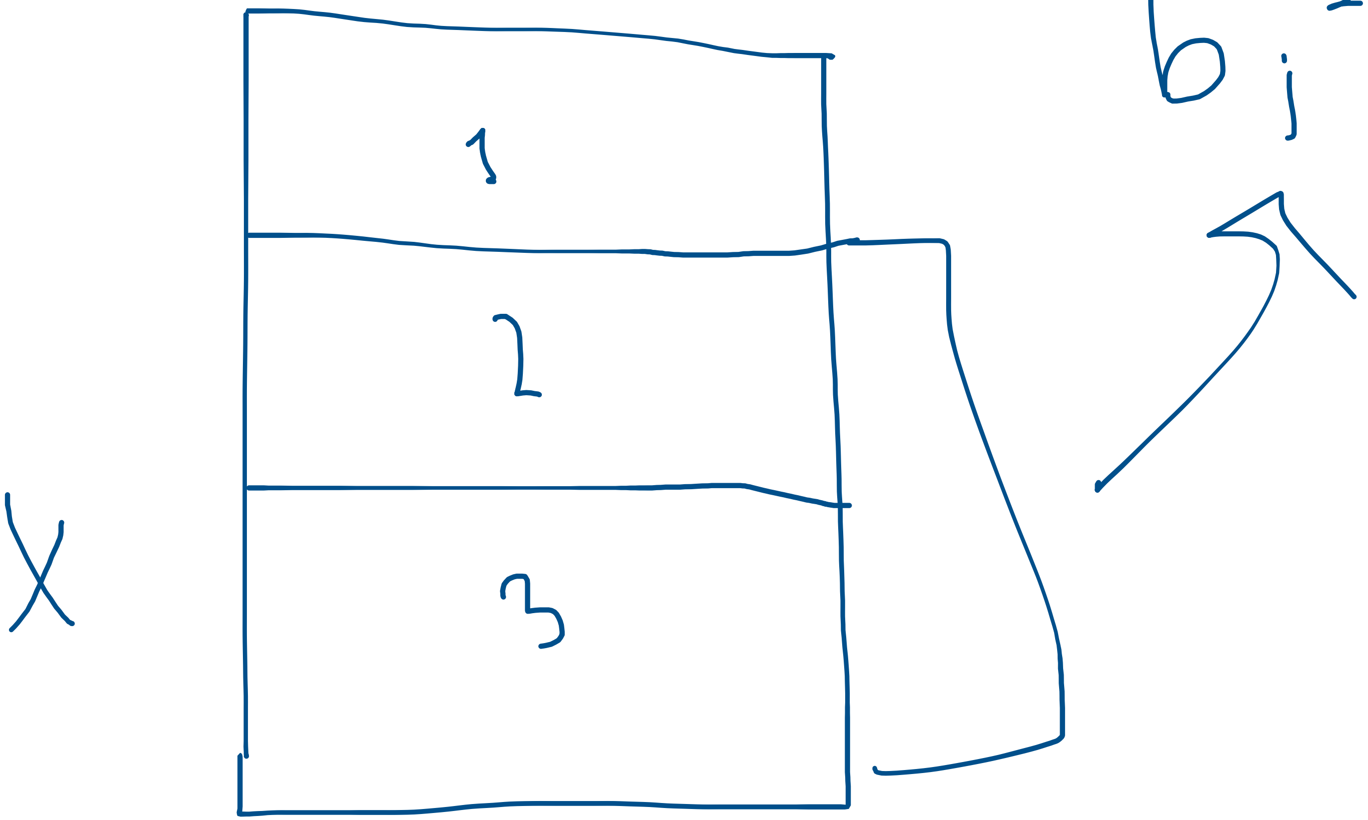
Стекинг

$$\sum L(y_i, a(b_1(x_i), b_2(x_i), \ldots, b_N(x_i))) \to \min_a$$

- Прогнозы алгоритмов объявляются новыми признаками, и поверх них обучается ещё один алгоритм (который иногда называют мета-алгоритмом)



$b_j^{-1}(x) \to X_2 \; X_3$



$$\sum_{K=1}^{K} \sum L\left(y_{i,q}, \left(b_1^{-k}(x), b_2^{-k}(x), \dots\right)\right)$$

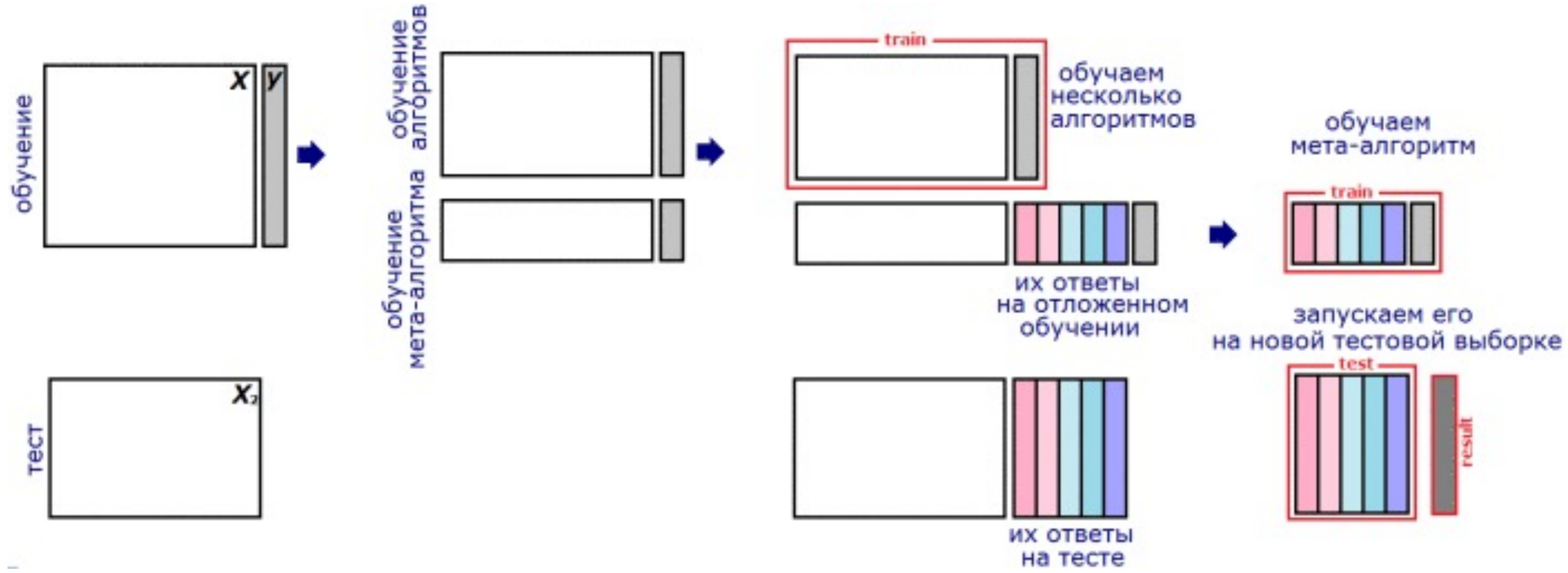
Блендинг

- Частным случаем стекинга является блендинг, в котором мета-алгоритм является линейным:

$$a(x) = \sum_{n=1}^{N} w_n b_n(x)$$

Это самый простой способ объединить несколько алгоритмов в композицию. Иногда даже блендинг без обучения весов (то есть вариант с $w_1 = \dots = w_n = 1/N$) позволяет улучшить качество по сравнению с отдельными базовыми алгоритмами.

Блендинг



обучение
X
y
тест
X_2
обучение алгоритмов
обучение мета-алгоритма
train
обучаем несколько алгоритмов
их ответы на отложенном обучении
их ответы на тесте
обучаем мета-алгоритм
train
запускаем его на новой тестовой выборке
test
result

Дополнительные материалы

- Статья про стекинг и блендинг (рус.)
- Про приближение функции потерь в XGBoost рядом Тейлора (англ.)
- Реализация XGBoost с нуля (рус.)
- Сравнение CatBoost, lightGBM, XGBoost (англ.)
- Простой пример использования lightGBM (рус.)
- Примеры использования lightGBM из документации (англ.)
- Советы по настройке параметров lightGBM (англ.)

Дополнительные материалы

- Очень классная лекция про CatBoost (рус.)
- Туториал по CatBoost (англ.)
- Туториал по CatBoost (рус.)
- Туториал по CatBoost с заданиями (англ.) За решение заданий до конца апреля можно получить 5 бонусных баллов.
- Лекция про AdaBoost (англ.)