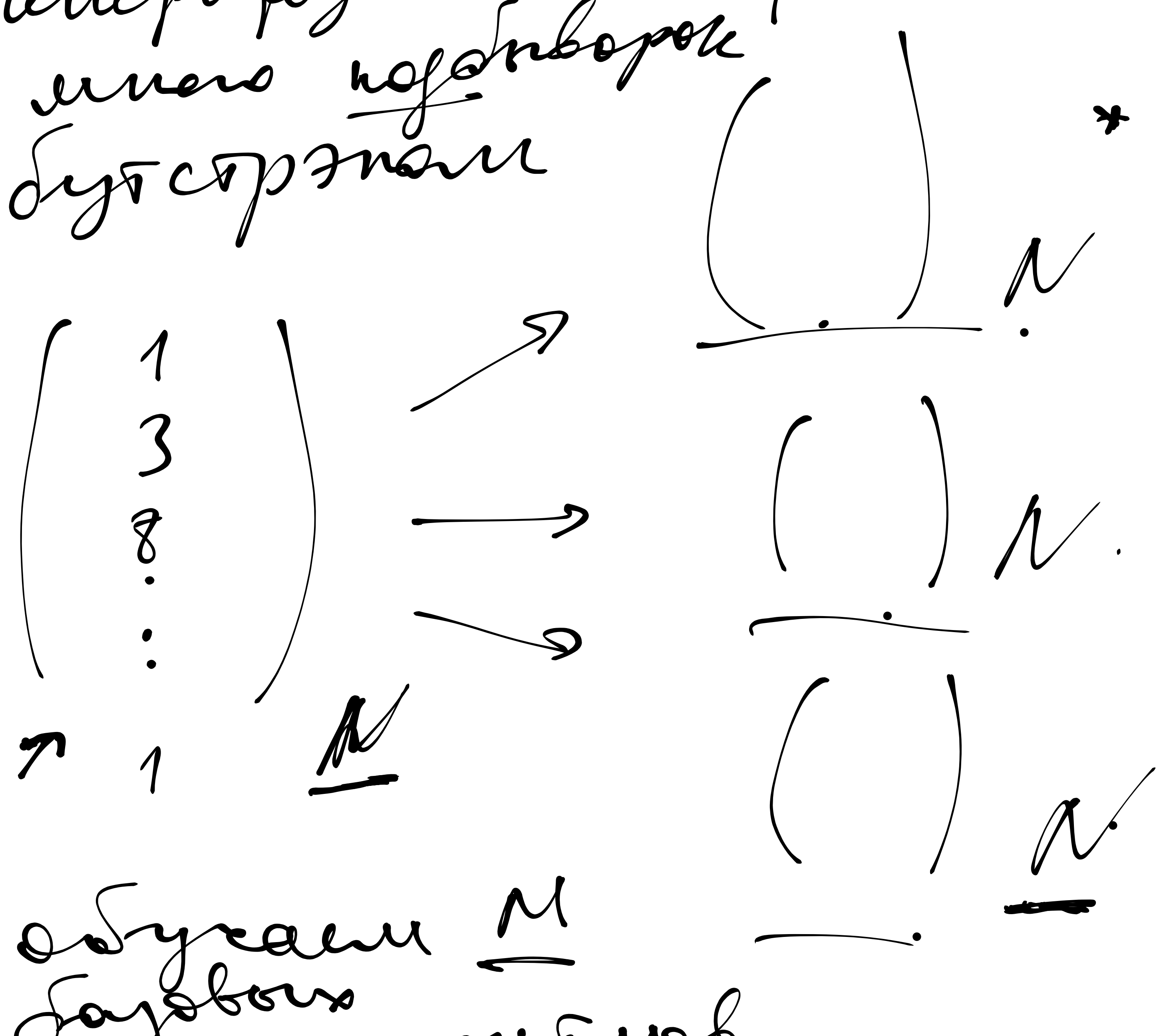
Ансамбли

① <u>Цель</u>

Хотим снизить разброс базовых алгоритмов

<u>bagging</u>	<u>Random Forest</u>
• генерируем много подвыборок бутстрэпом (M)	* — " — " —
	* выбираем признаки из случ. подмнож-ва



• обучаем M базовых алгоритмов

• усредняем ответы.

Гиперпараметры: M — число базовых алгоритмов

$\sqrt{p}$, $p/3$ — m — число признаков для разбиения

Out of bag samples

OOB error

Задачка

В среднем как много наблюдений не попадет в бутстрап. подвыборку.

$X = \{ x_1 \dots x_n \}.$

$X_1^* = \{ x_1^{*,1} \dots x_n^{*,1} \}.$

$X_2^* = \{ x_1^{*,2} \dots x_n^{*,2} \}$

$\vdots$

X_M^*

i-ый $- \frac{1}{n}$

$\left(1 - \frac{1}{n}\right)^n$ — вероятность того, что i-ый объект не попал в выборку

$$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^n \approx 0.37$$

Замеч.-ый предел:

$$\lim_{x\to\infty}\left(1+\frac{1}{x}\right)^x = e$$

$$\lim_{x\to\infty}\left(1+\frac{k}{x}\right)^x = e^k$$

$$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^{n\cdot m} = \left(e^{-1}\right)^m$$

$$e^{-1} \approx 1/2.7 \approx 0.37$$

E (числа объектов, не попавших в конечн. подвыборку)

$$n \cdot 0.37$$

№	1	2	3	4
x_1	1	2	3	4
y_i	6	6	12	18

$X_1^* = \{1, 1, 2, 3\}.$

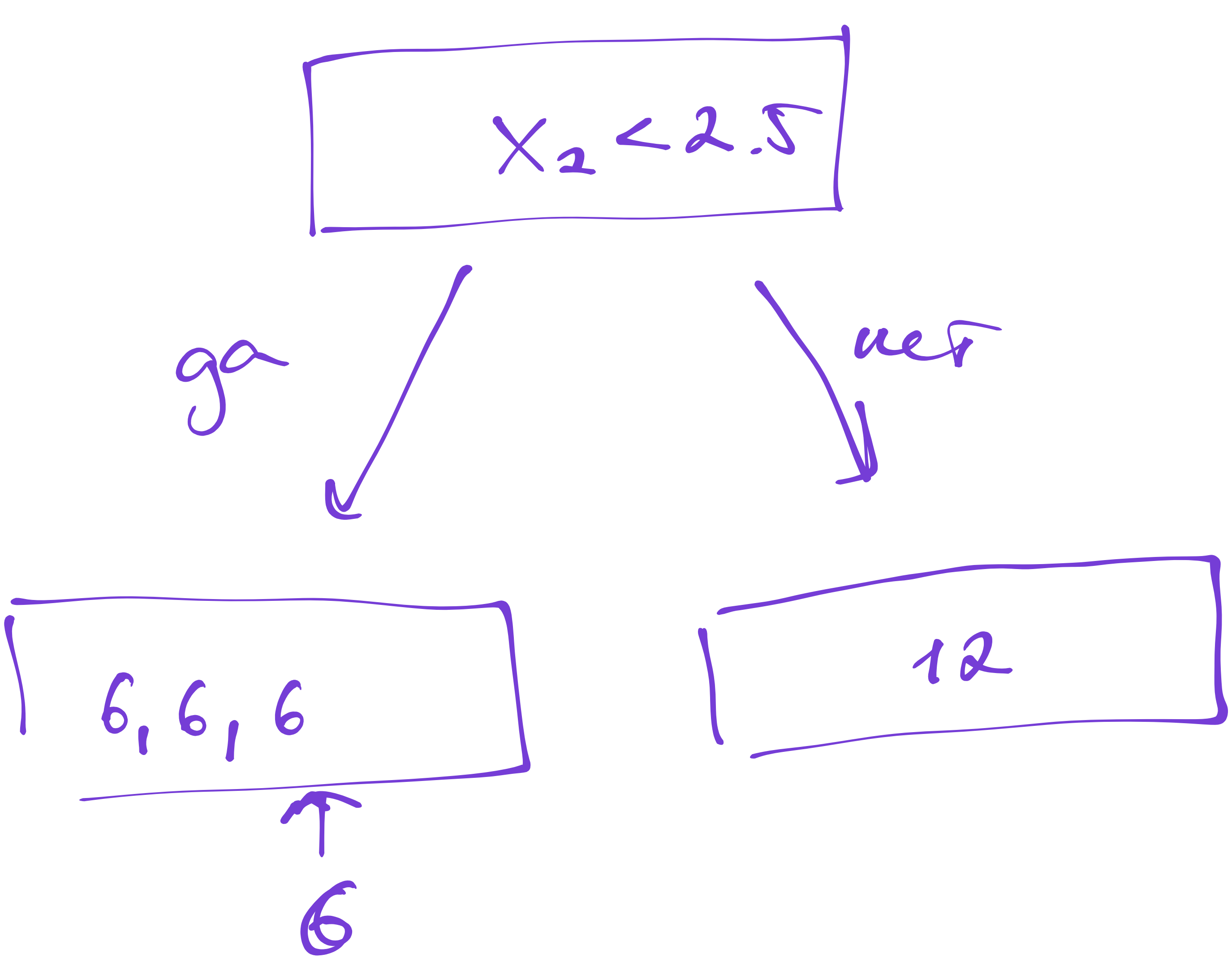
$X_2^* = \{2, 3, 4, 4\}.$

① минимизируя сумму квадратов ошибок

② $M = 2.$

x_1^*	1	1	2	3
y_i	6	6	6	12.

) max-depth = 1



x_1^*	2	3	4	4
y_i	6	12	18	18

$$\boxed{x_1 < 3.5}$$

да — нет

$$\boxed{6,\ 12} \qquad \boxed{18,\ 18}$$

$\hat{y}$ — 18

$$(\hat{y} - 6)^2 + (\hat{y} - 12)^2 \to \min_{y}$$

$$\hat{y} = \frac{6 + 12}{2} = 9$$

x_1	1	2	3	4
y_i	6	6	12	18

x_1	$\hat{y}_1$	$\hat{y}_2$	$\bar{y}$ прогноз композиции
1	6	9	$(6+9)/2 \approx 7.5$
2			
3			
4			

③ Градиентный бустинг

Идея: послед-но

① $a_0 := u = b_0(X)$ — константа, которая минимизир-т сумму квадратов ошибок

$$\arg\min_u \sum_{i=1}^{\ell} L(y_i, u) \to \min_u$$

② Обучаем M базовых алгоритмов

Что предсказывают $b_1 \dots b_m$?

параметр обуч-ия

③ $a_m(X) = \sum_{i=0}^{m} b_i \gamma_i$; $\gamma_0 = 0$

$= b_0(X) + \sum_{i=1}^{m} b_i \gamma_i$

$$b_1(X) = \underset{b}{\operatorname{argmin}} \sum (b(X) - s_i^{(1)})^2$$

сдвиг

$$s_i^{(1)} = -\frac{\partial L}{\partial a_0}$$

$$a_1 = b_0 + \gamma_1 \cdot b_1$$

$$b_2 = \underset{b}{\operatorname{argmin}} \sum_{i=1}^{\ell} (b(X) - s_i^{(2)})^2.$$

Квадратичная ф-ия потерь

$$L(y_i, a_{N-1}) = \frac{1}{2} \sum_{i=1}^{\ell} (y_i - a_{N-1})^2 \to \min_{a_{N-1}}$$

$$s_i^N = -\frac{\partial L}{\partial a_{N-1}} = -(-(y_i - a_{N-1})) =$$

$$= y_i - a_{N-1} \approx \text{остатоб}$$

$$b_N = \underset{b}{\operatorname{argmin}} \sum \left(b_i - (y_i - a_{N-1})\right)^2$$

Строим композицию из деревьев.

цена квартир

price	возраст	число комнат	S^2	$r^{(1)}$	$r^{(2)}$
480	5	3	90	336	
1040	11	2	60	-224	
350	14	5	80	466	
1810	8	8	120		
400	12	6	50		

$a_0 = b_0$

① $b_0 = \sum L(y_i, u) =$

$$= \sum_{i=1}^{5} (y_i - u)^2$$

$$\hat{u} = \sum y_i \cdot \frac{1}{5} = \frac{480 + 1040 + \dots}{5} = 816$$

② Строим дерево, которое будет предсказывать $r^{(1)}$

max-depth = 1

$\boxed{\text{число комнат} > 4}$

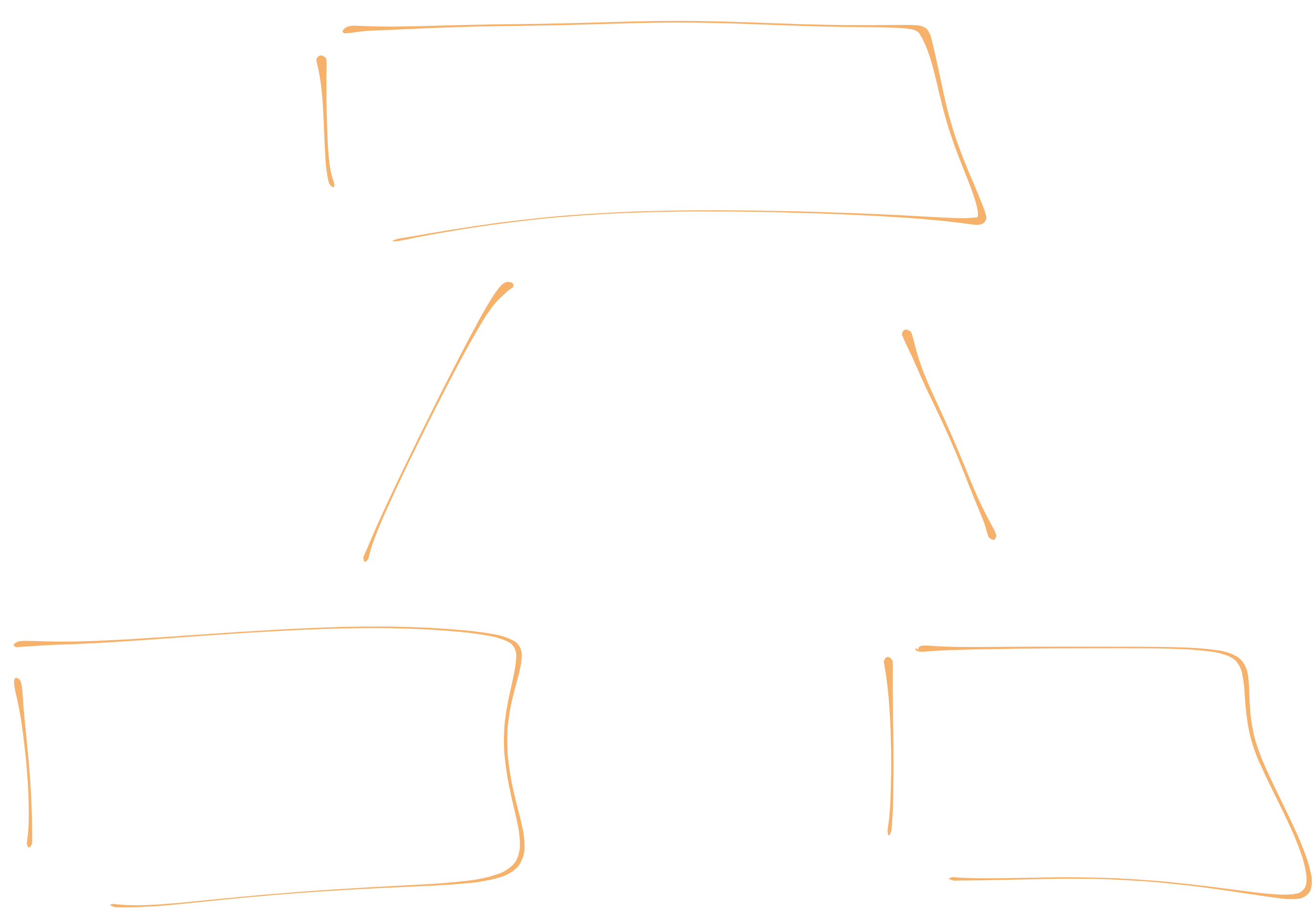
да → среднее: $\boxed{466 \dots}$

нет → среднее: $\boxed{336, -224}$

$r^{(2)}$

$$a_1 = b_0 + \overset{0.01}{\gamma_1} \cdot r^{(1)}$$

$$r^{(2)} = y_i - a_1$$



$$a_2 = b_0 + 0.01\, r^{(1)} + 0.01\, r^{(2)}$$

известно

b_N – известно

$\boxed{a_{N-1}}$

γ_N – неизвестен параметр обучения

В общем случае

$$\gamma_N = \arg\min \sum_{i=1}^{\ell} L\left(y_i, a_{N-1} + \gamma_N \cdot b_N\right) =$$

реал.

сумма модели

$$= \sum_{i=1}^{\ell} \left(\overbrace{y_i - a_{N-1}}^{s_i^{(N)}} - \gamma_N b_N\right)^2 \to \min$$

$$w_i^{(n)} = w_i^{(n-1)} - \eta \frac{\partial L}{\partial w_i^{(n-1)}} \to \min_{\eta}$$

или предсказали

Почему не делаем?

$$= \sum_{i=1}^{\ell} \left(s_i^{(N)} - \gamma_N b_N\right)^2 \to \min_{\gamma_N}$$

$$\gamma_N = \frac{\sum_{i=1}^{\ell} s_i b_N}{\sum b_N^2}$$

$$L(y, z) = |y - z| \quad - \ ?$$

$$g_i = -\frac{\partial L}{\partial z} = \operatorname{sign}(y - z)$$

AdaBoost — 1995

базовый алгоритм прибл. антигр. ф-ции в-от гр.

XgBoost

+

весa объектов зависят от антигр. ф-ции [illegible]

Light GBM

CatBoost (2017) → mean target encoding