Разбор домашнего задания №2 Тренировки по ML

Young&&Yandex ШАД

girafe ai

Arkadii Lysiakov

Outline

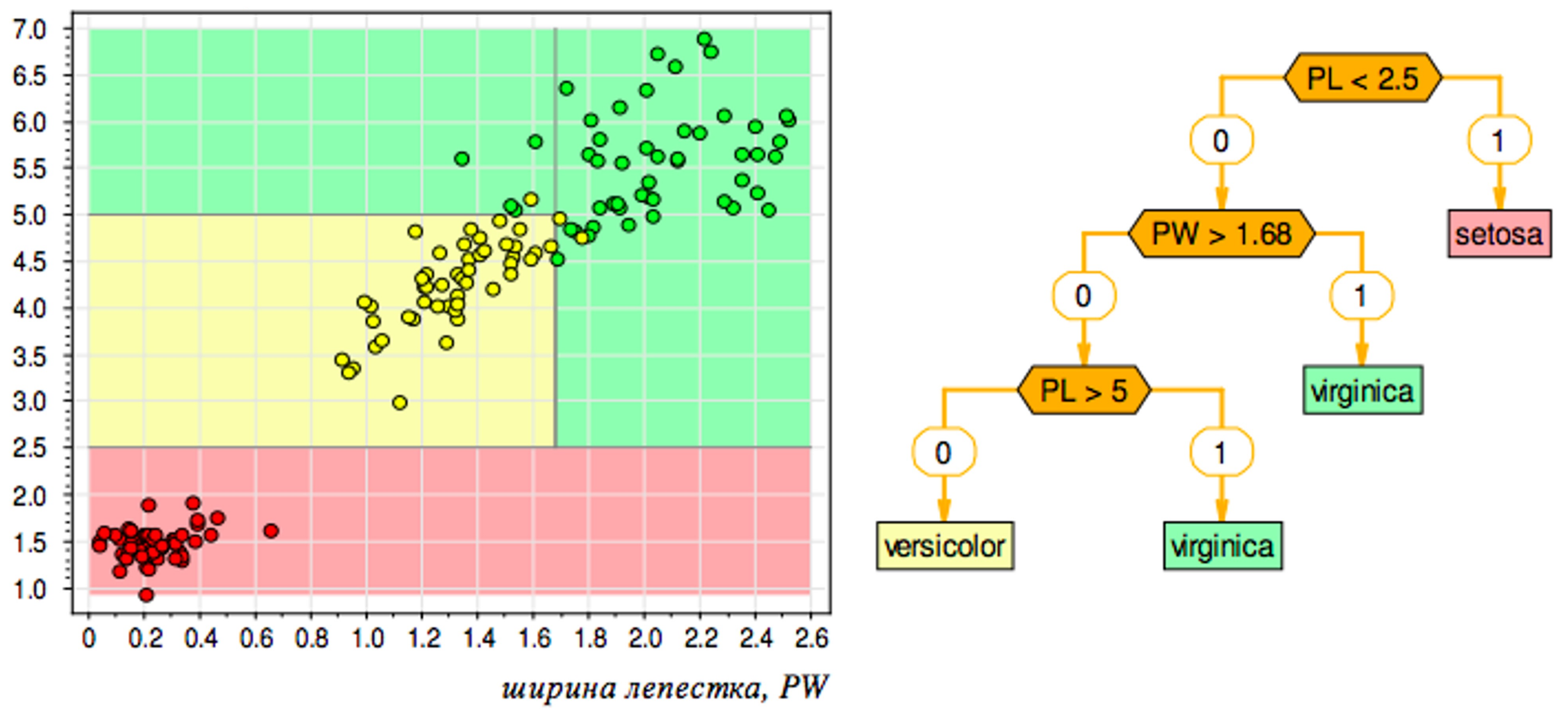
01 Decision tree

02 Bootstrap and Bagging

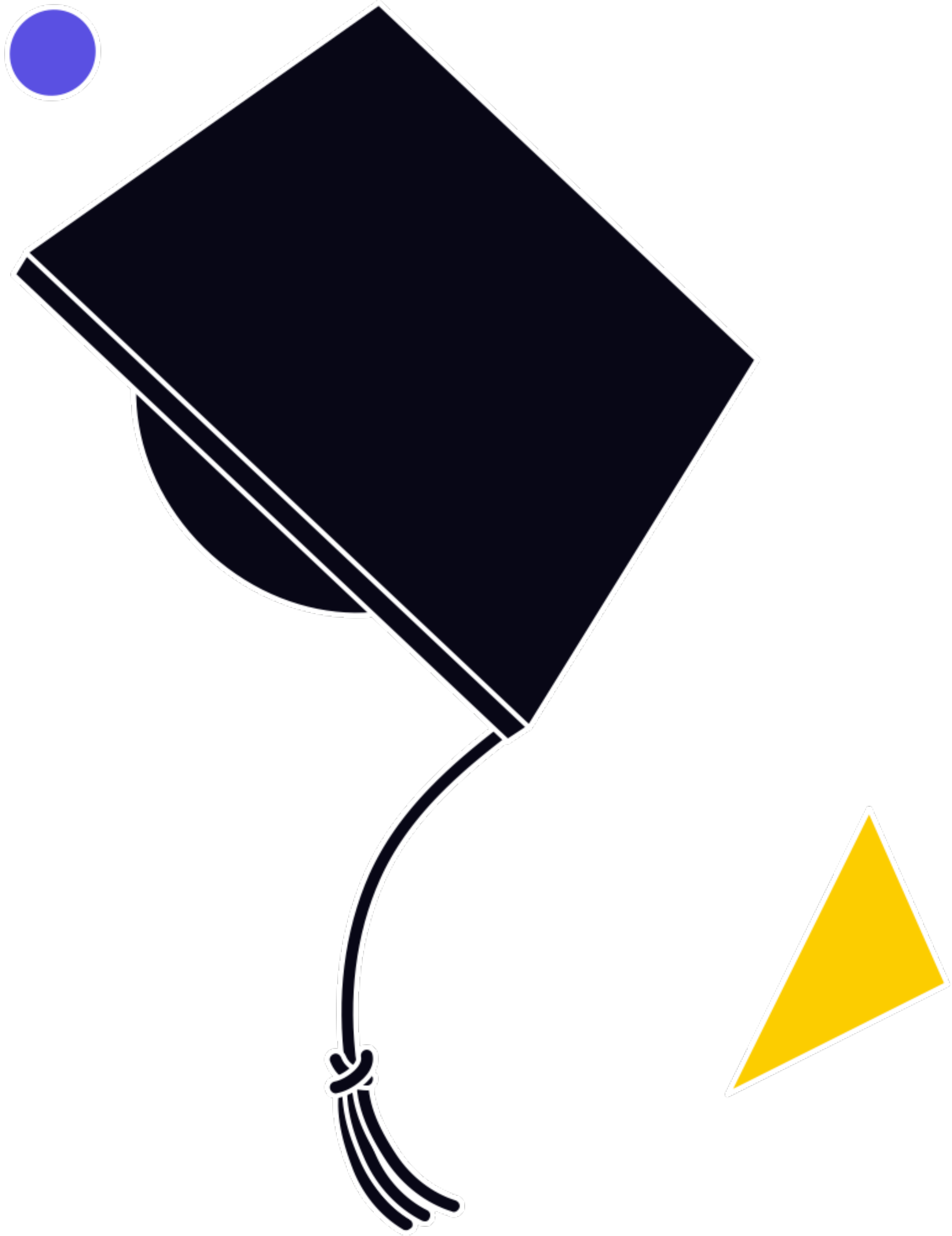
03 Random Forest

04 Boosting

Decision tree for Iris data set



setosa	$r_1(x) = [PL \leqslant 2.5]$
virginica	$r_2(x) = [PL > 2.5] \wedge [PW > 1.68]$
virginica	$r_3(x) = [PL > 5] \wedge [PW \leqslant 1.68]$
versicolor	$r_4(x) = [PL > 2.5] \wedge [PL \leqslant 5] \wedge [PW < 1.68]$



How to split data properly?

$\mathbb{Q}$

$x_j < t$

$\mathbb{T}$ $\mathbb{F}$

What is H?

$$\frac{|\mathbb{T}|}{|\mathbb{Q}|}H(\mathbb{T}) + \frac{|\mathbb{F}|}{|\mathbb{Q}|}H(\mathbb{F}) \longrightarrow \min_{j,t}$$

Information criteria

H(R) is measure of "heterogeneity" of our data.

Consider multiclass classification problem:

Obvious way:
Misclassification criteria:

$$H(\mathbb{T}) = 1 - \max_k(\{p_k\})$$

1. Entropy criteria:

$$H(\mathbb{T}) = -\sum_k p_k \log p_k$$

2. Gini impurity:

$$H(\mathbb{T}) = 1 - \sum_k p_k^2$$

Information criteria

H(R) is measure of “heterogeneity” of our data.

Consider regression problem:

1. Mean squared error

$$H(\mathbb{T}) = \min_c \frac{1}{|\mathbb{T}|} \sum_k (y^{(k)} - c)^2$$

Bootstrap

Consider dataset X containing m objects.

Pick m objects with return from X and repeat in N times to get N datasets.

Error of model trained on Xj:

$$\varepsilon_j(\boldsymbol{x}) = b_j(\boldsymbol{x}) - y(\boldsymbol{x}), \quad j = 1, \dots, N,$$

Then

$$\mathbb{E}_{\boldsymbol{x}} \left[b_j(\boldsymbol{x}) - y(\boldsymbol{x})\right]^2 = \mathbb{E}_{\boldsymbol{x}} \varepsilon_j^2(\boldsymbol{x}).$$

The mean error of N models:

$$E_1 = \frac{1}{N} \sum_{j=1}^{N} \mathbb{E}_{\boldsymbol{x}} \varepsilon_j^2(\boldsymbol{x}).$$

Bootstrap

This is a lie

Consider the errors ~~unbiased and uncorrelated:~~

$$\mathbb{E}_{\boldsymbol{x}}[\varepsilon_j(\boldsymbol{x})] = 0;$$

$$\mathbb{E}_{\boldsymbol{x}}[\varepsilon_i(\boldsymbol{x})\varepsilon_j(\boldsymbol{x})] = 0, \quad i \neq j.$$

$$E_N = \mathbb{E}_{\boldsymbol{x}} \left(\frac{1}{N} \sum_{j=1}^{N} b_j(\boldsymbol{x}) - y(\boldsymbol{x}) \right)^2$$

$$= \mathbb{E}_{\boldsymbol{x}} \left(\frac{1}{N} \sum_{j=1}^{N} \varepsilon_j(\boldsymbol{x}) \right)^2$$

$$= \frac{1}{N^2} \mathbb{E}_{\boldsymbol{x}} \left(\sum_{j=1}^{N} \varepsilon_j^2(\boldsymbol{x}) + \sum_{i \neq j} \varepsilon_i(\boldsymbol{x})\varepsilon_j(\boldsymbol{x}) \right)$$

$$\boxed{= \frac{1}{N} E_1.}$$

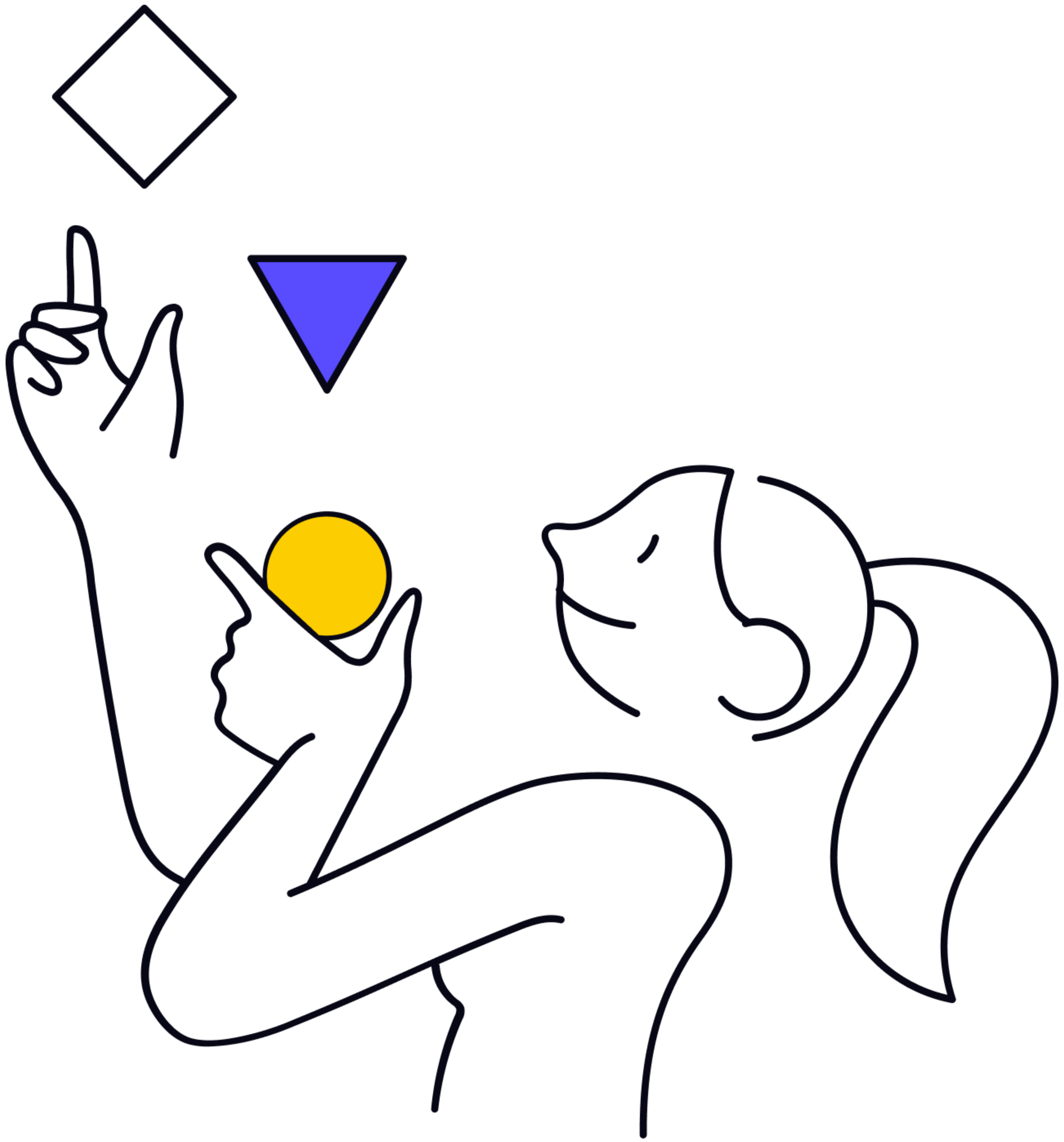
Error decreased by N times!

The final model averages all predictions:

$$a(\boldsymbol{x}) = \frac{1}{N} \sum_{j=1}^{N} b_j(\boldsymbol{x}).$$

Bagging = Bootstrap aggregating

Decreases the variance if the basic algorithms are not correlated



Bootstrap

Consider the errors unbiased and uncorrelated:

$$\mathbb{E}_{\boldsymbol{x}}[\varepsilon_j(\boldsymbol{x})] = 0;$$

$$\mathbb{E}_{\boldsymbol{x}}[\varepsilon_i(\boldsymbol{x})\varepsilon_j(\boldsymbol{x})] = 0, \quad i \neq j.$$

The final model averages all predictions:

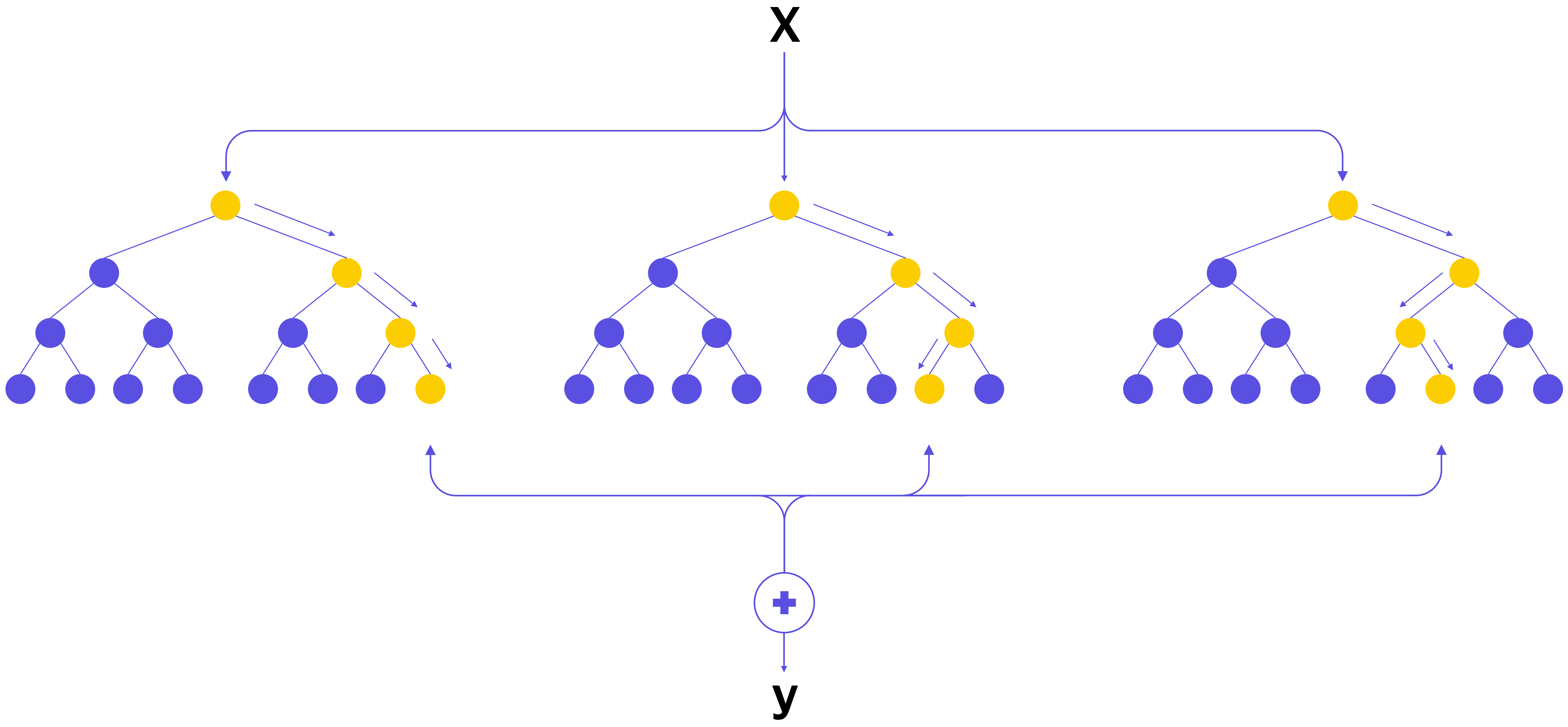
$$a(\boldsymbol{x}) = \frac{1}{N}\sum_{j=1}^{N} b_j(\boldsymbol{x}).$$

$$\begin{aligned} E_N &= \mathbb{E}_{\boldsymbol{x}}\left(\frac{1}{N}\sum_{j=1}^{N} b_j(\boldsymbol{x}) - y(\boldsymbol{x})\right)^2 \\ &= \mathbb{E}_{\boldsymbol{x}}\left(\frac{1}{N}\sum_{j=1}^{N} \varepsilon_j(\boldsymbol{x})\right)^2 \\ &= \frac{1}{N^2}\mathbb{E}_{\boldsymbol{x}}\left(\sum_{j=1}^{N} \varepsilon_j^2(\boldsymbol{x}) + \sum_{i \neq j} \varepsilon_i(\boldsymbol{x})\varepsilon_j(\boldsymbol{x})\right) \\ &\boxed{= \frac{1}{N}E_1.} \end{aligned}$$

Error decreased by N times!

Random Forest

Bagging + RSM = Random Forest



Random Forest

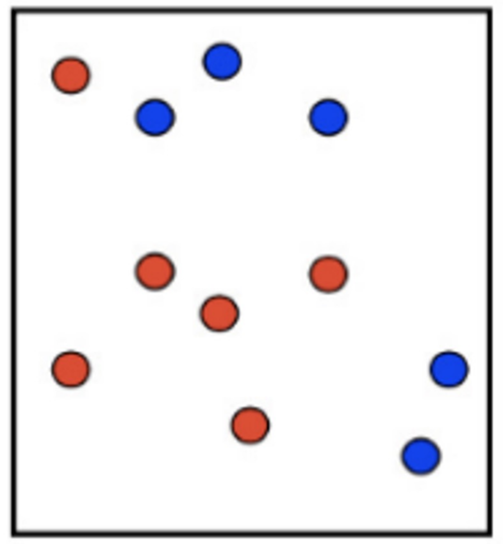
01 One of the greatest "universal" models

02 There are some modifications: Extremely Randomized Trees, Isolation Forest, etc.

03 Allows to use train data for validation: OOB

$$\mathrm{OOB} = \sum_{i=1}^{\ell} L\left(y^{(i)}, \frac{1}{\sum_{n=1}^{N}[\boldsymbol{x}^{(i)} \notin \boldsymbol{X}_n]} \sum_{n=1}^{N}[\boldsymbol{x}^{(i)} \notin \boldsymbol{X}_n] b_n(\boldsymbol{x}^{(i)})\right)$$

Boosting: AdaBoost



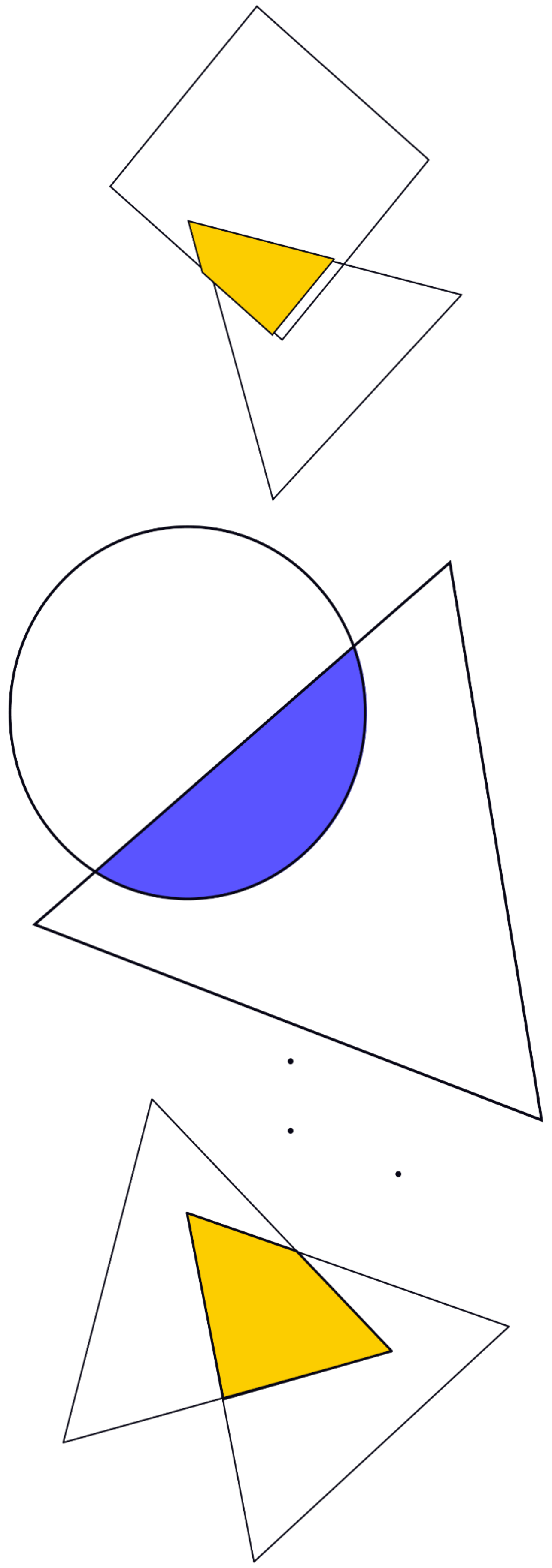
$$\hat{f}_T(\boldsymbol{x}) = \sum_{t=1}^{T} \rho_t h_t(\boldsymbol{x})$$

$$L(y^{(i)}, \hat{f}_T(\boldsymbol{x}^{(i)})) = \exp\left(-y^{(i)} \hat{f}_T(\boldsymbol{x}^{(i)})\right) = \exp\left(-y^{(i)} \sum_{t=1}^{T} \rho_t h_t(\boldsymbol{x}^{(i)})\right)$$

$$= \exp \overset{\text{const on step T}}{\boxed{\left(-y^{(i)} \sum_{t=1}^{T-1} \rho_t h_t(\boldsymbol{x}^{(i)})\right)}} \cdot \exp\left(-y^{(i)} \rho_T h_T(\boldsymbol{x}^{(i)})\right)$$

$$= w^{(i)} \cdot \exp\left(-y^{(i)} \rho_T h_T(\boldsymbol{x}^{(i)})\right)$$

Gradient boosting: theory



$$\hat{f}_{T-1}(\boldsymbol{x}) = \sum_{t=0}^{T-1} g_t(\boldsymbol{x}),$$

$$r_t^{(i)} = -\left[\frac{\partial L(y^{(i)}, f(\boldsymbol{x}^{(i)}))}{\partial f(\boldsymbol{x}^{(i)})}\right]_{f(\boldsymbol{x})=\hat{f}_T(\boldsymbol{x})}, \quad \text{for } i = 1, \ldots, n,$$

$$\boldsymbol{\theta}_T = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(r_t^{(i)} - h(\boldsymbol{x}^{(i)}, \boldsymbol{\theta})\right)^2,$$

$$\rho_t = \arg\min_{\rho} \sum_{i=1}^{n} L\left(y^{(i)}, \hat{f}_{t-1}(\boldsymbol{x}^{(i)}) + \rho \cdot h(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}_T)\right)$$

Thanks for attention!

Questions?

girafe
ai