Bias-variance decomposition Gradient boosting

Vladislav Goncharenko
ML researcher

girafe ai

ITMO, spring 2024

Outline

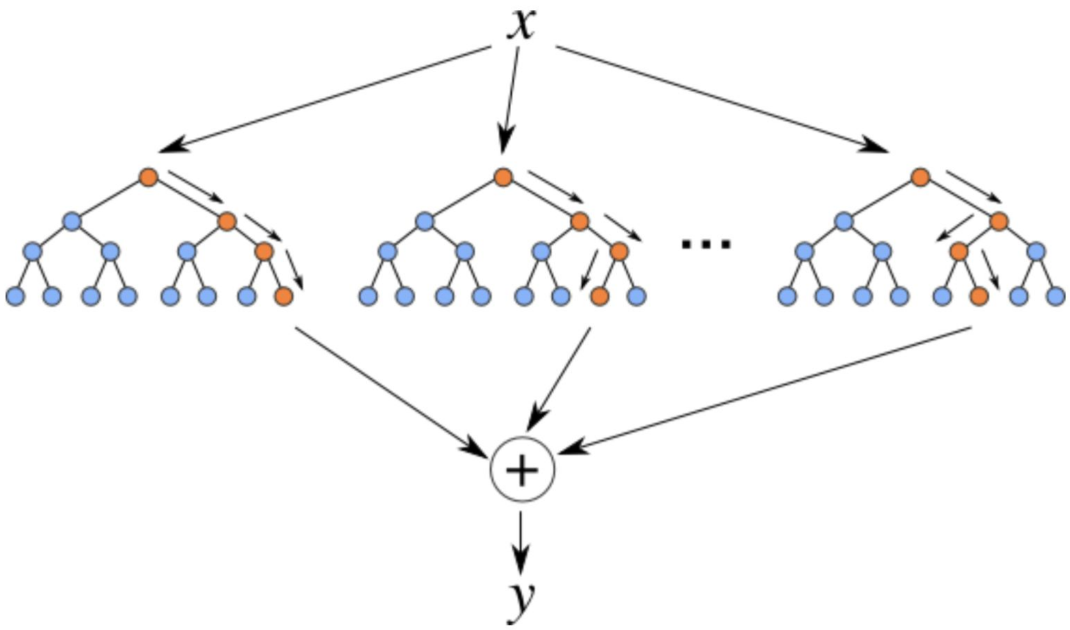
1. Intuitions
2. Gradient boosting theory
3. Examples
4. Libraries
5. Feature importances
6. Hyperparameter optimization

Ensembling recap

00

Random Forest

Bagging + RSM = Random Forest



Random Forest

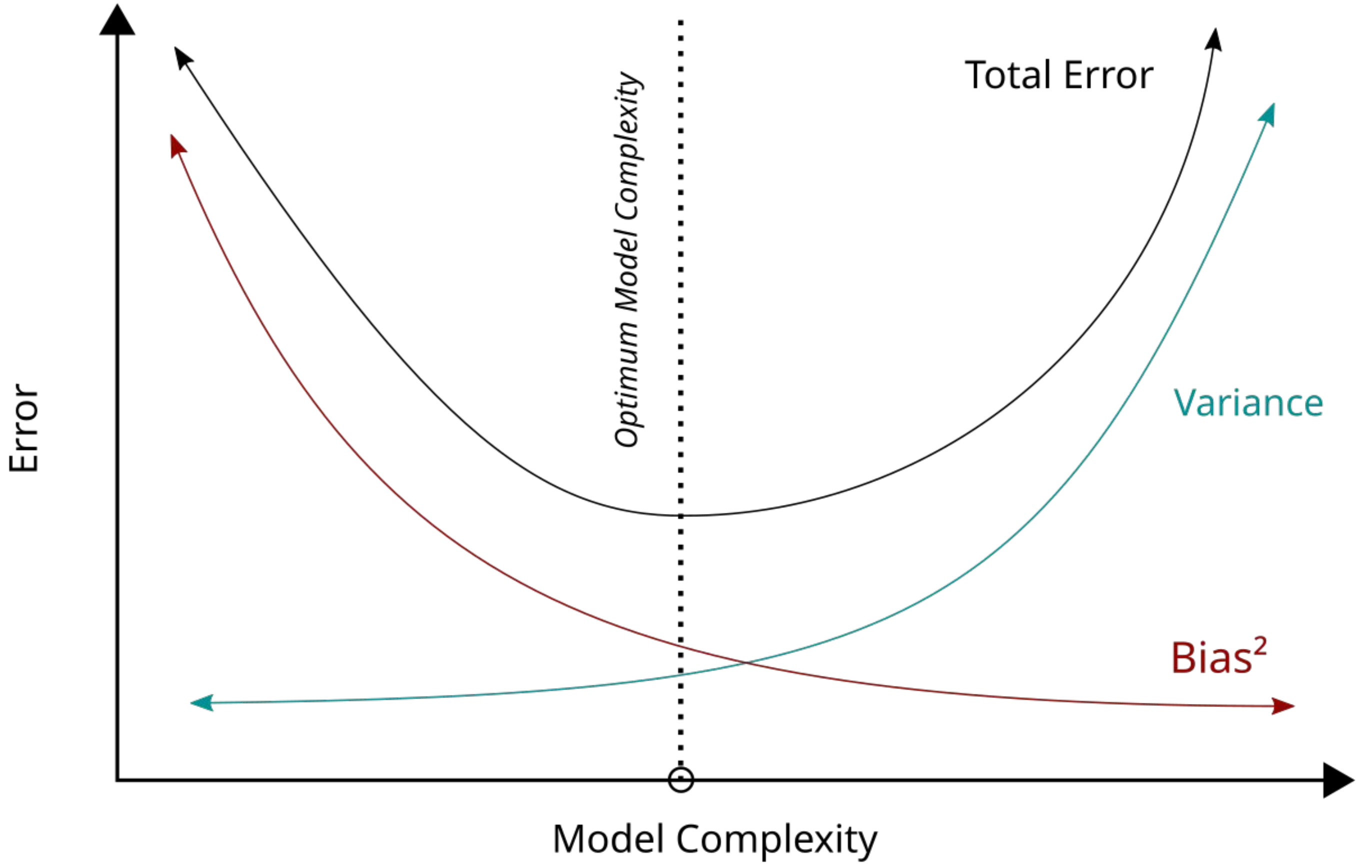
- One of the greatest “universal” models
- There are some modifications: Extremely Randomized Trees, Isolation Forest, etc.

Bias-variance decomposition

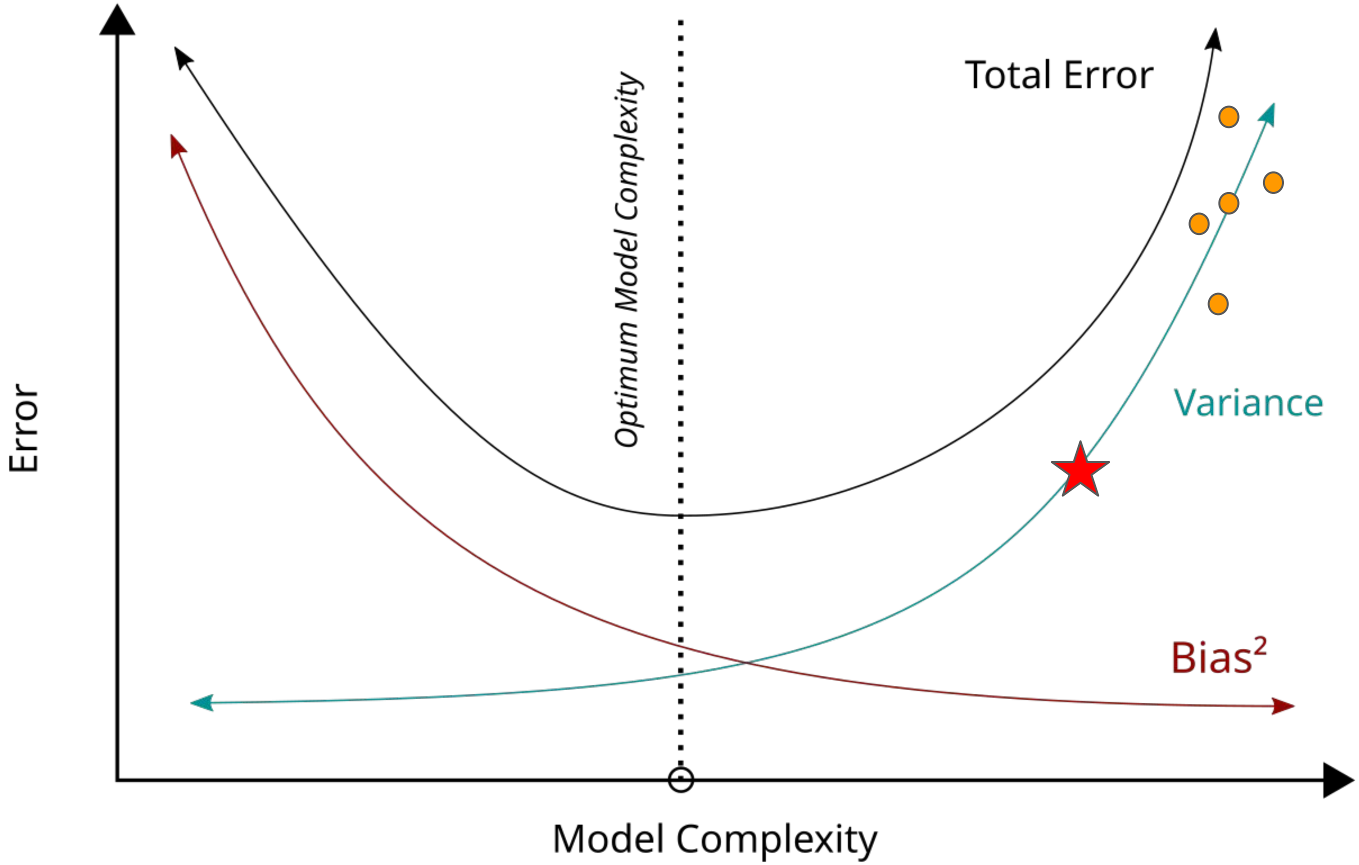
girafe
ai

01

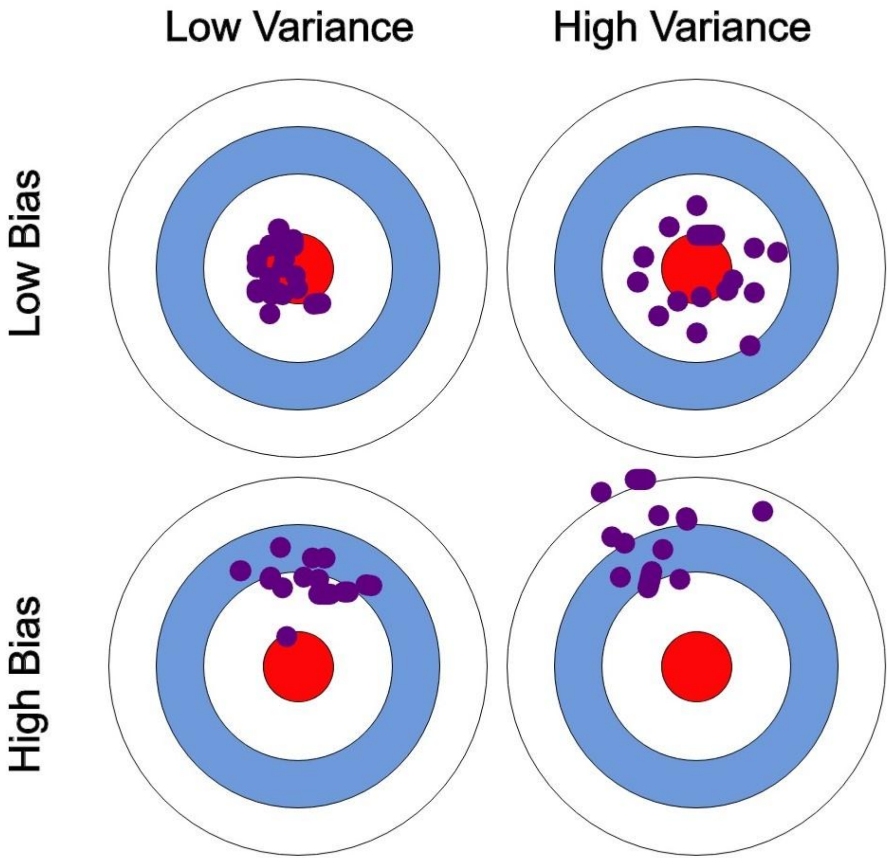
Bias-variance tradeoff



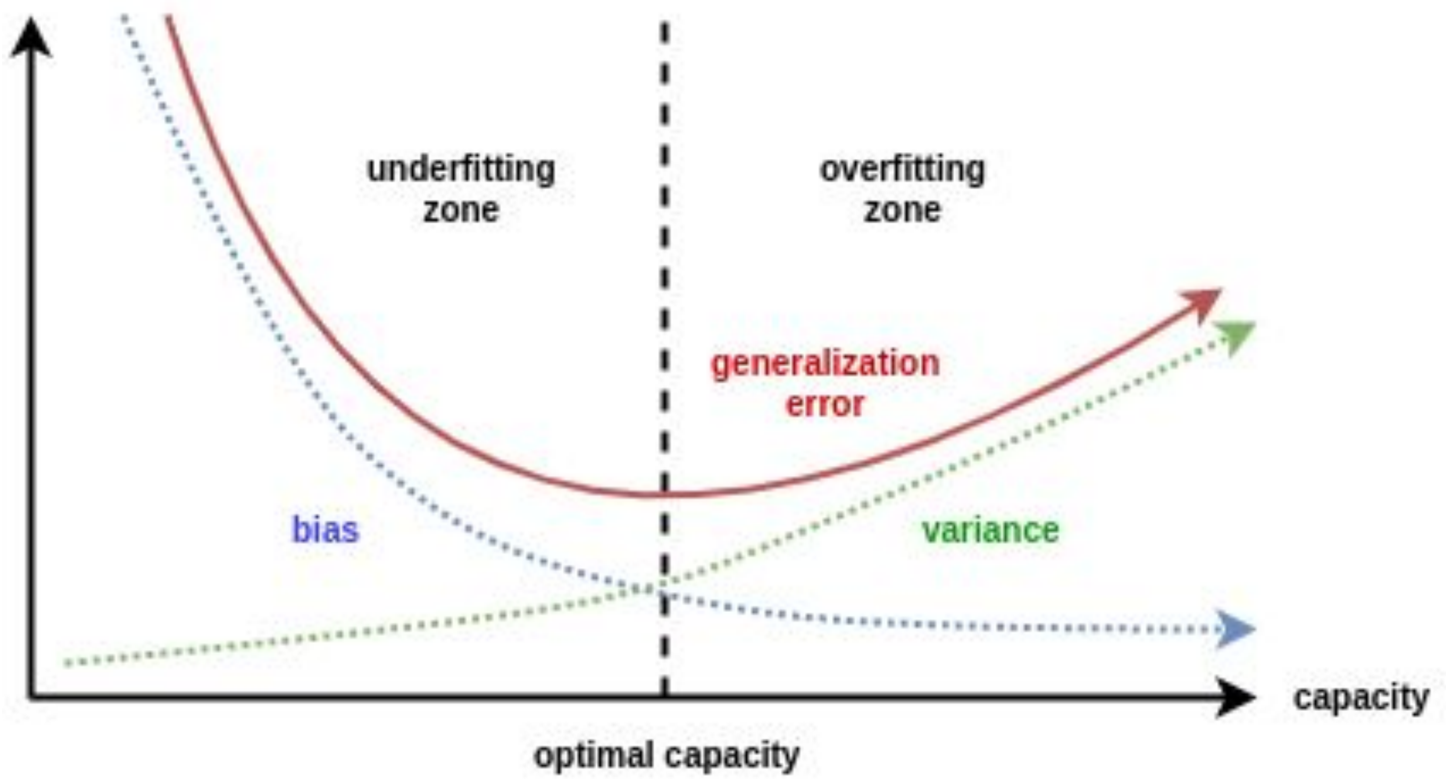
Bagging motivation



Bias-variance tradeoff



Bias-variance tradeoff



Randomness in error

1. Random sampling in training procedure
   a. for Linear model - initial weights
   b. for Tress - feature and threshold sampling
   c. for SGD - sampling of batches
2. Computational errors due to limited precision arithmetics
3. Target noise
   a. epsilon in regression model
4. Objects sampling

GB derivation for MSE

$$L \in R; y \in R^n; p \in R^n$$

$$

\textstyle{\frac{\partial}{\partial{p}}} L(y, p) = [{\frac{\partial}{\partial{p_1}}} L(y, p), {\frac{\partial}{\partial{p_2}}} L(y, p), ...]

$$

$$

1/n \cdot \textstyle{\frac{\partial}{\partial{p}}} \sum_i (y_i - p_i)^2 = 1/n [-2 (y_1 - p_1), -2 (y_2 - p_2), ...]

$$

GB derivation for MSE

$$f_1(x) = fit(x; \bar{y}); \quad g_i = \sum_{k=1}^{i} f_i(x)$$

$$\bar{f}_1 = \begin{pmatrix} f_1(x_1) \\ f_2(x_2) \\ \vdots \end{pmatrix}; \quad \bar{g}_i = \begin{pmatrix} g_i(x_1) \\ g_i(x_2) \\ \vdots \end{pmatrix}$$

$$L(\bar{y}; \bar{g}_2) \to \min$$

$$\frac{\partial L(\bar{y}; \bar{g}_2)}{\partial \bar{f}_2} = \frac{\partial L(\bar{y}; \bar{g}_1 + \bar{f}_2)}{\partial \bar{f}_2} =$$

$$= \frac{\partial L(\bar{y}; \bar{q})}{\partial \bar{q}} \cdot \frac{\partial (\bar{g}_1 + \bar{f}_2)}{\partial \bar{f}_2} =$$

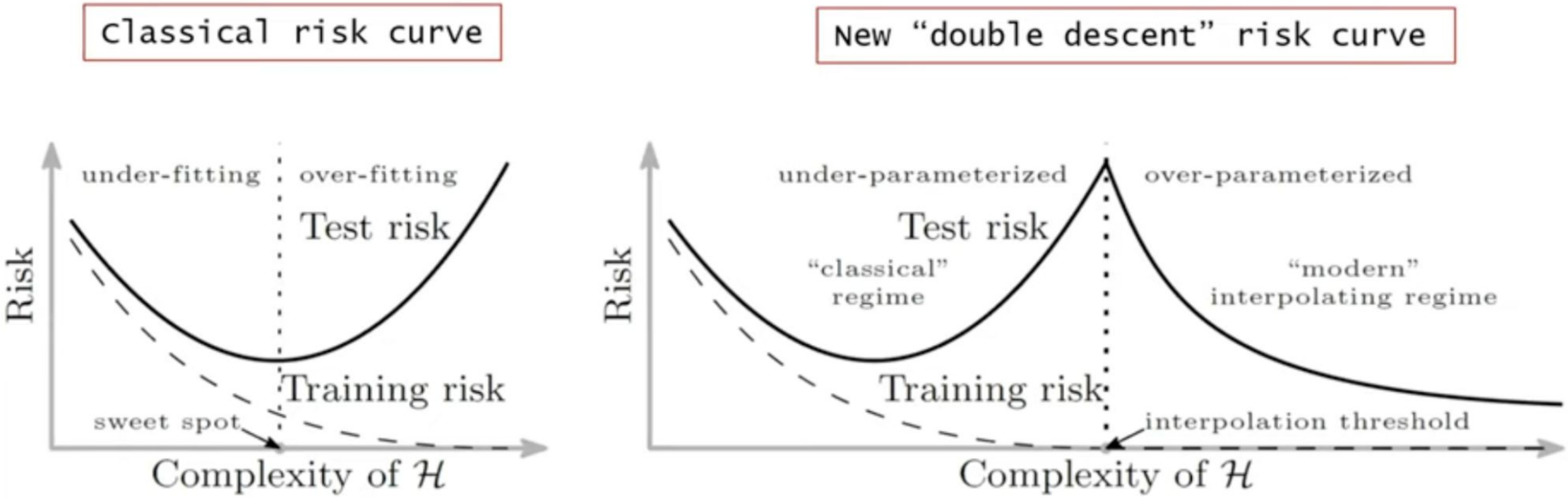
$$= \left. \frac{\partial L(\bar{y}; \bar{q})}{\partial q} \right|_{q = g_1}$$

Bias-variance decomposition

$$\mathrm{MSE} = \left(f(x) - \mathbb{E}\left[\hat{f}\,(x)\right]\right)^2 + \mathbb{E}\left[\left(\mathbb{E}\left[\hat{f}\,(x)\right] - \hat{f}\,(x)\right)^2\right] + \sigma^2$$

$$= \mathrm{Bias}\left(\hat{f}\,(x)\right)^2 \; + \; \mathrm{Var}\left[\hat{f}\,(x)\right] \; + \; \sigma^2$$

Contemporary hypothesis



For contemporary LLMs and other big models "double descent" theory is being developed, see https://www.youtube.com/watch?v=waJOSLNhHII or https://www.youtube.com/watch?v=5-QjjOYfeSI

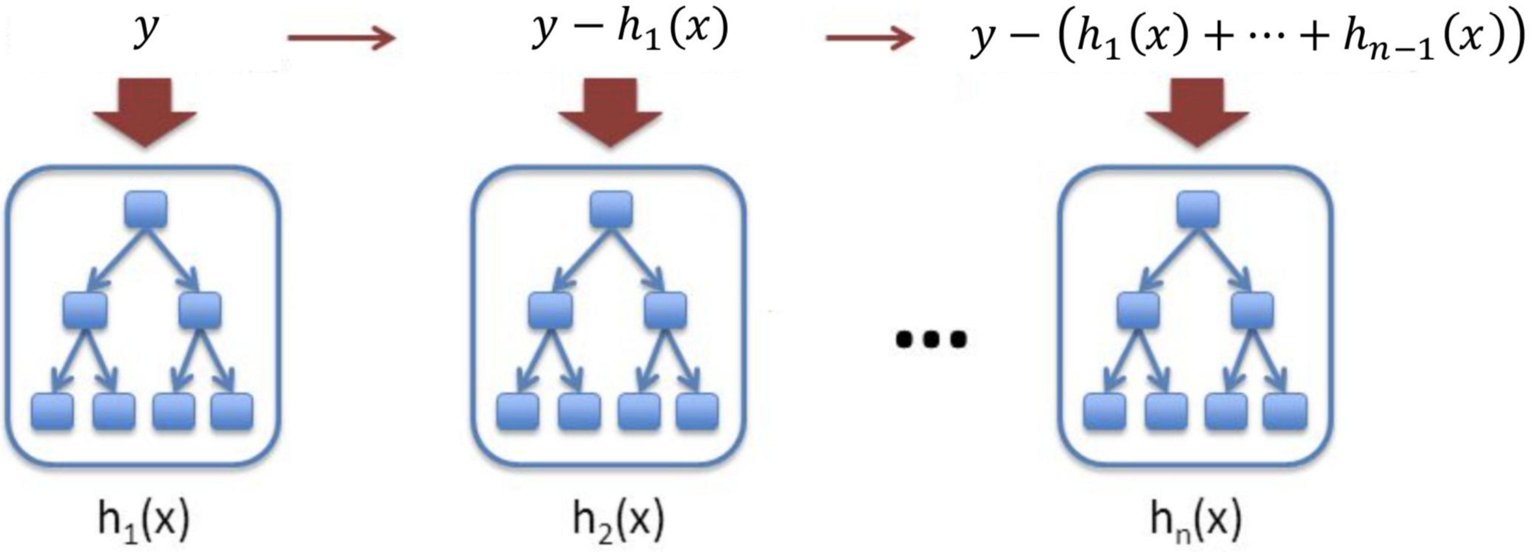
Boosting intuition

girafe
ai

02

Boosting for _MSE_

$$a_n(x) = h_1(x) + \dots + h_n(x)$$

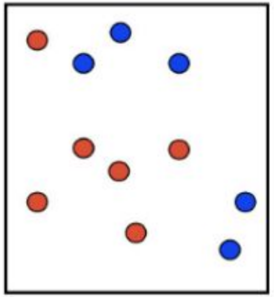


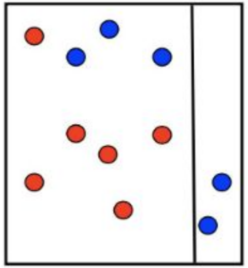
* in case of MSE loss

Boosting: intuition

Binary classification

Use decision stumps



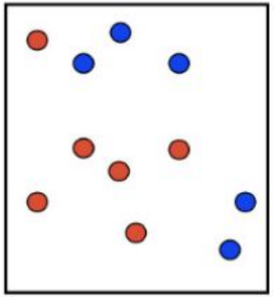


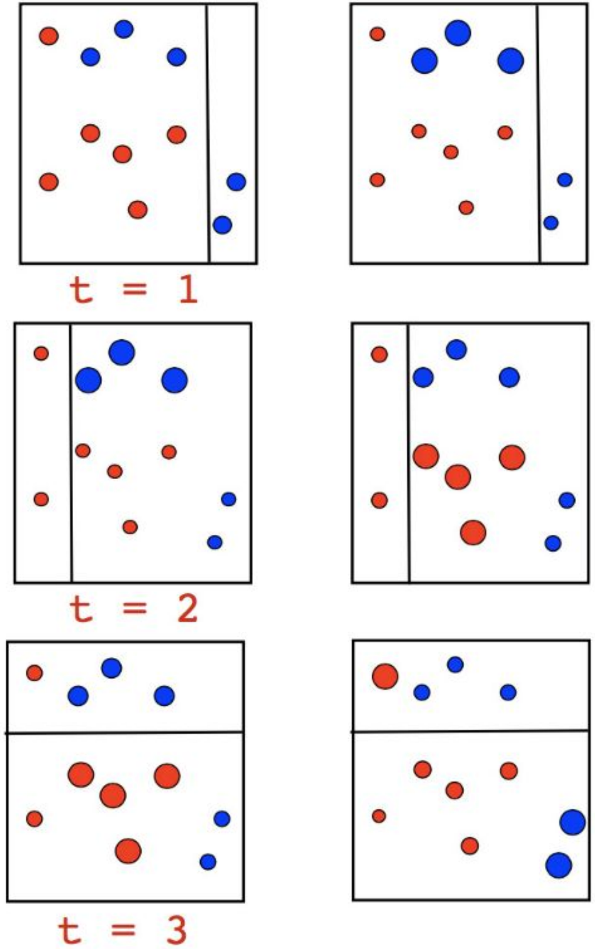
t = 1

Boosting: intuition

Binary classification

Use decision stumps.

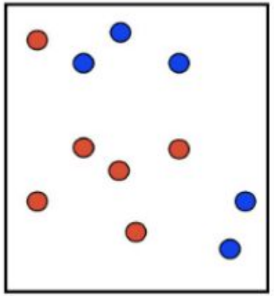


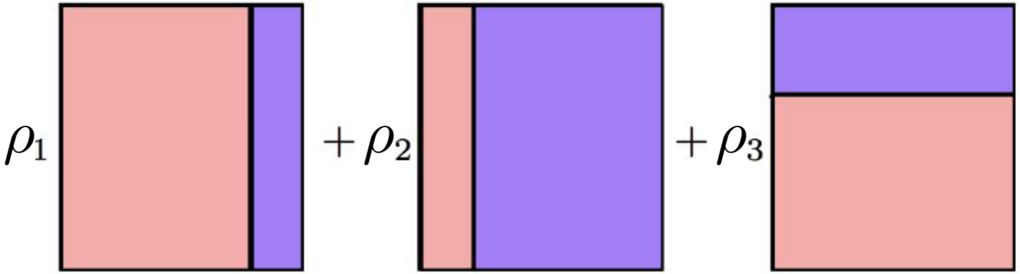


Boosting: intuition

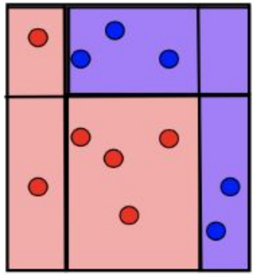
Binary classification

Use decision stumps.





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



Gradient boosting theory

girafe
ai

02

Gradient boosting: theory

Denote dataset $\{(x_i, y_i)\}_{i=1,\ldots,n}$, loss function $L(y, f)$.

Optimal model:

$$\hat{f}(x) = \arg\min_{f(x)} L(y, f(x)) = \arg\min_{f(x)} \mathbb{E}_{x,y}[L(y, f(x))]$$

Let it be from parametric family:

$$\hat{f}(x) = f(x, \hat{\theta}),$$

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_{x,y}[L(y, f(x, \theta))]$$

Gradient boosting: theory

$$\hat{f}(x) = \sum_{i=0}^{t-1} \hat{f}_i(x),$$

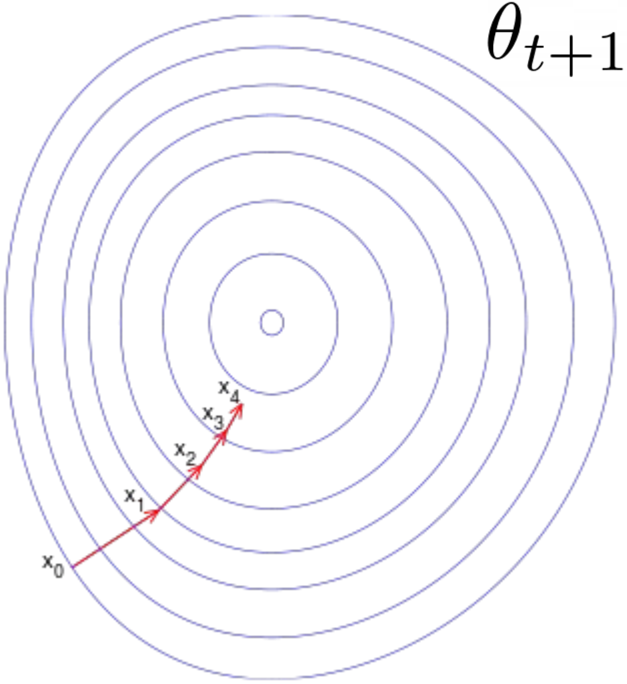
$$(\rho_t, \theta_t) = \arg\min_{\rho,\theta} \mathbb{E}_{x,y}[L(y, \hat{f}(x) + \rho \cdot h(x, \theta))],$$

$$\hat{f}_t(x) = \rho_t \cdot h(x, \theta_t)$$

What if we could use gradient descent in space of our models?

Gradient boosting: theory

$$\theta_{t+1} = \theta_t - \text{learning rate} \cdot \frac{\partial}{\partial \theta} Loss$$



What if we could use gradient descent in space of our models?

Gradient boosting: theory

$$\hat{f}(x) = \sum_{i=0}^{t-1} \hat{f}_i(x),$$

$$r_{it} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{f}(x)}, \quad \text{for } i = 1, \ldots, n,$$

$$\theta_t = \arg\min_{\theta} \sum_{i=1}^{n} (r_{it} - h(x_i, \theta))^2,$$

$$\rho_t = \arg\min_{\rho} \sum_{i=1}^{n} L(y_i, \hat{f}(x_i) + \rho \cdot h(x_i, \theta_t))$$

Gradient boosting: theory

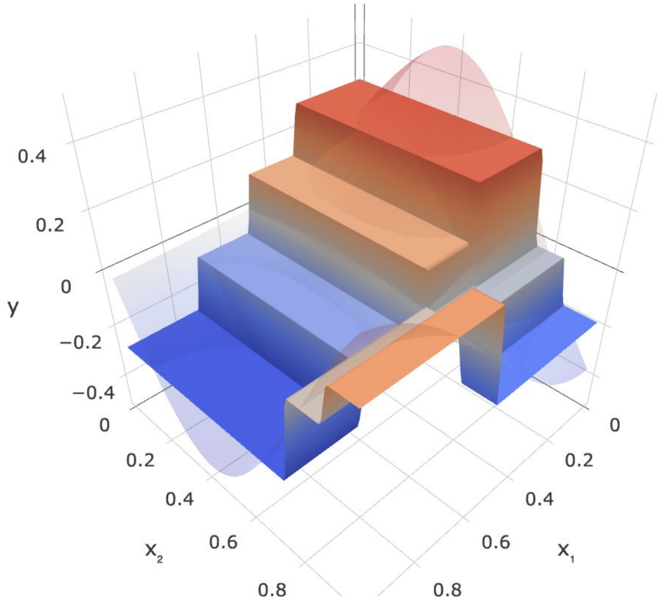
In linear regression case with MSE loss:

$$r_{it} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{f}(x)} = -2(\hat{y}_i - y_i) \propto \hat{y}_i - y_i$$

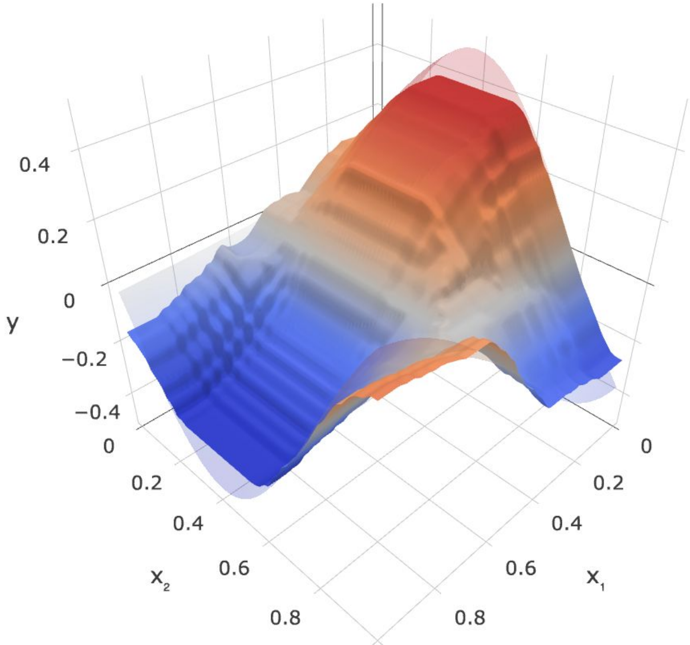
GB examples

03

Tree vs GB demo



One tree



Boosting

http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html

Gradient boosting

What we need:

- Data
- Loss function and its gradient
- Family of algorithms (with constraints if necessary)
- Number of iterations M
- Initial value (GBM by Friedman): constant

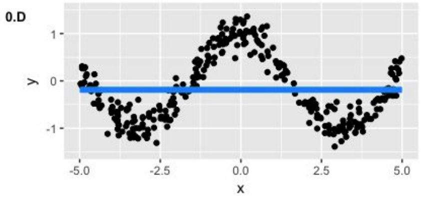
Gradient boosting: example

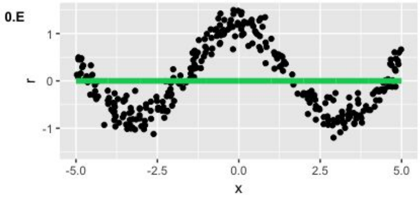
What we need:

- Data: toy dataset $y = cos(x) + \epsilon, \epsilon \sim \mathcal{N}(0, \frac{1}{5}), x \in [-5, 5]$
- Loss function: MSE
- Family of algorithms: decision trees with depth 2
- Number of iterations M = 3
- Initial value: just mean valu

https://habr.com/ru/company/ods/blog/327250/

Gradient boosting: example

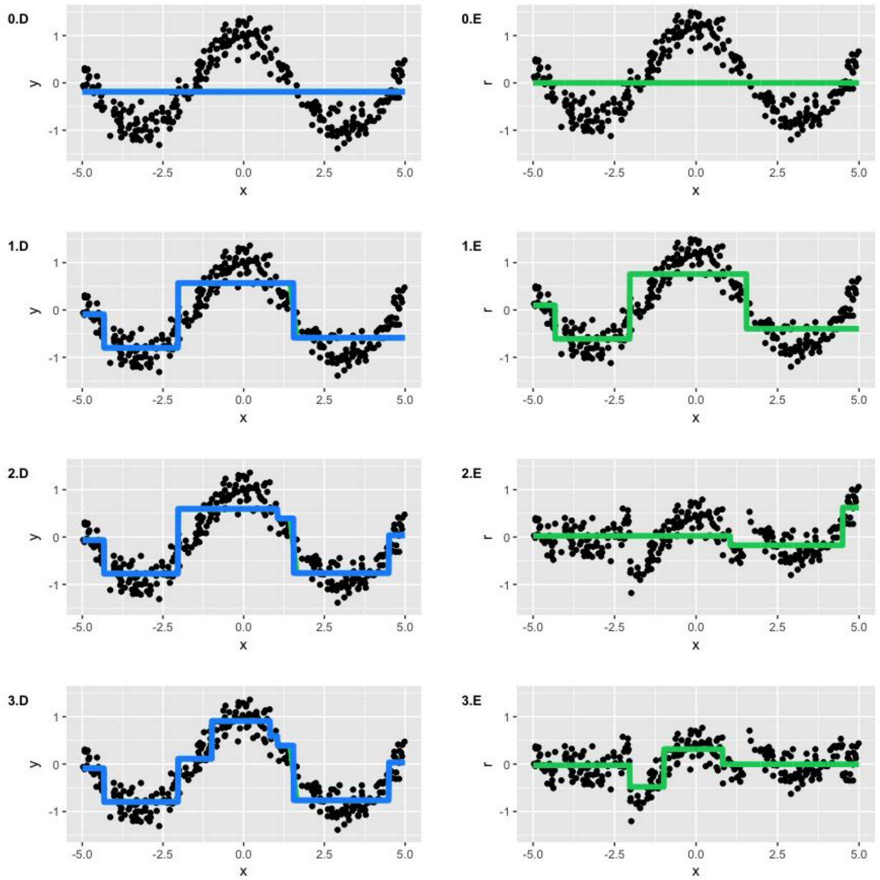




Left: full ensemble on each step.

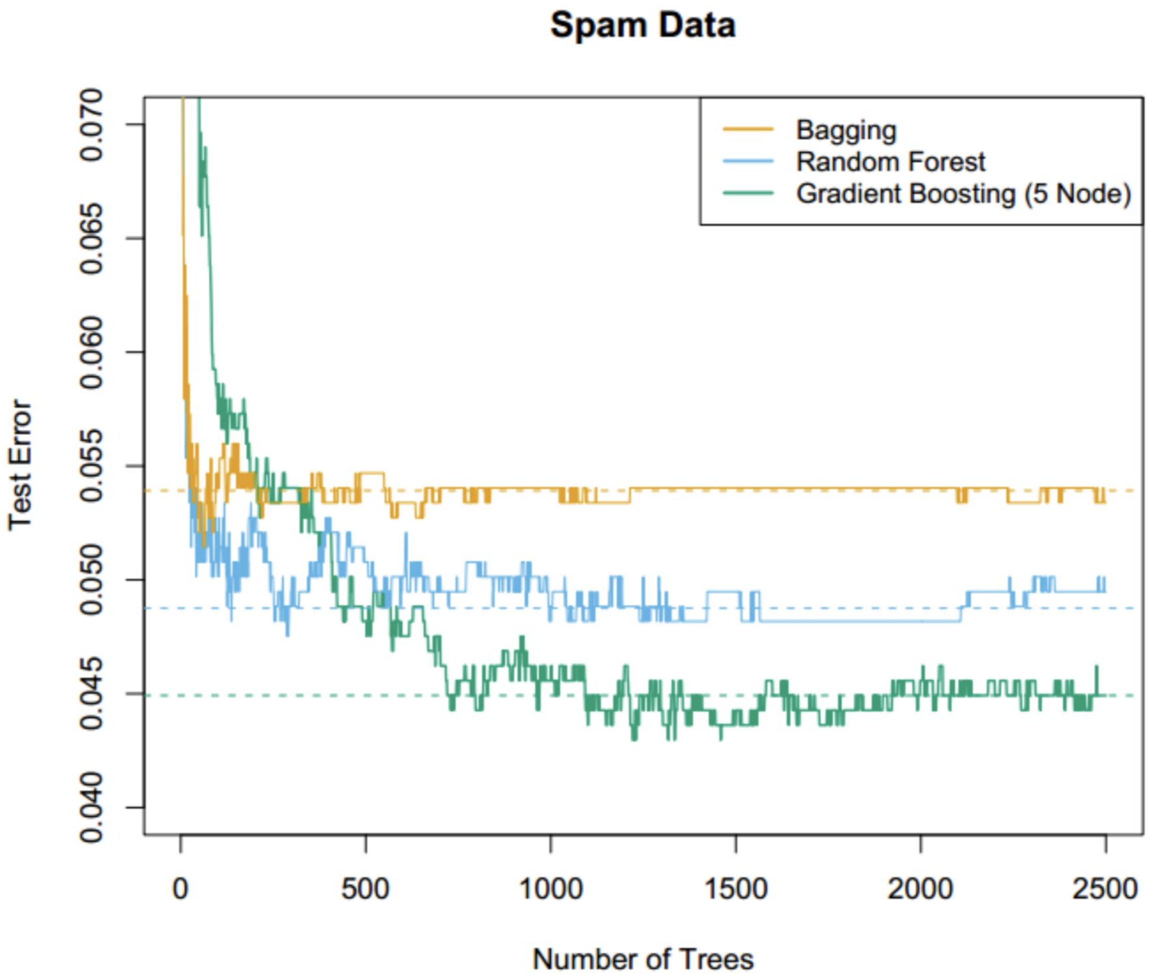
Right: additional tree decisions.

Gradient boosting: example



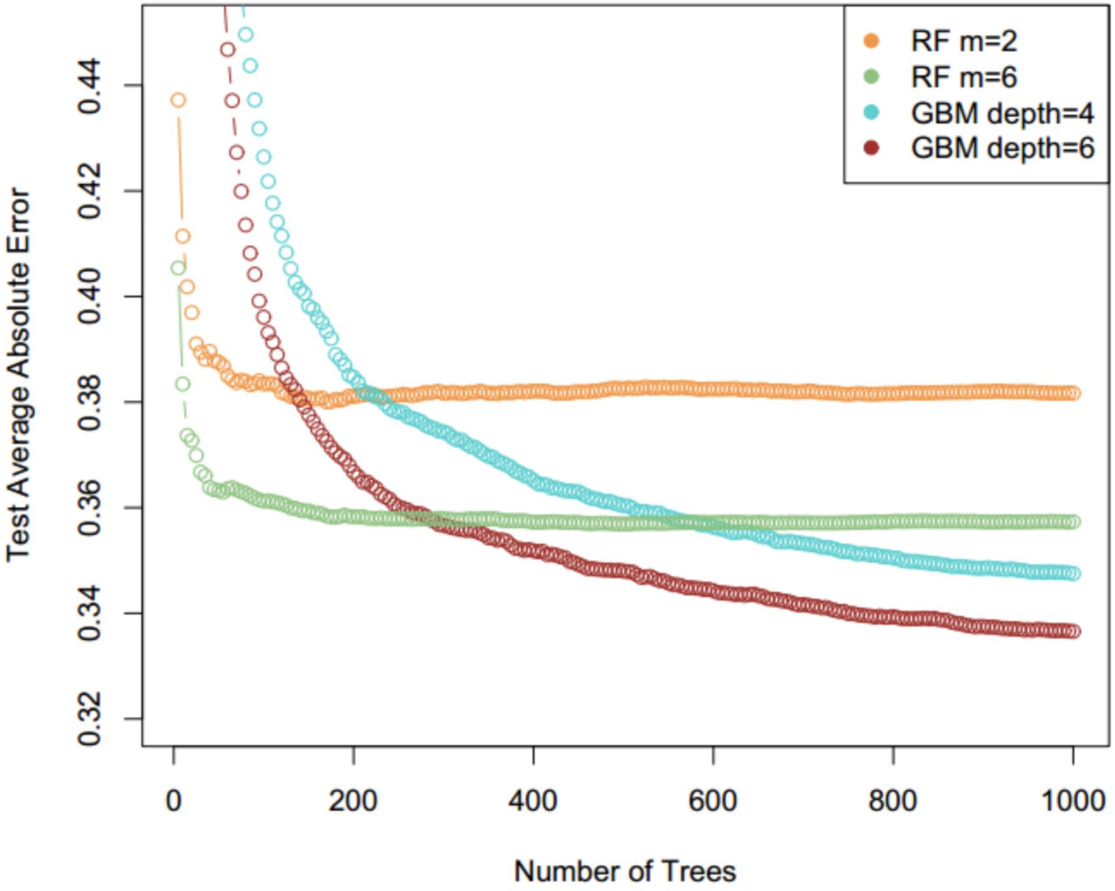
Left: full ensemble on each step.

Right: additional tree decisions



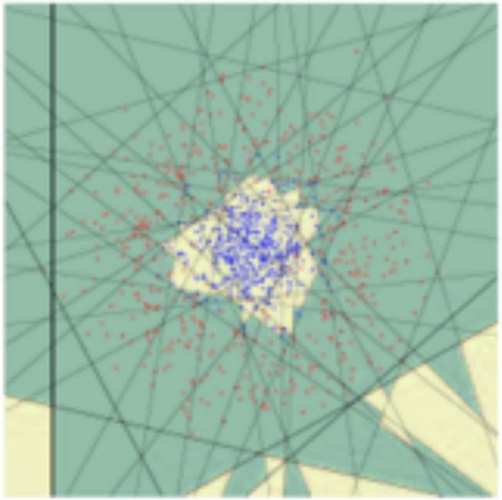
Spam Data
Bagging
Random Forest
Gradient Boosting (5 Node)
Test Error
0.040
0.045
0.050
0.055
0.060
0.065
0.070
0
500
1000
1500
2000
2500
Number of Trees

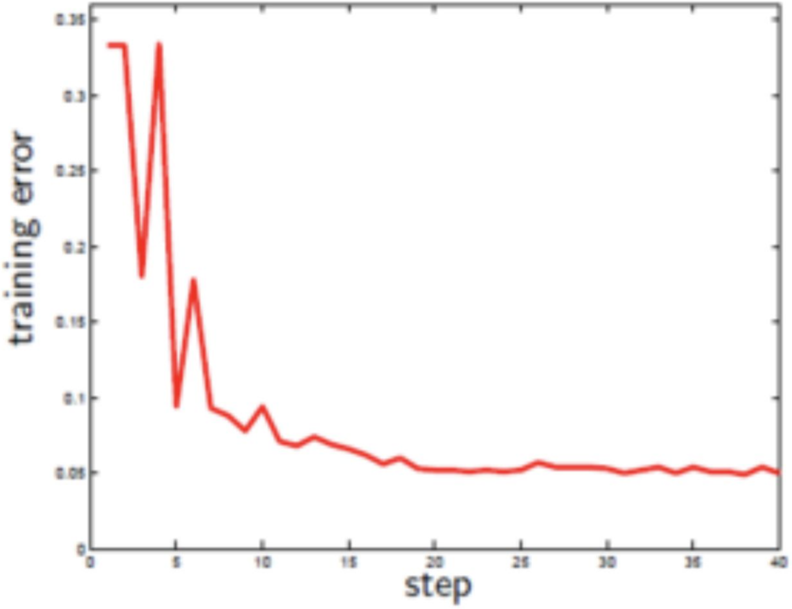
California Housing Data



Boosting with linear classification methods

$t = 40$





Ensembles parallelization

	Training	Inference
Bagging	parallel	parallel
Boosting	sequential	parallel

Libraries for GB

girafe
ai

04

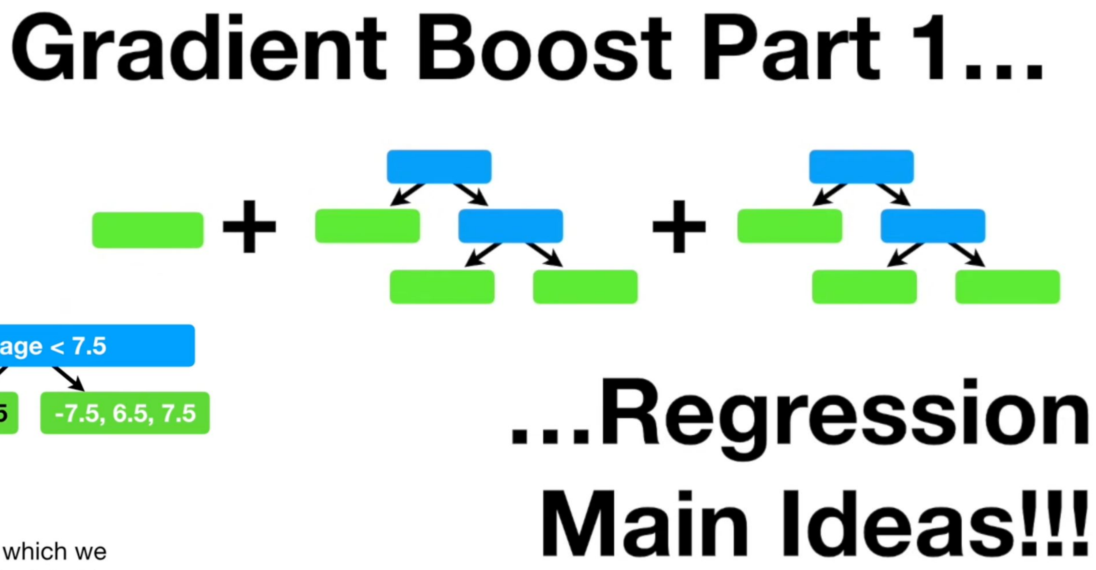
Main contemporary instruments

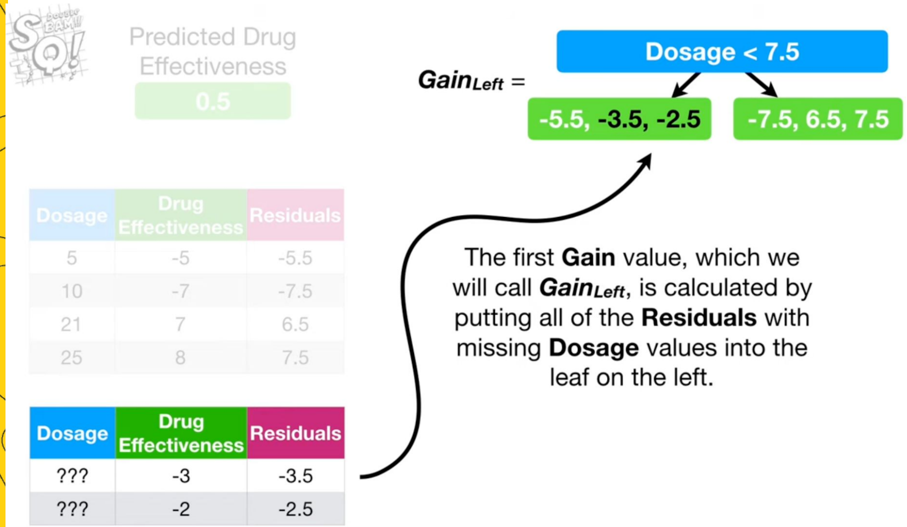
1. Catboost by Yandex
   https://catboost.ai/
   a. Explained by core developer for girafe-ai slides
2. LightGBM by Microsoft
   https://lightgbm.readthedocs.io/en/latest/index.html
3. XGboost by community
   https://xgboost.readthedocs.io/en/stable/

Definitely not sklearn!

Boosting explained in verse!

1. [Boosting explained]
2. [XGBoost expained]

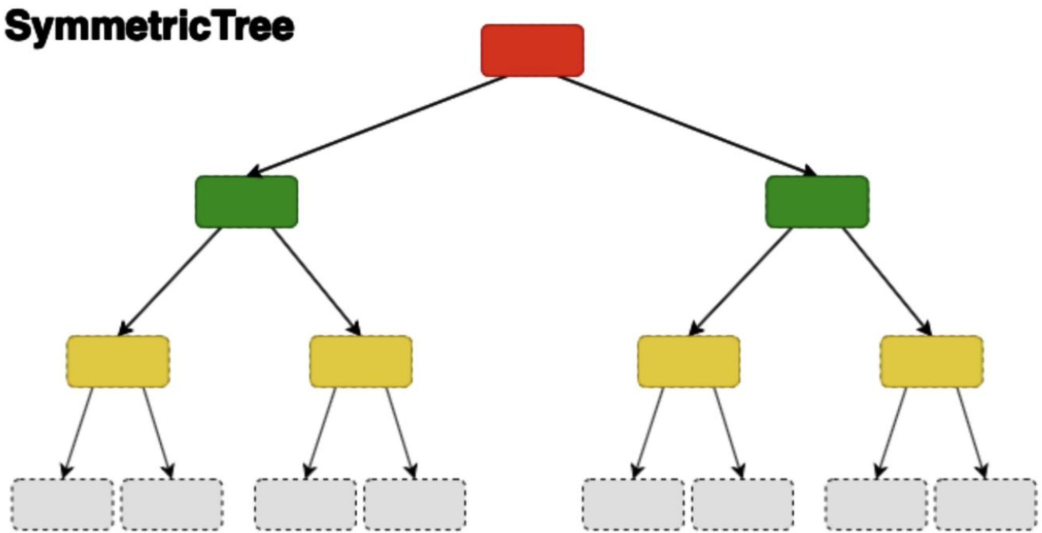




Dive to Catboost

- https://www.youtube.com/watch?v=s4GWmfB9VTA
- https://github.com/girafe-ai/journal-club/blob/master/slides/08%20CatBoosting.pdf

Форма дерева – CatBoost – SymmetricTree



More on boosting

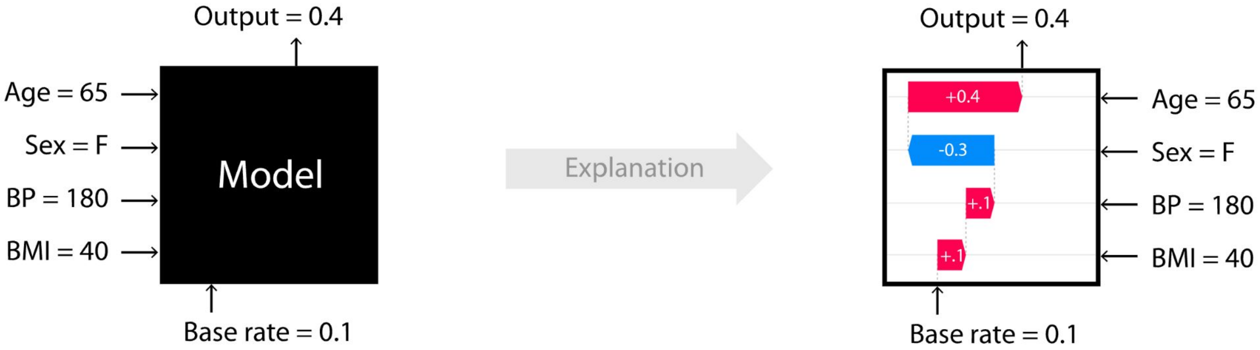
- https://habr.com/ru/companies/ods/articles/645887/
- https://neptune.ai/blog/when-to-choose-catboost-over-xgboost-or-lightgbm
- https://towardsdatascience.com/catboost-vs-lightgbm-vs-xgboost-c80f40662924
- https://www.springboard.com/blog/data-science/xgboost-random-forest-catboost-lightgbm/
- https://towardsdatascience.com/performance-comparison-catboost-vs-xgboost-and-catboost-vs-lightgbm-886c1c96db64

Feature importances

girafe
ai

05

Shap values



https://github.com/shap/shap

Shap values

$$\phi_i = \sum_{S \subseteq \{1,\ldots,p\}\{i\}} \underbrace{\frac{|S|!(p-|S|-1)!}{p!}}_{\text{Weight}} \underbrace{[val(S \cup \{i\}) - val(S)]}_{\text{Marginal contribution of player i to coalition S}}$$

p! = number of ways to form a coalition of p players

|S| = number of players in coalition S

|S|! = number of ways coalition S can form

(p-|S|-1)! = number of ways players can join after player i joins

https://youtu.be/UJeu29wq7d0

Shap values calculation

Coalition values

$C_{12} = 10{,}000$
$C_{1} = 7{,}500$
$C_{2} = 5{,}000$
$C_{0} = 0$

Marginal contribution

The increase in a coalition's value due to a player joining that coalition

$C_{12} - C_{2} = 5{,}000$
$C_{1} - C_{0} = 7{,}500$

$(5{,}000+7{,}500)/2$
$=$ **\$6,250**

$C_{12} - C_{1} = 2{,}500$
$C_{2} - C_{0} = 5{,}000$

$(2500+5000)/2$
$=$ **\$3,750**

Shap values calculation

$$C_{123} = 10,000$$
$$C_{0} = 0$$

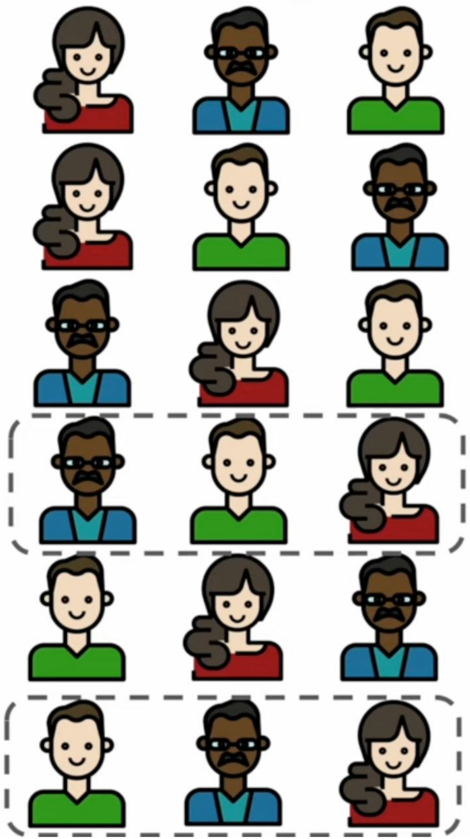
$$C_{12} = 7,500$$
$$C_{13} = 7,500$$
$$C_{23} = 5,000$$

$$C_{1} = 5,000$$
$$C_{2} = 5,000$$
$$C_{3} = 0$$

$$C_{123} - C_{23} = 5,000$$
$$C_{12} - C_{2} = 2,500$$
$$C_{13} - C_{3} = 7,500$$
$$C_{1} - C_{0} = 5,000$$

$$5,000^*(\tfrac{1}{3}) + 2,500^*(\tfrac{1}{6}) + 7,500^*(\tfrac{1}{6}) + 5,000^*(\tfrac{1}{3})$$
$$= \mathbf{\$5,000}$$

Shap values calculation



$C_{123} - C_{23} = 5{,}000$

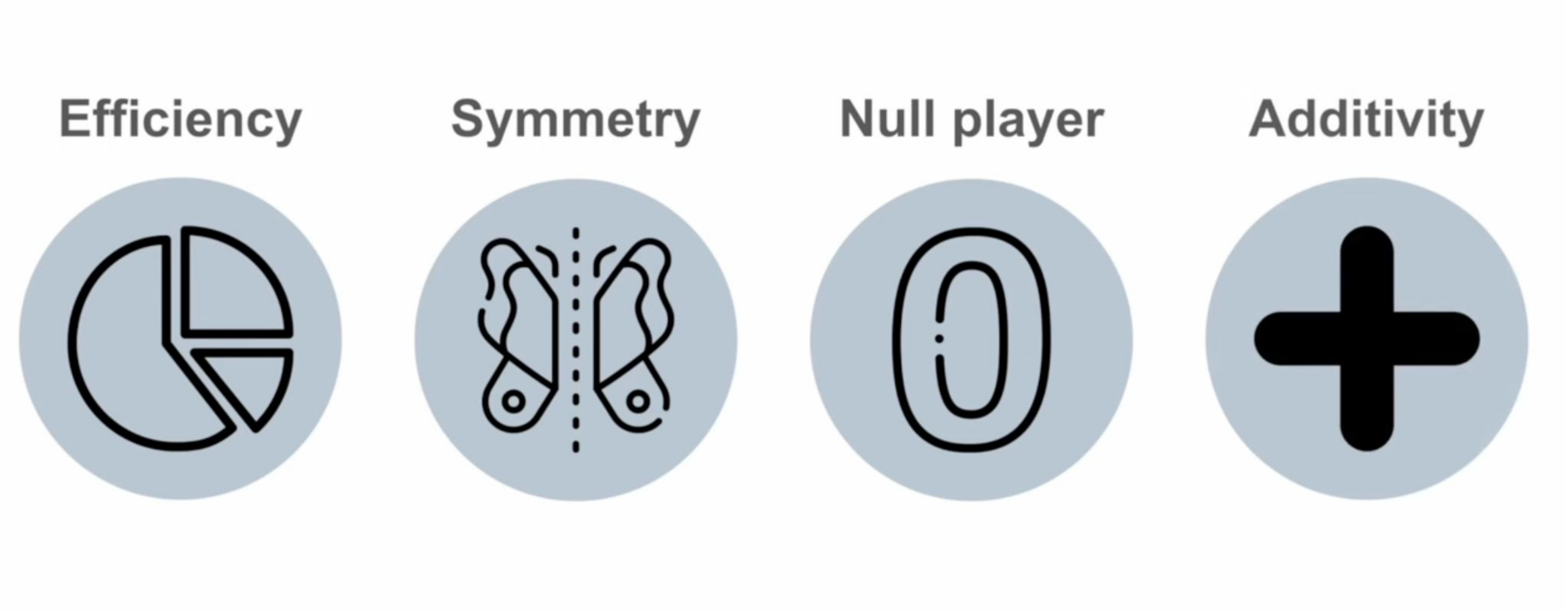
$P(C_{123} - C_{23})$ = probability that player 1 makes a marginal contribution to a coalition of player 2 and 3

$3! = 6$

$2! * 1! = 2$

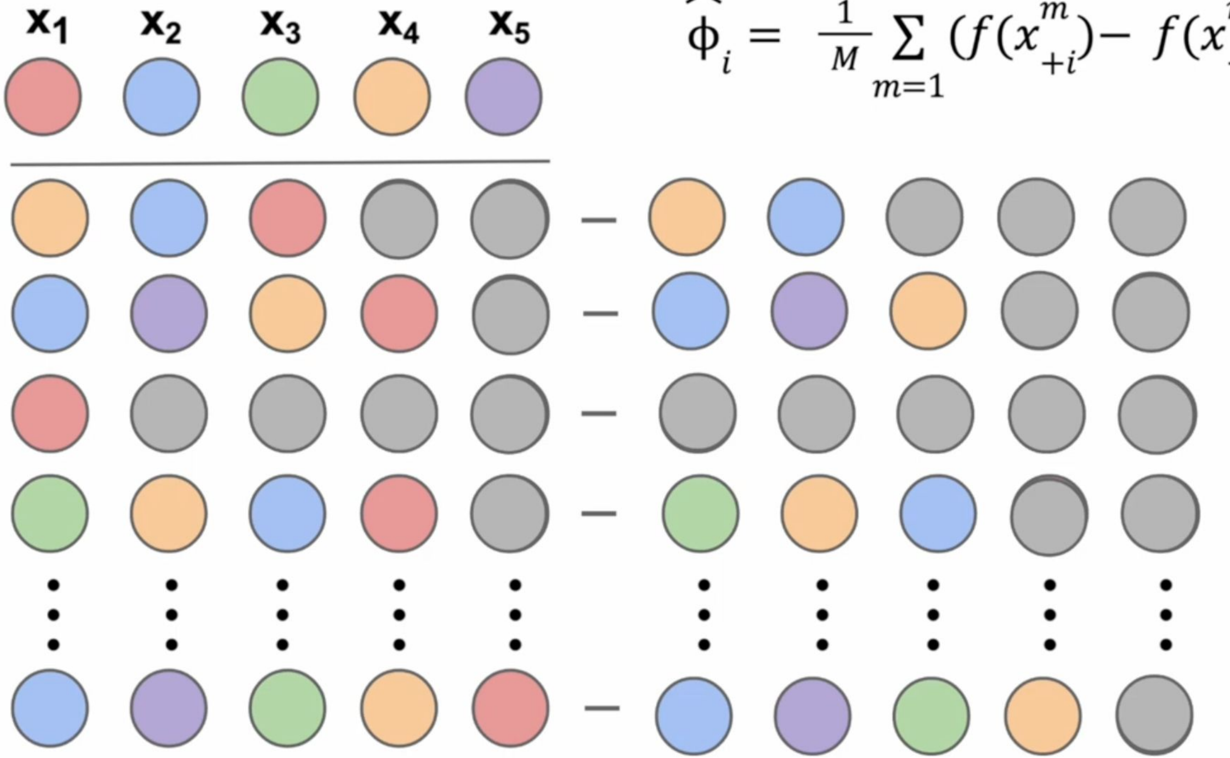
$P(C_{123} - C_{23}) = 2/6 = 1/3$

Shap values axioms

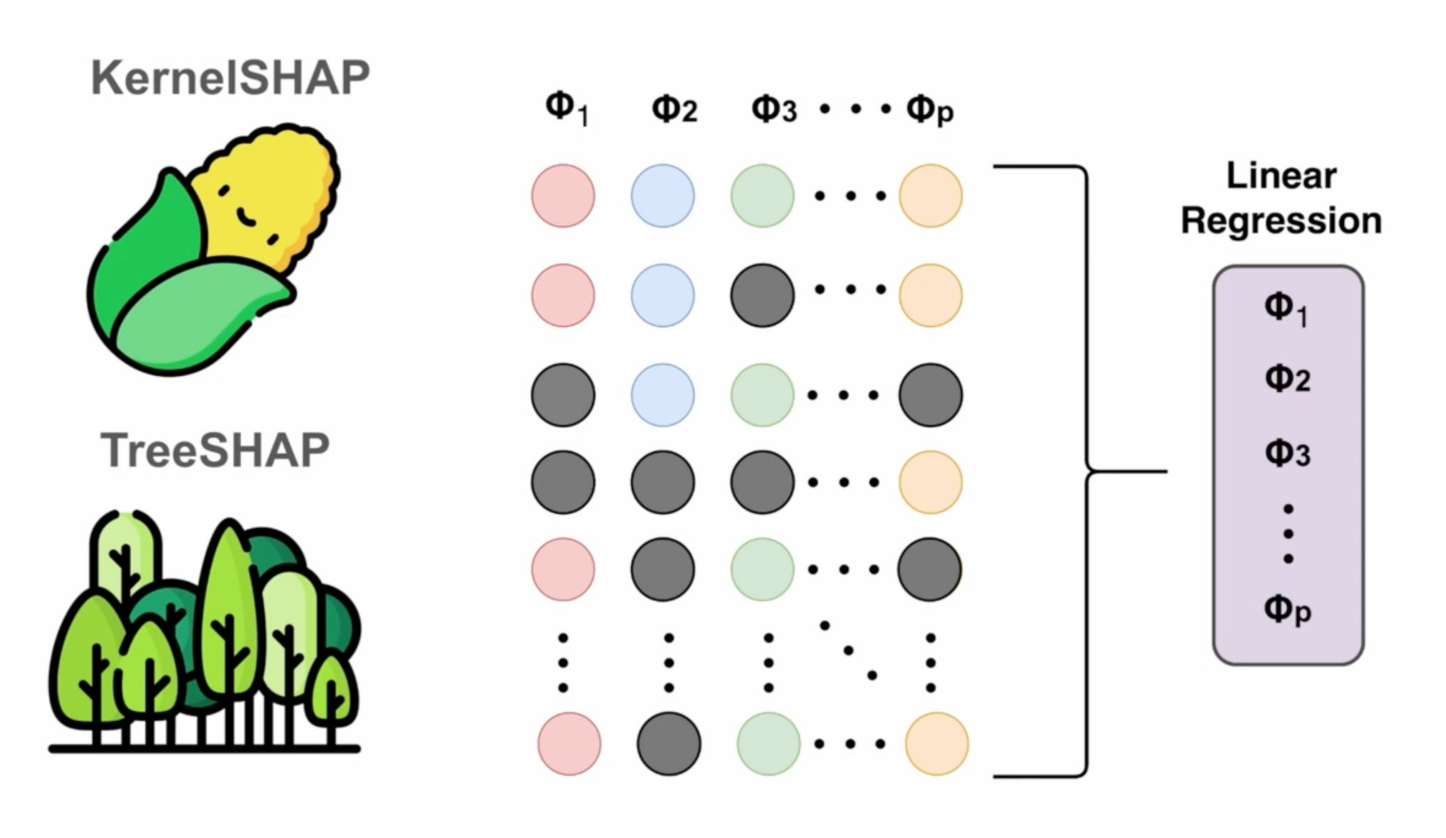


Shap values for ML

$$\widehat{\phi}_i = \frac{1}{M}\sum_{m=1}^{M}\left(f(x_{+i}^{m}) - f(x_{-i}^{m})\right)$$



Shap values for ML



Hyperparameter optimization

girafe
ai

06

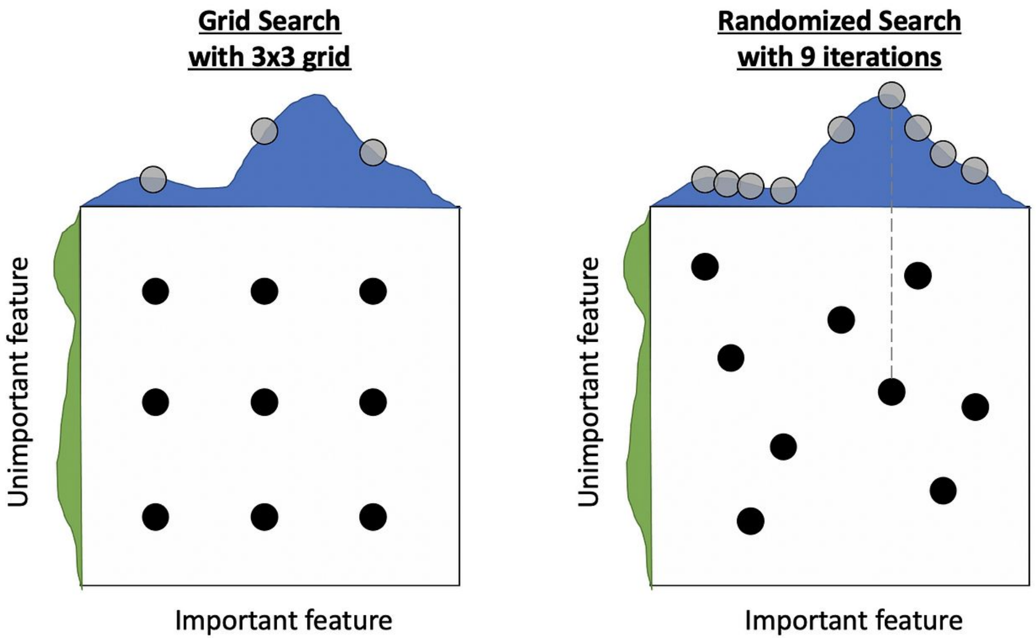
Optimization note

In optimization theory methods are associated with the order of derivatives they use. Main ones are:

- first order optimization (gradient based optimization)
  - use gradient of optimized function
  - e.g. SGD which we discussed
- second order optimization (Newton's method)
  - use Hessian matrix
  - they are quite slow
- zero order (black box optimization)
  - don't need gradient, only values of optimized function
  - that's what we are interested in today

0-order optimization approaches

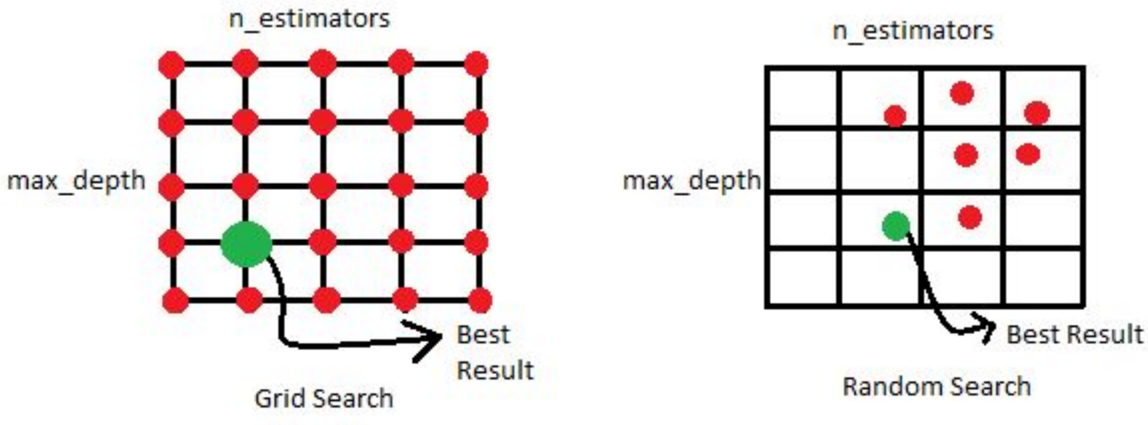
1. Manual trials
2. Grid search
3. Random search



In theory

0-order optimization approaches

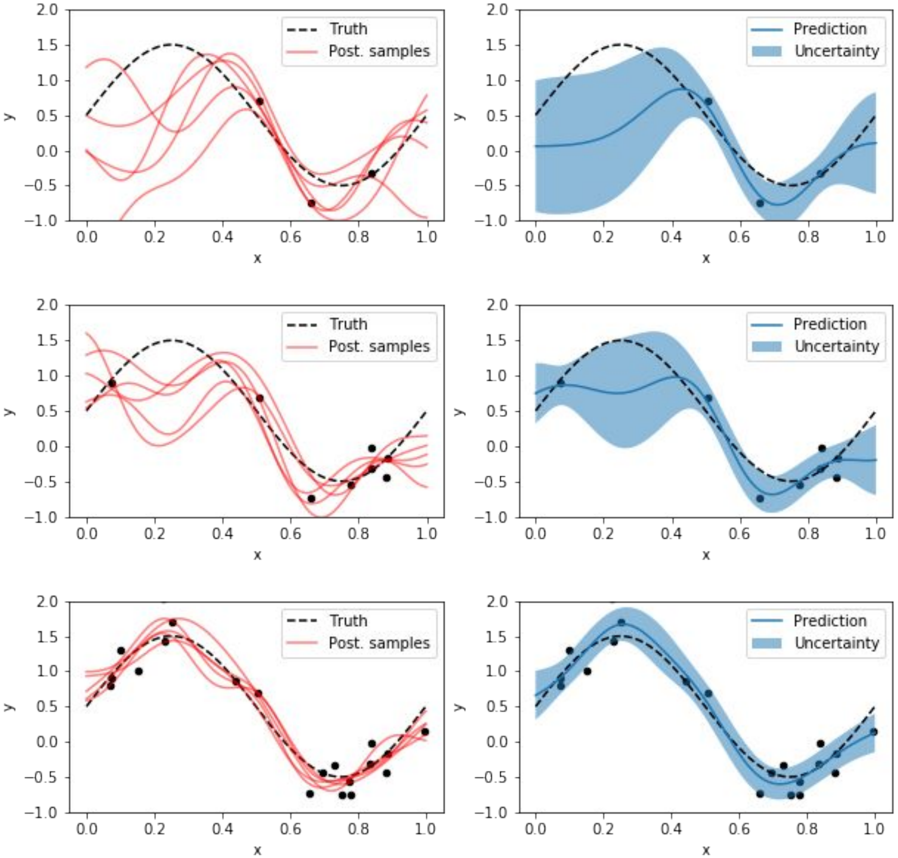
1. Manual trials
2. Grid search
3. Random search



In practice

0-order optimization approaches

1. Manual trials
2. Grid search
3. Random search
4. Bayesian methods
5. Evolutionary methods



Main libraries

- [Hyperopt]()
- [Optuna]()

Revise

1. Intuitions
2. Gradient boosting theory
3. Examples
4. Libraries
5. Feature importances
6. Hyperparameter optimization

Thanks for attention!

Questions?

girafe
ai