Machine Learning Lecture 8: Knowledge distillation

Young && Yandex ШАД

girafe ai

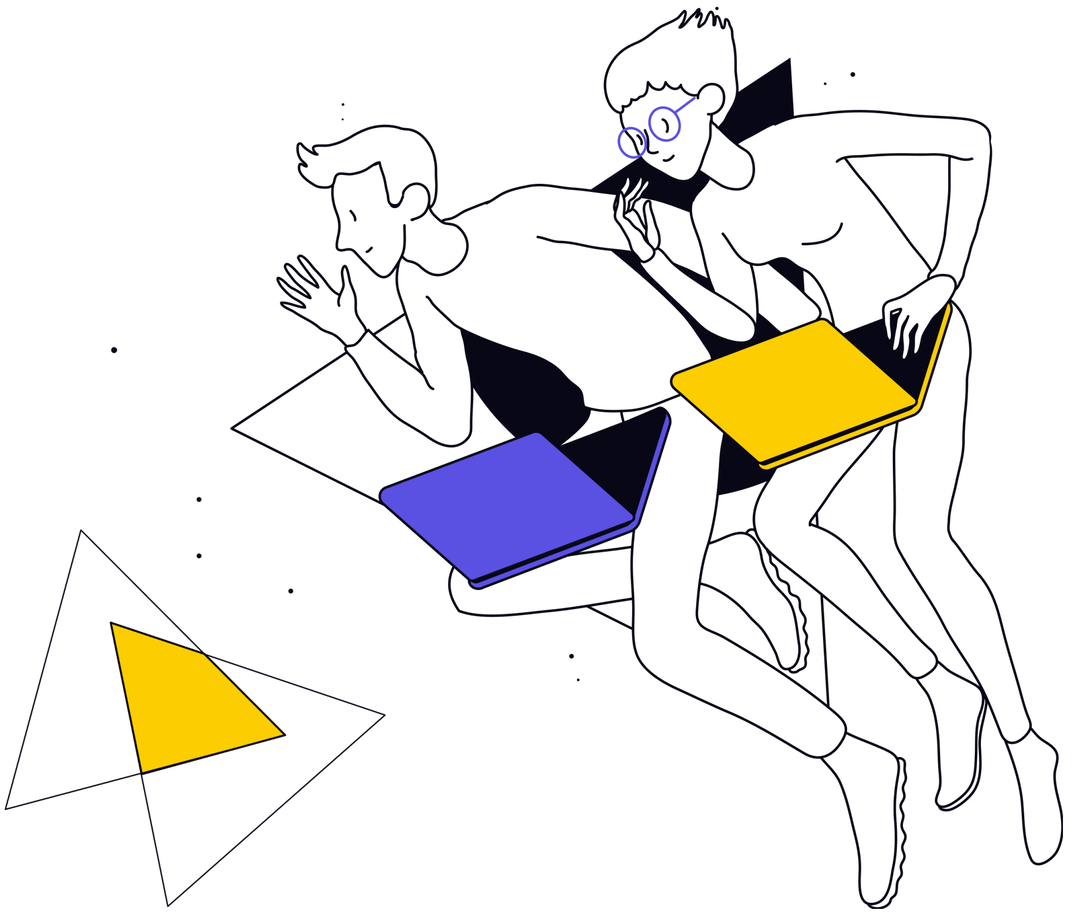
Radoslav Neychev

Outline

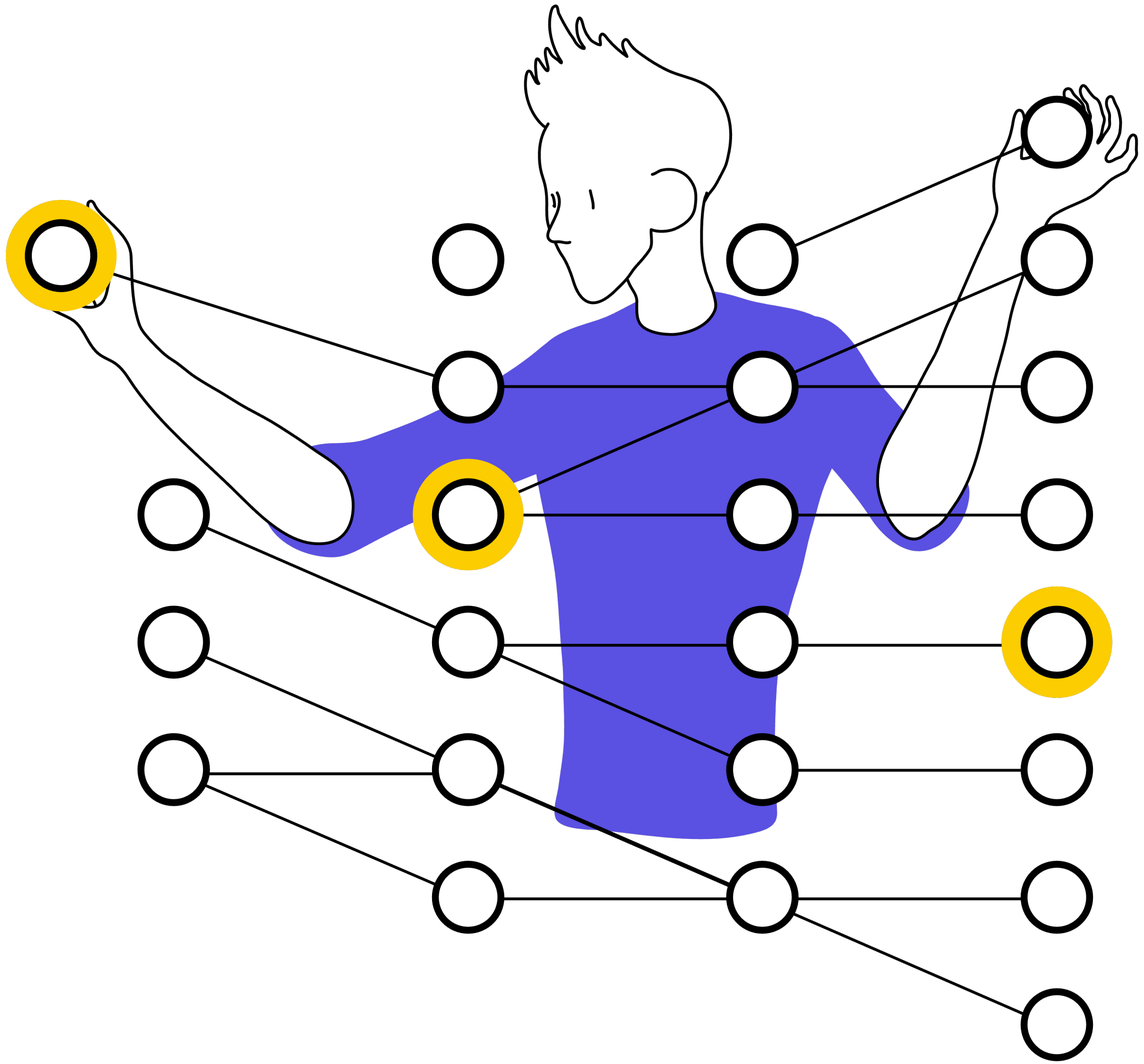
01 Knowledge distillation

02 General overview of supervised learning models

03 Outro



Knowledge distillation



01

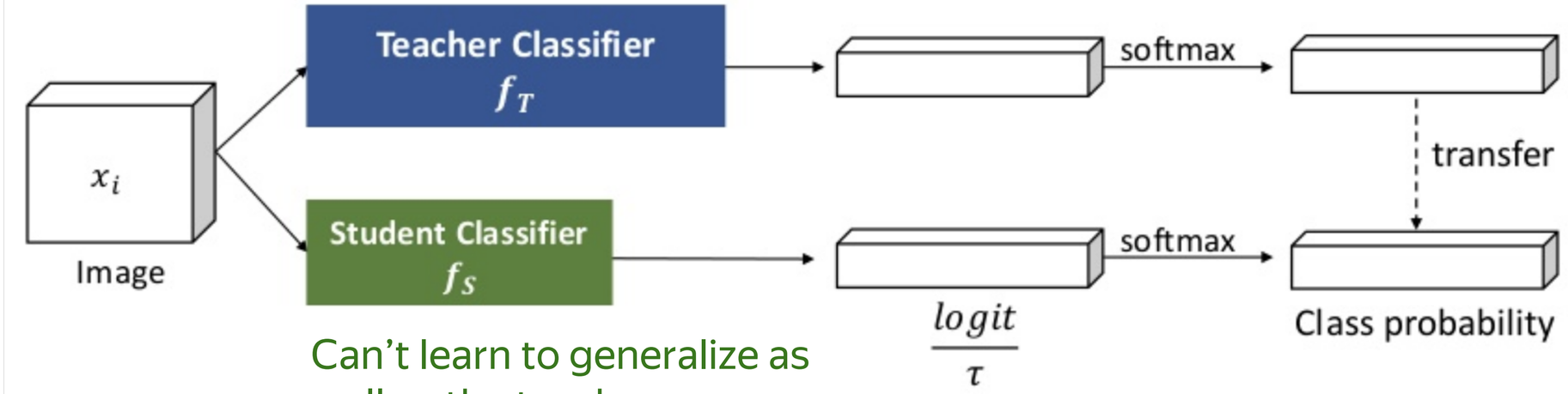
Knowledge distillation

Do they have the same “life purpose”
and solve the same problems?

Image credits: commons.wikimedia.org (left), Ben Sale/Flickr.com (right), copied from
https://nat-geo.ru/nature/insects/bylo-stalo-kak-vyglyadyat-babochki-i-ikh-gusenitsy/

Knowledge distillation

Learns to generalize and perform well



Can’t learn to generalize as well as the teacher, but still can perform well

Knowledge distillation

Denote **teacher** and **student** models.

Student model has logits z_i and corresponding probabilities q_i , derived with the softmax operation:

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

where *T* stays for the temperature.

Teacher model has logits v_i and corresponding probabilities p_i.

Knowledge distillation

Let’s derive the cross-entropy gradient on **student** logits using the **teacher** predictions as targets:

$$\frac{\partial C}{\partial z_i} = \frac{1}{T}(q_i - p_i) = \frac{1}{T}\left(\frac{e^{z_i/T}}{\sum_j e^{z_j/T}} - \frac{e^{v_i/T}}{\sum_j e^{v_j/T}}\right)$$

If the temperature is high, the following equation takes place:

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{T}\left(\frac{1 + z_i/T}{N + \sum_j z_j/T} - \frac{1 + v_i/T}{N + \sum_j v_j/T}\right)$$

Knowledge distillation

Logits can be centered, so

$$\sum_j z_j = \sum_j v_j = 0$$

Then the gradient takes form:

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{T}\left(\frac{1 + z_i/T}{N + \sum_j z_j/T} - \frac{1 + v_i/T}{N + \sum_j v_j/T}\right) \approx \frac{1}{NT^2}\left(z_i - v_i\right)$$

$$\frac{dC}{dz_i} = \frac{1}{NT^2}(z_i - v_i) \sim (z_i - v_i) = M\frac{d(z_i - v_i)^2}{dz_i}$$

Constant

Bias-Variance decomposition

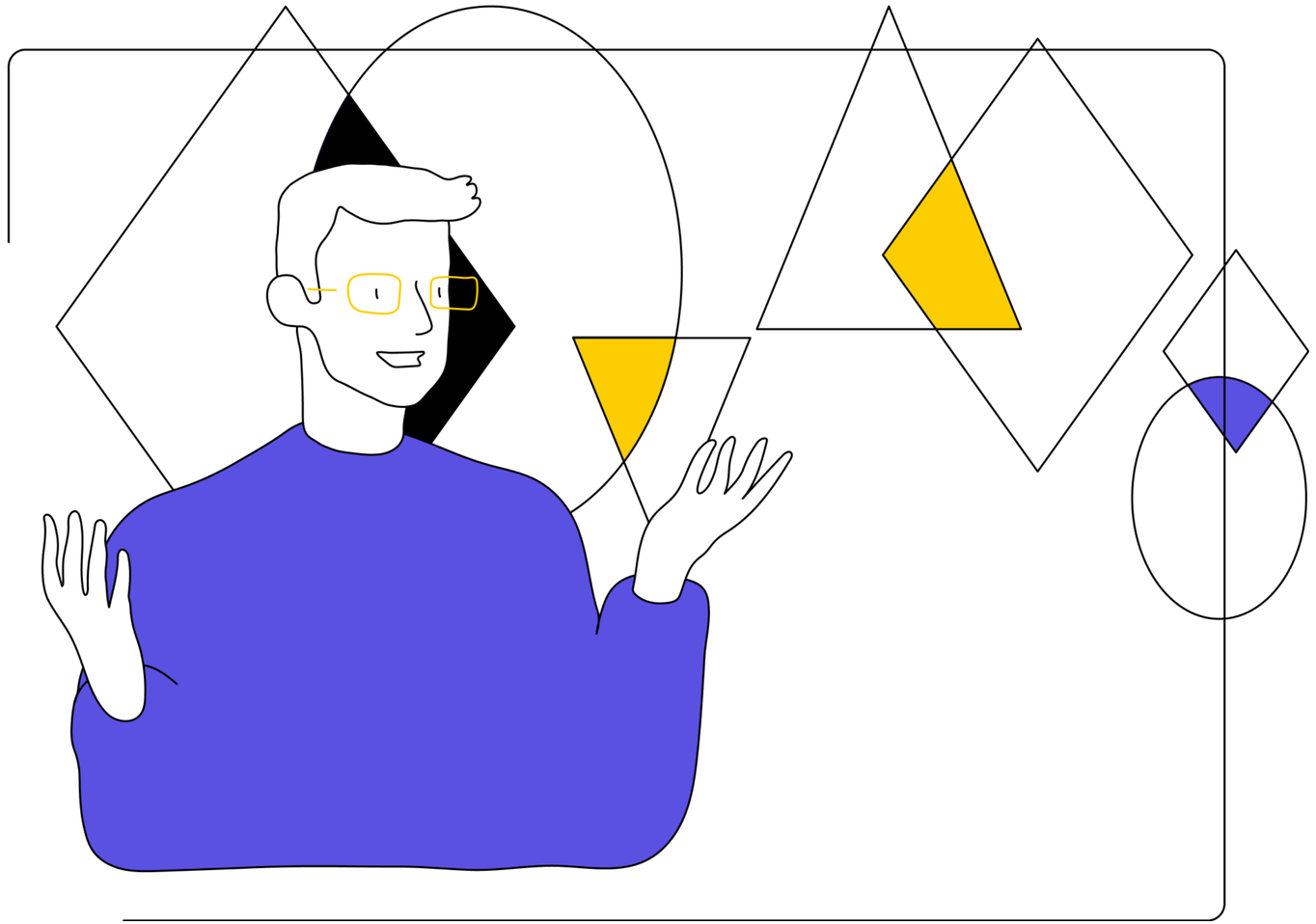
$$L(\mu) = \underbrace{\mathbb{E}_{x,y}\left[\left(y - \mathbb{E}[y \mid x]\right)^2\right]}_{\text{Noise}} +$$

$$+ \underbrace{\mathbb{E}_x\left[\left(\mathbb{E}_X\left[\mu(X)\right] - \mathbb{E}[y \mid x]\right)^2\right]}_{\text{Bias}} + \underbrace{\mathbb{E}_x\left[\mathbb{E}_X\left[\left(\mu(X) - \mathbb{E}_X\left[\mu(X)\right]\right)^2\right]\right]}_{\text{Variance}}$$

This exact form of bias-variance decomposition is correct for square loss in regression

However, it is much more general. See extra materials for more exotic cases

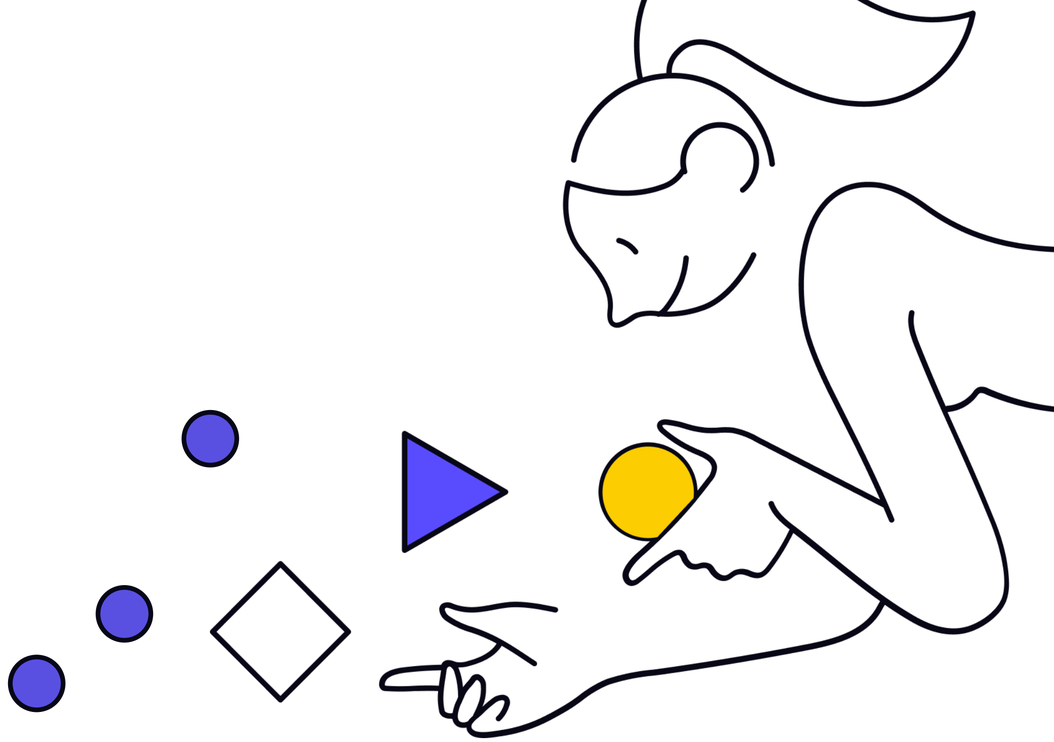
Feature importance estimation: naïve approach



02

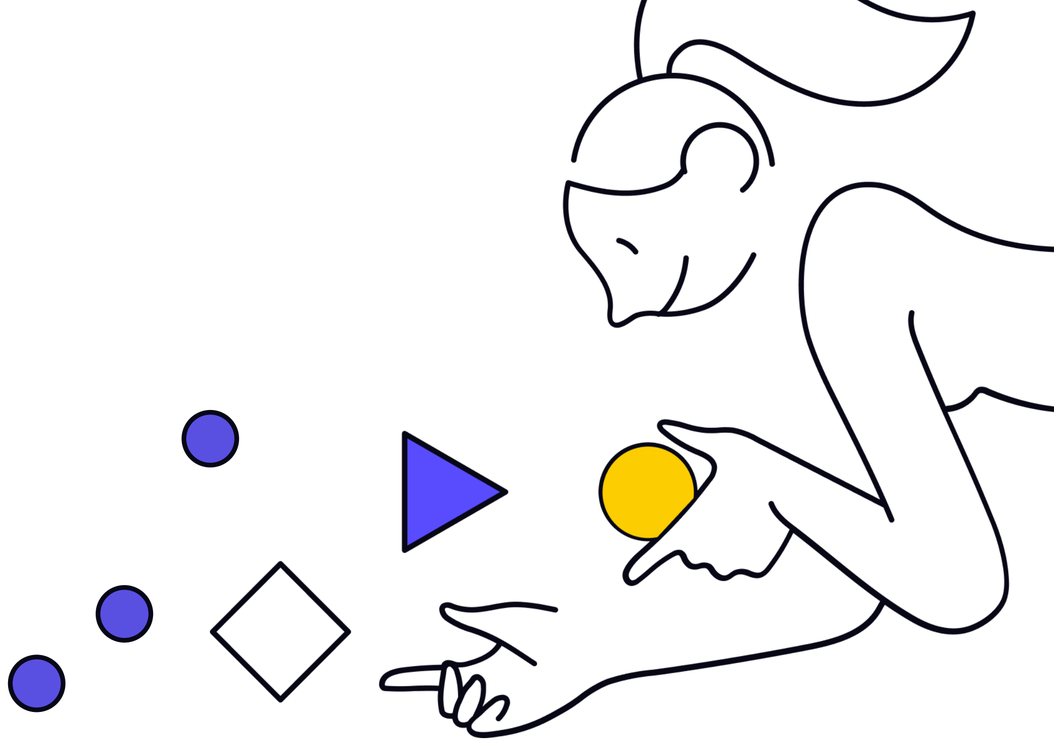
List of methods so far

1. kNN – non-parametric method



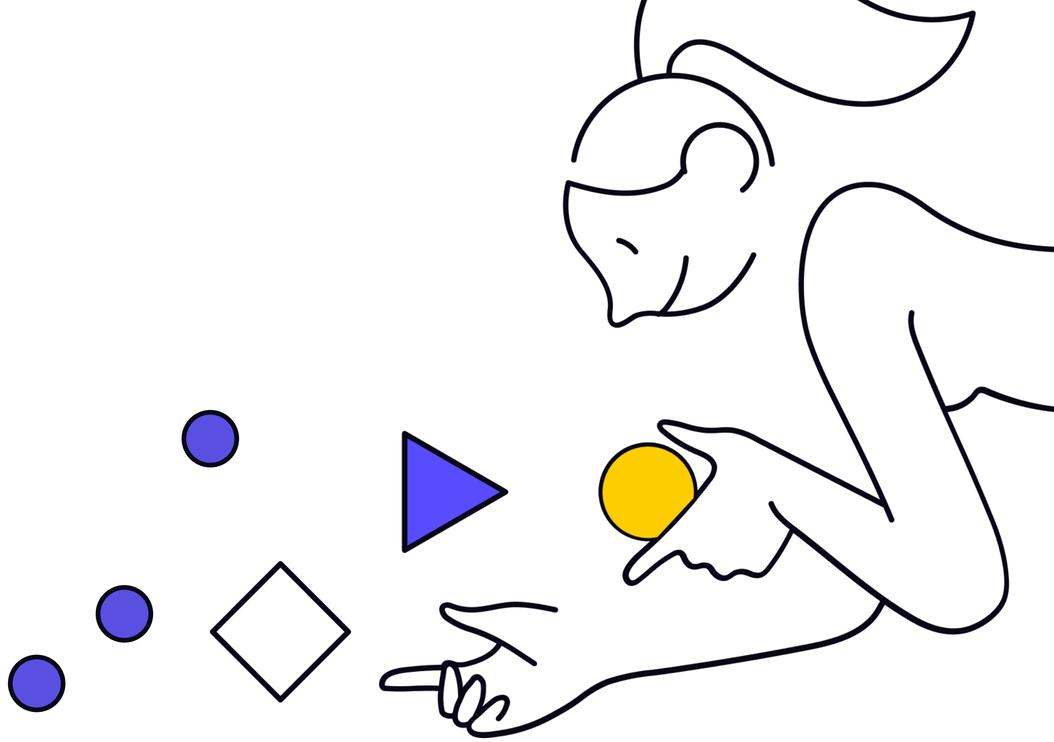
List of methods so far

1. kNN – non-parametric method
2. Linear (logistic) regression



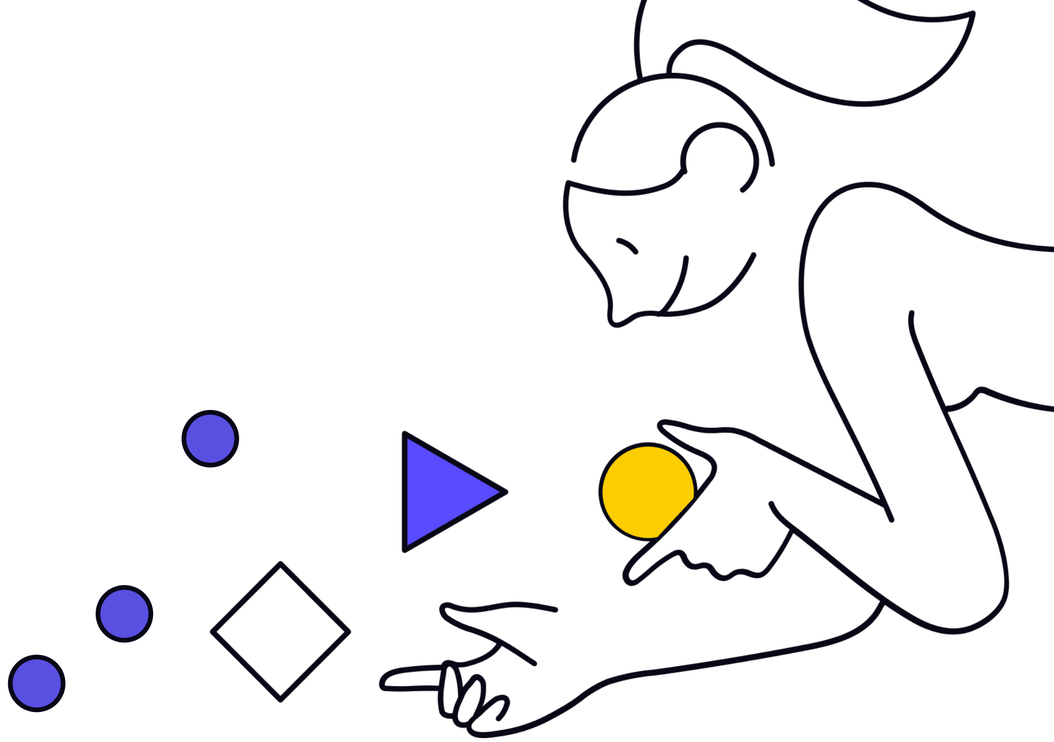
List of methods so far

1. kNN – non-parametric method
2. Linear (logistic) regression
3. Decision Trees



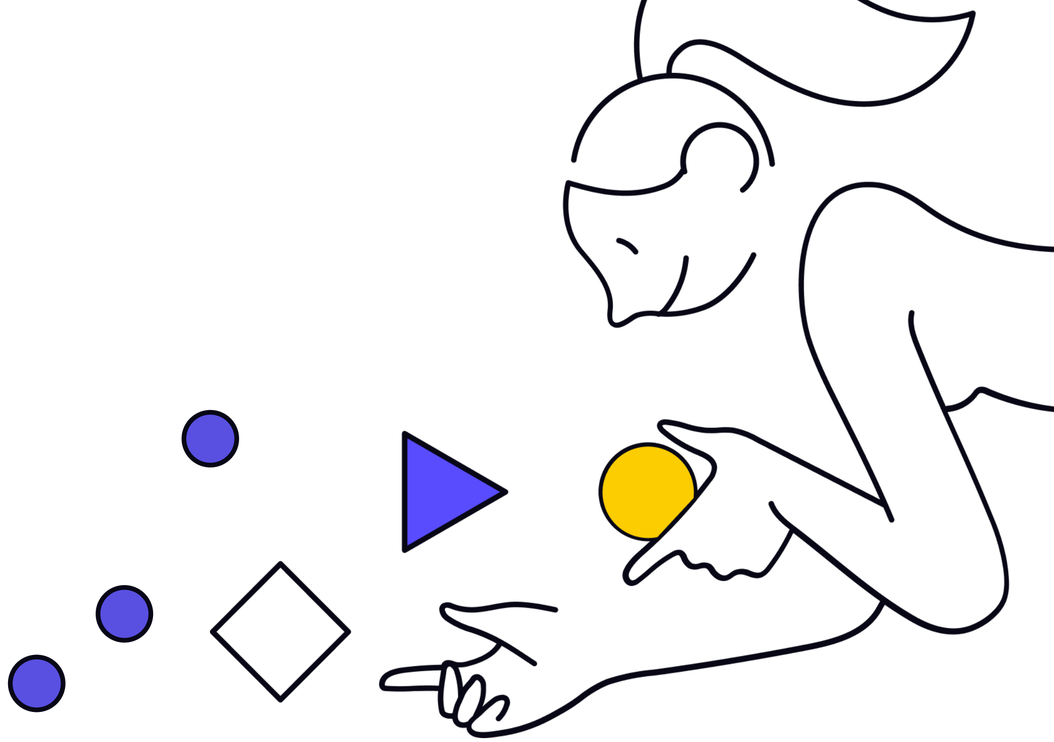
List of methods so far

1. kNN – non-parametric method
2. Linear (logistic) regression
3. Decision Trees
4. Bagging ensembles



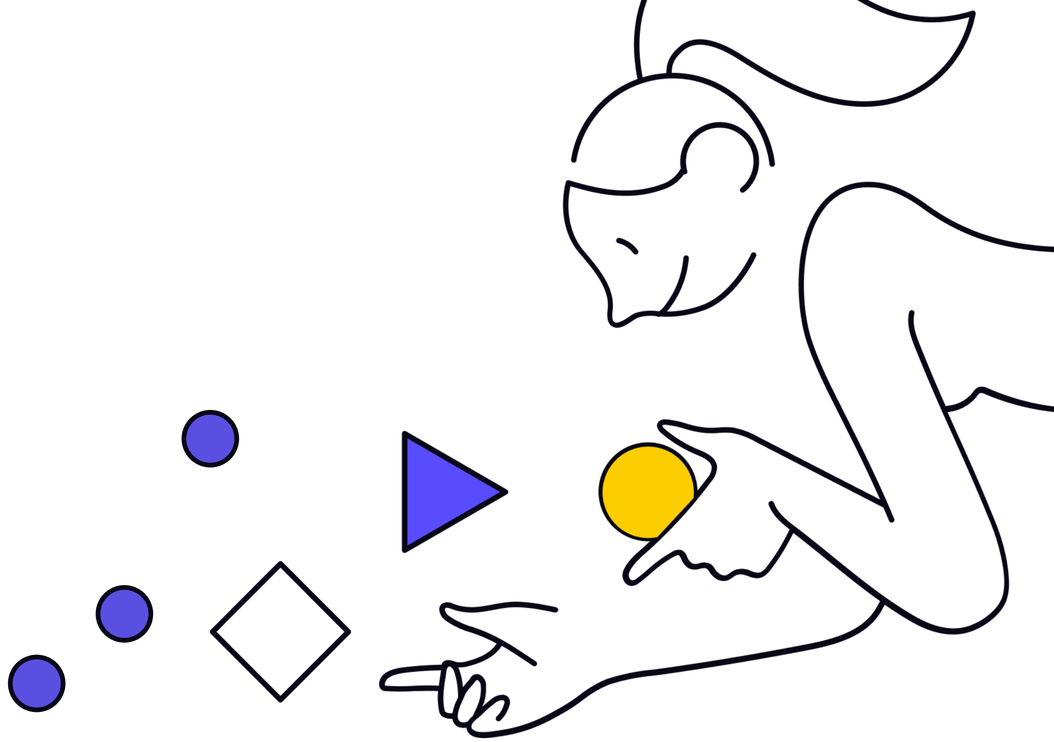
List of methods so far

1. kNN – non-parametric method
2. Linear (logistic) regression
3. Decision Trees
4. Bagging ensembles
5. Boosting ensembles



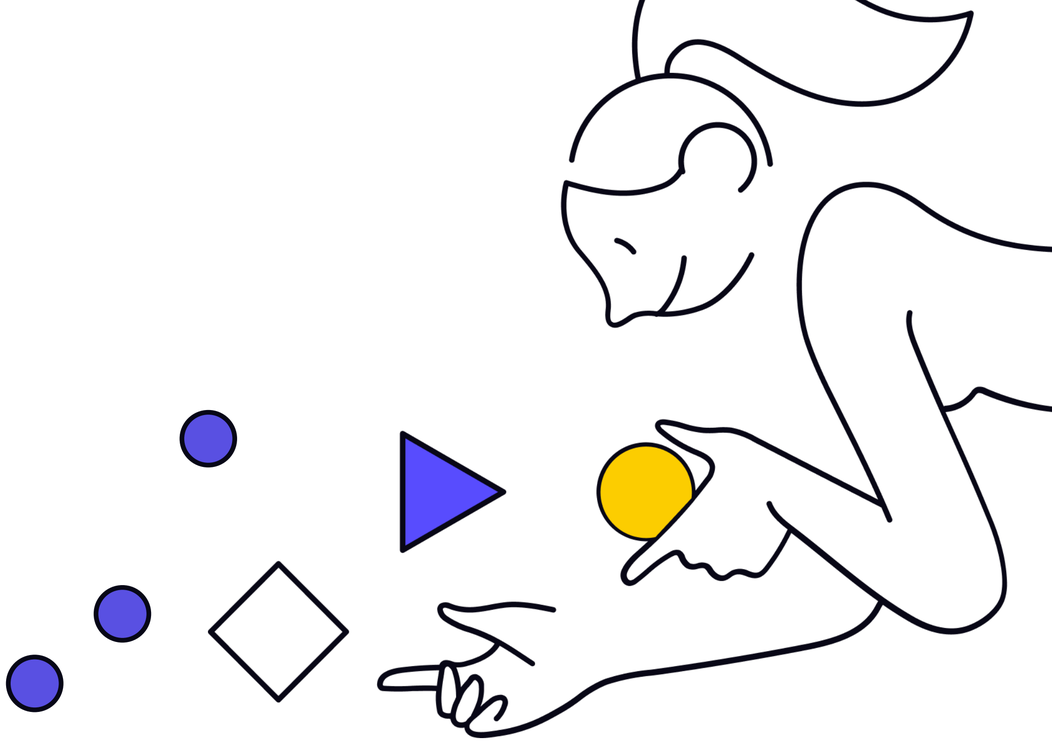
List of methods so far

1. kNN – non-parametric method
2. Linear (logistic) regression
3. Decision Trees
4. Bagging ensembles
5. Boosting ensembles
6. Stacking and blending



List of methods so far

1. kNN – non-parametric method
2. Linear (logistic) regression
3. Decision Trees
4. Bagging ensembles
5. Boosting ensembles
6. Stacking and blending
7. Neural Networks



Todays lecture recap

01 Bias-Variance decomposition

02 Feature importance estimation: naïve approach

03 LIME

04 Shapley values; SHAP

05 Neural networks predictions explanation

Thanks for attention!

Radoslav Neychev