Machine Learning Lecture 7: Feature importance estimation, predictions explanation

Young && Yandex ШАД

girafe ai

Radoslav Neychev

Outline

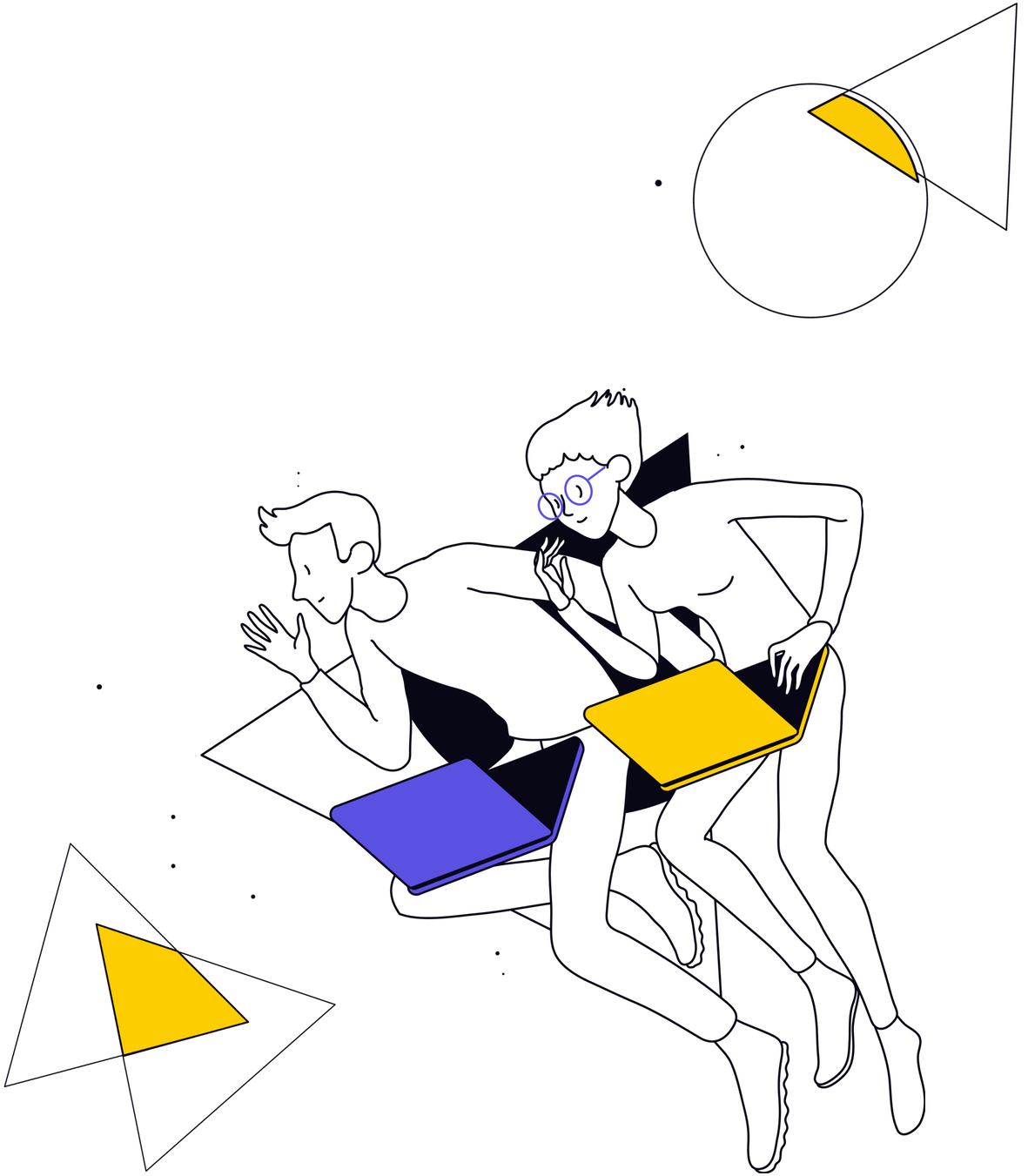
01 Bias-Variance decomposition

02 Feature importance estimation: naïve approach

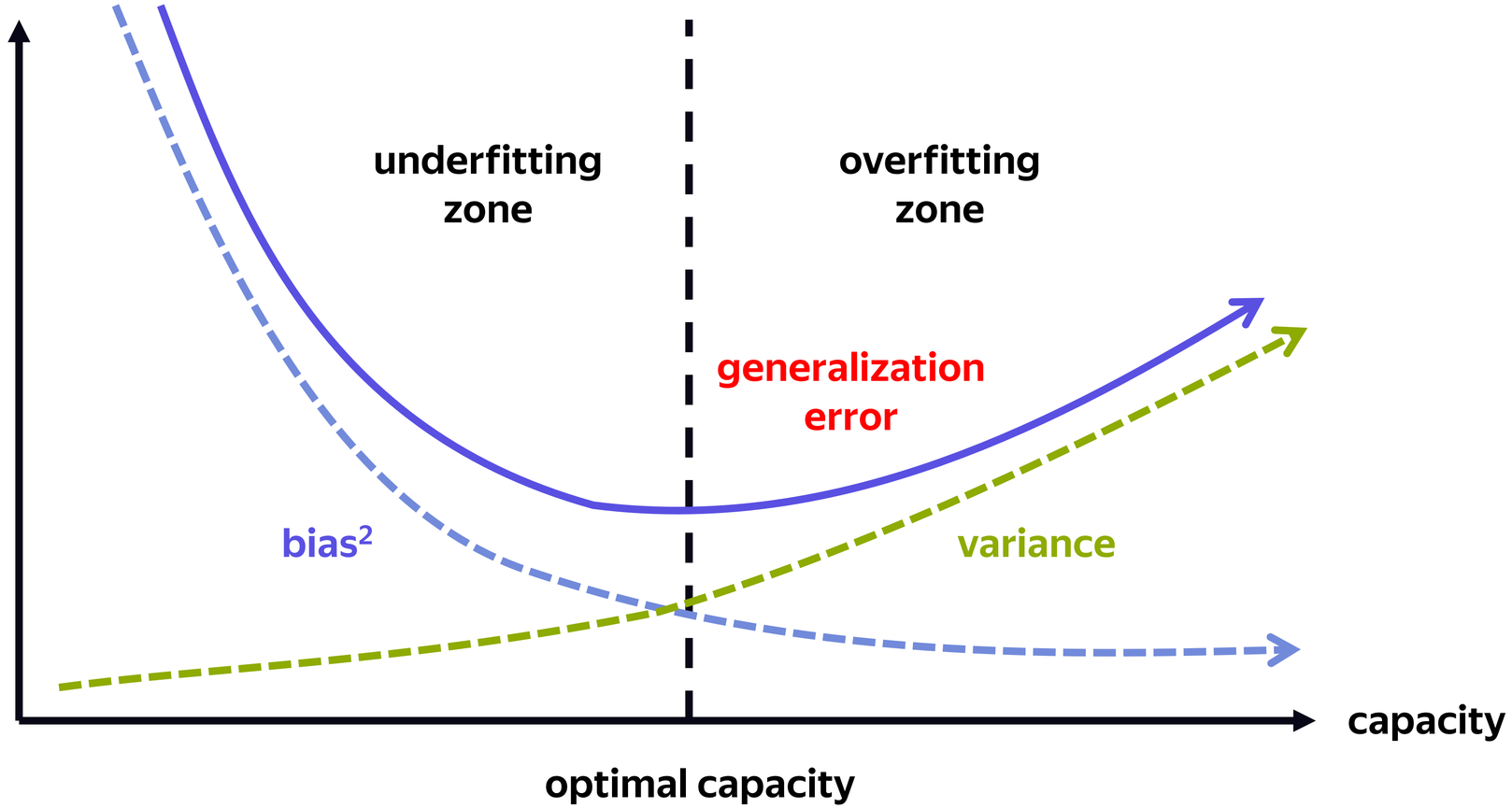
03 LIME

04 Shapley values; SHAP

05 Neural networks predictions explanation



Bias-Variance tradeoff



Bias-Variance decomposition

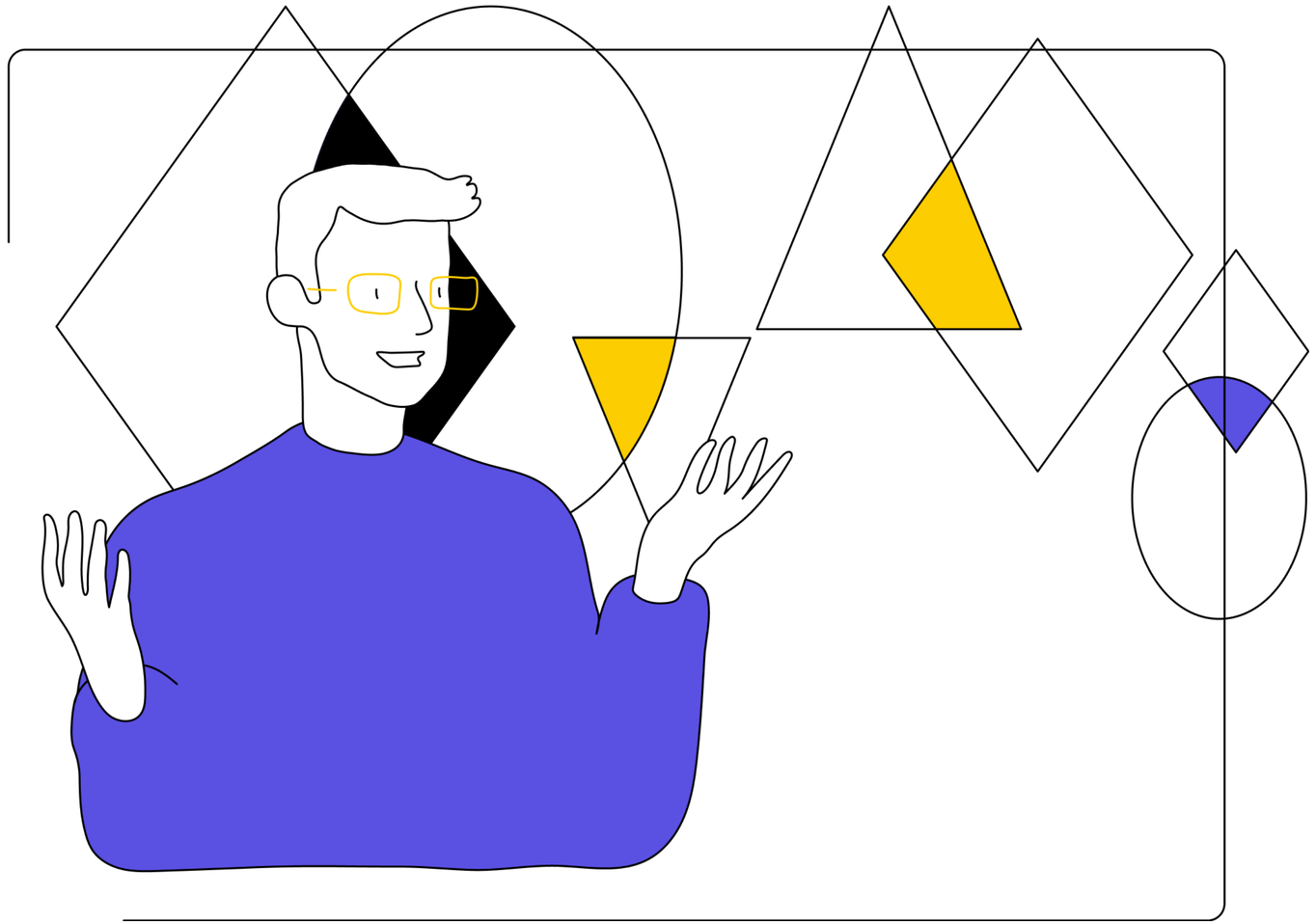
$$L(\mu) = \underbrace{\mathbb{E}_{x,y}\left[\left(y - \mathbb{E}[y \mid x]\right)^2\right]}_{\text{Noise}} +$$

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

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

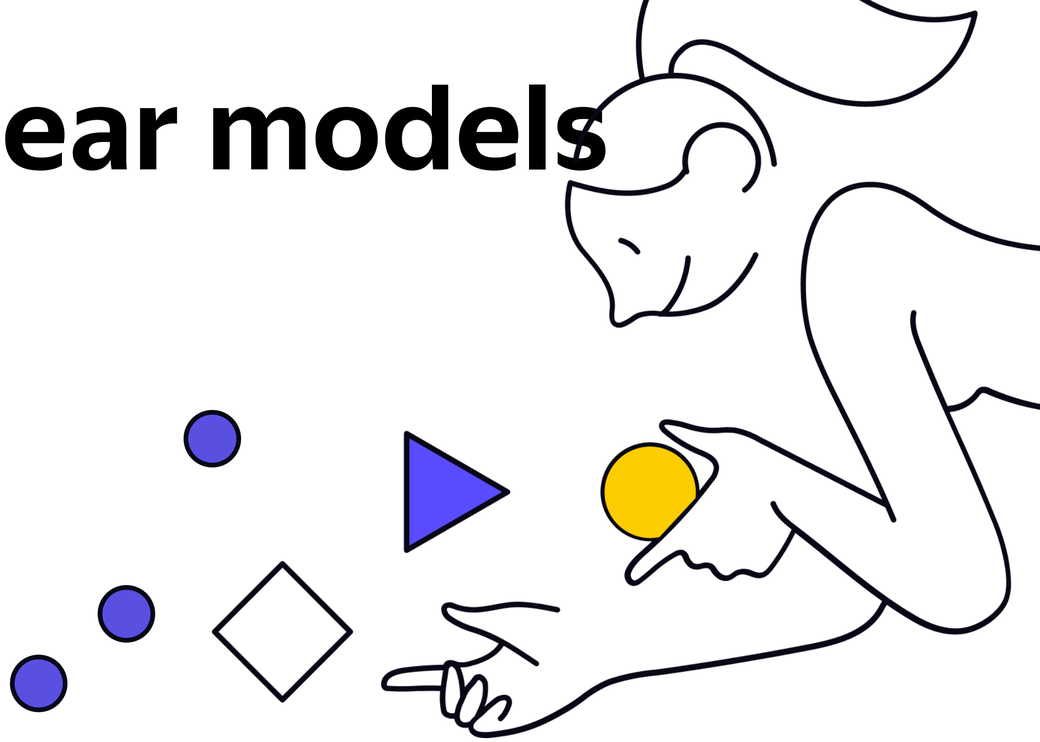
However, it is much more general. See extra materials for more exotic cases

Feature importance estimation: naïve approach



02

Importance estimation: linear models



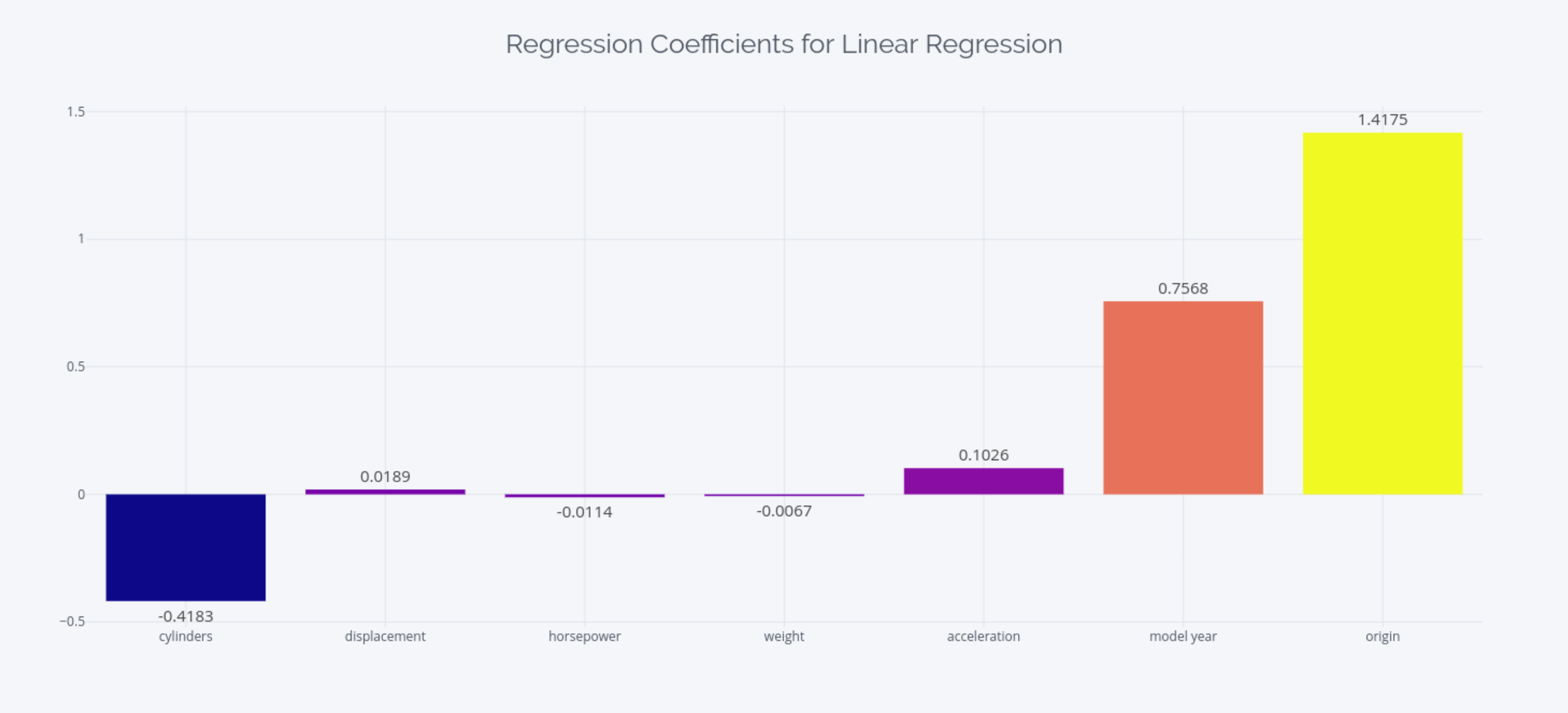
Denote linear model, trained on the provided data

$$\boldsymbol{y} = \sum_{i=1}^{p} w_i x_i + b = \boldsymbol{w}^\top \boldsymbol{x} + b$$

for regression problem

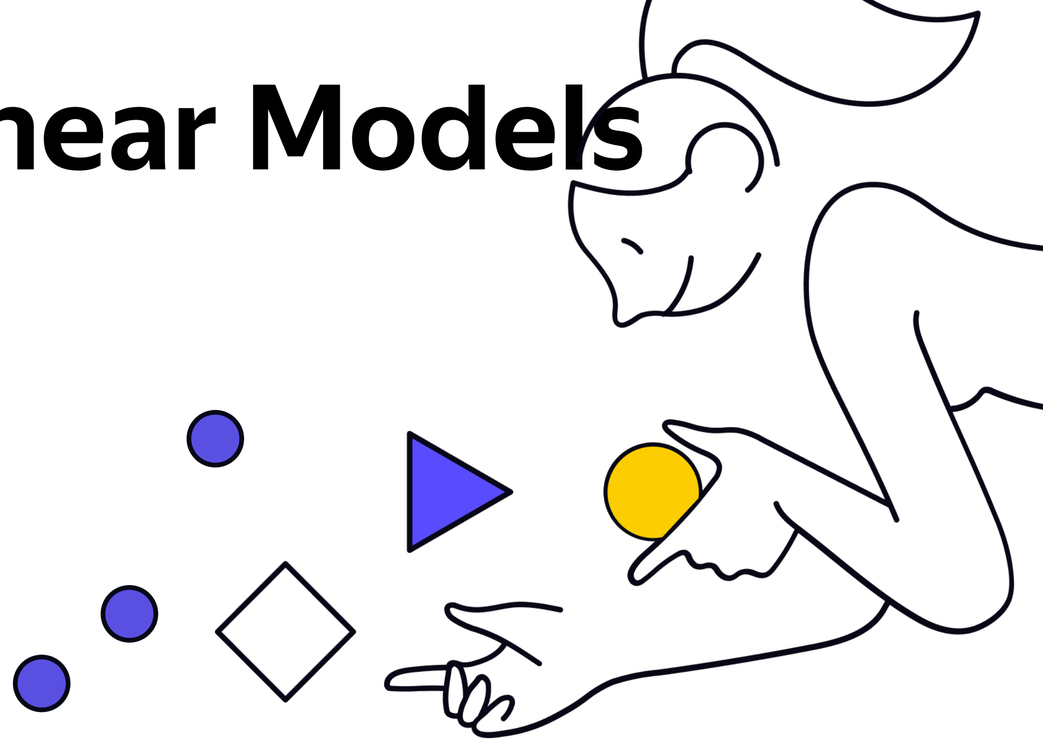
$$p = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$$

for binary classification problem



Source: Explainability: Cracking open the black box, Part 1, KDnuggets

Importance estimation: Linear Models



Denote linear model, trained on the provided data

$$\boldsymbol{y} = \sum_{i=1}^{p} w_i x_i + b = \boldsymbol{w}^\top \boldsymbol{x} + b$$

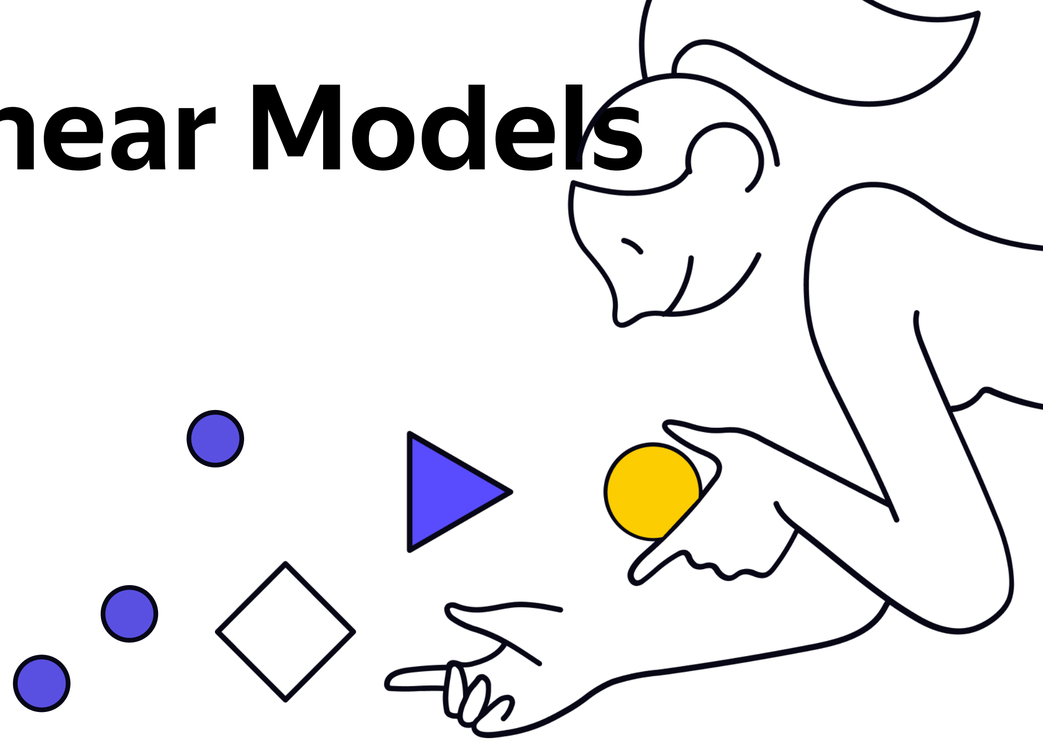
for regression problem

$$p = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$$

for binary classification problem

If features are **normalized**, w_i shows the importance of i-th feature for the defined model.

Importance estimation: Linear Models



Denote linear model, trained on the provided data

$$\boldsymbol{y} = \sum_{i=1}^{p} w_i x_i + b = \boldsymbol{w}^\top \boldsymbol{x} + b$$

for regression problem

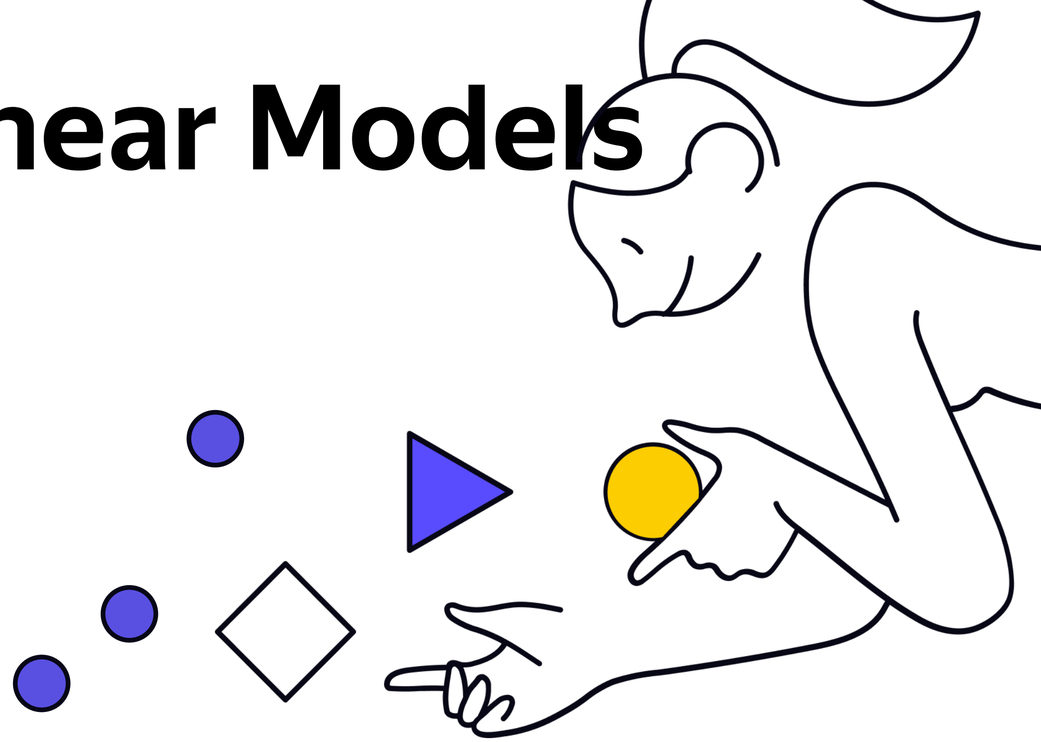
$$p = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$$

for binary classification problem

Standardized coefficient $w_i \cdot \dfrac{\mathrm{std}(x_i)}{\mathrm{std}(y)}$ shows how many std of target will change if i-th feature std changes.

Source: Explainability: Cracking open the black box, Part 1, KDnuggets

Importance estimation: Linear Models



Assume there is also a regularization:

$$Q(\boldsymbol{w}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_2^2$$

for L2 case (Ridge)

$$Q(\boldsymbol{w}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1$$

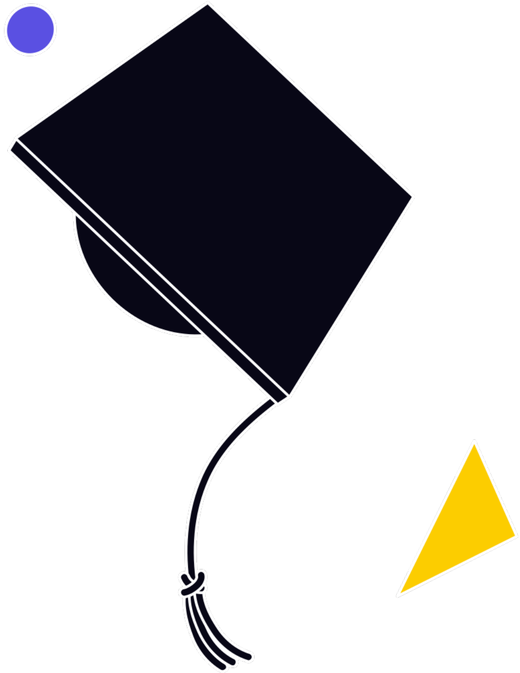
for L1 case (LASSO)

Rule of Thumb:

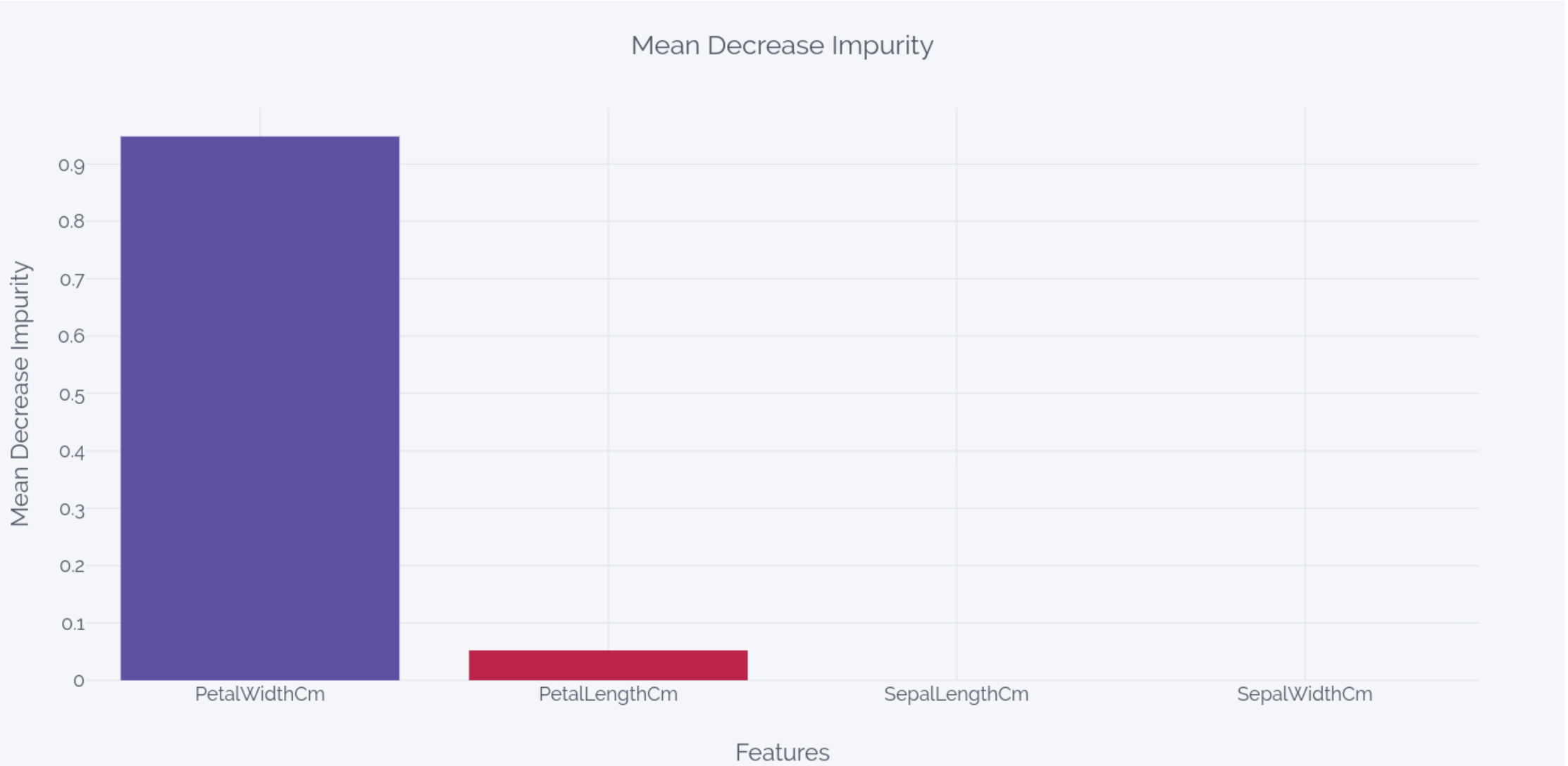
- Ridge – for multiple features slightly affecting the target
- LASSO – for few features affecting target a lot

Importance estimation: Decision Trees

Denote Fisher's Iris dataset

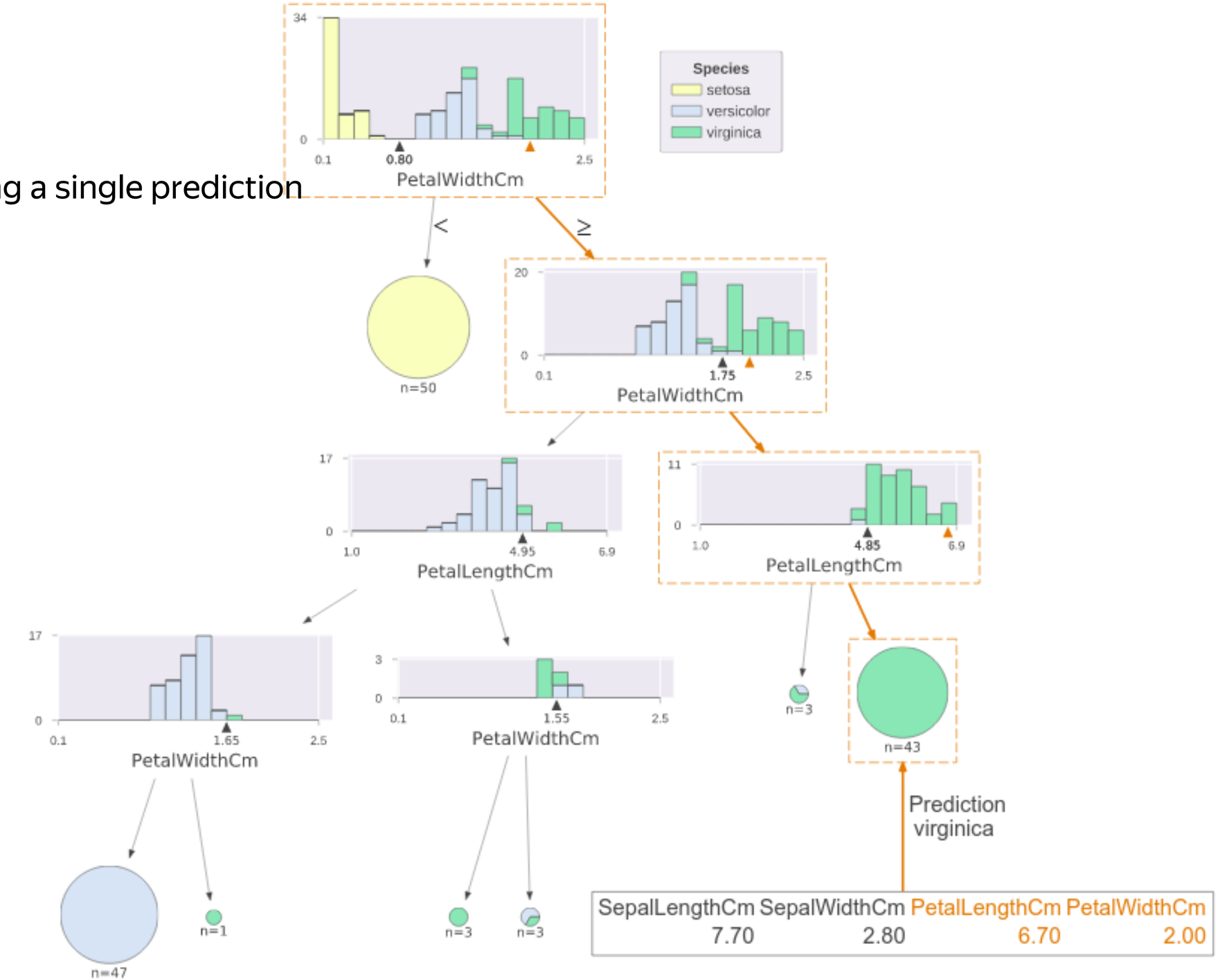


[Source]



Source: Explainability: Cracking open the black box, Part 1, KDnuggets

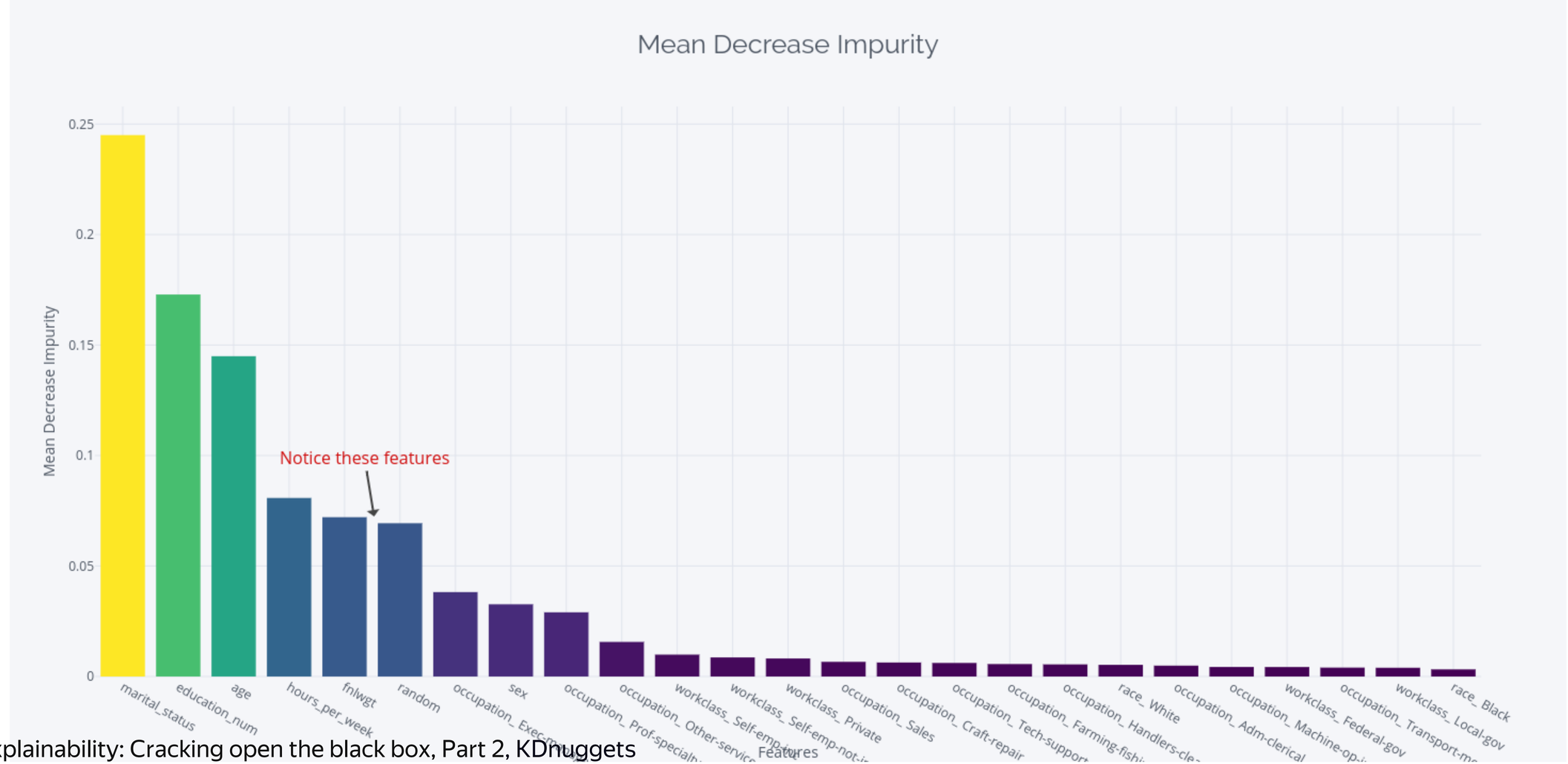
Explaining a single prediction



Source

Importance estimation: Trees (ensembles)

Information gain or Gini impurity/Entropy decrease

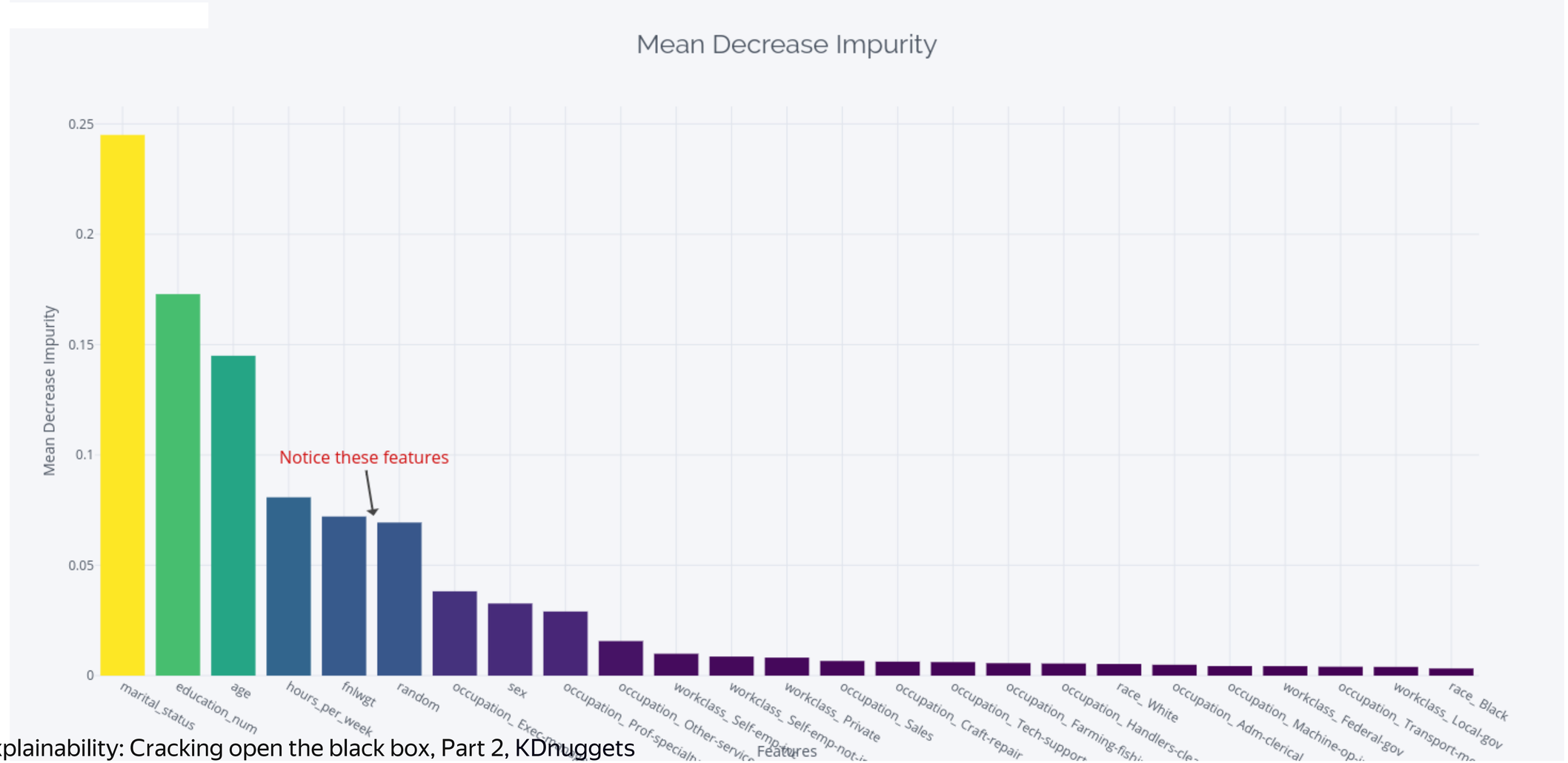


Source: Explainability: Cracking open the black box, Part 2, KDnuggets

Importance estimation: Trees (ensembles)

Information gain or Gini impurity/Entropy decrease – biased feature importance estimation method.

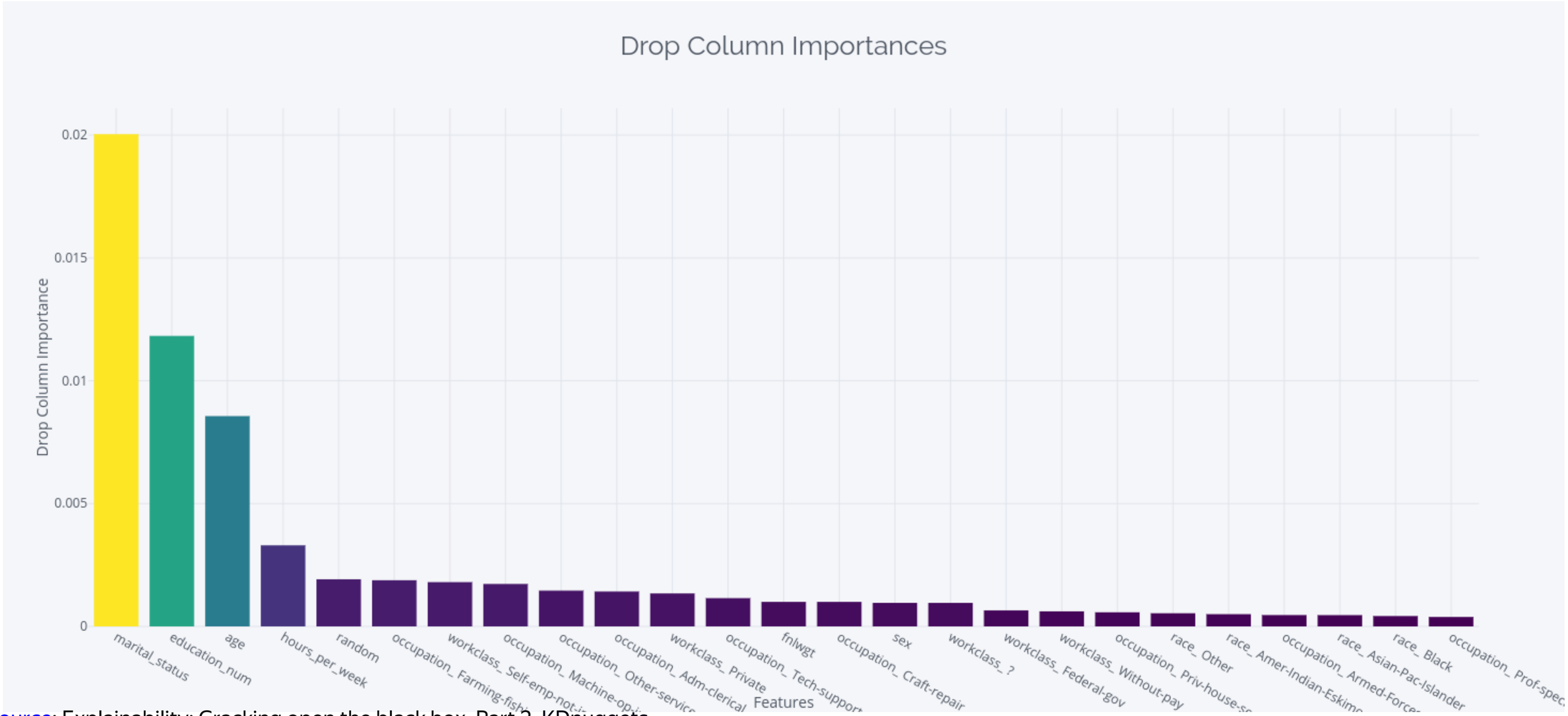
“It favours continuous features and features with high cardinality”.



Source: Explainability: Cracking open the black box, Part 2, KDnuggets

Importance estimation: model-agnostic methods

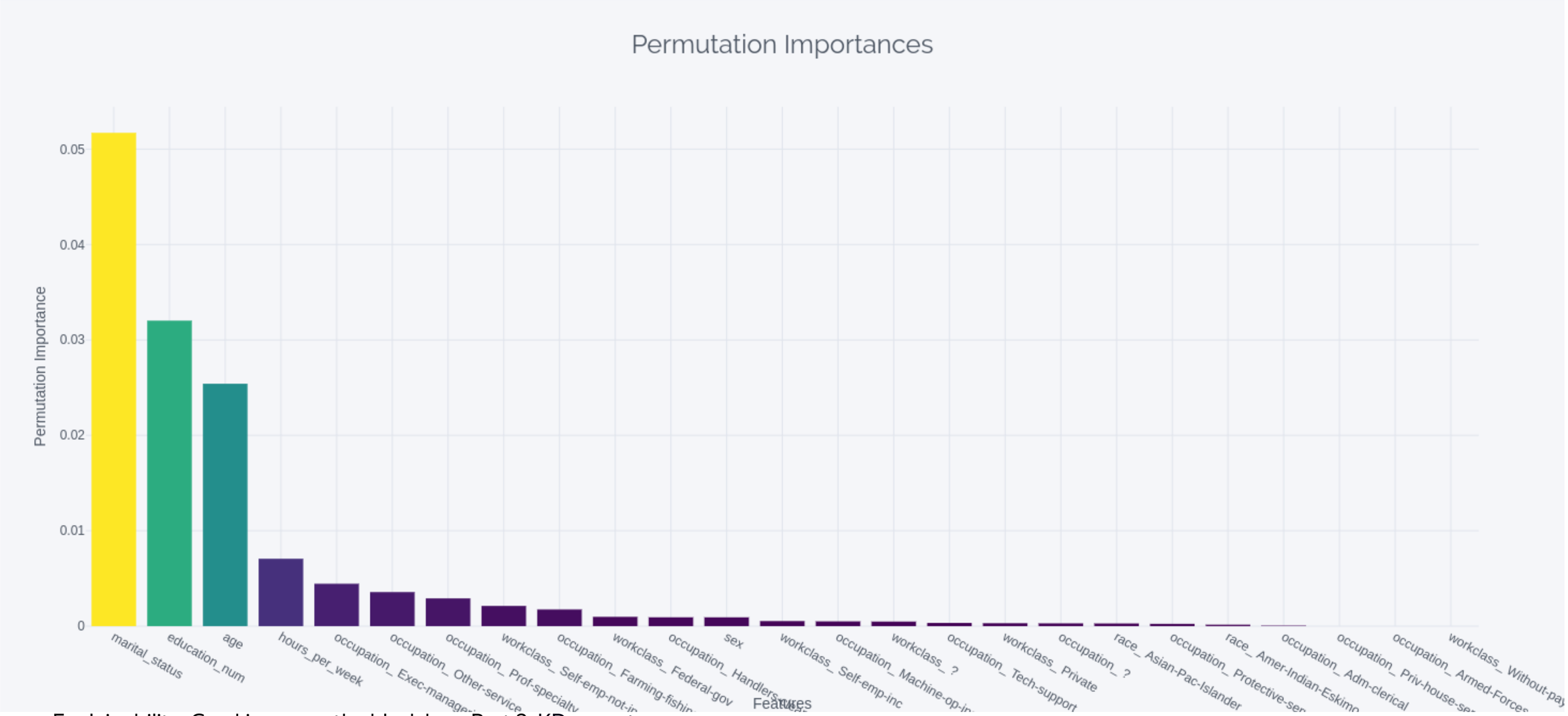
Drop feature/Drop column importance: simply train model without every feature.



Source: Explainability: Cracking open the black box, Part 2, KDnuggets

Importance estimation: model-agnostic methods

Permutation importance – shuffle feature values within the dataset and evaluate trained model.



Source: Explainability: Cracking open the black box, Part 2, KDnuggets

LIME – Local Interpretable Model-agnostic Explanations

03

Fidelity-Interpretability Trade-off

01 The model to be explained: $f : \mathbb{R}^k \longrightarrow \mathbb{R}$

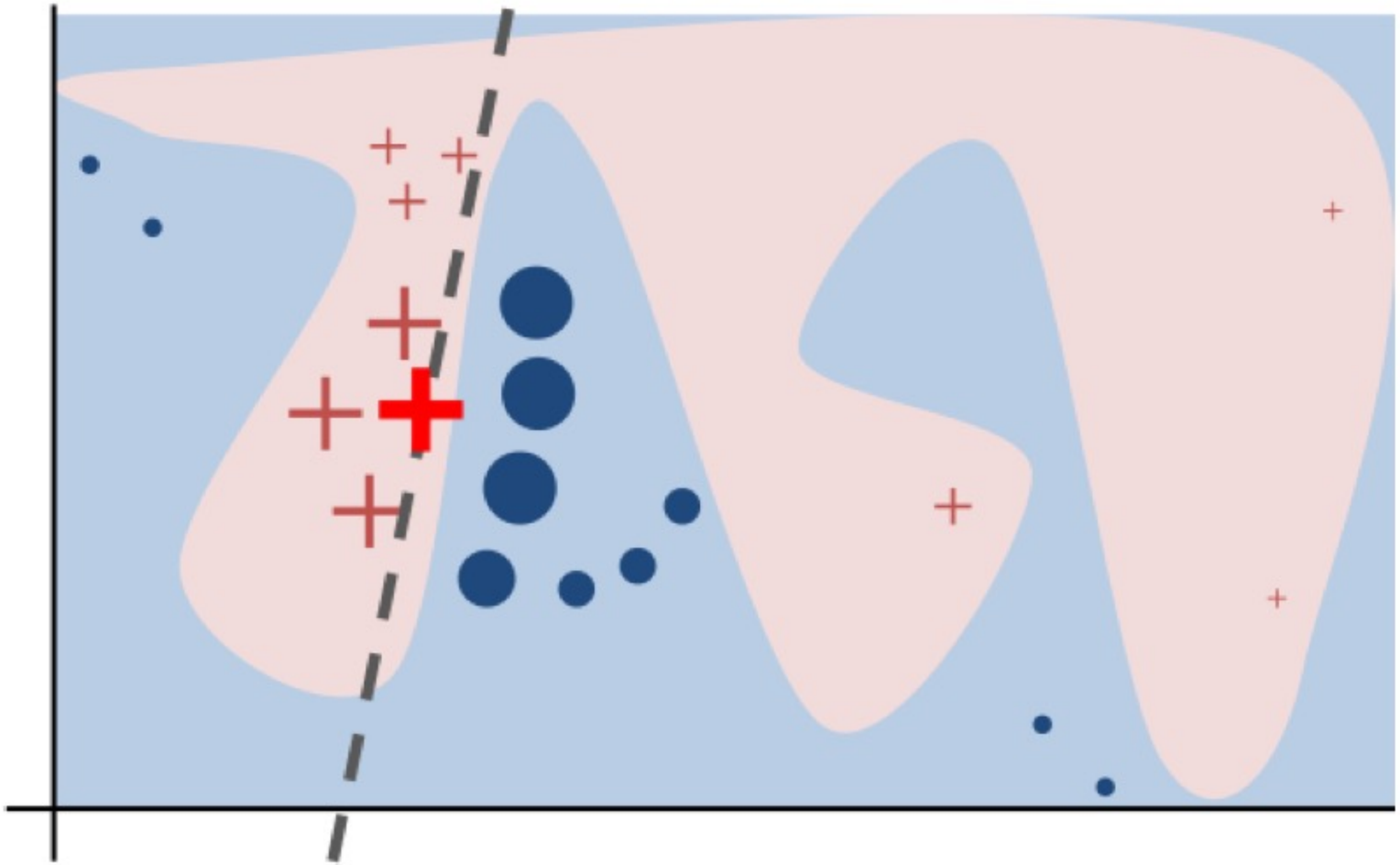
02 The local explainable model $g \in G$, $g \approx f$ in some $\boldsymbol{x}$ neighborhood of

$\pi_{\boldsymbol{x}}(\boldsymbol{z})$ denotes proximity of $\boldsymbol{z}$ to $\boldsymbol{x}$

$\Omega(g)$ measures the complexity of g

03 $\mathcal{L}(f, g, \pi_{\boldsymbol{x}})$ measures how unfaithful g is in approximating f in the locality $\pi_{\boldsymbol{x}}$

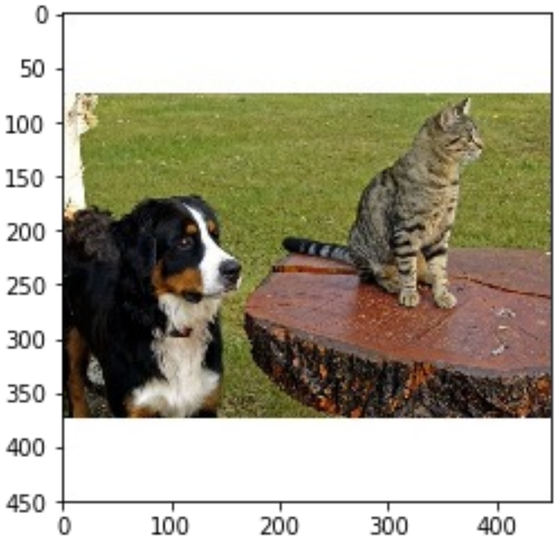
LIME explanation is $\xi(\boldsymbol{x}) = \arg\min_{g \in G} \left(\mathcal{L}(f, g, \pi_{\boldsymbol{x}}) + \Omega(g) \right)$

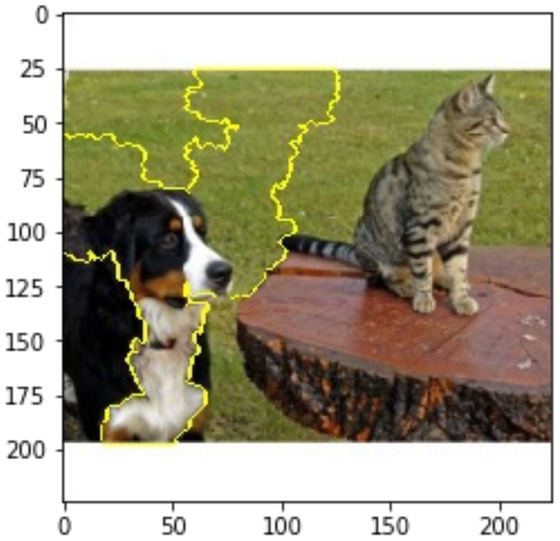


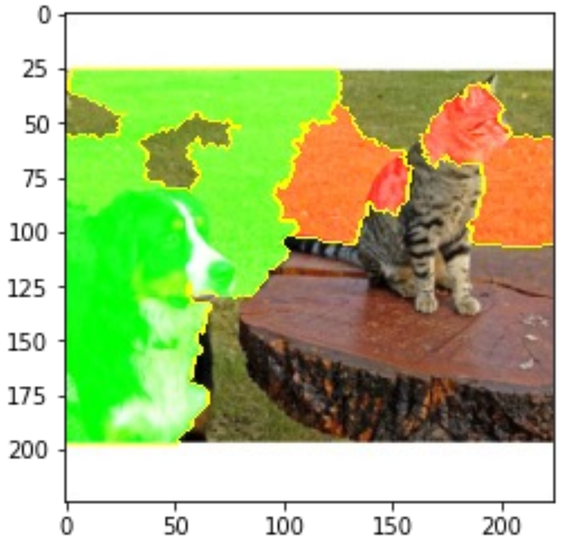
Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets pre- dictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

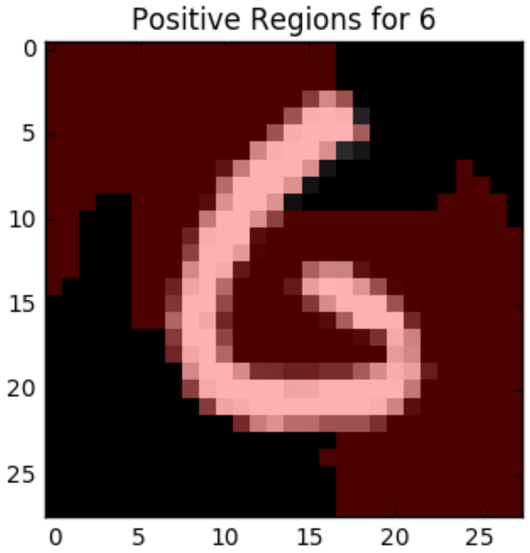
Source:"Why Should I Trust You?": Explaining the Predictions of Any Classifier

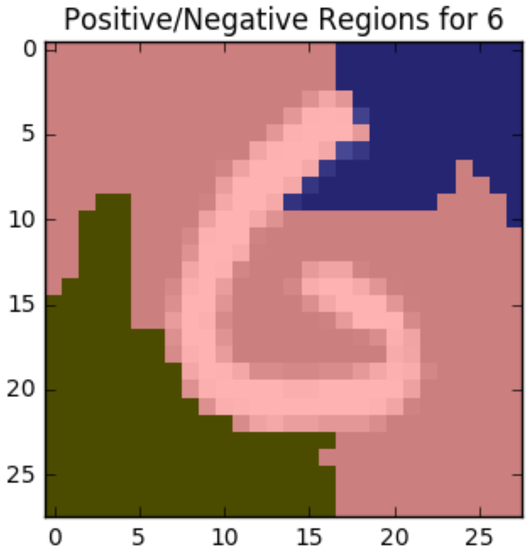
LIME examples





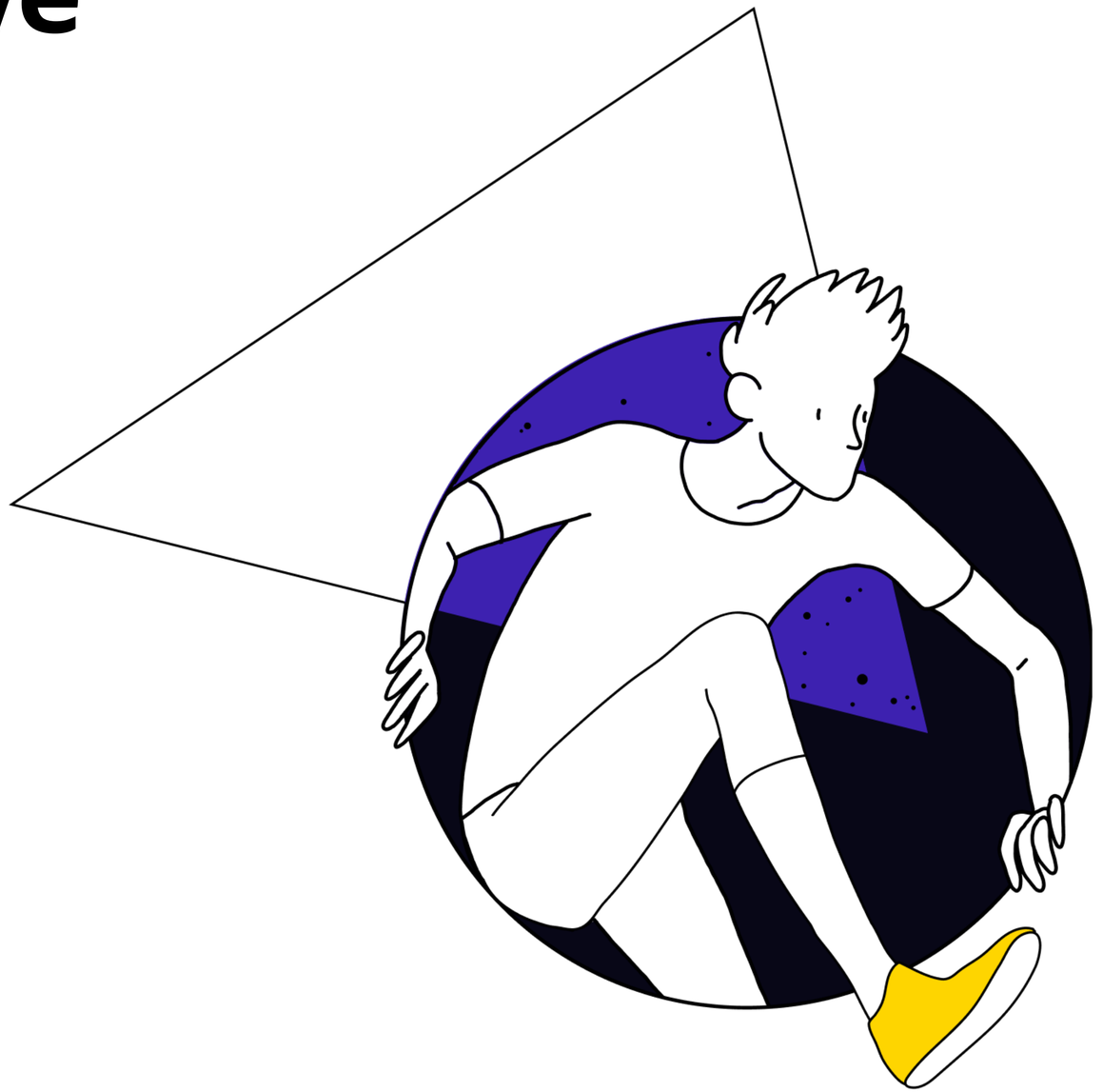






Source: LIME repository

Shapley values.
SHAP – SHapely Additive exPlanations



04

Shapley values estimation

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right]$$

where $S \subseteq F$ is features subset, F is set of all features, $f_{S \cup \{i\}}$ trained on feature subset and i-th feature.

Desirable properties:

01 Local accuracy: $f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i$

02 Missingness: $x'_i = 0 \implies \phi_i = 0$

03 Consistency: $f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i)$ means $\phi_i(f', x) \geq \phi_i(f, x)$

Shapley values: theorem

The only possible explanation model which satisfies properties 1-3 and follows the definition is:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

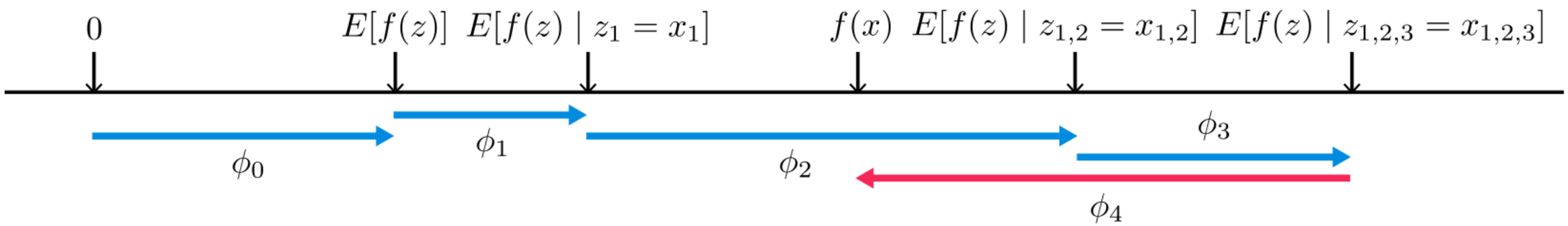
where $|z'|$ is the number of non-zero entries in z', and $z' \subseteq x'$ represents all z' vectors where the non-zero entries are a subset of the non-zero entries in x'.

01 Local accuracy: $f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i$

02 Missingness: $x'_i = 0 \implies \phi_i = 0$

03 Consistency: $f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i)$ means $\phi_i(f', x) \geq \phi_i(f, x)$

SHAP

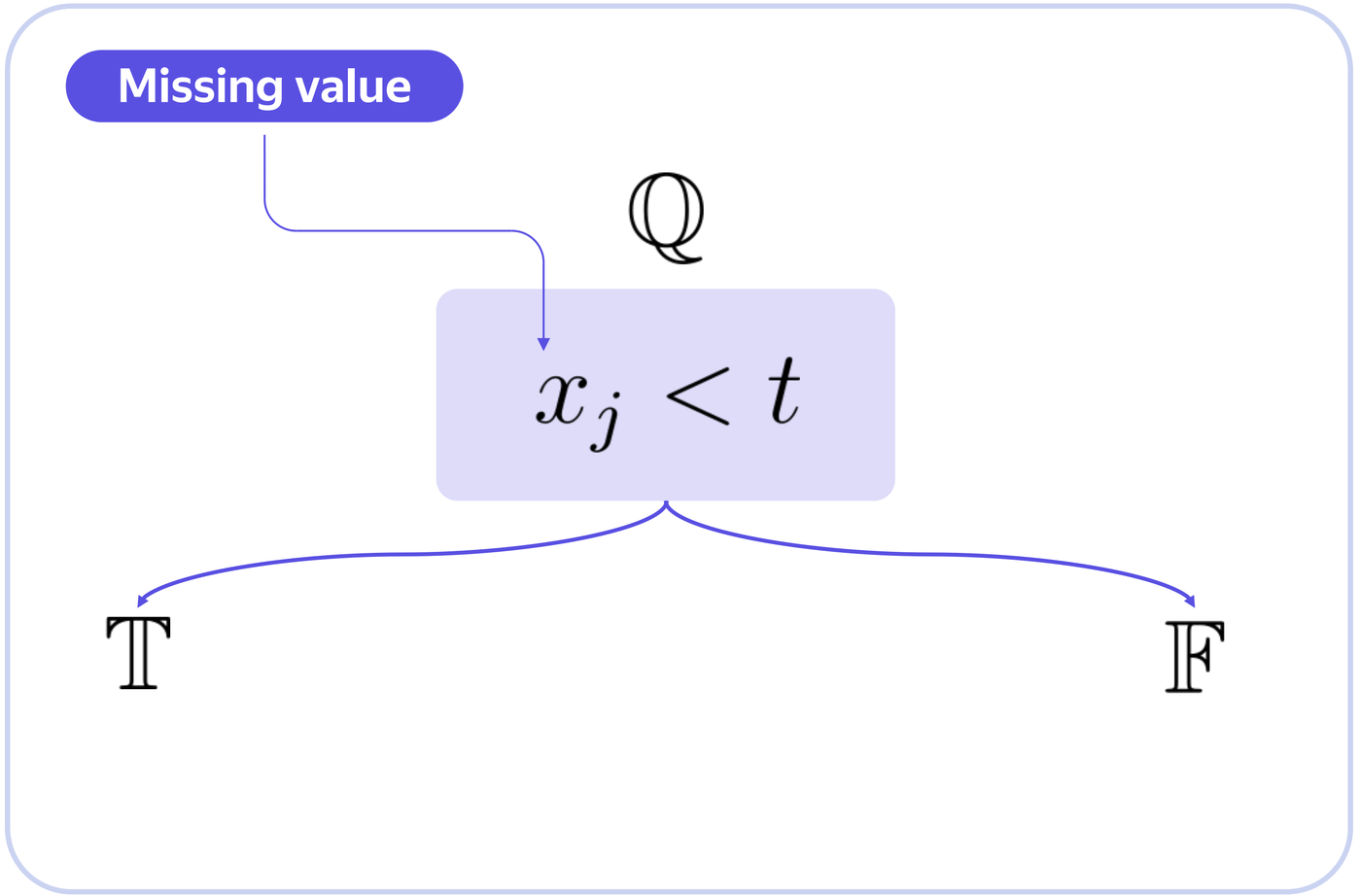


SHAP (SHapley Additive exPlanation) values attribute to each feature the change in the expected model prediction when conditioning on that feature. They explain how to get from the base value E[f(z)] that would be predicted if we did not know any features to the current output f (x). This diagram shows a single ordering. When the model is non-linear or the input features are not independent, however, the order in which features are added to the expectation matters, and the SHAP values arise from averaging the φi values across all possible orderings.

Source: A Unified Approach to Interpreting Model Predictions

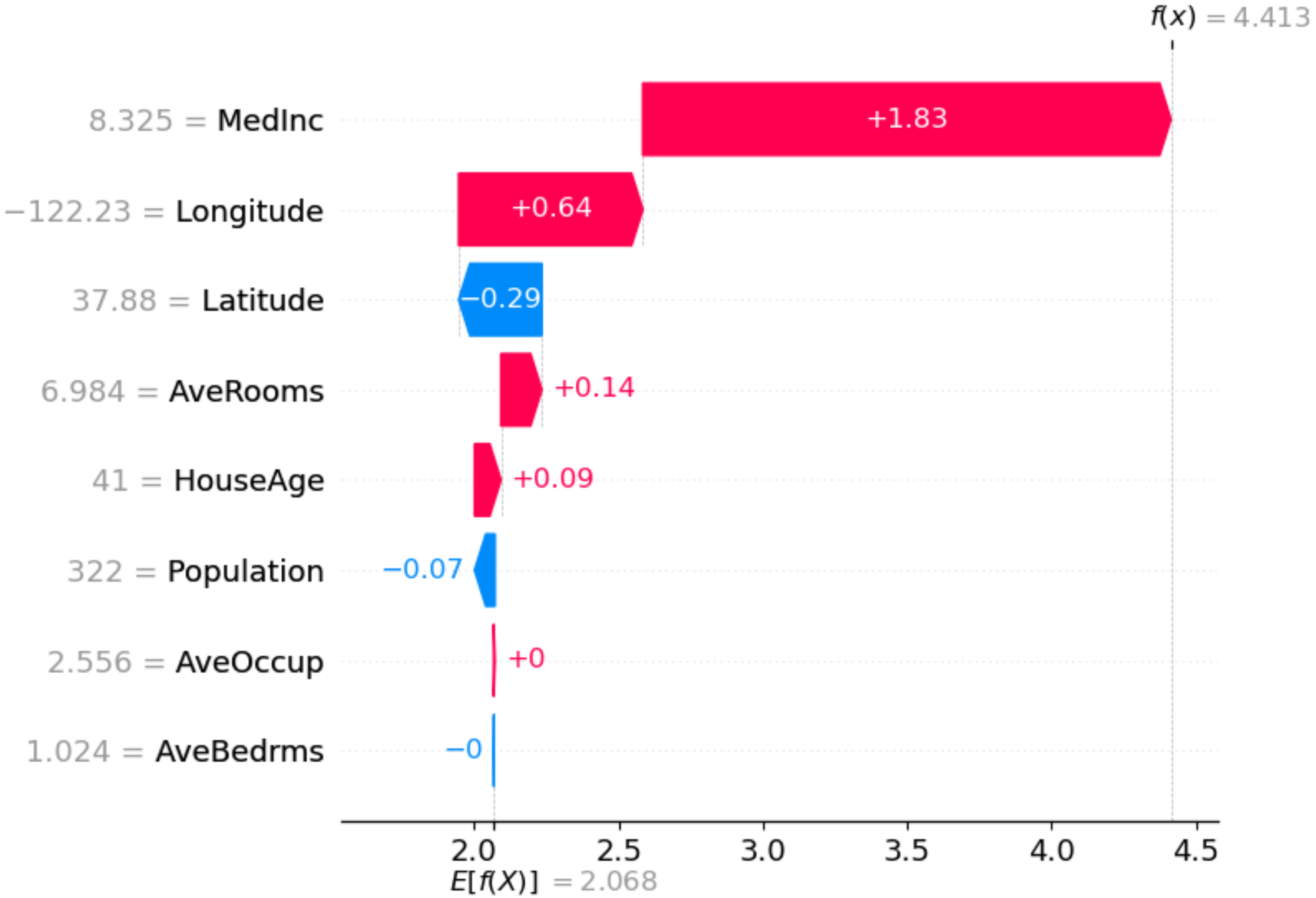
Recap: missing values in Decision Trees

If the value is missing, one might use both sub-trees and average their predictions



$$\hat{y} = \frac{|\mathbb{T}|}{|\mathbb{Q}|}\hat{y}_{\mathbb{T}} + \frac{|\mathbb{F}|}{|\mathbb{Q}|}\hat{y}_{\mathbb{F}}$$

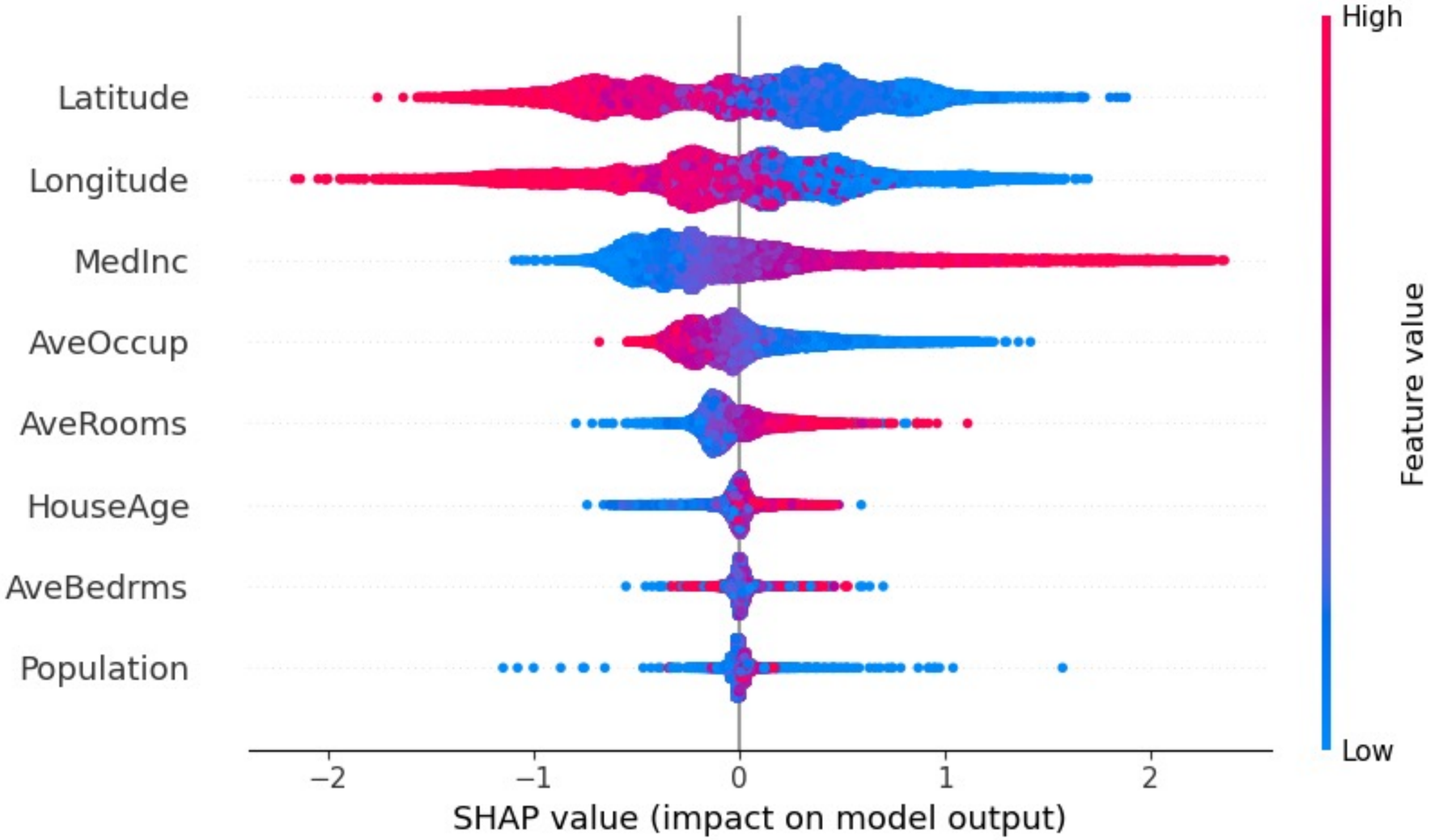
SHAP



Tree ensemble example

Source: shap repository

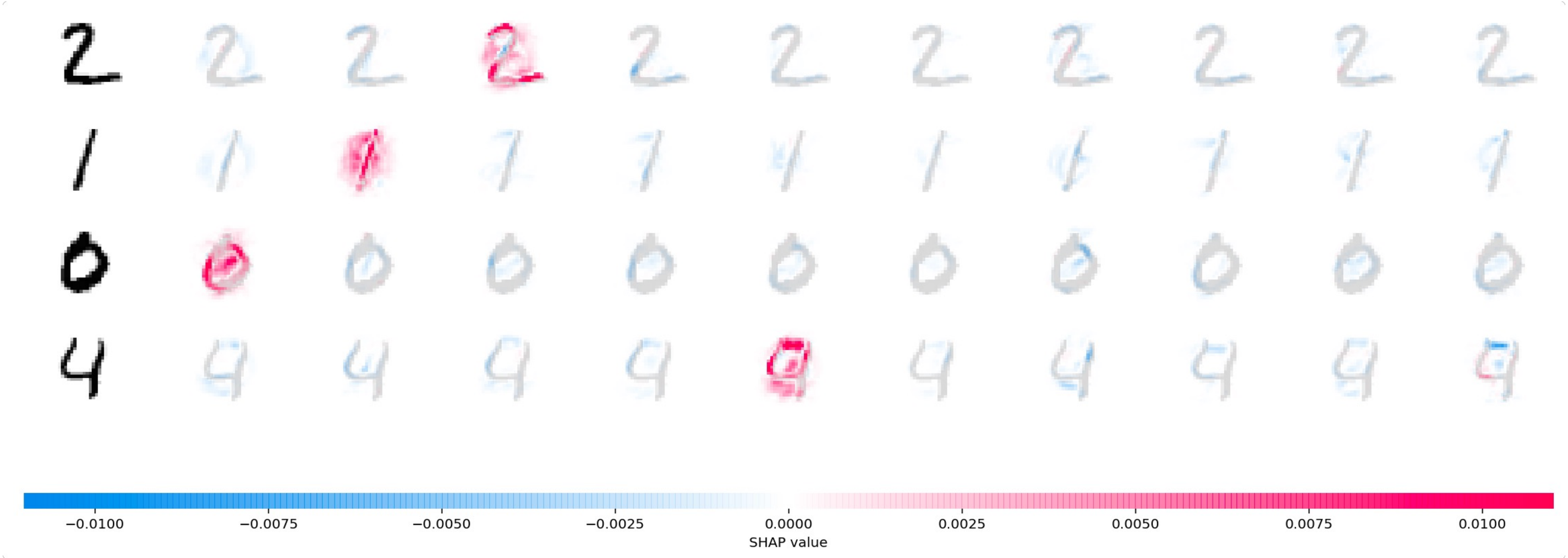
SHAP



SHAP values for every sample and every feature

Source: shap repository

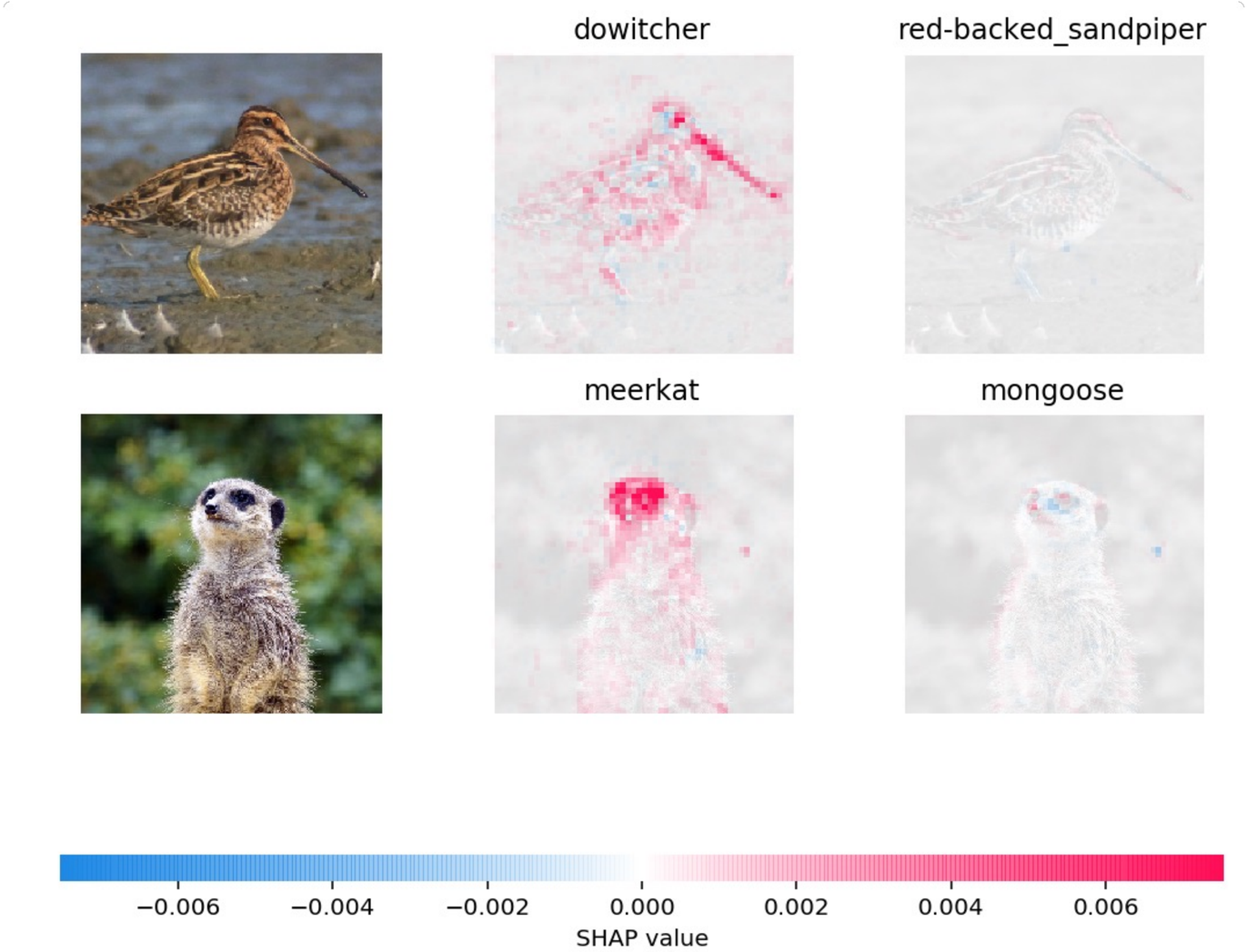
SHAP



DeepExplainer on MNIST data

Source: shap repository

SHAP

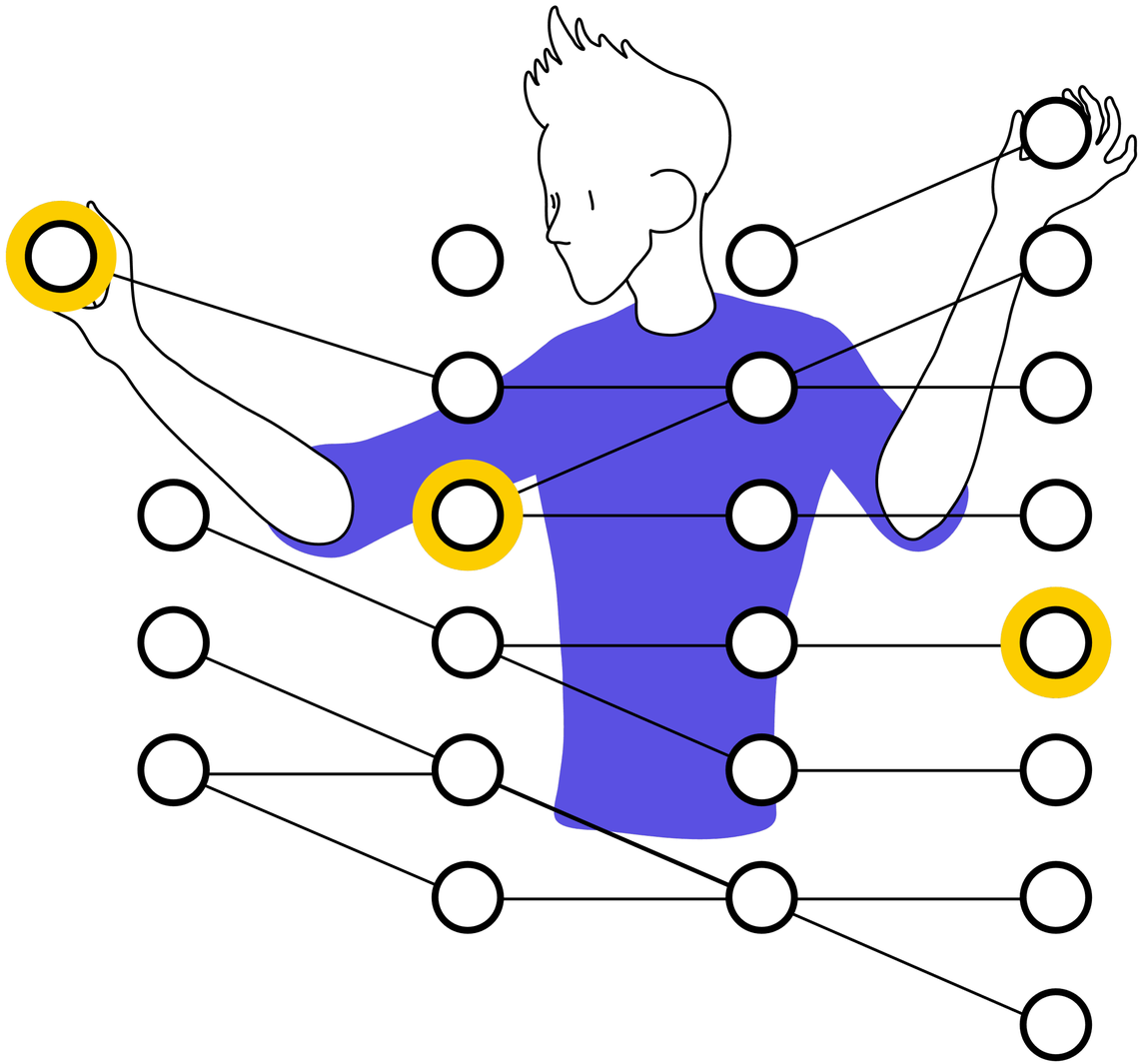


Gradient explainer on image data.

Illustration shows how 7th intermediate layer of the VGG16 ImageNet model impacts the output probabilities

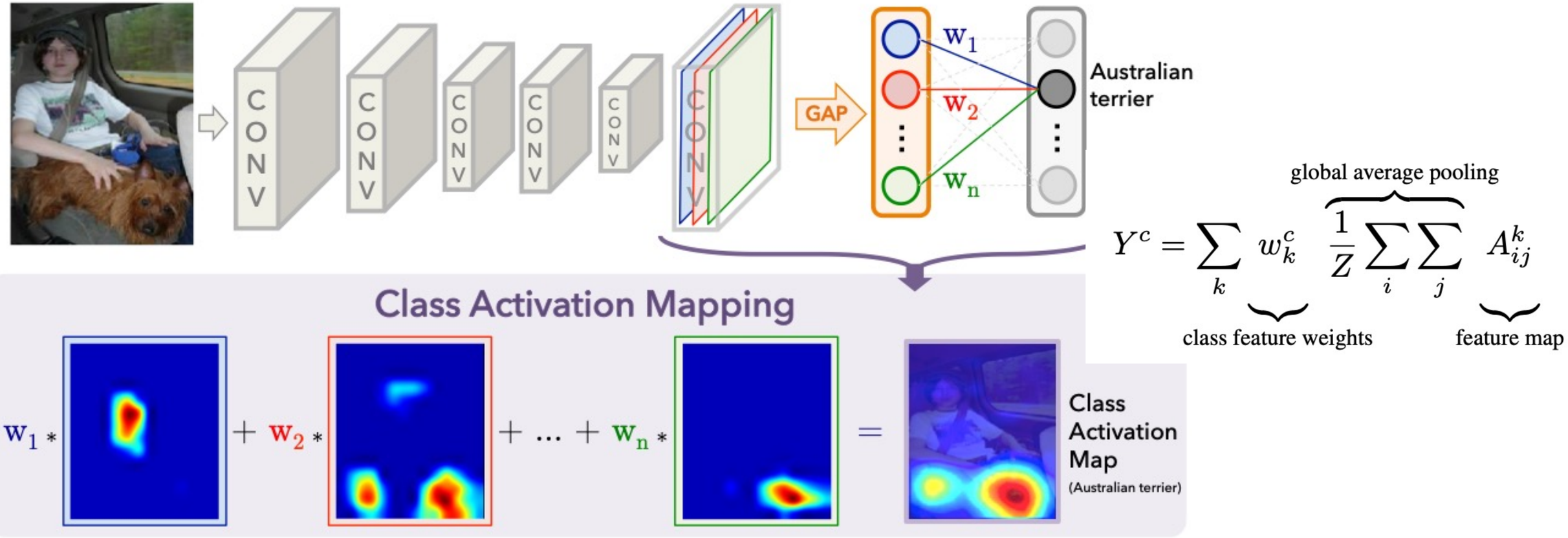
Source: shap repository

(Grad-)CAM(++)



05

CAM – Class Activation Mapping



$$Y^c = \sum_k \underbrace{w_k^c}_{\text{class feature weights}} \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{A_{ij}^k}_{\text{feature map}}$$

Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

Source: Learning Deep Features for Discriminative Localization

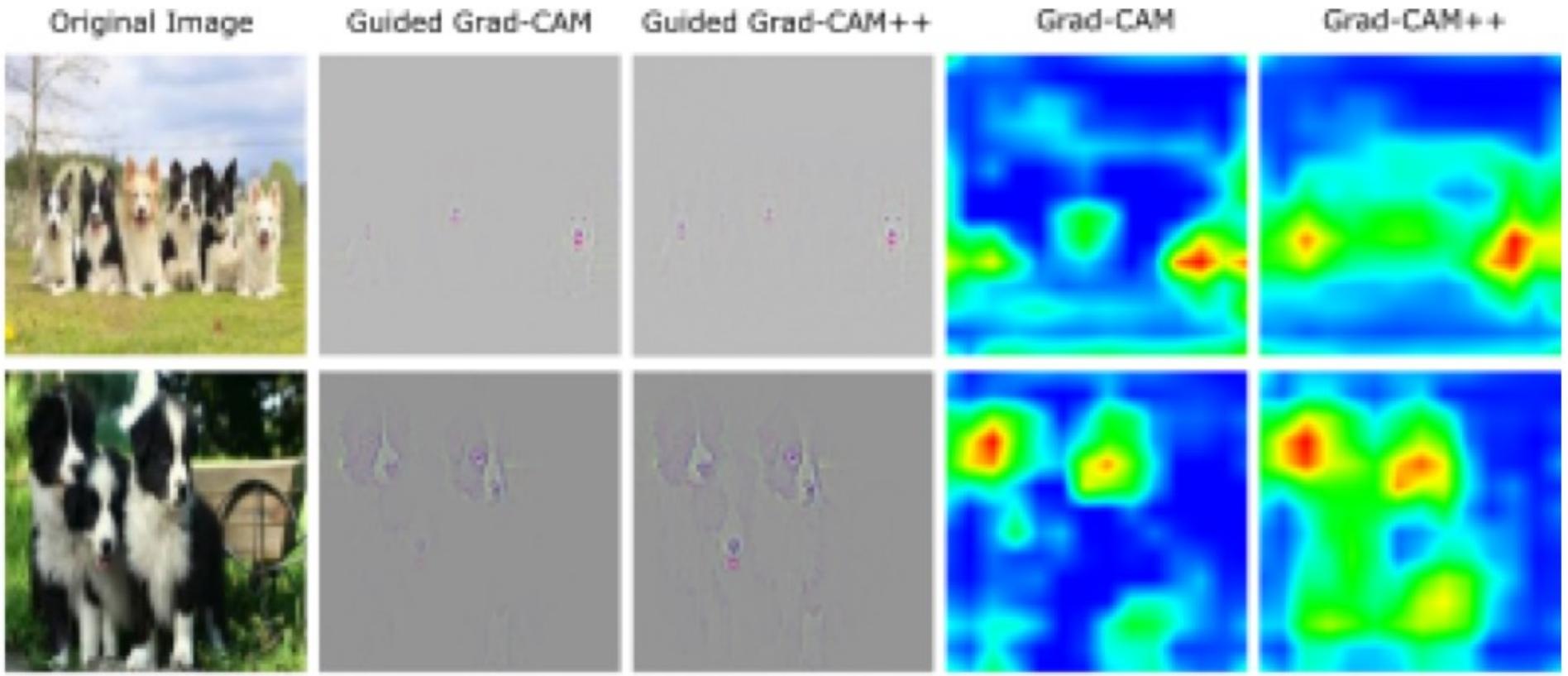
Grad-CAM

(a) Original Image (b) Cat Counterfactual exp (c) Dog Counterfactual exp

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

Source: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Grad-CAM++



$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} . relu(\frac{\partial Y^c}{\partial A_{ij}^k})$$

$$\alpha_{ij}^{kc} = \frac{1}{\sum_{l,m} \frac{\partial y^c}{\partial A_{lm}^k}} \quad if \quad \frac{\partial y^c}{\partial A_{ij}^k} = 1$$

$$= 0 \quad \text{otherwise}$$

Source: Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks

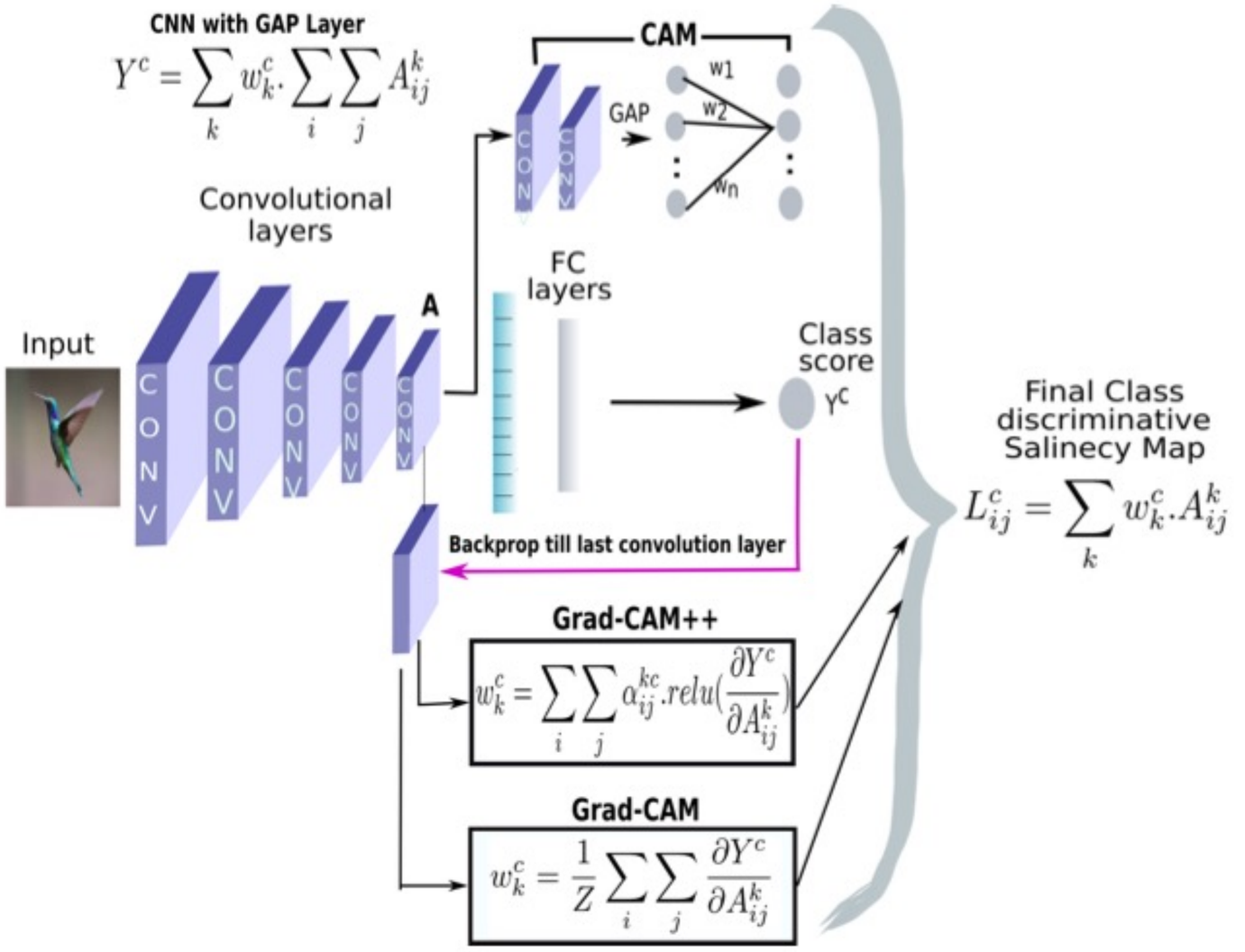


Fig. 3. An overview of all the three methods – CAM, Grad-CAM, Grad-CAM++ – with their respective computation expressions.

Source: Grad-CAM++: Improved Visual Explanations for Deep Convolutional 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