# Explainable AI

Christian Salvatore
Scuola Universitaria Superiore IUSS Pavia

christian.salvatore@iusspavia.it

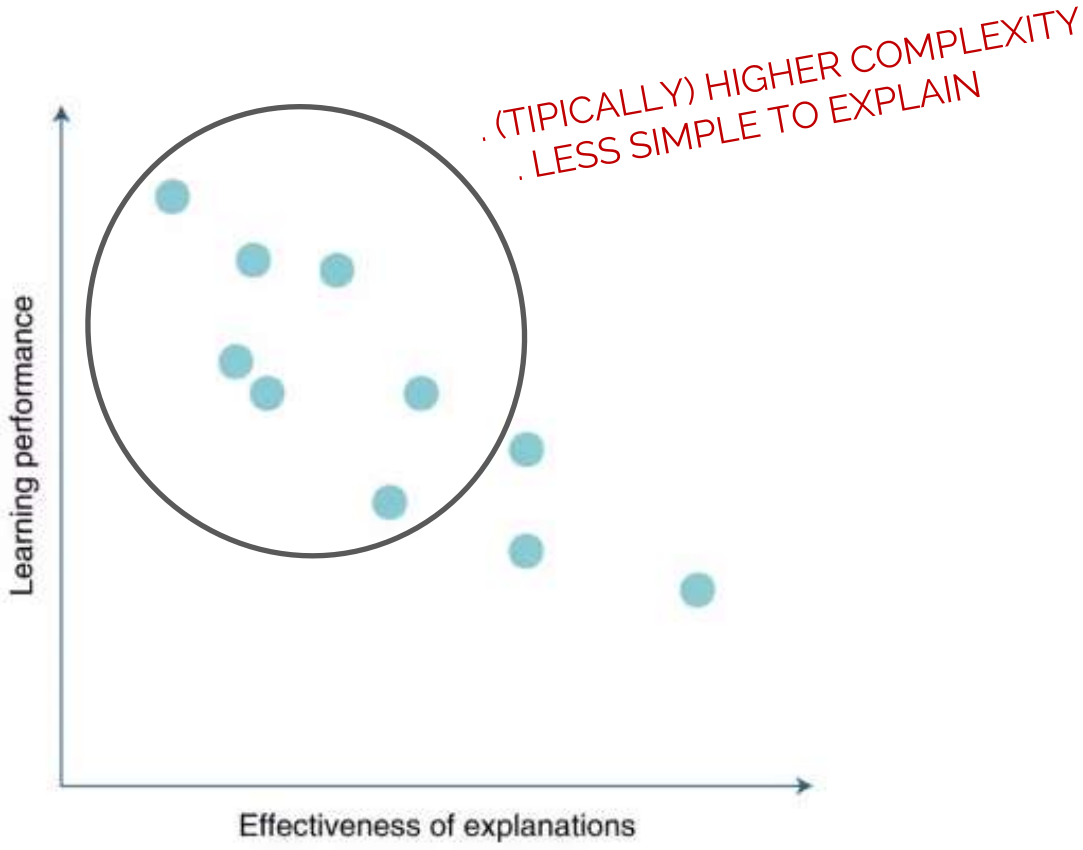
# Improve model explainability

## Interpretability

To observe the inner mechanics of the AI/ML method.
To understand the model's weights and features that are learned/used to determine the output of a classifier.

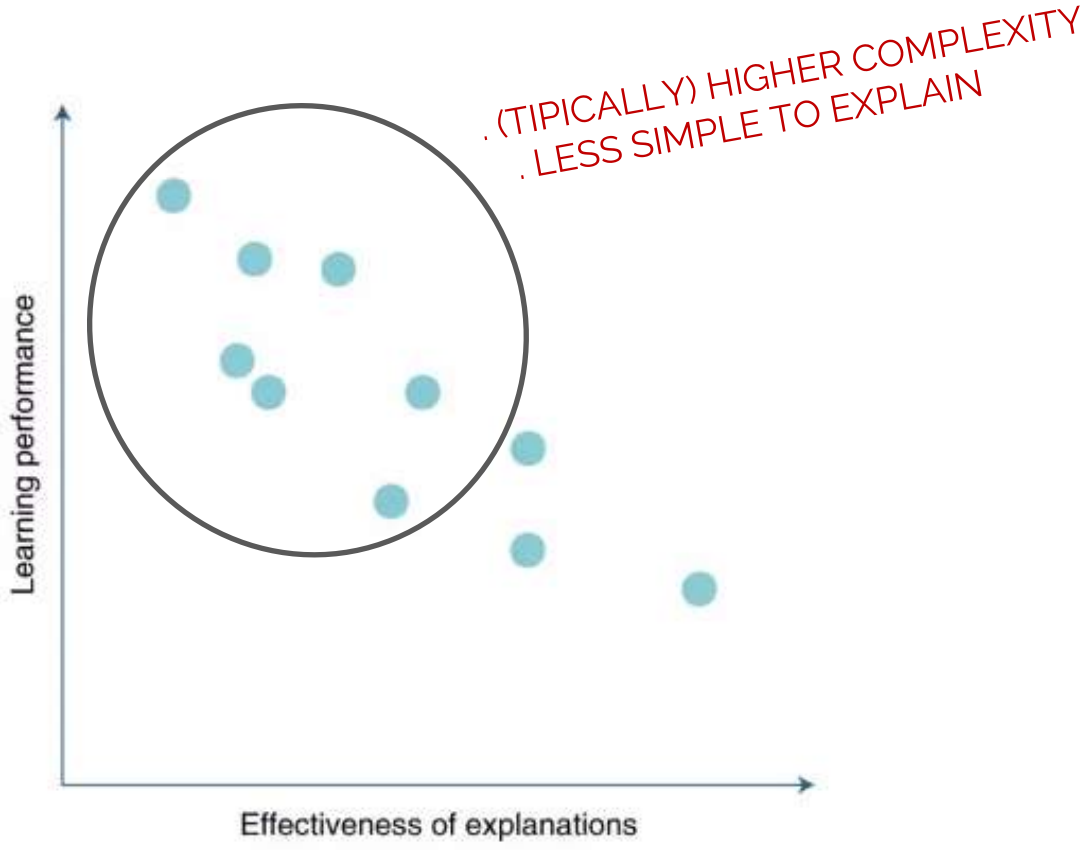
# Improve model explainability

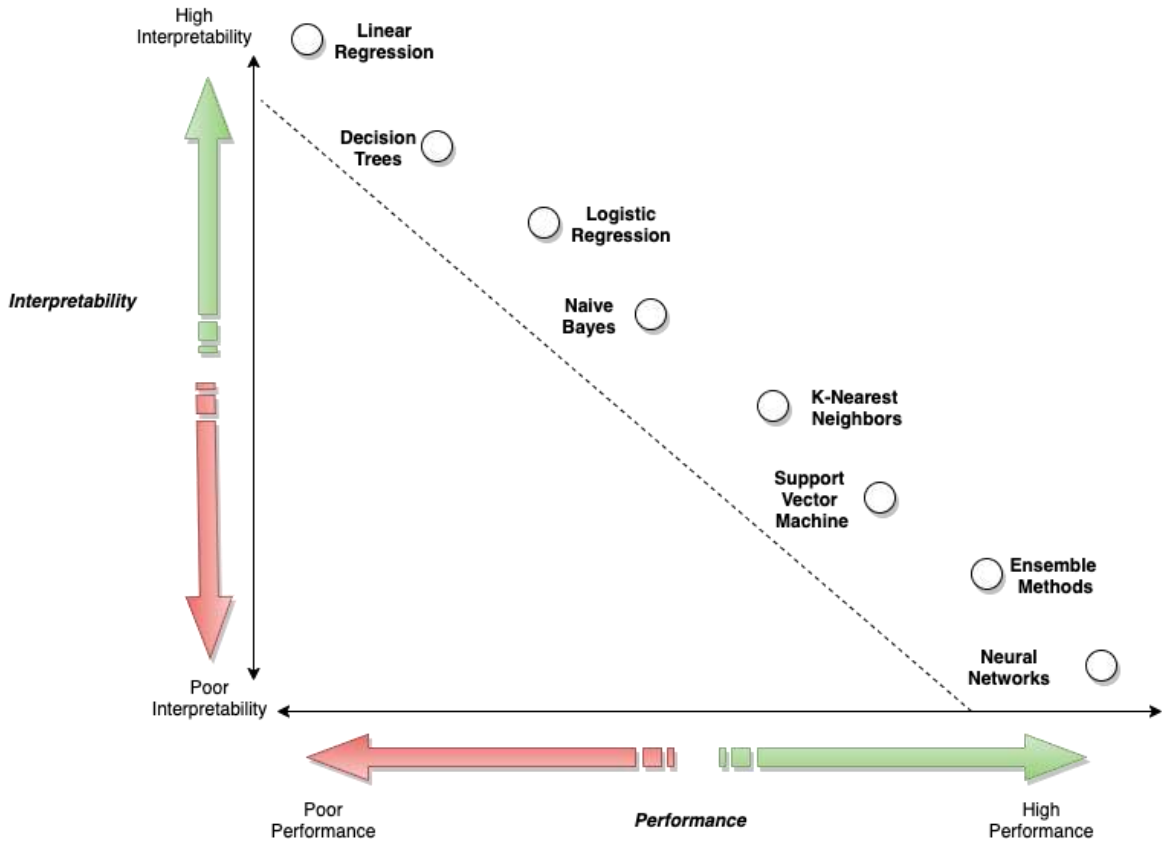
## Performance-vs-interpretability tradeoff



# Improve model explainability

## Performance-vs-interpretability tradeoff



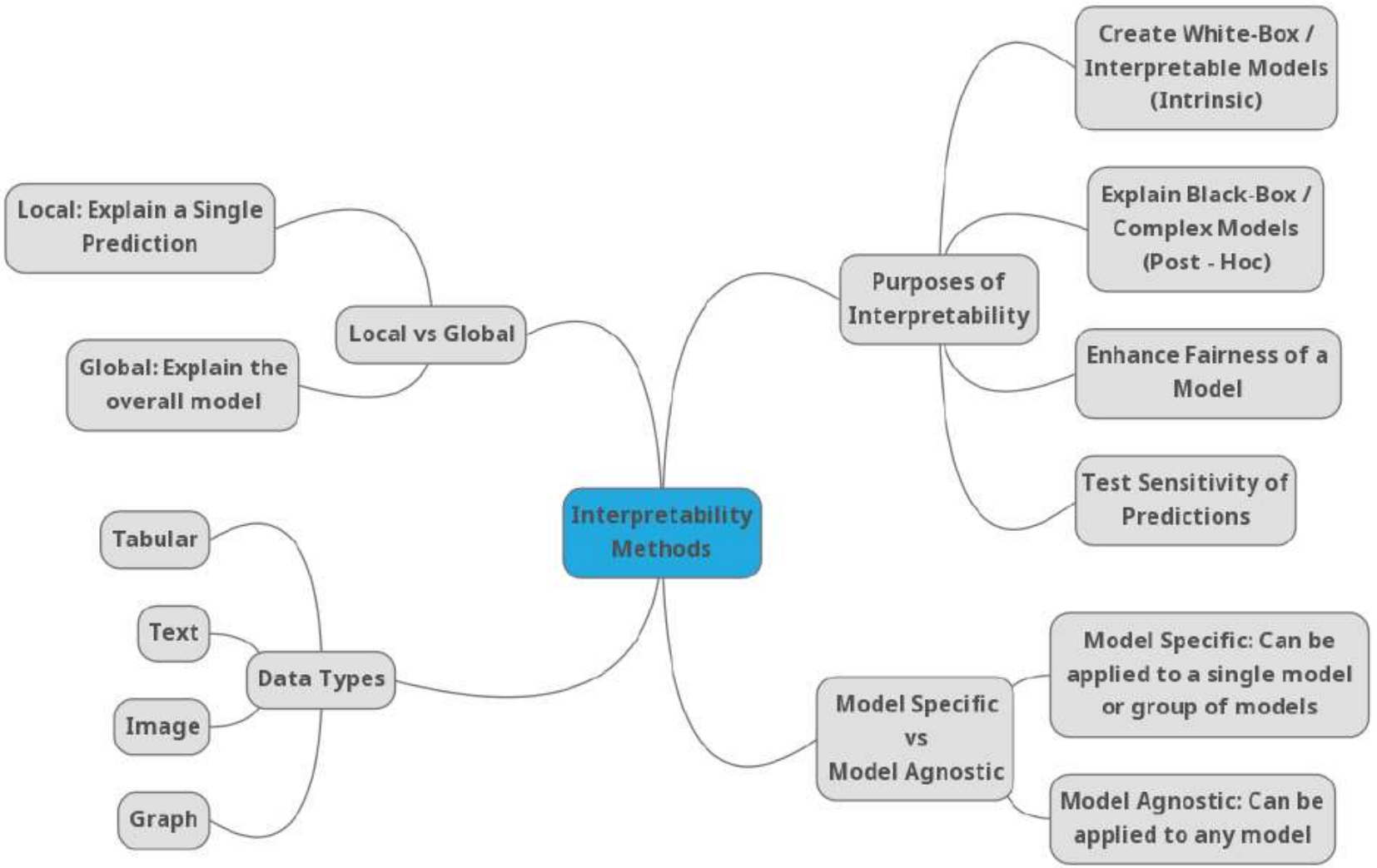

https://docs.aws.amazon.com/whitepapers/latest/model-explainability-aws-ai-ml/interpretability-versus-explainability.html

# Improve model explainability

## Explainability

To explain the behavior of a model at the decision level in human terms.

# Improve model explainability




entropy

Review

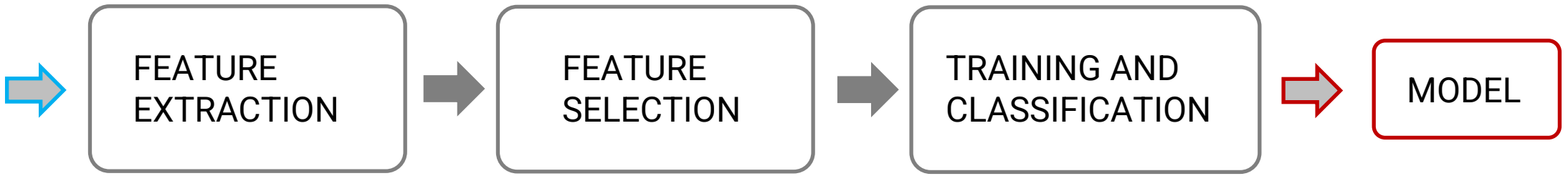
**Explainable AI: A Review of Machine Learning Interpretability Methods**

Pantelis Linardatos *, Vasilis Papastefanopoulos and Sotiris Kotsiantis

Figure 2. Taxonomy mind-map of Machine Learning Interpretability Techniques.

# Improve model explainability

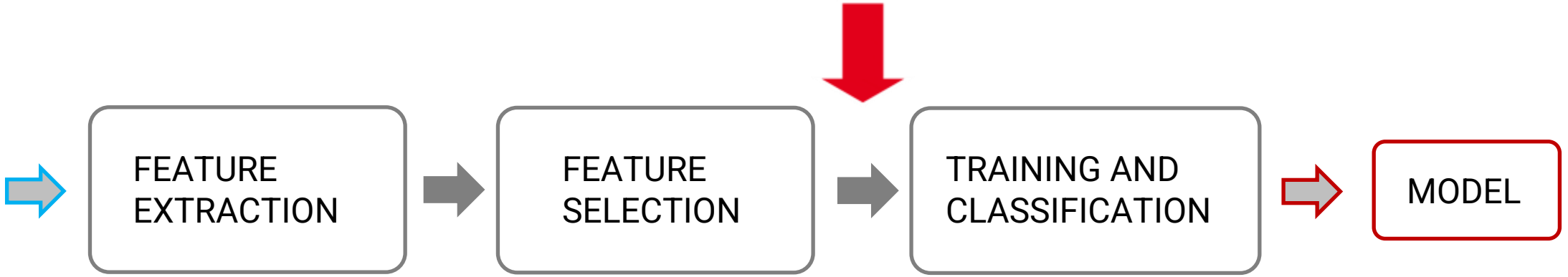
CONVENTIONAL MACHINE LEARNING



# Improve model explainability

## 1. Feature-selection level

CONVENTIONAL MACHINE LEARNING

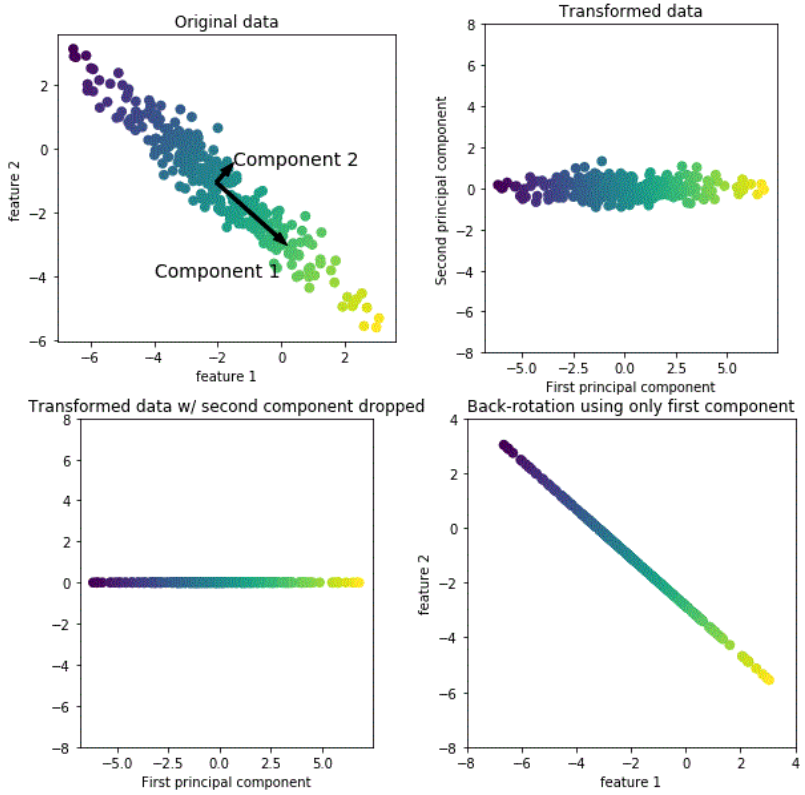


# Improve model explainability

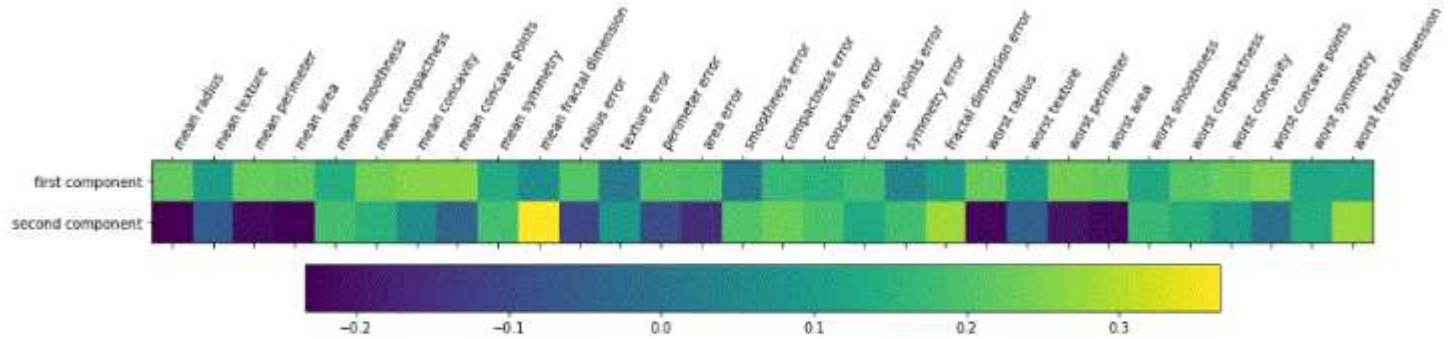
## 1. Feature-selection level

CONVENTIONAL MACHINE LEARNING

(CHARACTERISTICS OF) SELECTED FEATURES




Backprojecting PCA components

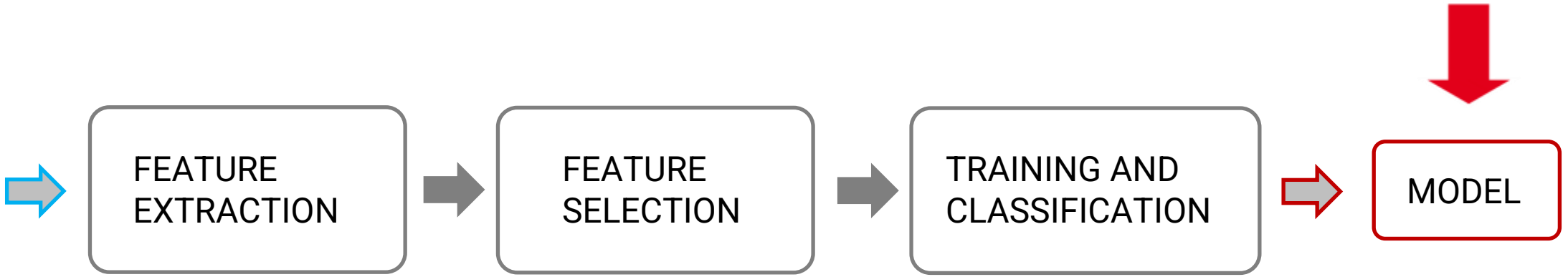



Above picture establishes (a) 1st Component has got all variables positively correlated meaning has got strong correlation (b) 2nd component has mixed sign meaning not so strong correlation (c) mixture of variables orients axes in such a way that 1st to n components have decresing order of relationship

# Improve model explainability

## 2. Post-hoc wrt the output of the training process

CONVENTIONAL MACHINE LEARNING
DEEP LEARNING



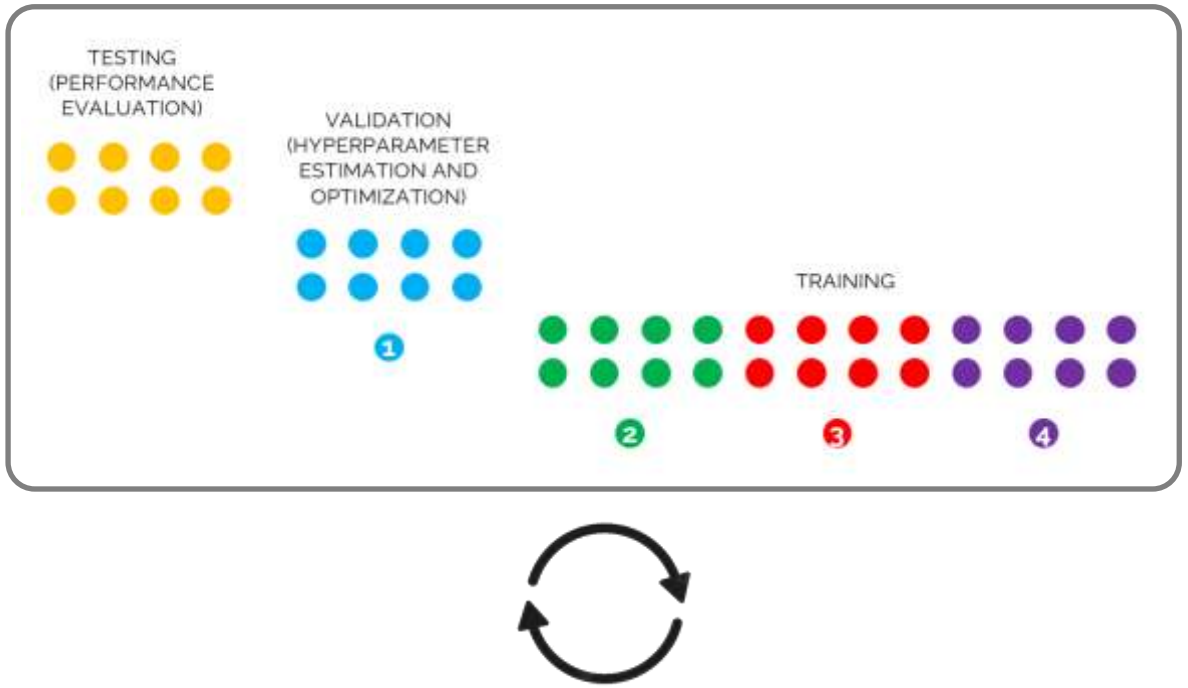
# Improve model explainability

## 2. Post-hoc wrt the output of the training process

CONVENTIONAL MACHINE LEARNING

SELECTED FEATURES RANKED BY FREQUENCY IN (NESTED) CV

- For each loop i of the (nested) cross validation, perform feature selection
- Train the model i using the selected set of features
- Optimize the set of selected features by minimizing/maximizing a given parameter (e.g., model performance)
- Consider the set of selected features corresponding to the optimal models and compute the selection frequency for each feature
- The highest the selection frequency, the highest the importance of the feature



# Improve model explainability

## 2. Post-hoc wrt the output of the training process

CONVENTIONAL MACHINE LEARNING
DEEP LEARNING

Table 2. Interpretability Methods to Explain any Black-Box Model.

| Ref | Tool | Category | Local vs. Global | Model Specific vs. Model Agnostic | Data Type | Citations/ Year | Year |
|---|---|---|---|---|---|---|---|
| [45] | lime<br>Eli5<br>InterpretML<br>AIX360<br>Skater | PH | L | Agnostic | img<br>txt<br>tab | 845.6 | 2016 |
| [59] | PDPbox<br>InterpretML<br>Skater | PH | G | Agnostic | tab | 589.2 | |
| [48] | shap<br>alibi<br>AIX360<br>InterpretML | PH | L & G | Agnostic | img<br>txt<br>tab | 504.5 | |
| [50] | alibi<br>Anchor | PH | L | Agnostic | img<br>txt<br>tab | 158.3 | |
| [53] | alibi | PH | L | Agnostic | tab<br>img | 124.5 | |
| [60] | PyCEbox | PH | L & G | Agnostic | tab | 53.3 | 2015 |
| [58] | L2X | PH | L | Agnostic | img<br>txt<br>tab | 50.3 | 2018 |
| [57] | Eli5 | PH | G | Agnostic | tab | 41.5 | 2010 |
| [51] | alibi<br>AIX360 | PH | L | Agnostic | tab<br>img | 34.3 | 2018 |
| [61] | Alibi | PH | G | Agnostic | tab | 23.2 | 2016 |
| [54] | alibi | PH | L | Agnostic | tab<br>img | 17 | 2019 |
| [62] | pyBreakDown | PH | L | Agnostic | tab | 8.3 | 2018 |
| [62] | pyBreakDown | PH | G | Agnostic | tab | 8.3 | 2018 |
| [47] | DLIME | PH | L | Agnostic | img<br>txt<br>tab | 7.5 | 2019 |
| [56] | AIX360 | PH | L | Agnostic | tab | 7 | 2019 |
| [52] | AIX360 | PH | L | Agnostic | tab<br>img | 3 | 2019 |

# Improve model explainability

## 2. Post-hoc wrt the output of the training process

CONVENTIONAL MACHINE LEARNING
DEEP LEARNING

### LIME: Local Interpretable Model-agnostic Explanations

- For any given instance and its corresponding prediction, generate a set of simulated randomly-sampled data around the neighbourhood of input instance (“perturbed” instances)
- Classify these instances (using the model to “explain”)
- Weight these classifications by their proximity to the “real” input instance
- Train a simple, interpretable model using simulated instances and corresponding classification labels (e.g., a decision tree)
- Interpret this newly trained (local) model: this will result in an approximated interpretation of the original black-box model

# Improve model explainability

## 2. Post-hoc wrt the output of the training process

CONVENTIONAL MACHINE LEARNING
DEEP LEARNING

LIME: Local Interpretable Model-agnostic Explanations

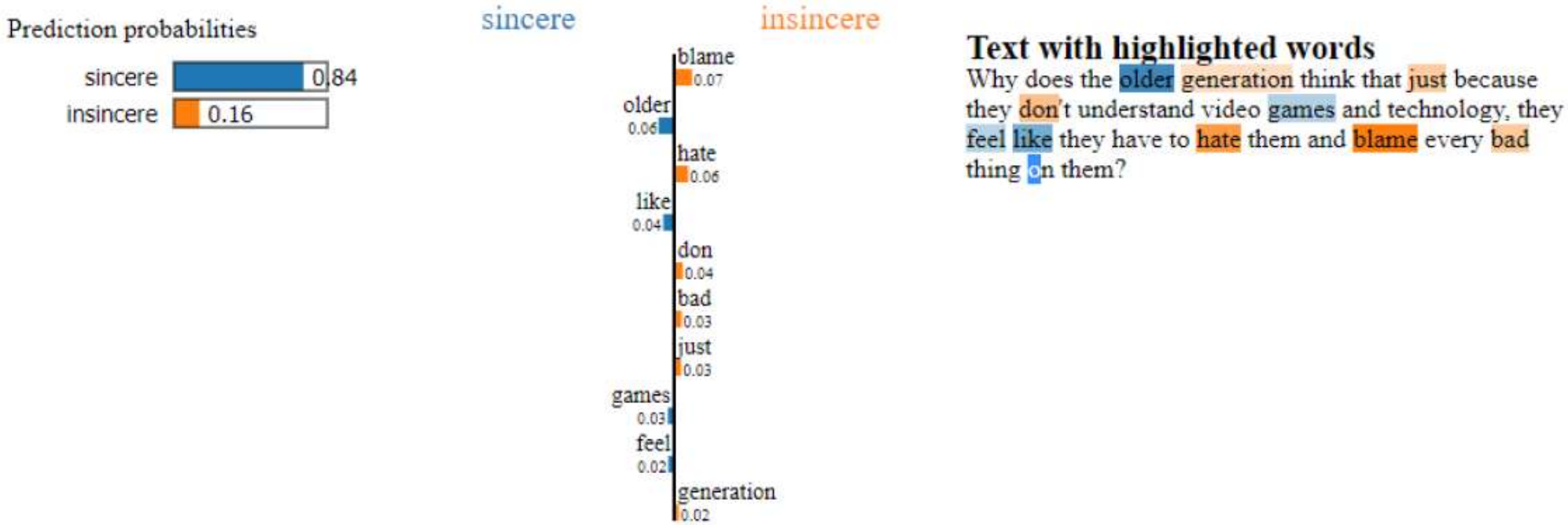



**Figure 4.** Local interpretable model-agnostic explanations (LIME) is used to explain the rationale behind the classification of an instance of the Quora Insincere Questions Dataset.

# Improve model explainability

## 2. Post-hoc wrt the output of the training process

CONVENTIONAL MACHINE LEARNING
DEEP LEARNING

SHAP: Shapley Additive Explanations

- Classic result in game theory on distributing the total gain from a cooperative game
- Introduced by Lloyd Shapley in 1953 (Nobel Prize in Economics in 2012)

How it works:

### Cooperative Game

- Players {1, ..., M} collaborating to generate some **gain**
  - Think: Employees in a company creating some profit
  - Described by a **set function v(S)** specifying the gain for any subset $S \subseteq \{1, ..., M\}$

- **Shapley values** are a fair way to attribute the total gain to the players
  - Think: Bonus allocation to the employees
  - Shapley values are commensurate with the player's contribution

On courtesy of Dr. Ankur Taly

# Improve model explainability

## 2. Post-hoc wrt the output of the training process

CONVENTIONAL MACHINE LEARNING
DEEP LEARNING

SHAP: Shapley Additive Explanations

### Shapley Value Algorithm [Conceptual]

$$\phi_i(v) = \mathop{\mathbb{E}}_{\boldsymbol{O}\sim\pi(M)}[v(\text{pre}_i(\boldsymbol{O}) \cup \{i\}) - v(\text{pre}_i(\boldsymbol{O}))]$$

- Consider all possible permutations $\pi(M)$ of players (**M! possibilities**)
- In each permutation $\boldsymbol{O} \sim \pi(M)$
  - Add players to the coalition in that order
  - Note the marginal contribution of each player i to set of players before it in the permutation, i.e., $v(pre_i(\boldsymbol{O}) \cup \{i\}) - v(pre_i(\boldsymbol{O}))$
- The average marginal contribution across all permutations is the Shapley Value

On courtesy of Dr. Ankur Taly

# Improve model explainability

## 2. Post-hoc wrt the output of the training process

CONVENTIONAL MACHINE LEARNING
DEEP LEARNING

SHAP: Shapley Additive Explanations

### Example

A company with two employees **Alice** and **Bob**

- No employees, no profit [v({}) = 0]
- Alice alone makes 20 units of profit [v({Alice}) = 20]
- Bob alone makes 10 units of profit [v({Bob}) = 10]
- Alice and Bob make 50 units of profit [v({Alice, Bob}) = 50]

**What should the bonuses be?**

On courtesy of Dr. Ankur Taly

# Improve model explainability

## 2. Post-hoc wrt the output of the training process

CONVENTIONAL MACHINE LEARNING
DEEP LEARNING

SHAP: Shapley Additive Explanations

### Example

A company with two employees **Alice** and **Bob**

- No employees, no profit [v({}) = 0]
- Alice alone makes 20 units of profit [v({Alice}) = 20]
- Bob alone makes 10 units of profit [v({Bob}) = 10]
- Alice and Bob make 50 units of profit [v({Alice, Bob}) = 50]

**What should the bonuses be?**

| Permutation | Marginal for Alice | Marginal for Bob |
|---|---|---|
| Alice, Bob | 20 | 30 |
| Bob, Alice | 40 | 10 |
| **Shapley Value** | **30** | **20** |

On courtesy of Dr. Ankur Taly

# Improve model explainability

## 2. Post-hoc wrt the output of the training process

CONVENTIONAL MACHINE LEARNING
DEEP LEARNING

### SHAP: Shapley Additive Explanations

**Additivity**
“SHAP values are additive, which means that the contribution of each feature to the final prediction can be computed independently and then summed up. This property allows for efficient computation of SHAP values, even for high-dimensional datasets.”

**Local accuracy**
“SHAP values add up to the difference between the expected model output and the actual output for a given input. This means that SHAP values provide an accurate and local interpretation of the model's prediction for a given input.”

**Missingness**
“SHAP values are zero for missing or irrelevant features for a prediction. This makes SHAP values robust to missing data and ensures that irrelevant features do not distort the interpretation.”

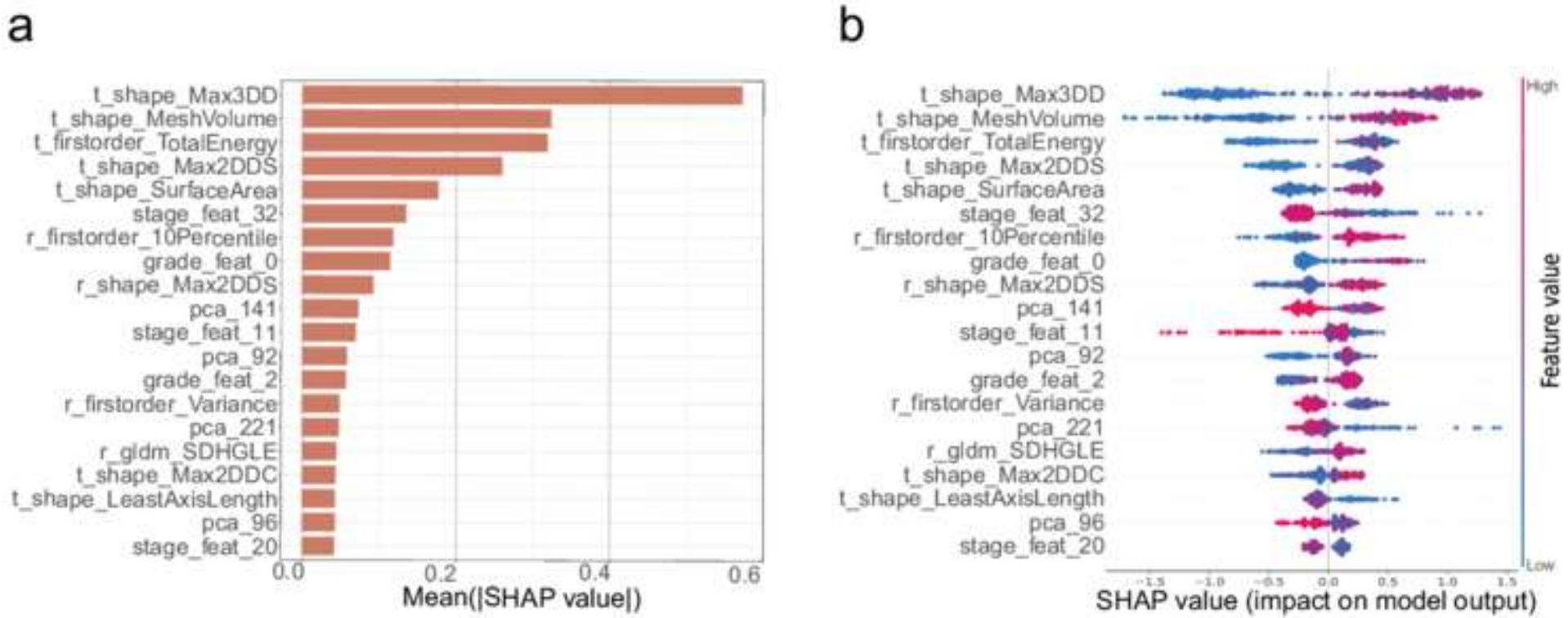
**Consistency**
“SHAP values do not change when the model changes unless the contribution of a feature changes. This means that SHAP values provide a consistent interpretation of the model's behavior, even when the model architecture or parameters change.”

# Improve model explainability

## 2. Post-hoc wrt the output of the training process

CONVENTIONAL MACHINE LEARNING
DEEP LEARNING

SHAP: Shapley Additive Explanations



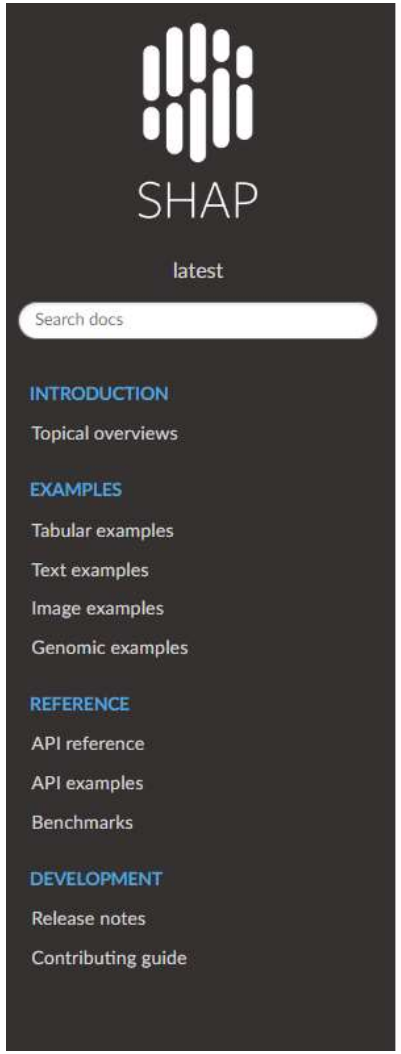
# Improve model explainability

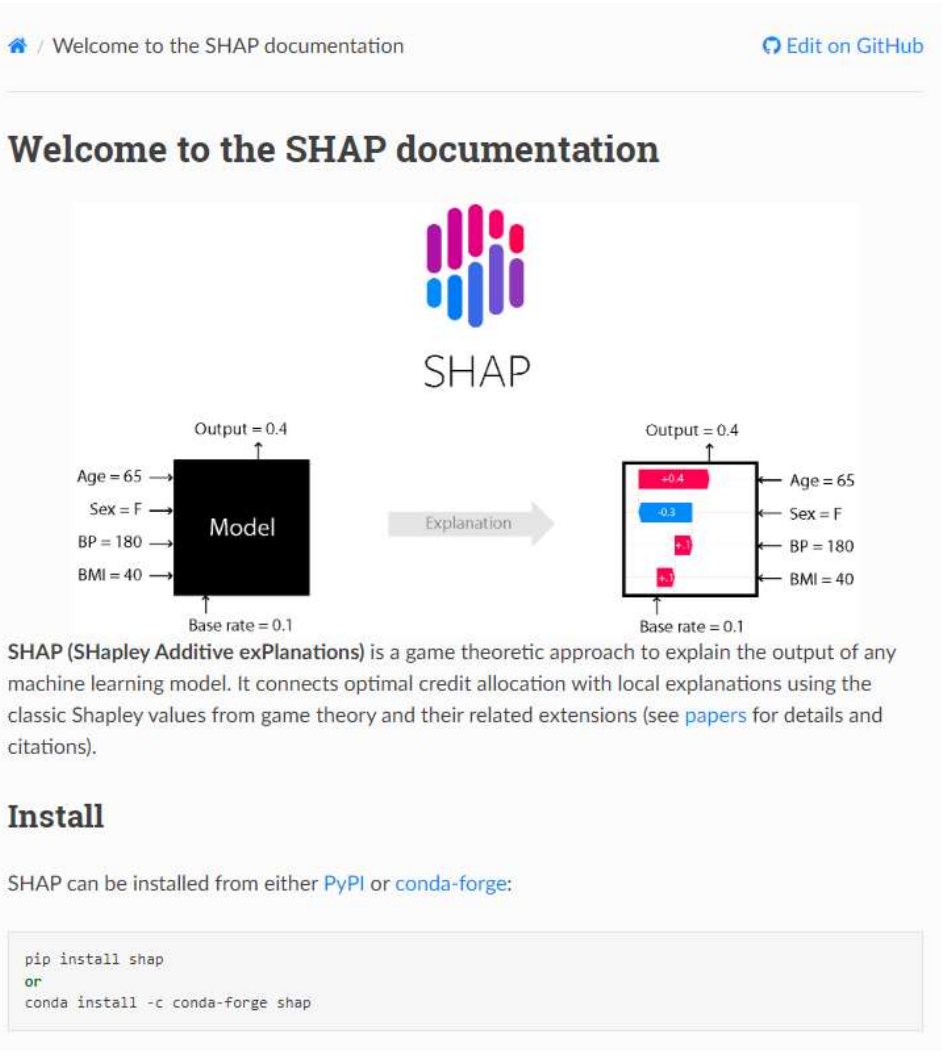
## 2. Post-hoc wrt the output of the training process

CONVENTIONAL MACHINE LEARNING
DEEP LEARNING

SHAP: Shapley Additive Explanations

https://shap.readthedocs.io/en/latest/

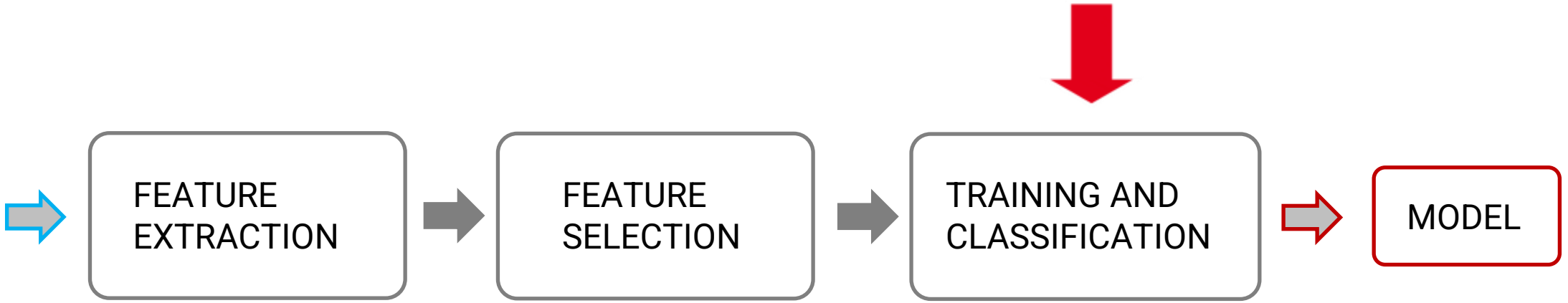


# Improve model explainability

## 3. Embedded into the classifier

CONVENTIONAL MACHINE LEARNING



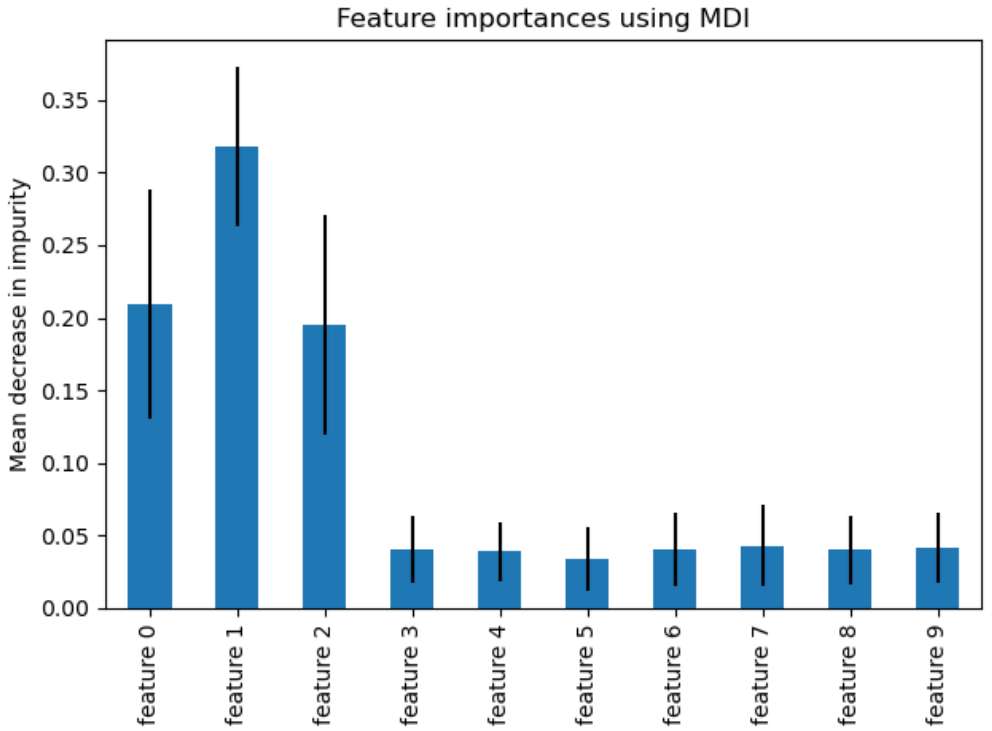
# Improve model explainability

## 3. Embedded into the classifier

CONVENTIONAL MACHINE LEARNING

### RANDOM-FOREST FEATURE IMPORTANCE

- Compute the decrease in Gini-impurity given by a feature at a given node
- Compute the mean-decrease of each feature for a given tree
- Compute the mean-decrease of each feature across all trained trees in the random forest to get MDI feature importance



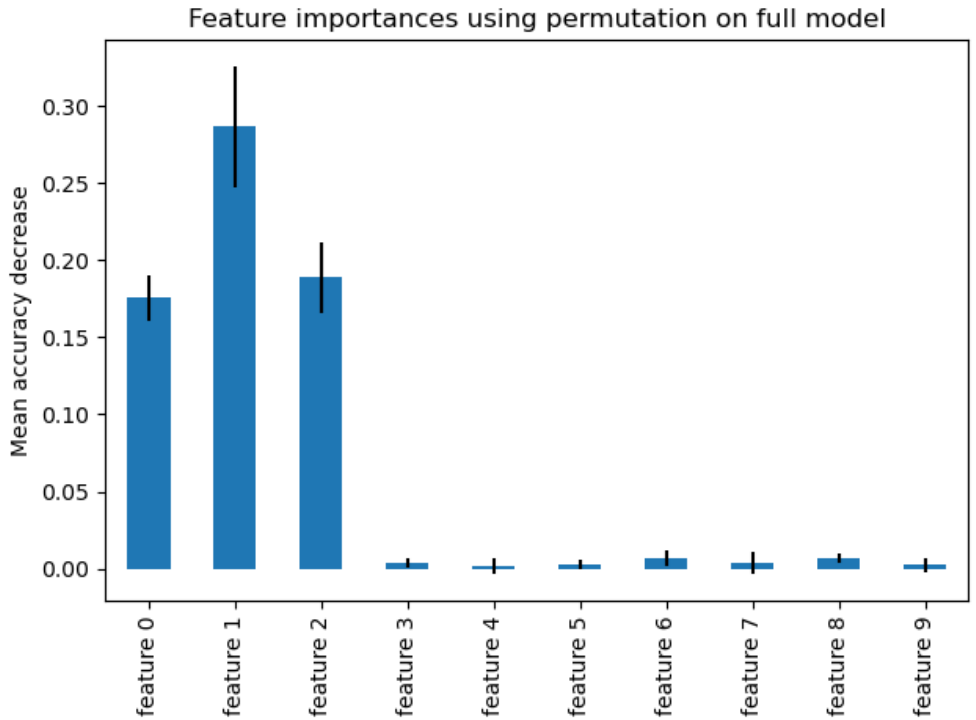
# Improve model explainability

## 3. Embedded into the classifier

CONVENTIONAL MACHINE LEARNING

RANDOM-FOREST FEATURE
IMPORTANCE BASED ON PERMUTATIONS

- Compute the feature importance on permuted out-of-bag (OOB) samples based on mean decrease in the accuracy (or other metrics)
- The highest the performance decrease, the highest the importance of a feature

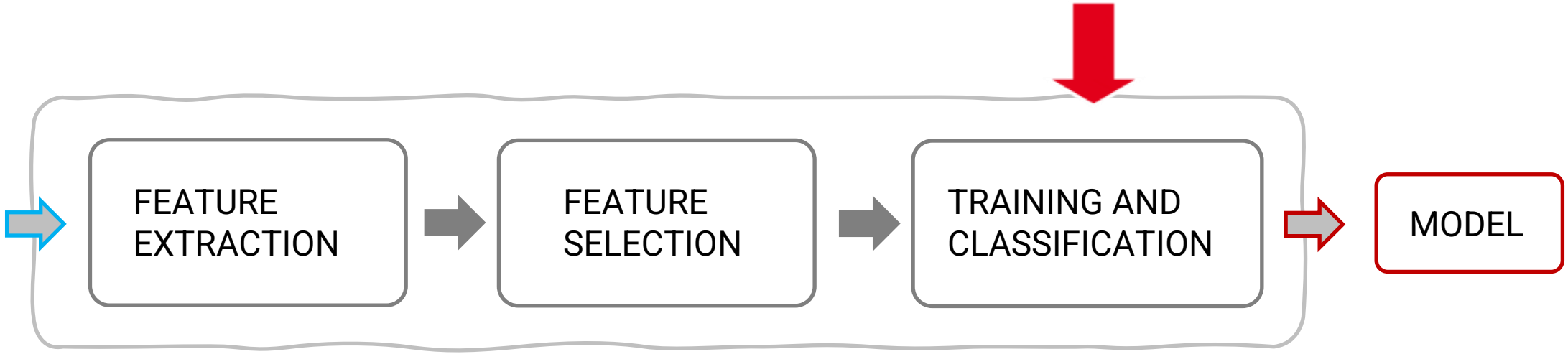


# Improve model explainability

## 3. Embedded into the classifier

DEEP LEARNING

BASED ON DEEPNETS LAYER INFORMATION



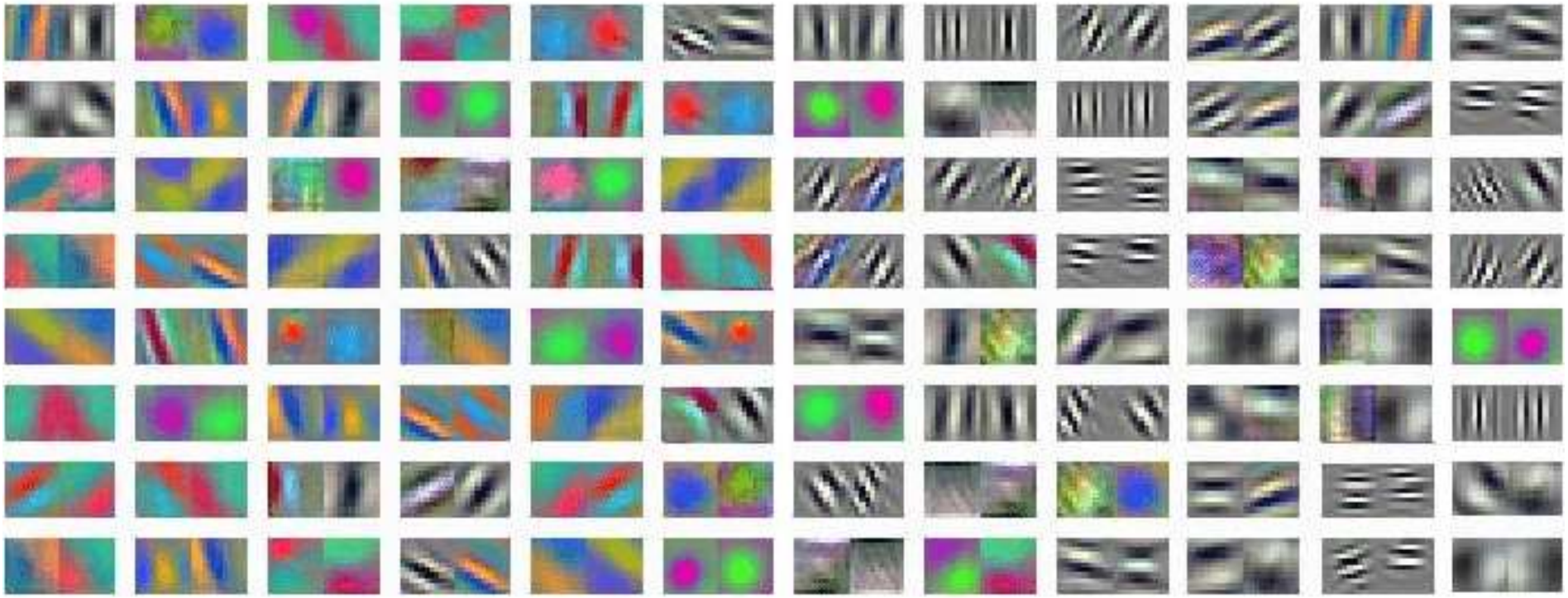
# Improve model explainability

DEEP LEARNING

## 3. Embedded into the classifier

BASED ON DEEPNETS LAYER
INFORMATION -> WEIGHT FILTERS

ALEXNET CONV1 LEARNED FILTERS

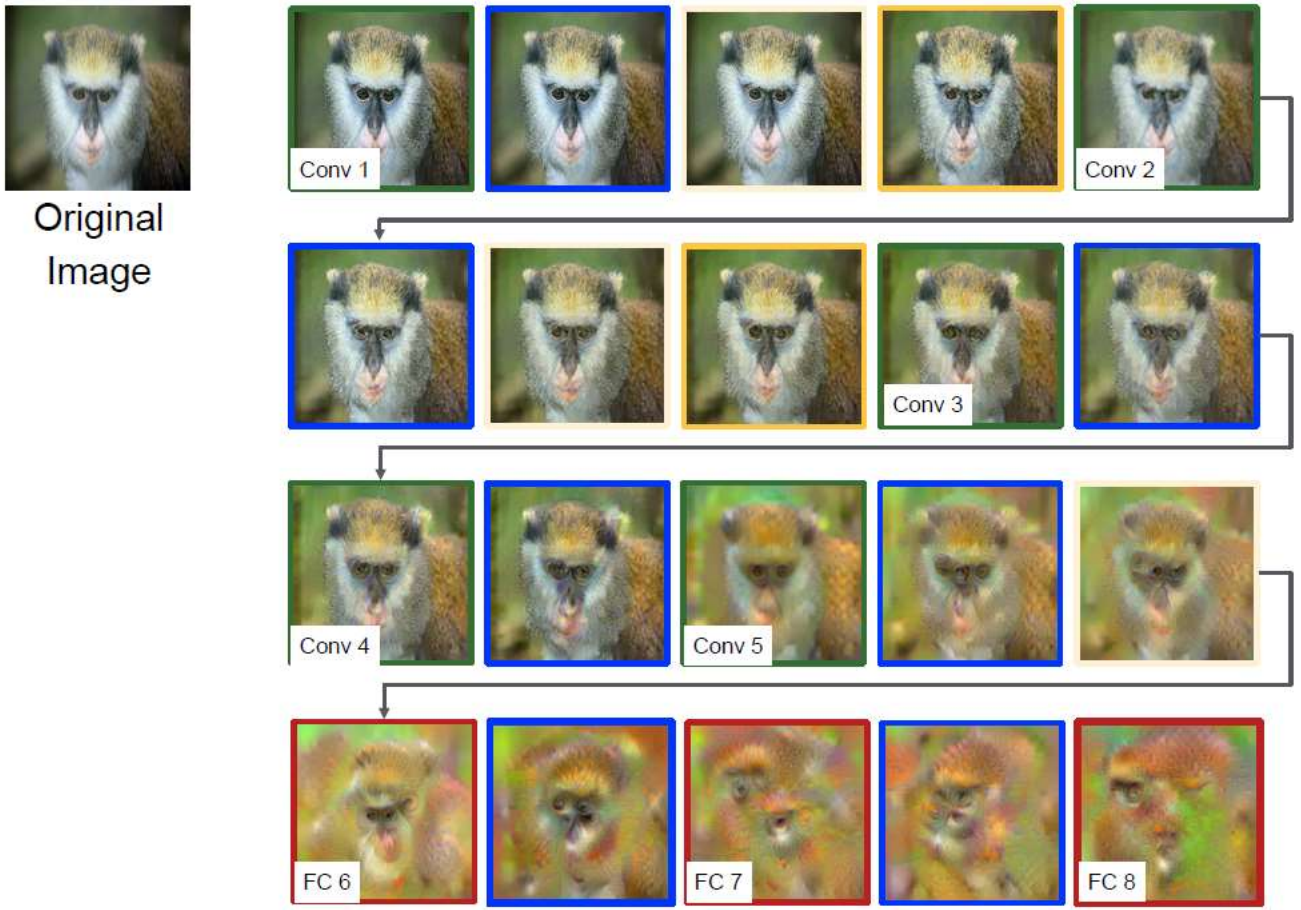


# Improve model explainability

## 3. Embedded into the classifier

DEEP LEARNING

BASED ON DEEPNETS LAYER
INFORMATION -> TRANSFORMED INPUT



# Improve model explainability

## 3. Embedded into the classifier

DEEP LEARNING

**Table 1.** Interpretability Methods to Explain Deep Learning Models.

| Ref | Tool | Category | Local vs. Global | Model Specific vs. Model Agnostic | Data Type | Citations/ Year | Year |
|---|---|---|---|---|---|---|---|
| [32] | DeepExplain<br>iNNvestigate<br>tf-explain | PH | L | Specific | img | 1548.3 | 2014 |
| [35] | Grad-CAM<br>tf-explain | PH | L | Specific | img | 797.8 | 2017 |
| [34] | CAM | PH | L | Specific | img | 607.8 | 2016 |
| [31] | iNNvestigate | PH | L | Specific | img | 365.3 | 2014 |
| [23] | DeepExplain<br>iNNvestigate<br>tf-explain | PH | L | Specific | img | 278.3 | 2013 |
| [27] | DeepExplain<br>iNNvestigate<br>Integrated Gradients<br>tf-explain<br>alibi<br>Skater | PH | L | Specific | img<br>txt<br>tab | 247 | 2017 |
| [40] | Deep Visualization Toolbox | PH | L | Specific | img | 221.7 | 2( |
| [37] | DeepExplain<br>iNNvestigate<br>The LRP Toolbox<br>Skater | PH | L | Specific | img<br>txt | 217.8 | 2( |
| [29] | DeepExplain<br>DeepLift<br>iNNvestigate<br>tf-explain<br>Skater | PH | L | Specific | img | 211.5 | 2017 |
| [41] | iNNvestigate | PH | L | Specific | img | 131.5 | 2017 |
| [38] | iNNvestigate<br>tf-explain | PH | L | Specific | img | 113.3 | 2017 |
| [42] | tcav | PH | L | Specific | img | 95 | 2018 |
| [43] | rationale | PH | L | Specific | txt | 81.4 | 2016 |
| [36] | Grad-CAM++ | PH | L | Specific | img | 81 | 2018 |
| [39] | RISE | PH | L | Specific | img | 43.3 | 2018 |
| [44] | iNNvestigate | PH | L | Specific | img | 41.8 | 2017 |

# Improve model explainability

## 3. Embedded into the classifier

DEEP LEARNING

GradCAM

"Used to indicate the discriminative regions of an image used by a CNN to identify the category of the image"

Grad-CAM uses the gradient information flowing into the last convolutional layer of the model to obtain localization map and understand the importance of each pixel of the input image for a specific class. Let's assume, $L^c_{Grad-CAM} \in \mathbb{R}^{u \times v}$ represents the localization map with width $u$ and height $v$ for class $c$. To calculate $L^c_{Grad-CAM}$, the gradient of the score for class $c$ (before softmax), with respect to feature map $k$, $A^k$, of the last convolutional layer is calculated and global average pooled to obtain neuron importance weight, $\alpha^c_k$:

$$\alpha^c_k = \frac{1}{N} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}$$

Furthermore, a weighted combination of forward activation maps followed by ReLU is obtained:

$$M = \text{ReLU}\left(\sum_k \alpha^c_k A^k\right).$$

This results in a coarse heatmap of the same size as the convolutional feature maps. Using ReLU allows us to capture features that have positive influence on the class of interest, i.e. pixels whose intensity should be increased in order to increase $y^c$. Negative pixels are likely to belong to other categories in the image. Without ReLU, localization maps sometimes highlight more than just the desired class and achieve lower localization performance.

https://github.com/nejlag/Grad-CAM

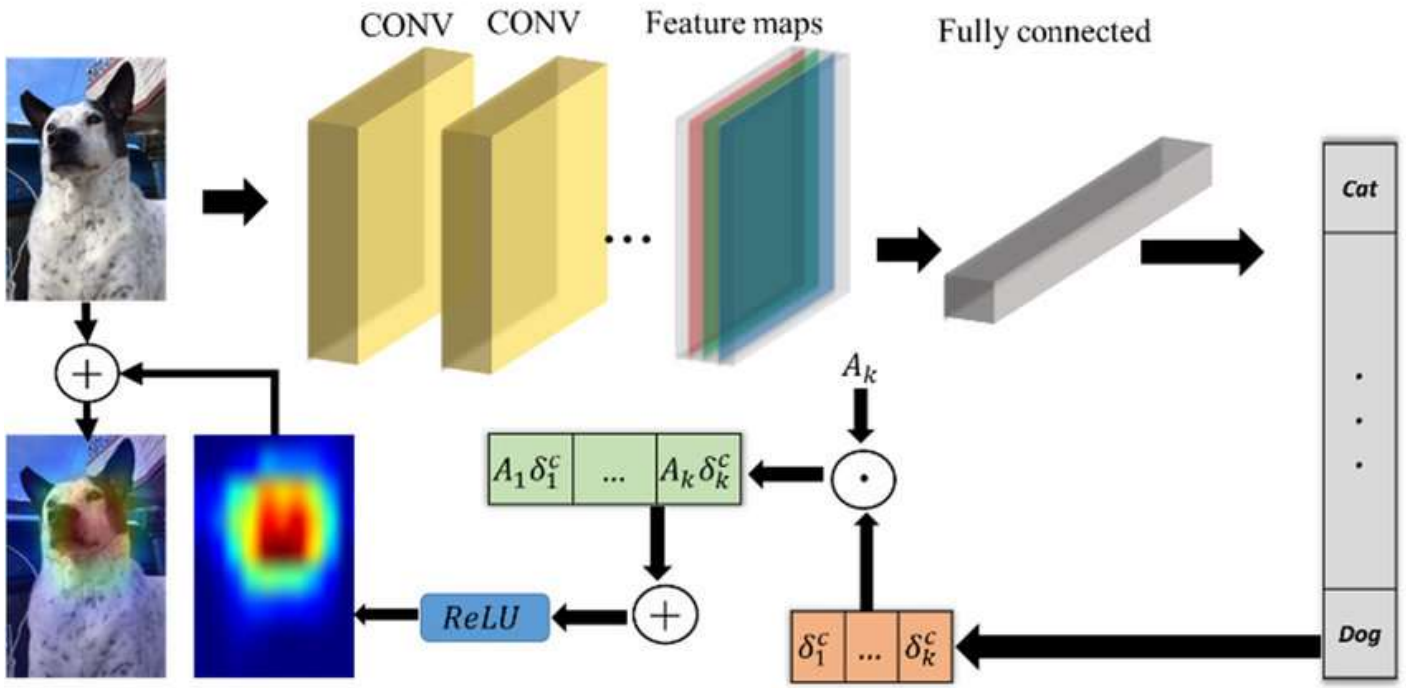
# Improve model explainability

DEEP LEARNING

## 3. Embedded into the classifier

GradCAM

"Used to indicate the discriminative regions of an image used by a CNN to identify the category of the image"




GRAD-CAM HEATMAP

# Improve model explainability

DEEP LEARNING

## 3. Embedded into the classifier

GradCAM "Used to indicate the discriminative regions of an image used by a CNN to identify the category of the image"