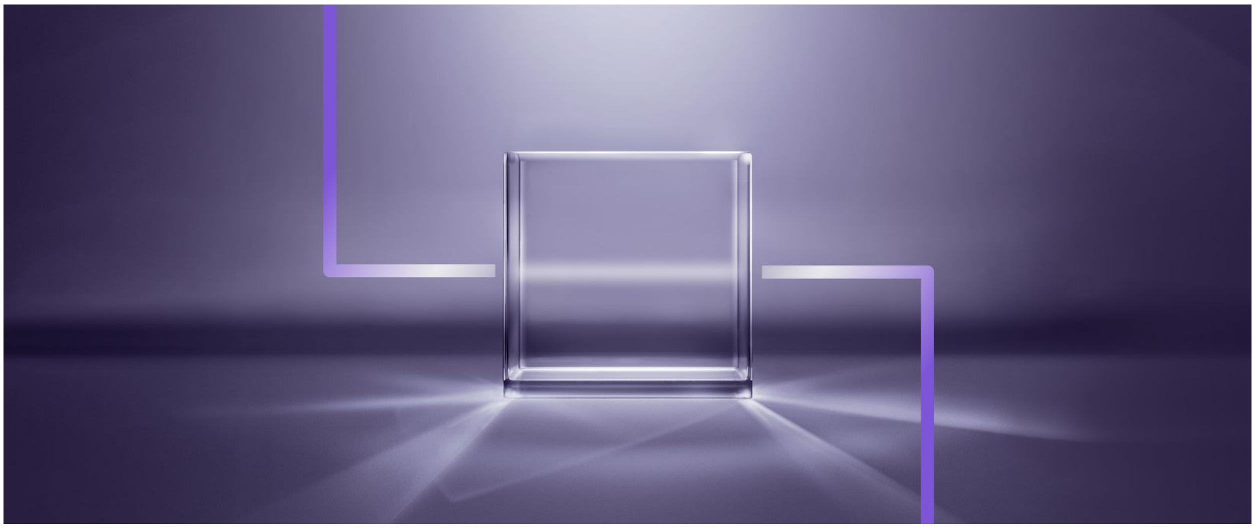
Appendix: Explainable AI Software Packages and Toolkits



This appendix is restricted to post-hoc explainability methods, as they are the main focus of the XAI community. Many of the most well-known methods have a publicly available implementation. However, these implementations are often only intended for reproducing results in the original paper, and not necessarily meant for deployment in a production environment.

A snapshot of common post-hoc explainability methodologies characterized according to the FATE taxonomy proposed earlier in this series

Method	Target Type/ Scope/ Complexity	Drivers	Explanation family	Estimator Applicability/ Mechanism	Comment	Code
LIME	Functional/ Local/ Simple label and numeric	Input features	Importance scores	Agnostic/ Perturbation	See also extension to local decision rules	Yes
SHAP	Functional/ Local/ Simple label	Input features	Importance scores	Agnostic and specific[1]/ Perturbation	See also extensions for graph-structured data and deep	Yes

[1] Shapley values could be estimated for a black-box model. However, specific model architectures, e.g. tree ensembles, could lend themselves to more efficient estimation approaches.

	and numeric				networks	
Feature Visualization	Mechanistic/ Global/ Individual neurons or a layer	Input features	Importance scores	Specific to neural networks/ Activation optimization	See also extension to activation atlas	Yes
TCAV	Mechanistic/ Global/ Simple label	User defined concepts	Importance scores	Specific to neural networks/ Backward prop	See also extension to discover the concepts	Yes
LRP	Functional/ Local/ Simple label and numeric	Input features	Importance scores	Specific to neural networks/ Backward prop	See also extension to deep taylor decomposition	Yes
Influence Functions	Functional/ Local/ NA[2]	Training samples	Importance scores	Agnostic/ Perturbation	See also extensions to set-wise influence and its applicability study	No
PDP	Functional/ Global/ Simple label and numeric	Input features	Dependency plot	Agnostic/ Perturbation	See also extension to interactive dependency plots and related work on partial importance	Yes
ICE	Functional/ Local/ Simple label and numeric	Input features	Dependency plot	Agnostic/ Perturbation	See also extension to individual conditional importance	Yes

[2] The influence function score estimates impact of a training sample on a model's loss for a given test sample regardless of its output complexity

BETA	Functional/ Local/ Simple label and numeric	Input features	Decision set	Agnostic/ Proxy		No
DeepLift	Functional/ Local/ Simple label and numeric	Input features	Importance scores	Specific to neural networks/ Backward prop		Yes
Grad-CAM	Functional/ Local/ Simple label and numeric	Input features	Importance scores	Specific to neural networks/ Backward prop	See also extension to spatio-temporal data using Grad-CAM++	Yes
DeepRED	Functional/ Global/ Simple label and numeric	Input features	Decision tree	Specific to neural networks/ Proxy		No
GAN Lab	Mechanistic/ Global/ 2D distributions	Training samples	Custom visualization	Specific to GANs/ NA		Yes
SOCRAT	Functional/ Locall/ Sequence of labels	(Sequence of) Input features	Importance scores	Agnostic/ Perturbation		No

In addition, there are a number of software packages that contain implementation of several explainability methods. The majority of these packages are dedicated to explaining neural network models developed with the TensorFlow and Keras platforms. In addition, almost all of them, except for the IML, have a Python interface.

Overview of the off-the-shelf explainability software packages

Software package	Explainability methods included	Platform/Interface	Comment
Skater	LIME, PDP, LRP, IG, Bayesian rule lists, Tree surrogates	TensorFlow and Keras/ Python	Provides some (local and global) post-hoc and modelling explainability methods

DeepExplain	Saliency maps, Gradient * Input, IG, DeepLIFT. LRP, Occlusion, Shapley value sampling	TensorFlow and Keras/ Python	Provides a set of state-of-the-art gradient and perturbation-based (feature) attribution methods
ELI5	Global feature attribution through permutation importance, LIME	Scikit-learn and XGBoost and LightGBM and lightning/ Python	Provides local and global feature attribution explanation support for several ML frameworks
iNNvestigate	Saliency maps, SmoothGrad, deconvnet, Guided backprop, PatternNet, Gradient * Input, LRP, IG, DeepLIFT	TensorFlow and Keras/ Python	Provides a comprehensive set of backward propagation based methods for explaining neural networks
Keras-viz	Activation maximization, Saliency maps, Class activation maps	TensorFlow and Keras/ Python	A high-level toolkit for visualizing and debugging neural networks
Lucid	Feature visualization, Saliency maps, Activation grids, Channel attribution, Neuron interaction grids, Class activation atlas	TensorFlow/ Python	A collection of infrastructure and tools mainly for obtaining mechanistic explanations of neural networks
IML	Global feature attribution through permutation importance, PDP, ICE, Accumulated local effects, Tree surrogate, LIME, SHAP	R	Provides a set of model-agnostic explainability methods
What-If	PDP, Counterfactual explanations	TensorFlow/ Python	An interactive visual interface designed to probe neural network models
TensorWatch	LIME, Grad-CAM, Gradient * Input, DeepLIFT, SmoothGrad, Guided Backprop	PyTorch and TensorFlow/Python	Provides a set of state-of-the-art, mainly backpropagation-based, (feature) attribution methods

In general, the choices for the off-the-shelf and mature explainability toolkits are relatively limited at the moment. This is particularly the case in contrast to other areas of AI such as computer vision and NLP. We hope this shortcoming to be alleviated in the future due to the ever-increasing and high demand for explainability in various application domains of AI.