AI Explainability (XAI) Open Source Tools

|  |  |  |  |
| --- | --- | --- | --- |
| Name and link | Short description | Supported XAI methods | Supported DL frameworks |
| [DeepExplain](https://github.com/marcoancona/DeepExplain) | unified framework for state-of-the-art gradient and perturbation-based attribution methods | **Gradient-based attribution methods**   * [**Saliency maps**](https://arxiv.org/abs/1312.6034) * [**Gradient \* Input**](https://arxiv.org/abs/1605.01713) * [**Integrated Gradients**](https://arxiv.org/abs/1703.01365) * [**DeepLIFT**](https://arxiv.org/abs/1704.02685), in its first variant with Rescale rule (\*) * [**ε-LRP**](http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0130140) (\*) | **Tensorflow** as well as **Keras** with Tensorflow backend. Support for **PyTorch** is planned. |
| [iNNvestigate](https://github.com/albermax/innvestigate) | common interface and out-of-the-box implementation for many analysis methods |  function: **gradient, smoothgrad:** [SmoothGrad](https://arxiv.org/abs/1706.03825)   signal: **deconvnet:** [DeConvNet](https://arxiv.org/abs/1311.2901), **guided:** [Guided BackProp](https://arxiv.org/abs/1412.6806), **pattern.net:** [PatternNet](https://arxiv.org/abs/1705.05598)   attribution: **input\_t\_gradient, deep\_taylor[.bounded]:** [DeepTaylor](https://www.sciencedirect.com/science/article/pii/S0031320316303582?via%3Dihub), **pattern.attribution:** [PatternAttribution](https://arxiv.org/abs/1705.05598), **lrp.\*:** [LRP](http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0130140), **integrated\_gradients:** [IntegratedGradients](https://arxiv.org/abs/1703.01365), **deeplift.wrapper:** [DeepLIFT (wrapper around original code, slower)](http://proceedings.mlr.press/v70/shrikumar17a.html) | based on Keras and therefore requires a supported [Keras-backend](https://keras.io/backend/) |
| [SHAP](https://github.com/slundberg/shap) | game theoretic approach to explain the output of any machine learning model | * TreeExplainer (TE) * DeepExplainer (DE): SHAP and DeepLIFT * GradientExplainer (GE) * KernalExplainer (KE) | * TE: XGBoost, LightGBM, CatBoost, scikit-learn and pyspark * DE: Tensorflow and Keras * GE: TF, Keras and PyTorch * KE: skikit-learn |
| [ELI5](https://github.com/TeamHG-Memex/eli5)  [Documentation](https://eli5.readthedocs.io/en/latest/) | debugging/inspecting machine learning classifiers and explaining their predictions | for inspecting black-box models:   * [TextExplainer](https://eli5.readthedocs.io/en/latest/tutorials/black-box-text-classifiers.html#lime-tutorial) allows to explain predictions of any text classifier using [LIME](https://eli5.readthedocs.io/en/latest/blackbox/lime.html#eli5-lime) * [Permutation Importance](https://eli5.readthedocs.io/en/latest/blackbox/permutation_importance.html#eli5-permutation-importance) method can be used to compute feature importance for black box estimators.   Two ways to look at a classification or a regression model:   1. inspect model parameters 2. inspect an individual prediction of a model | [scikit-learn](https://eli5.readthedocs.io/en/latest/libraries/sklearn.html#library-scikit-learn); [XGBoost](https://eli5.readthedocs.io/en/latest/libraries/xgboost.html#library-xgboost); [LightGBM](https://eli5.readthedocs.io/en/latest/libraries/lightgbm.html#library-lightgbm); [CatBoost](https://eli5.readthedocs.io/en/latest/libraries/catboost.html#library-catboost); [lightning](https://eli5.readthedocs.io/en/latest/libraries/lightning.html#library-lightning); [sklearn-crfsuite](https://eli5.readthedocs.io/en/latest/libraries/sklearn_crfsuite.html#library-sklearn-crfsuite); [Keras](https://eli5.readthedocs.io/en/latest/libraries/keras.html#library-keras) |
| [Skater](https://github.com/oracle/Skater) | unified framework to enable model Interpretation for all forms of model to help one build an Interpretable machine learning system |  | [scikit-learn](https://eli5.readthedocs.io/en/latest/libraries/sklearn.html#library-scikit-learn) only? |
| [Yellowbrick](https://www.scikit-yb.org/en/latest/)  [Documentation](https://www.scikit-yb.org/en/latest/) | Visual analysis and diagnostic tools to facilitate machine learning model selection | Not for DL? | [scikit-learn](https://eli5.readthedocs.io/en/latest/libraries/sklearn.html#library-scikit-learn) only |
| [Lucid](https://github.com/tensorflow/lucid) | A collection of infrastructure and tools for research in neural network interpretability | Feature visualization; Attribution. See [**Feature Visualization**](https://distill.pub/2017/feature-visualization/)  Very rich library! | Needs saved inference graph:  TF, Keras |
| [Captum](https://github.com/pytorch/captum)  [Main page](https://captum.ai/) | Model interpretability and understanding for PyTorch | IntegratedGradients; NeuronIntegratedGradient; InputXGradient; SmoothGrad; NoiseTunnel; NeuronConductance; LayerConductance; DeepLift; NeuronIntegratedGradients; GradientShap, DeepLiftShap; InternalInfluence; Saliency, NeuronGradient; GradCAM, Guided GradCAM; Deconvnet; Guided Backpropagation | PyTorch. It has quick integration for models built with domain-specific libraries such as torchvision, torchtext, and others. |