Ukrainian Catholic University

Faculty of Applied Sciences

Data Science Master Programme

Interpretability libraries comparison

Responsible Data Science project report

Authors:
Konstantyn Ovchynnikov
Dmitrii Glushko
Yuriy Lizak
Olena Shevchenko

May 14, 2019

1 Introduction

Today we are surrounded by applications of Machine learning. It helps us quickly write letters, corrects grammar, recommends movies and products, unlocks our phone. We are getting used to it; some of us do not really care how such algorithms make decisions, others do. Though some applications are harmless, meaning a mistake will not have serious consequences (e.g., a movie recommender system), others might heavily influence our life by making a mistake (e.g., autonomous driving). Generally, we might want to know 'what was predicted?' and 'why the prediction was made.' To answer the second question, the interpretability comes in handy.
According to *Miller*[4]: **Interpretability is the degree to which a human can understand the cause of a decision.** Another definition[1] is **Interpretability is the degree to which a human can consistently predict the models result.** So the interpretability is a measure of an answer to 'why' question, is it only? Various teams around the world work on the understanding of model decision making to make black box models more transparent. Different algorithms had been developed and open sourced. In this work, we want to explore the functionality of popular libraries and analyze their interpretability power.

2 Related work

The easiest way to achieve interpretability is to use only a subset of algorithms that create interpretable models. Linear regression, logistic regression, and the decision tree are commonly used interpretable models.[5] Such models are well described in terms of their explanatory power, so we will not pay additional attention to these models.
With the increasing popularity of deep learning, various teams work of mechanisms to achieve model agnostic interpretability[6].The team from the University of Washington *Ribeiro et al.*[7] presented the local interpretable model-agnostic explanations (LIME) that work with any classifier.

3 Problems

We want to investigate the interpretability systems based on two different tasks - Credit Scoring and Image Classification. Link to the Github project[2]

3.1 Credit Scoring

A credit score is a numerical expression based on a level analysis of a person's credit files, to represent the creditworthiness of an individual. Traditionally, a credit score was primarily based on credit report information typically sourced from credit bureaus.
About the Dataset: Lending Club Dataset
Lending Club is a US peer-to-peer lending company, headquartered in San Francisco,

California. Lending Club is the world's largest peer-to-peer lending platform. The company states that $15.98 billion in loans had been originated through its platform up to 31 December 2015.

4 Model-agnostic systems

In their work *Ribeiro et al.*[6] describes desirable properties of model-agnostic explanation systems. Authors point out three such features:

- Model flexibility - can work with both linear models and deep networks
- Explanation flexibility - not limited to a certain form of explanation
- Representation flexibility - should be able to use a different feature representation as the model being explained.

4.1 SHAP

SHAP (SHapley Additive exPlanations) presented by *Lundberg and Lee* [3] at NIPS 2017 as a unified way to interpret model predictions. SHAP is a unified approach to explain the output of any machine learning model. SHAP connects game theory with local explanations, representing the only possible consistent and locally accurate additive feature attribution method based on expectations.
We will use Python Shap library on two tasks (credit scoring, image classification), to understand It's abilities, advantages and disadvantages.
Shap has model flexibility - this library can work with Tree ensembles, Linear models, and Deep Learning models also. By the time, Shap is one of the most famous libraries, it has good explanation flexibility. There is an ability to look at single observation (or multiple) to understand, which features influenced the result in which way (higher/lower).
Shap API is straightforward.

```
# Explainer can work with any Python model
explainer = shap.KernelExplainer(model.predict_proba, subsample)
# Extracting shap values matrix
shap_values = explainer.shap_values(subsample)
```

Representation flexibility is important part of this research, so we was trying to use all visualization abilities of the library to get better understanding of prediction results and features influence.
First plot was used to get feature influence and which range of values created this influence. At this plot we can see, for example, how higher number of delinquincies impacted the model.

After looking at big picture, library gives the ability to look at single observation output and features influence.
Library has ability to build dependence plots. For example we can look at the most influencial features.

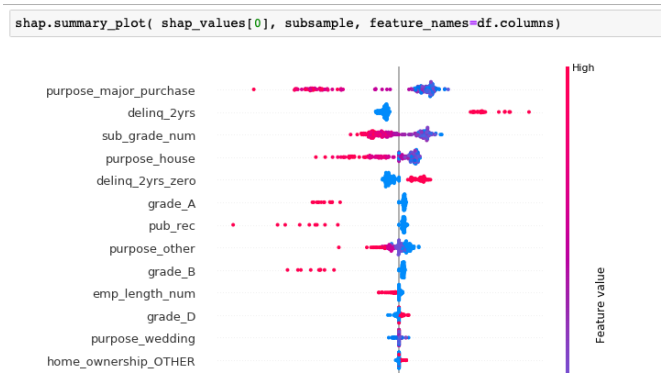

Figure 1: Summary plot by all features in credit scoring task.

Figure 2: Example of single observation explanation.

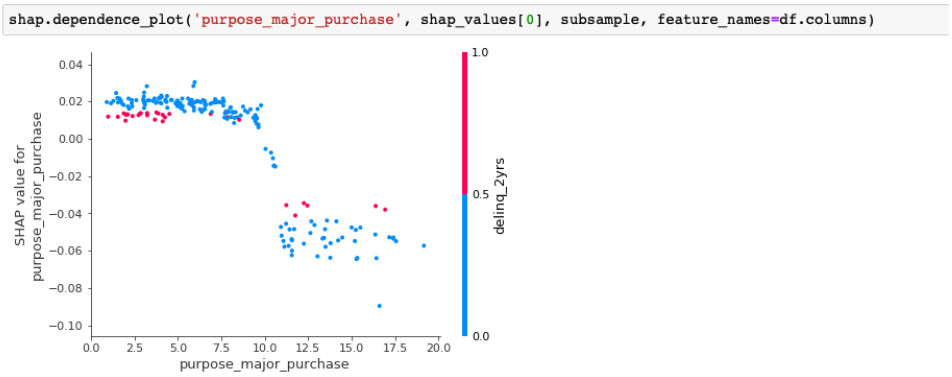

Figure 3: Example of dependence plot for scoring task.

4.2 SKATER

Skater is an open-source python library for model interpretation. The project started at the beginning of 2017 and is currently under active development in the beta phase.
Skater supports interpretation on two levels: global - using information from the whole dataset, local - using information about one prediction to approximate functions based on inputs and outputs. Global interpretation is supported by the next algorithms:

- Model agnostic Feature Importance
- Model agnostic Partial Dependence Plots
- Scalable Bayesian Rule Lists
- Tree Surrogates

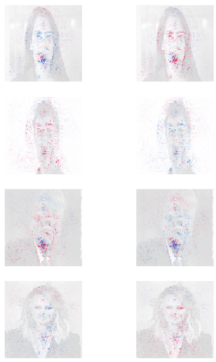

Figure 4: Deep learning model feature maps. Explains visually which features was used to make predictions.

Local interpretation is realized using:

- Local Interpretable Model Explanation (LIME)
- Layer-wise Relevance Propagation (e-LRP): image
- Integrated Gradient: image and text
- Scalable Bayesian Rule Lists
- Tree Surrogates

The framework supports the most popular Python libraries. It can work with models from scikit-learn, XGBoost, Keras, Tensorflow.
In this project, Skater was applied to interpret Gender Classification model based on the VGGNet Architecture. 3 methods were used to explain the model:

- Epsilon-LRP
- Integrated Gradient
- Occlusion

Layer-wise Relevance Propagation is a method that identifies important pixels by running a backward pass in the neural network. It starts from the output and weight the neurons that contribute the most. How to implement it can be found in [9].

Integrated Gradients is a method to find the relation between a deep model's prediction and its features. Like in our example it is a relation to the pixels. The method is based on the paper [8].

Occlusion as an algorithm that computes direct relevance of the input features by removing or masking them, running a forward pass and measuring the difference between old and new output. Explained in detail by [10].

Corresponding code of using Skater on VGGNet can be found in *interpretability/skater_library.ipynb* ipython notebook.

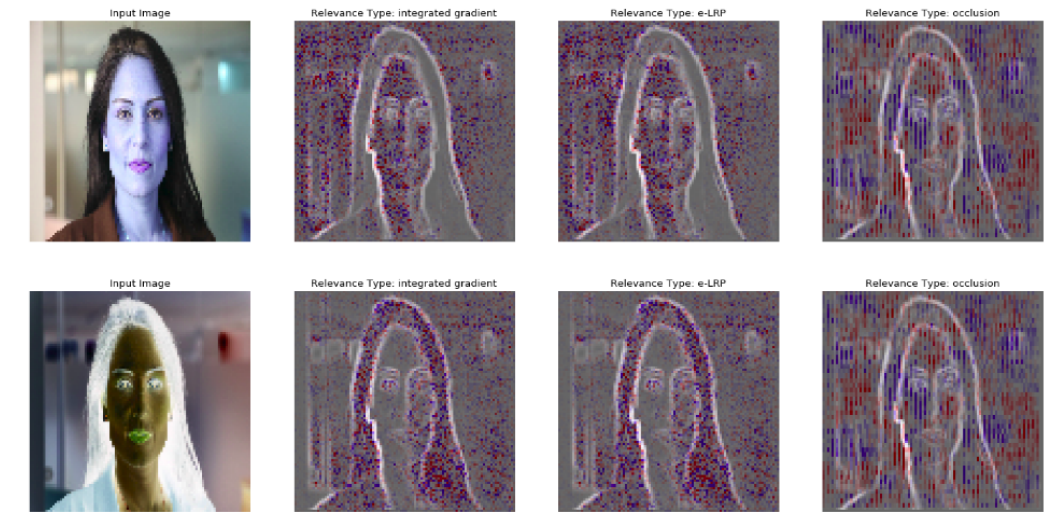

Figure 5: Local VGGNet model interpretability with Skater

All three methods are trying to find out the importance of the pixels for the specific prediction. Looking at the result with the woman example we can see that the current model has a strong relation between hair pixels and the prediction. Both Integrated Gradient and Epsilon-LRP show almost the same result. In contrast, looking at the occlusion method, it is hard to get a good interpretation of the model. So using Skater with local interpretation approaches it is easy to add some interpretability to CNN networks even having few examples.

5 Conclusions

We've tried several libraries with two different (deep learning, tree) models. Most of them had model flexibility so we can work with both models. Shap library has more flexibility in representation features and visualization techniques. It is more supported by the community, too. It's a disadvantage - library works slow.
Skater has also flexibility in the available functionality. The framework has different interpretation algorithms and methods available for different models including deep neural networks like CNN. But it's still in the beta phase and the functionality sometimes looks raw. However, it is in active development. Summarising the frameworks we analyzed, there are a lot of different algorithms and methods that are already implemented and work nice with existing libraries like sci-kit-learn, Keras, Tensorflow making live of the data scientist easier and giving time to stay focused on the domain part of the problem. But from another point of view, all these techniques are pretty generic so far and can't cover all the possible cases, meaning that the data scientist can always encounter the situation where the specific solution must be applied with using the specific domain knowledge.

6 Peer Review

Work done:

- Dmitrii Glushko: Gender Classification model preparation, Skater library analysis

and report section

- Konstantyn Ovchynnikov: Credit Scoring model preparation, SHAP library analysis and report section
- Yuriy Lizak: LIME library analysis
- Olena Shevchenko: report skeleton, introduction, related work

References

[1] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. "Examples are not enough, learn to criticize! Criticism for Interpretability". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee et al. Curran Associates, Inc., 2016, pp. 2280–2288. URL: http://papers.nips.cc/paper/6300-examples-are-not-enough-learn-to-criticize-criticism-for-interpretability.pdf.

[2] Ovchynnikov Konstantyn et al. "Interpretability libraries comparison Github Project". In: 2019. URL: https://github.com/costefan/resp-ds-2019.

[3] Lundberg, Scott M, and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[4] Tim Miller. "Explanation in Artificial Intelligence: Insights from the Social Sciences". In: *CoRR* abs/1706.07269 (2017). arXiv: 1706.07269. URL: http://arxiv.org/abs/1706.07269.

[5] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/. 2019.

[6] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Model-agnostic interpretability of machine learning". In: *arXiv preprint arXiv:1606.05386* (2016).

[7] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *CoRR* abs/1602.04938 (2016). arXiv: 1602.04938. URL: http://arxiv.org/abs/1602.04938.

[8] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: 2017. URL: https://arxiv.org/abs/1703.01365.

[9] "Tutorial: Implementing Deep Taylor Decomposition / LRP". In: URL: http://www.heatmapping.org/tutorial/.

[10] Matthew D. Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Networks". In: 2014.