P. Mishra, *Explainable AI Recipes*
https://doi.org/10.1007/978-1-4842-9029-3_1

1. Introducing Explainability and Setting Up Your Development Environment

Pradeepta Mishra[1]
(1) Bangalore, Karnataka, India

Industries in which artificial intelligence has been applied include banking, financial services, insurance, healthcare, manufacturing, retail, and pharmaceutical. There are regulatory requirements in some of these industries where model explainability is required. Artificial intelligence involves classifying objects, recognizing objects to detect fraud, and so forth. Every learning system requires three things: input data, processing, and an output. If the performance of any learning system improves over time by learning from new examples or data, it is called a *machine learning system*. When the number of features for a machine learning task increases or the volume of data increases, it takes a lot of time to apply machine learning techniques. That's when deep learning techniques are used.

Figure 1-1 represents the relationships between artificial intelligence, machine learning, and deep learning.

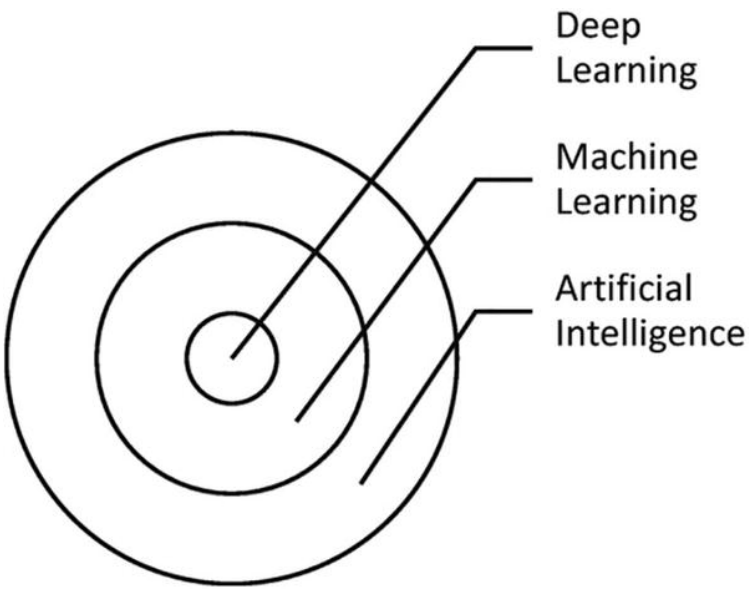

Figure 1-1 Relationships among ML, DL, and AI

After preprocessing and feature creation, you can observe hundreds of thousands of features that need to be computed to produce output. If we train a machine learning supervised model, it will take significant time to produce the model object. To achieve scalability in this task, we need deep learning algorithms, such as a recurrent neural network. This is how artificial intelligence is connected to deep learning and machine learning.

In the classical predictive modeling scenario, a function is identified, and the input data is usually fit to the function to produce the output, where the function is usually predetermined. In a modern predictive modeling scenario, the input data and output are both shown to a group of functions, and the machine identifies the best function that approximates well to the output given a particular set of input. There is a need to explain the output of a machine learning and deep learning model in performing regression- and classification-related tasks. These are the reasons why explainability is required:

- *Trust*: To gain users' trust on the predicted output
- *Reliability*: To make the user rely on the predicted output
- *Regulatory*: To meet regulatory and compliance requirements
- *Adoption*: To increase AI adoption among the users
- *Fairness*: To remove any kind of discrimination in prediction
- *Accountability*: To establish ownership of the predictions

There are various ways that explainability can be achieved using statistical properties, probabilistic properties and associations, and causality among the features. Broadly, the explanations of the models can be classified into two categories, global explanations and local explanations. The objective of local explanation is to understand the inference generated for one sample at a time by comparing the nearest possible data point; global explanation provides an idea about the overall model behavior.

The goal of this chapter is to introduce how to install various explainability libraries and interpret the results generated by those explainability libraries.

Recipe 1-1. SHAP Installation

Problem

You want to install the SHAP (shapely additive explanations) library.

Solution

The solution to this problem is to use the simple `pip` or `conda` option.

How It Works

Let's take a look at the following script examples. The SHAP Python library is based on a game theoretic approach that attempts to explain local and as well as global explanations.

```
pip install shap
```

or

```
conda install -c conda-forge shap
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simpl
Collecting shap
  Downloading shap-0.41.0-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (569 kB)
     |████████████████████████████████| 569 kB 8.0 MB/s
Requirement already satisfied: tqdm>4.25.0 in /usr/local/lib/python3.7/dist-packages (from shap)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from shap) (1.3
Collecting slicer==0.0.7
  Downloading slicer-0.0.7-py3-none-any.whl (14 kB)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.7/dist-packages (from shap)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from shap) (1.7
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist-packages (from shap
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from shap) (1.21
Requirement already satisfied: numba in /usr/local/lib/python3.7/dist-packages (from shap) (0.56
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.7/dist-packages (from sh
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: llvmlite<0.40,>=0.39.0dev0 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: setuptools<60 in /usr/local/lib/python3.7/dist-packages (from num
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importl
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pand
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-d
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from scik
Installing collected packages: slicer, shap
Successfully installed shap-0.41.0 slicer-0.0.7
```

Recipe 1-2. LIME Installation

Problem

You want to install the LIME Python library.

Solution

You can install the LIME library using `pip` or `conda`.

How It Works

Let's take a look at the following example script:

```
pip install lime
```

or

```
conda install -c conda-forge lime
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simpl
Collecting lime
  Downloading lime-0.2.0.1.tar.gz (275 kB)
     |████████████████████████████████| 275 kB 7.5 MB/s
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from lime)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from lime) (1.21
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from lime) (1.7
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from lime) (4.64
Requirement already satisfied: scikit-learn>=0.18 in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: scikit-image>=0.12 in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.7/dist-packages (from sci
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.7/dist-packages (fron
Requirement already satisfied: pillow!=7.1.0,!=7.1.1,>=4.3.0 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from sc
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (fron
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matp
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (fron
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-d
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from scik
Building wheels for collected packages: lime
```

```
  Building wheel for lime (setup.py) ... done
  Created wheel for lime: filename=lime-0.2.0.1-py3-none-any.whl size=283857 sha256=674ceb94cdcb
  Stored in directory: /root/.cache/pip/wheels/ca/cb/e5/ac701e12d365a08917bf4c6171c0961bc880a818
Successfully built lime
Installing collected packages: lime
Successfully installed lime-0.2.0.1
```

Recipe 1-3. SHAPASH Installation

Problem

You want to install SHAPASH.

Solution

If you want to use a combination of functions from both the LIME library and the SHAP library, then you can use the SHAPASH library. You just have to install it, which is simple.

How It Works

Let's take a look at the following code to install SHAPASH. This is not available on the Anaconda distribution; the only way to install it is by using `pip`.

```
pip install shapash
```

Recipe 1-4. ELI5 Installation

Problem

You want to install ELI5.

Solution

Since this is a Python library, you can use `pip`.

How It Works

Let's take a look at the following script:

```
pip install eli5
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simpl
Collecting eli5
  Downloading eli5-0.13.0.tar.gz (216 kB)
     |████████████████████████████████| 216 kB 6.9 MB/s
Requirement already satisfied: attrs>17.1.0 in /usr/local/lib/python3.7/dist-packages (from eli5
Collecting jinja2>=3.0.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
     |████████████████████████████████| 133 kB 42.7 MB/s
Requirement already satisfied: numpy>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from eli5
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from eli5) (1.7
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from eli5) (1.15.0
Requirement already satisfied: scikit-learn>=0.20 in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: graphviz in /usr/local/lib/python3.7/dist-packages (from eli5) (0
Requirement already satisfied: tabulate>=0.7.7 in /usr/local/lib/python3.7/dist-packages (from e
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.7/dist-packages (from j
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from scik
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (f
```

```
Building wheels for collected packages: eli5
  Building wheel for eli5 (setup.py) ... done
  Created wheel for eli5: filename=eli5-0.13.0-py2.py3-none-any.whl size=107748 sha256=3e02d416b
  Stored in directory: /root/.cache/pip/wheels/cc/3c/96/3ead31a8e6c20fc0f1a707fde2e05d49a80b1b4b
Successfully built eli5
Installing collected packages: jinja2, eli5
  Attempting uninstall: jinja2
    Found existing installation: Jinja2 2.11.3
    Uninstalling Jinja2-2.11.3:
      Successfully uninstalled Jinja2-2.11.3
ERROR: pip's dependency resolver does not currently take into account all the packages that are
flask 1.1.4 requires Jinja2<3.0,>=2.10.1, but you have jinja2 3.1.2 which is incompatible.
Successfully installed eli5-0.13.0 jinja2-3.1.2
```

Recipe 1-5. Skater Installation

Problem

You want to install Skater.

Solution

Skater is an open-source framework to enable model interpretation for various kinds of machine learning models. The Python-based Skater library provides both global and local interpretations and can be installed using `pip`.

How It Works

Let's take a look at the following script:

```
pip install skater
```

Recipe 1-6. Skope-rules Installation

Problem

You want to install Skopes-rule.

Solution

Skope-rules offers a trade-off between the interpretability of a decision tree and the modeling power of a random forest model. The solution is simple; you use the `pip` command.

How It Works

Let's take a look at the following code:

```
pip install skope-rules
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simpl
Collecting skope-rules
  Downloading skope_rules-1.0.1-py3-none-any.whl (14 kB)
Requirement already satisfied: numpy>=1.10.4 in /usr/local/lib/python3.7/dist-packages (from sko
Requirement already satisfied: scikit-learn>=0.17.1 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: pandas>=0.18.1 in /usr/local/lib/python3.7/dist-packages (from sk
Requirement already satisfied: scipy>=0.17.0 in /usr/local/lib/python3.7/dist-packages (from sko
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pand
```

```
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from scik
Installing collected packages: skope-rules
Successfully installed skope-rules-1.0.1
```

Recipe 1-7. Methods of Model Explainability

Problem

There are various libraries and many explanations for how to identify the right method for model explainability.

Solution

The explainability method depends on who is the consumer of the model output, if it is the business or senior management then the explainability should be very simple and plain English without any mathematical formula and if the consumer of explainability is data scientists and machine learning engineers then the explanations may include the mathematical formulas.

How It Works

The levels of transparency of the machine learning models can be categorized into three buckets, as shown in Figure 1-2.

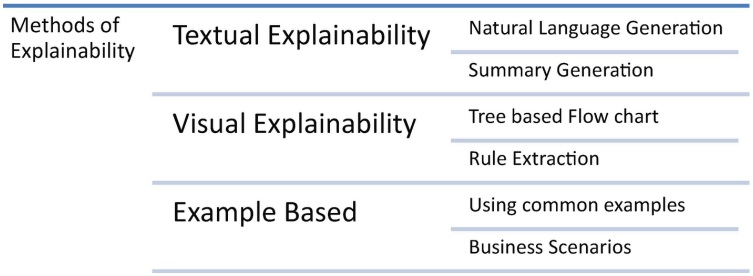

Figure 1-2 Methods of model explainability

Textual explanations require explaining the mathematical formula in plain English, which can help business users or senior management. The interpretations can be designed based on model type and model variant and can draw inferences from the model outcome. A template to draw inferences can be designed and mapped to the model types, and then the templates can be filled in using some natural language processing methods.

A visual explainability method can be used to generate charts, graphs such as dendrograms, or any other types of graphs that best explain the relationships. The tree-based methods use `if-else` conditions on the back end; hence, it is simple to show the causality and the relationship.

Using common examples and business scenarios from day-to-day operations and drawing parallels between them can also be useful.

Which method you should choose depends on the problem that needs to be solved and the consumer of the solution where the machine learning model is being used.

Conclusion

In various AI projects and initiatives, the machine learning models generate predictions. Usually, to trust the outcomes of a model, a detailed explanation is required. Since many people are not comfortable explaining the machine learning model outcomes, they cannot reason out the decisions of a model, and thereby AI adoption is restricted. Explainability is required from regulatory stand point as well as auditing and compliance point of view. In high-risk use cases such as medical imaging and object detection or pattern recognition, financial prediction and fraud detection, etc., explainability is required to explain the decisions of the machine learning model.

In this chapter, we set up the environment by installing various explainable AI libraries. Machine learning model interpretability and explainability are the key focuses of this book. We are going to use Python-based libraries, frameworks, methods, classes, and functions to explain the models.

In the next chapter, we are going to look at the linear models.