Sorry, our blog is currently experiencing some technical issues.

We are working hard to get it up and running as soon as possible.

Please visit us in a couple of days.

VISIT OUR HOMEPAGE →

March 25, 2020

Consulting
Digital Transformation
Project Management
Research and Development
Technology Law
IT Outsourcing
Life at Codete
Case Studies
Codete News
HR and Recruitment
Personal Development
Other
Software Development
Backend Development
DevOps
Frontend Development
Mobile Development
QA and Testing
UX Design
Web Development
Technology
Artificial Intelligence
Big Data
Blockchain
Cloud Computing
Cybersecurity
Data Science
Machine Learning
Tech Trends
Virtual and Augmented Reality
SEARCH

Explainable AI (XAI) is a hot topic right now. We've recently seen a boom in AI, and that's mainly because of the Deep Learning methods and the difference they've made. There are many more use cases of AI now compared to the times before Deep Learning was introduced.

The problem that we're facing today is that many methods work, but we don't elaborate on the details of whatever is done under the hood. But it's very important to understand how the prediction is done, not just to understand the architecture of the method.

Explainable AI is useful for:

domain experts,
regulatory agencies,
managers and executive board members,
data scientists,
users that use machine learning, with or without the awareness.

The domain experts or the users of an AI or machine learning model (like doctors, for example) trust the model itself, gain scientific knowledge. The regulatory agencies certify the model's compliance with the legislation in force. The managers assess the model's regulatory compliance and need to understand the possible corporate applications of AI. The data scientists ensure and improve product efficiency or develop new functionalities.

Every other user affected by the model's decision wants to understand the situation and verify if the decision is fair.

Goals for an explainable AI model to fulfill

There are many goals an XAI model should fulfill.

However, not every goal can be met with every method, and each goal has a different target audience.

Here are a few examples:

The domain experts and other users affected by the model should be able **trust** it.
We should be able to **transfer the knowledge** that we can gain from the model to other problems or challenges.
We should understand the models enough to ensure **privacy** of the data used for training the model and what's done with the data during the prediction process. The European Union is already working on a directive on privacy and machine learning.
Every model should be done in a way that doesn't affect any minorities. In other words, every model should be **fair and ethical**.
Every model should be **robust** and **informative**. We should be confident that the prediction is valuable and related to the user's decision.
And finally, every model should be **accessible** to non-technical people. They should understand how it works and, in some cases, be able to **interact** with it.

Not every XAI model needs to fulfill all of the goals, and not every model that meets one of the goals above is an *XAI* model.

Levels of transparency of an XAI model

The transparency of a model can be divided into **three levels**, depending on how transparent the model is.

→ Transparency level 1:

We should achieve a model that is fully **simulatable**, which means it can be fully simulated by a human. Simulatable models are the most demanded type. Most machine learning projects use shallow and scikit-learn methods to increase the chances that their model will be simulatable.

→ Transparency level 2:

The second level of transparency is reached when a model can be **decomposed**. This means that we are able to divide the model into parts and explain how each part works and how it processes the data. In many models, especially in models based on neural networks, we can only explain just one part of the whole model in detail.

→ **Transparency level 3:**

The last level is **algorithmic transparency**, which means that we understand how the model produces the output. In most cases, it can be achieved with simple methods easily understood by the user.

How to explain a model?

There are many ways to explain how a model works (it's also called *post-hoc explainability*). Typically, we use text to explain it, but we can also use symbols and formulas.The most popular method for explanation is based on **charts**. It's an easily interpretable way for humans to understand how a model works. We simply take a subspace of the model and explain it in a couple of different ways.

Another easy to understand method for explanation is doing it through **an example**. We take some input data and explain what happens with it during the process, step by step.

If the model is too complex, we may simplify it and explain the way it works on a **simplified** model.

Black-box method example

For the black-box method example, we'll use a three layer network. Each layer is a dense. It's simple, but still complex enough to be considered a black-box method, even if we achieve almost 99% of accuracy.

```
from keras.datasets import mnist
from keras import models
from keras import layers
from keras.utils import to_categorical

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
train_images = train_images.reshape((60000, 28 * 28))
train_images = train_images.astype('float32') / 255
test_images = test_images.reshape((10000, 28 * 28))
test_images = test_images.astype('float32') / 255

train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)

```

```
network = models.Sequential()
network.add(layers.Dense(784, activation='relu', input_shape=(28 * 28,)))
network.add(layers.Dense(784, activation='relu', input_shape=(28 * 28,)))
network.add(layers.Dense(10, activation='softmax'))
network.compile(optimizer='adam',
            loss='categorical_crossentropy',
            metrics=['accuracy'])

network.fit(train_images, train_labels, epochs=5, batch_size=128)

test_loss, test_acc = network.evaluate(test_images, test_labels)
print('test_acc:', test_acc, 'test_loss', test_loss)
```

We can draw the weights of the layers. It's very hard or even impossible to interpret them.

```
print(network.layers[1].get_weights())
```

The code above will produce about tens of lines like the one below:

```
[array([[ 0.00166182, 0.04952418, 0.08845846, ..., 0.00472951,
-0.04272532, 0.04789469],
[-0.0524085 , -0.03233211, -0.0232333 , ..., -0.0056492 ,
0.04325055, -0.06916995],
[-0.01691317, 0.02450813, 0.06632359, ..., -0.06094756,
-0.14761966, -0.02945693],
...,
```

These are the values of the weights of just one layer. In many neural networks, the number of weights is counted in millions. It's hard to explain each weight, in many cases even impossible. That's why we call such methods black-box methods.

White-box method example

There are plenty of white-box methods. One method that is well-known and easy to interpret is the decision tree. Modified versions are used in Kaggle competitions with success.

It has many advantages:

- it's simple,
- it's interpretable,
- it's easy to convert to a set of rules,
- it's a feature importance tool.

We can easily draw the tree and see all the decisions that are made on each node.

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree

X, y = load_iris(return_X_y=True)

dtc = DecisionTreeClassifier()

dtc.fit(X, y)
```

It can be effortlessly replaced with a set or rules (*if* statements).

```
from sklearn.tree import export_graphviz
import graphviz

dot_data = export_graphviz(dtc, out_file=None,
                           filled=True, rounded=True,
                           special_characters=True)
graph = graphviz.Source(dot_data)
graph
```

Python tools for explainable AI

In Python, we have a number of tools to understand how the model works.

Here are a few of them:

eli5 — it's a tool for debugging and explaining the prediction of a model. With eli5, we can easily debug the models using scikit-learn or xgboost. It also works with NLP models as a tool for text and feature importance explanation. You can download eli5 here.

XAI — it's a tool based on 8 Responsible AI principles. It can be used for data analysis and model evaluation. The data analysis is a similar solution to pandas profiling. Model evaluation can be used together with scikit-learn and keras. You can download XAI here.

IBM aix360 — it's a tool provided by IBM with many examples and tutorials. It covers several explainability algorithms including data, local and global direct explanation. You can download IBM aix360 here.

Shap — it's a tool that uses the SHapley Additive exPlanations method. The Shap method is a game-theoretic approach to explaining the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extension. You can download Shap here.

Lime — it's a tool able to explain any black box classifier with two or more classes. It can also explain a network based on images. It's good for deep networks. You can download Lime here.

Skater — it's a tool that can be used for various models including NLP, ensemble, and image recognition models. It uses lime for image interpretation. You can download Skater here.
TensorBoard — it's a tool used for explaining models based on TensorFlow. TensorBoard uses the log files generated by the TensorFlow model during the training phase. You can read more about TensorBoard here.

If you need experienced specialists in Artificial Intelligence to help you with explainable AI, don't hesitate to reach out to us at mail@codete.com.

You may also be interested in:
Machine Learning vs. Data Science: Similarities and Differences You Need to Know
Machine Learning Libraries Overview: Top 10 Libraries and Frameworks You Should Know
Data Prefetching in Web Applications – a Survey of Available Methods

Karol Przystalski karol.przystalski

Karol Przystalski is CTO and founder of Codete. He obtained a Ph.D in Computer Science from the Institute of Fundamental Technological Research, Polish Academy of Sciences, and was a research assistant at Jagiellonian University in Cracow. His role at Codete is focused on leading and mentoring teams.
share:

We were unable to load Disqus. If you are a moderator please see our troubleshooting guide.
share:

contact our tech consultant
E-MAIL
related posts
Blockchain
Digital Transformation
Technology
Blockchain Technology Explained: What Is It and How Does It Work?
February 9, 2021
Digital Transformation
Tech Trends
Technology
Tech Trends Forecast for 2021

January 5, 2021

Digital Transformation

Tech Trends

Technology

7 Technology Trends in the Fashion Industry

December 22, 2020