By Paul Naidoo, EOLAS

Peeking into the ‘black box’ with Explainable AI (XAI)

Artificial Intelligence (AI) allows us to draw insight from complex, multi-modal data in ways that a classical analysis simple cannot. However, these abilities come at the expense of increasing model complexity, and obfuscation of what is driving the decisions/conclusions these models reach. This “black box” effect means that understanding their decision-making processes becomes increasingly challenging. This is where Explainable AI (XAI) comes into play.

Explainable AI (sometimes called Interpretable ML) refers to techniques and methods that make the behaviour and decisions of AI systems more transparent and understandable to humans. By providing insights into how AI models arrive at their conclusions, XAI helps bridge the gap between complex algorithms and human comprehension.

Increasing Trustworthiness

As well as empowering developers with an understanding of how their model is working, one the greatest benefits of XAI is its ability to enhance the trustworthiness of AI systems. It achieves this by increasing:

1. **Transparency**: XAI provides insights into the inner workings of AI models, allowing users to see how decisions are made. This transparency helps users feel more confident in the system's reliability and fairness.
2. **Accountability**: With explainable AI, it becomes easier to identify and address errors or biases in the decision-making process. This accountability ensures that AI systems can be held to ethical standards and regulatory requirements.
3. **User Confidence**: When users understand how AI systems operate, they are more likely to trust and adopt these technologies. Clear explanations can alleviate concerns about the unpredictability or opacity of AI decisions.

XAI methods

Explainable AI is a growing field. The following is an (non-exhaustive) summary of some key Explainable AI techniques, indications of when you might prefer to use each and links to free and open-source Python modules which can be used for each analysis. Note that there may be alternative modules for each method. Those listed simply provide a starting point for the reader.

1. LIME (Local Interpretable Model-agnostic Explanations)

Summary: LIME explains individual predictions by approximating the model locally with an interpretable model. It perturbs the input data and observes the changes in predictions to understand the model's behaviour.

When to Use: LIME is useful when you need to explain specific predictions of any black-box model, especially when dealing with tabular data or text.

Module: LIME

2. SHAP (SHapley Additive exPlanations)

Summary: SHAP values are based on cooperative game theory and provide a unified measure of feature importance. It calculates the contribution of each feature to the prediction by considering all possible feature combinations.

When to Use: SHAP is preferred when you need consistent and fair explanations across different models and datasets. It's particularly useful for understanding global feature importance and interactions.

Module: SHAP

3. Grad-CAM (Gradient-weighted Class Activation Mapping)

Summary: Grad-CAM visualizes the regions of an input image that are important for the model's prediction by using the gradients of the target output with respect to the last convolutional layer.

When to Use: Grad-CAM is ideal for explaining predictions of convolutional neural networks (CNNs) in image classification tasks.

Module: grad-cam

4. Integrated Gradients

Summary: This technique attributes the prediction of a deep network to its input features by integrating the gradients of the output with respect to the input along a path from a baseline to the input.

When to Use: Integrated Gradients are useful for models where you need to understand the contribution of each input feature, especially in scenarios where baseline comparisons are meaningful.

Module: captum (pytorch), tf-keras-vis (keras)

5. Counterfactual Explanations

Summary: Counterfactual explanations provide insights by showing how the input data needs to be modified to change the prediction. They answer "what-if" questions by identifying minimal changes to the input that would alter the output.

When to Use: Use counterfactual explanations when you need actionable insights or when you want to understand the decision boundaries of the model.

Module: DiCE

Choosing the Right Technique

- **For Local Explanations**: Use LIME or Anchors when you need to explain individual predictions.
- **For Wholistic Explanations**: Use SHAP to understand overall feature importance and interactions.
- **For Image Data**: Use Grad-CAM for visual explanations of CNNs.
- **For Deep Networks**: Use Integrated Gradients to attribute predictions to input features.
- **For Actionable Insights**: Use Counterfactual Explanations to understand how to change outcomes.