Why should I trust you? 🙄

Deep learning has tremendous potential to identify inherent patterns and feature representations hidden in large pools of data. They are being used in all spheres of our life. However, what is the first thought we have, hearing the word deep learning?

Black box. Isn't it? 👽

Imagine applying for a role in an organization, where the applications are shortlisted by a system using machine learning algorithms. Unfortunately, you get rejected (I sincerely hope that you don't)

Wouldn't you want to know why were you rejected?

Now, given the potential of deep learning to solve major problems, should we stop using it for complex applications which requires reasoning? 🤔

This is where explainable AI (XAI) comes into the picture. In formal terms, "XAI is a set of tools, techniques, and frameworks intended to help users and designers of AI systems understand their predictions, including how and why the systems arrived at them."

Now, that you understand the significance of XAI, let's dive deeper and understand the levels of explainability 😀

- **User benefit**: this type of explanation is designed to inform the user about an output. For example, *why was your application rejected*?
- **Societal acceptance**: such an explanation is designed to generate trust and acceptance within the society. If an unexpected result is generated, the model needs to be able to provide an explanation for such as result
- **Global understanding**: providing explanations about the basis of predictions or results being generated by the model.

We shall explore the local (User benefit) aspect of interpretability.

Some of the popularly used methods are as:

- **Layer-wise relevance propagation (LRP):** It operates by propagating the prediction backward in the neural network, using a set of purposely designed propagation rules
- **Counterfactual method**: what could have happened if the input to the model had been changed in a certain way
- **Local Interpretable Model Agnostic explanations (LIME):**
- **Generalized additive model (GAM)**
- **Rationalization**
- **Shapley Additive Explanations (SHAP)**

Reference: https://www.aisoma.de/5-methods-for-explainable-ai-xai/

LIME (Local Interpretable Model Agnostic) and **SHAP (Shapley Additive Explanations)** are the two primary approaches of local interpretability.

Let us start exploring **LIME.** I am excited 😬 …. Are you??

The article is hereafter divided into 3 major sections:

1. Intuition of LIME
2. Mathematics of LIME
3. Implementation of LIME in python

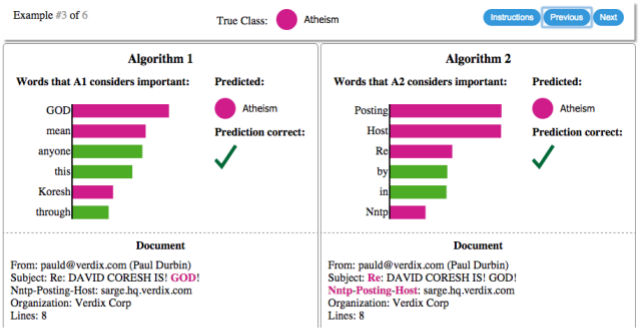


Figure 2: Explaining individual predictions of competing classifiers trying to determine if a document is about "Christianity" or "Atheism". The bar chart represents the importance given to the most relevant words, also highlighted in the text. Color indicates which class the word contributes to (green for "Christianity", magenta for "Atheism").

Which of the two models is more reliable?

LIME helps us to rationalize the prediction of the model by interpreting the most significant features

Ref: https://arxiv.org/pdf/1602.04938v3.pdf

How does LIME solve the problem of rationalization in complex problems?

LIME tries to approximate the prediction result of the model for any given data point using simple linear models. It first generates surrogate dataset (neighborhood points) around the sample datapoint. It is based on the idea of representing the non-interpretable inputs (such as word embedding, pixels in an image) into human interpretable inputs such as one hot encoded representation and super pixels (high level representation of a cluster of pixel). It tries to ensure that the explanation is short enough to be interpretable by humans.

"The overall goal of LIME is to identify an interpretable model over the interpretable representation that is locally faithful to the classifier"

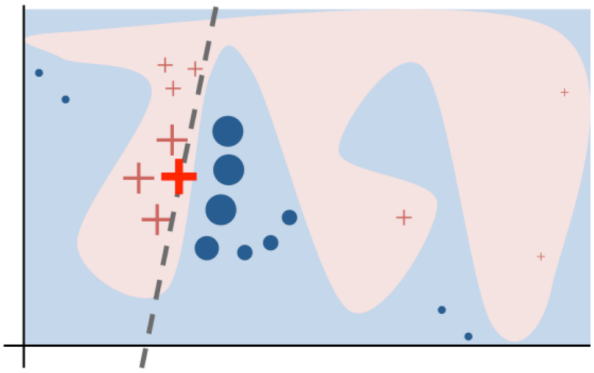
Mathematically, LIME is represented as an optimization problem

$$\xi(x) = \text{argmin}_{g \in G} \; L(f, g, \pi_x) + \Omega(g)$$

Let us dive deeper to understand the notations, $g \in G$, where G is a class of potentially interpretable models. As not every g will be simple enough to easily interpret, $\Omega(g)$ denotes the complexity of the model G. It acts a regularizer.

Interpretable explanations need to use a representation that is understandable to humans.

An illustration of this process is given below. The original model's decision function is represented by the blue/pink background, and is clearly nonlinear. The bright red cross is the instance being explained (let's call it X). We sample perturbed instances around X, and weight them according to their proximity to X (weight here is represented by size). We get original model's prediction on these perturbed instances, and then learn a linear model (dashed line) that approximates the model well in the vicinity of X. Note that the explanation in this case is not faithful globally, but it is faithful locally around X.



We denote $\mathbf{x} \in \mathbf{R}^d$ in the original representation of an instance, as $\mathbf{x}' \in \{0,1\}^{d'}$ to denote binary vector for its interpretable representation. Also, **g** is $\{0,1\}^{d'}$, i.e; g acts over the absence/presence of the interpretable components. π_x is the **proximity measure** between an instance z to x, which defines the locality around x

The function $\mathbf{f}: \mathbf{R}^d \rightarrow \mathbf{R}$, f(x) is the probability that x belongs to certain class for the model being interpreted.

Thus, $L(f, g, \pi_x)$ measures how unfaithful g is in approximating f in the locality defined by π_x

How do we implement LIME?

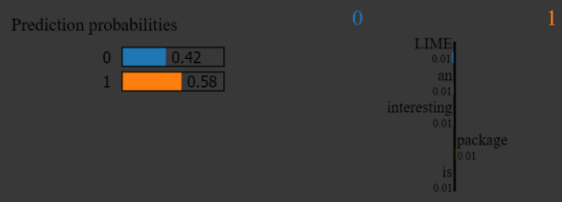
LIME has a beautiful implementation in python and is available through the PyPI package manager.

Using the LIME implementation in python, we can easily explore the interpretability of model based on text, image or tabular data

Please check the notebook link. Lime text explainer is used to explain the predictions of pretrained bert sequence classifier of Hugging face library.

```
[80]  1 text_exp = txt.LimeTextExplainer(class_names=[0,1])
      2 exp = text_exp.explain_instance(inp[0], model, labels=[1], num_samples=1000)
```

```
1 exp.show_in_notebook()
```




```
[40]  1 from skimage.segmentation import mark_boundaries
      2
      3 temp_1, mask_1 = exp.get_image_and_mask(exp.top_labels[0], positive_only=True, num_features=5, hide_rest=True)
      4 temp_2, mask_2 = exp.get_image_and_mask(exp.top_labels[0], positive_only=False, num_features=10, hide_rest=False)
      5
      6 fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(10,10))
      7 ax1.imshow(mark_boundaries(temp_1, mask_1));
      8 ax2.imshow(mark_boundaries(temp_2, mask_2));
      9 ax1.axis('off')
     10 ax2.axis('off')
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
(-0.5, 223.5, 223.5, -0.5)
```

ref: https://itrexgroup.com/wp-content/uploads/2021/08/DarpaExplainabelAI.png

https://en.wikipedia.org/wiki/Explainable_artificial_intelligence

https://homes.cs.washington.edu/~marcotcr/blog/lime/

https://arxiv.org/pdf/1602.04938v3.pdf

I hope you found the article useful. Thank you for reading 😊