Course Summary: XAI - Introduction

Leroy Octave

1 Introduction to Machine Learning and XAI

Machine Learning (ML) allows computers to make predictions or decisions based on data rather than explicit instructions. It represents a shift from explicit programming, via algorithm, to data-driven programming. While ML has strengths, such as being faster than humans, consistent, cheap and scalable, it also introduces strong limitations : predictions are difficult to understand and insights are often hidden within a complex model. In this course, we will denote a model $\hat{f}$ applied to data x as $\hat{f}(x)$.

Machine Learning models are generally categorized into two main types : Black box models and Interpretable, or White box, models. Black box models have complex internal mechanisms that are not understandable (e.g Deep neural network) whereas interpretable models have internal logic that is clear and understandable.

Interpretability cannot be defined with a single formal mathematical definition due to the presence of humans in the process. In fact, lots of different and subjective definitions have emerged through the years. In the following we will consider the definition of Miller (2017) :

"Interpretability is the degree to which a human can understand the cause of a decision" (e.g. predict the model's output).

Interpretability can be achieved in various ways with various tools and strategies. Tools have been designed for white box model and black box model. Furthermore model-agnostic, where models are treated as black boxes, even if they are not, can be adopted.

Nowadays interpretability is highly important. In fact, high performance and accuracy are not sufficient for real-world tasks because problems are often incompletely specified. Moreover in numerous domains, such as Healthcare, the why matters as much as the what. Interpretability is crucial for several reasons. First for curiosity and learning, humans build mental models and need explanations to learn insights by the ML and understand events. Second interpretability and explanations influence beliefs and actions of users, they are necessary to persuade users. Finally, understanding a model helps detect inherited biases and discrimination. Interpretability enables several properties as Fairness (absence of discrimination), Privacy, Robustness, Causality and Trust.

2 Taxonomy and Evaluation

Interpretability methods are categorized based on their nature and their scope.

Nature :

- **Intrinsic** involves using simple models and interpretable by nature.
- **Post-hoc** are explanations generated after the model has been trained.

Scope :

- **Global** explains the overall behaviour of the model.
- **Local** explains individual predictions of an instance.

Several methods of interpretability exist and can be categorized into :

- **Attribution Methods** assign a numerical score to input features to determine which parts of the inputs matter the most for the prediction. They include perturbation-based methods (LIME, SHAP) and gradient-based method.
- **Concept-Based Explanations** aim to identify high-level human ideas or concepts present in the model's logic. Explanations remain on understandable ideas and concepts.
- **Example-Based Explanations** use specific data points, similar or contrasting to justify classification. It can be counterfactuals, showing how changing variables values flip the outcome or prototype, showing typical examples of a class.
- **Rule-Based & Textual Rationalization** simplifies the logic of the model into sets of rules or use LLM to generated natural language explanations of the reasoning.

An effective explanation method should have several important properties :

- **Accuracy** assesses how the explanation predicts the true outcome. It's a quite rare property.
- **Fidelity** assesses how the explanation matches the behaviour of the black-box models. High fidelity is essential.
- **Consistency** explanations are similar across different models
- **Stability** implies that similar inputs should have similar explanations. A high stability is always desirable for robustness.
- **Comprehensibility** assesses how well the explanation is understood by humans. Comprehensibility is the most critical and difficult property to measure. Often it's proxied by the explanation size, the numbers of features used or the number of rules.

3 Interpretable Models

Interpretable models can be explained easily, due to their simple architectures, by analysing their internal structure and parameters.

3.1 Linear Regression

Linear regression is the simplest model for regression. It predict a continuous target variable as a weighted sum of the input features. The model for an instance i is defined as :

$$y^{(i)} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j^{(i)} + \epsilon^{(i)}$$

Where $y^{(i)}$ is the target value, β_j is the learned weight for feature j, β_0 is the bias and $\epsilon^{(i)}$ is the error term.

The goal of linear regression is to estimate the coefficients β that best explain the relationship between the features and the target. An Ordinary Least Squares (OLS), minimizing the squared prediction error, is used to train the model :

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} \left(y^{(i)} - \beta_0 - \sum_{j=1}^{p} \beta_j x_j^{(i)} \right)^2$$

However, linear regression relies on strong assumptions : relationship between features and target is linear, errors must be independent and normally distributed, and the variance of errors must be constant. Making the errors symmetric and frequently small.

Linear regression is interpretable by nature. In fact, the prediction is a direct sum of feature contributions. Each coefficients β_j quantifies the effect of the feature j. To assess how well the regression model explains variability of the target variable one use the coefficient of determination R^2. R^2 measures the fraction of the target variable's explained by the model.

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum (y^{(i)} - \hat{y}^{(i)})^2}{\sum (y^{(i)} - \bar{y})^2}$$

R^2 have value between 0 and 1. If R^2 is equal to 1 the model explain all variability,while 0 means it is no better than predicting the mean. However R^2 always increase when adding features even useless ones. Therefore, an adjusted R^2 is used to correct this bias :

$$\bar{R}^2 = R^2 - (1 - R^2)\frac{p}{n - p - 1}$$

R^2 can be used to compare models on a same dataset and check sanity of the model before interpreting the coefficients.

Lot of tools enable to interpret linear regression, for example feature importance is assessed using t-statistics.

$$t_{\hat{\beta}_j} = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}$$

t-statistic test if a coefficient is significantly different from zero. A large absolute value, $|t|$, indicates a strong and reliable effect. A small value indicates the effect may be related to noise. The standard error (SE) measure the uncertainty of the estimate, which depend on noise, data size and feature correlation.

Moreover we can visualize the effect of coefficients via Weight Plot and Effect Plot. Weight Plot display raw coefficients, with confidence interval of 95%, which can be miss-leading due to the used of different scales per features. It's better to standardize the data before training the model to obtain scale-independent t-statistics.

Another visualization is the Effect Plot, visualizing the contribution of a feature to the prediction :

$$\text{effect}_j^{(i)} = \hat{\beta}_j x_j^{(i)}$$

Effect plots are boxplots that allow global explanations, via distributions of effects, and local explanations via contributions of each variable for a specific instance.

Linear explanations have several advantages if assumptions hold. In fact they are truthful, simple, easy to communicate and transparent. However, those approach suffer from poor contrastive power and low selectivity by default. Moreover linear explanation can be improved by mean-centering features, using effect coding for categorical variables or use sparse linear models. .

3.2 Logistic Regression

Logistic regression comes from linear regression but models the probability $\mathbf{P}(y = 1|x)$ rather than predicting the class directly, ensuring the outputs remain within the interval $[0, 1]$. To achieve the probabilistic output, the linear regression is transformed using the sigmoid function:

$$\sigma(\nu) = \frac{1}{1 + \exp(-\nu)}$$

The logistic regression is defined as:

$$\mathbf{P}(y^{(i)} = 1) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1^i + ... + \beta_p x_p^i))}$$

Logistic regression assumes a linear relationship between the input features and the log-odds, not the probabilities directly. Explanation comes from this relationship. In fact, the relationship between probabilities and the linear model is non-linear due to the sigmoid function.

$$\log \frac{\mathbf{P}(y = 1)}{1 - \mathbf{P}(y = 1)} = \beta_0 + \beta_1 x_1^i + ... + \beta_p x_p^i$$

In logistic regression, effects are multiplicative regarding the odds. In fact, if a feature x_j increases by one, the odds are multiplied by a factor of $\exp(\beta_j)$. β_0 represents the log-odds when all numerical features are zero.

3.3 Decision Trees

Decision Trees are supervised learning models used for both classification and regression. Decision Trees recursively split the feature space with their nodes. Each internal node tests a feature against a threshold t (e.g., $x_j \leq t$), and each leaf node provides a final prediction: for classification the prediction is the majority class, for regression the prediction is the average target value in the leaf.

In order to make a prediction for an instance x, the model traverses the tree starting from the root. At each node, it evaluates the split condition and moves to the corresponding child node until a leaf is reached.

Training involves learning the best feature and threshold to split the data at each node in order to minimize an impurity measure. Common ones are Entropy or Gini impurity:

$$Gini = 1 - \sum_{k=1}^{K} p_k^2$$

Decision trees are inherently explainable by following the path to leaves and creating rules. Moreover, there are no hidden transformations or latent representations. A global explanation is given by the full tree structure, revealing how features interact, and feature importance is derived from the impurity reduction of a split. A local explanation can be done by the path of a specific instance.

3.4 Decision Rules

A decision rule is a simple "IF-THEN" statement composed of one or multiple conditions and a prediction. If the conditions are met, the prediction is applied. A predictive model can consist of a single decision rule or a set of multiple rules. However, in Machine Learning, the rules are not written by human experts but are learned automatically from data. Moreover, a default rule, with no conditions, is often used to ensure full coverage of the input space when no other rules apply.

Decision rules are highly interpretable by nature. In fact, they are similar to natural language, mirror human reasoning, can be inspected independently, and have no hidden transformations or latent representations. However, to remain interpretable, the model must use intelligible features, contain few conditions per rule, and limit the total number of rules.

4 Model-Agnostic Methods

Model-agnostic explanation methods aim to explain the predictions of a Machine Learning model without using any information about its internal structure. They treat the model as a black box, requiring access only to the input features x and the model predictions $\hat{f}(x)$. Since they do not rely on model parameters, the same method can be applied to any model, from linear regression to neural networks, enabling fair comparisons of interpretability across different models.

4.1 Partial Dependence Plots

Partial Dependence Plots (PDP) visualize the average effect of one or two features on the prediction of a machine learning model. They explain how the prediction changes on average when a feature value changes and the relationship of the prediction with those features. PDPs provide a global explanation of the model, summarizing the model behaviour across the entire dataset.

Let X_S be the features of interest and X_C the remaining features. The partial dependence function is the expectation of the model prediction over the marginal distribution of X_C:

$$\hat{f}_{x_S}(x_S) = \mathbb{E}_{x_C}[\hat{f}(x_S, x_C)] = \int \hat{f}(x_S, x_C) dP(x_C)$$

We fix X_S to a specific value or a grid of values and average the predictions over the distribution of all other features X_C. In practice, the expectation is approximated using the training data and the Monte Carlo approximation:

$$\hat{f}_{x_S}(x_S) = \frac{1}{n} \sum_{i=1}^{n} \hat{f}(x_S, x_C^{(i)})$$

PDPs are intuitive, easy to implement, and provide a causal interpretation for the model. However, they are limited to one or two features and make a strong assumption of independence of X_S and X_C. If features are correlated, PDPs may average over impossible combinations, leading to biased feature effects.

4.2 Marginal Plots

Marginal Plots (M-Plots) address the issue of correlated features, which is a limitation of PDPs, by averaging conditionally. M-Plots show what the model predicts for instances that have a specific feature value and mix the effect of the feature of interest and of correlated features. The key difference between M-plots and PDPs lies in the distribution used to average:

$$\hat{f}_{x_S,M}(x_S) = \mathbb{E}_{X_C|X_S=x_S}[\hat{f}(x_S, X_C)]$$

In practice, one cannot force a value because a specific combination might not exist in the dataset. Instead, for a value v_k, a local neighbourhood of instances, defined by a bandwidth σ, is selected, and an average prediction is calculated within that neighbourhood:

$$\hat{f}_{x_S,M}(v_k) = \frac{1}{|\mathcal{N}(v_k)|} \sum_{i \in \mathcal{N}(v_k)} \hat{f}(x^{(i)})$$

4.3 Global Surrogate

A global surrogate is an interpretable model g trained to approximate the predictions of a complex black-box model f: $g(x) \approx f(x)$ for any input x. Explanations and interpretations of the internal logic of the simple model g give understanding of the behaviour of the complex model f. To train the global surrogate, one selects a dataset X and computes the predictions of the black-box model $\hat{f}(X)$. Then, one chooses an interpretable model class g (e.g., linear model, decision trees), trains g on the dataset $(X, \hat{f}(X))$ and evaluates the quality of g by computing R^2. The surrogate model g never sees the true ground truth labels; it only learns the behaviour of the black-box model.

This method presents several advantages: it is fully model-agnostic, allows flexibility in choosing the interpretable model, produces easy explanations to communicate, and provides quantifiable fidelity via R^2. However, it explains the model rather than the real-world data, and no consensus on what constitutes an acceptable R^2 threshold is defined.

4.4 LIME

The LIME method aims at explaining locally a highly complex black-box model rather than globally. In fact, LIME explains one single prediction of a black-box model in a way that humans can understand. The intuition is that while a complex model is non-linear globally, it can be approximated by a linear model locally, in the neighbourhood of an instance. To generate explanations, LIME distinguishes between two representations of the data: the original representation and the interpretable representation. The original representation, $x \in \mathbb{R}^d$, is the raw input used by the model (e.g., word embeddings), often not human-readable. The interpretable representation, $x' \in \{0,1\}^d$, is a binary vector indicating the presence or absence of human-understandable components (e.g., words in text).

To understand which parts of an input x are responsible for a prediction, LIME uses a perturbation-based approach. It generates perturbed samples z' around x' by randomly turning off interpretable features. These perturbed samples z' are mapped back to the original input space as z to be processed by the black-box model. Furthermore, a similarity kernel $\pi_x(z)$ assigns weights to these samples based on their proximity to the original instance x, ensuring the explanation remains local.

$$\pi_x(z) = \exp\left(-\frac{D(x,z)^2}{\sigma^2}\right)$$

Where D represents a distance between x and z.

The objective function to train g, which minimizes the local weighted loss, is defined as:

$$\mathcal{L}(f, g, \pi_x) = \sum_{(z,z') \in \mathcal{Z}} \pi_x(z) \left(f(z) - g(z')\right)^2$$

Where $f(z)$ is the prediction of the black-box model, $g(z')$ is the prediction of the surrogate model, and $\pi_x(z)$ is the locality weighting.

Moreover, to make the explanation human-readable, LIME enforces sparsity via a K-Lasso, ensuring a small number of the most important interpretable features are selected for explanation.

4.5 SHAP

SHapley Additive exPlanations (SHAP) is based on the cooperative game theory of Shapley. This method treats the model predictions as a game where input features are players cooperating to produce a prediction. In this framework, the players are a set of features $D = \{1, \ldots, d\}$, a coalition is any subset $S \subseteq D$ of features, and a characteristic function $v : 2^D \to \mathbb{R}$ defines the value produced by the coalition S. The goal of SHAP is to determine the contribution of each player or feature to the total value, and to identify the essential players.

The core mechanism for attributing value is the marginal contribution. It is the change in value when a player i joins an existing coalition S :

$$v(S \cup \{i\}) - v(S)$$

The contribution of any player i is defined as :

$$\phi_i(v) = \mathbb{E}_\pi \left[v(S^i_\pi \cup \{i\}) - v(S^i_\pi) \right]$$

Where π is a random permutation of all players and S^i_π is the set of players that appear before i in the ordering π. In fact, the Shapley value is the average of the marginal contributions across all possible permutations of players. The explicit formula to compute the Shapley value is :

$$\phi_i(v) = \sum_{S \subseteq D \setminus \{i\}} \frac{|S|!(d - |S| - 1)!}{d!} \left[v(S \cup \{i\}) - v(S) \right]$$

Where S is a coalition without player i.

Moreover, the Shapley value satisfies four specific fairness axioms:

- **Efficiency :** The total value is fully distributed among players.
- **Symmetry :** Interchangeable players, who contribute equally to all coalitions, receive the same credit.
- **Null Player :** A player who never adds value to any coalition receives zero credit.
- **Linearity :** The Shapley value of a linear combination of games is the linear combination of the Shapley values.

However, SHAP in Machine Learning adapts the Shapley theory to Machine Learning by mapping players to input features and game value to model prediction. In ML, the value function is slightly different. In fact, the value of a coalition S, a subset of features, denoted $F(x_S)$, is the expected output of the model given the known features, marginalizing over the missing features $x_{\bar{S}}$:

$$v(S) = F(x_S) = \mathbb{E}_{x_{\bar{S}}|x_S} \left[f(x_S, x_{\bar{S}}) \right] = \sum_{x_{\bar{S}}} f(x_S, x_{\bar{S}}) p(x_{\bar{S}}|x_S)$$

However, calculating the exact Shapley values has a complexity of $O(2^d)$, which is infeasible for large feature sets. Therefore, SHAP is approximated by sampling values and computing a Monte Carlo integration:

$$\mathbb{E}_{x_{\bar{S}}|x_S}[f(x_S, x_{\bar{S}})] \approx \mathbb{E}_{x_{\bar{S}}}[f(x_S, x_{\bar{S}})] \approx \frac{1}{m} \sum_{i=1}^{m} f(x_S, x^{(i)}_{\bar{S}})$$