# Trustworthy AI in Society

Ayush M

## 9 Importance

The European Union enacted the right to explanation which was incorporated in the EU General Data Protection Regulation (GDPR) in 2018: [...] In any case, such processing should be subject to suitable safeguards, which should include specific information to the data subject and the right to obtain human intervention, to express his or her point of view, to obtain an explanation of the decision reached after such assessment and to challenge the decision

**Note 1. Explainability more important then ever**

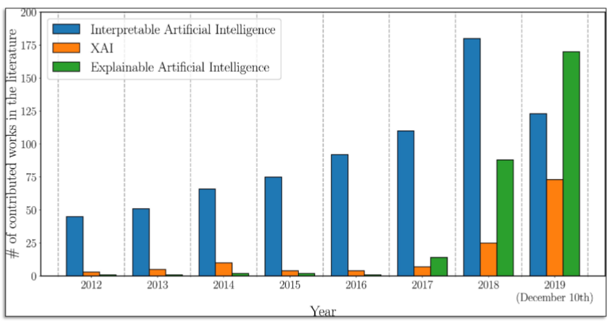


Figure 1: Explainability Trends

### 9.1 Explainability vs Interpretability

"An AI system is explainable if the task model is intrinsically interpretable (here the AI system is the task model) or if the non-interpretable task model is complemented with an interpretable and faithful explanation (here the AI system also contains a post-hoc explanation)
Accuracy Increases as Interpretability decreases, generally

**The Tidal Force**

$$F_{tide} = -GM_m m(\frac{\hat{d}}{d^2} - \frac{\hat{d}_0}{d_0^2}) \tag{1}$$

Where:

$$G = \text{Gravitational Constant}$$
$$d = \text{Object's Position Relative to Moon}$$
$$d_0 = \text{Earth's Center Relative to the moon}$$
$$M_m = \text{Mass of the moon}$$

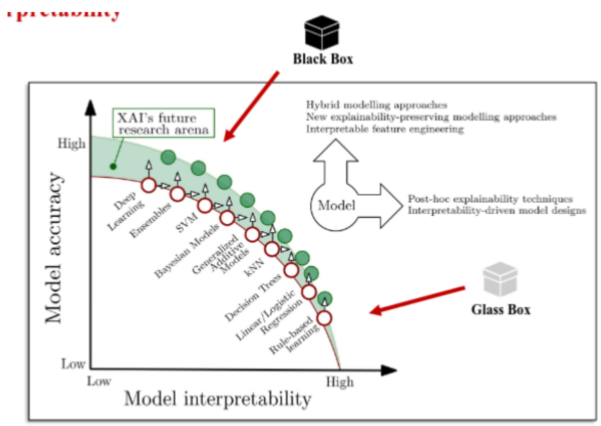


Figure 2: Enter Caption

# 10 Federated Learning

“Federated Learning is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider. Each client’s raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective.”

# 11 Differential Privacy

“Differential privacy addresses the paradox of learning nothing about an individual while learning useful information about a population.”

# 12 Glass Box Models

- Linear Models
- Generalized Additive Models
- Explainable Boosting Machines
- Decision Trees
- Rule Based Approaches

## 12.1 Linear Models

Literally just least squares ols. you get a bunch of betas as you fit a x to a y.
Regularization
Can use ridge, lasso, elastic net etc.

### 12.1.1 Bayesian Inference

Prior doesn’t need regularization

## 12.2 Log Reg

## 12.3 Generalized Additive Models

Sum of functions

Logistic Regression

$$\min_{\beta_0,\beta} \frac{1}{n}\sum_{i=1}^{n}\left(\log(1+e^{(\beta_0+x_i^\intercal\beta)}) - y_i(\beta_0+x_i^\intercal\beta)\right)$$

Logistic Regression with Regularization

$$\min_{\beta_0,\beta} \frac{1}{n}\sum_{i=1}^{n}\left(\log(1+e^{(\beta_0+x_i^\intercal\beta)}) - y_i(\beta_0+x_i^\intercal\beta)\right)+\lambda\left(\tfrac{1-\alpha}{2}\|\beta\|_2^2+\alpha\|\beta\|_1\right)$$

$\lambda \geq 0,\ \alpha \in [0,1]$ hyperparameters

Figure 3: Log reg formula

## 12.4 Explainable Boosting

Sum of functions and sum of cross-interaction functions

### 12.4.1 Regression Trees

Recursive Splitting of samples in every level of the tree such as to minimize error

### 12.4.2 Tree pruning

Avoid overfitting with a fully grown tree

### 12.4.3 Classification Tree

Uses error metrics based on purity of a region. Generally Unstable

## 12.5 RUG

Builds rules (boolean) to classify, very interpretable, not as good performance

Misclassification error: $1 - \max_k \{\hat{p}_{mk}\}$

Cross entropy: $- \sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$

Gini index: $\sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$

[h]

Figure 4: Classification Tree

# 13 Unboxing

# 14 LIME

Explains any model by approximating with an interpretable model. Explains locally.

## 14.1 Working

- Create perturbed samples around the local point $x_i$
- Compute $f(x_i)$ for each perturbed sample from black box model
- assign weights to the samples based on distance from $x_i$

$$\pi_{x_0}(z_i) = \exp\left(-\frac{D(x_0, z_i)^2}{\sigma^2}\right)$$

- minimize weighted error + model complexity (lasso)

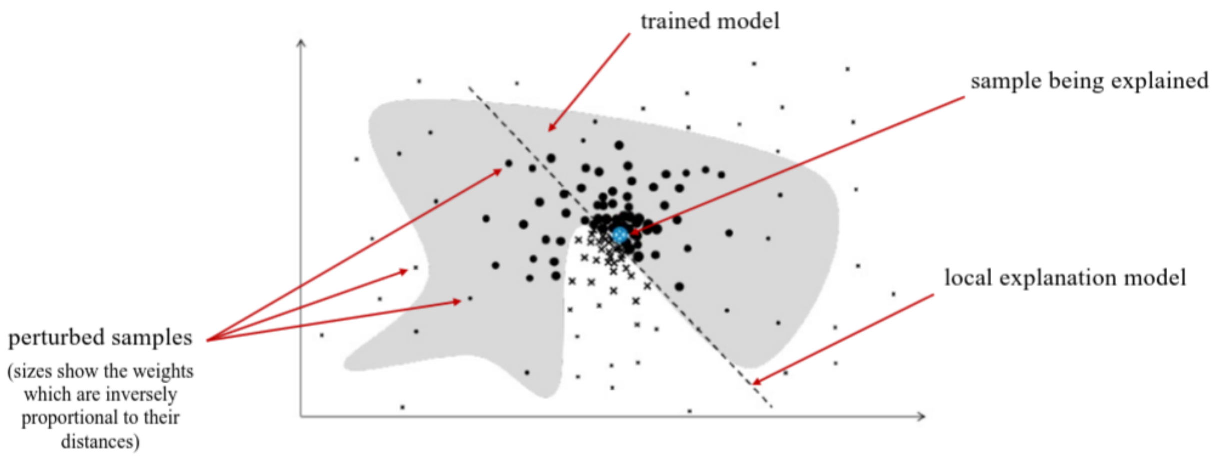


Figure 5: Lime working

### 14.2 SP-LIME

Submodular pick LIME takes some local explanations and constructs a global explanation

- Compute Feature Importance for feature j and sample i

$$I_j = \sqrt{\sum_{i=1}^{n} |W_{ij}|}$$

- define a coverage function c(V, W, I)
- greedily pick n explanations which increase coverage the most

## 15 SHAP

SHAP gives each feature an importance value for a local prediction. slower than lime, but more stable and guarantees consistency

## 16 Counterfactual Explanations

Local explanation (seen before like LIME and SHAP). Tells you what you need to change for one sample to give it the opposite decision (changes to data required to make -1 '+1')

### 16.1 LPP

minimize distance from counterfactual
h is the fitted model

$$mind(\hat{x}, x)$$
$$s.t. \hat{h}(x) \geq threshold,$$
$$x \in X$$

### 16.2 What an explanation needs

- Proximity: how close counterfactual is to x must be nearby
- Sparsity : CE should differ from x in few features
- Coherence: CE should be mapped back to input feature space after one hot encoding
- Actionability: Has to be things an individual can acutally change
- Data Manifold Closeness : CEs should be close to the observed data
- Causality: Any known causal relationships must be reflected
- Diversity: A set of explanations which differ in atleast one feature

## 17 Optimization with Constraint Learning

Trust region constraints: basically force the solution to be in the vicinity of the observed data and then you add other constraints so that the solution of the lpp has everything an explanation needs

## 18 Robust Counterfactual Explanation

Provide infinitely many solutions
Derive a method which guarantees full robustness
Set an uncertainty set for all counterfactuals which could be solutions

**Note 2.** Problem: This causes infinite constraints, with no dual possibility

### 18.1 Master and Adversarial Approach

master: solve the problem with a subset of the constraints, adversarial: find the constraint with max violation, and then put the max violated back into the master problem and repeat

## 19 Symbolic Regression

Find a equation that best fits the dataset

### 19.1 Genetic Programming

Function can be represented as a tree, and then new trees are made by mixing leaves with a genetic algorithm

### 19.2 Linear Optimization

### 19.3 ECSEL (Explainable Classification via Signomial Equation Learning)

Signomial function is a sum of monomials