Introduction to Responsible Machine Learning*

Lecture 1: Self-explainable Machine Learning Models

Patrick Hall

The George Washington University

May 14, 2022

Contents

Grading and Policy

- Grading:
  - $\frac{6}{10}$ Weekly Assignments
  - $\frac{3}{10}$ GitHub model card (Mitchell et al., 2019)
  - $\frac{1}{10}$ Class participation
- Project:
  - HMDA data using techniques from class
  - Individual or group (no more than 4 members)
  - Groups randomly assigned by instructor, with consideration of time zone
- Syllabus
- Webex office hours: 8:30 PM Tues (??)
- Class resources: https://jphall663.github.io/GWU_rml/

Overview

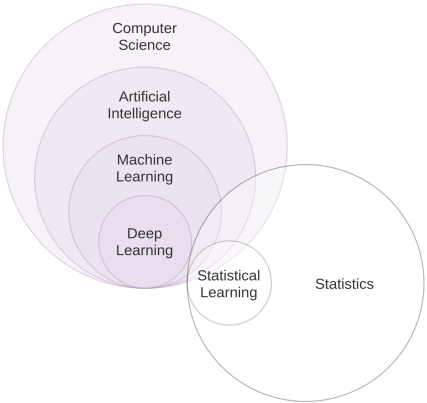
- **Class 1**: Self-explainable Models
- **Class 2**: Post-hoc Explanations
- **Class 3**: Fairness
- **Class 4**: Security
- **Class 5**: Model Debugging
- **Class 6**: Best Practices

Responsible Artificial Intelligence

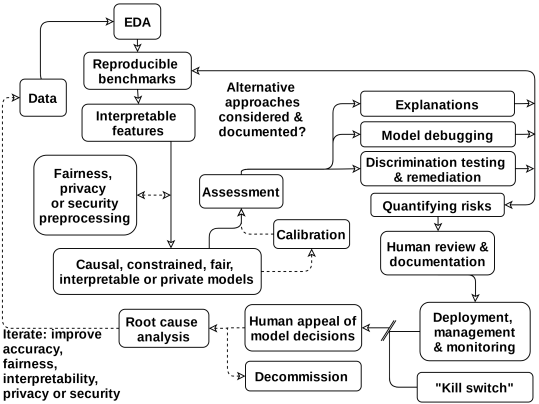
"Responsible Artificial Intelligence is about human responsibility for the development of intelligent systems along fundamental human principles and values, to ensure human-flourishing and well-being in a sustainable world."

— Virginia Dignum, ***Responsible Artificial Intelligence***

What About Machine Learning?

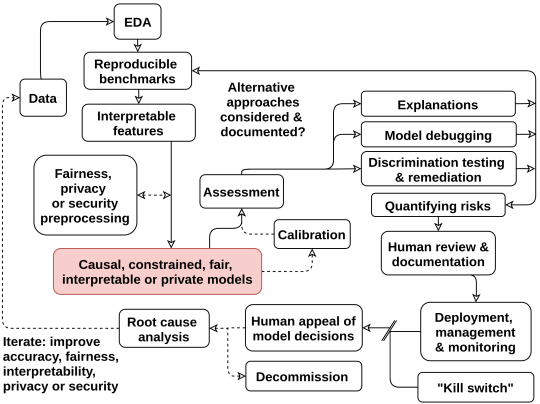


A Responsible Machine Learning Workflow



Source: *A Responsible Machine Learning Workflow*.

A Responsible ML Workflow: Self-explainable Models



Source: *A Responsible Machine Learning Workflow*.

Self-explainable ML Models

Interpretation: a high-level, meaningful mental representation that contextualizes a stimulus and leverages human background knowledge. An interpretable model should provide users with a description of what a data point or model output means *in context* (Broniatowski, 2021).

Explanation: a low-level, detailed mental representation that seeks to describe some complex process. An ML explanation is a description of how some model mechanism or output *came to be* (Broniatowski, 2021).

Self-explainable ML Models

There are many types of self-explainable ML models. Some might be directly interpretable to non-technical consumers. Some are only explainable to highly-skilled data scientists. Interpretability is not an on-and-off switch.

Self-explainable models are crucial for risk management, documentation, compliance, explanation of predictions to consumers, finding and fixing discrimination, and debugging other problems in ML modeling pipelines. Simply put, **it is very difficult to mitigate risks you don't understand**.

There is not necessarily a trade-off between accuracy and interpretability, especially for structured data.

Some Characteristics of Self-explainable ML Models

(Sudjianto and Zhang, 2021)

- **Additivity**: Whether/how model takes an additive or modular form. Additive decomposition of feature effects tends to be more explainable.
- **Sparsity**: Whether/how features or model components are regularized. Having fewer features or components tends to be more explainable.
- **Linearity**: Whether/how feature effects are linear. Linear or constant feature effects are easy to explain.
- **Smoothness**: Whether/how feature effects are continuous and smooth. Continuous and smooth feature effects are relatively easy to explain.
- **Monotonicity**: Whether/how feature effects can be modeled to be monotone. When increasing/decreasing effects are desired by expert knowledge they are easy to explain.
- **Visualizability**: Whether/how the feature effects can be directly visualized. Visualization facilitates the final model diagnostics and explanation.

Background

We will frequently refer to the following terms and definitions today:

- **Pearson correlation**: Measurement of the linear relationship between two input X_j features; takes on values between -1 and +1, including 0.
- **Shapley value**: a quantity, based in Game Theory, that accurately decomposes the outcomes of complex systems, like ML models, into individual components.
- **Partial dependence and individual conditional expectation (ICE)**: Visualizations of the behavior of X_j under some model g.

Background: Notation

Spaces

- Input features come from the set $\mathcal{X}$ contained in a P-dimensional input space, $\mathcal{X} \subset \mathbb{R}^P$. An arbitrary, potentially unobserved, or future instance of $\mathcal{X}$ is denoted x, $\mathrm{x} \in \mathcal{X}$.
- Labels corresponding to instances of $\mathcal{X}$ come from the set $\mathcal{Y}$.
- Learned output responses come from the set $\hat{\mathcal{Y}}$.

Background: Notation

Datasets

- The input dataset X is composed of observed instances of the set $\mathcal{X}$ with a corresponding dataset of labels Y, observed instances of the set $\mathcal{Y}$.
- Each i-th observation of X is denoted as $\mathrm{x}^{(i)} = [x_0^{(i)}, x_1^{(i)}, \ldots, x_{P-1}^{(i)}]$, with corresponding i-th labels in Y, $\mathrm{y}^{(i)}$, and corresponding predictions in $\hat{\mathrm{Y}}, \hat{\mathrm{y}}^{(i)}$.
- X and Y consist of N tuples of observations: $[(\mathrm{x}^{(0)}, \mathrm{y}^{(0)}), (\mathrm{x}^{(1)}, \mathrm{y}^{(1)}), \ldots, (\mathrm{x}^{(N-1)}, \mathrm{y}^{(N-1)})]$.
- Each j-th input column vector of X is denoted as $X_j = [x_j^{(0)}, x_j^{(1)}, \ldots, x_j^{(N-1)}]^T$.

Background: Notation

Models

- A type of machine learning (ML) model g, selected from a hypothesis set $\mathcal{H}$, is trained to represent an unknown signal-generating function f observed as X with labels Y using a training algorithm $\mathcal{A}$: $\mathrm{X}, \mathrm{Y} \xrightarrow{\mathcal{A}} g$, such that $g \approx f$.
- g generates learned output responses on the input dataset $g(\mathrm{X}) = \hat{\mathrm{Y}}$, and on the general input space $g(\mathcal{X}) = \hat{\mathcal{Y}}$.
- The model to be explained, tested for discrimination, or debugged is denoted as g.

Background: Gradient Boosting Machine

$$g^{\text{GBM}}(\mathbf{x}) = \sum_{b=0}^{B-1} T_b(\mathbf{x}; \Theta) \tag{1}$$

A GBM is a sequential combination of decision trees, T_b, where T_0 is trained to predict y, but all subsequent T are trained to reduce the errors of T_{b-1}.

Anatomy of Elastic Net Regression

Generalized linear models (GLM) have the same basic functional form as more traditional linear models, e.g. ...

$$g^{\text{GLM}}(x) = \beta_0 + \beta_1 x_0 + \beta_2 x_1 + \cdots + \beta_P x_{P-1} \tag{2}$$

... but are more robust to correlation, wide data, and outliers.

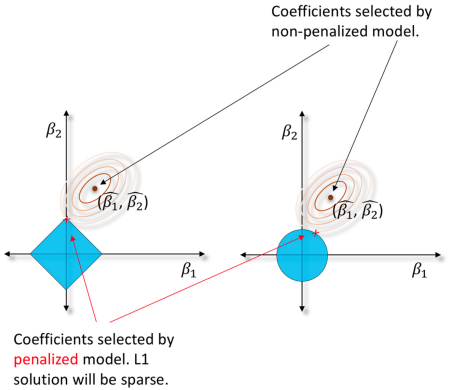
Anatomy of Elastic Net Regression: L1 and L2 Penalty

Iteratively reweighted least squares (IRLS) method with ridge (L_2) and LASSO (L_1) penalty terms:

$$\tilde{\beta} = \min_{\beta}\Big\{ \underbrace{\sum_{i=0}^{N-1}(y_i - \beta_0 - \sum_{j=1}^{P-1} x_{ij}\beta_j)^2}_{1} + \underbrace{\lambda}_{2} \sum_{j=1}^{P-1}\big(\underbrace{\alpha}_{3}\ \underbrace{\beta_j^2}_{4} + (1 - \underbrace{\alpha}_{3})\ \underbrace{|\beta_j|}_{5} \big) \Big\} \tag{3}$$

- 1: Least squares minimization
- 2: Controls magnitude of penalties
- 3: Tunes balance between L1 and L2
- 4: L_2/Ridge penalty term
- 5: L_1/LASSO penalty term

Graphical Illustration of Shrinkage/Regularization Method:



Generalized Additive Models and Explainable Boosting Machines

Generalized additive models (GAMs, Friedman, Hastie, and Tibshirani, 2001) extend GLMs by allowing an arbitrary function for each X_j:

$$g^{\text{GAM}}(\mathbf{x}) = \beta_0 + \beta_1 g_0(x_0) + \beta_2 g_1(x_1) + \cdots + \beta_P g_{p-1}(x_{P-1}) \tag{4}$$

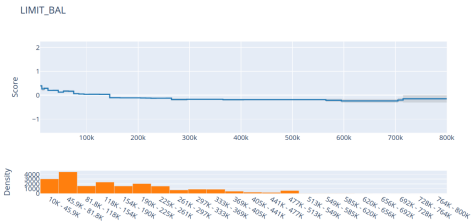
GAMs use spline approaches to fit each g_j.

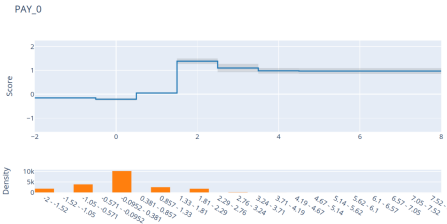
Later Lou et al., 2013 introduced an efficient technique for finding interaction terms ($\beta_{jk} g_{(j-1)(k-1)}(x_j \cdot x_k)$) to include in GAMs. This highly accurate technique was given the acronym GA2M.

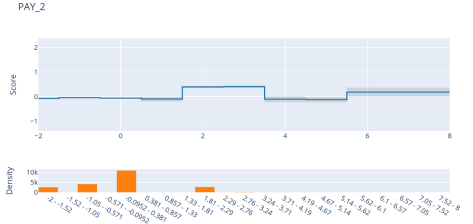
Recently Microsoft Research introduced the Explainable Boosting Machine (EBM) in the interpret package, in which GBMs are used to fit each g_j and g_{jk}. Higher order interactions are allowed, but used infrequently in practice.

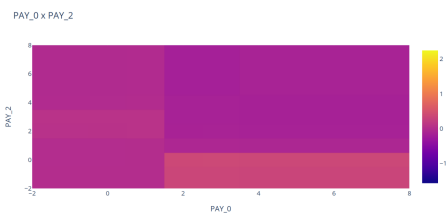
Because each input feature, or combination thereof, is treated separately and in an additive fashion, interpretability is very high.

Generalized Additive Models and Explainable Boosting Machines









Monotonic GBM (Gill et al., 2020)

Monotonic GBM (MGBM) constrain typical GBM training to consider only tree splits that obey user-defined positive and negative monotone constraints, with respect to each input feature, X_j, and a target feature, y, independently. An MGBM remains an additive combination of B trees trained by gradient boosting, T_b, and each tree learns a set of splitting rules that respect monotone constraints, Θ_b^{mono}. A trained MGBM model, g^{MGBM}, takes the form:

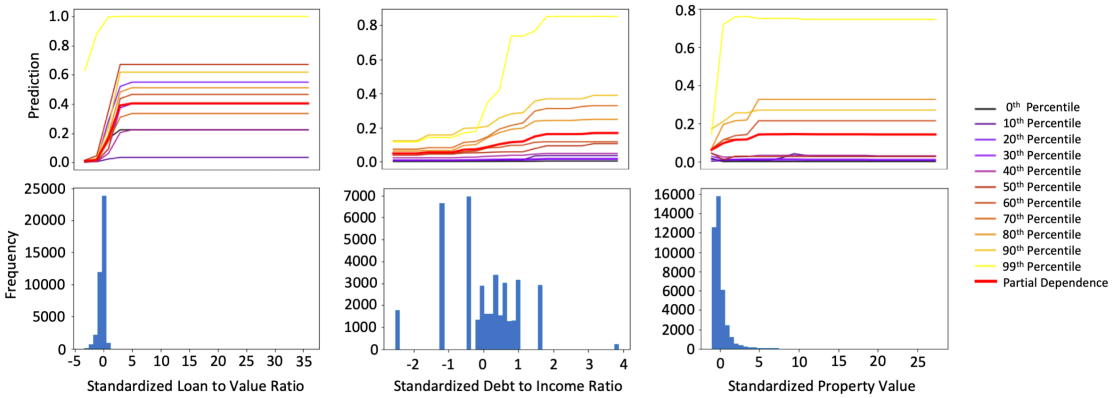
$$g^{\text{MGBM}}(\mathrm{x}) = \sum_{b=0}^{B-1} T_b\left(\mathrm{x}; \Theta_b^{\text{mono}}\right) \tag{5}$$

Monotone Constraints for GBM (Gill et al., 2020)

1. For the first and highest split in T_b involving X_j, any $\theta_{b,j,0}$ resulting in $T(x_j; \theta_{b,j,0}) = \{w_{b,j,0,L}, w_{b,j,0,R}\}$ where $w_{b,j,0,L} > w_{b,j,0,R}$, is not considered.
2. For any subsequent left child node involving X_j, any $\theta_{b,j,k\geq 1}$ resulting in $T(x_j; \theta_{b,j,k\geq 1}) = \{w_{b,j,k\geq 1,L}, w_{b,j,k\geq 1,R}\}$ where $w_{b,j,k\geq 1,L} > w_{b,j,k\geq 1,R}$, is not considered.
3. Moreover, for any subsequent left child node involving X_j, $T(x_j; \theta_{b,j,k\geq 1}) = \{w_{b,j,k\geq 1,L}, w_{b,j,k\geq 1,R}\}$, $\{w_{b,j,k\geq 1,L}, w_{b,j,k\geq 1,R}\}$ are bound by the associated $\theta_{b,j,k-1}$ set of node weights, $\{w_{b,j,k-1,L}, w_{b,j,k-1,R}\}$, such that $\{w_{b,j,k\geq 1,L}, w_{b,j,k\geq 1,R}\} < \frac{w_{b,j,k-\mathbf{1},L} + w_{b,j,k-\mathbf{1},R}}{2}$.
4. (1) and (2) are also applied to all right child nodes, except that for right child nodes $w_{b,j,k,L} \leq w_{b,j,k,R}$ and $\{w_{b,j,k\geq 1,L}, w_{b,j,k\geq 1,R}\} \geq \frac{w_{b,j,k-\mathbf{1},L} + w_{b,j,k-\mathbf{1},R}}{2}$.

Note that $g^{\text{MGBM}}(\text{x})$ is an addition of each full T_b prediction, with the application of a monotonic logit or softmax link function for classification problems. Moreover, each tree's root node corresponds to some constant node weight that by definition obeys monotonicity constraints, $T(x_j^{\alpha}; \theta_{b,0}) = T(x_j^{\beta}; \theta_{b,0}) = w_{b,0}$.

Partial Dependence and ICE:



A Burgeoning Ecosystem of Self-explainable Machine Learning Models

- Explainable Neural Network (XNN) (Vaughan et al., 2018)
- Rudin group:
  - *This looks like that* deep learning (Chen et al., 2019)
  - Scalable Bayesian rule list (Yang, Rudin, and Seltzer, 2017)
  - Optimal sparse decision tree (Hu, Rudin, and Seltzer, 2019)
  - Supersparse linear integer models (Ustun and Rudin, 2016)
  - and more ...
- rpart
- RuleFit (Friedman, Popescu, et al., 2008)
- skope rules

Model Selection

- Generally speaking, standard ML evaluation – including Kaggle leaderboards, are poor ways to assess ML model performance.
- However, Caruana, Joachims, and Backstrom, 2004 puts forward a robust model evaluation and selection technique based on cross-validation and ranking.

Fold	Metric	best_glm Value	best_mgbm Value	gbm11 Value	best_glm Rank	best_mgbm Rank	gbm11 Rank
0	F1	0.533181	0.551298	0.562353	3.0	2.0	1.0
0	accuracy	0.816246	0.817367	0.814006	2.0	1.0	3.0
0	auc	0.738625	0.776026	0.777570	3.0	2.0	1.0
0	logloss	0.468678	0.440775	0.438078	3.0	2.0	1.0
0	mcc	0.419924	0.420105	0.426918	3.0	2.0	1.0
1	F1	0.540865	0.554762	0.555283	3.0	2.0	1.0
1	accuracy	0.823882	0.826063	0.828244	3.0	2.0	1.0
1	auc	0.729674	0.776877	0.785956	3.0	2.0	1.0
1	logloss	0.465999	0.434170	0.428677	3.0	2.0	1.0
1	mcc	0.432722	0.445354	0.447637	3.0	2.0	1.0
2	F1	0.500593	0.516364	0.530343	3.0	2.0	1.0
2	accuracy	0.830907	0.833707	0.835946	3.0	2.0	1.0
2	auc	0.707507	0.760838	0.769493	3.0	2.0	1.0

Three models are ranked across different metrics and folds. The model with the highest rank, on average, across metrics and folds is the best model, `gbm11` in this case.

Acknowledgments

Thanks to Lisa Song for her assistance in developing these course materials.

References

Broniatowski, David A. (2021). "Psychological Foundations of Explainability and Interpretability in Artificial Intelligence." In: URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=931426.

Caruana, Rich, Thorsten Joachims, and Lars Backstrom (2004). "KDD Cup 2004: Results and Analysis." In: *ACM SIGKDD Explorations Newsletter* 6.2. URL: https://www.cs.cornell.edu/people/tj/publications/caruana_etal_04a.pdf, pp. 95–108.

Chen, Chaofan et al. (2019). "This Looks Like That: Deep Learning for Interpretable Image Recognition." In: *Proceedings of Neural Information Processing Systems (NeurIPS)*. URL: https://arxiv.org/pdf/1806.10574.pdf.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2001). ***The Elements of Statistical Learning***. URL: https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf. New York: Springer.

Friedman, Jerome H., Bogdan E. Popescu, et al. (2008). "Predictive Learning Via Rule Ensembles." In: *The Annals of Applied Statistics* 2.3. URL: https://projecteuclid.org/download/pdfview_1/euclid.aoas/1223908046, pp. 916–954.

Gill, Navdeep et al. (2020). "A Responsible Machine Learning Workflow with Focus on Interpretable Models, Post-hoc Explanation, and Discrimination Testing." In: *Information* 11.3. URL: https://www.mdpi.com/2078-2489/11/3/137, p. 137.

References

Hu, Xiyang, Cynthia Rudin, and Margo Seltzer (2019). "Optimal Sparse Decision Trees." In: *arXiv preprint arXiv:1904.12847*. URL: https://arxiv.org/pdf/1904.12847.pdf.

Lou, Yin et al. (2013). "Accurate Intelligible Models with Pairwise Interactions." In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.352.7682&rep=rep1&type=pdf. ACM, pp. 623–631.

Mitchell, Margaret et al. (2019). "Model Cards for Model Reporting." In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. URL: https://arxiv.org/pdf/1810.03993.pdf, pp. 220–229.

Sudjianto, Agus and Aijun Zhang (2021). "Designing Inherently Interpretable Machine Learning Models." In: *arXiv preprint arXiv:2111.01743*. URL: https://arxiv.org/pdf/2111.01743.pdf.

Ustun, Berk and Cynthia Rudin (2016). "Supersparse Linear Integer Models for Optimized Medical Scoring Systems." In: *Machine Learning* 102.3. URL: https://users.cs.duke.edu/~cynthia/docs/UstunTrRuAAAI13.pdf, pp. 349–391.

Vaughan, Joel et al. (2018). "Explainable Neural Networks Based on Additive Index Models." In: *arXiv preprint arXiv:1806.01933*. URL: https://arxiv.org/pdf/1806.01933.pdf.

Yang, Hongyu, Cynthia Rudin, and Margo Seltzer (2017). "Scalable Bayesian Rule Lists." In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. URL: https://arxiv.org/pdf/1602.08610.pdf.