Machine Learning: Overview and Applications

Vijay Nair

Outline

- Machine Learning: Brief overview
- Supervised ML :
  - Algorithms
  - Applications in Banking
- Opportunities
- Challenges
  - Interpretability

References

- Hu, L., et al. (2021) Supervised Machine Learning Techniques: **An Overview** with Applications to Banking*, International Statistical Review*, Volume 89 (Issue3) pages, p.573 - 604.
- Breiman, L. (2001a). **Random Forests.** *Machine Learning, 45*, 5-32.
- Breiman, L. (2001b). **Statistical Modeling: The Two Cultures** (with comments and a rejoinder by the author). *Statistical Science, 16*, 199-231.
- Friedman, J. (2001). Greedy Function Approximation: **A Gradient Boosting Machine**. *The Annals of Statistics, 29*, 1189-1232.
- Chen, T., & Guestrin, C. (2016). **XGBoost:** A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* San Francisco.
- Goodfellow, I., Bengio, Y., & Courville, A. (2015). ***Deep Learning****.* Cambridge, MA: MIT Press.
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, (pp. 4765-4774).
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " Why should i trust you?" **Explaining the predictions** of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, (pp. 1135-1144).
- Song, E., Nelson, B. L., & Staum, J. (2016). **Shapley Effects for Global Sensitivity Analysis**: Theory and Computation. *J. Uncertainty Quantification, 4*(1), 1060-1083.

Machine Learning and Artificial Intelligence

- **Machine Learning:**
  - Term coined by Arthur Samuel (IBM) in 1959
  - ML gives computers the ability to learn without being explicitly programmed
  - Study and construction of algorithms that can learn from data, identify features, recognize patterns, make predictions, and take actions
  - A key pathway to AI

- **Artificial Intelligence:** concerned with making computers behave like humans
  - Term coined by John McCarthy (MIT) around 1956
  - Study of " intelligent agents" [or systems] that "perceive" the environment and take actions that maximize [probability] of success [to achieve] some goal
  - Long history: formal reasoning in philosophy, logic, …
  - Resurgence of AI techniques in the last decade: advances in computing power, computing and data architectures, sizes of training data, and theoretical understanding
  - Deep Learning Neural Networks: At the core of recent advancements in AI, specifically for certain classes of ML tasks (Reinforcement L and Representation L)

Machine Learning Tasks

- Supervised Learning
  - Data with **“labels”**
  - **Regression and classification**

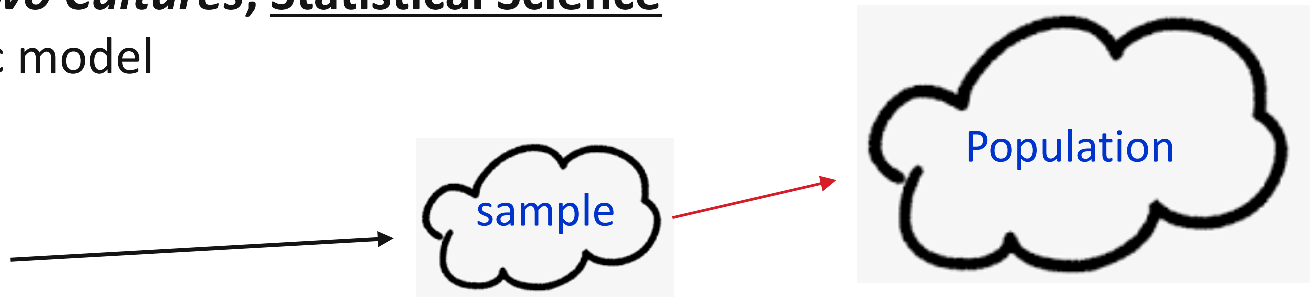
- Unsupervised Learning
  - Data with **no labels**
  - **Discover patterns or structure** in the data (anomalies, clusters, lower-dimensional representation)

- Reinforcement Learning
  - **Experiment** and **exploit** to **make “optimal” decisions** based on **reward** structure

- Others
- Semi-supervised, Positive-Unlabeled Learning, …
- Representation Learning
- Transfer Learning

Supervised Learning: Statistics vs ML paradigms

- **Leo Breiman (2001)** ***Statistical Modeling: The Two Cultures***, **Statistical Science**
  - Two paradigms: data model and algorithmic model



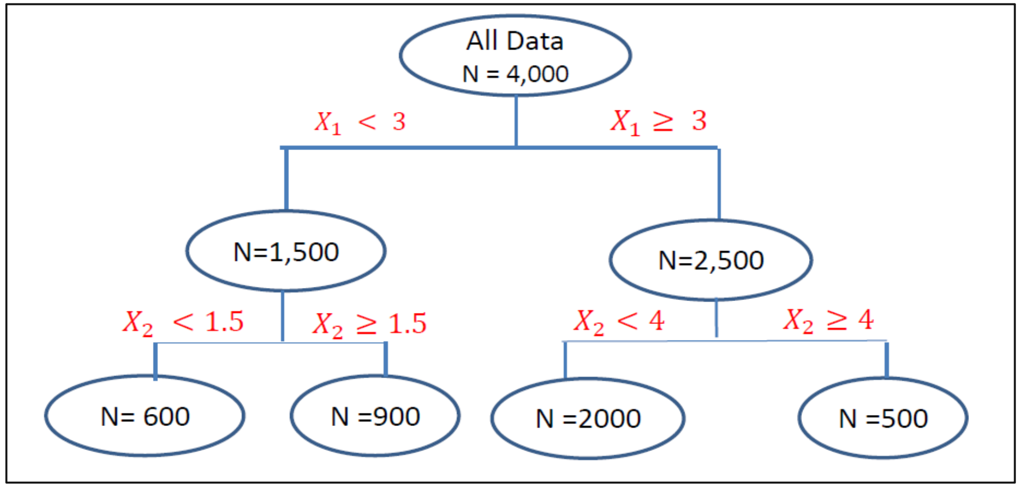
- Traditional statistics
  - Goal: “understand” the generative model
    - **Estimate** model **parameters** and **assess uncertainty**
    - Identify **key drivers** and **input-output relationships**
    - **Extensive tools and diagnostics** developed over time
    - Parametric **models** → **easier to interpret**

- Machine Learning
  - Goal: best predictive performance … generalization assessed on hold-out data
    - **Algorithmic** approach and **automation** of model building
      → variable selection, feature engineering, model training
    - Large samples
    - Not much focus on CI, hypothesis testing, …
  - **No** intrinsic **interest in** the **data generation process** (even if there’s such a thing!)

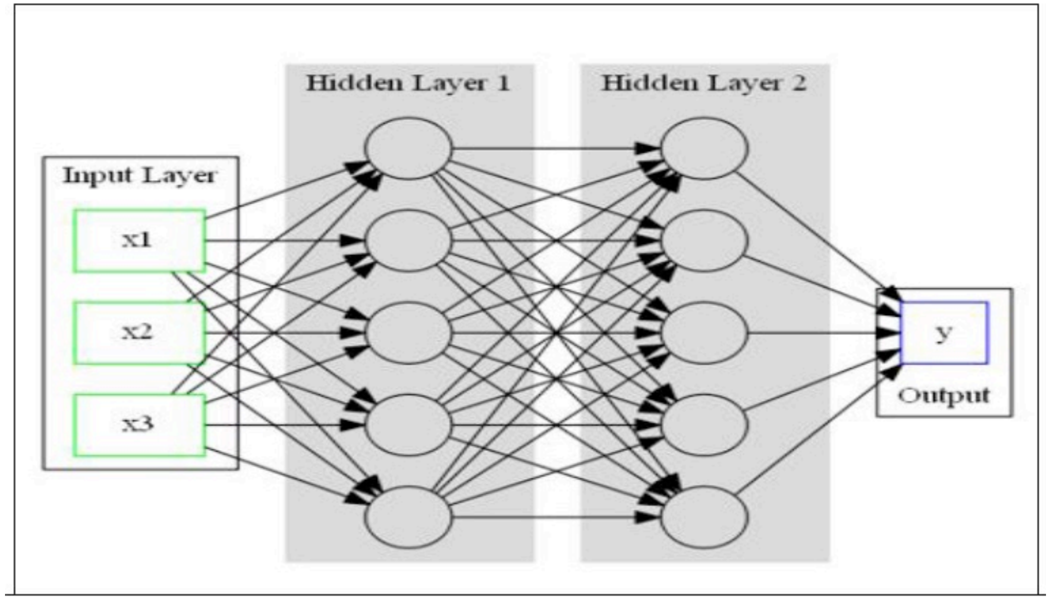
- **For regulated industries and safety-critical applications:**
  - Model interpretability is important

Supervised ML Algorithms

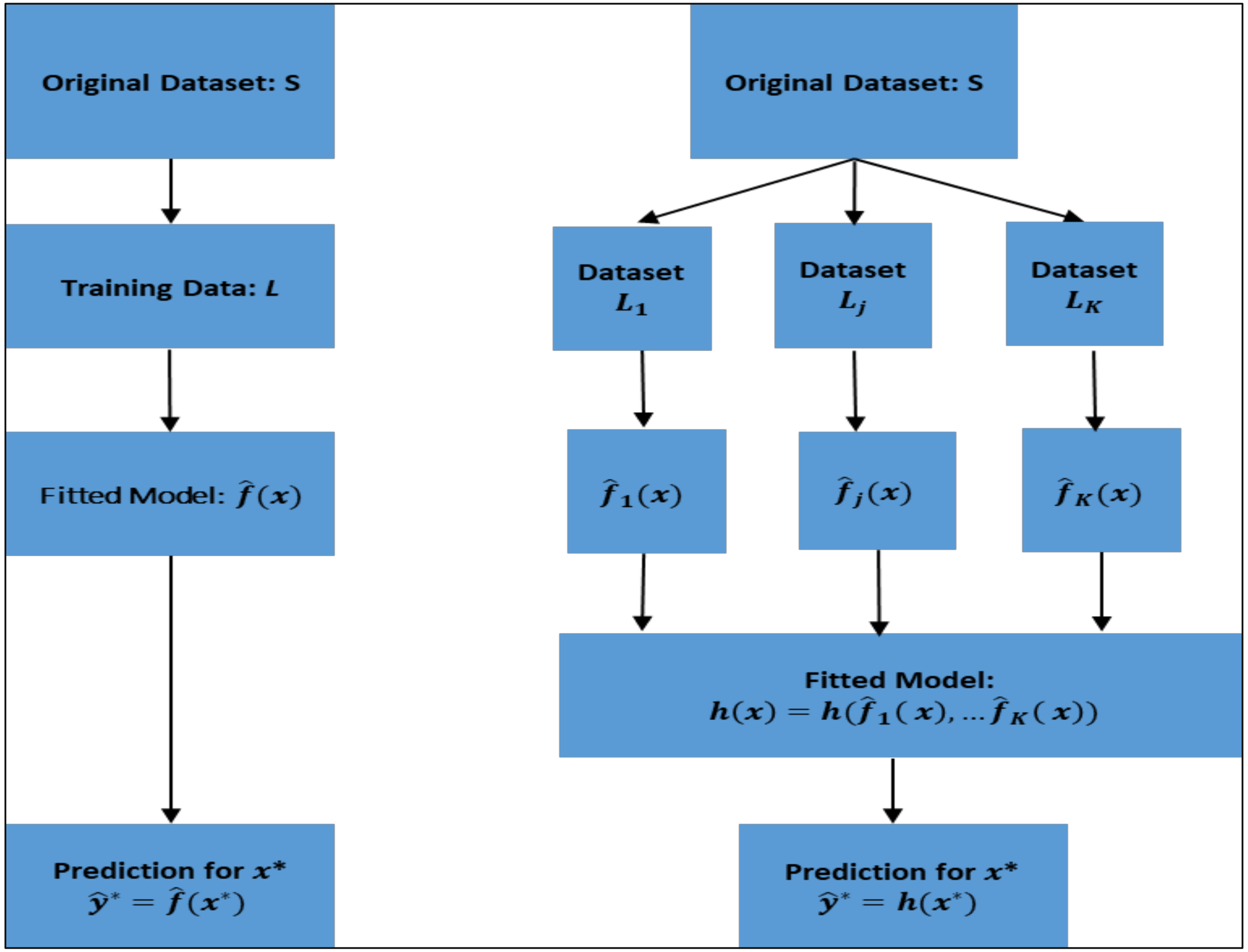
- **Ensemble algorithms**
  - Random Forests (RFs)
  - Gradient Boosting Machines (GBMs)
    - eXtreme Boosting (XGBoost)
  - Tree-based models
  - Piecewise constant within nodes



- **Feedforward Neural Networks**



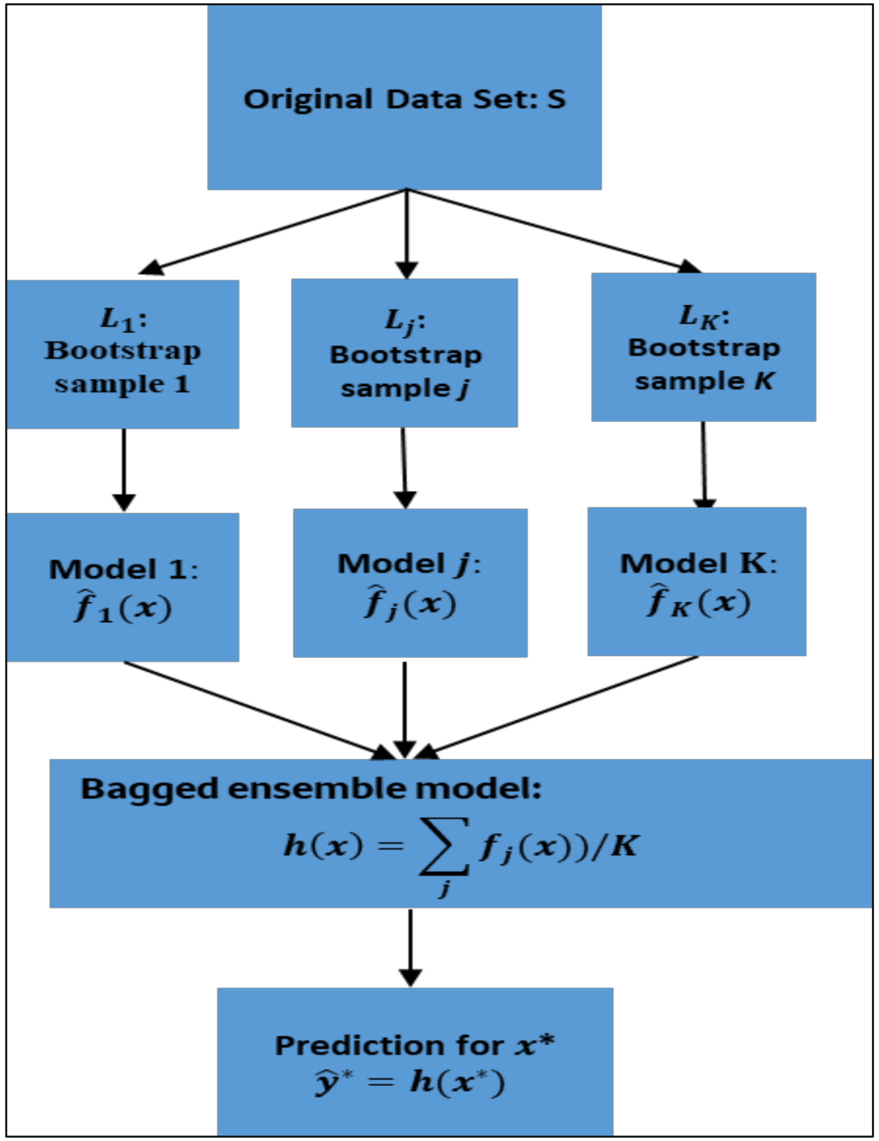
Ensemble Algorithms



Improve performance by **combining** outputs of **several individual algorithms** ("weak learners"):

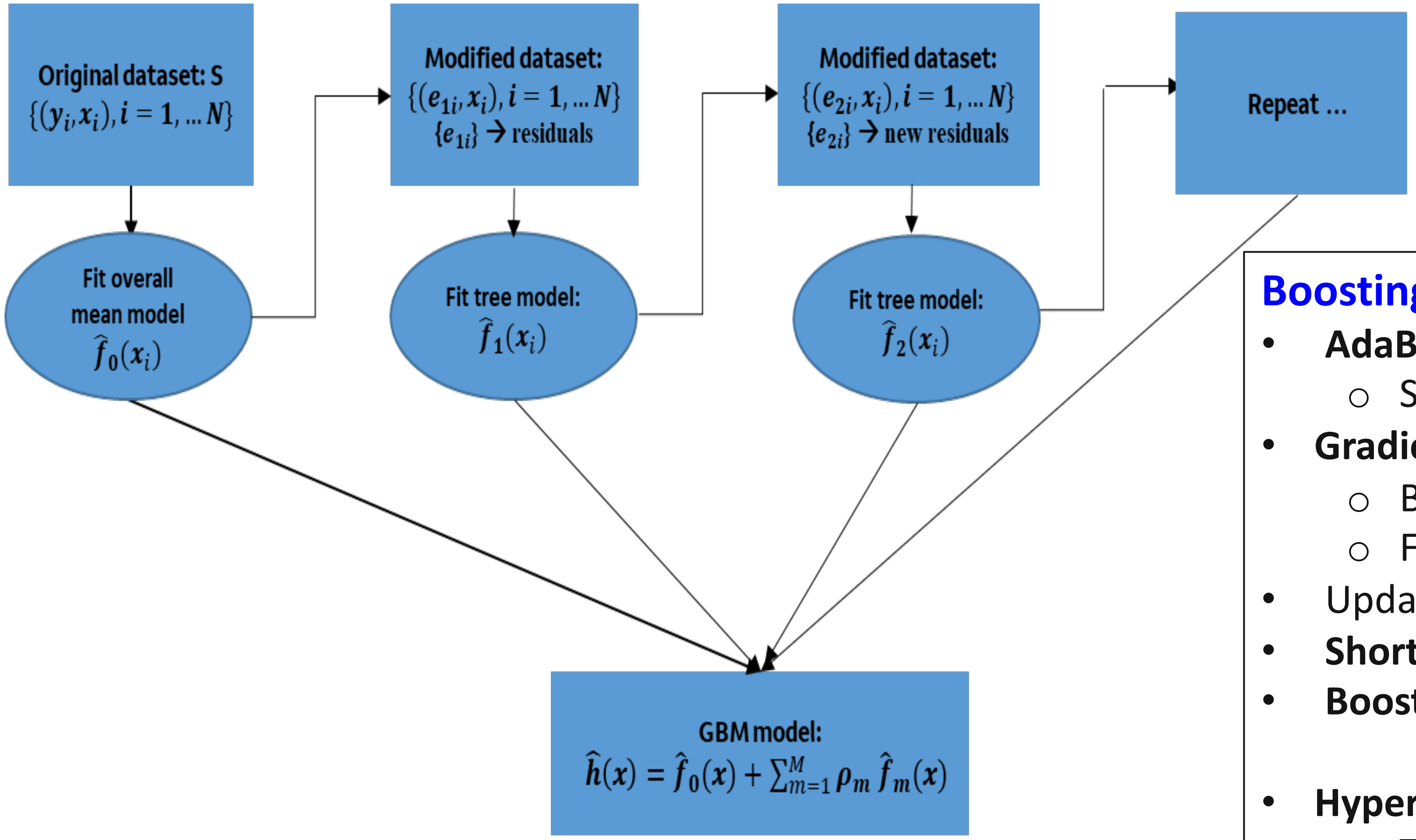
- **Bagging and Random Forest**
- **Boosting**
- **Other ensemble approaches:**
  - Model Averaging
  - Majority Voting
  - Stacking

Random Forest



- **Random Forest** (Breiman 2001)
  - Create **multiple datasets** by **bootstrap sampling of rows**
  - Build **deep trees** for each dataset
    - → fit piecewise constant models
    - → each tree has small bias (deep) but large variance
  - **Average** results across trees
    - → **reduce variance** and instability
- **Bootstrap aggregating** (bagging)
  - Column sub-sampling
    - → reduce correlations across trees
- **Hyper-parameters**
  - Tree depth
  - Number of trees
  - Row sampling ratio
  - Colum sampling ratio

Gradient Boosting Machine

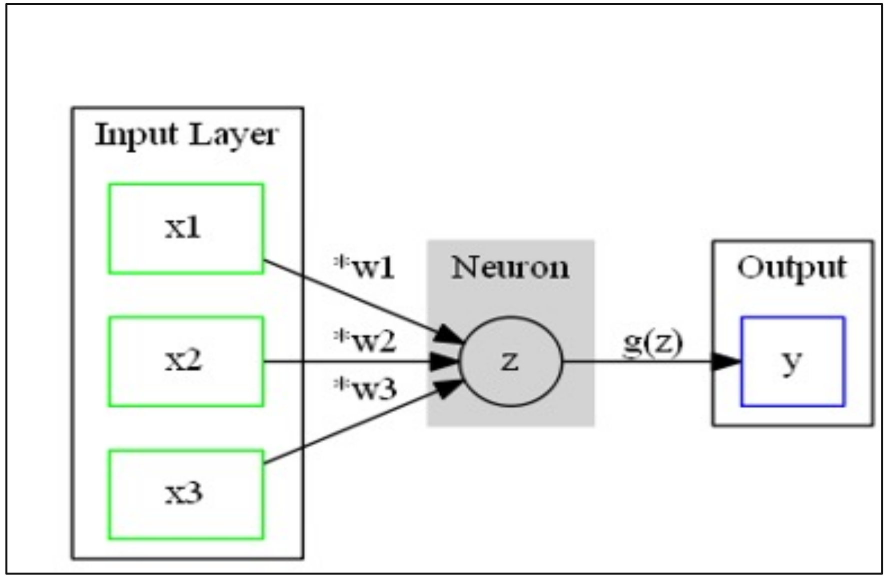


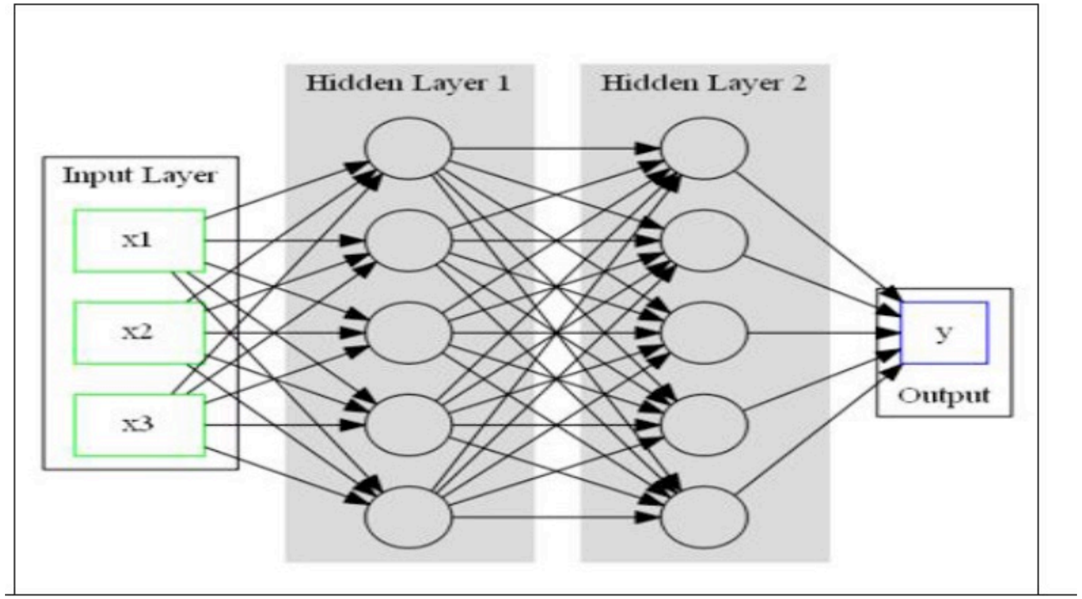
Boosting

- **AdaBoost**
  - Schapire (1990), Freund and S (1995)
- **Gradient boosting**
  - Breiman (1996), Friedman (2001)
  - Fit trees to **residuals sequentially**
- Updates in the **direction of negative gradient**
- **Short trees** → low variance, big bias
- **Boosting reduces bias**

- **Hyper-parameters** (same as RF)
  - Tree depth
  - Number of trees
  - Learning rate
  - Row sampling ratio
  - Colum sampling ratio

Feedforward Neural Networks (FFNN)





- **Mimic neuronal networks**

- **Activation function:** $\boldsymbol{g(w^T x)}$
  - Sigmoidal, Hyperbolic Tan, ReLU
  - Connection to **additive index models:**
    $f(\boldsymbol{x}) = g(w_1 x_1 + \ \dots + w_P x_P)$

- **FFNN architecture**
  - Nodes (Neurons)
  - Input, Output, and Hidden Layers
  - All nodes connected with others in next layer
- **Deep NNs**
  - Many layers
  - CNN, RNN, LSTM, …
  - BERT (Bidirectional Encoder Representations from Transformers)

Hyper-parameter Optimization

- **Batch or non-sequential techniques**
  - Grid search
  - Random Search
  - Designed experiments

- **Sequential Search**
  - Hyperband
  - Sequential model-based global optimization techniques
    - Bayesian optimization with Gaussian Process
    - Tree-structured Parzen estimator

- Can be time consuming with large number of hyper-parameters and datasets
- Need access to good computing environment

Applications in Banking

Areas:

- **Credit Risk:** Predicting **losses** – customers **not repaying debts or loans**: Mortgages, Auto-Loans, Student Loans, Credit cards, Small businesses, …
- **Credit Decisions: Activities related to loan applications**: credit scoring, marketing, collections, …
- **Revenue and Transactions:** Interest, servicing fees, deposits, withdrawals, electronic payments, etc.
- **Financial Crimes:** Fraud detection, Money laundering
- **Fair Lending**: Ensuring fair treatment of customers
- **Text and speech**: Conversations, complaints, emails, voice messages, chat-bots for assisting customers and employees

Statistical Techniques

- Dimension reduction; clustering, anomaly detection
- Parametric modelling for regression and classification
- Semi- and non-parametric regression models
- Regularization: Lasso, ridge, …
- Survival analysis; Time series forecasting

New Focus:

- Account level data → very large datasets with 100's of millions of observations and 1,000s of predictors
- Emphasis on "automated" feature engineering and model development
- Modeling new sources and types of data

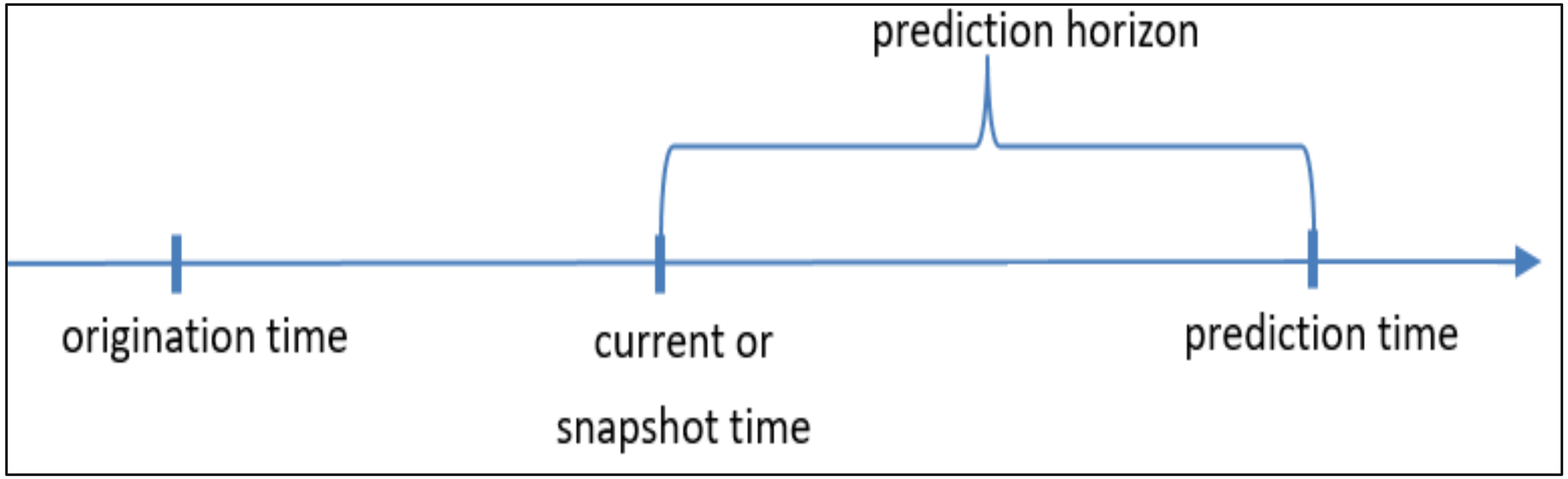
ML/AI Techniques:

- Auto-encoders, GANs, …
- Ensemble Tree-Based Algorithms: RFs and GBMs
- Feedforward Neural Networks
- Deep NNs for Natural Language Processing and Time Series Data

Application to Home Mortgage: Modeling "In-Trouble" Loans

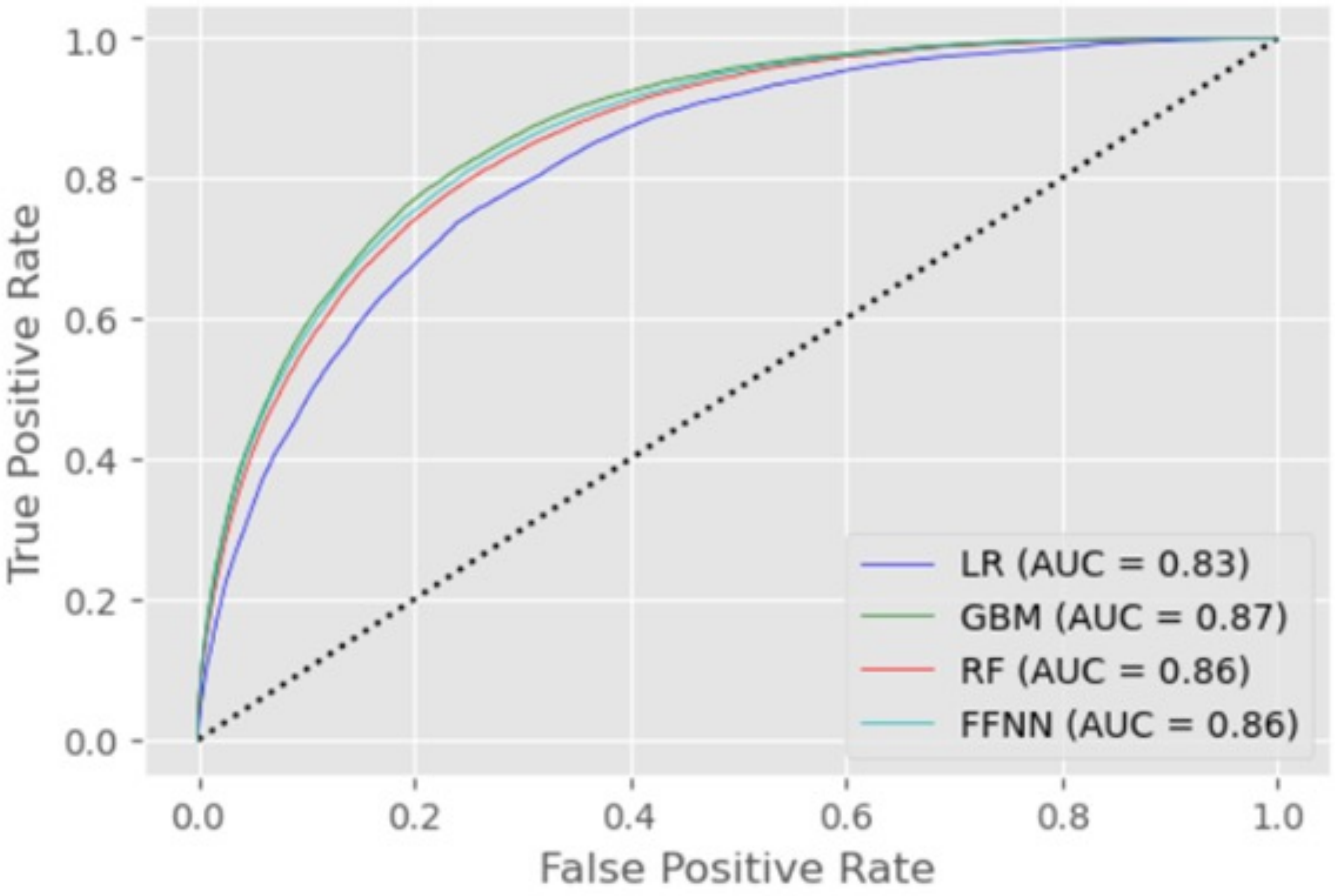
- **One portfolio: ~ 5 million observations**
- **Response: binary = loan is "in trouble"** (multiple failures and connections to competing risks)
- **20+ predictors:** credit history, type of loan, loan amount, loan age, loan-to-value ratios, interest rates at origination and current, loan payments up-to-date, etc. (origination and over time)

Modeling framework



Loan origination, current (snapshot) and prediction times

Comparison of Predictive Performance: ROC and AUC on Test Data



- **ML with 22 predictors**
- **LR model: eight carefully selected variables**
  - snapshot fico (credit history);
  - ltv (loan-to-value ratio);
  - ind_financial-crisis;
  - pred_unemp_rate;
  - two delinquency status variables;
  - horizon

How typical is this “lift” in our applications?

Natural Language Processing (NLP)

- Methods, algorithms, and systems for **analyzing "human language" data** (text, speech, conversations)
  - **Very challenging** …
- **Interdisciplinary area** that combines computer science, statistics, optimization, AI, linguistics, logic …
  - Earlier version → computational linguistics, speech recognition, …
- **Evolution:**
  - Rule-based, statistical …now **largely driven by deep neural networks**
- **Diverse applications**

Text Summarization

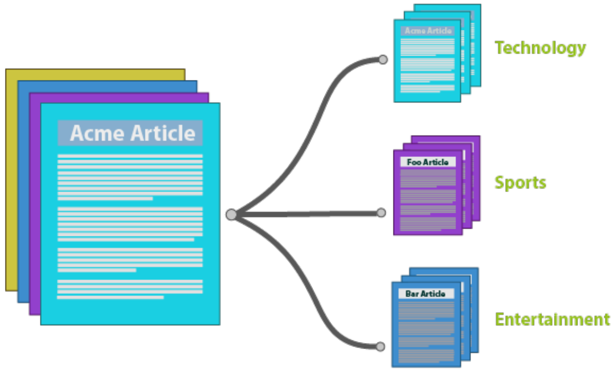
Machine Translation

Chatbots

- Alexa and Siri-like
- Conversational AI

Natural Language Generation

Text classification



Sentiment Analysis

Opportunities with ML

General:

- **Advent of "Big Data"**
  - ✓ **New sources of data**: social media, sensor networks, intelligent systems, …
    - Text, conversations, …

- **Advances in computing and data storage technologies**
  - ✓ Infrastructure for data collection, warehousing, transfer, and management
  - ✓ Efficient and scalable algorithms and associated technologies for analyzing large datasets
  - ✓ Open-source algorithms
  - ✓ Cloud storage and computing

    → Democratization of Data Science

Specific:

- Availability of large datasets and fast algorithms

  → flexible modeling … move away from restrictive parametric models
- SML algorithms:
  - ➢ Improved predictive performance
  - ➢ Semi-automated approach to feature engineering and model training → ideal for Big Data
- New data sources and computing technologies open up new opportunities
  - ➢ Text, speech, images, …
  - ➢ More timely information and decision making

ML: In Pursuit of Interpretability

- Major Challenge:
  - Predictor $\hat{f}(x)$ is implicitly defined, high-dimensional, and complex → hard to interpret results
  - Not an issue if only goal is prediction: recommender systems, fraud detection, …
  - Big issue for regulated industries and safety-critical applications
  - Typically dual goals: good predictive performance and interpretability

- Main Approaches:

I. Post hoc: Techniques for interpreting results after fitting model

II. Fitting and using surrogate models to explain complex results

   a) Born-again trees (piecewise constant) → Breiman
   b) Locally additive tress → Hu, Chen, Nair (2022)

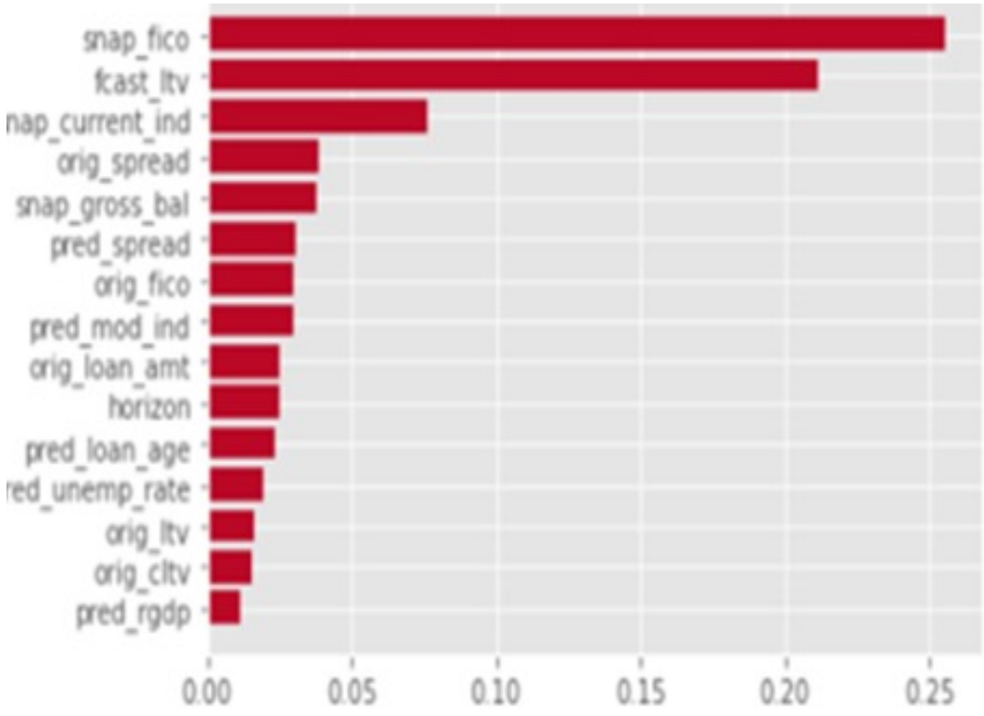
III. Inherently interpretable algorithms

Global: Variable Importance

Y	X1	X2	X3	X4	X5
2	1.5	0	4.5	10.2	3.0
4	2.7	1	5.3	8.7	4.2
8	3.3	1	7.2	19.3	17.6
3	1.9	0	3.3	7.8	21.2

- **Permutation based: Model agnostic**
  - Randomly permute the rows for variable (column) of interest while keeping everything else unchanged
  - Compute the change in prediction performance as the measure of importance.
- **Selected Others**
  - **Tree-based** importance metrics
    - Importance of a variable x_j → total reduction of impurity at nodes where x_j is used for splitting
    - For ensemble algorithms, average over all trees
  - **Global Shapley**
    - Based on Shapley decomposition (1953); Owen (2014)
    - Model agnostic but **computationally intractable**

Home Mortgage-XGB



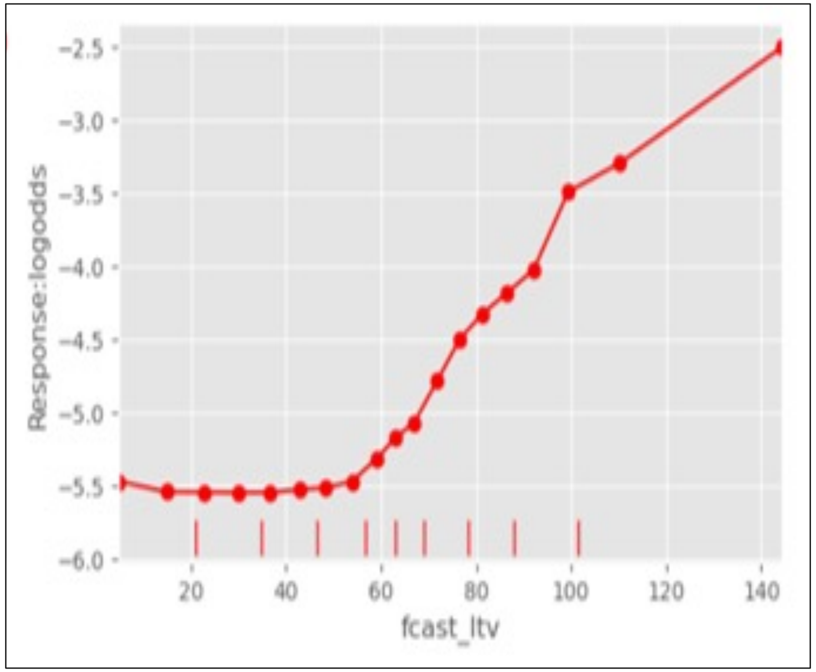
Input-Output Relationships: 1-D Partial Dependence Plots

- Understand how fitted response varies as a function of one or more variables of interest
- **One-dimensional Partial Dependence Plot (PDP)**
  - Variable of interest: x_j
  - Write the fitted model as $\hat{f}(x) = \hat{f}(x_j, \boldsymbol{x}_{-j})$
  - Fix x_j at c; compute the average of $\hat{f}$ over the entire data

$$g_j(x_j = c) = \frac{1}{N}\sum_{i=1}^{N} \hat{f}(x_j = c, x_{-j,i})$$

  - Plot $g_j(x_j)$ against x_j over a grid of values
  - One-dimensional summary
  - Interpretation: Effect of x_j averaged over other variables

Home Mortgage
1-D PDP for forecast_LTV



Local Explainability

- **Questions of Interest:**
  1. How can we interpret the response surface locally at selected points of interest?
  2. Given the predicted value at a point of interest $\hat{f}(\boldsymbol{x}^*) = \hat{f}(x_1^*, \dots, x_K^*)$, what are the contributions of the different variables $\{x_1, \dots x_K\}$ to the prediction?

- If fitted model is linear: $\hat{f}(\boldsymbol{x}) = b_0 + b_1 x_1 + \ \dots b_K x_k$, we can answer both questions using the regressions coefficients.

- Answer to 1: Model is linear → magnitudes and signs of regression coefficients provide explanation

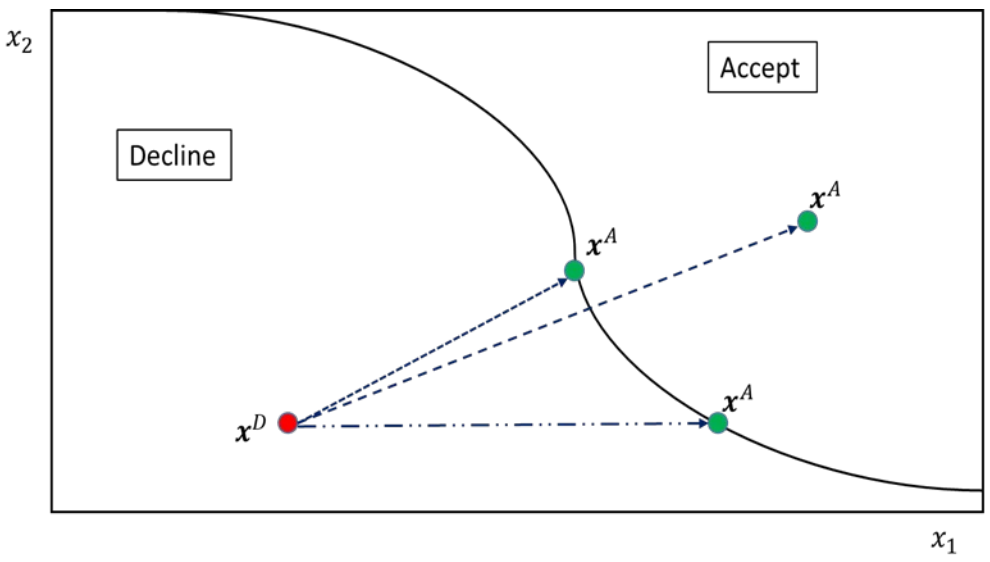
- Answer to 2: Contribution of $\boldsymbol{x_j^*}$ is $\boldsymbol{b_j x_j^*}$

- How to extend these interpretations to fitted models from complex ML algorithms?

- LIME, SHAP, B-Shap, etc.

“Adverse” action explanation on declined decisions to customers

- $\boldsymbol{x} = (x_1, \dots, x_K)$ K −dimensional credit attribute
- Use historical data $\{y_i, \boldsymbol{x}_i\}, i = 1, \dots n$ to develop model for probability of default (PoD)
- Fitted model for PoD – $p(\boldsymbol{x})$
- **Decision:**
  - Accept application with $\boldsymbol{x}^*$ if $p(\boldsymbol{x}^*) \leq \boldsymbol{\tau}$;
  - Decline otherwise
- Declined customers are entitled to an “explanation” by law



- **Problem formulation**
  - Take a **reference point** $\boldsymbol{x}^A$ in the “accept” region
  - Compute the difference: $[\boldsymbol{p}(\boldsymbol{x}^{\boldsymbol{D}}) - \boldsymbol{p}(\boldsymbol{x}^{\boldsymbol{A}})]$
  - **Attribute** the difference **to** the (important) **predictors**
  - Better to do in terms of $f(\boldsymbol{x}) = logit\ p(\boldsymbol{x})$
  - Decompose $[f(\boldsymbol{x}^{\boldsymbol{D}}) - f(\boldsymbol{x}^{\boldsymbol{A}})] = E_1(\boldsymbol{x}^{\boldsymbol{D}}, \boldsymbol{x}^{\boldsymbol{A}}) + E_2(\boldsymbol{x}^{\boldsymbol{D}}, \boldsymbol{x}^{\boldsymbol{A}}) + \dots + E_K(\boldsymbol{x}^{\boldsymbol{D}}, \boldsymbol{x}^{\boldsymbol{A}})$

    $E_k(\boldsymbol{x}^{\boldsymbol{D}}, \boldsymbol{x}^{\boldsymbol{A}})$ is allocation to (contribution by) k −th predictor

General expression for AA with Baseline Shapley

$$[f(\boldsymbol{x}^{\boldsymbol{D}}) - f(\boldsymbol{x}^{\boldsymbol{A}})] = E_1 + \cdots + E_K,$$

$$\boxed{E_k = E_k(\boldsymbol{x}^{D};\, \boldsymbol{x}^{A}) = \sum_{\boldsymbol{S}_k \subseteq \boldsymbol{K} \setminus \{k\}} \frac{|\boldsymbol{S}_k|!\,(|\boldsymbol{K}| - |\boldsymbol{S}_k|)!}{|\boldsymbol{K}|!} \left(f\left(x_k^D;\, \boldsymbol{x}_{\boldsymbol{S}_k}^{\boldsymbol{D}};\, \boldsymbol{x}_{\boldsymbol{K}\setminus \boldsymbol{S}_k}^{\boldsymbol{A}}\right) - f\left(x_k^A;\, \boldsymbol{x}_{\boldsymbol{S}_k}^{\boldsymbol{D}};\, \boldsymbol{x}_{\boldsymbol{K}\setminus \boldsymbol{S}_k}^{\boldsymbol{A}}\right)\right).}$$

- Application of Shapley concept (Shapley, 1951+)
  - Adapted to global explanation in ML (Owen 2014; and others)
  - Local explanation (Lundberg et al. 2018, others)
  - Computationally intractable
- Baseline Shapley (Sundararajan, M. and Najmi, A. (2020) – easier to compute
- Adaptation to Adverse Action (Nair et al. 2022)

AA Explanation with Two Predictors

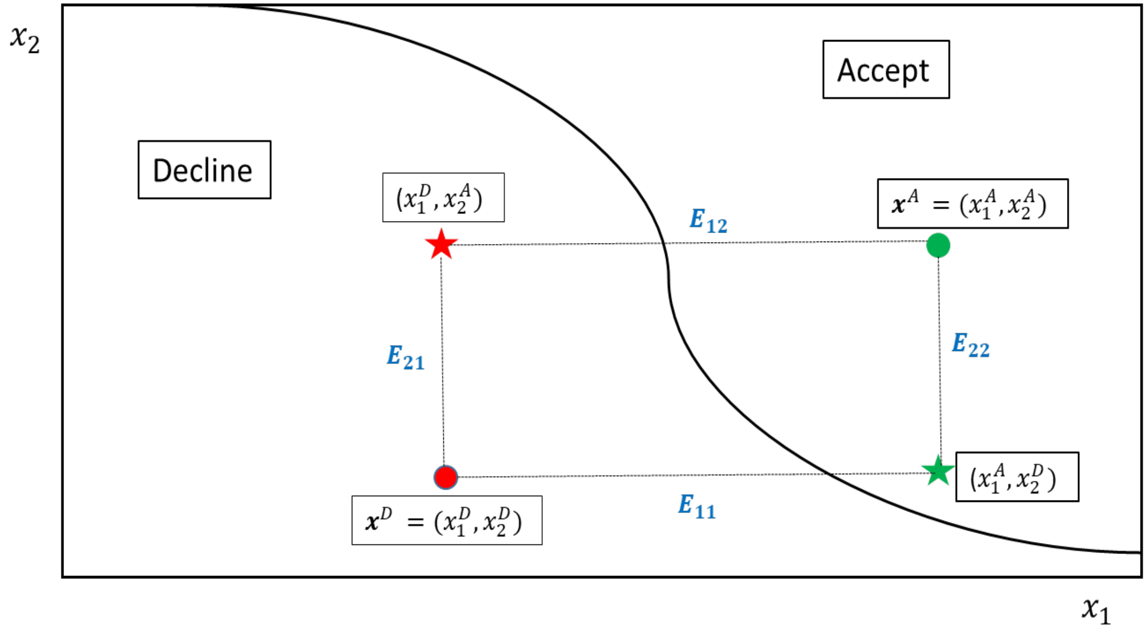
Linear model: $f(\boldsymbol{x}) = b_0 + b_1 x_1 + \cdots + b_K x_K$

$[f(\boldsymbol{x^D}) - f(\boldsymbol{x^A})] = b_1(x_1^D - x_1^A) + b_2(x_2^D - x_2^A) + \ \ldots$

GAM?

Interactions? $f(\boldsymbol{x}) = b_0 + b_1 x_1 + b_2 x_2 + b_{12} x_1 x_2$

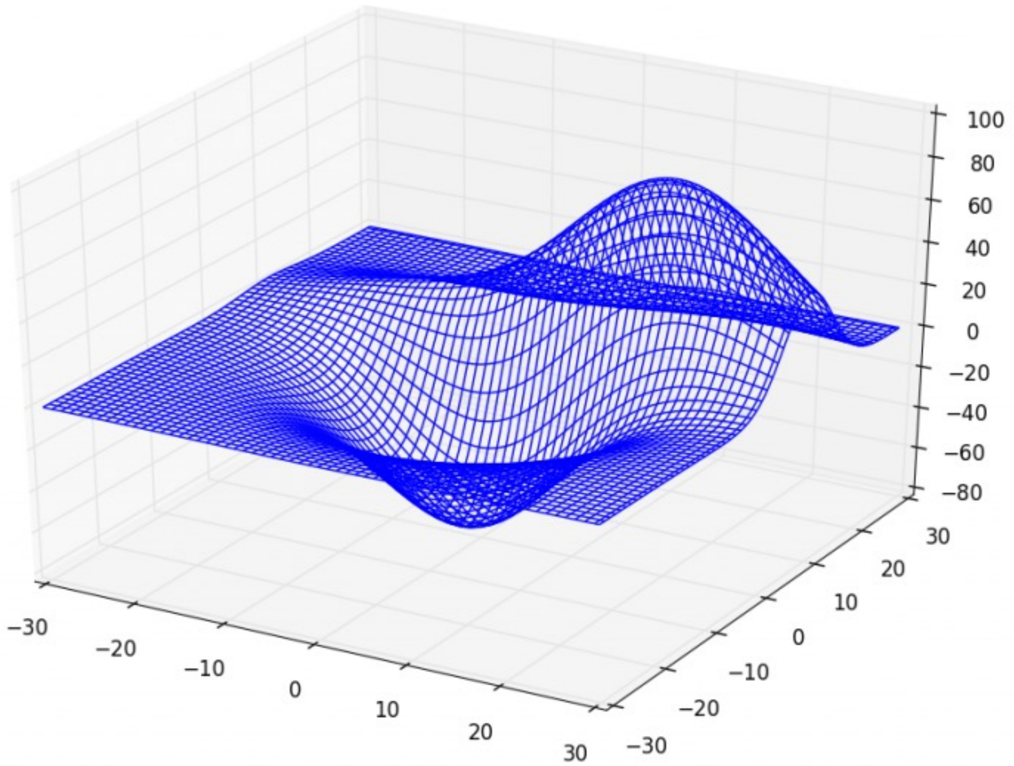
$b_1(x_1^D - x_1^A) + b_2(x_2^D - x_2^A) + b_{12}(x_1^D x_2^D - x_1^A x_2^A)$

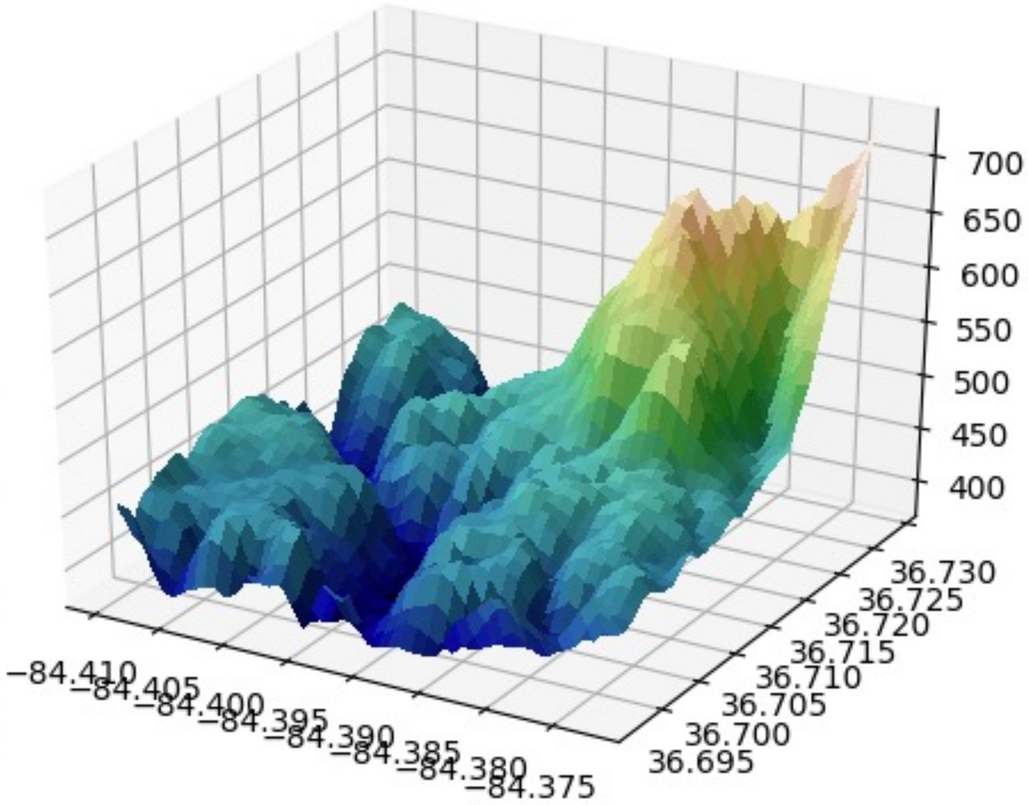


General: (Nair et al. 2022)

- $E_1 = \frac{1}{2}(E_{11} + E_{12}) \rightarrow \frac{1}{2}\{[f(x_1^D, x_2^D) - f(x_1^A, x_2^D)] + [f(x_1^D, x_2^A) - f(x_1^A, x_2^A)]\}$
- $E_2 = \frac{1}{2}(E_{21} + E_{22}) \rightarrow \frac{1}{2}\{[f(x_1^D, x_2^D) - f(x_1^D, x_2^A)] + [f(x_1^A, x_2^D) - f(x_1^A, x_2^A)]\}$

Issues





- **Most post-hoc tools** for studying input-output relationships are **lower-dimensional summaries**
  - **Limited** in **ability** to **characterize complex models** with local behavior
  - **Need better visualization** tools in high-dimensions
  - How to **automate visualization** → spirit of ML and AI.

- ML algorithms: **Function-fitting vs modeling**
  - High-dimensional ML – can do very good function fitting with large samples
  - What is a **role of a model**?

Correlation can create havoc!

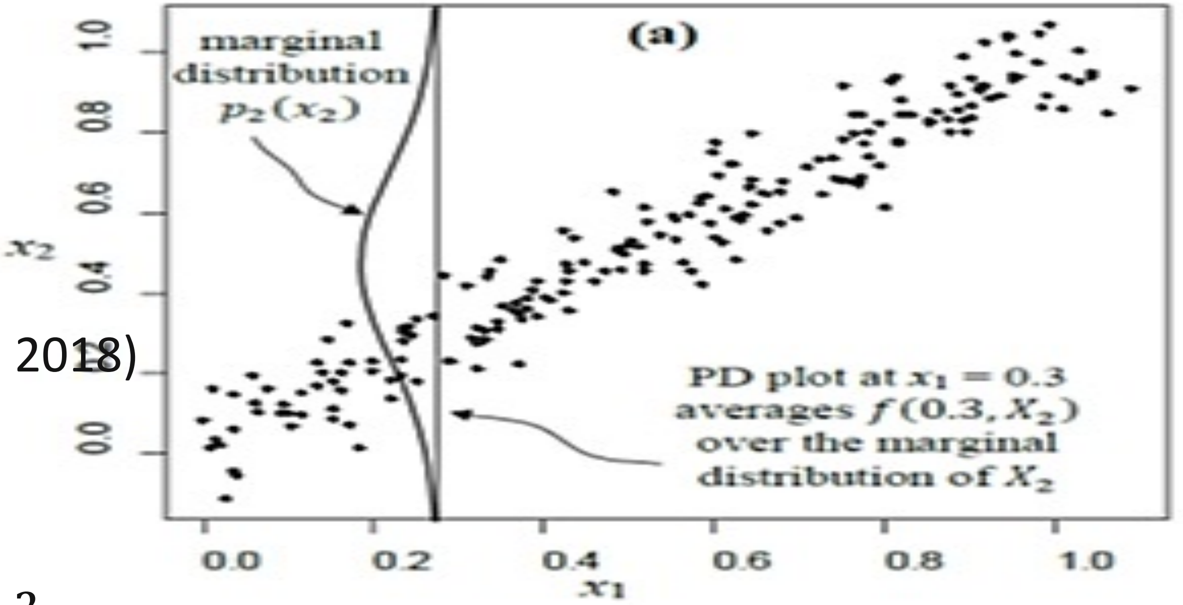
$\hat{f}(\boldsymbol{x}) = \hat{f}(\boldsymbol{x_j}, \boldsymbol{x_{-j}})$ is the fitted model

$$\hat{f}_{PD,j}(z) = \frac{1}{N}\sum_{i=1}^{N} \hat{f}(x_j = z, \boldsymbol{x_{-j,i}})$$

When predictors are highly correlated:

Performance of VI analyses and PDPs?

- Extrapolation
- Poor model fit outside data envelope
- Alternatives: ALE (Apley and Zhu, 2020), ATDEV (Liu et al. 2018)



Bigger issue: Model identifiability

$$f(x_1, x_2) = \beta_1 x_1 + \beta_2 x_2 + \beta_{12}\, x_1 x_2 \rightarrow g(x_1)\ ?$$

- Main effect → masked by quadratic term from interaction
- Different ML algorithms can capture the masking differently
- VI analysis → permute correlated variables jointly

These are known problems to statisticians → that's why there has been a lot of model diagnostics!
But the view in ML is to throw as many predictors as possible into the mix and automate model building.

Inherently interpretable models

- **Key characteristics**
  - **Parsimony** → easier to interpret
    - ✓**Sparsity** → few active effects or complicated relationships
    - ✓**Low-order interactions** → more than two hard to understand
  - **Analytic expression** → use **regression coefficients** for interpretation

- **Goals and challenges of complex ML models**
  - **Extract as much predictive performance** as possible
  - **No emphasis on interpretation** → lots of variables, complex relationships and interactions
  - No analytic expressions → **rely on low dimensional summaries** → **don't present the full picture**

- **Emerging view:**
  - **Low-order functional** (nonparametric) **models** are **adequate** in most of our applications

    → **tabular data in banking**
  - **Directly interpretable**
  - **Reversing emphasis on complex modeling**

    → **trade-off**: small improvements in **predictive performance vs interpretation**

Examples of “Low Order” Models

- **Functional ANOVA Models:**

$$f(\boldsymbol{x}) = g_0 + \sum_j g_j(x_j) + \sum_{j<k} g_{jk}(x_j, x_k) + \sum_{j<k<l} g_{jkl}(x_j, x_k, x_l) + \cdots$$

  - FANOVA models with low-order interactions are adequate for many of our applications
  - **Focus** on models with **functional main effects and second order interactions**
  - Stone (1994); Wahba and her students (see Gu, 2013)

    → use **splines** to estimate low-order functional effects non-parametrically
  - **Not scalable** to large numbers of observations and predictors
  - Recent approaches

    → use **ML architecture and optimization algorithms** to develop fast algorithms

FANOVA framework

$$f(\boldsymbol{x}) = g_0 + \sum_j g_j(x_j) + \sum_{j<k} g_{jk}(x_j, x_k)$$

- Model made up of mean g_0, **main effects $g_j(x_j)$, two-factor interactions $g_{jk}(x_j, x_k)$**
- **Interpretability**
  - Fitted model is **additive,** effects are enforced to be **orthogonal**
  - Components can be **easily visualized** and **interpreted directly**
  - Regularization or other techniques used to keep model parsimonious

- Two state-of-the-art ML algorithms for fitting these models:
  - **Explainable Boosting Machine** (Nori, et al. 2019) → boosted tress
  - **GAMI Neural Networks** (Yang, Zhang and Sudjianto, 2021) → specialized NNs
  - **GAMI-Tree** (Hu, Chen, and Nair, 2022) → specialized boosted model-based trees

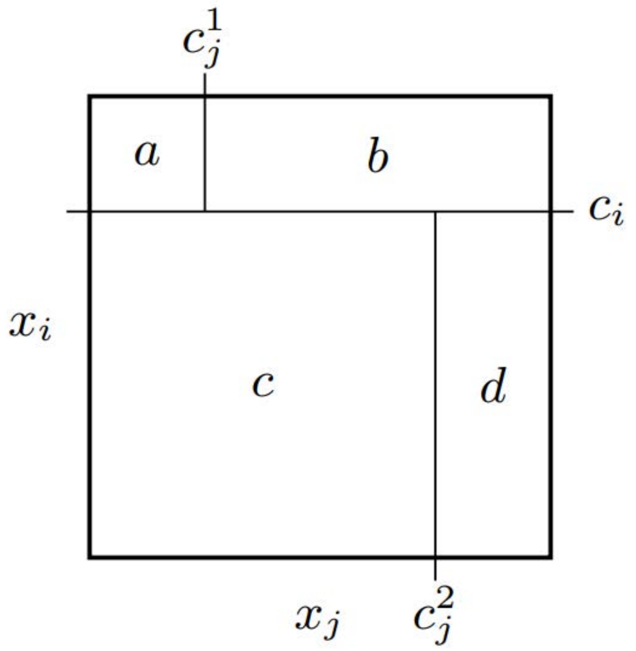
Nori, Jenkins, Koch and Caruana (2019). InterpretML: A Unified Framework for Machine Learning Interpretability. arXiv: 1909.09223
Yang, Zhang and Sudjianto (2021, Pattern Recognition): GAMI-Net. arXiv: 2003.07132

Explainable Boosting Machine

- **EBM** – Boosted-tree algorithm by Microsoft group (Lou, et al. 2013)

$$f(\boldsymbol{x}) = g_0 + \sum g_j(x_j) + \sum g_{jk}(x_j, x_k)$$

  – Microsoft InterpretML (Nori, et al. 2019)
  – fast implementation in C++ and Python



- **Multi-stage model training :**
  – 1: fit functional main effects non-parametrically
    – **Shallow tree boosting** with splits on the same variable for capturing a non-linear main effect
  – 2: fit pairwise interactions on residuals:
    a. Detect interactions using **FAST** algorithm
    b. For each interaction (x_j, x_k), fit function $g_{jk}(x_j, x_k)$ non-parametrically using a tree with depth two: 1 cut in x_j and 2 cuts in x_k, or 2 cuts in x_j and 1 cut in x_k (pick the better one)
    c. Iteratively fit all the detected interactions until convergence

Lou, Caruana, Gehrke and Hooker (2013). Accurate Intelligible Models with Pairwise Interactions. Microsoft Research

Explainable boosting machine: Example

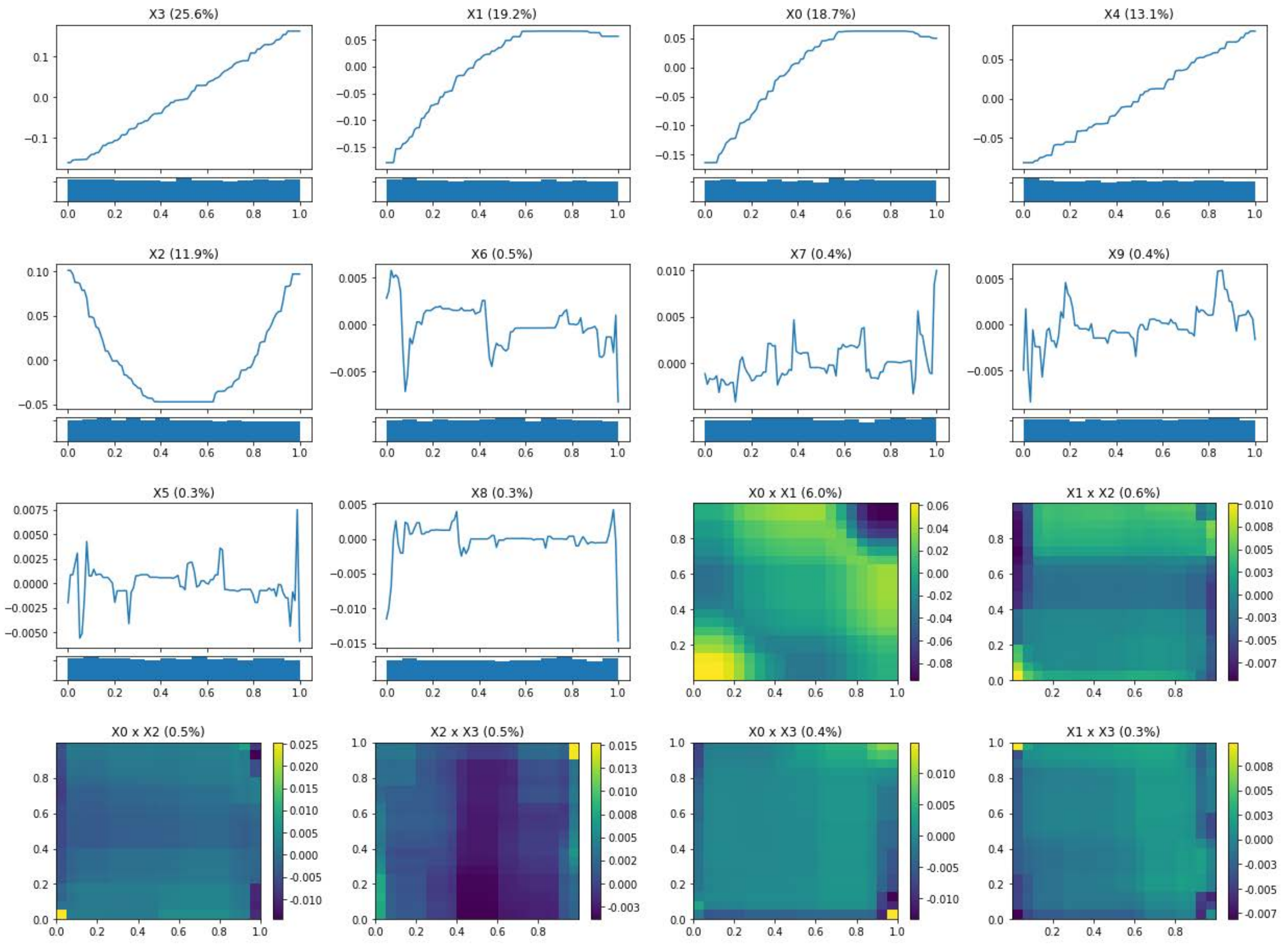
Friedman1 simulated data:

- sklearn.datasets.make_friedman1 n_samples=10000, n_features=10, and noise=0.1.
- Multivariate independent features $\boldsymbol{x}$ uniformly distributed on [0,1]
- Continuous response generated by

$$y(\boldsymbol{x}) = 10\sin(\pi x_0 x_1) + 20(x_2 - 0.5)^2 + 20x_3 + 10x_4 + \epsilon$$

depending only $\boldsymbol{x_0}$~$\boldsymbol{x_4}$

EBM Output with Test RMSE = 0.0284 and R2 = 97.39%



GAMI-Net

- NN-based algorithm for non-parametrically fitting

$$f(\boldsymbol{x}) = g_0 + \sum g_j(x_j) + \sum g_{jk}(x_j, x_k)$$

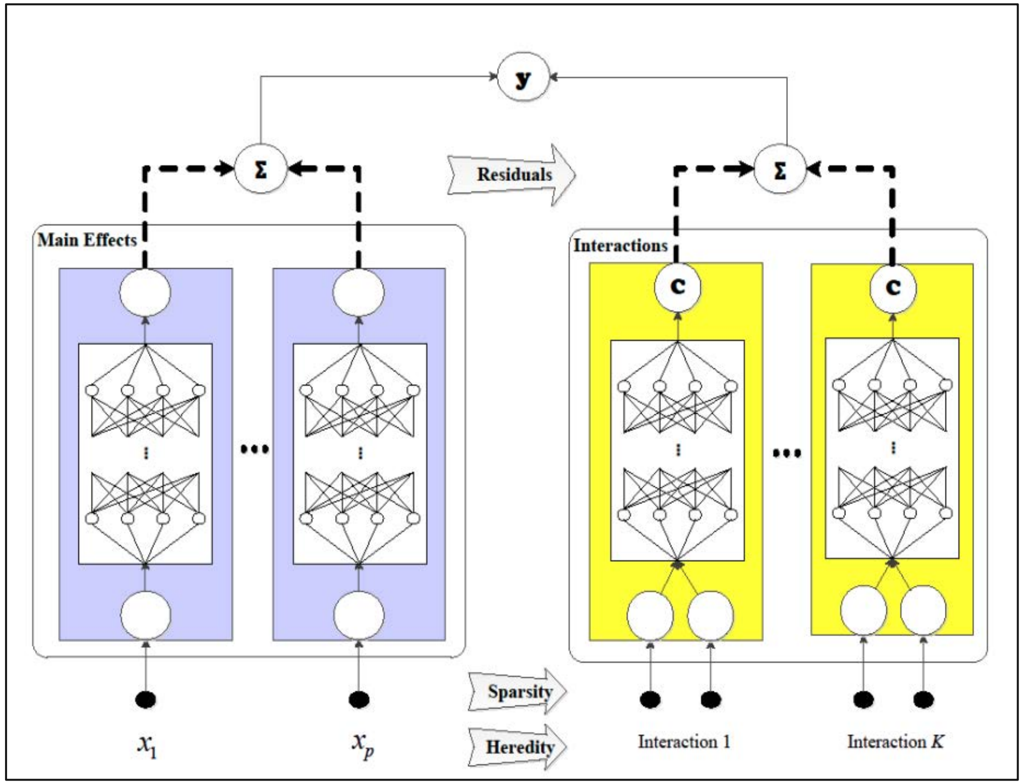
- **Multi-stage training algorithm:**

  1: estimate $\{g_j(x_j)\}$ → train main-effect subnets and **prune** small main effects

  2: estimate $\{g_{jk}(x_j, x_k)\}$ → compute residuals from main effects and train pairwise interaction nets

  - Select candidate interactions using heredity constraint
  - Evaluate their scores (by FAST) and select top-K interactions;
  - Train the selected two-way interaction subnets;
  - Prune small interactions

  3: retrain main effects and interactions simultaneously



Yang, Zhang and Sudjianto (2021, Pattern Recognition): GAMI-Net. arXiv: 2003.07132

Diagnostics: Effect importance and feature importance

- Each **effect importance** (before normalization) is given by

$$D(h_j) = \frac{1}{n-1}\sum_{i=1}^{n} g_j^2(x_{ij}), \qquad D(f_{jk}) = \frac{1}{n-1}\sum_{i=1}^{n} g_{jk}^2(x_{ij}, x_{ik})$$

- For prediction at $\boldsymbol{x}_i$, the **local feature importance** is given by

$$\phi_j(x_{ij}) = g_j(x_{ij}) + \frac{1}{2}\sum_{j \neq k} g_{jk}(x_{ij}, x_{ik})$$

- For GAMI-Net (or EBM), the **global feature importance** is given by

$$\mathrm{FI}(x_j) = \frac{1}{n-1}\sum_{i=1}^{n}\left(\phi_j(x_{ij}) - \overline{\phi_j}\right)^2$$

- The effects can be visualized by a line plot (for main effect) or heatmap (for pairwise interaction).

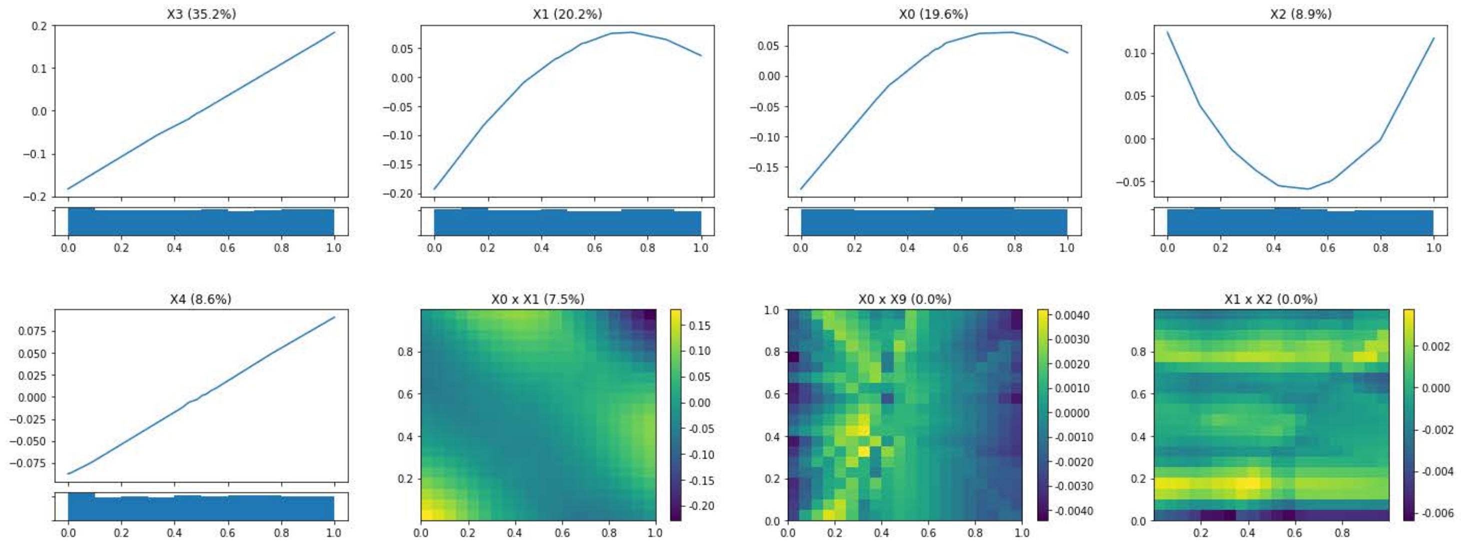
GAMI-Net: Example

Friedman1 data:

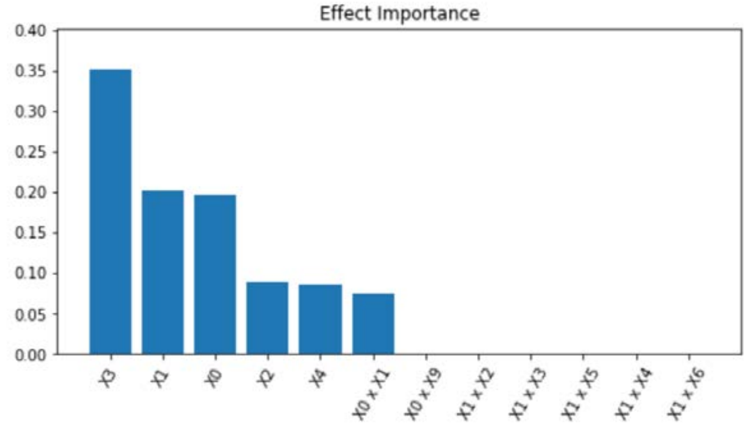
$$y(\boldsymbol{x}) = 10\sin(\pi x_0 x_1) + 20(x_2 - 0.5)^2 + 20x_3 + 10x_4 + \epsilon$$

Same data generated as for EBM example.

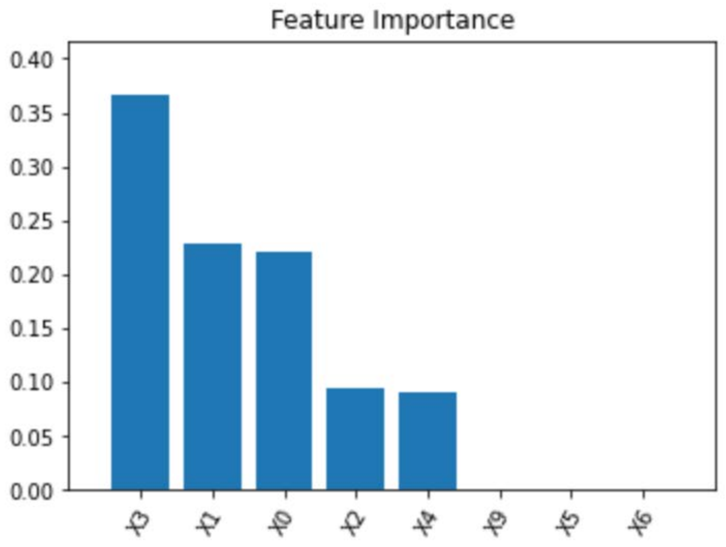
GAMI-Net Output with Test RMSE = 0.0058 and R2 = 99.89%




```
model_gaminet.show_effect_importance()
```




```
model_gaminet.show_feature_importance()
```

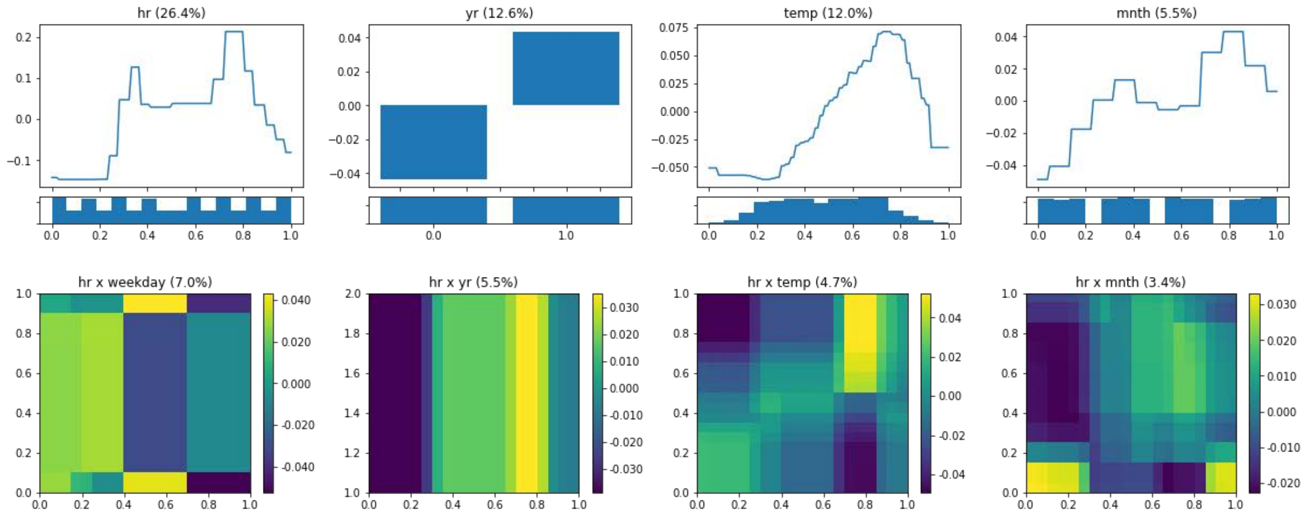


Comparisons: Bike Sharing Data

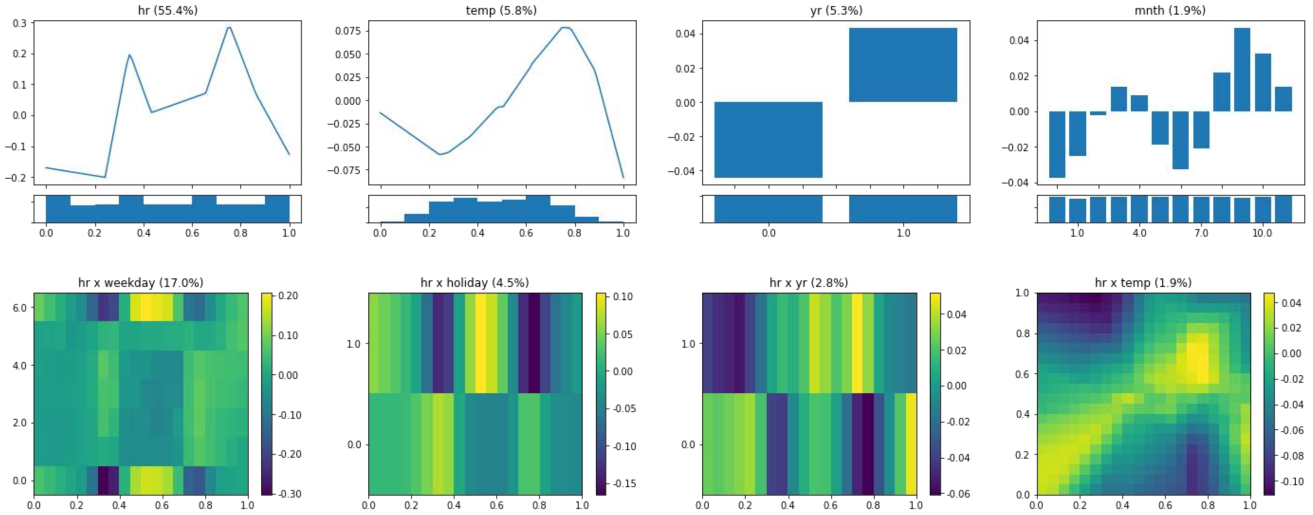
Bike sharing data:

- Another [popular benchmark UCI dataset]() consisting of hourly count of rental bikes between years 2011 and 2012 in Capital bikeshare system.
- Sample size: 17379
- The features include weather conditions, precipitation, day of week, season, hour of the day, etc.
- The response is count of total rental bikes.

EBM Output with test RMSE = 0.0825 and R2 = 80.58%



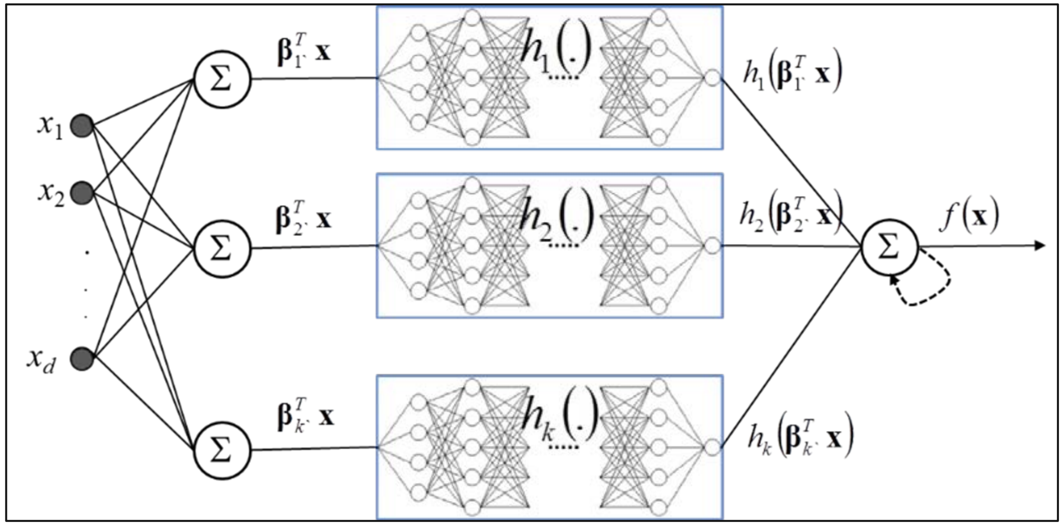
GAMI-Net Output with test RMSE = 0.0595 and R2 = 89.89%



Another example of “Low Order” Models:

- **Additive Index Models:**

$$f(\boldsymbol{x}) = g_1(\boldsymbol{\beta}_1^T \boldsymbol{x}) + g_2(\boldsymbol{\beta}_2^T \boldsymbol{x}) + \ldots + g_K(\boldsymbol{\beta}_K^T \boldsymbol{x})$$



  - Generalization of GAMs:

$$f(\boldsymbol{x}) = g_1(x_1) + g_2(x_2) + \ldots + g_P(x_P)$$

  - Incorporates certain types of interactions
  - **Projection pursuit regression** (Friedman and Stuetzle, 1981)
  - **Need for scalable algorithms** with large datasets and many predictors

  - Use specialized **neural network architecture and associated fast algorithms**
    - e**X**plainable **N**eural **N**etworks (xNNs) → Vaughan, Sudjianto, … Nair (2020)

Summary

- Advent of “Big Data” and advances in computing → many opportunities
  - Large datasets → flexible models → better performance
  - Automated feature engineering and selection
  - Exploit information in new sources of data (text)
- Challenges
  - Computational
  - Overfitting, model robustness, generalizability, …
  - Incorporating shape constraints and subject matter knowledge
  - Interpretability
  - Fairness and Bias