WELLS FARGO

Workshop on Machine Learning and Application

Module 2: Modeling tabular data, explanation and diagnostics

March 24, 2023

Vijay Nair
Joint with Anwesha Bhattacharyya

Advanced Technologies for Modeling Group
Corporate Model Risk

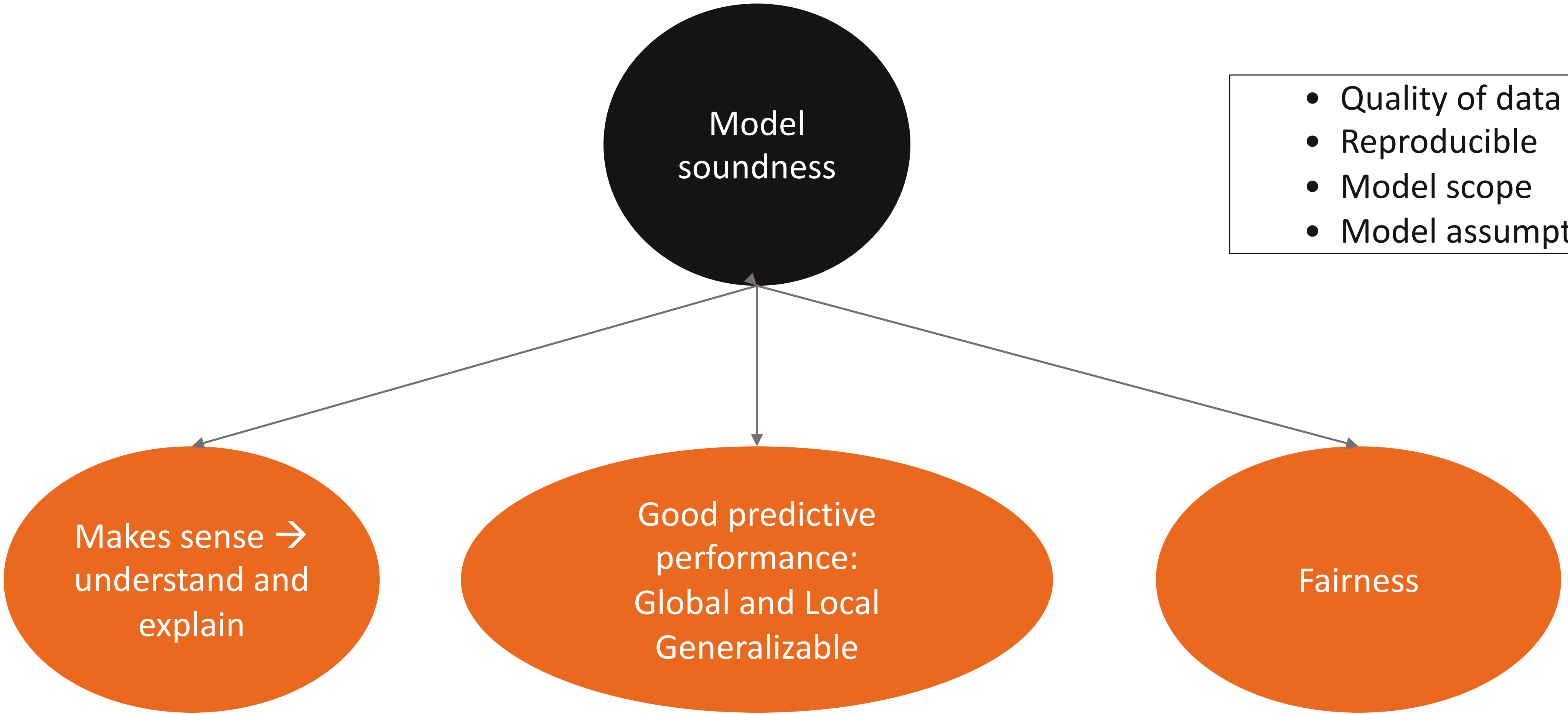
Outline

- Introduction: Challenges and concepts
- Model interpretability
  - Post hoc techniques for model explanation
  - Inherently-interpretable ML algorithms
- Model diagnostics
- Discussion

Opportunities and Challenges with ML

- Flexible modeling … works well with large datasets
  - ➢ Better predictive performance
  - ➢ Automated approach to feature engineering
    - Saves time
    - Useful in new applications with insufficient prior knowledge on feature engineering
- BUT … Predictor $\hat{f}(x)$ is implicitly defined, high-dimensional, and complex
  - Hard to interpret results
  - Not an issue if goal is only prediction: recommender systems, fraud detection, …
  - Big issue for regulated industries and safety-critical applications
  - Banks have dual goals: good predictive performance and ensure results make sense
- Must understand model, results, and develop insights
  - Why? Provide explanations to multiple stakeholders
    - ✓ Model must make sense → consistent with subject-matter knowledge
    - ✓ For certain applications, model must be "fair"
    - ✓ Model must be generalizable
      - ▪ Identify areas of poor model fit, fix problem, or develop mitigation strategies
    - ✓ Model must be robust: not overfit or pick up artifacts in the data

Selected model soundness concepts



- Quality of data
- Reproducible
- Model scope
- Model assumptions

Model soundness

- Makes sense
  - Model and results are interpretable and can be explained to stakeholders
  - Results are consistent with subject-matter expertise
- Good predictive performance
  - In comparison to other algorithms (benchmark models for high-risk rank)
  - Global as well as local
  - Generalizable to potential new environments (when it should)
    - Stable to certain changes when it should be
    - Good sensitivity to certain changes in key predictors
- Robust
  - Not overly flexible and unstable
  - Does not overfit training data (globally or locally)
- Fair
  - Does not discriminate based on protected attributes
  - Results are fair to customers and other stakeholders
- Above points are not mutually exclusive

Outline

- Introduction: Challenges and concepts
- Model interpretability
  - Post hoc techniques for model explanation
  - Inherently-interpretable ML algorithms
- Diagnostics for model weakness
  - Predictive performance
    - Global and local
    - Generalizability
  - Robustness
- Bias and fairness
- Discussion

Making sense: Understanding model results

Main approaches:

I. Post hoc: Techniques for interpreting results after fitting the ML algorithm
   a) Global – Important predictors; input-output relationships
   b) Local – how does model behave locally; contribution of predictors to a particular prediction

II. Inherently interpretable algorithms
   a) Low-order functional ANOVA models → primary focus
   b) Additive index models

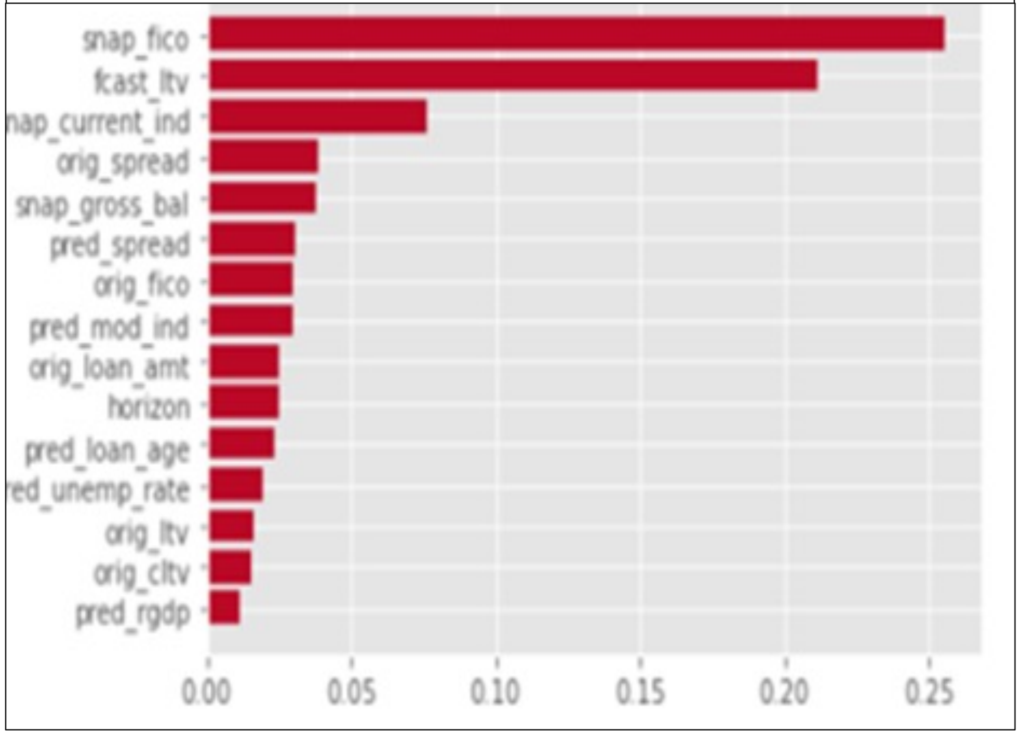
III. Fitting and using surrogate models to explain complex results (skip)
   a) Born-again trees (piecewise constant) → Breiman
   b) Locally additive tress → Hu, Chen, Nair (2022)

Post hoc global: Identifying important predictors/features

- **Permutation based: Model agnostic**
  - L. Breiman, "Random Forests", Machine Learning, pp 5-32, 2001.
  - 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.
  - Issues:
    - Double counting of interactions
    - Correlated predictors → Option: joint 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

Home Mortgage-XGB



- **Selected Others** (Many in literature)
  - **Tree-based** importance metrics
    - Importance of a variable x_j based on impurity
      - Total reduction of impurity at all nodes where x_j used for splitting
    - For ensemble algorithms, average over all trees
  - **Global Shapley**
    - Based on Shapley decomposition (1953);
    - Owen (2014) and others applied it to ML feature importance
    - Model agnostic but **computationally intractable**

Understanding input-output relationships: 1-dimensional partial dependence plots

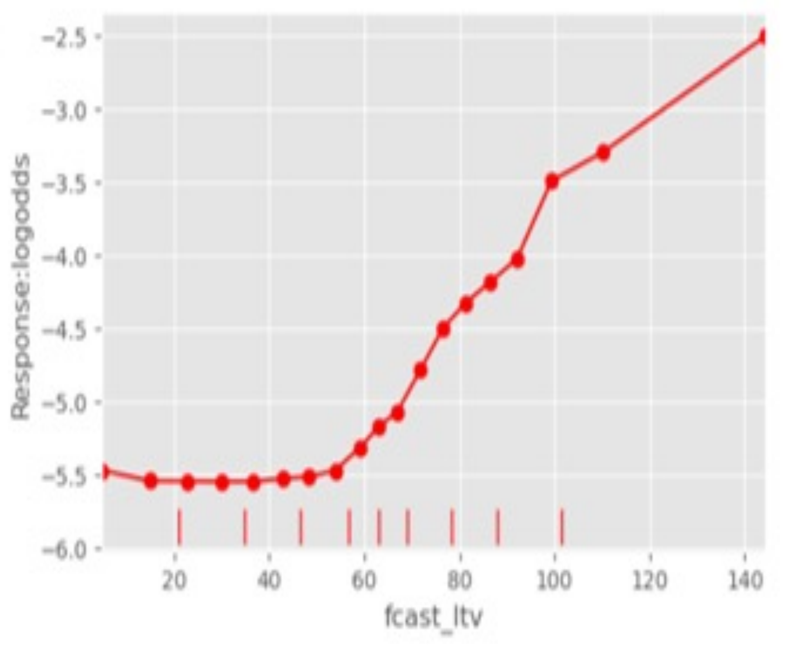
- Understand how fitted response varies as a function of the variable of interest
- **One-dimensional Partial Dependence Plot (PDP)**
  - Friedman, J. H (2001). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 29 (5): 1189-1232
  - Variable of interest: x_j
  - Write 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

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

  - Plot $f_{pdp,j}(x_j)$ against x_j over a grid of values $c_1, \ldots\ c_m$
  - One-dimensional summary
  - Interpretation: Effect of x_j averaged over other variables

- Accumulated local effects plots (in highly correlated cases)
  - Reference

Home Mortgage
1-D PDP for Forecasted LTV



Assessing interactions

I. ICE (individual conditional expectation) plots

Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2013). Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. *eprint arXiv:1309.6392*

II. Two-dimensional partial dependence plots

III. H-statistics for quantifying two-dimensional interactions

Friedman, J. H (2001). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 29 (5): 1189-1232.

- One can use ICE plots to detect the presence of interactions, but they do not give further insights
- Items II and II examine interactions with specific pairs of variables
- They can be extended to higher-dimensional interactions

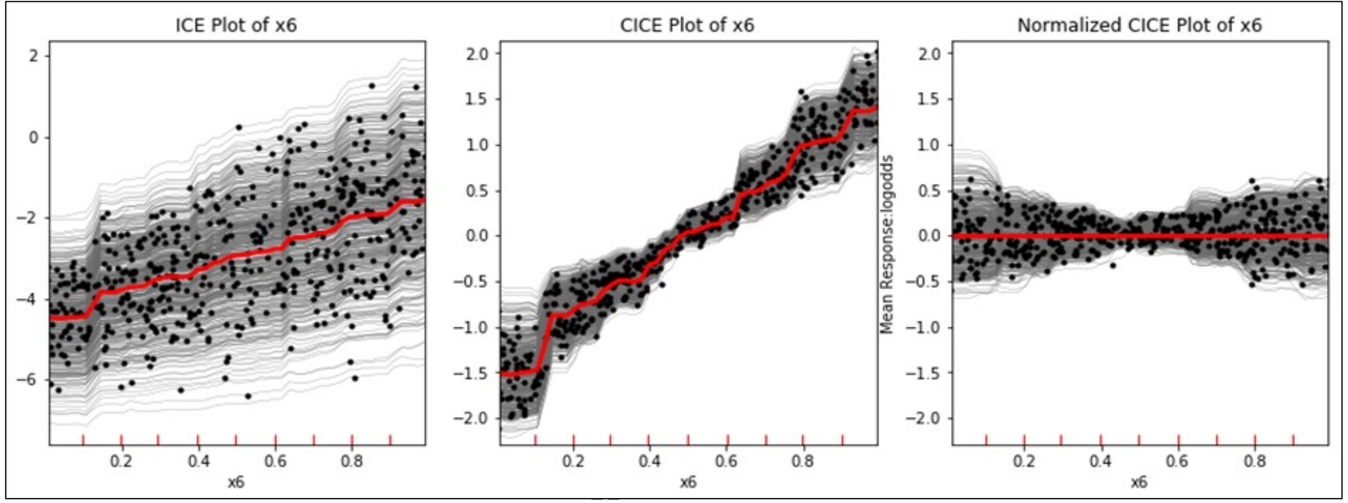
Individual Conditional Expectation Plots

- 1-d partial dependence plot $f_{pdp,j}(x_j)$ shows the average over the entire data

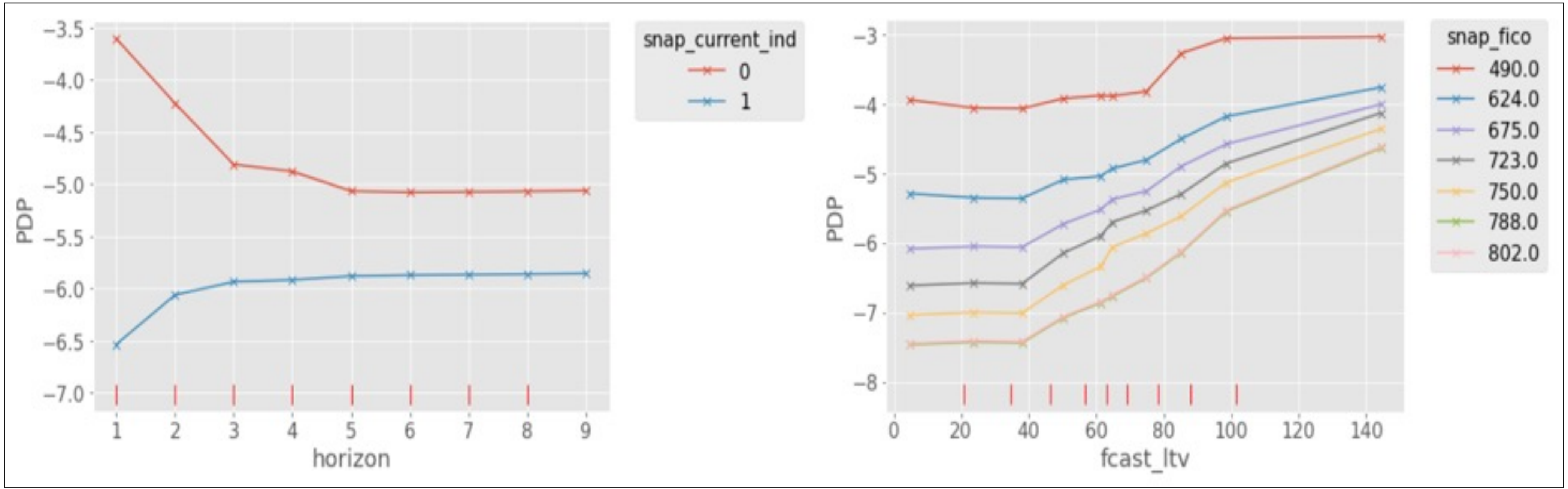
$$f_{pdp,j}(x_j) = \frac{1}{N}\sum_{i=1}^{N} \hat{f}(x_j = c, x_{-j,i})$$

- When there are interaction effects, $\hat{f}(x_j, x_{-j,i})$will have different patterns for different x_{ji}.
- So averaging will lose the interaction information.
- The ICE plot is a plot of all the N curves $\hat{f}(x_j, x_{-j,i}), i = 1, 2, \dots, N$.
- Each curve is localized for a single ith observation.
- It allows us to see if there is any change of the input-output relationships for x_j, thus to see any interaction effect.

- ICE plot for a simulation example.
  - True model: $log\left(\frac{p_i}{1-p_i}\right) = -8 + 1.6\,x_1 + 4\sqrt{x_2} - x_3^2 + x_5^2 + 2.4\,x_6 + 2x_1x_6$
  - The dots show the plot of $\hat{f}(\boldsymbol{x}_i)$ against the observed data x_{-6i}.
  - The black curves are the ICE curves over a grid of x_6.
  - The red curve is the PDP, which is the average of all ICE curves.
  - CICE plots are centered versions
  - If we further subtract the PDP, we get the normalized CICE.
  - It shows the remainder interaction effects after subtracting the main effects of x_6 and x_C



Input-output relationships:
Two-dimensional partial dependence plots



- One way to display 2D-PDPs
- Multiple 1D-PDPs for the variable on the x-axis
- Multiple curves: Each represent fixed values of the second variable
- Non-parallel curves show interactions

- Other ways to display information include contour plots and heat maps

H-statistics to measure two-dimensional interactions

- H_{jk} to measure the interaction between x_j and x_k

$$H_{jk}^2 = \frac{\sum_{i=1}^{N}\left[f_{pdp}(x_{ij},x_{ik}) - f_{pdp}(x_{ij}) - f_{pdp}(x_{ik})\right]^2}{\sum_{i=1}^{N} f_{par}^2(x_{ij},x_{ik})}, \quad H_{jk} = \sqrt{H_{jk}^2}$$

  - $f_{pdp}(x_{ij}, x_{ik})$, $f_{pdp}(x_{ij})$, $f_{pdp}(x_{ik})$ are the centered two and one-dimensional partial dependence functions
  - H_{jk}^2 is the proportion of variation in $f_{par}(x_{ij}, x_{ik})$ unexplained by an additive model – relative measure
  - There is no easy way assess if it is large or small

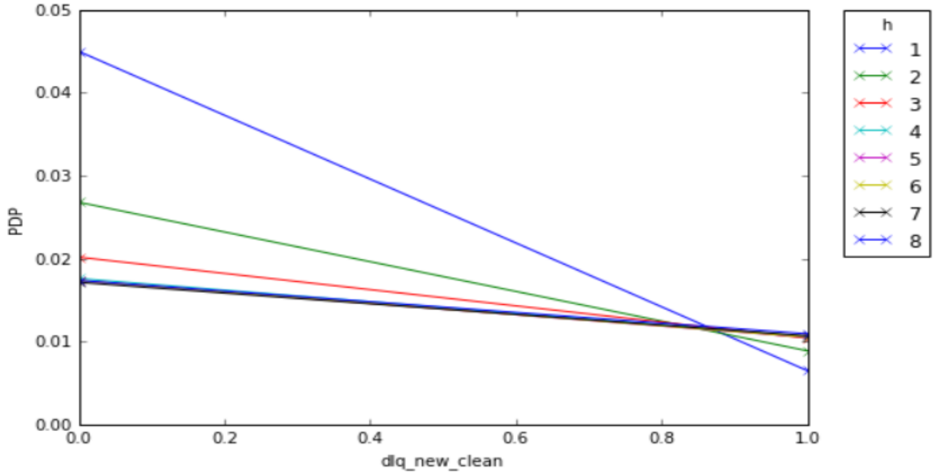
- One can also use an absolute version of H-statistic (without the denominator)

$$\widetilde{H}_{jk}^2 = \frac{1}{N}\sum_{i=1}^{N}\left[f_{pdp}(x_{ij}, x_{ik}) - f_{pdp}(x_{ij}) - f_{pdp}(x_{ik})\right]^2, \; \widetilde{H}_{jk} = \sqrt{\widetilde{H}_{jk}^2}$$

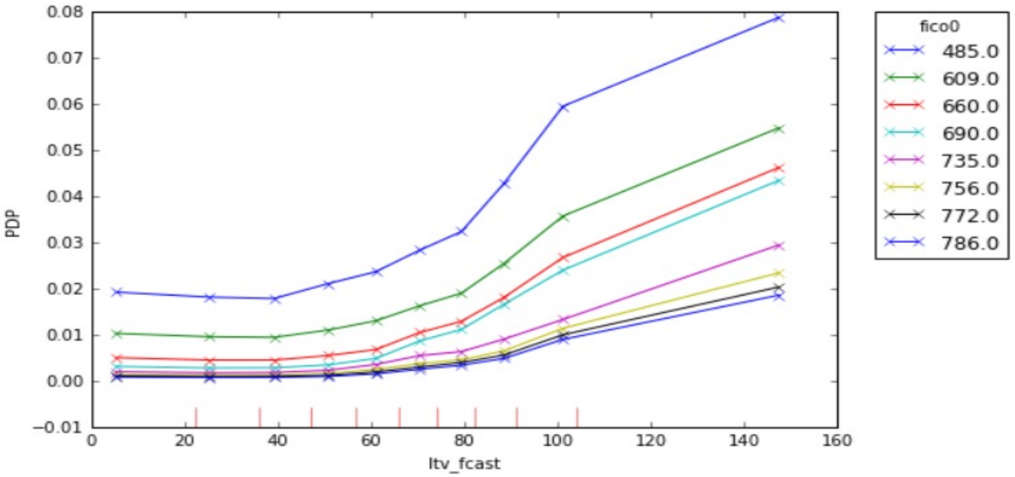
Illustration: 2-D PDPs and H-statistics: Home Lending Example

- Interactions between FICO and LTV_forcast (left), h and dlq_new_clean(right).

	fico0	ltv_fcast	dlq_new_clean	unemprt	totpersincyy	h	premod_ind
fico0	NaN	0.1630	0.1224	0.0820	0.0360	0.0339	0.1107
ltv_fcast	0.1630	NaN	0.0518	0.0291	0.0286	0.0186	0.0843
dlq_new_clean	0.1224	0.0518	NaN	0.0101	0.0071	0.2296	0.0003
unemprt	0.0820	0.0291	0.0101	NaN	0.0232	0.0122	0.0094
totpersincyy	0.0360	0.0286	0.0071	0.0232	NaN	0.0068	0.0661
h	0.0339	0.0186	0.2296	0.0122	0.0068	NaN	0.0192
premod_ind	0.1107	0.0843	0.0003	0.0094	0.0661	0.0192	NaN



Delinquency vs horizon



LTV_fcast vs fico

Techniques for local explanation

- **Two questions of Interest:**

  1. How does the model behave locally at a point of interest?
  2. Consider the predicted value at a point of interest $\hat{f}(\boldsymbol{x}^*) = \hat{f}(x_1^*, \ldots, x_K^*)$:
     What are the contributions of the different variables/features $\{x_1, \ldots x_K\}$ to this prediction?

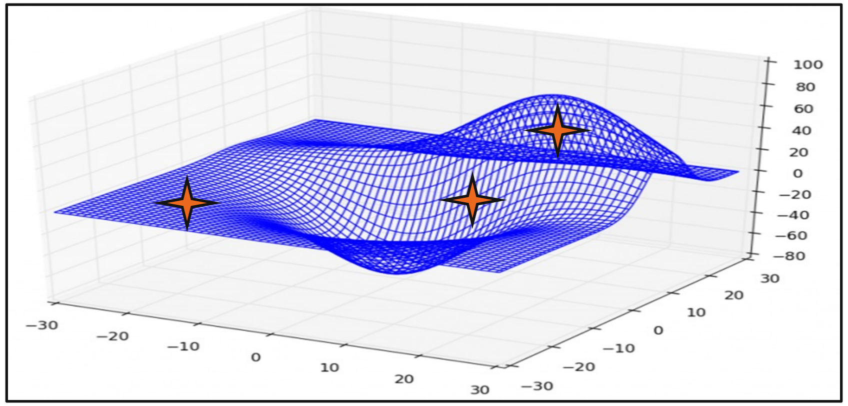
  We will see an example of this in credit applications.

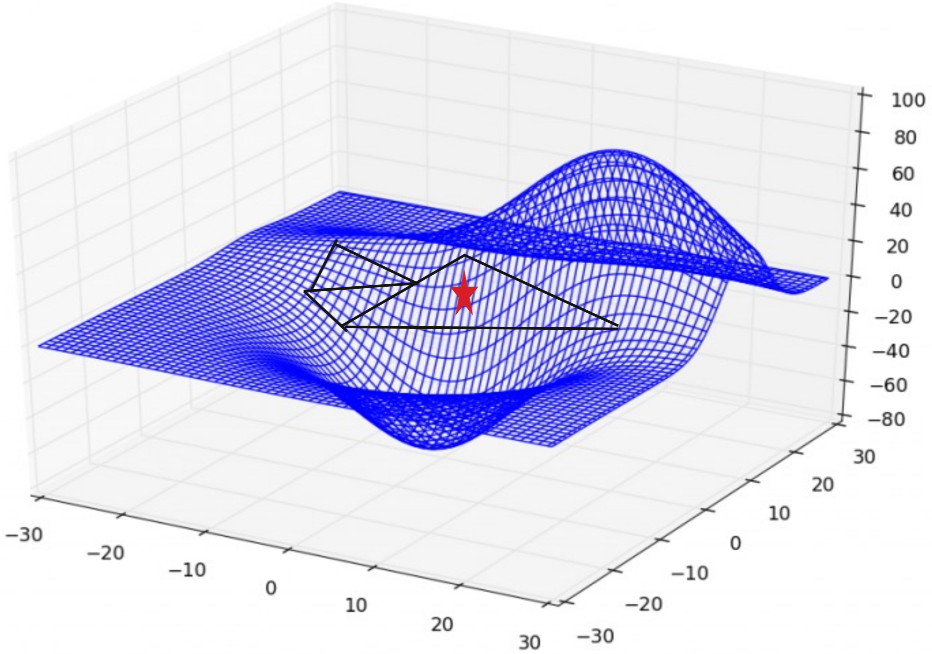
- If fitted model is linear: $\hat{f}(\boldsymbol{x}) = b_0 + b_1 x_1 + \ \ldots b_K x_k$,
  we can answer both questions using the regressions coefficients.
  - Answer to 1: Model is linear → magnitudes and signs of regression coefficients provide explanation
  - Answer to 2: Contribution of x_j^* is $b_j x_j^*$

- BUT … how to extend these interpretations to ML algorithms?

Techniques for local explanation: Question 1

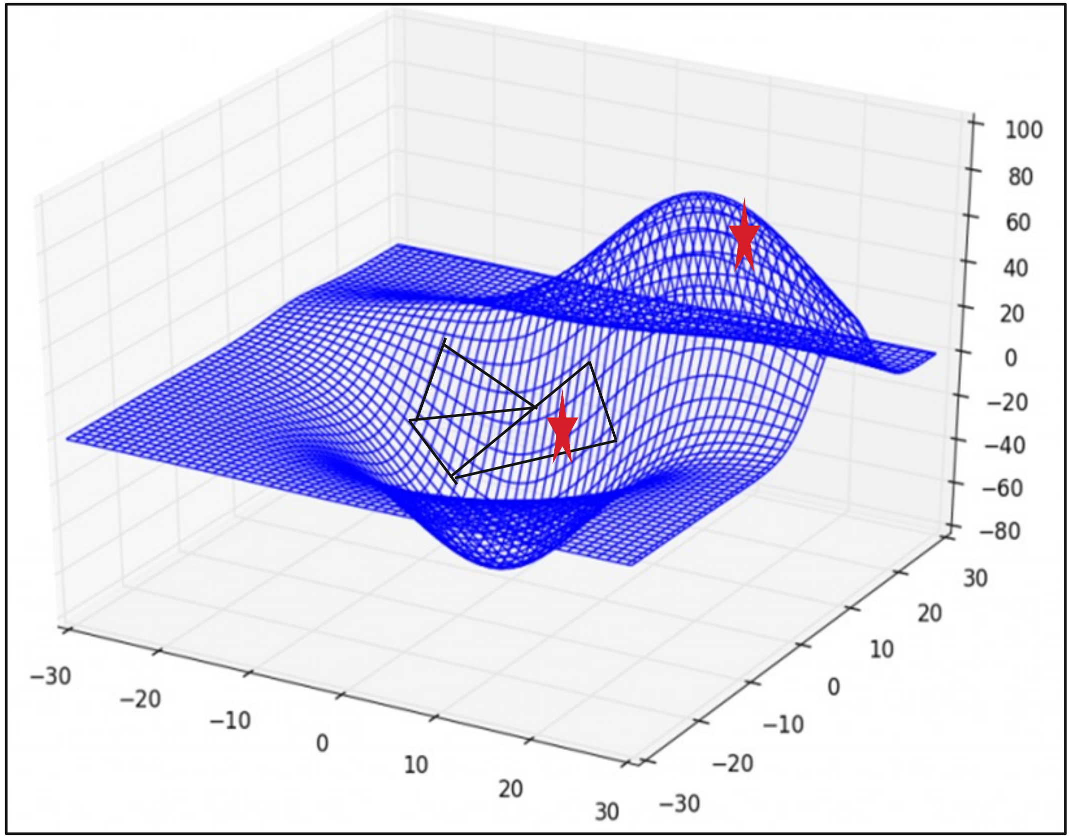
- How can we interpret the response surface locally at selected points of interest?
- LIME
  - Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you?: Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 1135-1144)
  - Fit a linear model locally around the point and use for interpretation
  - Above reference suggests a particular way to fit a local model – but there are simple ways to do it.
- FFNNs based on RELU activation function
  - Essentially partitions predictor space into regions and fits a linear regression model within
  - Difference with LIME: local linear model developed for each point
  - FFNN-RELU yields same linear model for all points in region
- Piecewise constant tree
  - Single regression tree or RF or XGB)
  - Same as FFNN but splits into rectangular regions





Techniques for local explanation: Question 2

- Consider the predicted value at a point of interest $\hat{f}(\boldsymbol{x}^*) = \hat{f}(x_1^*, \dots, x_K^*)$
- What are the individual contributions of $\{x_1, \dots x_K\}$ to this prediction?
- Involves comparison of prediction to a reference point (“average”)
- Approaches in previous slide **cannot** be used
- Many techniques in the literature
- Common approaches based on local Shapley decomposition (called SHAP)
  - Lundberg and Lee (2017)
  - Many variations: Kernel SHAP, Tree SHAP
  - Recommend Baseline SHAP
  - Sundarajan and Najmi (2019)



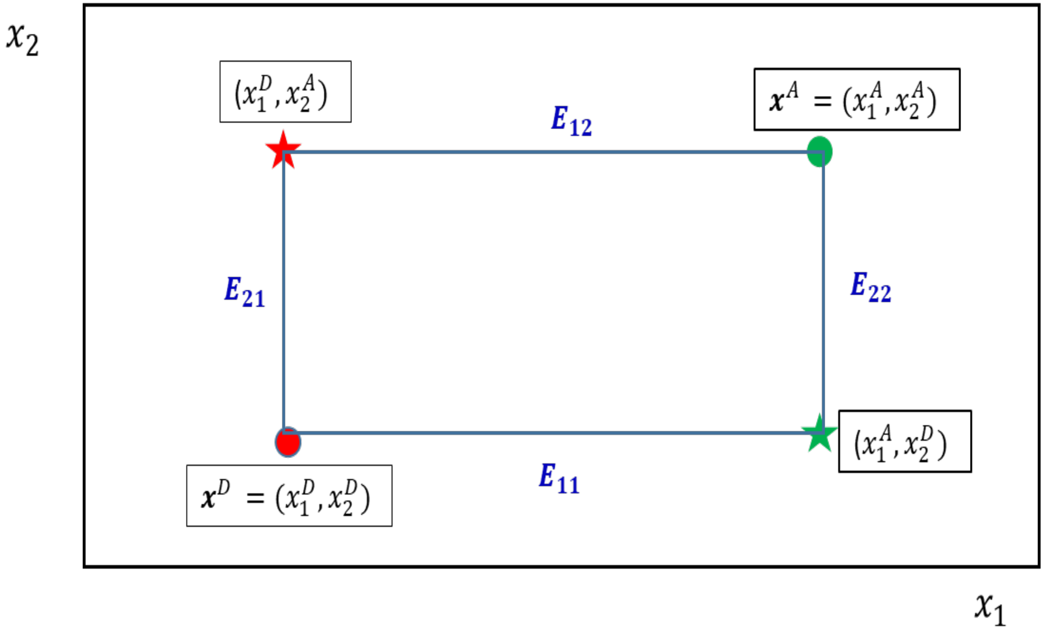
General expression for B-SHAP

- Let $f(\boldsymbol{x})$ be the fitted model with K variables
- Consider two points of interest in the predictor space: point of interest $\boldsymbol{x^D}$ and a reference point $\boldsymbol{x^A}$
- The goal is to decompose the difference $[f(\boldsymbol{x^D}) - f(\boldsymbol{x^A})]$ and attribute it to the difference variables $(x_1, \dots x_K\}$
- Baseline-SHAP decomposition

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

- Looks formidable

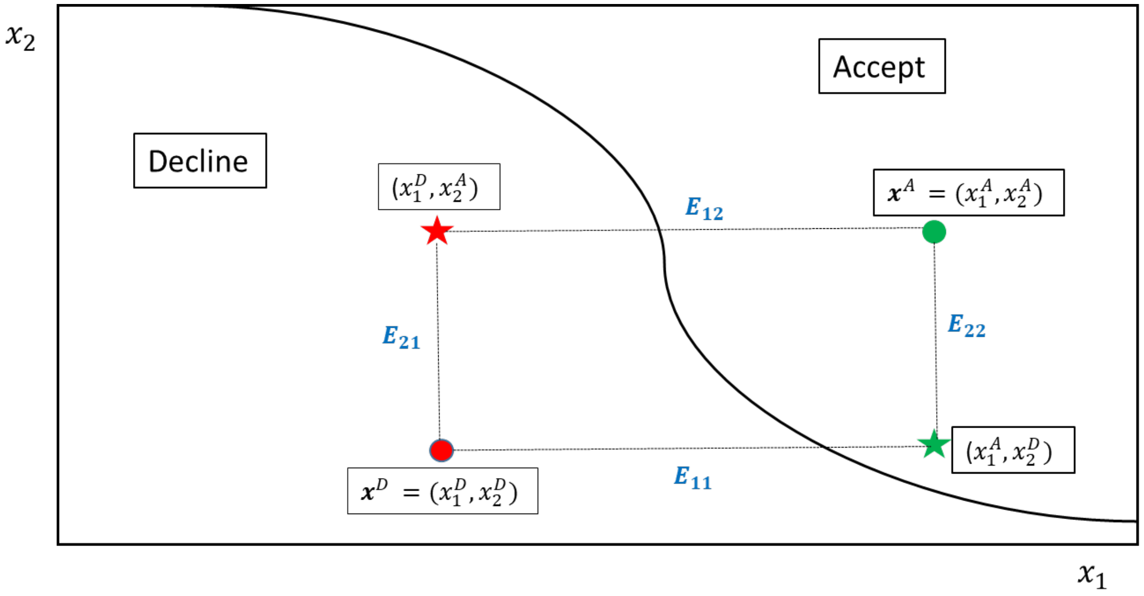
- Consider simple case with two variables: $K = 2$
- Decomposing $\left[f(x_1^D, x_2^D) - f\left(x_1^A, x_2^A\right)\right]$ into contributions by x_1 and x_2
- Consider contribution by x_1
- All possible subset: $\{\phi, 1, 2, \{1,2\}\}$; $\boldsymbol{S}_1 = \{\phi, 2\}$
- $E_1 = \frac{1}{2}\{[f\left(x_1^D, x_2^A\right) - f\left(x_1^A, x_2^A\right)] + [f(x_1^D, x_2^D) - f\left(x_1^A, x_2^D\right)]\}$



Case with two predictors: Motivation from first principles

$$\left[f(x_1^D, x_2^D) - f\left(x_1^A, x_2^A\right)\right] = E_{11} + E_{22}$$

$$\left[f(x_1^D, x_2^D) - f\left(x_1^A, x_2^A\right)\right] = E_{21} + E_{12}$$



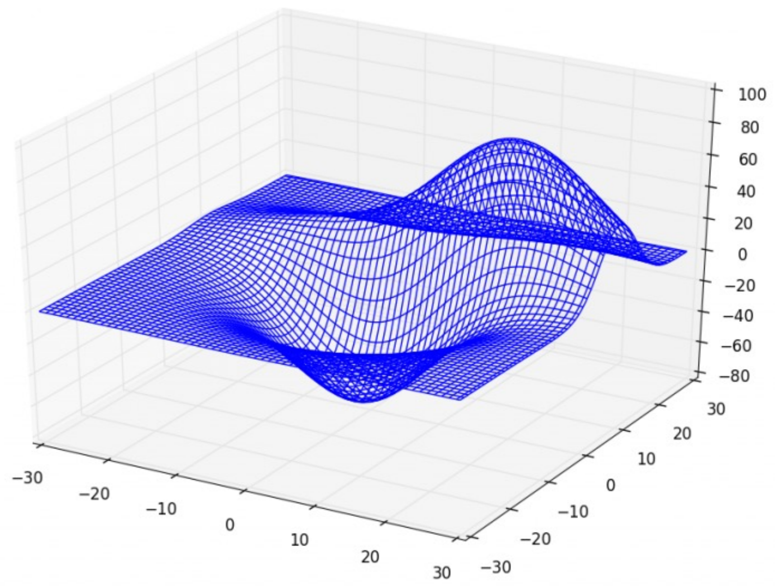
- $E_{11} = \left[f(x_1^D, x_2^D) - f\left(x_1^A, x_2^D\right)\right]$
- $E_{12} = \left[f\left(x_1^D, x_2^A\right) - f\left(x_1^A, x_2^A\right)\right]$
- $E_1 = \frac{1}{2}(E_{11} + E_{12})$

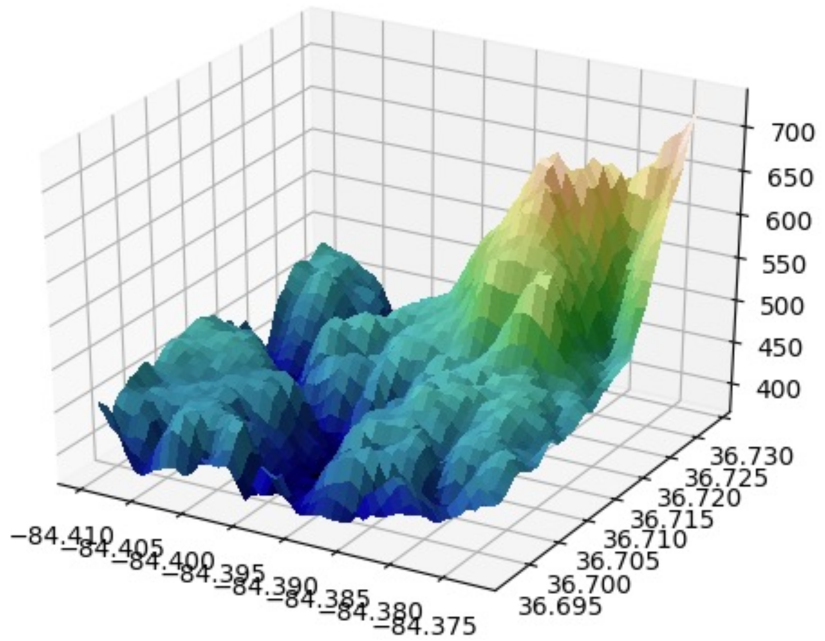
- $E_1 = \frac{1}{2}(E_{11} + E_{12}) \rightarrow \frac{1}{2}\left\{\left[f(x_1^D, x_2^D) - f\left(x_1^A, x_2^D\right)\right] + \left[f\left(x_1^D, x_2^A\right) - f\left(x_1^A, x_2^A\right)\right]\right\}$

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

- What happens in linear model with no interactions?
  - $f(\boldsymbol{x}) = b_0 + b_1 x_1 + b_2 x_2$
  - $E_1 = b_1\left(x_1^D - x_1^A\right)$

Issues with ML algorithms and post hoc explanantions





- **Most post-hoc tools** for studying input-output relationships are **lower-dimensional summaries**
  - **Limited** in **ability** to **characterize complex models** that may have different local behaviors
  - **Need better visualization** tools in high-dimensions

- 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!

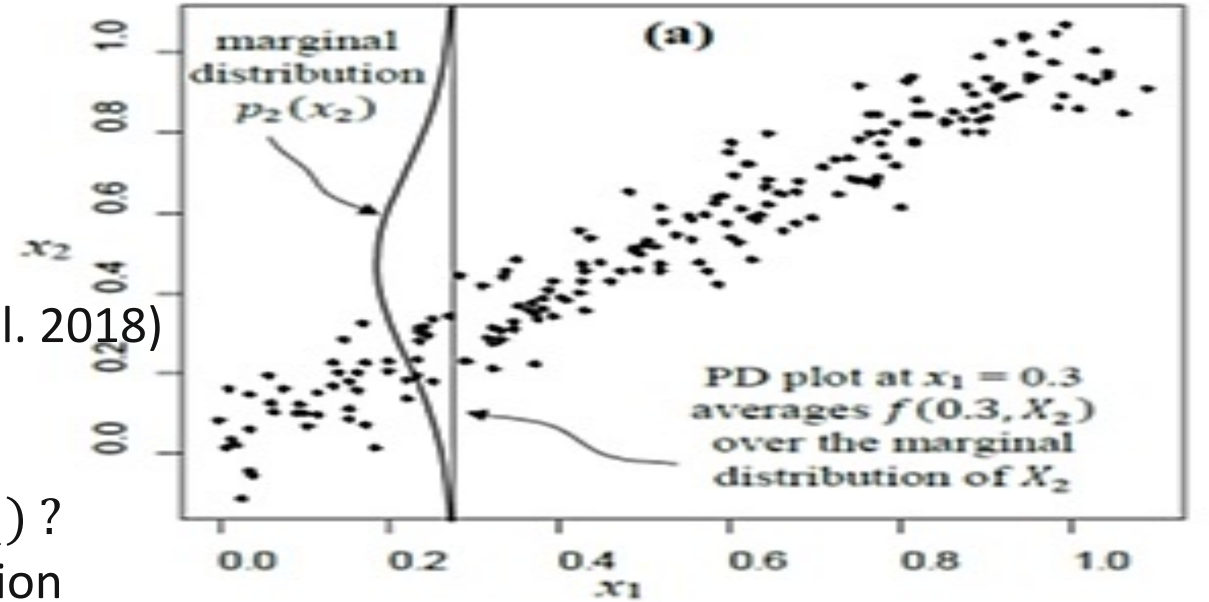
$\hat{f}(\boldsymbol{x}) = \hat{f}(\boldsymbol{x_j}, \boldsymbol{x_{-j}})$ is the fitted model

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

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

No easy answers!

Adverse action in credit applications: An illustration of local explanation

Adverse action (AA) in credit applications

- AA occurs in different contexts: lending, insurance, job application, etc.
- Credit Lending:
  - **Regulation B of Equal Credit Opportunity Act** – Both consumers and businesses
    - **A refusal to grant credit in substantially the amount or terms requested in an application** … ;
    - **A termination of an account** or an **unfavorable change in the terms of an account** … ;
    - A **refusal to increase the amount of credit** available to an applicant ...
- **US Fair Credit Reporting Act**
  - Covers only consumers
  - Broader scope: credit, insurance, employment, government license or benefit, …
- Applicants are **legally entitled to get an explanation** for a negative decision
  - i) specific **principal reason(s)** for action; or
  - ii) disclose to the applicant they have the right to request reason(s) for denial

Adverse action (AA) explanation and Reason Codes

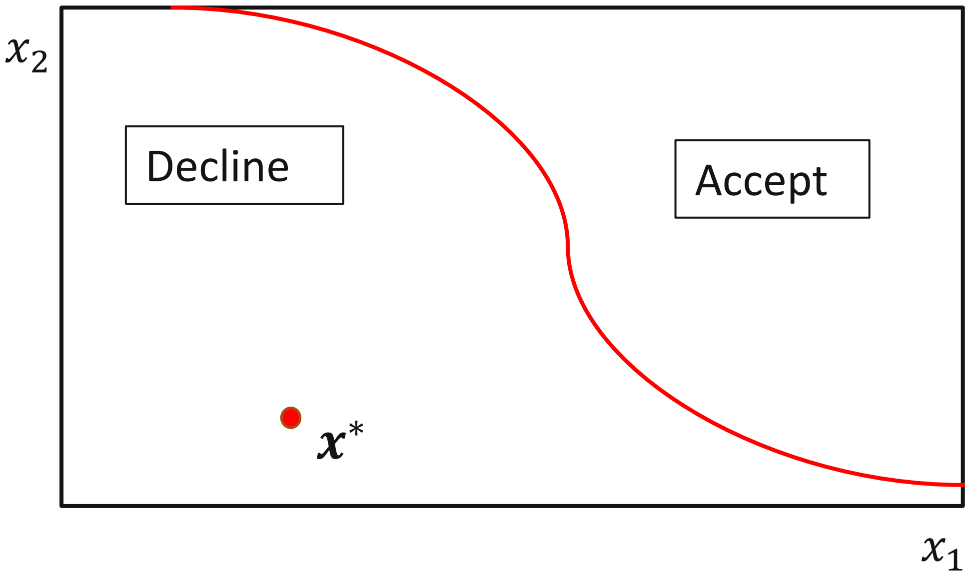
- Nature of explanation depends on stage of review and decision processes
  - – Incomplete or unverifiable information
  - – Decision based on judgement, credit system, or combination

- **Examples of Reason Codes**

Issues with application	After assessment
• Credit application incomplete	• Insufficient income
• Unable to verify credit references	• Limited credit experience
• Length of employment	• Number of recent inquiries on credit bureau report
	• Delinquent credit obligations
	• Value of collateral not sufficient

Decision based on predictive models: Problem formulation

- $\boldsymbol{x} = (x_1, \ldots, x_K)$
- K −dimensional attribute used for credit decision
- Use historical data $\{y_i, \boldsymbol{x}_i\}, i = 1, \ldots n$ to develop model for probability of default (PoD)
- Fitted model for PoD: $p(\boldsymbol{x})$
- Decision:
  - Accept application with attributes $\boldsymbol{x}^*$ if $p(\boldsymbol{x}^*) \leq \boldsymbol{\tau}$;
  - Decline otherwise



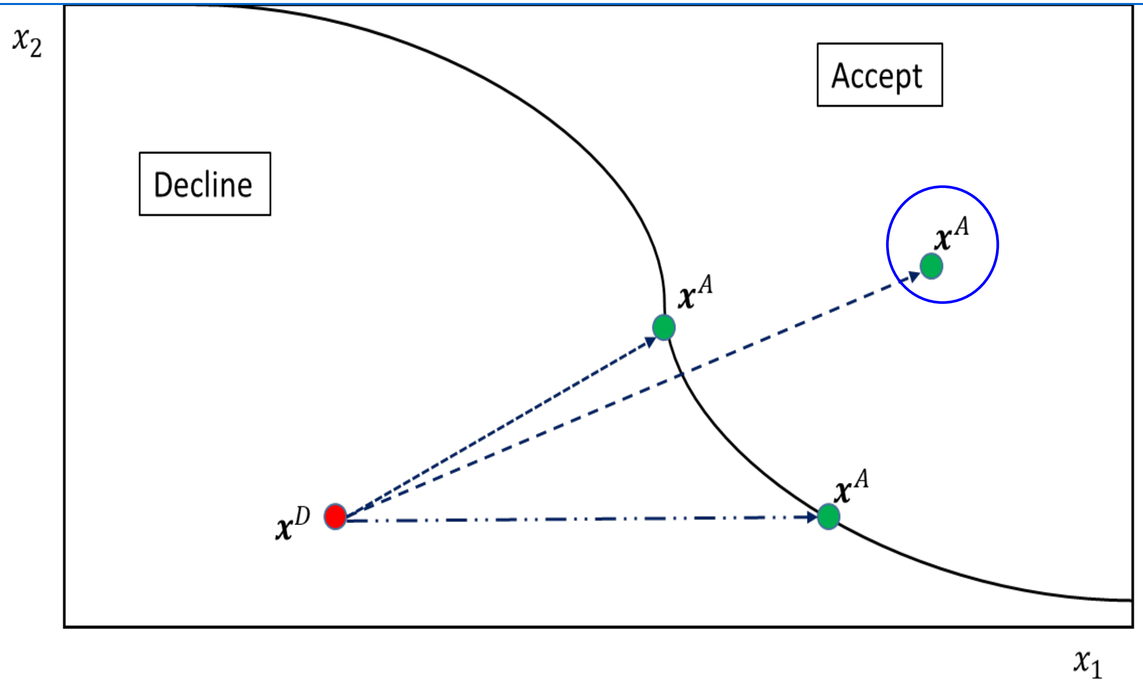
Adverse action (AA) explanation: Reference point

- $\boldsymbol{x^D}$ attribute of declined application

- Adverse action explanation:
    - Choose a reference point $\boldsymbol{x^A}$
    - Difference: $[\boldsymbol{p}(\boldsymbol{x^D}) - \boldsymbol{p}(\boldsymbol{x^A})]$
    - Attribute difference to predictors

- Choice of reference points in "accept" space
    - Internal point
    - On the boundary

- Choices on the boundary
    - Varies with declined application
    - Uncertainty of decision boundary



Selecting of a reference point for comparison

Adverse action explanation

- $\boldsymbol{x^D}$ attribute of declined application

- Pick a reference point $\boldsymbol{x^A}$
- Decompose difference: $[\boldsymbol{p}(\boldsymbol{x^D}) - \boldsymbol{p}(\boldsymbol{x^A})]$
  and allocate to each of the K predictors

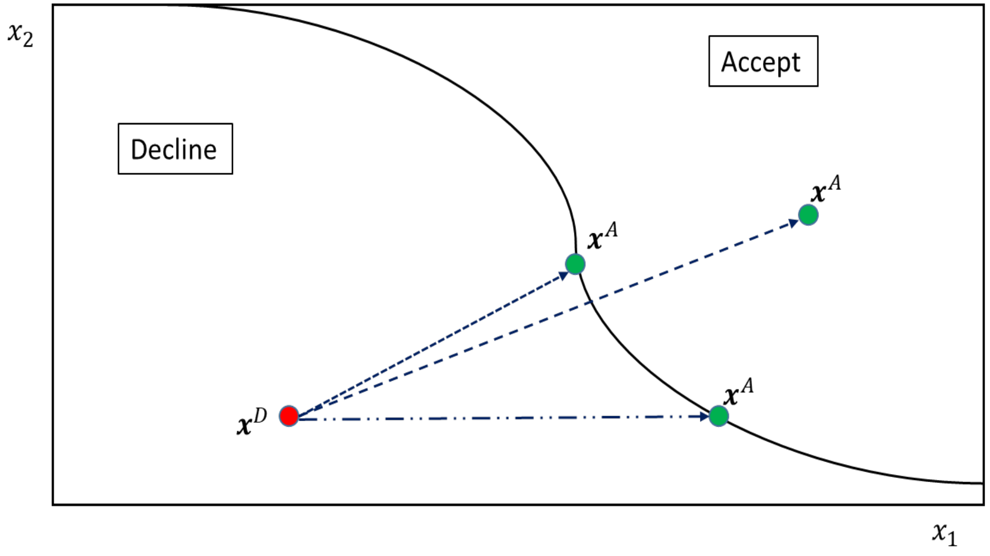


Figure: Selection of a reference point for comparison

- Better to do it in terms of $f(\boldsymbol{x}) = logit\ p(\boldsymbol{x})$
  - $[f(\boldsymbol{x^D}) - f(\boldsymbol{x^A})] = E_1(\boldsymbol{x^D}, \boldsymbol{x^A}) + E_2(\boldsymbol{x^D}, \boldsymbol{x^A}) + \ldots + E_K(\boldsymbol{x^D}, \boldsymbol{x^A})$
    where $E_k(\boldsymbol{x^D}, \boldsymbol{x^A})$ is the allocation to (contribution by) k −th predictor

    From here on, denote $E_k(\boldsymbol{x^D}, \boldsymbol{x^A})$ as $\boldsymbol{E_k}$

Illustrative Example

Variable Name	Description	Monotone in probability of default
Response: $y =$ **default indicator**	$y = 1$ if account defaulted and $y = 0$ if it did not default	
Predictors		
x1 = avg bal cards *std*	Average monthly debt standardized: amount owed by applicant) on all of their credit cards over last 12 months	**N**
x2 = credit age *std*	Age in months of first credit product standardized: first credit cards, auto-loans, or mortgage obtained by the applicant	**Y = Decreasing**
x3 = pct over 50 uti	Percentage of open credit products (accounts) with over 50% utilization	**N**
x4 = tot balance *std*	Total debt standardized: amount owed by applicant on all of their credit products (credit cards, auto-loans, mortgages, etc.)	**N**
x5 = uti open card	Percentage of open credit cards with over 50% utilization	**N**
x6 = num acc 30d past due 12 months	Number of non-mortgage credit-product accounts by the applicants that are 30 or more days delinquent within last 12 months (Delinquent means minimum monthly payment not made)	**Y = Increasing**
x7 = num acc 60d past due 6 months	Number of non-mortgage credit-product accounts by the applicants that are 30 or more days delinquent within last 6 months	**Y = Increasing**
x8 = tot amount currently past due *log*	Total debt standardized: amount owed by applicant on all of their credit products – credit cards, auto-loans, mortgages, etc.	**Y = Increasing**
x9 = num credit inq 12 month	Number of credit inquiries in last 12 months. An inquiry occurs when the applicant's credit history is requested by a lender from the credit bureau. This occurs when a consumer applies for credit.	**Y = Increasing**
x10 = num credit card inq 24-month	Number of credit card inquiries in last 24 months. An inquiry occurs when the applicant's credit history is requested by a lender from the credit bureau. This occurs when a consumer applies for credit.	**Y = Increasing**

Simulated Data:
- 50,000 accounts
- Default or not in 18 months
- 10 predictors
- Distributions of predictors mimic bureau data

Fitted Model:
- Feedforward NN
- Constrained to be monotone in indicated variables

Correlation

x1 = avg bal cards std
x2 = credit age std flip
x3 = pct over 50 uti
x4 = tot balance std
x5 = uti open card
x6 = num acc 30d past due 12 months
x7 = num acc 60d past due 6 months
x8 = tot amount currently past due log
x9 = num credit inq 12 month
x10 = num credit card inq 24-month

- Block correlation among similar features
- High-levels

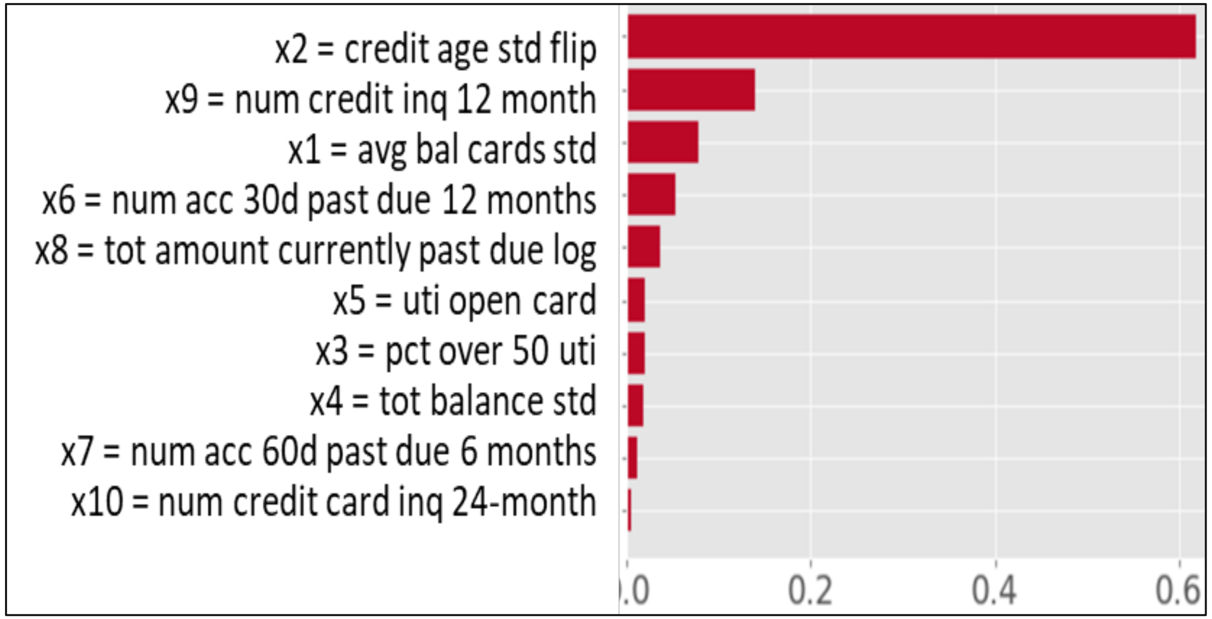
Training Monotone Neural Network

- Iterative algorithm: Fit with a penalty for monotonicity; certify; and iterate
- 50,00 accounts → training: 80%, validation and training: 10% each
- Final model: three hidden layers with dimensions [35, 15, 5]; learning rate (LR) = 0.001
- For comparison:
  - fitted unconstrained Feedforward Neural Network (FFNN) [23, 35, 15]; LR = 0.004

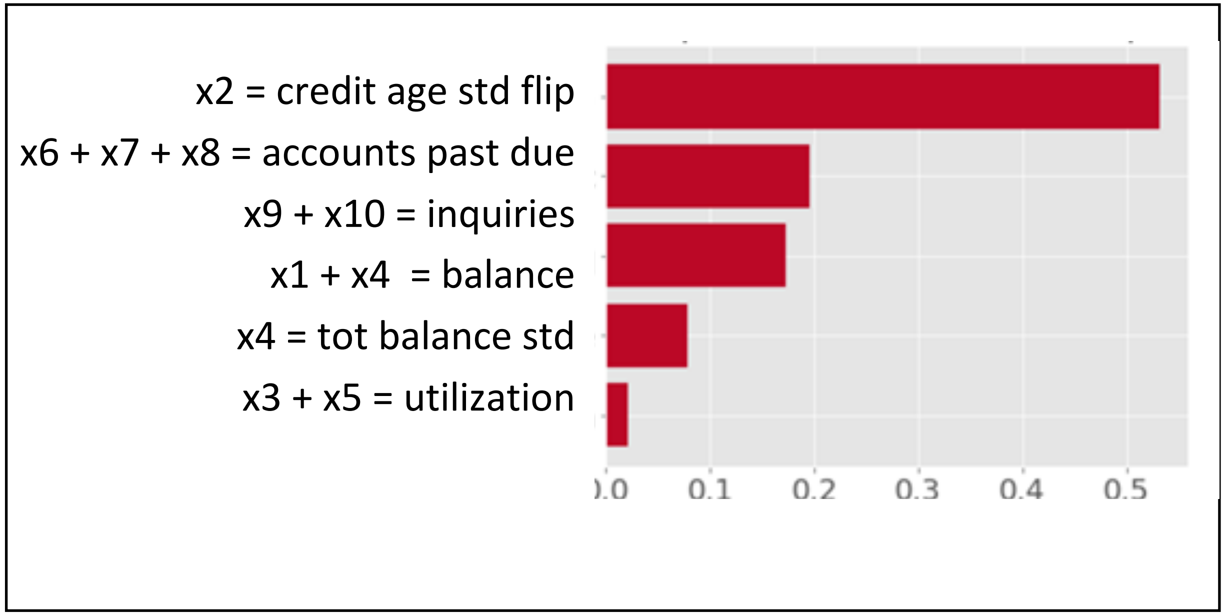
Training and Test AUCs for the Two Algorithms

Algorithm	Training AUC	Test AUC
FFNN	0.810	**0.787**
M-NN	0.807	**0.797**

Variable Importance for Mono-NN: All and Correlated

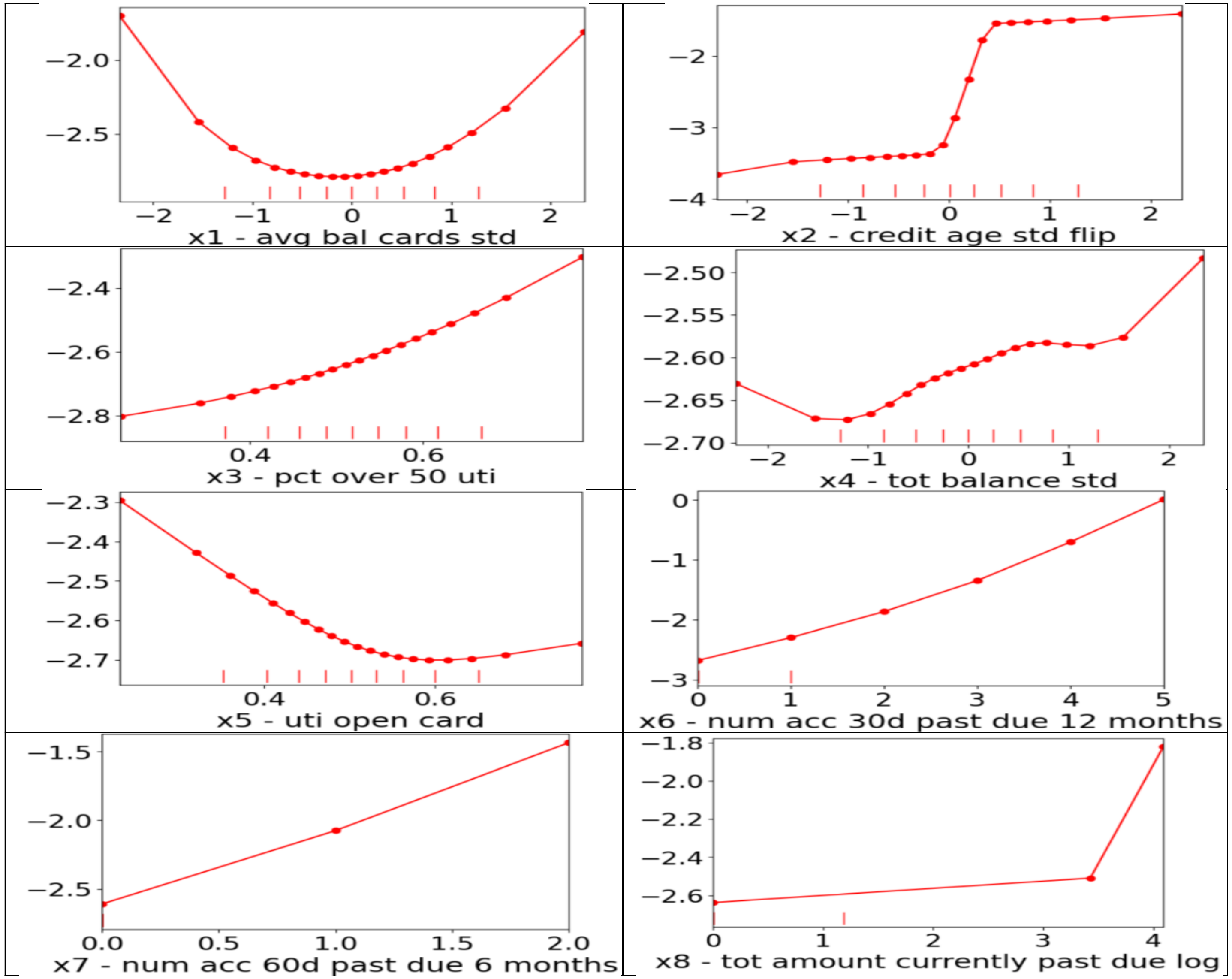


Individual Variable Importance



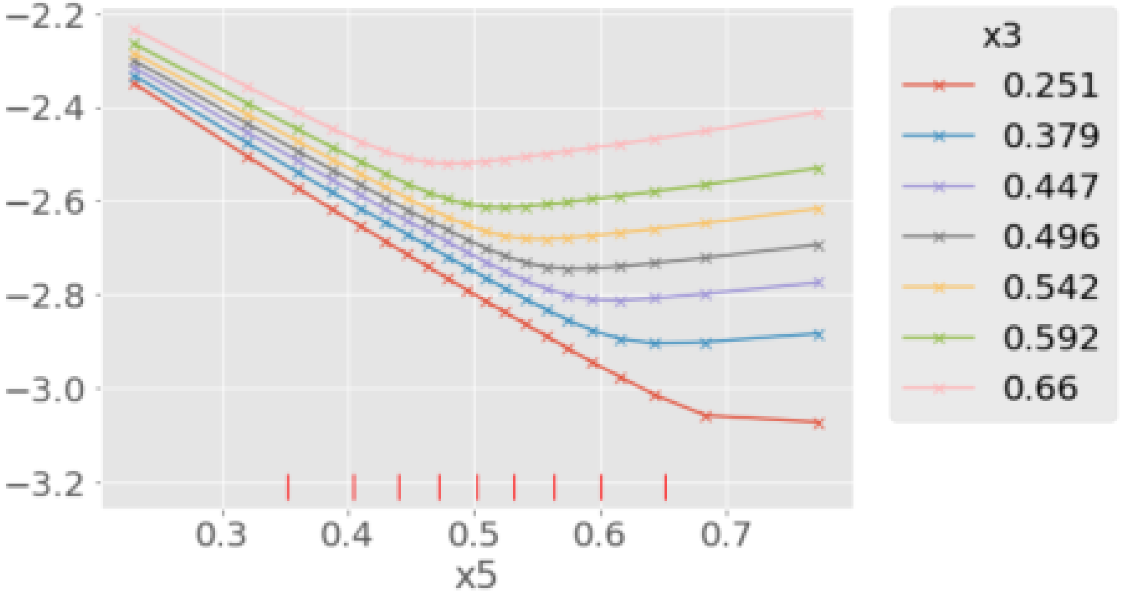
Joint Variable Importance for Correlated predictors

PDPs for Mono-NN

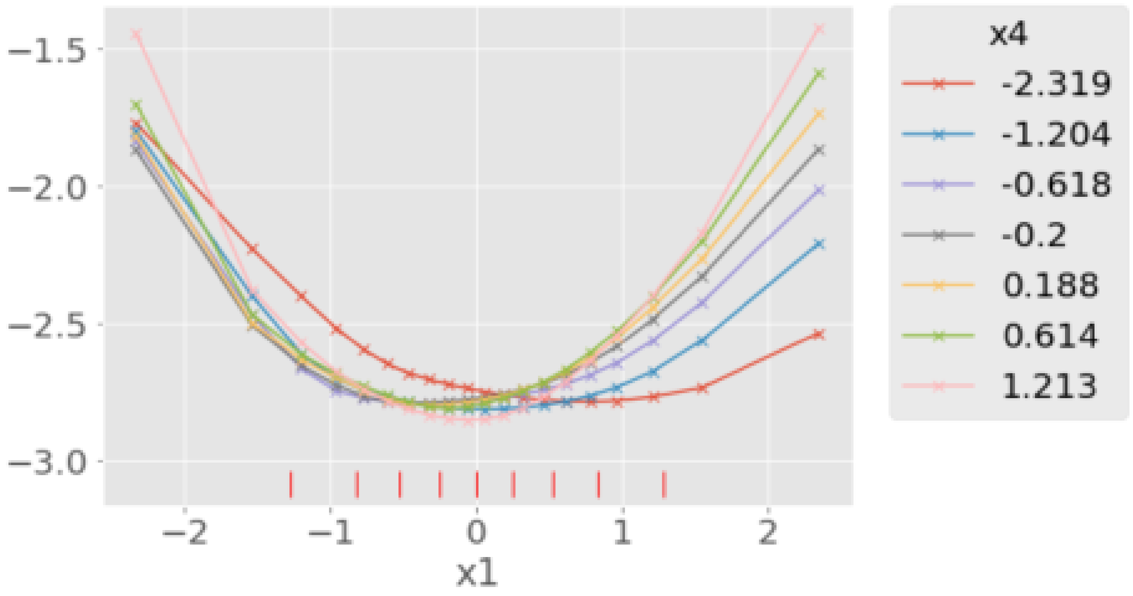


Two-dimensional PDPs of Variables with Interactions

x5 = uti open card   x3 = pct over 50 uti



x1 = avg bal cards std   x4 = tot balance std



AA explanation: Decision rule – decline if $p(x) > 0.25$

Predictors	$\boldsymbol{x^A}$	$\boldsymbol{x_1^D}$	M-NN Attributions for $\boldsymbol{x_1^D}$	$\boldsymbol{x_2^D}$	M-NN Attributions for $\boldsymbol{x_2^D}$
x1 = avg bal cards std	-0.006	0.674	0.112 (3.5%)	0.519	0.028 (0.5%)
x2 = credit age std flip	-0.733	0.886	**1.928 (59.5%)**	0.431	**1.565 (26.5%)**
x3 = pct over 50 uti	0.518	0.531	0.001 (0.0%)	0.522	-0.001 (0.0%)
x4 = tot balance std	-0.001	0.562	−0.008 (0.2%)	1.968	-0.201 (−3.4%)
x5 = uti open card	0.501	0.577	0.012 (0.4%)	0.525	-0.024 (−0.4%)
x6 = num acc 30d past due 12 months	0.000	0.000	0.0 (0.0%)	4.000	**1.850 (31.3%)**
x7 = num acc 60d past due 6 months	0.000	0.000	0.0 (0.0%)	2.000	**0.984 (16.6%)**
x8 = tot amount currently past due std	0.000	0.000	0.0 (0.0%)	4.379	**1.712 (28.9%)**
x9 = num credit inq 12 month	0.000	3.000	**1.010 (31.2%)**	0.000	0.0 (0.0%)
x10 = num credit inq 24 month	0.000	4.000	0.186 (5.7%)	0.000	0.0 (0.0%)
$\hat{\boldsymbol{p}}(\boldsymbol{x})$	0.016	0.294		0.858	
$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{logit}(\hat{\boldsymbol{p}}(\boldsymbol{x}))$	−4.117	−0.876		1.797	

AA explanation: Decision rule – decline if $p(x) > 0.25$

Predictors	x^A	x_1^D	M-NN Attributions for x_1^D
x1 = avg bal cards std	-0.006	0.674	0.112 (3.5%)
x2 = credit age std flip	-0.733	0.886	**1.928 (59.5%)**
x3 = pct over 50 uti	0.518	0.531	0.001 (0.0%)
x4 = tot balance std	-0.001	0.562	−0.008 (0.2%)
x5 = uti open card	0.501	0.577	0.012 (0.4%)
x6 = num acc 30d past due 12 months	0.000	0.000	0.0 (0.0%)
x7 = num acc 60d past due 6 months	0.000	0.000	0.0 (0.0%)
x8 = tot amount currently past due std	0.000	0.000	0.0 (0.0%)
x9 = num credit inq 12 month	0.000	3.000	**1.010 (31.2%)**
x10 = num credit inq 24 month	0.000	4.000	0.186 (5.7%)
$\hat{p}(x)$	0.016	0.294	
$f(x) = logit(\hat{p}(x))$	−4.117	−0.876	

- Can modify to get combined explanation for groups of correlated predictors

Groups of predictors	M-NN Attributions for x_1^D
balance	0.126 (3.9%)
credit age std flip	**1.925 (59.4%)**
utilization	0.018 (0.5%)
num acc	0.000 (0.0%)
num inq	**1.173 (36.2%)**

AA explanation: Decision rule – decline if $p(x) > 0.25$

Predictors	x^A	x_2^D	M-NN Attributions for x_2^D
x1 = avg bal cards std	-0.006	0.519	0.028 (0.5%)
x2 = credit age std flip	-0.733	0.431	**1.565 (26.5%)**
x3 = pct over 50 uti	0.518	0.522	-0.001 (0.0%)
x4 = tot balance std	-0.001	1.968	-0.201 (−3.4%)
x5 = uti open card	0.501	0.525	-0.024 (−0.4%)
x6 = num acc 30d past due 12 months	0.000	4.000	**1.850 (31.3%)**
x7 = num acc 60d past due 6 months	0.000	2.000	**0.984 (16.6%)**
x8 = tot amount currently past due std	0.000	4.379	**1.712 (28.9%)**
x9 = num credit inq 12 month	0.000	0.000	0.0 (0.0%)
x10 = num credit inq 24 month	0.000	0.000	0.0 (0.0%)
$\hat{p}(x)$	0.016	0.858	
$f(x) = logit(\hat{p}(x))$	−4.117	1.797	

- Combined explanation for groups of correlated predictors

Groups of predictors	M-NN Attributions for x_2^D
balance	-0.328 (-5.5%)
credit age std flip	**1.785 (30.2%)**
utilization	-0.018 (-0.3%)
past due	**4.476 (75.7%)**
num inq	0.000 (0.0%)

Outline

- Introduction: Challenges and concepts
- Model interpretability
  - Post hoc techniques for model explanation
  - Inherently-interpretable ML algorithms
- Diagnostics for model weakness
  - Predictive performance
    - Global and local
    - Generalizability
  - Robustness
- Bias and fairness
- Discussion

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**

Example 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** $\boldsymbol{g_j(x_j)}$**, two-factor interactions** $\boldsymbol{g_{jk}(x_j, x_k)}$
- **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

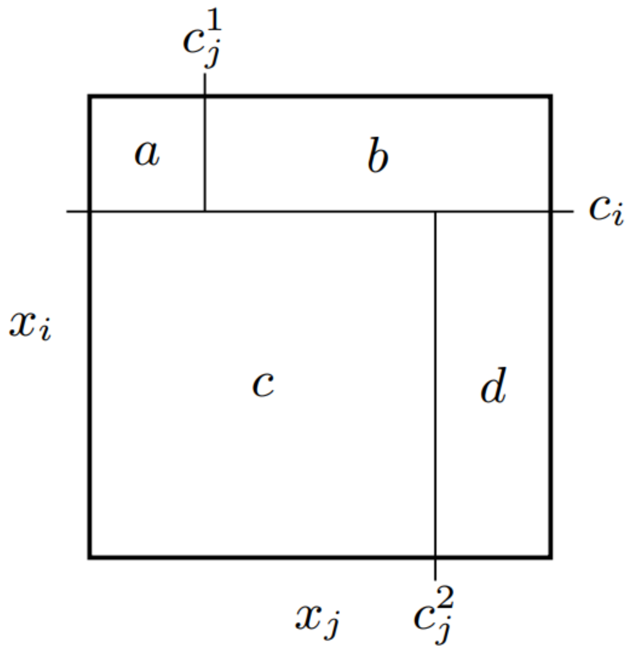
Explainable Boosting Machine

- **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

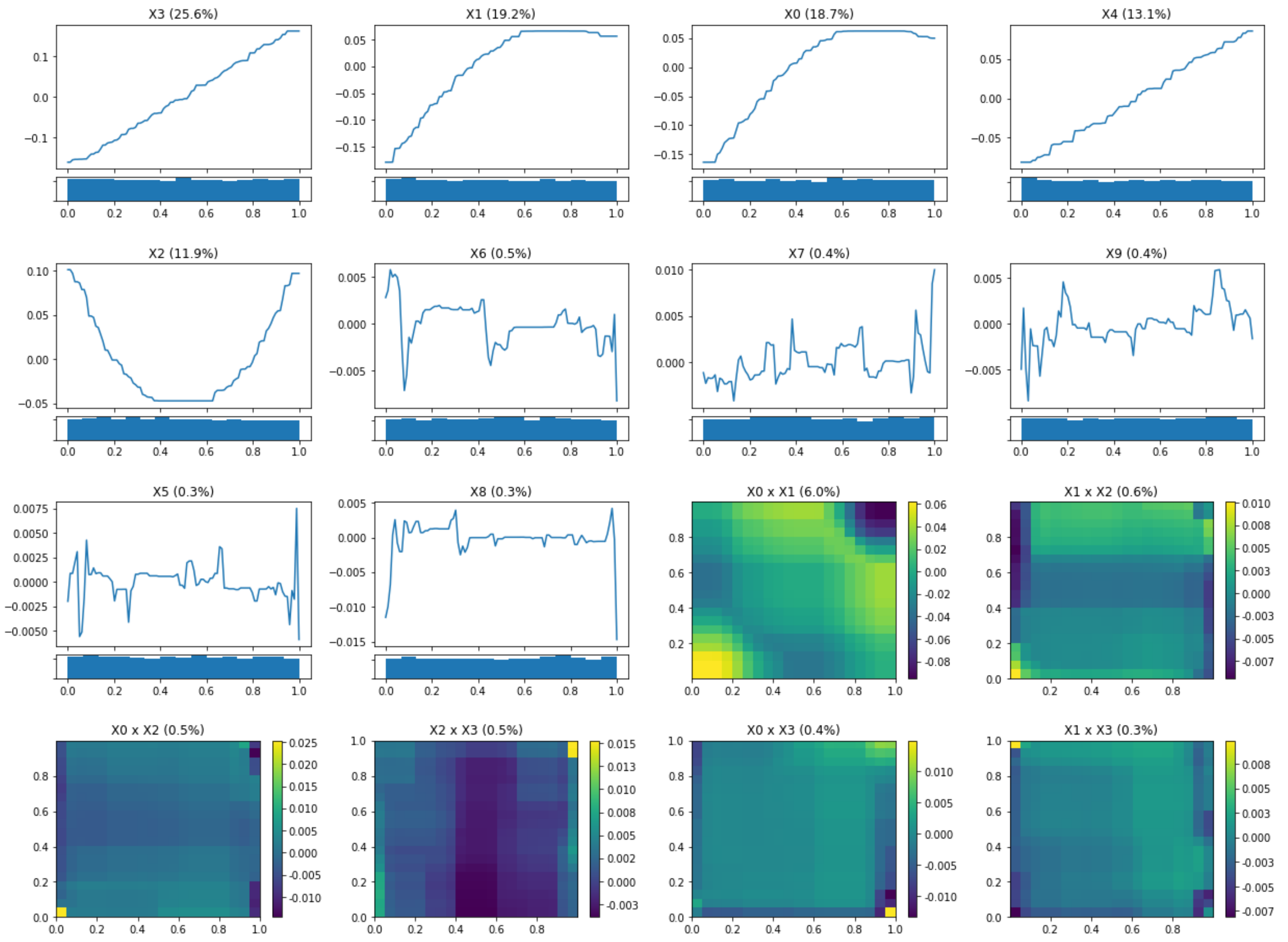
Friedman1 simulated data:

- 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 $\boldsymbol{x_0} \sim \boldsymbol{x_4}$

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



GAMI-Net

- NN-based algorithm for non-parametrically fitting

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

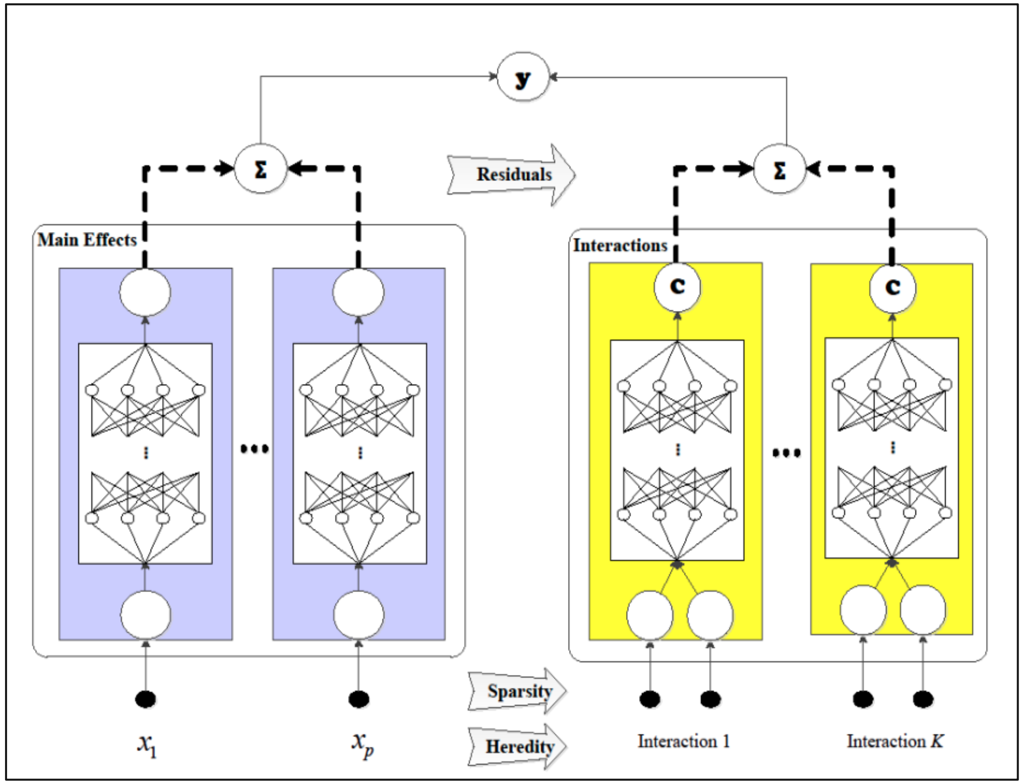
- **Multi-stage training algorithm:**

  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).

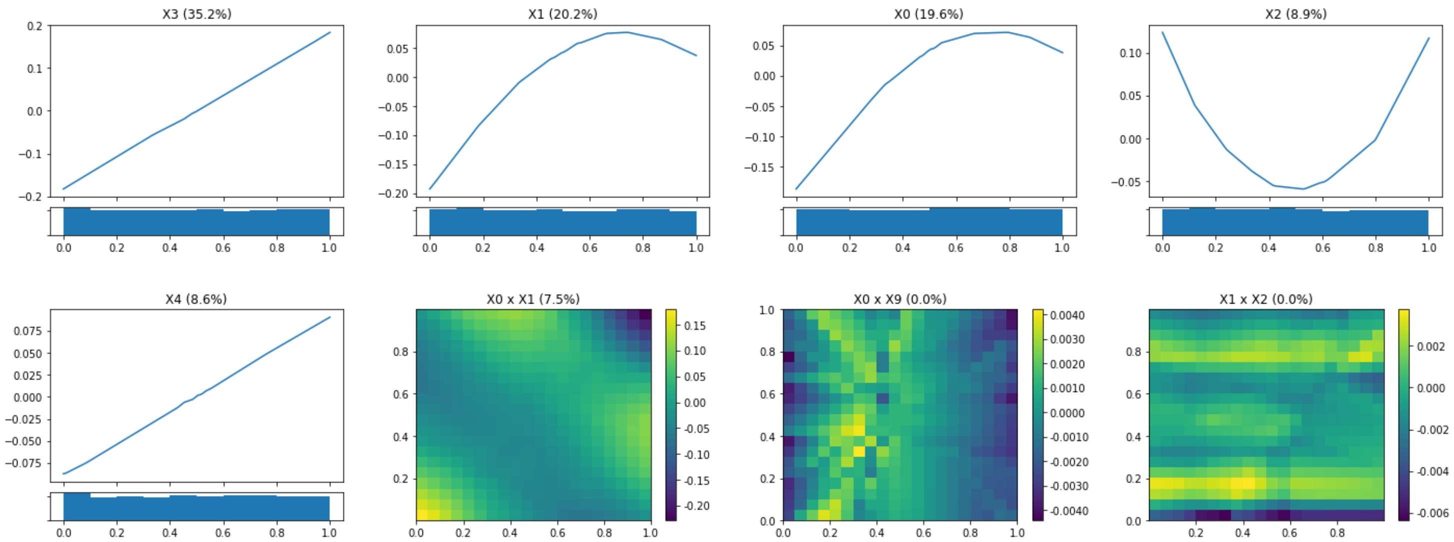
GAMI-Net: Example

Friedman1 data:

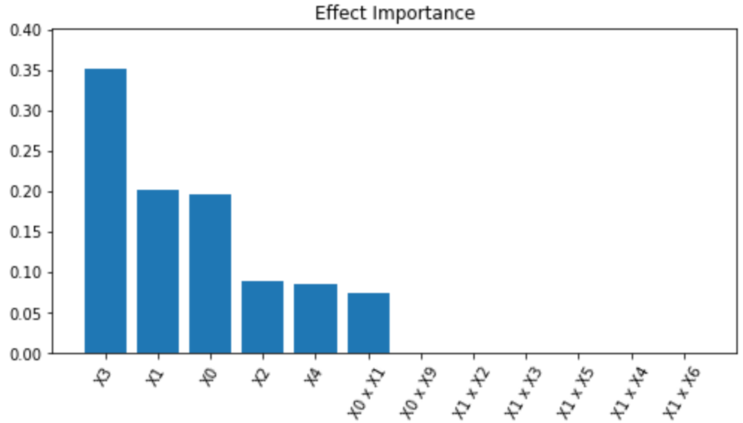
$$y(\boldsymbol{x}) = 10\sin(\pi x_0 x_1) + 20(x_2 - 0.5)^2 + 20x_3 + 10x_4 + \epsilon$$

Same data generated as for EBM example.

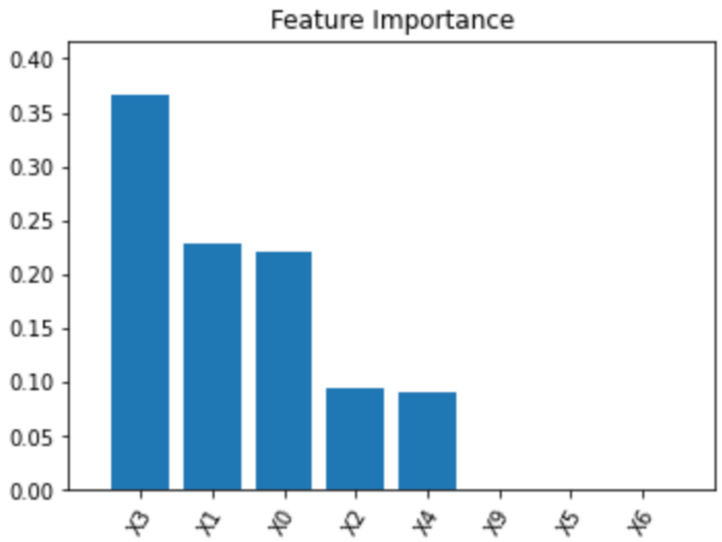
GAMI-Net Output with Test RMSE = 0.0058 and R2 = 99.89%




```
model_gaminet.show_effect_importance()
```




```
model_gaminet.show_feature_importance()
```

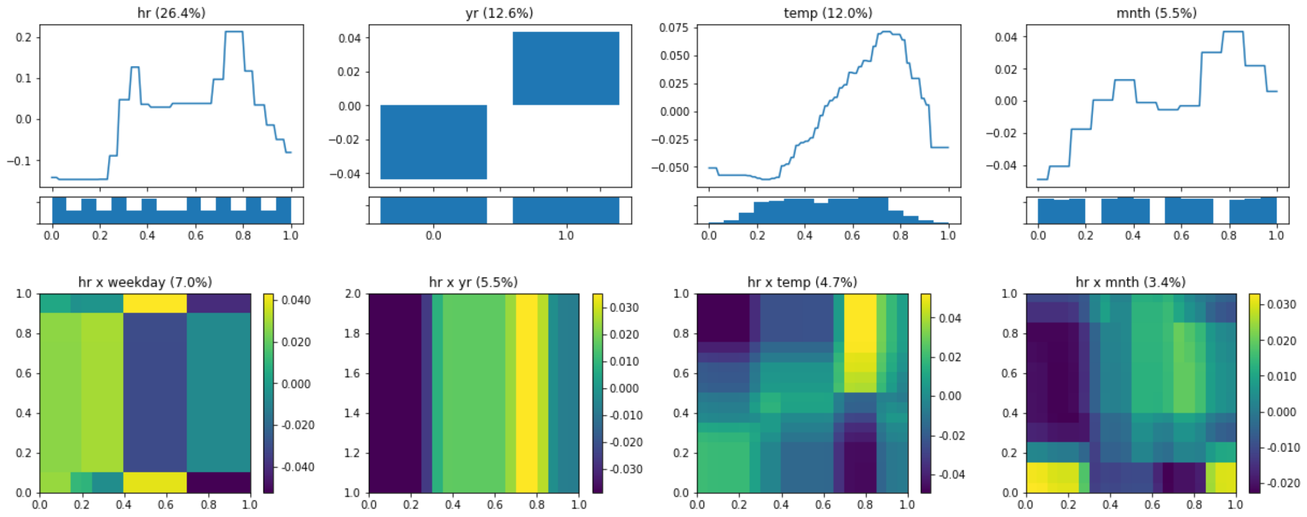


Comparisons: Bike Sharing Data

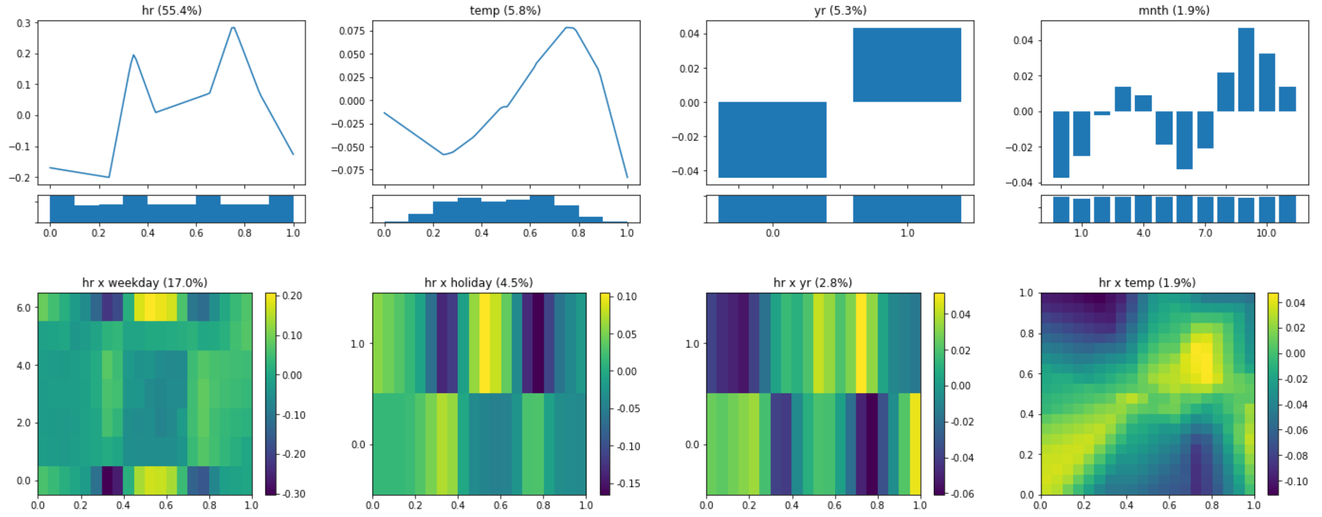
Bike sharing data:

- 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:

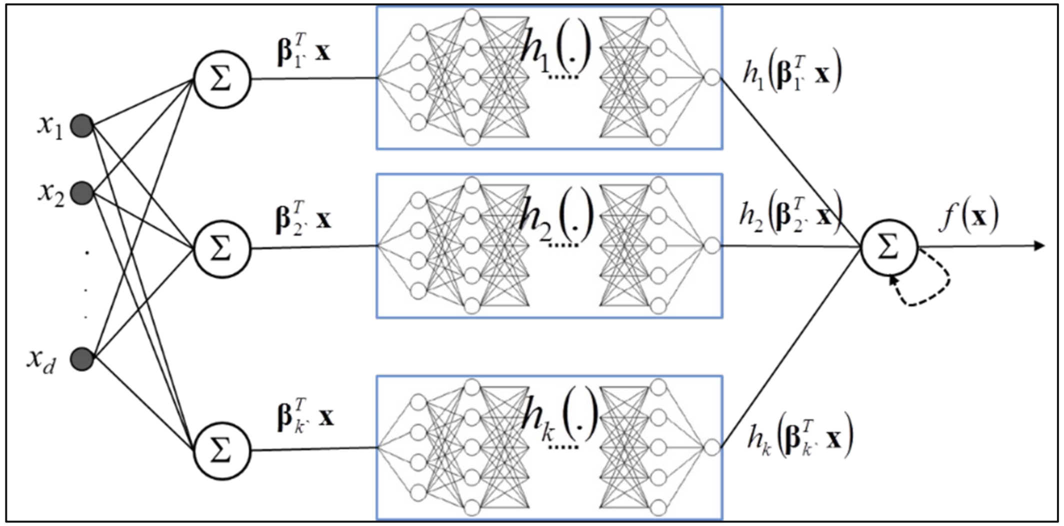
- **Additive Index Models:**

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

  - 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)

Outline

- Introduction: Challenges and concepts
- Model interpretability
  - Post hoc techniques for model explanation
  - Inherently-interpretable ML algorithms
- Model diagnostics
- Discussion

Outline: Model diagnostics

- Overview of diagnostics
- Global predictive performance
- Model performance weakness
  - Local analysis using supervised partitioning
  - Unsupervised analysis of residuals
- Generalizability on unseen data
- Uncertainty quantification
- Model stability assessment (Robustness)
  - Perturbations in the X-space
  - Perturbations in Y-space

Model Weaknesses

Types	Possible causes

Poor Predictive Performance
• Overall (compared with other algorithms)
• Locally in some regions
	• Limitations of algorithm
	  • Missing important predictors/interactions
	  • Not flexible enough to capture varying structure
	• Heterogeneity in the data
	  • Model changes over time
	  • Different segments with varying behavior
	  • Data sparsity in certain regions

Inconsistent with subject-matter knowledge	• Missing important predictors/ interactions • Wrong predictors/interactions in model • Reason: high correlation • Incorrect feature engineering • Desired monotonicity not enforced
Does not generalize well in hold-out data	• Out-of-time: model changes over time • Tails of dataset: behavior in tail different from training data • In data-sparse regions: not enough data • Poor extrapolation behavior – piecewise constant trees

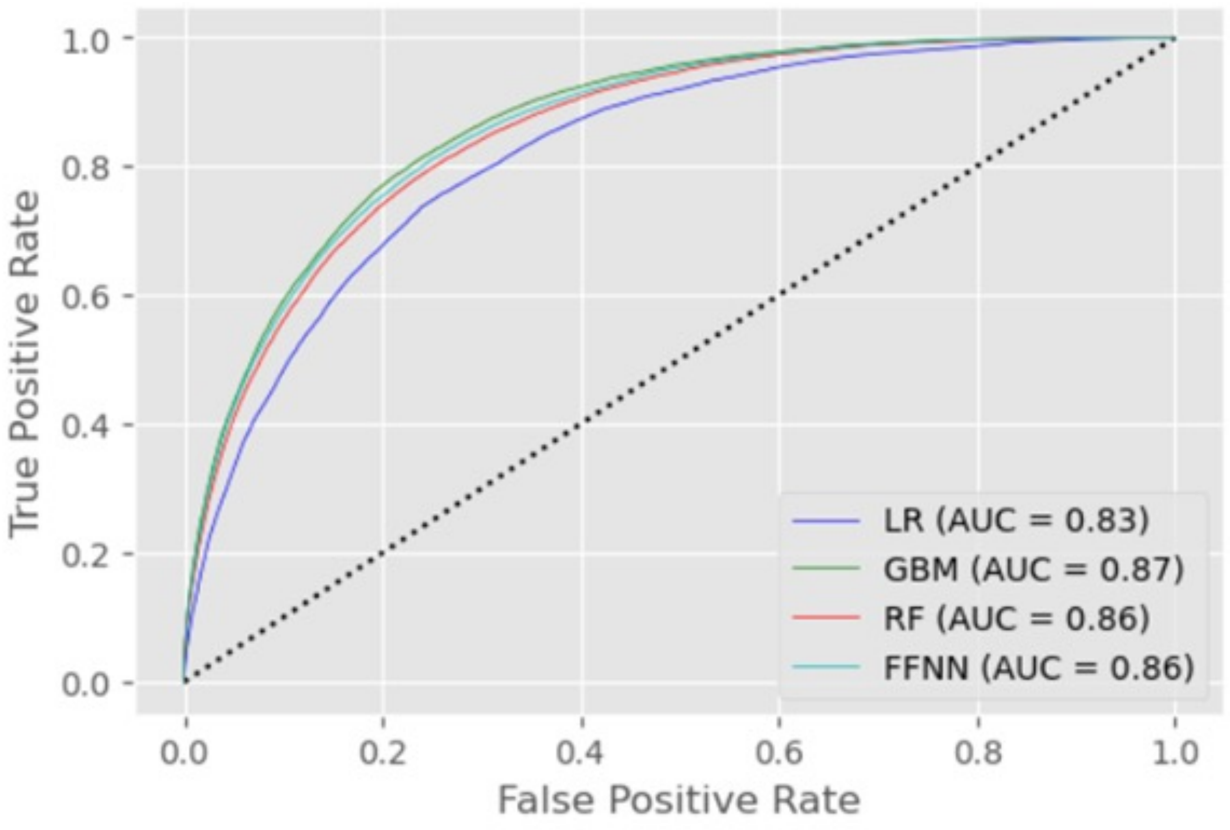
Lack of Robustness
• Overfitting
• Sensitive to small input perturbations
	• Model is too flexible
	  • Overly “parametrized”
	  • Needs regularization

Assessing Model Performance

- Traditional approaches listed below:
- **Which of these can be used with modern ML algorithms?**
- **Predictive Performance Assessment**
  - Performance metrics (MSE, R-squared, AUC, etc.)
  - Std errors and p-values of estimated coefficients
  - Hold-out sample (in-sample and out-of-sample) prediction accuracy
  - Comparisons with challenge/benchmark models, …
- **Model Diagnostics**
  - Checking model assumptions: Linearity/nonlinearity, interactions, regime change, …
  - Checking error structure: equal variance, independence, stationarity, seasonality, etc.
    - Use of residual plots, QQ plots, …

Comparing global predictive performance

- How does the algorithm's predictive performance compare against peers – "benchmark model(s)"
- Illustrative example:
  - Home mortgage loans
  - Response: binary ("in trouble loans"
  - Twenty-two predictors
    - loan-to-value ratio;
    - credit score over time;
    - Before or after financial crisis
    - Unemployment rate;
    - income;
    - delinquency status, etc.
- Comparison of Logistic Regression against
  - Random Forest
  - Gradient Boosting
  - Feedforward Neural Networks
- 4-5% improvement → Is this good improvement?
- Why?
  - Missing predictors?
  - Interactions, transformations?
  - use this information to improve
- Typically, look at multiple metrics



Model weakness and residual analysis

Analyzing residuals of fitted model

- Rationale
  - If original model does not fit well, there will be structure in the residuals
  - Analyze the residuals in different ways to identify possible remaining structure and understand possible reasons
- Fit a different global model to the residuals
  - For example: original model is logistic regression → we fit (new) XGB model to residuals
  - Does the combined fit produce considerable improvement in predicted performance
  - If yes, identify possible reasons:
    - important predictors/interactions present in new model for residuals?
    - wrong feature engineering in original model?
- Supervised partitioning of residuals
  - Fit a tree to identify regions where the model does not fit as well
  - Try to understand causes of the poor fit
- Unsupervised analysis of residuals
  - Just pick the top K% of residuals (in absolute value) and examine how the predictors for the top ones are different
- Definition of residual
  - Obvious for continuous response: residual $r_i = (y_i - \hat{y}_i)$
  - Not clear for binary response → several choices and challenges

Global analysis of residuals using simulation study

- Functional form

$$f(x) = \beta_0 x_0 + \cdots + \beta_7 x_7 + \beta_8 x_8^2 + \beta_9 x_9^2 + \beta_{01} x_0 x_1 + \beta_{23} x_2 x_3 + \beta_{02} x_0 x_2 + \beta_{14} x_1 x_4$$

- Original model(target model) : Random Forest
- Fit XGB model to residuals in **test data**
- **Diagnose** results from second model using
  - Variable importance
  - PDP and ICE Plots

Results of fitting XGB model to residuals from RF target model

$$f(x) = \beta_0 x_0 + \cdots + \beta_7 x_7 + \beta_8 x_8^2 + \beta_9 x_9^2 + \beta_{01} x_0 x_1 + \beta_{23} x_2 x_3 + \beta_{02} x_0 x_2 + \beta_{14} x_1 x_4$$

RF Target Model

Predictive Performance

	MSE	R-Squared
Train	1.009	0.890
Test	3.481	0.617

Very poor performance in test data

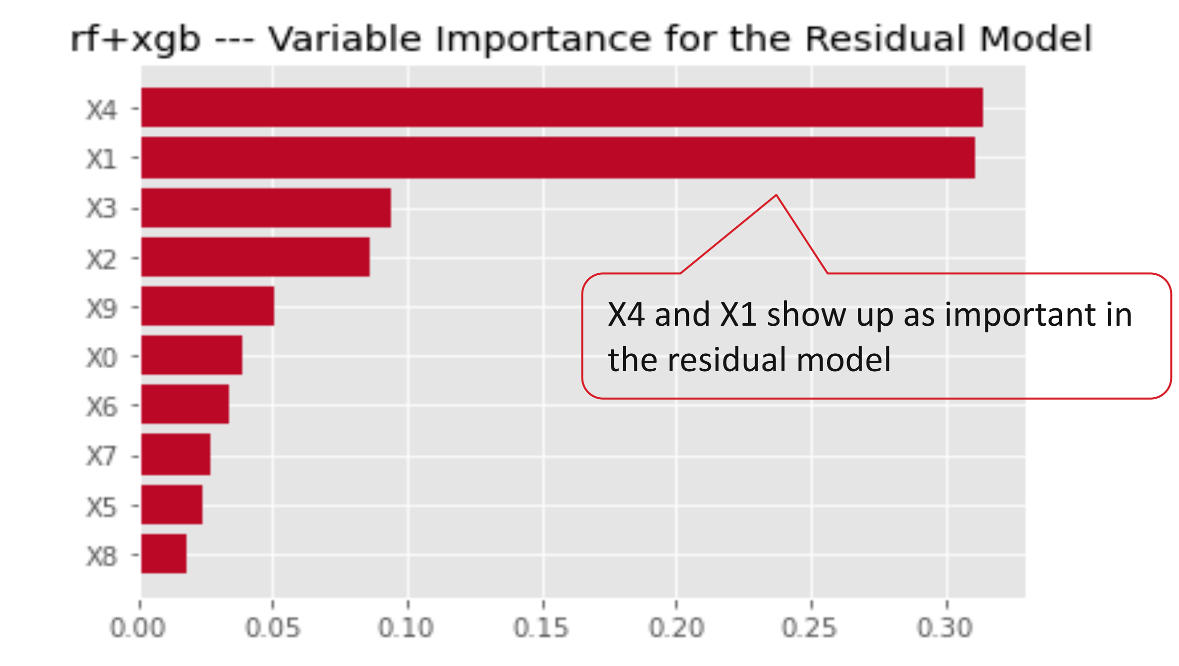
Combined performance after fitting XGB to Residuals

Predictive Performance

	MSE	Rsquared
Test	1.356	0.851

Diagnosing reasons for poor performance of RF

- Variable importance analysis of XGB Residual Model
- **Possible explanation:**
  - Interaction term $x_1 x_4$ not captured well
  - Simulation case → confirms truth
  - In practice, have to dig deeper to verify



Results of fitting XGB model to residuals from RF target model (cont'd)

$$f(x) = \beta_0 x_0 + \cdots + \beta_7 x_7 + \beta_8 x_8^2 + \beta_9 x_9^2 + \beta_{01} x_0 x_1 + \beta_{23} x_2 x_3 + \beta_{02}\, x_0 x_2 + \beta_{14}\, x_1 x_4$$

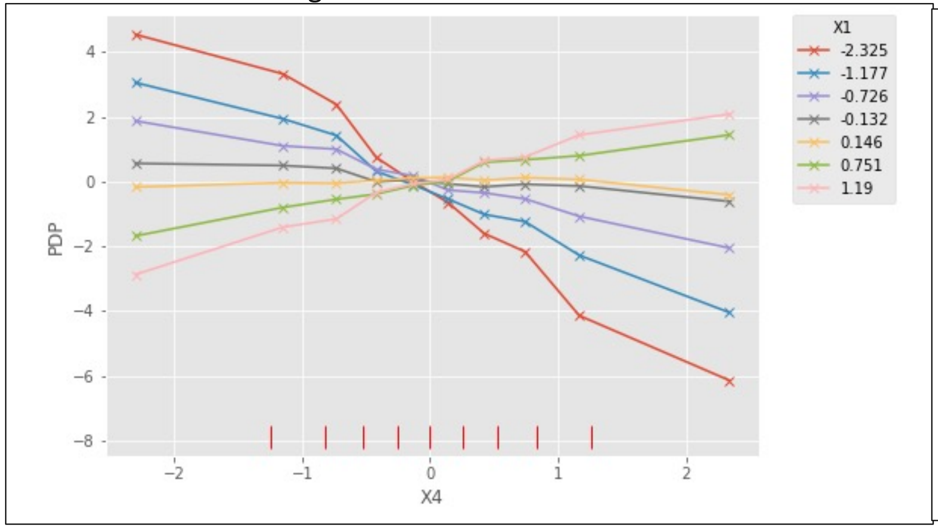
Digging deeper into interactions …

Unscaled H-statistics

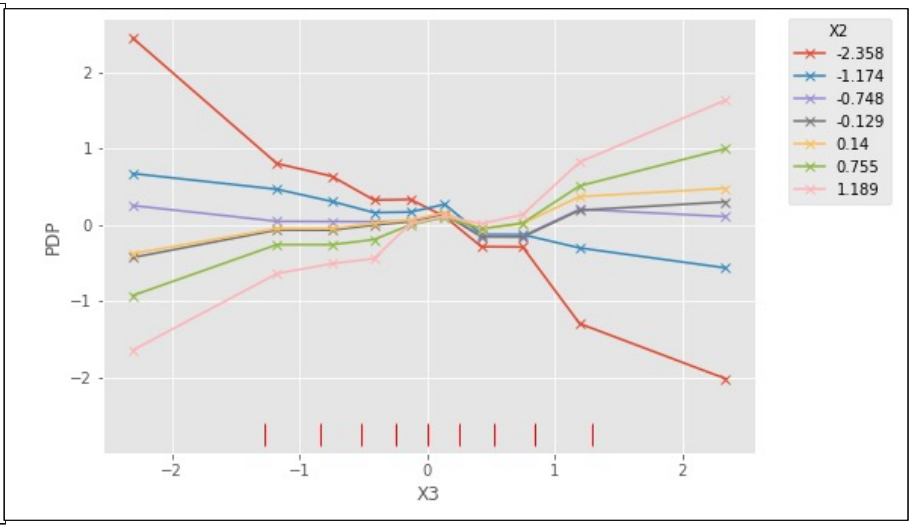
X1 - X4	1.0601
X2 - X3	0.3775
X0 - X1	0.1600
X0 - X2	0.1588
X1 - X8	0.0782
X5 - X7	0.0563
X2 - X4	0.0563
X1 - X7	0.0512

H-stats for leftover interactions

rf+xgb ---2d-PDP of X1 – X4



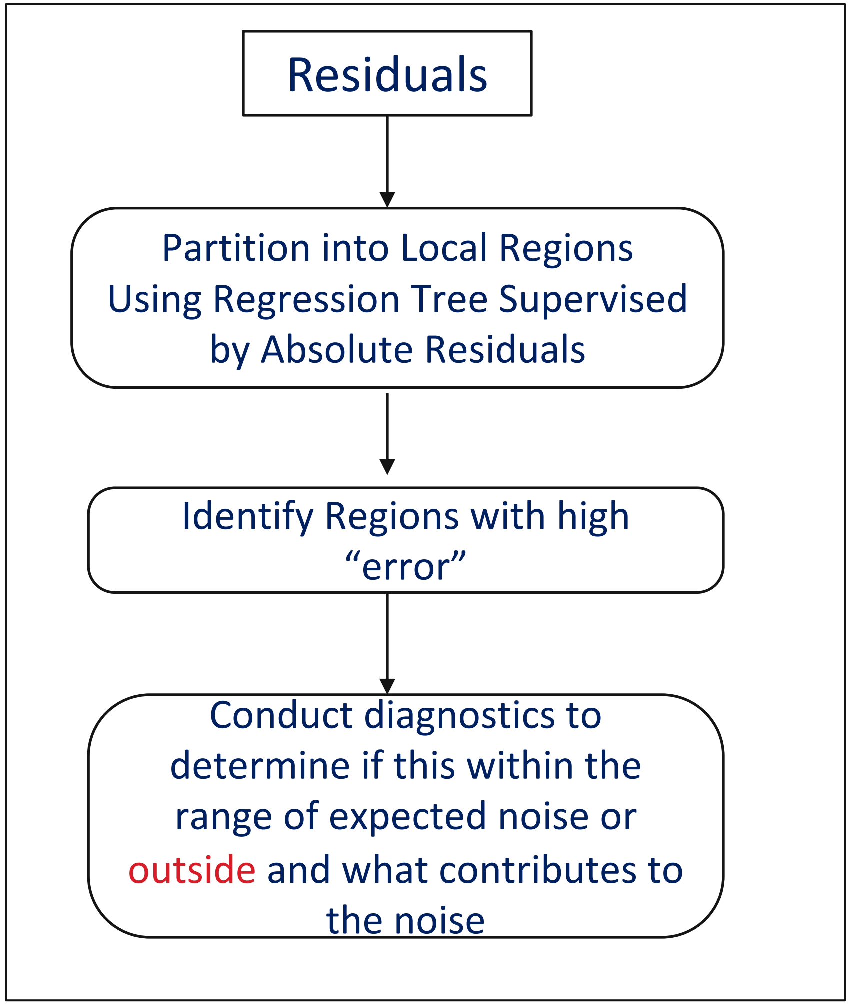
rf+xgb ---2d-PDP of X2 – X3



2D-PDPs show leftover Interactions from RF model for **x1-x4** and **x2-x3**

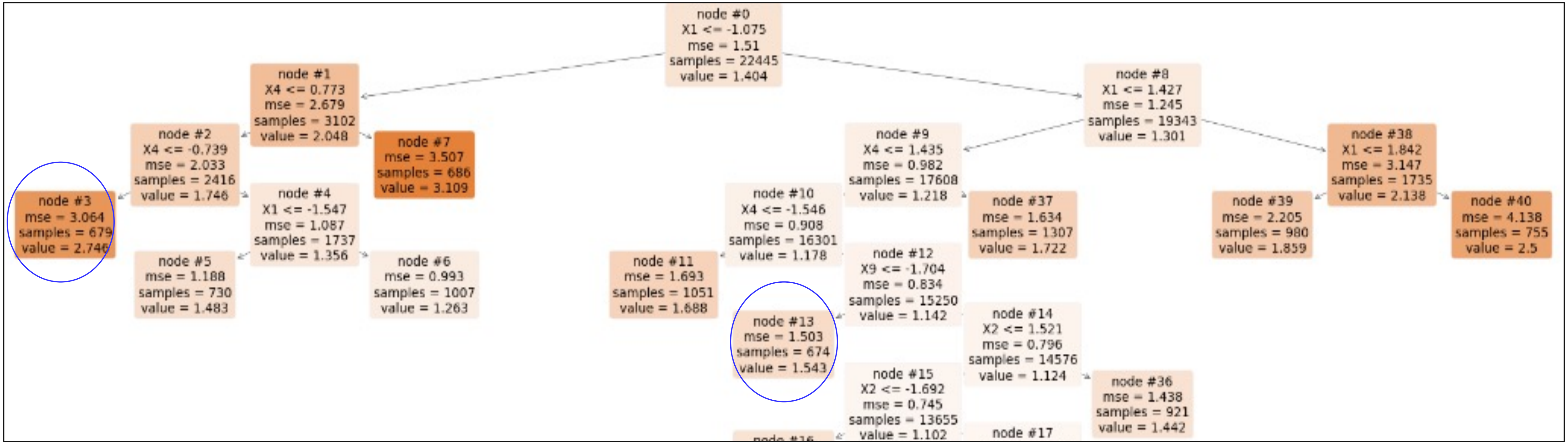
Supervised partitioning of residuals

- Goal
  - Identify local regions in feature space where target model has poor fit.
  - Identify the root cause of poor performance
    - non-captured non-linearity, interaction effects, etc.
- Strategy
  - Fit regression tree → $|residuals|$
  - Identify:
    - Leaf nodes with high error (MAE, MSE, log-loss, etc.)
    - Determine if high error is within expected noise or not
    - If not, determine the causes (features contributing to this)
- Diagnostics for last two steps
  - Identify features and splits along the path of node with error
  - Use tests of hypothesis (informally) to determine if error is high
  - Examine which of the features in the splits and their interactions are important



Fitting decision tree in our example

$$f(x) = \beta_0 x_0 + \cdots + \beta_7 x_7 + \beta_8 x_8^2 + \beta_9 x_9^2 + \beta_{01} x_0 x_1 + \beta_{23} x_2 x_3 + \beta_{02}\, x_0 x_2 + \beta_{14}\, x_1 x_4$$



- Multiple splits on **X1** and **X4** → hint that their effect might not be captured correctly.
- **X9** and **X2** show up in the splits as well.

- Leaf nodes #3, #7, #40, #37, #11 and #13 are among the nodes with highest MAE.
- We dig deeper into nodes #3 and #13 in the next slide.

ANOVA tables for leaf nodes #3 and #13

ANOVA Table for Node 13

	df	sum_sq	mean_sq	F	PR(>F)
C(X9)	1	1170	1170	405.8	1.91E-89
C(X4)	2	206.5	103.3	35.82	2.93E-16
C(X1)	2	125.6	62.81	21.79	3.51E-10
C(X1):C(X4)	4	11931	2983	1035	0.00E+00
C(X4):C(X9)	2	18.1	9.048	3.139	4.34E-02
C(X1):C(X9)	2	6.515	3.258	1.13	3.23E-01
Residual	22431	64658	2.883	NaN	NaN

- X9, X1 and X4 are the **split variables** that create leaf node #13.
- X9: high mean squared and low p-value. Its **main** effect is not captured
- The same for X1*X4. The **interaction** effect is not captured

- X1 and X4 are the split variables for leaf node #3.
- X1*X4: high mean squared and low p-value
- The **interaction** effect is not captured

ANOVA Table for Node 3

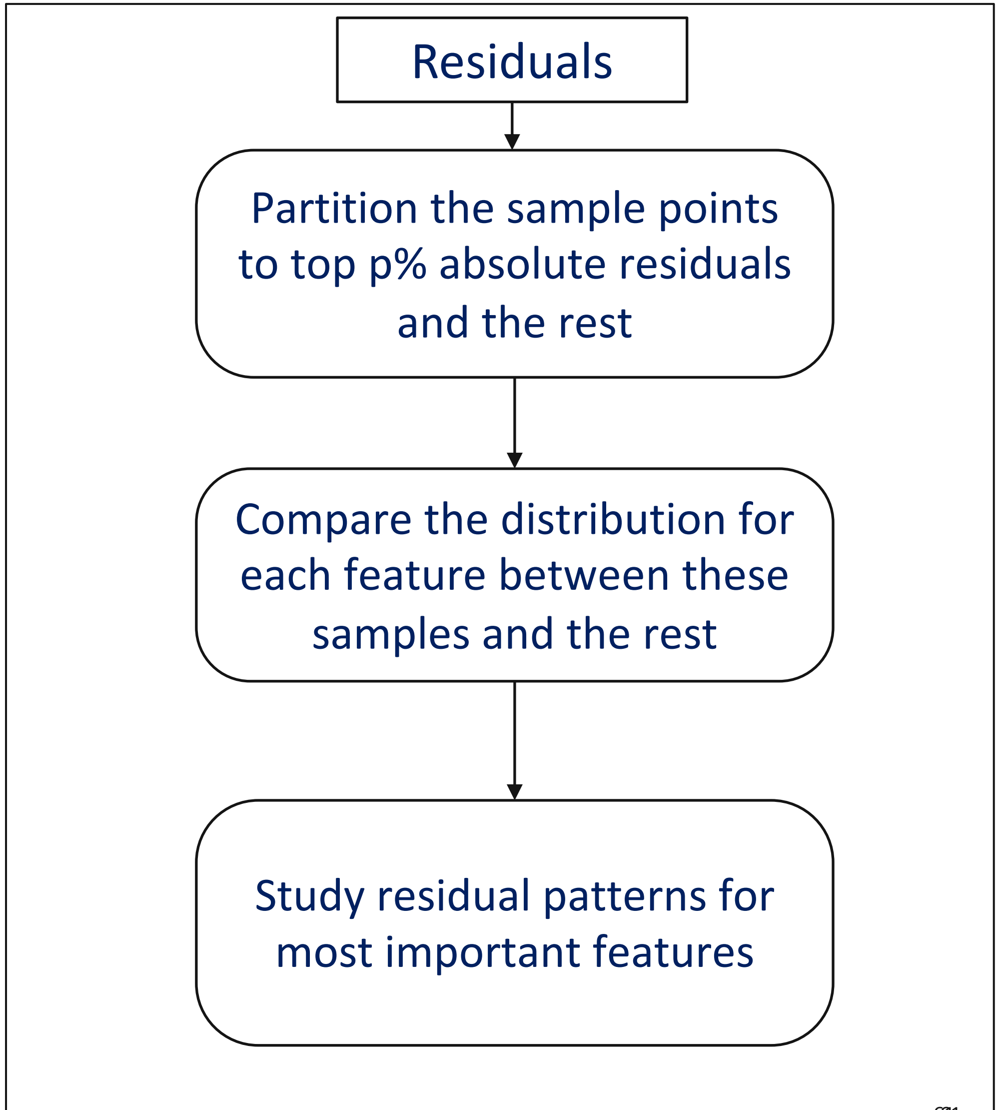
	df	sum_sq	mean_sq	F	PR(>F)
C(X4)	1	242.266	242.266	76.785	2.04E-18
C(X1)	1	58.352	58.352	18.494	1.71E-05
C(X1):C(X4)	1	7011.095	7011.095	2222.121	0.00E+00
Residual	22441	70804.415	3.155	NaN	NaN

Unsupervised analysis of residuals: Examining top p% absolute residuals

- Goal
  - Identify the differences in predictors corresponding to top residuals and the rest

- Strategy
  - Pick observations corresponding to top p% $|residual|$
  - Examine the change in distribution for each feature between worst p% and the rest using a metric like PSI
  - Pick the features with high PSI
  - Study the patterns in the residuals for these features

- Comment
  - Particularly useful when number of features is large



Unsupervised partition using same simulation example

$$f(x) = \beta_0 x_0 + \cdots + \beta_7 x_7 + \beta_8 x_8^2 + \beta_9 x_9^2 + \beta_{01} x_0 x_1 + \beta_{23} x_2 x_3 + \beta_{02} x_0 x_2 + \beta_{14} x_1 x_4$$

Results for top p = 5% of the residuals

Clear shift in histograms for X1 and X4.
Also, some shift for X2, X3 and X9.

Comparison of histograms for each feature for high residual samples and the rest of data

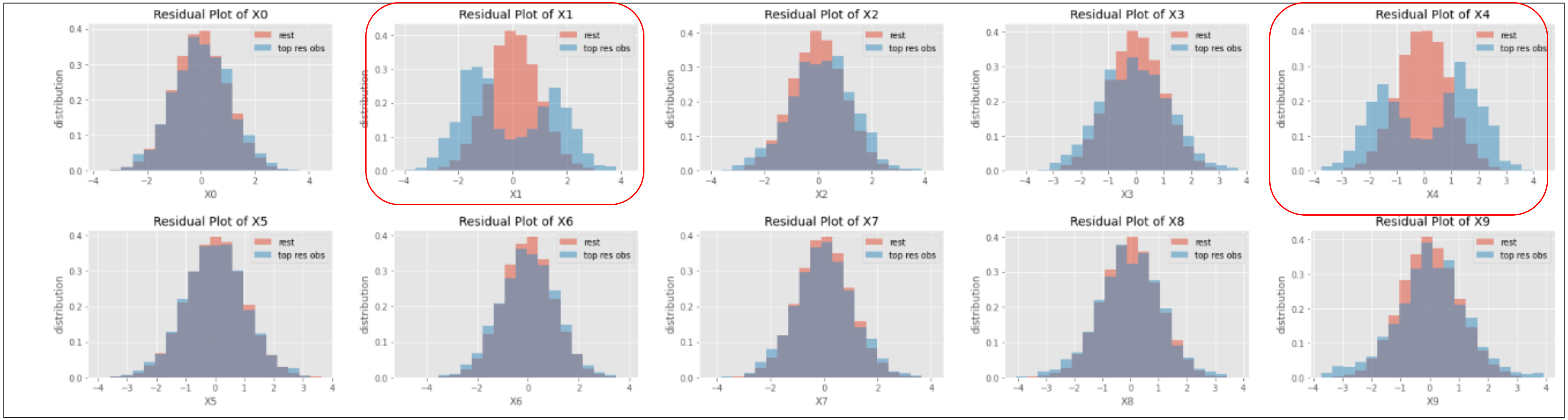


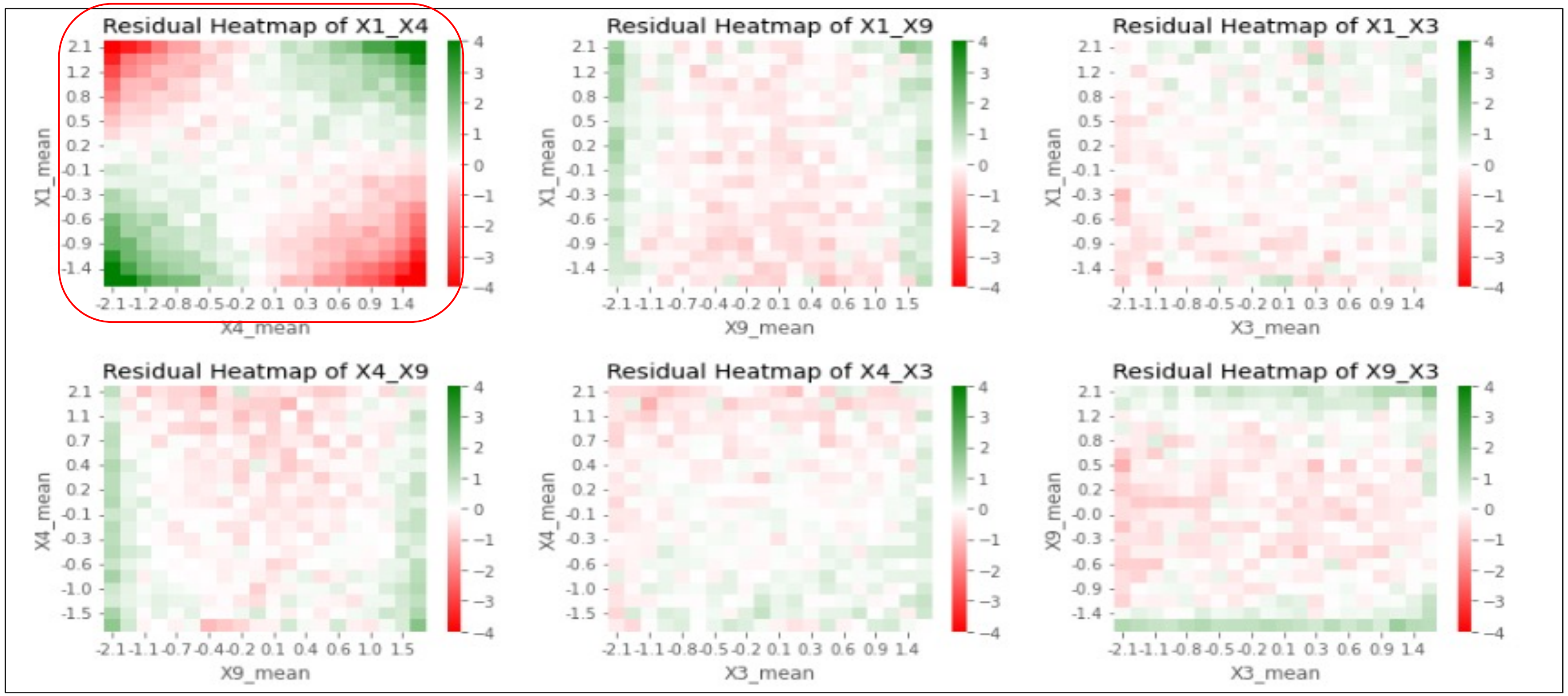
Table of PSI measure for each feature

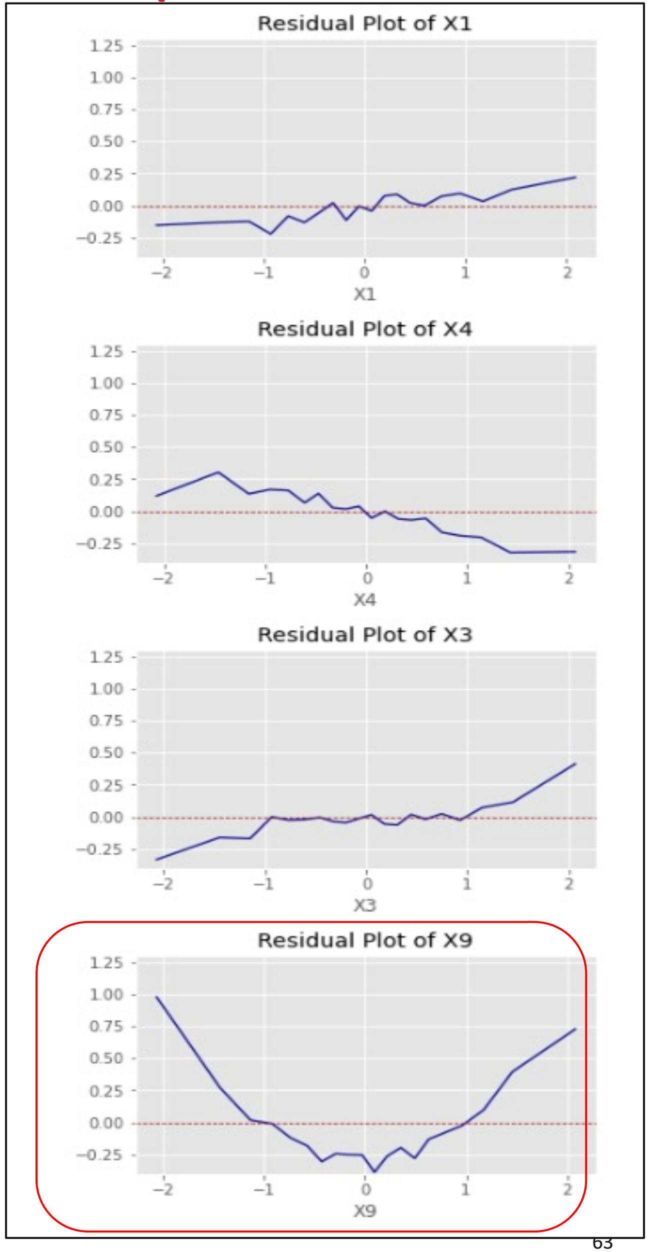
Feature	X1	X4	X3	X9	X2	X8	X0	X6	X7	X5
PSI Valuses	1.110	1.078	0.109	0.093	0.074	0.035	0.018	0.017	0.013	0.007

X1 and X4 are important

Diagnostics: Examine residual plots for leading features (X1, X4, X3, X9)

Interaction between X1 and X4 is noticeable

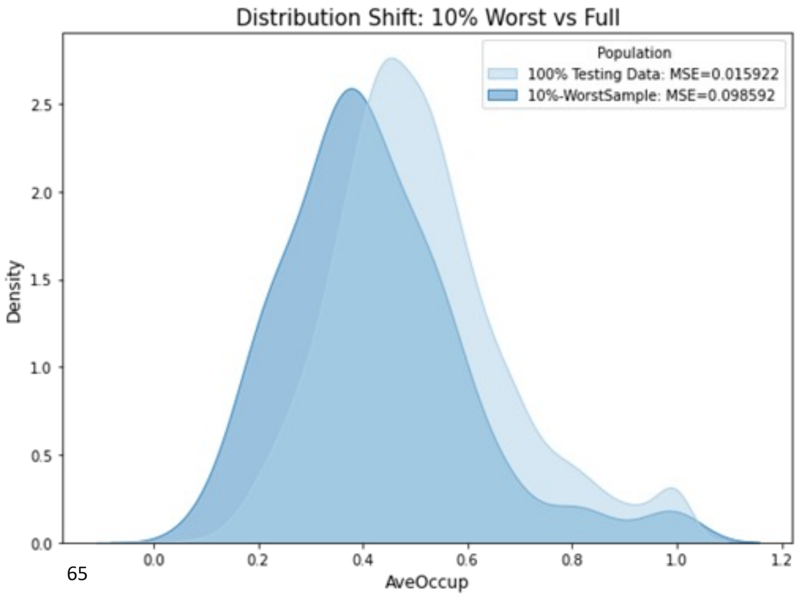




Quadratic pattern between X9 and the residuals

Generalizability:
Model performance on unseen data

Generalizability to out of distribution data



- Development, $P(.)$ vs. Production, $Q(.)$

$$P(x,y) \overset{?}{\Leftrightarrow} Q(x,y)$$

- Covariate drift

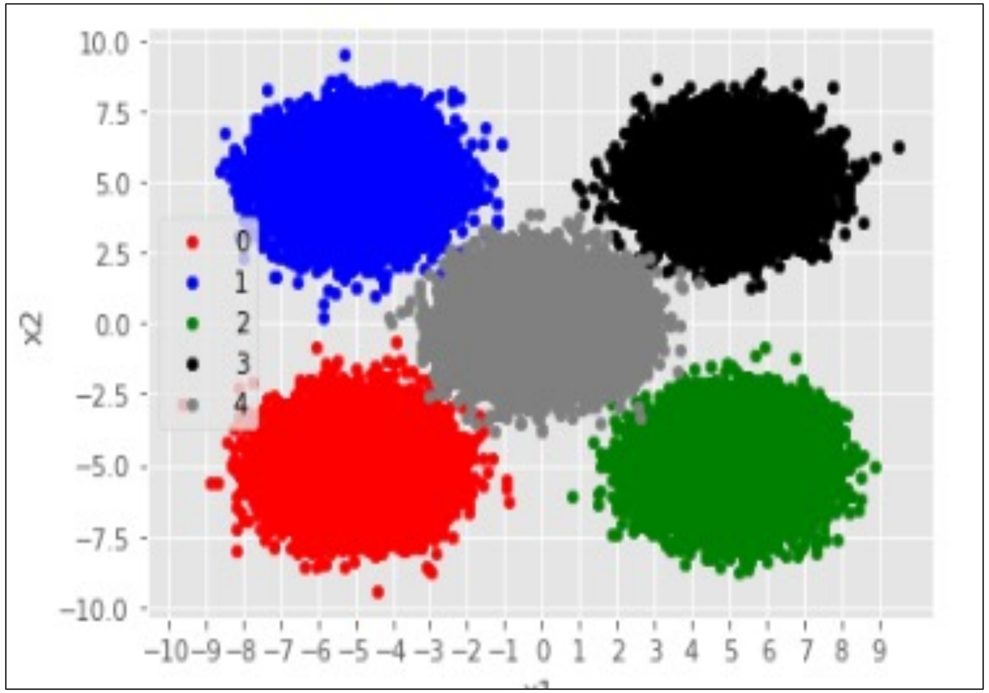
$$P(y|x) == Q(y|x)$$
$$\boldsymbol{P(x) \neq Q(x)}$$

- Concept drift

$$P(y|x) \neq Q(y|x)$$

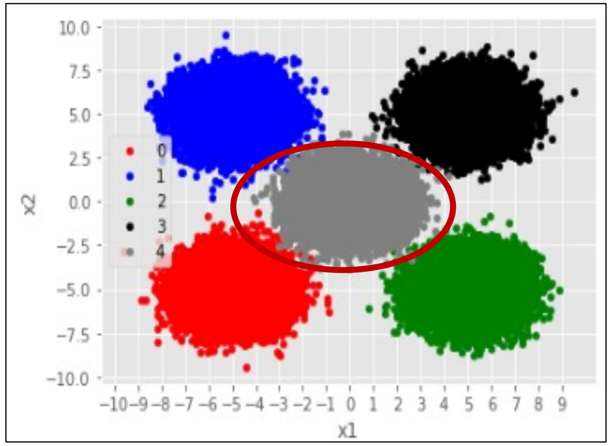
Performance with unseen data

- Extrapolation: Test data at inference time is totally outside of the training envelope for all/some features
  - Beyond the tails of (one/more than one) feature
  - Separate cluster



- Simulation study
  - $f(\boldsymbol{x}) = \beta_1 x_1 + \cdots + \beta_5 x_5 + \beta_6 x_1 x_2$
  - X1 and X2 simulated from clusters
  - X3, X4, X5 are from Gaussian distribution
  - $\beta_1 = 0.5, \beta_2 = 0.3, \beta_3 = -0.9, \beta_4 = 1.2, \beta_5 = -1, \beta_6 = 0.1$

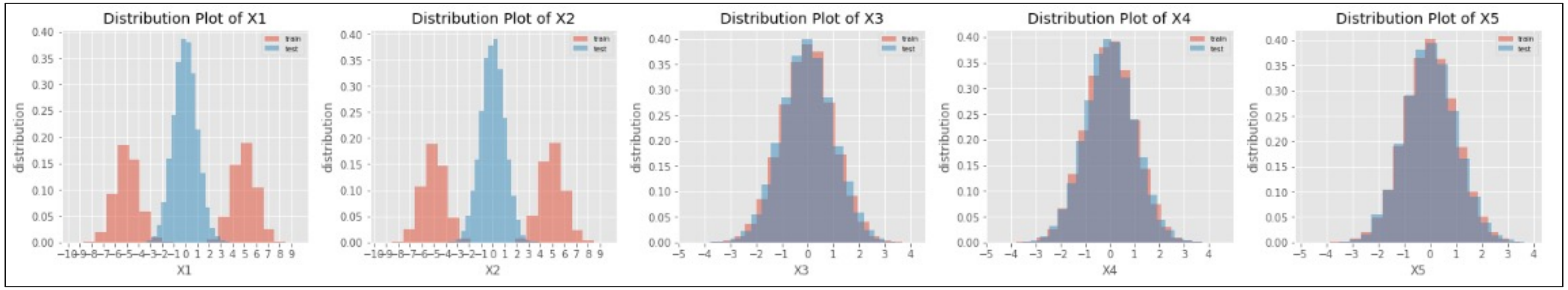
Performance of XGB and FFNN models for different **held out test clusters (1)**



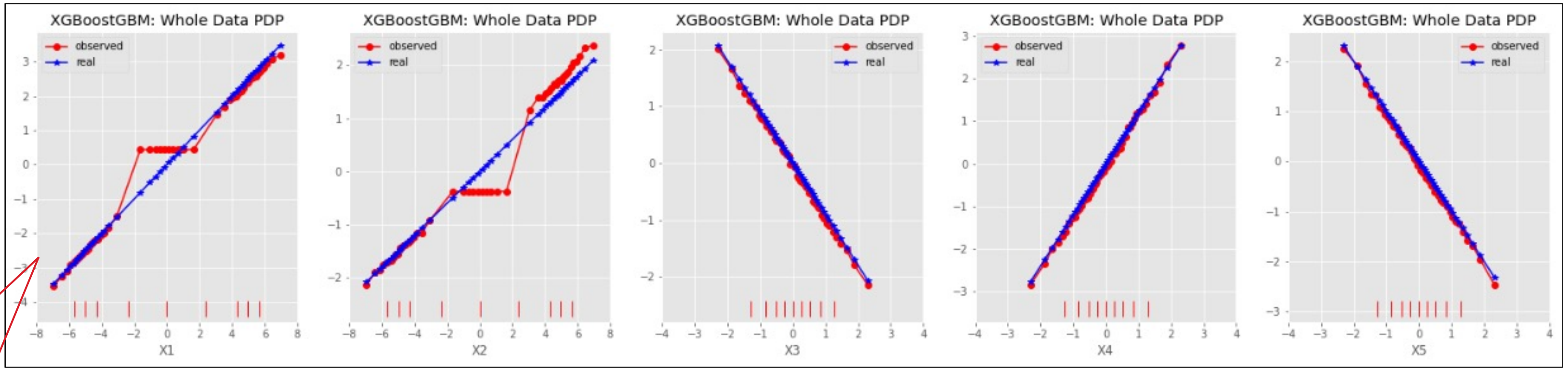
interpolation

	XGB	FFNN
Train MSE	0.967	1.003
Valid MSE	1.055	1.022
Test MSE	**1.503**	1.186

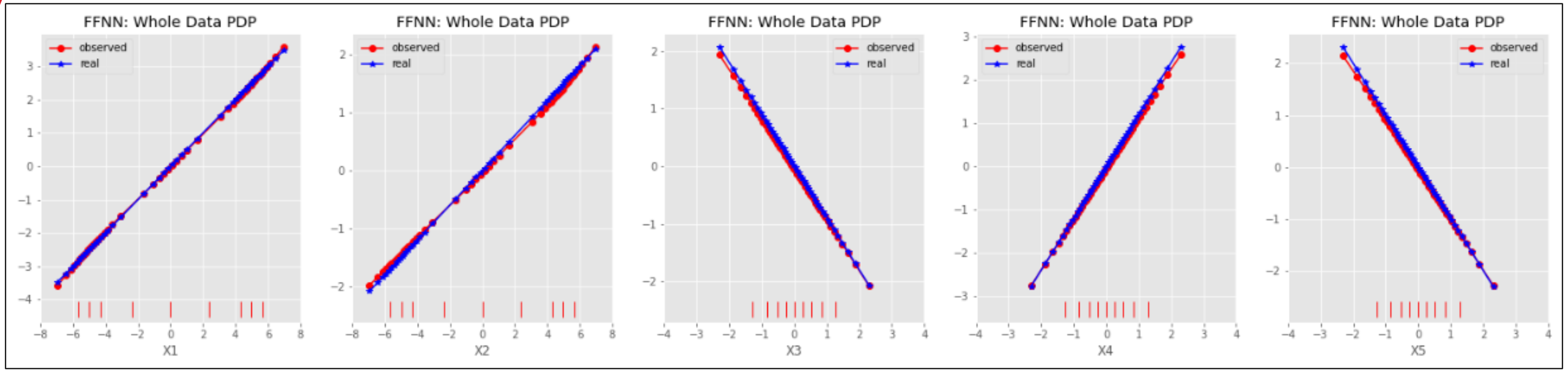
$$f(x) = 0.5x_1 + 0.3x_2 - 0.9x_3 + 1.2x_4 - x_5 + 0.1x_1x_2$$



• Observed True



XGB

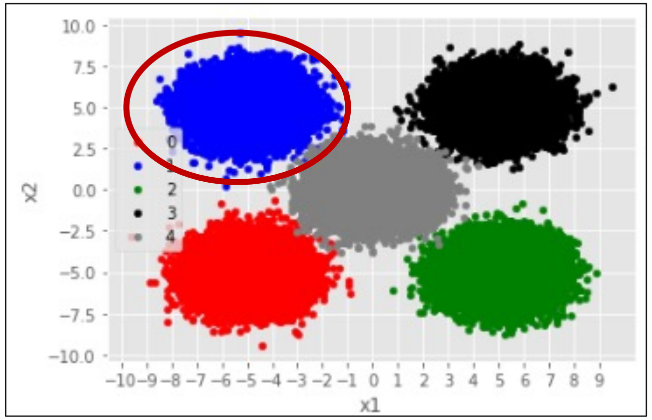


FFNN

- XGB due to its piecewise constant nature interpolates accordingly and results in larger MSE
- FFNN interpolates correctly since true underlying function is linear
- Slight under-estimation for other covariates in both algorithms

Performance of XGB and FFNN models for different **held out test clusters** (2)

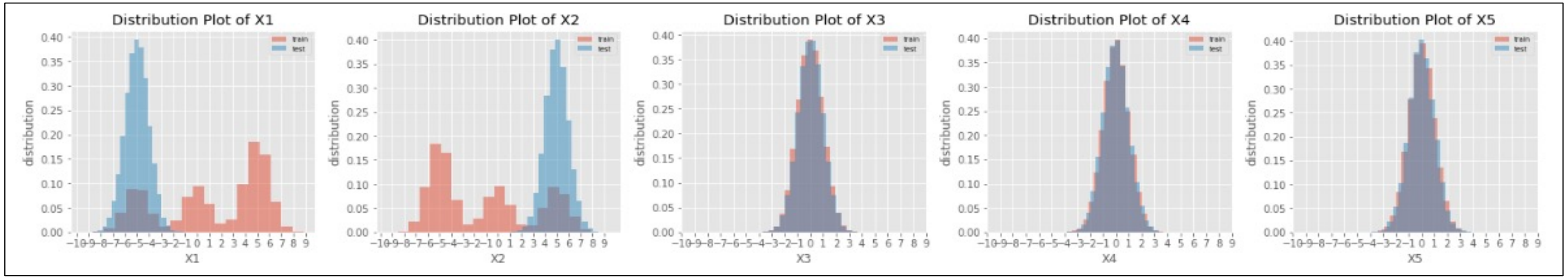
$$f(x) = 0.5x_1 + 0.3x_2 - 0.9x_3 + 1.2x_4 - x_5 + 0.1x_1x_2$$

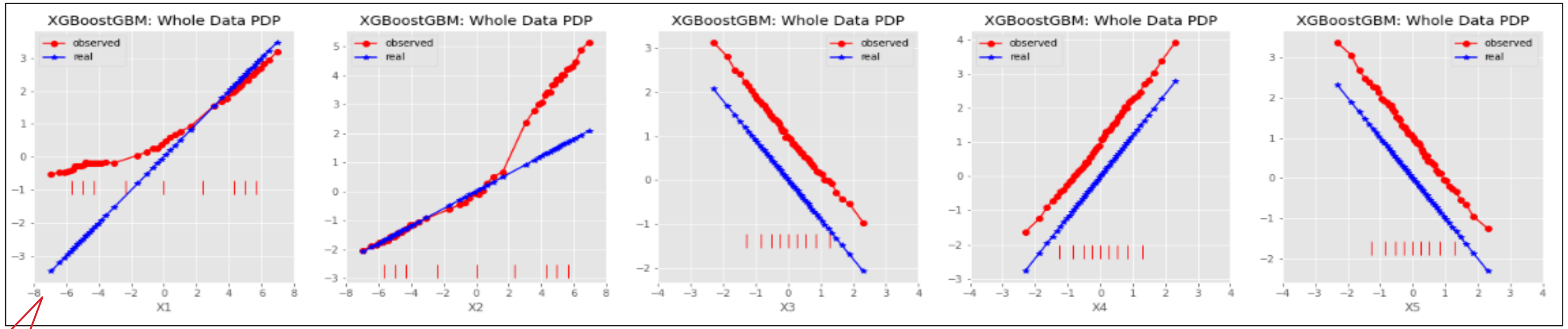


extrapolation

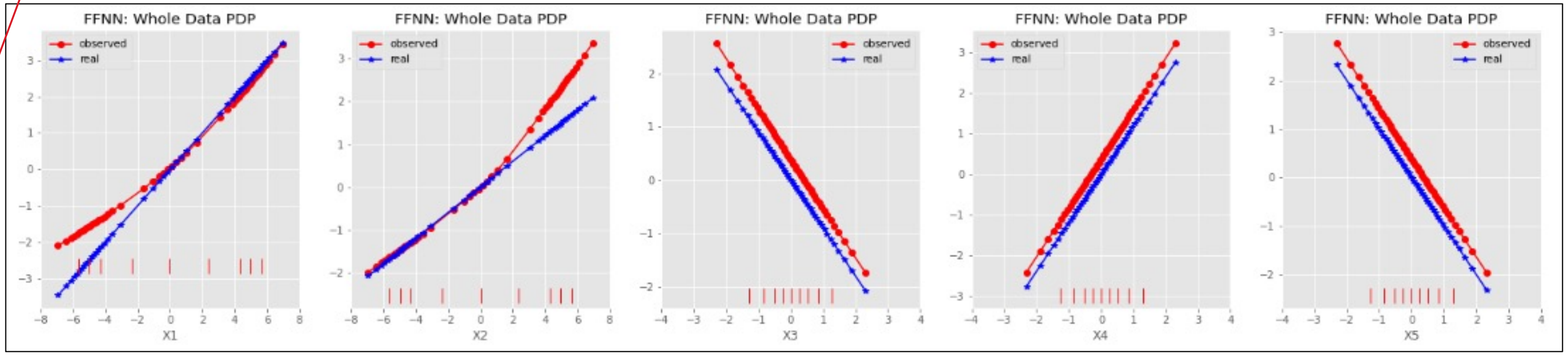
	XGB	FFNN
Train MSE	0.938	1.016
Valid MSE	1.047	1.007
Test MSE	**28.390**	**5.773**

- Over-estimation in all covariates
- XGB estimates a flat effect of X1 on the tail
- XGB performs worse than FFNN as the true functional form is linear





XGB



FFNN

- Observed
- True

Uncertainty quantification

Uncertainty in model predictions

- We should measure model performance beyond simply using accuracy measures such as (MSE, AUC, etc.
- Three major sources of uncertainty
  - Noise in the data: large noise in data causes uncertainty in predictions
  - Data sparsity: overall low observations or insufficient observations in certain regions of data
  - Mis-specified model: unaccounted for effects

- In linear regression, uncertainty is quantified with prediction intervals computed under the model assumptions
- In ML models. we want to construct such intervals without any distributional assumptions

Prediction interval

- Given training data $\{Z_i = (X_i, Y_i)\}_{i=1}^{n}$ and error level α, we want interval C$(X_{n+1}, Z_{1\ldots n})$ such that

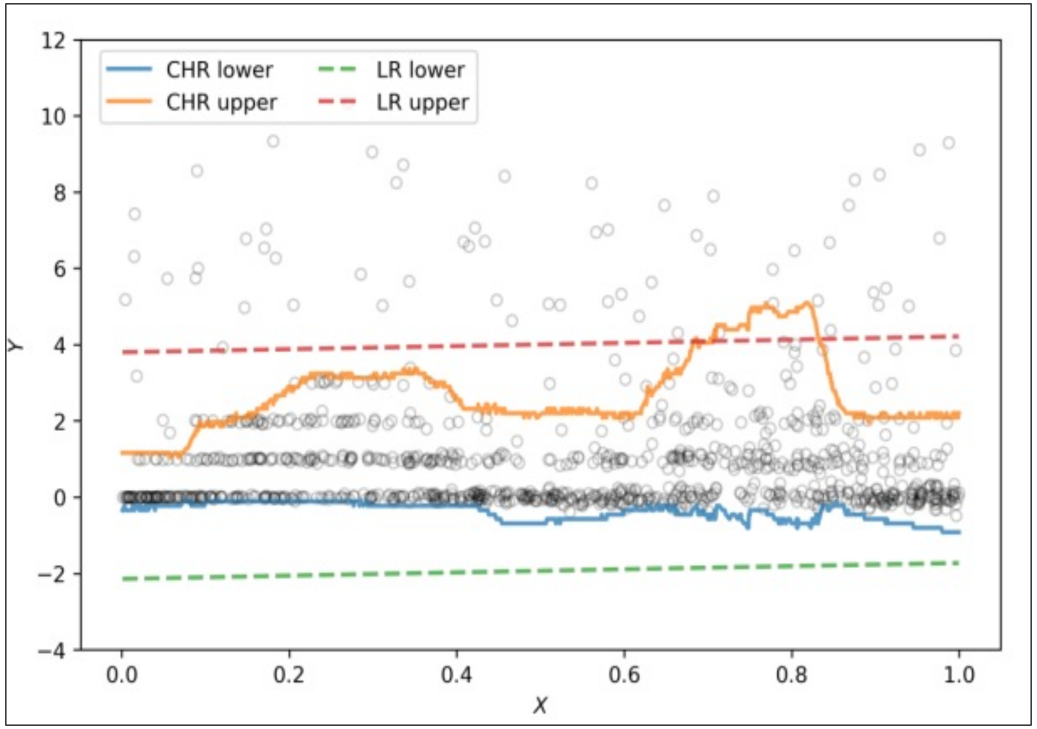
$$P\big(Y_{n+1} \in C(X_{n+1}, Z_{1\ldots n})\big) \geq 1 - \alpha$$

- Wider prediction interval → less reliable prediction

- Quantification of uncertainty can be done using Conformal Prediction to produce **distribution free prediction interval**

- Based on theory: given set of *exchangeable* real numbers $s_1, \ldots s_{n+1}$, where $s_i = g(z_i)$,

$$P(rank(s_{n+1}; s_1, \ldots s_n) \leq \lceil (n+1)(1-\alpha) \rceil) \geq 1 - \alpha$$



Conformal prediction

- Relies on a pre-defined conformal score function $S()$
- Given a trained model $\hat{f}$, compute conformal scores on a *hold out dataset* $S_i = S(x_i, y_i, \hat{f})$
- Get calibrated α quantile on scores S_i given as $\hat{q}_\alpha$
- Prediction interval for x_{test} given as $C(x_{test}) = \{y: S(x_{test}, y, \hat{f}) \le \hat{q}_\alpha\}$

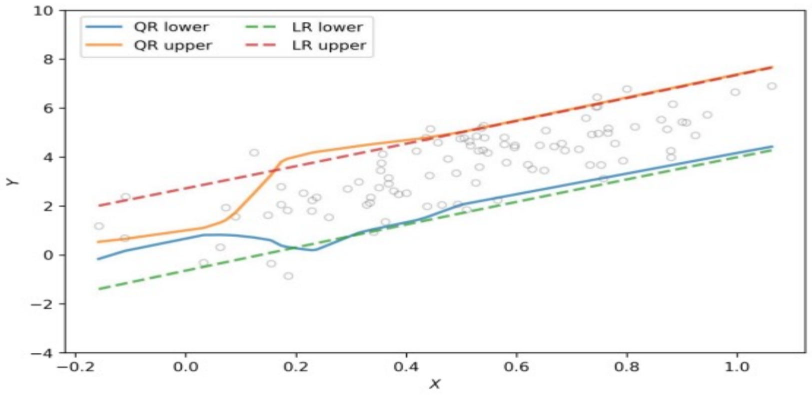
Given the exchangeability assumptions, it is guaranteed that

$$P(y_{test} \notin C(x_{test})) \le \alpha$$

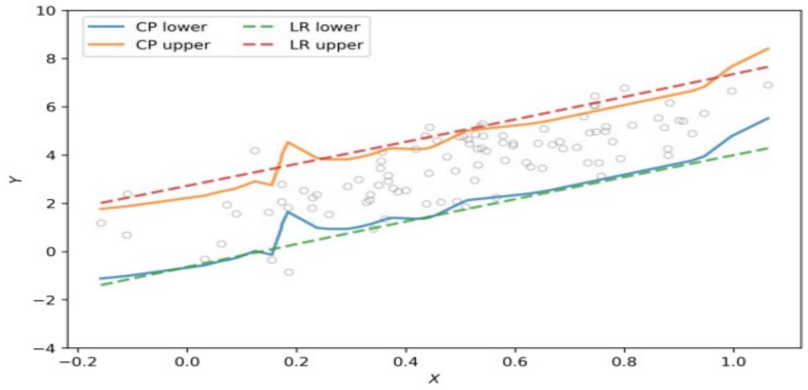
Conformal prediction

- Prediction using quantile regressions:
  - Fits two quantile regression on training data and computes interval based on these
  - no guaranteed coverage for finite samples
- Split conformal prediction(SCP):
  - $S(x, y) = \left|y - \hat{f}(x)\right|$
  - equal length, not adaptive intervals
- Conformal Quantile Regression(CQR):
  - $S(x, y) = \max(\hat{f}_{\frac{\alpha}{2}}(x) - y, y - \hat{f}_{1-\frac{\alpha}{2}}(x))$
  - adaptive and provides coverage guarantee

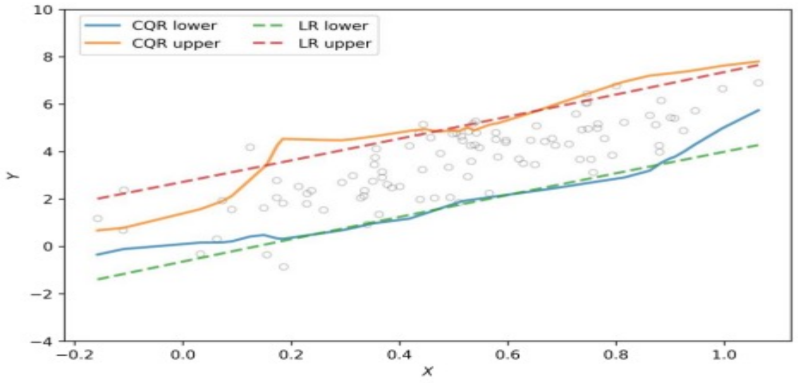
Prediction interval using quantile regression



Prediction interval SCP



Prediction interval CQR



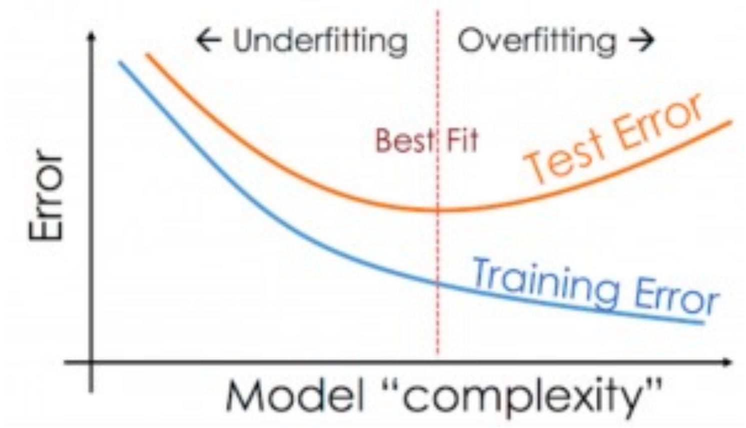
Robustness:
Assessing stability of results to small perturbations

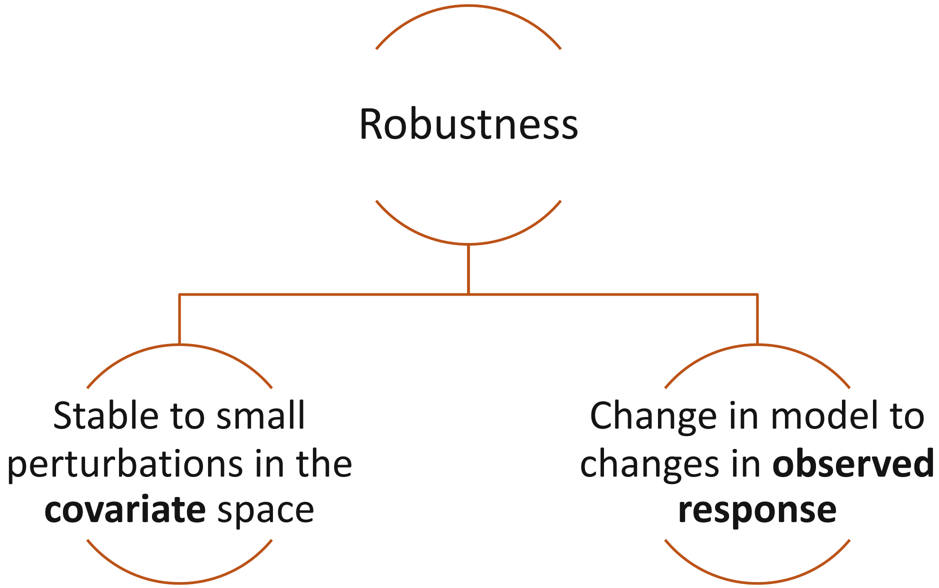
History of model robustness

- There has been a long focus on “data robustness” in statistical modeling and data analysis
  - Robustness to outliers, influential points, etc.
  - Rank based methods, robust techniques (trimmed mean, median, influence functions, etc.).
- Notion of “model robustness” in parametric models
  - Model misspecification
  - Degrees of freedom
- Renewed interest in ML literature
  - “Stability of machine learning algorithms” – W.Sun, et al.
  - “Stability and generalization of learning algorithms that converge to global optima” – Z.Charles, et al.
  - “Under-specification presents challenges for credibility in modern machine learning” - A. D’Amour, et al.

Concept of model robustness

- Over-fitted models have high generalizability error
- This is usually measured as the gap between training and testing error.
- As a consequence of over-fitting a model may show large variability in predictions due to
  - Small perturbations in the X-space
  - Small changes in the Y-space
- Goal of robustness study:
  - Quantify stability of model to small perturbations
  - Identify points/regions which contribute to lack of stability in model.
  - Understand whether the lack of robustness is due to sensitivity of model to different covariates or simply over-fitting to noise in these regions.

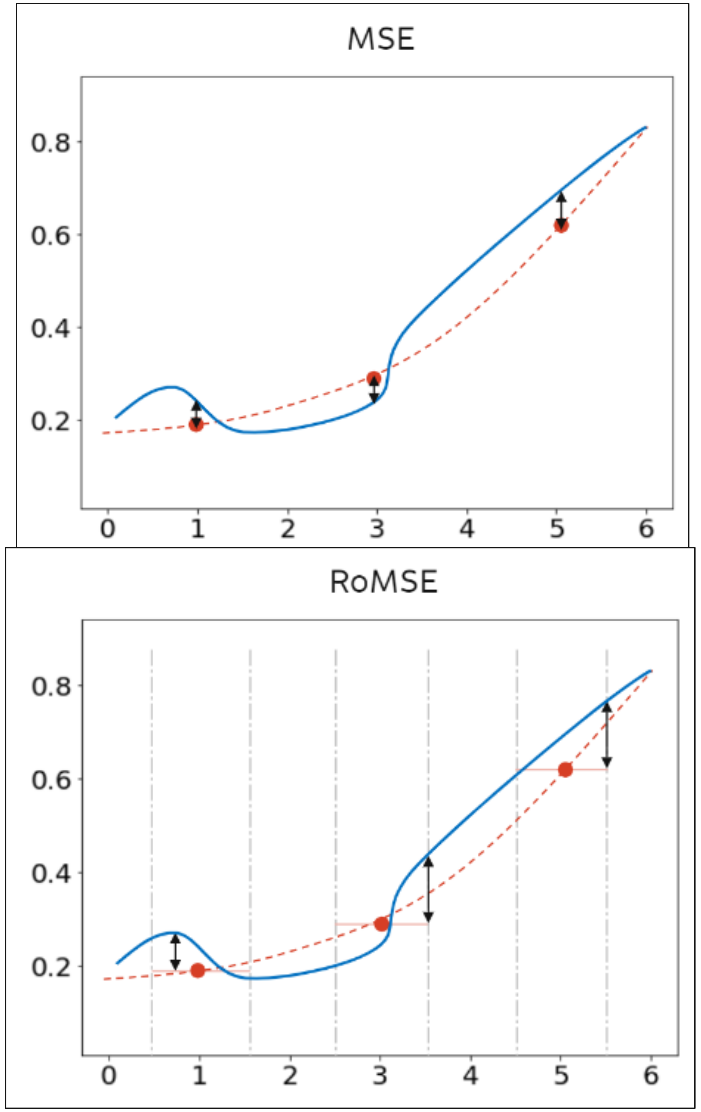




X space perturbations – worst case analysis

Key assumption: **Robust model is able to maintain stable outputs against small perturbations in input.**

- Worst case perturbation
  - Look at small neighborhood of given observation
  - Find worst case prediction on perturbing observation in that neighborhood
  - Define metrics based on the worst case scenario.
  - RoMSE = $\frac{1}{N}\Sigma_{i=1}^{N} \max_{d_i \in D_i}(f(x_i + d_i) - y_i)^2$
  - RoAUC = $AUC(\{(f_i^*)^{y_i}(f_i^{**})^{1-y_i}\}, \{y_i\}),$
    - $f_i^* = \min_{d_i \in D_i} f(x_i + d_i), f_i^{**} = \max_{d_i \in D_i} f(x_i + d_i)$
- Constrained optimization problem
  - White-box: assumes knowledge of model architecture and parameters
  - Black-box: only requires able to call model and generate prediction on perturbed inputs.
- True response is not known, so we compare to original response
- **Looking at worst case scenario exaggerates drop in performance**

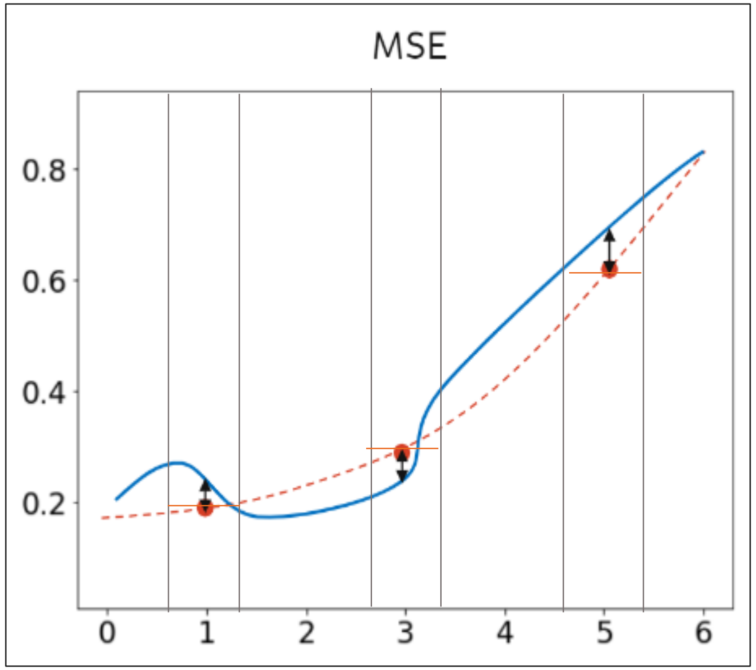


Average measures of robustness

Key assumption: **Robust model is able to maintain stable outputs against small perturbations in input.**

Multiple random perturbations

- Randomly sample within a small perturbation region
- Look at summary statistics of the deviation
  - $\hat{y}(X_i + \delta_{ij}) - \hat{y}(X_i)$: captures variability in prediction
  - $\hat{y}(X_i + \delta_{ij}) - y_i$: combination of true error and prediction variability
- Potential summaries include mean, median, quantile, maximum, standard deviation, IQR



$$\hat{y}_{ik} - \hat{y}_i$$

Deviance for each perturbation *k* of an observation *i*

$$rPPV_i = \sqrt{\frac{\Sigma_{k=1}^{K}(\hat{y}_{ik} - \hat{y}_i)^2}{K}}$$

Summarize the deviance observation *i:* root **P**erturbed **P**rediction **V**ariance

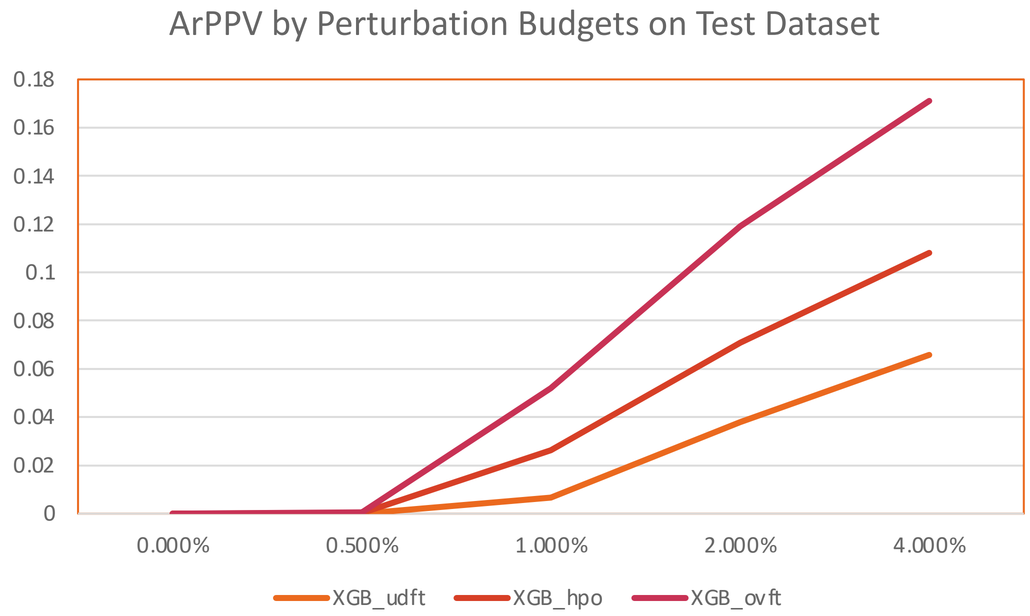
$$ArPPV = \frac{1}{N}\Sigma_{i=1}^{N} rPPV_i$$

Measure of robustness for given model and budget

- Alternatively, compute MSE on perturbed data and corresponding predictions and obtain average of perturbed MSE $\frac{1}{K}\Sigma_k \frac{1}{N}(\hat{y}_{ik} - y_i)^2$

Choice of neighborhood

- All tests of robustness rely on **local** perturbations
- Budgets control local neighborhood
  - Larger budgets → large neighborhood
  - Small budget → small neighborhood
- Budgets can be expressed as percentage $b\%$, where it refers to a percentage on range, IQR, standard deviation, etc.
  - Example: perturb observation by 2% standard deviation
- When perturbations are large, response should change. Hence comparing to original prediction/response only makes sense for small budgets.
- At larger budgets these metrics cannot assess model stability.



Perturbations strategies

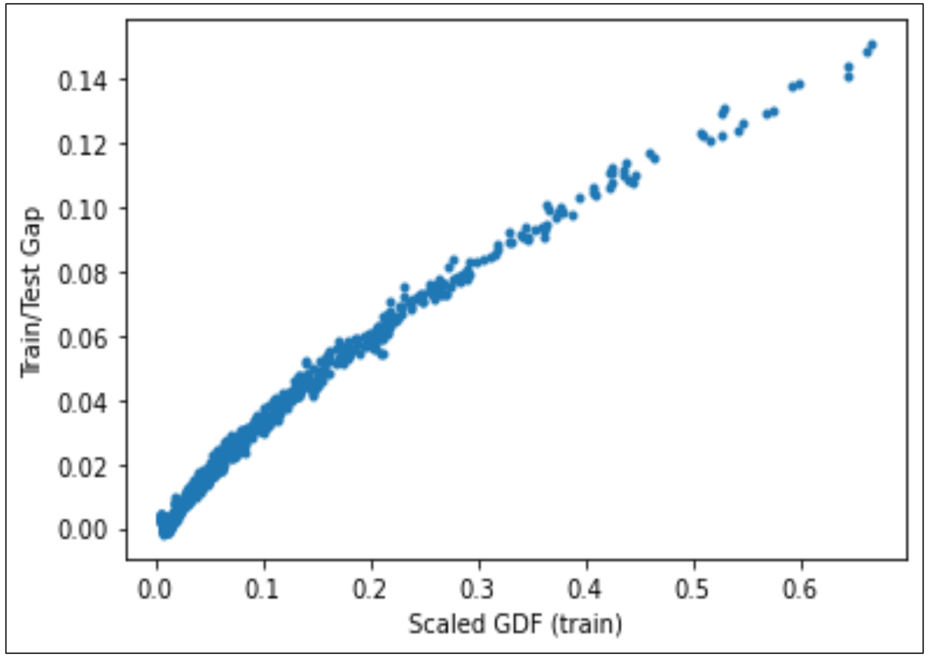
- **Continuous variables**
  - Model based perturbation
  - Add noise following Gaussian distribution (**budget is a percentage of std dev**)
    - **Correlated noise** respecting variable associations
    - **Independent noise**
  - Add noise from Uniform distribution(budget is percentage or range/IQR)

- **Discrete variables**
  - Perturb in original scale
    - **Round up** perturbed variables to **avoid invalid values**
  - Quantile based perturbation
    - Convert values to quantile scale
    - Perturb and round up to nearest value
    - Invert to original value
    - **Helps to perturb values for long tailed distribution**

- **Categorical variables**
  - **Marginal** perturbations (perturb each categorical variable independently)
    - With some probability change the category using user-defined **transition matrix** or transition matrix created from counts in the dataset
  - Joint perturbations respecting sense of local perturbations
    - Use some **pseudo distance method** to define distance between the categorical variables of two observations
    - Perturb to a combination that occurs in the local **neighborhood** based on the pseudo-distance measure.

Generalized degree of freedom (DF) as measure of robustness

- In linear models, model complexity is measured through the number of variables in the model (DF)
  - Linear model : $Y = X\beta + \epsilon$
  - $\hat{\beta} = (X'X)^1 X'Y$ and $\hat{\mu}(Y) = X(X'X)^1 X'Y$
  - **DF:** $\boldsymbol{p} = tr(H) = \sum_i h_{ii} = \sum_i \frac{\partial \hat{\mu}(y_i)}{\partial y_i}$, $H = X(X'X)^{-1}X'$
  - Therefore DF≡ sum of sensitivities of fitted values to observed values.

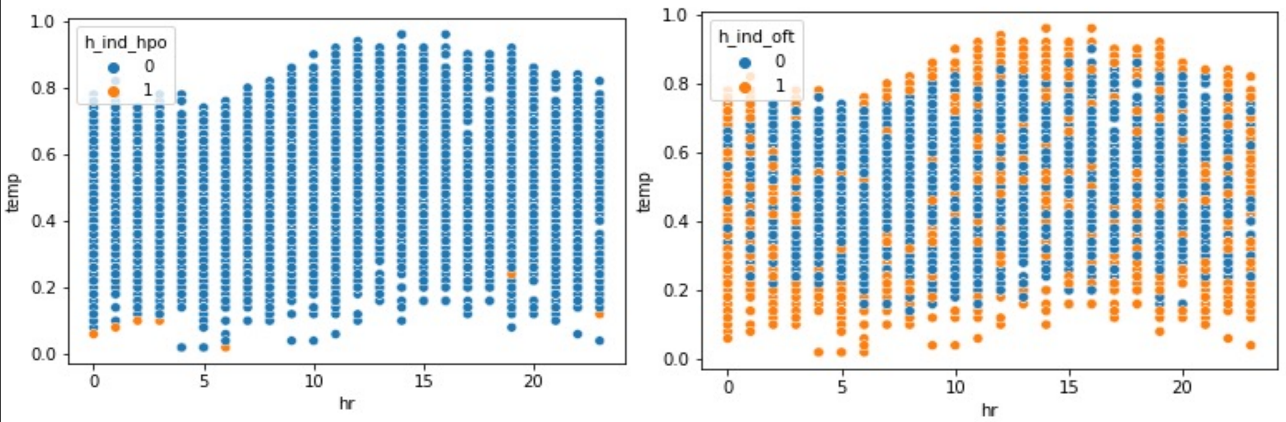


- Generalized Degrees of Freedom (GDF)(Ye 1998):
  - $D(M) = \sum_i h_i^M$ for model M, where
  - $h_i^M = \frac{\partial E_\mu[\hat{\mu}(Y_i)]}{\partial Y_i}$
- Compute DF by Monte Carlo method
  - For $t = 1, \dots, T$:
    - Generate perturbations $\Delta_t = (\delta_{t1}, \dots, \delta_{tn})$ from independent $N(0, \tau^2)$
    - Evaluate $\hat{\mu}(Y + \Delta_t)$ based on the modeling procedure M.
  - Calculate $\hat{h}_i^M$ as the regression slope from $\hat{\mu}(Y_i + \delta_{ti}) = \alpha + \hat{h}_i^M \delta_{ti}$, $t = 1, \dots, T$
  - Estimate $D(M)$ by $\widehat{D}(M) = \sum_i \hat{h}_i^M$
- High computational complexity due to multiple(T) refit of model based on the perturbations.
- Strong relationship with train and test gap

Illustration: Assessing robustness of two XGB models

- Illustration on DC Bike Share data
- Two models fitted to bike share data:
  - tuned XGB model (hpo)
  - overfitted XGB model (oft)
- Compute leverages on each point

Temp vs hr plots colored by high leverage points



Hpo

Oft

Orange: points with hstat> 0.8

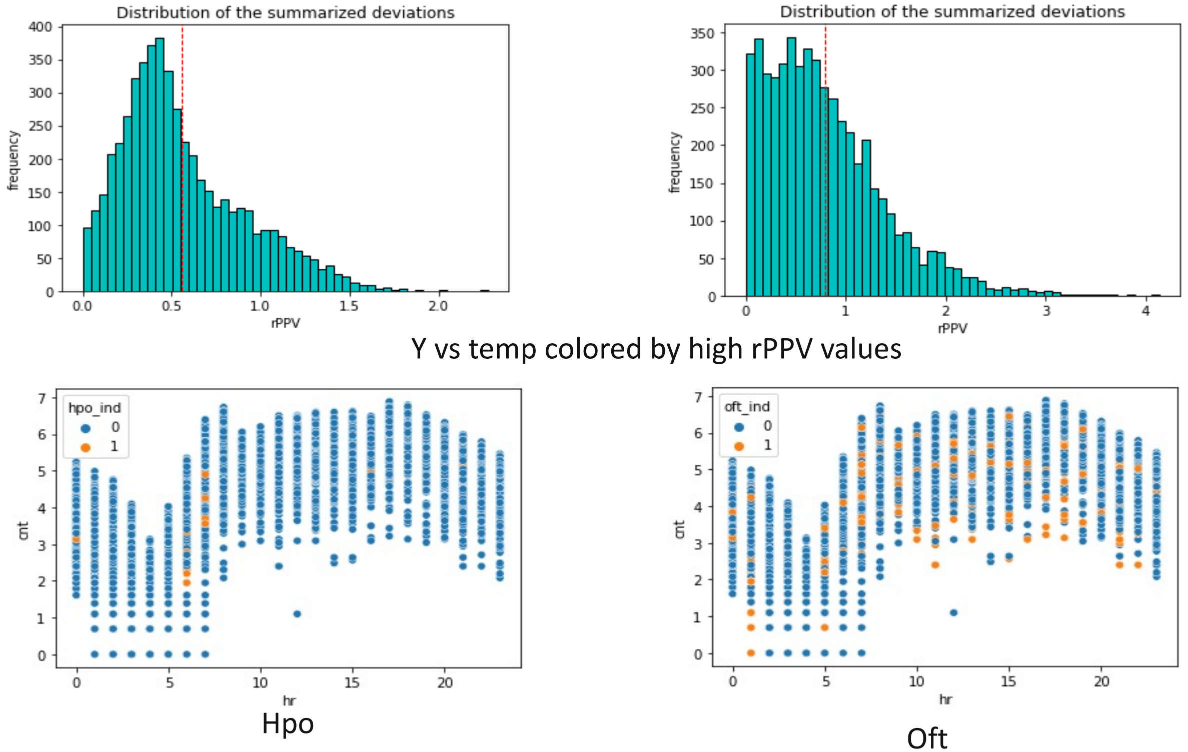
- Over-fitted (OFT) model has many high leverage points indicating that it is a more complex model
- High leverage occur mostly in the outer regions of the data envelope

Model	Train mse	Test mse	Mse gap	ArPPV	GDF
Hpo	0.117	0.186	0.069	0.56	1674.07
Oft	0.066	0.204	0.138	0.79	3551.70

Assessing robustness of two XGB models (contd.)

- Illustration on [DC Bike Share data]
- Two models fitted to bike share data:
  - tuned XGB model (hpo)
  - overfitted XGB model (oft)
- Examine robustness when one pertur variable temperature with budget 2 % std dev.

Histogram of rPPV metrics



Orange: points with rPPV > 1.5

- Over-fitted model (**oft**) has a thick and long tail on the summary measures of instability to perturbations (rPPV)compared to tuned model

- In tuned model (**hpo**), high rPPV points are located at regions where the counts sharply change with hour. Thus the high values can be attributed to sensitivity

- In **oft** model, high rPPV points are present during nearly all hours of the day showing a general lack of stability.

Model	Train mse	Test mse	Mse gap	ArPPV	GDF
Hpo	0.117	0.186	0.069	0.56	1674.07
Oft	0.066	0.204	0.138	0.79	3551.70

Summary

- We have looked at multiple causes for model weakness
  - Not capturing true effect
  - Missing interactions
  - Over-fitting
  - Data outside known envelope
- We have shown multiple methods to capture
  - Model weakness due to missed out true effects
  - Lack of uncertainty in model predictions
  - Lack of stability in fitted response
- Ongoing research area