WELLS FARGO

Machine Learning Model Validation

Risk Americas Workshop
New York, NY

Agus Sudjianto and Vijay Nair
Corporate Model Risk, Wells Fargo

May 9, 2022

Agenda

- **9:00 – 9:30: Introduction** – Agus Sudjianto
- **9:30-10:45: Machine Learning and Explainability** – Vijay Nair and Sri Krishnamurthy
- 10:45-11:00: **Break**
- **10:45-11:45: Unwrapping ReLU Networks** – Agus Sudjianto
- **11:45-12:45 Inherently Interpretable Models** – Vijay Nair and Sri Krishnamurthy
- **12:45-1:15: Lunch Break**

- **1:15-2:15: Outcome Testing** – Agus Sudjianto
- **2:15-3:15 Hands-on Exercises** – Sri Krishnamurthy
- **3:15-3:30: Break**
- **3:30-4:30 Bias and Fairness** – Nick Schimdt
- **4:30-5:00: ModelOp Presentation** – Jim Olsen

Overview

1. Introduction: Risk Dynamics, Conceptual Soundness and Outcome Testing
2. **Supervised Machine Learning: Algorithms and Explainability**
3. Deep ReLU Networks and Inherent Interpretation
4. Inherently Interpretable Models
5. Outcome Testing

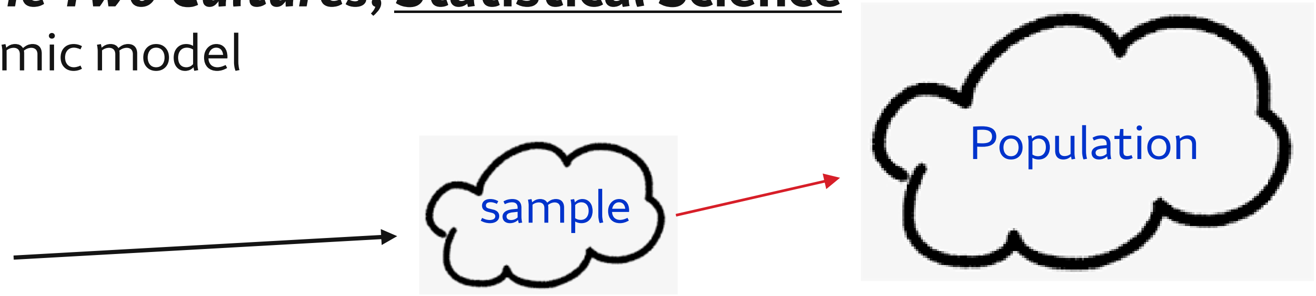
2. Supervised Machine Learning: Algorithms and Explainability

- Supervised ML algorithms
  - Ensemble algorithms
  - Feedforward neural networks (FFNNs)
  - Application
- Post hoc explainability techniques
  - Global
  - Local
  - Challenges with post hoc techniques
- PiML Demo

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



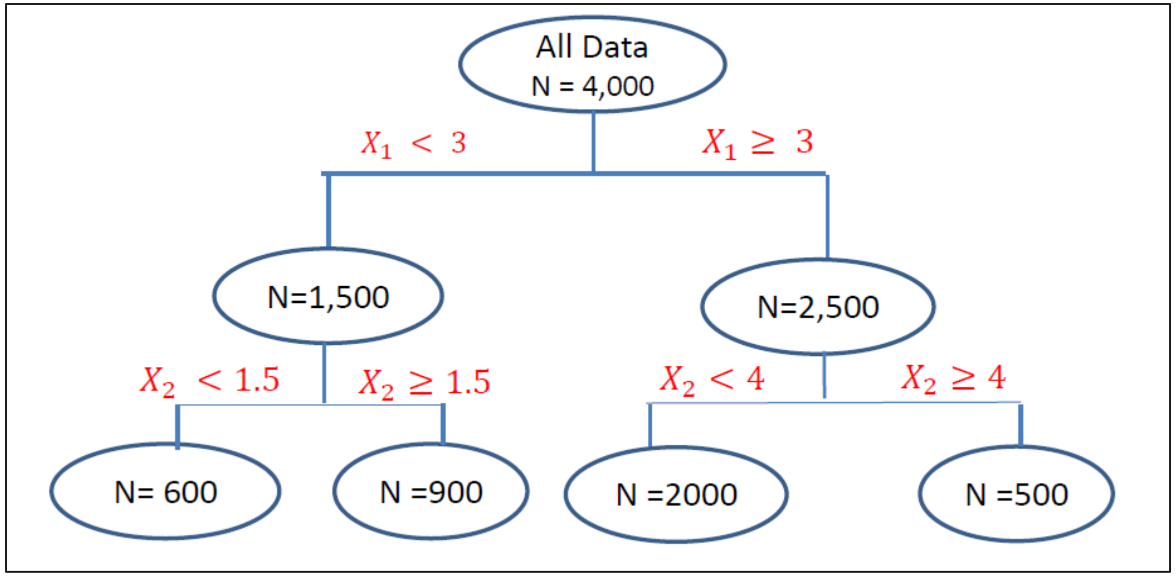
    - ○ **Estimate** model **parameters** and **assess uncertainty**
    - ○ Identify **key drivers** and **input-output relationships**
    - ○ **Extensive tools and diagnostics** developed over time
    - ○ Parametric **models** → **easier to interpret**

- 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

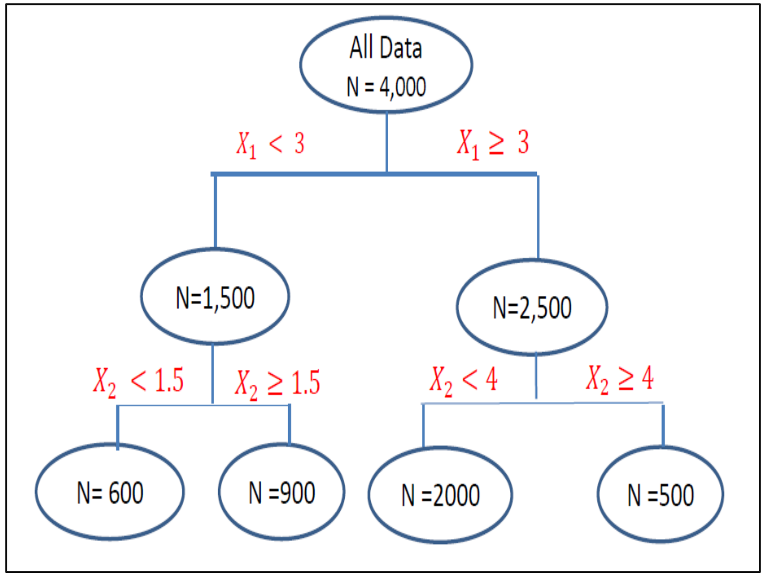
- **Ensemble algorithms**
  - Random Forests (RFs)
  - Gradient Boosting Machines (GBMs)
  - eXtreme Boosting (XGBoost)
    - Tree-based models
    - Piecewise constant within nodes



- Feedforward Neural Networks

Fitting trees with piecewise constant model

1) Root node → entire dataset with 4,000 observations. Compute overall fit metrics
   a) For continuous response: MSE
   b) For binary: Gini index or cross-entropy loss
2) Take each predictor in turn. Take a set of quantiles and consider splitting at that quantile: for example $X_1 < 3$ *or* ≥ 3. Compute fit metrics for the split.
3) Compare all the possible splits, take the one with maximum reduction in fit metric and split at that point.
4) Consider the resulting split nodes. Repeat steps 2 and 3.
5) Stop when some criteria is met: maximum reduction in fit metric, number of nodes, tree depth, and minimum number of observations in leaf node



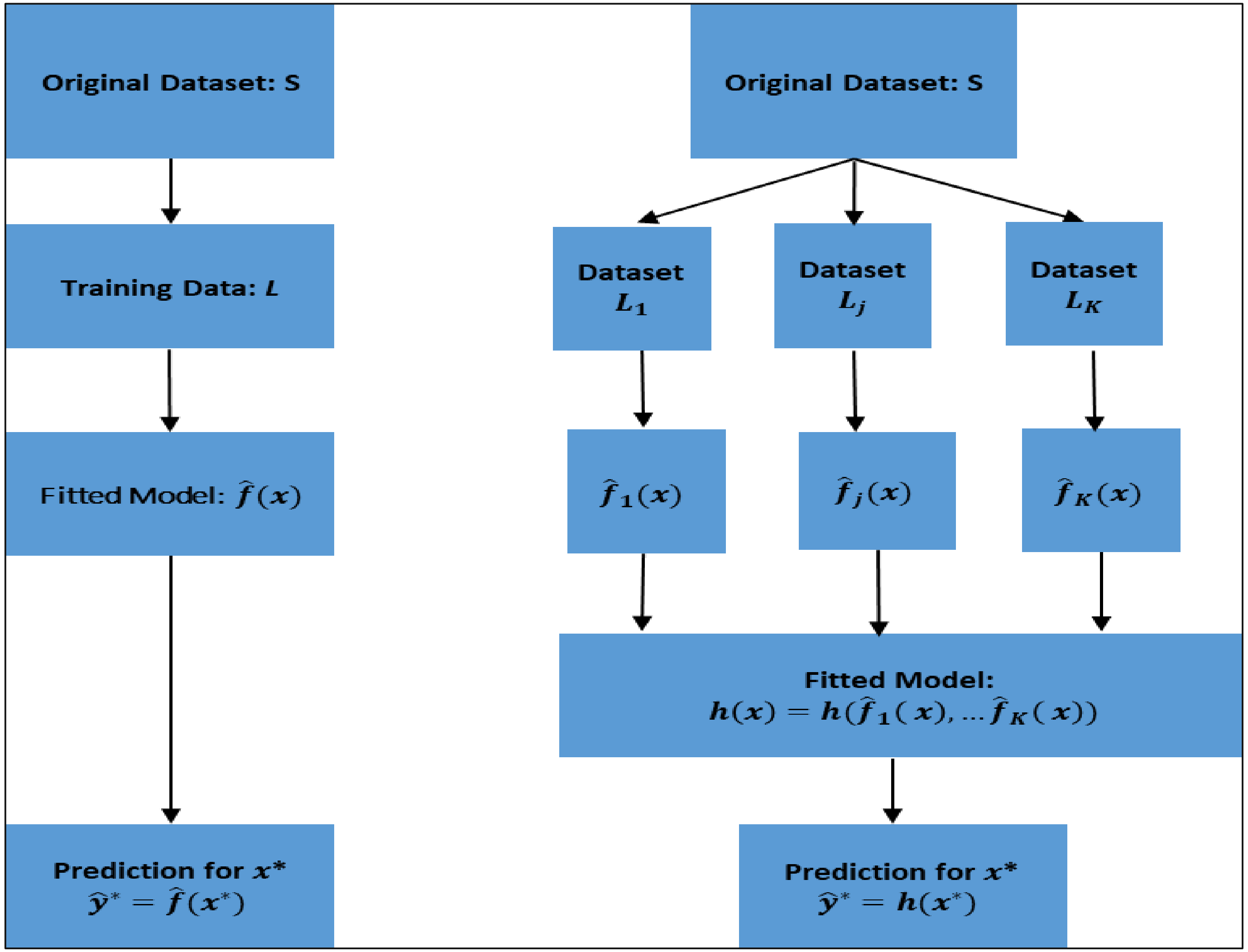
Result can be viewed as a piecewise constant regression model:

$$\hat{Y} = c_1 \ I\{X_1 < 3, X_2 < 1.5\} + c_2 \ I\{X_1 < 3, X_2 \geq 1.5\} + c_3 \ I\{X_1 \geq 3, X_2 < 4\} + c_4 \ I\{X_1 \geq 3, X_2 \geq 4\}$$

Most software packages use binary splits.

There are algorithms with more than two splits (CHAID by Kass, 1980).

Ensemble Algorithms



Original Dataset: S

Training Data: L

Fitted Model: $\hat{f}(x)$

Prediction for x^*
$\hat{y}^* = \hat{f}(x^*)$

Original Dataset: S

Dataset L_1

Dataset L_j

Dataset L_K

$\hat{f}_1(x)$

$\hat{f}_j(x)$

$\hat{f}_K(x)$

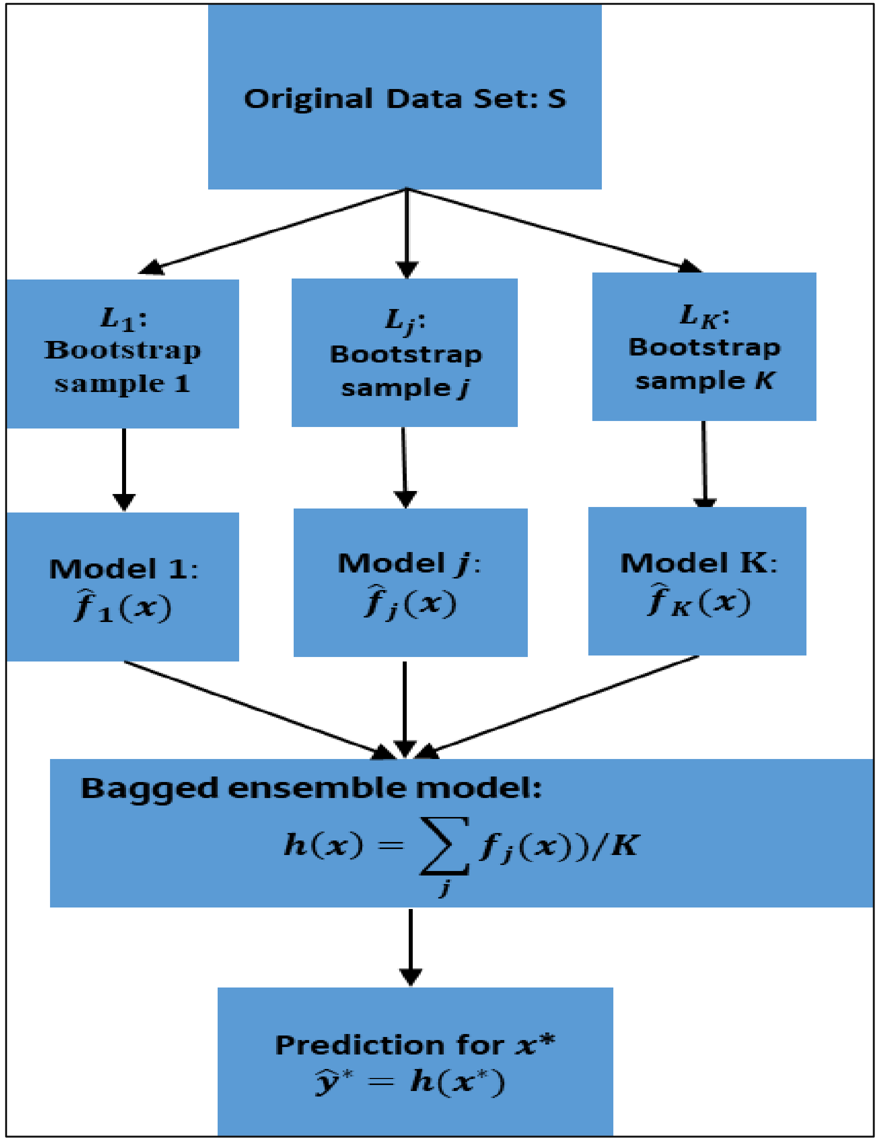
Fitted Model:
$h(x) = h(\hat{f}_1(x), \ldots \hat{f}_K(x))$

Prediction for x^*
$\hat{y}^* = h(x^*)$

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



Original Data Set: S

L_1: Bootstrap sample 1

L_j: Bootstrap sample j

L_K: Bootstrap sample K

Model 1: $\hat{f}_1(x)$

Model j: $\hat{f}_j(x)$

Model K: $\hat{f}_K(x)$

Bagged ensemble model:

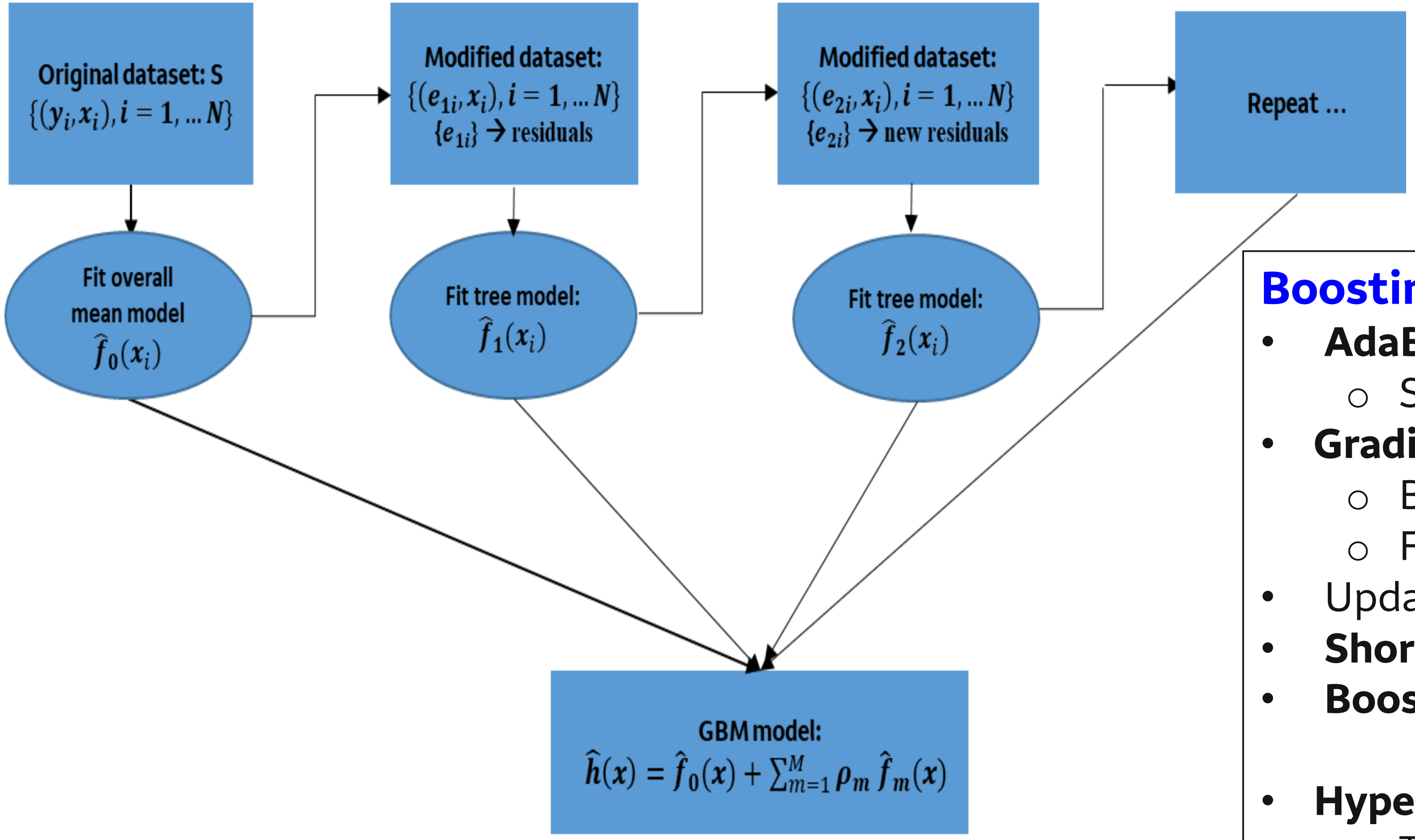
$$h(x) = \sum_j f_j(x))/K$$

Prediction for x^*

$$\hat{y}^* = h(x^*)$$

- **Random Forest** (Breiman and Cutler, 1994)
  - ○ 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



Original dataset: S
$\{(y_i, x_i), i = 1, \dots N\}$

Fit overall mean model
$\hat{f}_0(x_i)$

Modified dataset:
$\{(e_{1i}, x_i), i = 1, \dots N\}$
$\{e_{1i}\}$ → residuals

Fit tree model:
$\hat{f}_1(x_i)$

Modified dataset:
$\{(e_{2i}, x_i), i = 1, \dots N\}$
$\{e_{2i}\}$ → new residuals

Fit tree model:
$\hat{f}_2(x_i)$

Repeat ...

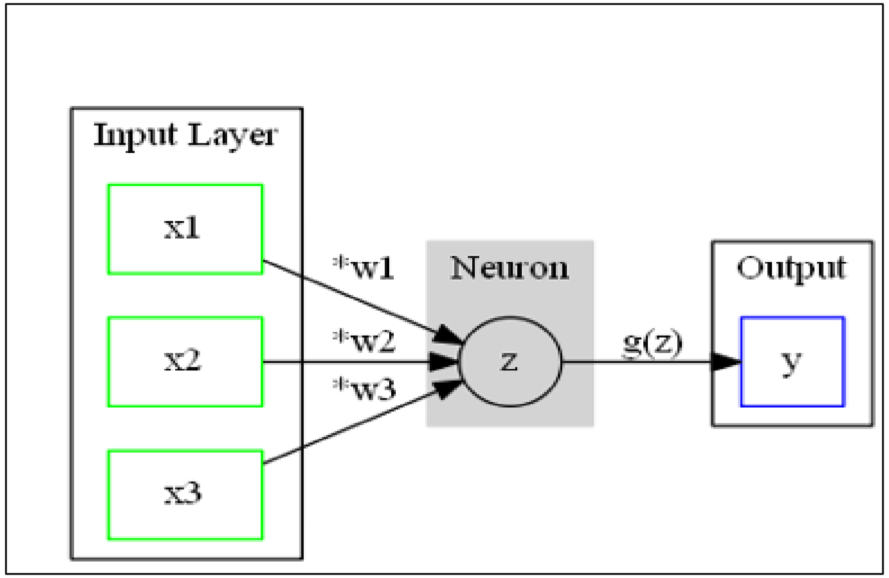
GBM model:
$\hat{h}(x) = \hat{f}_0(x) + \sum_{m=1}^{M} \rho_m \hat{f}_m(x)$

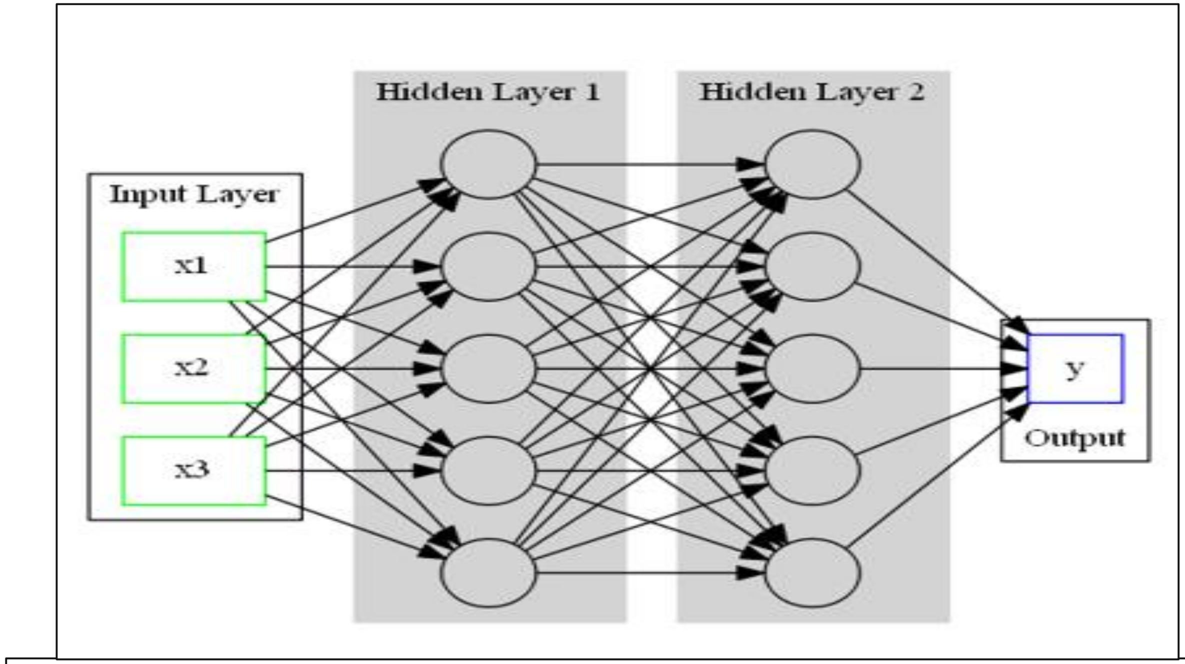
Boosting

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

- **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-parameters for FFNNs

Algorithms	HP	Description
FFNN Hidden layers have batch normalization and dropouts added.	Learning rate (lr)	• Learning rate in the Keras ADAM optimizer
	Layers	• Number of layers
	Neurons	• Number of neurons in each layer
	L1	• L_1 penalty value
	L2	• L_2 penalty value
	Dropout	• A form of regularization added to the hidden layers
	Batch size	• Number of samples per gradient update.

Hyper-Parameter Optimization

Batch vs sequential

Batch: Grid search, random search, designed experiments

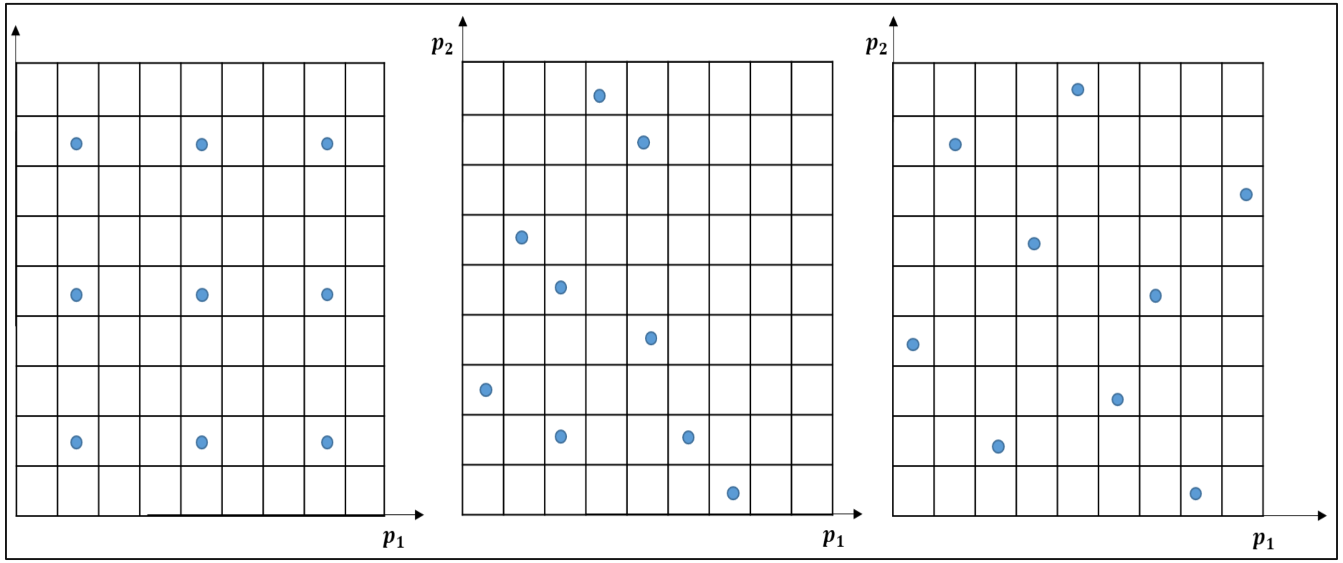
Sequential:

○ Bayesian optimization with Gaussian process (BOGP)

○ Tree-structured Parzen estimator (TPE)

Mixed: Hyperband

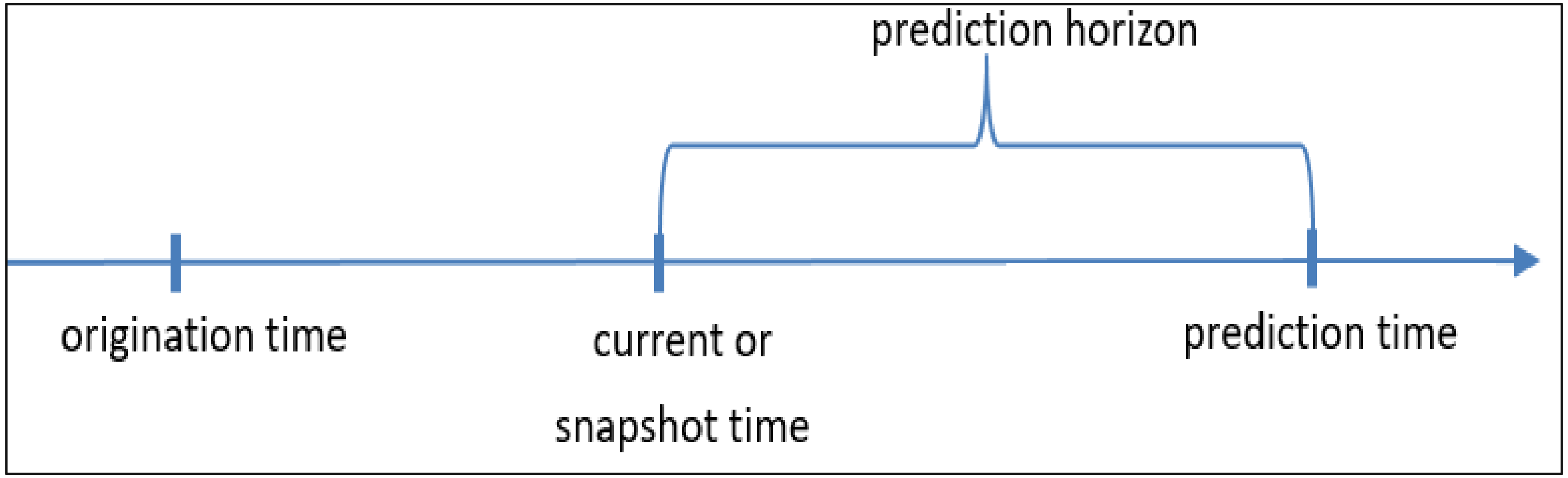
Visual comparison of: grid search (left); random search (middle); and Latin hypercube (right)



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, delinquency *loan payments up-to-date), etc.
  - Time varying predictors: at origination and over time

Modeling framework



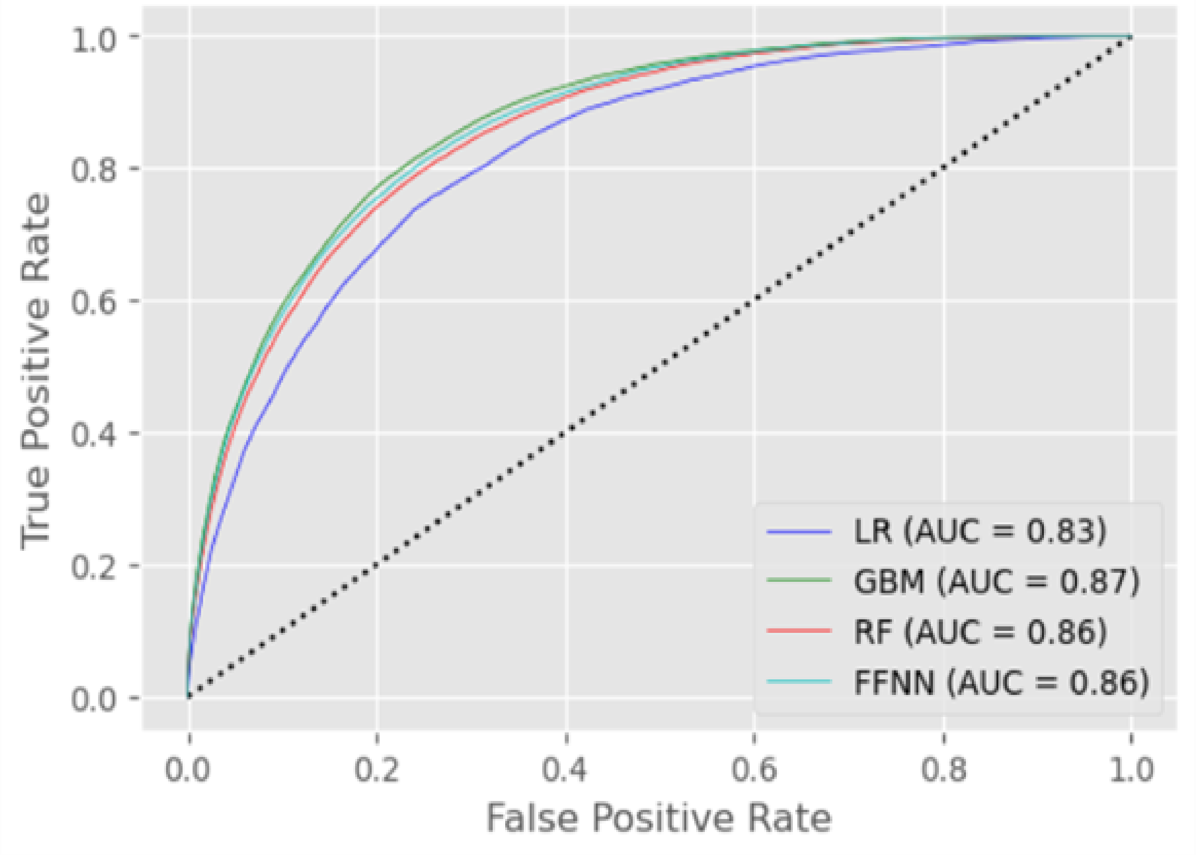
Loan origination, current (snapshot) and prediction times

Home lending data: Response binary (loan is in “trouble”)

Variable	Definition
horizon	prediction horizon (difference between prediction time and snapshot time) in quarters
snap_fico	credit score at snapshot (current) time
orig_fico	credit score at loan origination
snap_ltv	loan to value (ltv) ratio at current time
fcast_ltv	loan to value (ltv) ratio forecasted at prediction time
orig_ltv	origination loan to value ratio
orig_cltv	origination combined ltv
snap_current_ind	“current” (payments are up-to-date) indicator: 1= loan is current; 0 means loan is delinquent,
snap_early_delq_ind	“early delinquency” (loan is delinquent for a few months but not close to default)
pred_loan_age	age of loan (in months) at prediction time
snap_gross_bal	gross loan balance at snapshot time
orig_loan_amt	total loan amount at origination time
pred_spread	spread (difference between note rate and market mortgage rate) at prediction time
orig_spread	spread at origination time
orig_arm_ind	Indicator: 1 if loan is adjustable rate mortgage (ARM); 0 otherwise
pred_mod_ind	modification indicator: 1 means prediction time before 2007Q2 (financial crisis); 0 if after
pred_unemp_rate	unemployment rate at prediction time
pred_hpi	house price index (hpi) at prediction time
orig_hpi	hpi at origination time
pred_home_sales	home sales data at prediction time
pred_rgdp	real GDP at prediction time
pred_totpersinc_yy	total personal income growth (from year before prediction to prediction time)

Comparison of Predictive Performance

ROCs and AUCs on Test Data



- ML algorithms 22 predictors
- LR model: eight carefully selected variables

How typical is this “lift” in our applications?

ML: Post hoc explainability

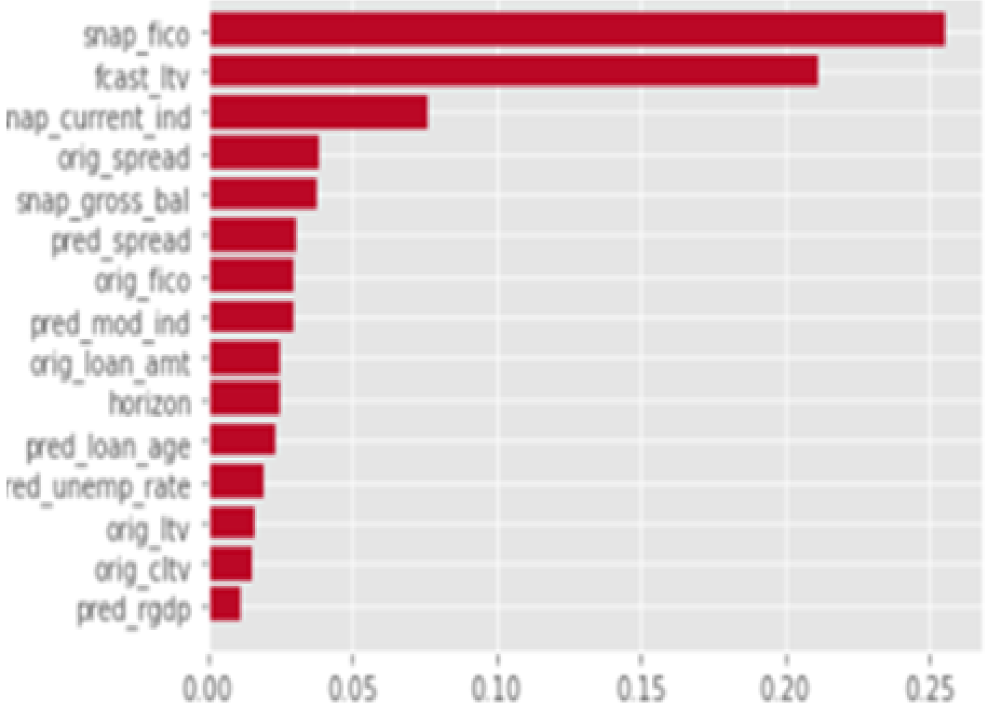
- What is post hoc explainability?
  - Apply interpretation tools after model development
  - Many of these tools are model agnostic
- Global Explainability
  - Examining relative importance of variables → Variable Importance Analysis
  - Understanding input-output relationships: Low-dimensional explanations
    - One- and Two-Dimensional Partial Dependence Plots
    - One- and Two-Dimensional Accumulated Local Effects Plots
    - H-statistics for Interactions
- Local Explainability: Two different local explanations
- PiML Demo

Global: Variable Importance

- **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**

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



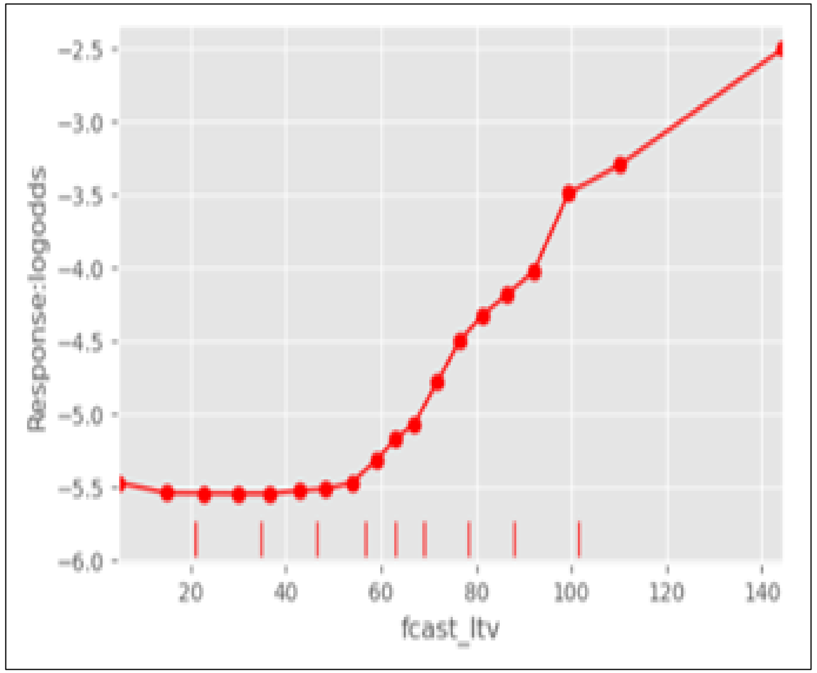
Input-Output Relationships: 1-D Partial Dependence Plots

- 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

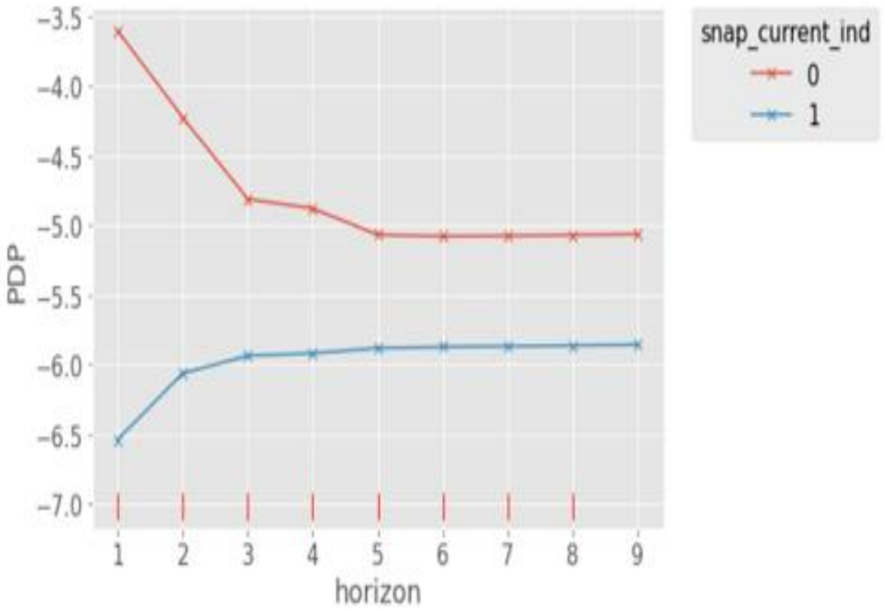
$$g_j(x_j = c) = \frac{1}{N}\sum_{i=1}^{N} \hat{f}(x_j = c, x_{-j,i})$$

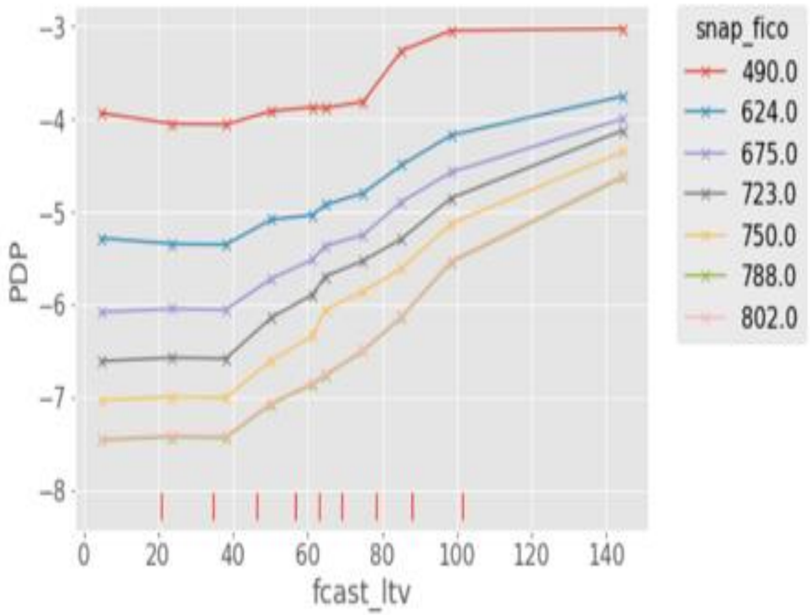
  - Plot $g_j(x_j)$ against x_j over a grid of values
  - One-dimensional summary
  - Interpretation: Effect of x_j averaged over other variables

Home Mortgage
1-D PDP for forecast_LTV



Home Mortgage: Selected 2-D PDPs





- One way to display 2D-PDPs

  → we shall see a different way using heat maps later)

- Multiple 1D-PDPs for the variable on the x-axis
- Multiple curves represent fixed values of the second variable
- Non-parallel curves show interactions

Accumulated Local Effect (ALE) Plots

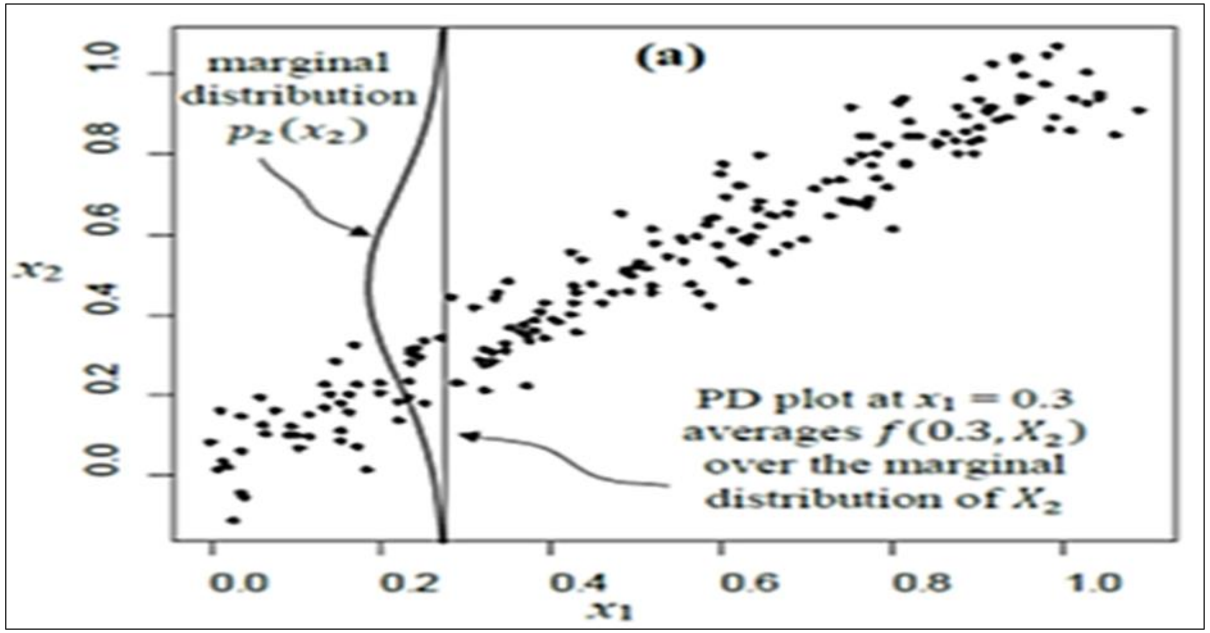
- PDPs can be misleading in the presence of high correlation

  → extrapolation outside data envelope

- ALE plots were developed as alternative to PDPs for highly correlated situations
  - Apley and Zhu (20200
  - Based on conditional expectations
    → avoids extrapolation
  - Reduces to PDP when predictors are independent

- We will see examples of ALE plots later

Figure from Apley and Zhu (2020)



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

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 $\hat{f}(\boldsymbol{x}^*) = \hat{f}(x_1^*, \ldots, x_K^*)$, what are the contributions of the different variables $\{x_1, \ldots x_K\}$ to the prediction?

- 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.

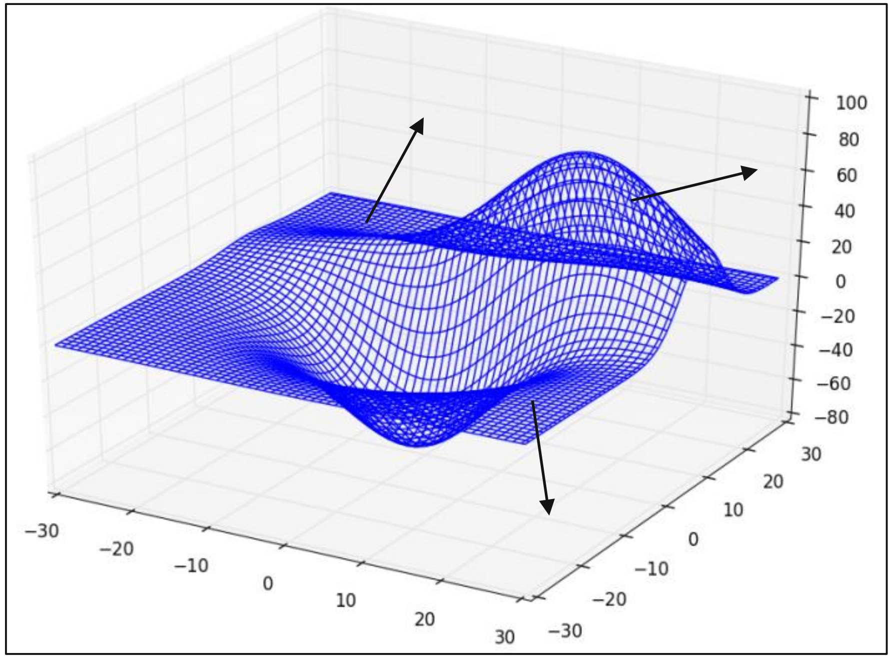
- Answer to 1: Model is linear → magnitudes and signs of regression coefficients provide explanation

- Answer to 2: Contribution of $\boldsymbol{x}_j^*$ is $\boldsymbol{b}_j \boldsymbol{x}_j^*$

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

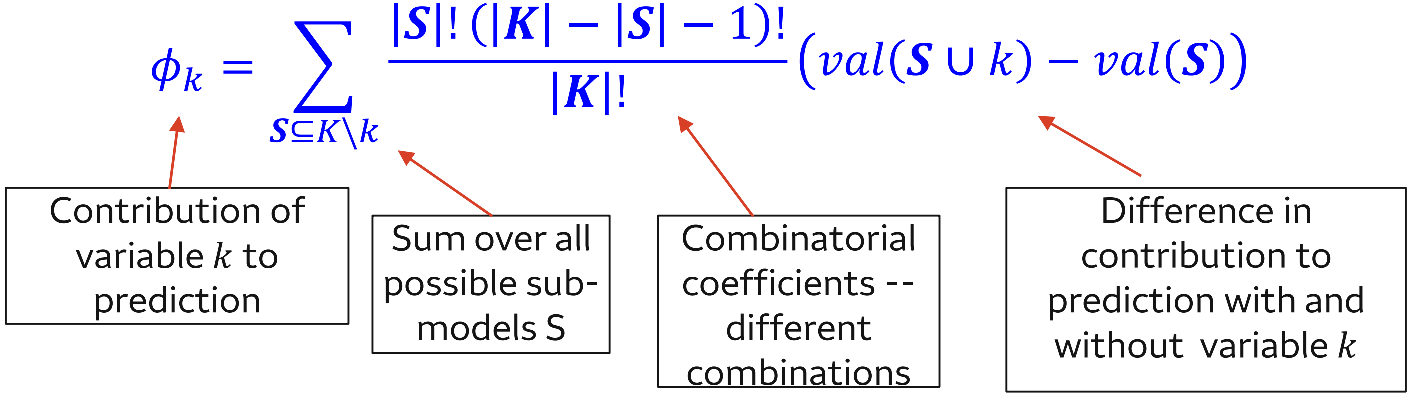
Local Explainability Q1: Use local linear model

- **Local Interpretable Model-Agnostic Explanations (LIME)** by Ribeiro et al.(2016).
- Can be used for complex situations like explaining classification boundary, etc.
- In this application, idea is simple:
  - At each point of interest x^*, fit a local linear model
  - Use the linear model $b_0 + b_1\, x_1^* + \ \ldots + b_K\, x_K^*$ for explanation
- Proposed approach for fitting local linear model:
  - Given point of interest x^*
  - Simulate points $\mathrm{z}_1, \ldots z_m$ in a neighbourhood of x^*;
  - Compute $\hat{f}(\mathrm{z}_1)\ \ldots \hat{f}(\mathrm{z}_m)$;
  - Fit a weighted multiple linear model to the points $\{z_k, \hat{f}(\mathrm{z}_k), \mathrm{k} = 1, \ldots \mathrm{m}\}$ with weights inversely proportional to the distance between z_k and x^*;
  - Use the fitted local linear model $b_0 + b_1\, x_1^* + \ \ldots + b_K\, x_K^*$ for interpretation



Local Explainability Q2: SHAP

- **Shapley** decomposition proposed by Shapley (1953)
- Good properties: efficiency, symmetry, additivity, etc.
- **Adaptation to answer Q2**
  - SHapley Additive exPlanations or **SHAP** values
  - Lundberg and Lee (2016)
  - Computationally challenging
  - Approximations proposed
    - KernelSHAP and TreeSHAP
    - Based on unrealistic assumptions
    - Differing results and often not reliable



$$\phi_k = \sum_{\boldsymbol{S} \subseteq K \setminus k} \frac{|\boldsymbol{S}|!\,(|\boldsymbol{K}| - |\boldsymbol{S}| - 1)!}{|\boldsymbol{K}|!} \left(val(\boldsymbol{S} \cup k) - val(\boldsymbol{S}) \right)$$

Contribution of variable k to prediction

Sum over all possible sub-models S

Combinatorial coefficients -- different combinations

Difference in contribution to prediction with and without variable k

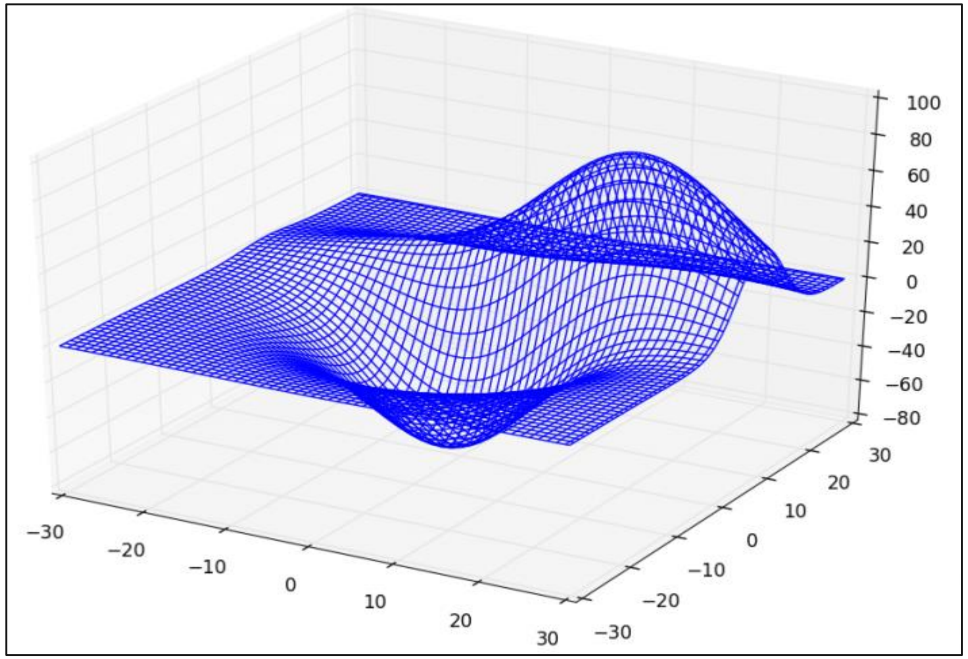
Example: $\hat{f}(x_1, x_2)$ → explanation at point $\boldsymbol{x}^*$

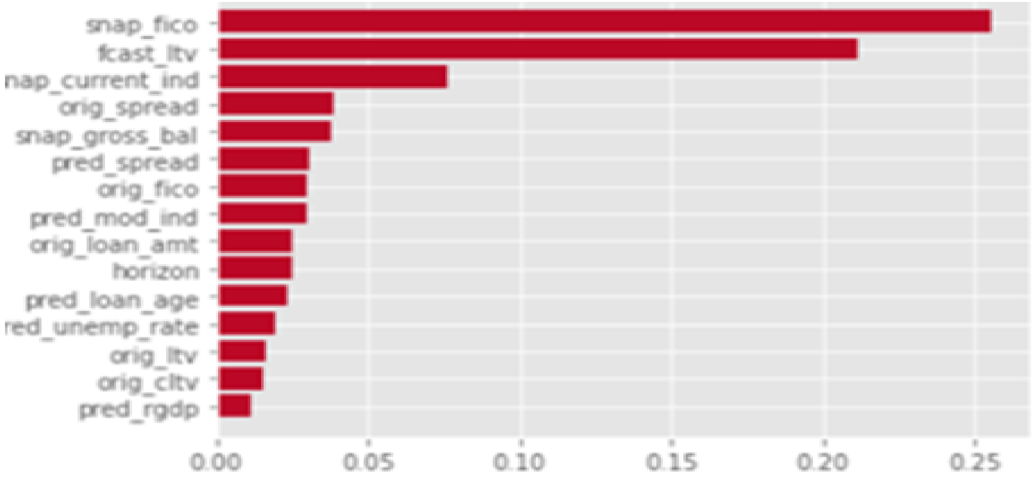
$\phi_1 = [\hat{f}(x_1^*, x_2^*)$ - $\hat{f}(x_{10}, x_2^*]) + [\hat{f}(x_1^*, x_{20})\ \hat{f}(x_{10}, x_{20})]$

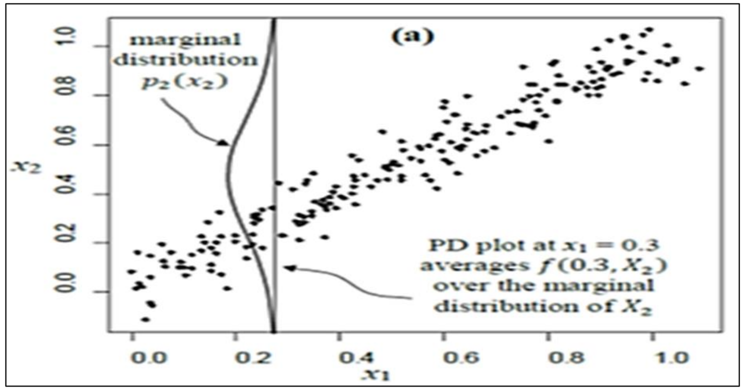
x_{10} and x_{20} are selected reference points – usually means

Challenges with post hoc explainability

- Low dimensional summaries of complex models
- Don’t tell the full story → mask structure
- ML algorithms with similar predictive performance can have different low-dimensional explanations
- Permutation based variable importance include both main effects and interactions
- Extrapolation with correlated predictors misleading answers if fitted model is unreliable outside the data envelope







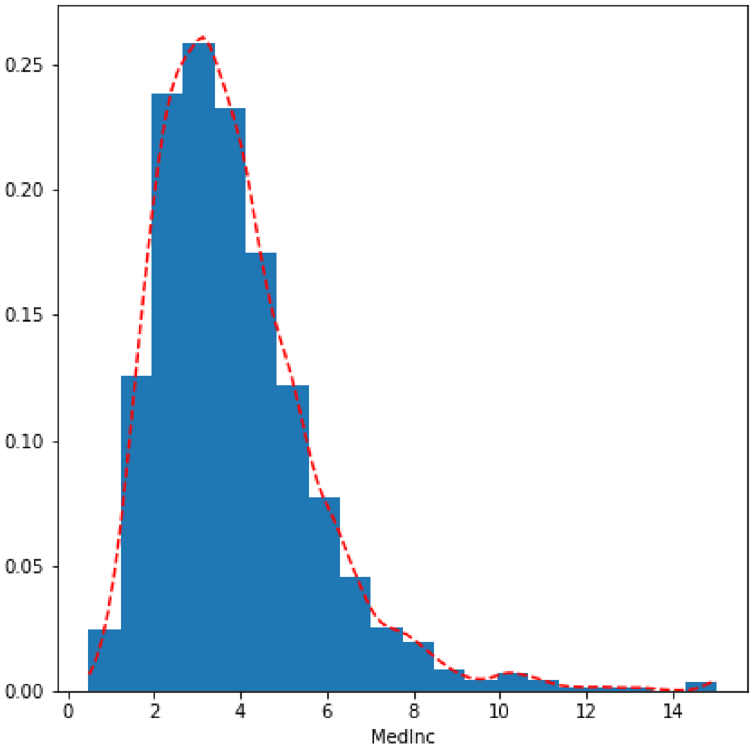
California Housing Data

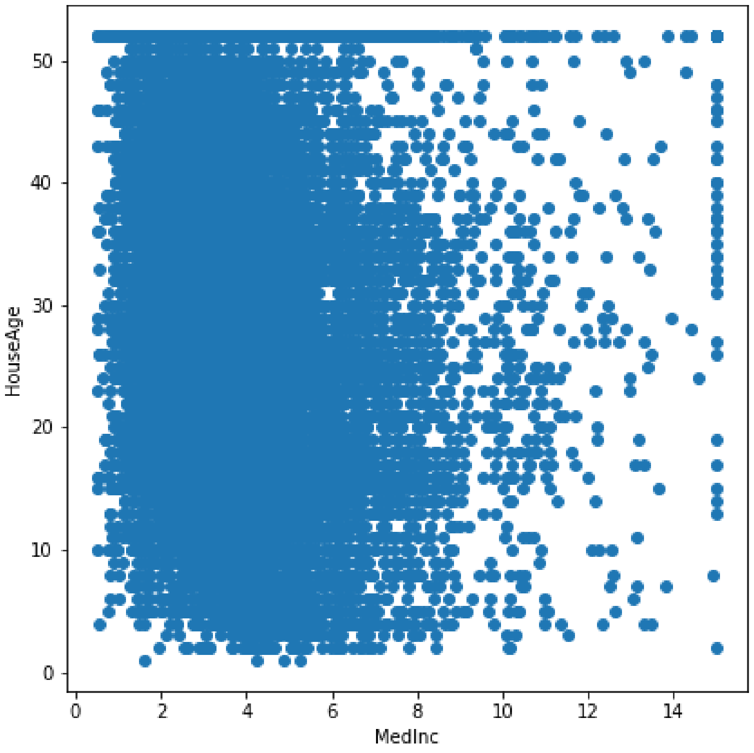
- Source:
  - Pace, R. K. and Barry, R. Sparse Spatial Autoregressions, Statistics and Probability Letters, (1997) 291-297
  - Dataset can be obtained from the StatLib repository
  - Available in PiML
- Description
  - Data from 1990 Census for California
  - Data for housing of "block groups" – geographically compact area
  - Geographical area of block varies inversely with population density
  - Total of 20,640 observations: response (house price) and 8 predictors
  - Goal: Model house price as a function of predictors

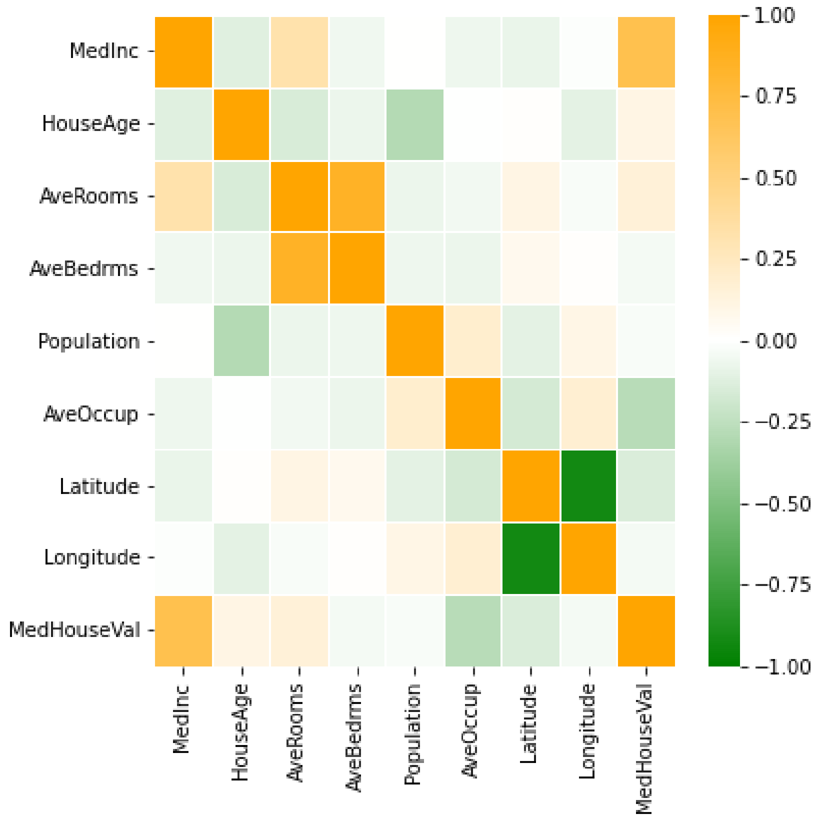
  - Response: Median house price per block (analyzed as log(median price)
  - Predictors:
    - median income, median age of house
    - total rooms, total bedrooms
    - population, number of households
    - latitude and longitude of block

California Housing: Illustrations

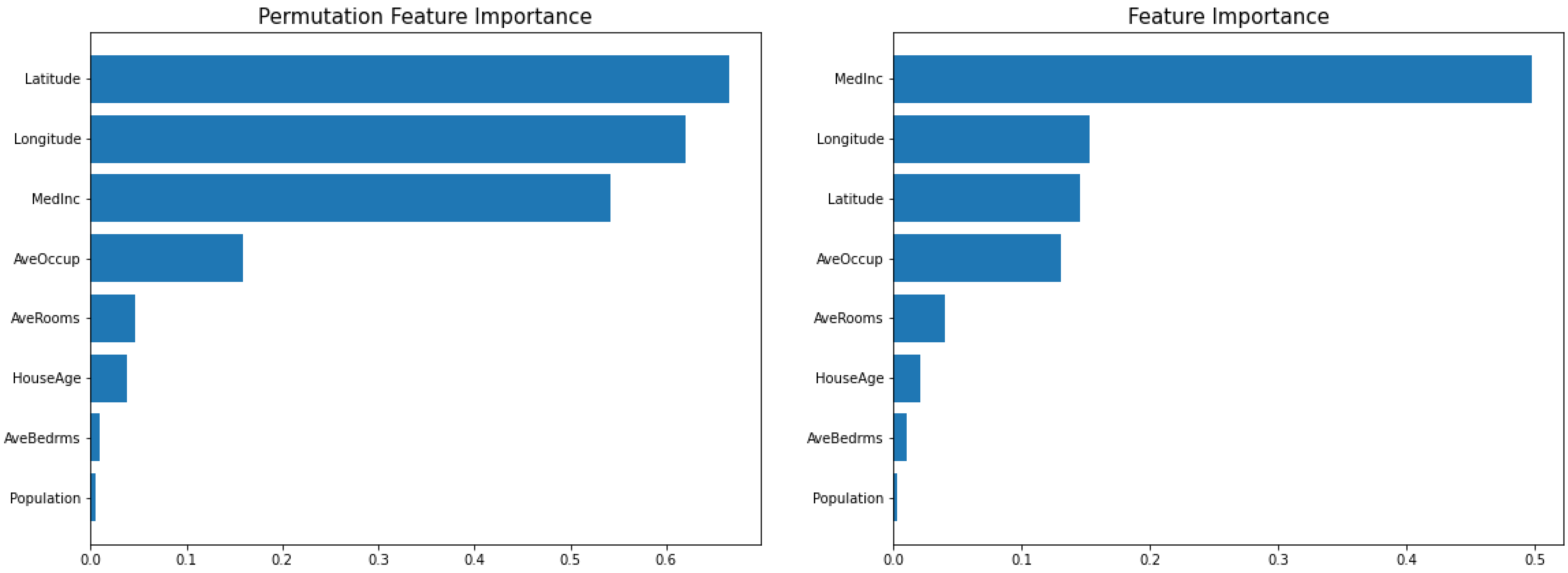
- Pre-processing and exploratory analysis







Comparison of Permutation and Interpretable ML Feature Importance



Python Interpretable Machine Learning: PiML

Python toolbox for Interpretable ML
Application to model development and validation

Python Interpretable Machine Learning: PiML

- PiML supports various machine learning models in the following two categories:

- Low-code environment: Inherently interpretable algorithms and model diagnostics
  - EBM: Explainable Boosting Machine
    - (Nori, et al. 2019; Lou, et al. 2013)
  - GAMI-Net: Generalized Additive Model with Structured Interactions
    - (Yang, Zhang and Sudjianto, 2021)
  - ReLU-DNN: Deep ReLU Networks using Aletheia Unwrapper
    - (Sudjianto, et al. 2020)

- High-code environment: Black-box algorithms and post-hoc diagnostics
  - LightGBM or XGBoost
  - RandomForest
  - DNNs with softmax/tanh activations

- Installation details will be provided later

Datasets in PiML

Taiwan Credit Data

https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

- Default data of credit card clients (n=30,000) in Taiwan from 2005-04 to 2005-09.
- Response: 1=customers defaulted, 0 did not default

Var	Description	Var	Description
X1	amount of given credit (NT dollars):	X12	amount of bill statement (NT dollar), Sept 2005
X2	gender (1 = male; 2 = female)	X13	amount of bill statement (NT dollar), Aug 2005
X3	education (1 = graduate school; 2 = university; 3 = high school; 4 = others).	X14	amount of bill statement (NT dollar), July 2005
X4	marital status (1 = married; 2 = single; 3 = others)	X15	amount of bill statement (NT dollar), June 2005
X5	age (year)	X16	amount of bill statement (NT dollar), May 2005
X6	repayment (delinquency) status in Sept, 2005	X17	amount of bill statement (NT dollar), April, 2005
X7	repayment status in Aug, 2005;	X18	amount paid in Sept, 2005
X8	repayment status in July, 2005;	X19	amount paid in Aug, 2005
X9	repayment status in June 2005;	X20	amount paid in July, 2005
X10	repayment status in May 2005;	…	
X11	repayment status in April, 2005;	X23	amount paid in April, 2005

Bike Rentals

Public dataset hosted on **UCI machine learning repository** that has been analyzed by various people

Dataset has **17,379 observations** from Capital Bikeshare system

Response: hourly (and daily) bike rental counts for two years 2011-12 → analyzed as log-count

Predictors: weather and time information

Original 17 variables → reduced to 11

season (1:winter; 2:spring, 3:summer, 4:fall)	***mnth*** (month = 1 to 12)
hr (hour = 0 to 23)	***holiday*** (1 if yes and 0 if not)
weekday (0 = sunday to 6 = saturday)	***workingday*** (1 if working and 0 if not)
weathersit (1:clear, 2: misty + cloudy; 3: light snow; 4 :heavy rain)	***temp*** (normalized temperature)
atemp (ambient temp, normalized);	***hum*** (humidity)
windspeed (normalized)	

California Housing Data

- Source:
  - Pace, R. K. and Barry, R. Sparse Spatial Autoregressions, Statistics and Probability Letters, (1997) 291-297
  - Dataset can be obtained from the StatLib repository
  - Available in PiML
- Description
  - Data from 1990 Census for California
  - Data for housing of "block groups" – geographically compact area
  - Geographical area of block varies inversely with population density
  - Total of 20,640 observations: response (house price) and 8 predictors
  - Goal: Model house price as a function of predictors

  - Response: Median house price per block (analyzed as log(median price)
  - Predictors:
    - median income, median age of house
    - total rooms, total bedrooms
    - population, number of households
    - latitude and longitude of block