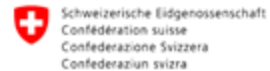


DIGITAL FINANCE

This project has received funding from the Horizon Europe research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 101119635



State Secretariat for Education,
Research and Innovation SERI



**Funded by
the European Union**



White Box AI - Intrinsic Explainability

Faizan Ahmed



Funded by
the European Union

Reading

- **Mandatory Reading Material**

- Molnar, Christoph. *Interpretable machine learning*. 2020. [Section 6-11]
<https://christophm.github.io/interpretable-ml-book/>
- Explainable AI with Python Chapter "Intrinsic Explainable Models"
https://link.springer.com/chapter/10.1007/978-3-030-68640-6_3 [Access through University Library]

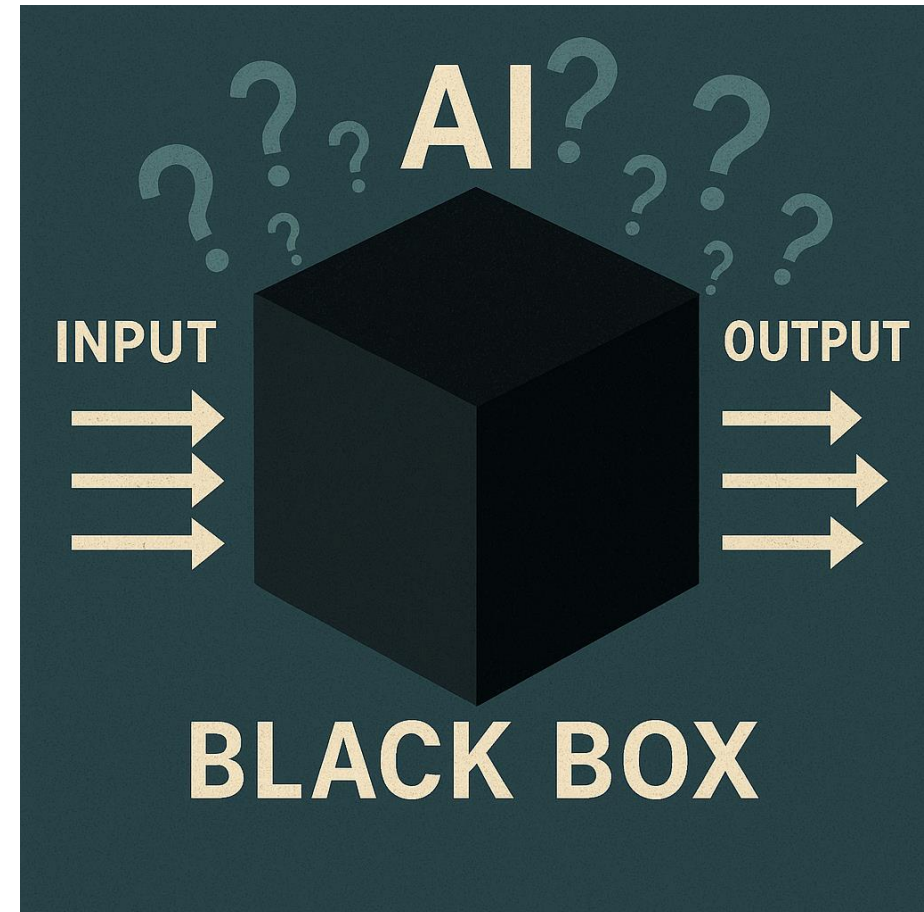
- **Recommended Reading Material**

- Lipton, Zachary C. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." *Queue* 16.3 (2018): 31-57.
<https://arxiv.org/abs/1606.03490>
- Fisher, Aaron, Cynthia Rudin, and Francesca Dominici. "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously." <http://arxiv.org/abs/1801.01489> (2018).
- Wei, Pengfei, Zhenzhou Lu, and Jingwen Song. "Variable importance analysis: a comprehensive review." *Reliability Engineering & System Safety* 142 (2015): 399-432



What is it all about?

- Modern AI, especially deep learning, can achieve incredible performance. However, its decision-making process is often opaque and difficult for humans to understand. This lack of transparency can lead to:
 - Vulnerability to bad data or design.
 - Distrust from stakeholders and the public.
 - Significant legal and ethical risks.

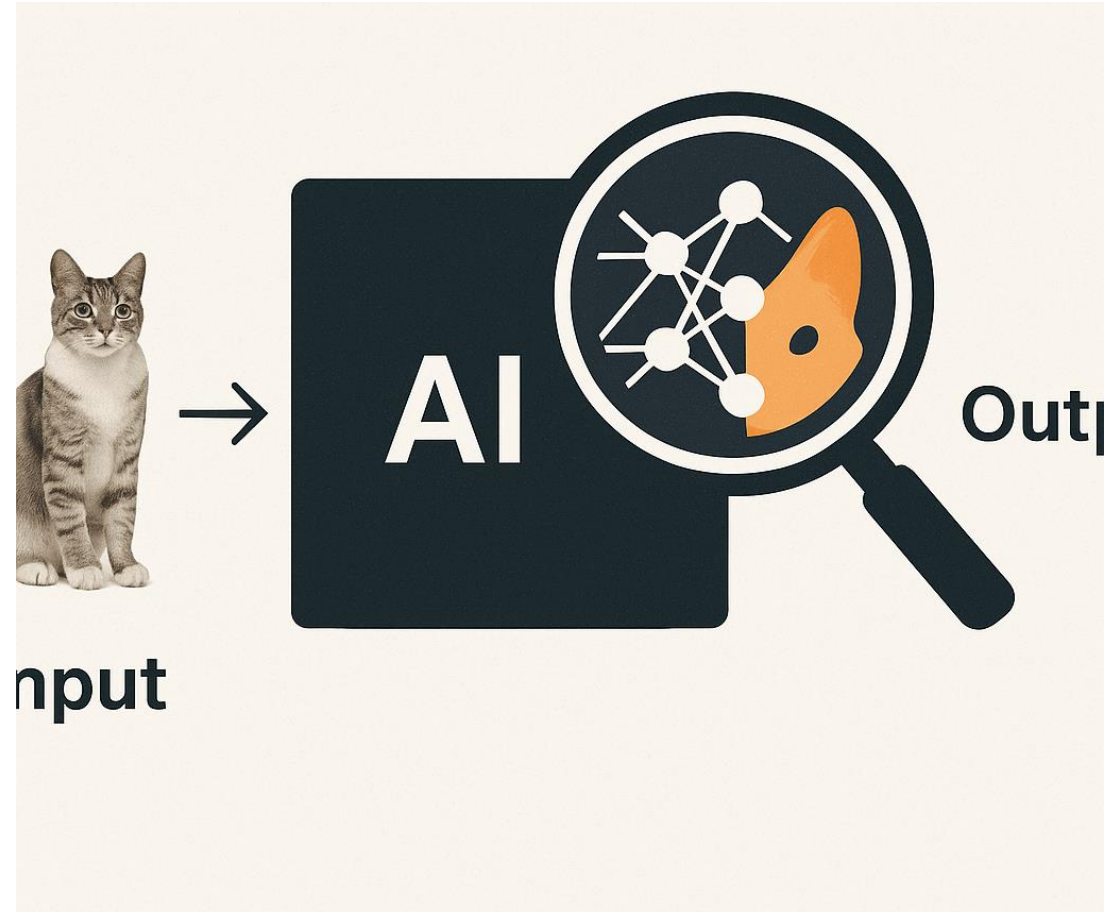


What is Explainable AI (XAI)?

Explainable AI, or XAI for short is

- a set of methods and tools that can be adopted to make Machine Learning (ML) models
 - understandable to human beings.

As in reasoning about a decision with humans, XAI provides the tools to understand why a decision or output a machine learning model makes.



XAI – Manifesto

F.A.S.T.



Fair → AI decisions must be unbiased.

Example: A hiring algorithm should not favor male candidates over equally qualified female candidates.



Accountable → AI must explain and justify its decisions.

Example: A medical AI recommending a cancer treatment must show which symptoms and lab results led to the suggestion.



Secure → AI must be robust against attacks.

Example: A self-driving car's vision system should not be tricked by stickers on a stop sign into misclassifying it.



Transparent → Inner workings must be interpretable.

Example: A loan approval AI should show which features (income, credit score, debt ratio) most influenced the decision.



Taxonomy of Explainability (What, How, To Whom)

What to Explain (Content):

- **Local:** Why was *this specific* decision made?
- **Global:** How does the model work *in general*?
- **Counterfactual:** What needs to change to get a different outcome?

How to Explain (Communication):

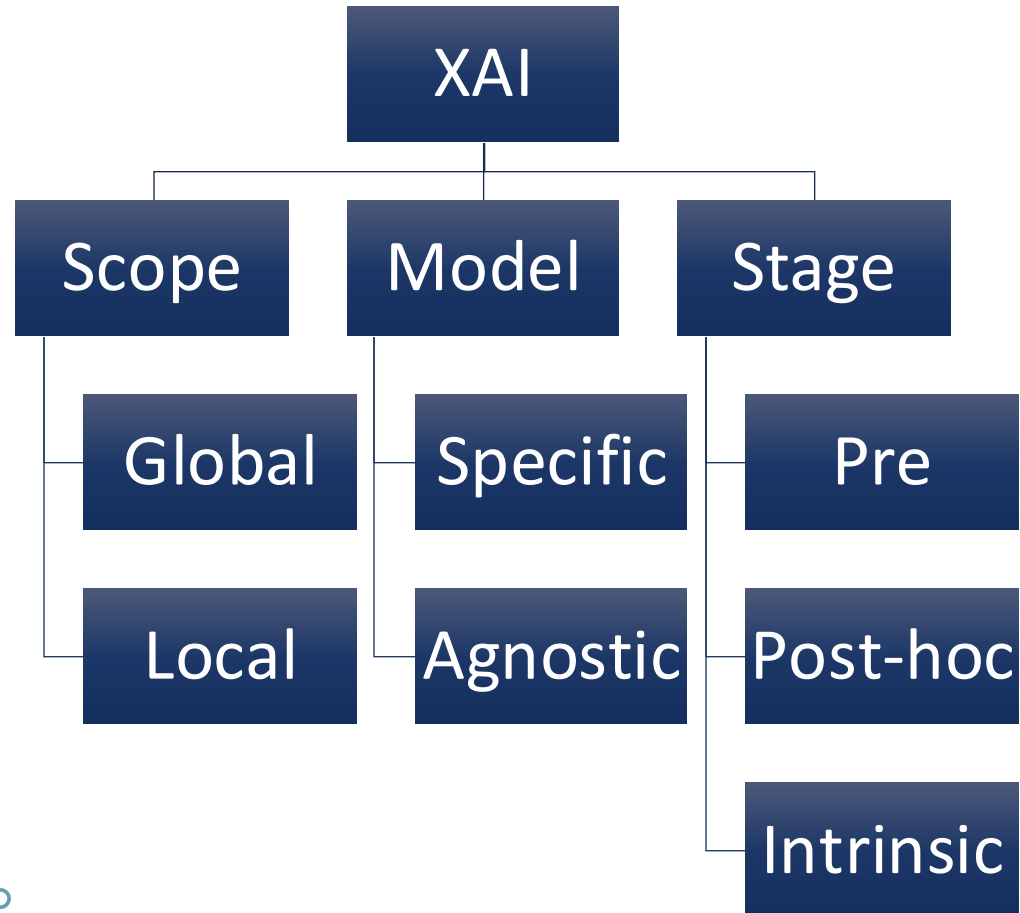
- Text, Graphics (charts, decision trees), or Multimedia.

To Whom to Explain (Stakeholders):

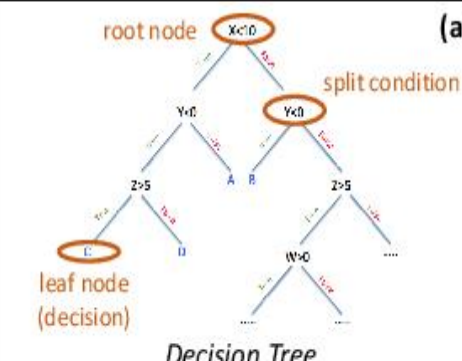
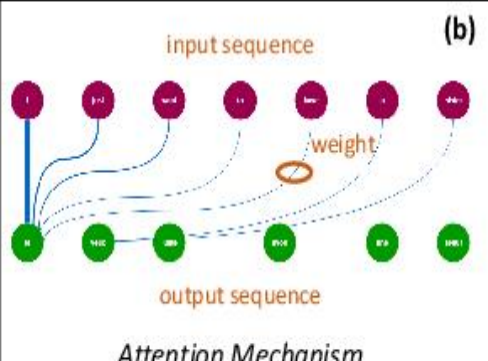
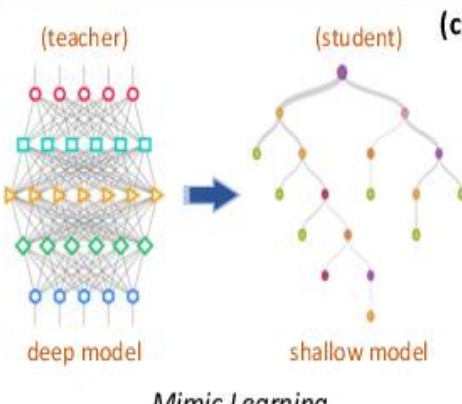
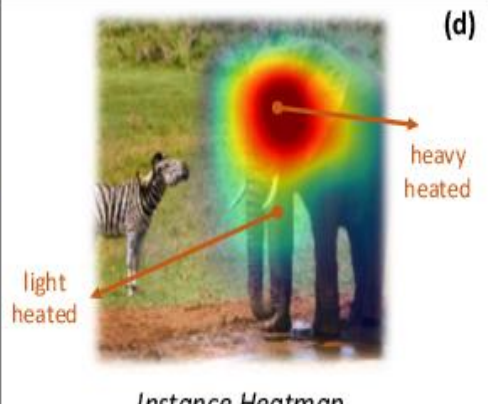
- Explanations must be tailored to the audience (e.g., technical for developers, simple for customers).



Taxonomy of XAI



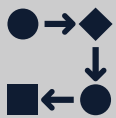
- <https://arxiv.org/pdf/1907.06831.pdf>

		Interpretation Scope	
		Global	Local
Interpretation Manner	Intrinsic	 <p>(a) Decision Tree</p>	 <p>(b) Attention Mechanism</p>
	Posthoc	 <p>(c) Mimic Learning</p>	 <p>(d) Instance Heatmap</p>

Intrinsically Interpretable Models



Models that are interpretable by design.



No post-processing steps are needed to achieve interpretable.



Linear Regression

A **linear regression model** predicts a target variable as a **weighted sum of input features**. It assumes a linear relationship between inputs and output.

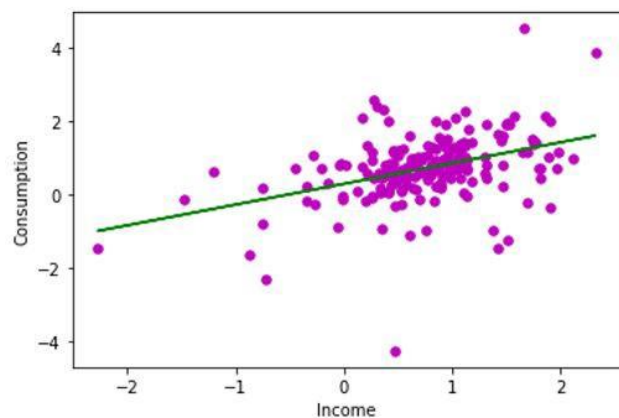
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

- Minimize the **squared differences** between actual and predicted values.

$$\hat{\beta} = \arg \min_{\beta_0, \dots, \beta_p} \sum_{i=1}^n \left(y^{(i)} - \left(\beta_0 + \sum_{j=1}^p \beta_j x_j^{(i)} \right) \right)^2$$

- **Assumption:** Linearity, normality, Homoscedasticity (constant variance), Independence, Fixed features, Absence of multicollinearity





	Consumption	Income	Production	Savings	Unemployment
1970-01-01	0.615986	0.972261	-2.452700	4.810312	0.9
1970-04-01	0.460376	1.169085	-0.551525	7.287992	0.5
1970-07-01	0.876791	1.553271	-0.358708	7.289013	0.5
1970-10-01	-0.274245	-0.255272	-2.185455	0.985230	0.7
1971-01-01	1.897371	1.987154	1.909734	3.657771	-0.1
...
2015-07-01	0.664970	0.801663	0.380606	3.180930	-0.3
2015-10-01	0.561680	0.740063	-0.845546	3.482786	0.0
2016-01-01	0.404682	0.519025	-0.417930	2.236534	0.0
2016-04-01	1.047707	0.723721	-0.203319	-2.721501	-0.1
2016-07-01	0.729598	0.644701	0.474918	-0.572858	0.0

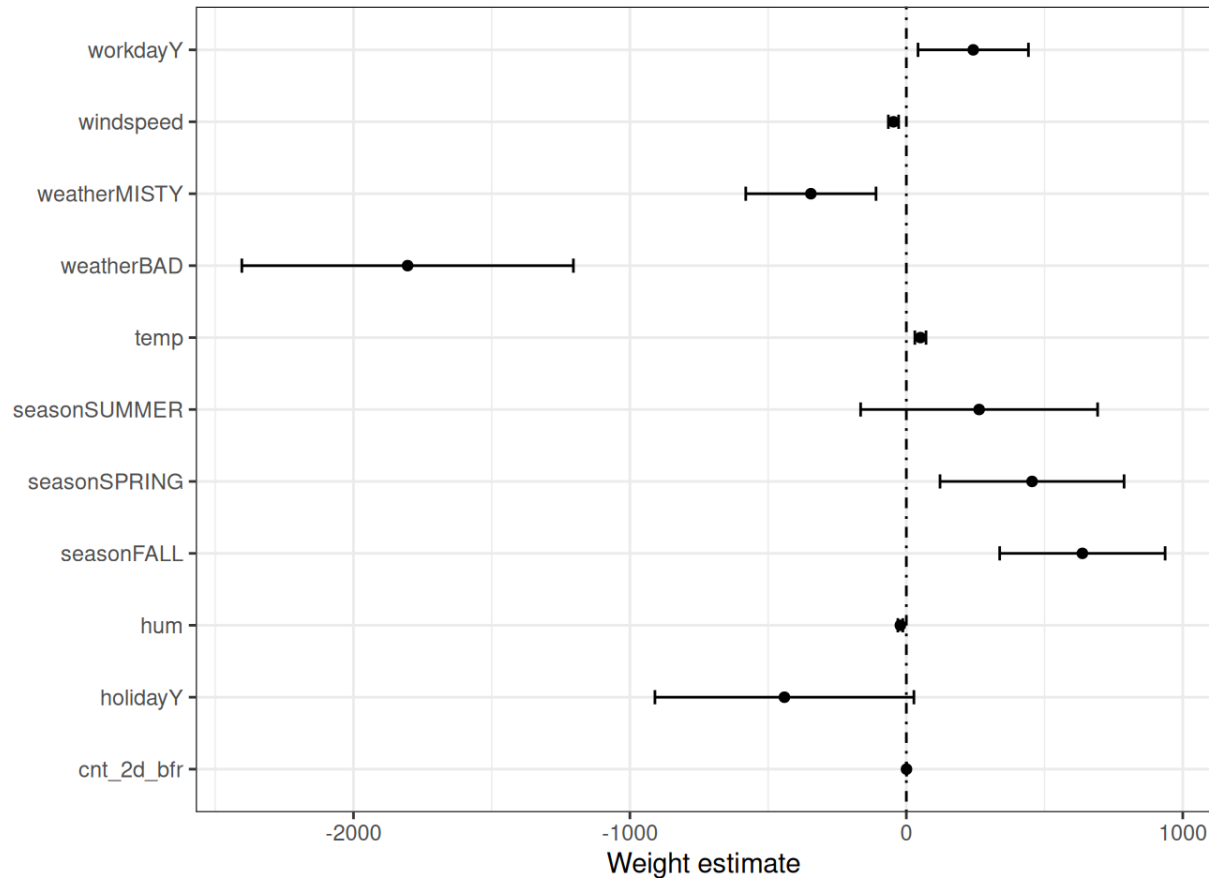
Linear Regression Example

$$\text{Consumption} = a_1 \text{Income} + a_2 \text{Production} + a_3 \text{Savings} + a_4 \text{Unemployment} + b$$

Intercept	0.2673
Income	0.7145
Production	0.0459
Savings	-0.0453
Unemployment	-0.2048



Linear Regression- Interpretations



- **Numerical feature:** An increase of feature x_j by one unit increases the prediction for y by β_j units when all other feature values remain fixed.
- **Categorical feature:** Changing feature x_j from the reference category to the other category increases the prediction for y by β_j when all other features remain fixed.
- **Feature Importance:** The importance of a feature in a linear regression model can be measured by the absolute value of its t-statistic.

$$t_{\hat{\beta}_j} = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}$$



Property	Assessment
Completeness	Full completeness achieved without the need of trading-off with interpretability being an intrinsic explainable model
Expressive power	Correlation coefficients provide a direct interpretation of the linear regression weights
Translucency	High, we can look directly at the internals to provide explanations
Portability	Low, explanations rely specifically on linear regression machinery
Algorithmic complexity	Low, no need of complex methods to generate explanation
Comprehensibility	Good level of human-understandable explanations to build as much confidence as possible

Linear Regression

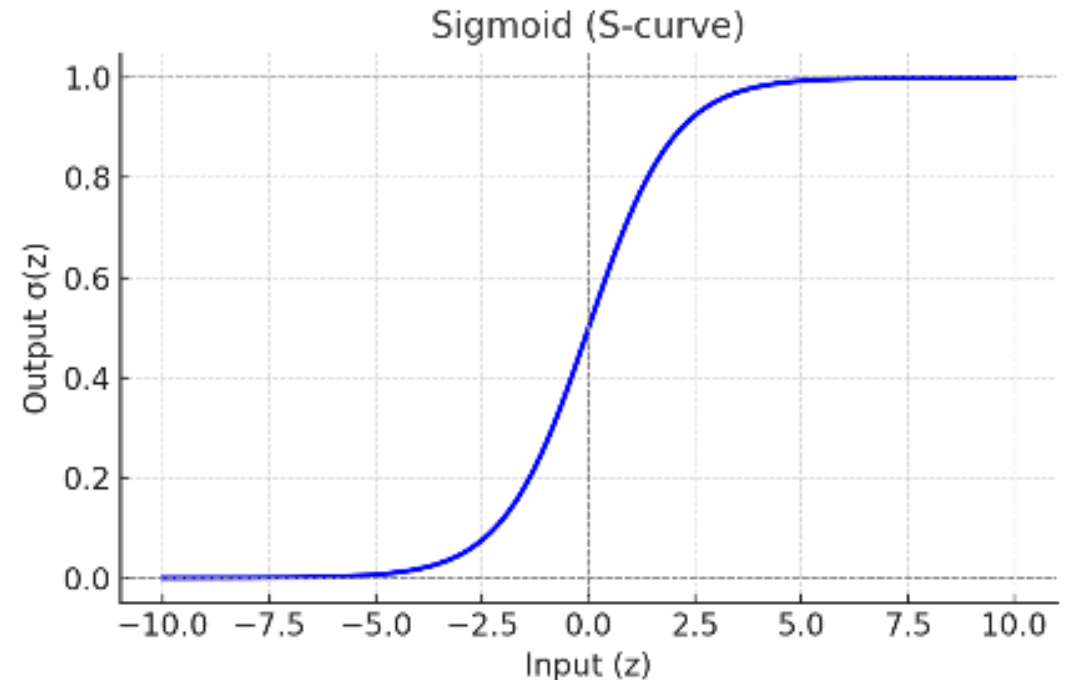
- Further reading:
 - **Sparse linear models-**



Logistic Relation

- To predict the probability of an event (e.g., tumor is malignant) given input features (e.g., tumor size, age, etc.).
 - Linear regression could predict probabilities >1 or $<0 \rightarrow$ not valid.
 - Need something that **always stays between 0 and 1.**
- S-curve or Sigmoid
 - Always between 0 and 1
 - Mathematically for any real number z

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



Logistic Relation

- Given features x_1, x_2, \dots, x_p , assuming they combines linearly

$$z = \beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_p x_p^{(i)}$$

- The formula**

$$P(y^{(i)} = 1) = \frac{1}{1 + \exp\left(-(\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_p x_p^{(i)})\right)}$$

- To retrieve coefficient for interpretation*

$$odds = \log\left(\frac{P(Y = 1)}{1 - P(Y = 1)}\right) = (\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_p x_p^{(i)})$$

- Log is difficult to interpret so [What happen when there is a unit change in the feature value]

$$\frac{odds(x_k + 1)}{odds(x_k)} = \exp(\beta_k)$$



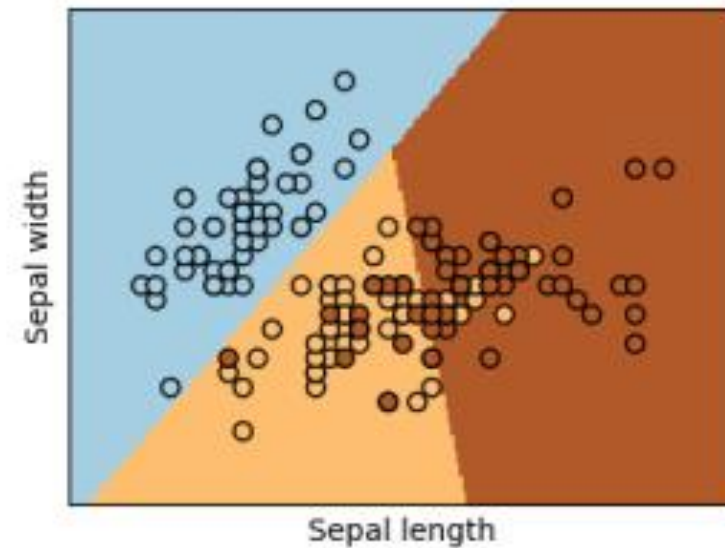
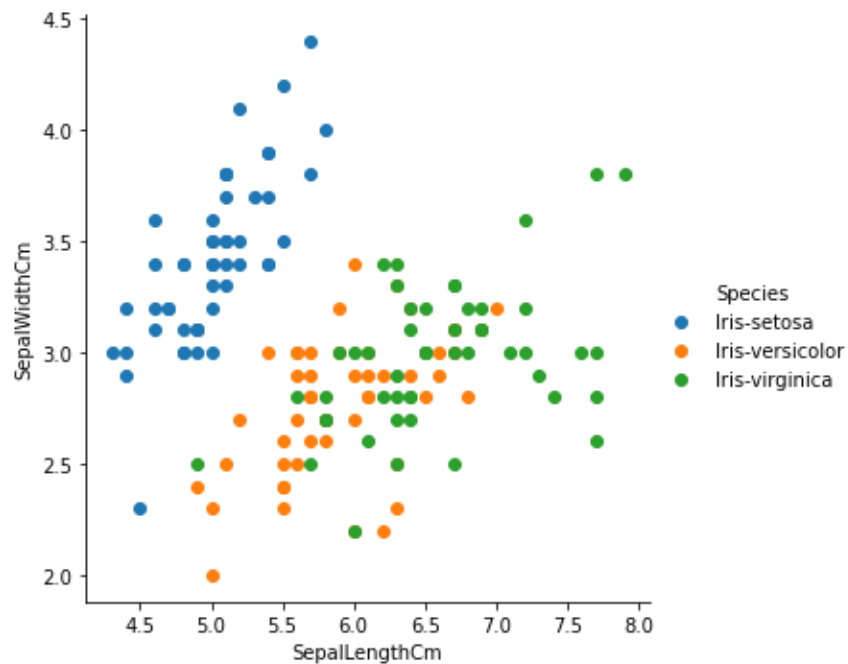
Explaining Logistic Regression

Feature type	Interpretation
Numerical	1-unit increase changes odds by factor of $\exp(\beta)$ - β is coefficient of feature of interest
Binary categorical	Reference category coded as 0; switching changes odds by $\exp(\beta)$
Categorical (>2)	Use one-hot encoding (L-1 dummies); each dummy works like binary feature
Intercept	Odds when numerics=0 and categoricals=reference; usually not meaningful



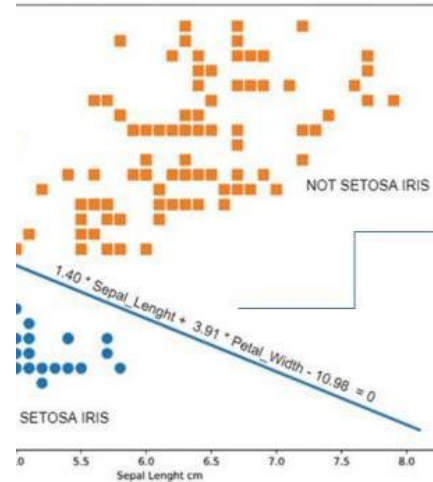
Explaining Logistic Regression- Example

- The Iris dataset predicts the species of a flower (Setosa, Versicolor, Virginica) from sepal and petal measurements.



Logistic Regression - in Practice

- For Setosa, a 1 cm increase in sepal width multiplies odds by 4.1.
- For Virginica, a 1 cm increase in petal length multiplies odds by 11.



$$\log \frac{P(Y = \text{Setosa})}{P(Y = \text{non-Setosa})} = \log \text{odds}(Y = \text{Setosa}) = m_0 + m_1 (\text{Sepal Length}) + m_2 (\text{Petal Width})$$

$$\frac{\text{odds}(x_1 + 1)}{\text{odds}(x_1)} = \exp(m_1)x_1 = \text{sepal length}$$

$$\frac{\text{odds}(x_2 + 1)}{\text{odds}(x_2)} = \exp(m_2)x_2 = \text{petal width}$$

	<i>Setosa</i>	<i>Versicolor</i>	<i>Virginica</i>
	1.44936119	1.61875962	0.21035554
	4.09477593	0.20667587	0.20414071
	0.11623966	1.55210045	11.00944064
	0.38491152	0.33653155	8.63283183



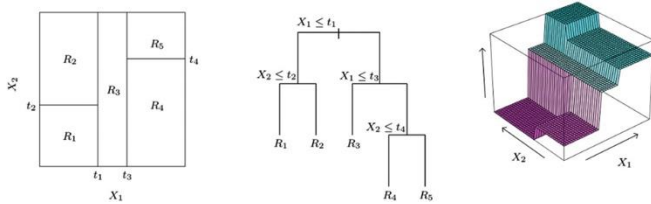
Logistic Regression

Property	Assessment
Completeness	Full completeness achieved without the need of trading-off with interpretability being an intrinsic explainable model
Expressive power	Less than linear regression case. Interpretation of coefficients is not so straightforward
Translucency	As any intrinsic explainable model, we can look at the internals. Weights are used to provide explanations but not so directly as in linear regression case
Portability	Method is not portable, specific for logistic regression
Algorithmic complexity	Low but not trivial as in linear regression case
Comprehensibility	Explanations are human understandable also for not technical people



Decision Trees

- ❑ Regression models fails if:
 - ❑ there is a non-linear relations
 - ❑ And features interact with each other

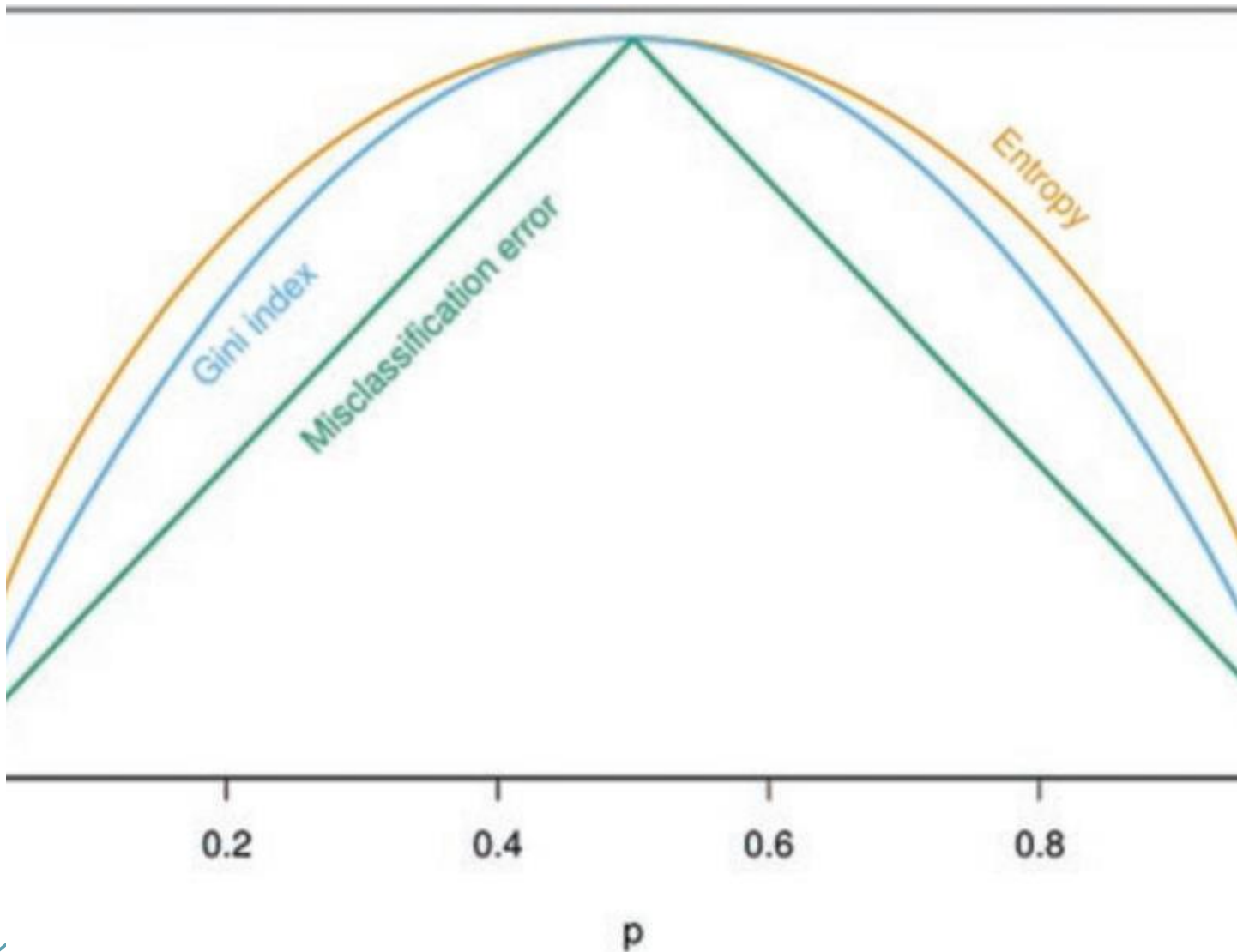


- **Decision Trees**

- Categorizes data through repeated splits based on feature values.
- Forms groups that predict outcomes in final nodes.
- Data is split multiple times creating subsets.
- Final subsets are called terminal or leaf nodes.
- Intermediate subsets are known as internal or split nodes.
- Outcome prediction in leaf nodes uses average outcome of training data.
- CART (Classification and Regression Trees) is the most popular.



Decision Trees - Different Methods to Split



- Impurity quantification
 - Gini equation: $1 - \sum_{i=1}^C (p_i)^2$
 - Shannon Entropy: $\sum_{i=1}^C -p_i \log_2(p_i)$
 - Classification Error: $1 - \max(p_i)$
 - C is the total number of classes, p_i is the probability of a random observation being class i in the remaining observation



Decision Trees

Decision Tree Interpretation

- Start at the **root node** and follow the branches (edges).
- Each edge represents a **condition** (e.g., "Feature \leq Threshold").
- **All conditions are connected with AND.**
- At the **leaf node**, the outcome is the **predicted value** (e.g., mean of instances in that node).

Feature Importance in Decision Trees

- For each feature:
- Check how much it **reduced variance** or **Gini index** at its splits.
- Sum all reductions where the feature was used.
- Scale total importance values to **100%**.
- Result: Each feature's importance = its **share of overall model impact**.

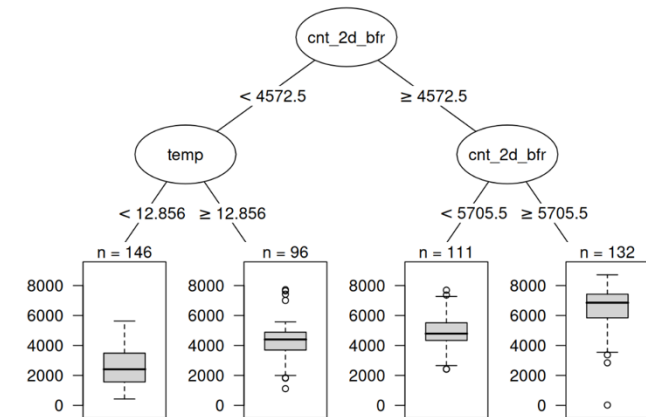


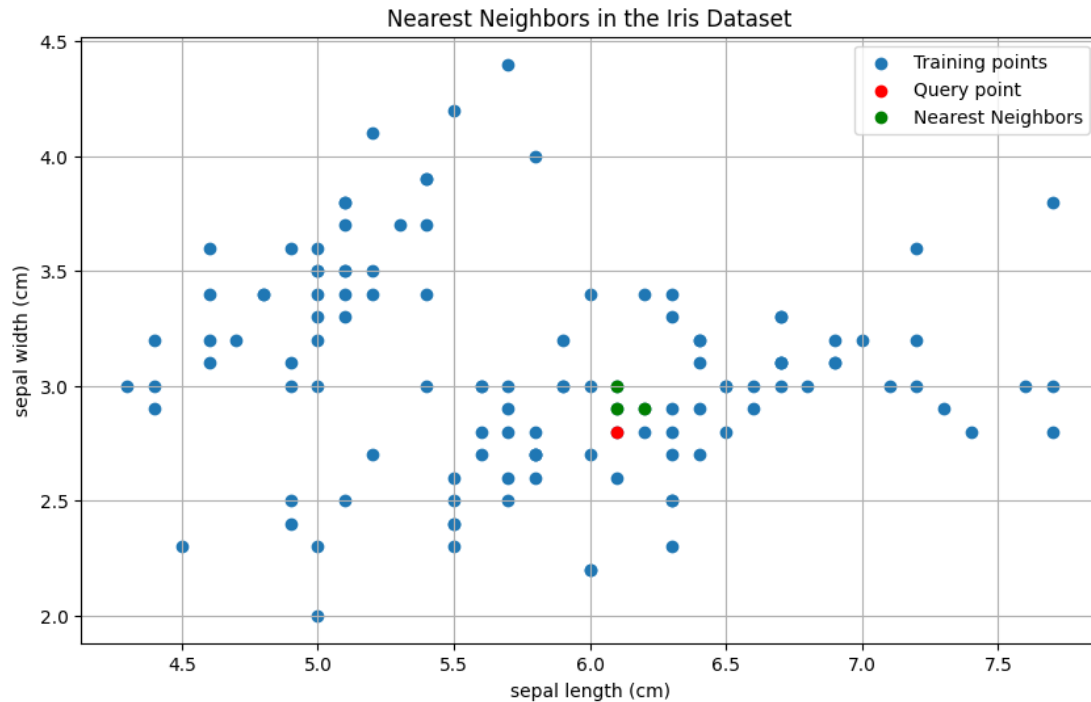
Figure 9.2: Regression tree fitted on the bike rental data. The boxplots show the distribution of bike counts in the terminal node.



Property	Assessment
Completeness	Full completeness achieved without the need of trading off with interpretability being an intrinsic explainable model
Expressive power	High expressive power; in fact DTs mimic to some extent human reasoning
Translucency	Intrinsic explainable easy to guess results
Portability	In fact many models are derived from decision trees such as Random Forest and boosted trees so DT results can be incorporated in such models
Algorithmic complexity	Decision trees are NP-complete, but we resort to heuristic for fast evaluation
Comprehensibility	Easy explanations to humans

Properties of explanations





K-Nearest Neighbors (KNN)

- KNN use the nearest neighbors of a data point for prediction.
 - **Regression:** Takes the average outcome of the neighbors.
 - **Classification:** Assigns the most common class of the nearest neighbors.
- Selecting the right number of neighbors (k).
- Choosing the distance metric to define the neighborhood.



Property	Assessment
<u>Completeness</u>	Full completeness achieved without the need of trading-off with interpretability being an intrinsic explainable model
Expressive power	High expressive power in terms of counterfactual and contrastive explanations
Translucency	Intrinsic explainable easy to guess results
Portability	KNN has a unique class in its own not portable
Algorithmic complexity	Simple training, complex inference step
Comprehensibility	Easy explanations to humans

Properties of explanations



Current trends and challenges

Model (Year)	Interpretability Approach	Application Domain
IGANN (2024)	Additive neural network (shape functions) + boosting (ELM-based training) for high accuracy and transparency.	Tabular data (e.g. productivity, credit, recidivism)
tiSFM (2023)	CNN architecture with motif-based filters & layers – parameters directly correspond to sequence motifs.	Genomics (DNA/RNA functional sequence modeling)
MoE-X (2025)	Mixture-of-Experts layer redesigned to be wide & sparse for disentangled neurons; uses ReLU experts + sparsity-aware routing.	NLP (language modeling, tested on chess moves and text)
InterpretCC (2024)	Conditional computation with feature-level gating or group-level expert routing – only human-relevant features/groups activated per sample.	Tabular, time-series, text (human-centric domains like education, health)
IDEAL (2025)	Prototype-based classification using frozen foundation model features; classifies by similarity to learned prototypes.	Vision (image classification, transfer & continual learning)
WYM (Why Match?) (2023)	Decision units (paired or unpaired feature tokens) as atomic inputs; interpretable matcher computes each unit's impact on a match/non-match decision.	Data integration (entity matching across databases)
B-cos Networks (2022)	Standard CNN layers replaced by B-cos transforms that enforce weight–input alignment; the network reduces to a single linear mapping aligned with meaningful features.	Vision (image recognition – e.g. integrated into ResNet, DenseNet on ImageNet)



Basic Explainability - Feature Importance, PDP, ICE

Reading

- **Mandatory Reading Material**

- Molnar, Christoph. *Interpretable machine learning*. 2020. [Section 23, 24, 19, 20, 13]
<https://christophm.github.io/interpretable-ml-book/>

- **Recommended Reading Material**

- Lipton, Zachary C. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." *Queue* 16.3 (2018): 31-57.
<https://arxiv.org/abs/1606.03490>

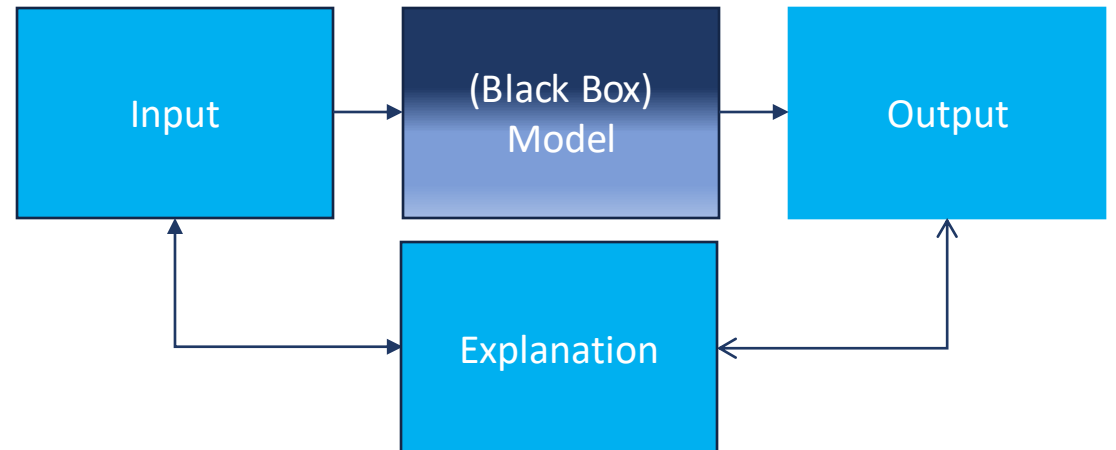
- **Libraries**

- MMD-critic <https://github.com/BeenKim/MMD-critic>
- ALE Plots: <https://github.com/blent-ai/ALEPython>



Model Agnostic Methods

- These are methods to produce explanations without relying on ML model internals, i.e. the ML model is treated like a black box.



Permutation Importance

- Measures the increase in the prediction error of the model after the feature values are permuted
- How: **only a column (feature) of the training data is shuffled and make the prediction again but with the shuffled values.**
- Note: we are creating a mismatch from the true data by shuffling only one column, i.e. the whole row is not shuffled.
- By shuffling a particular column only, if the output predictions falls significantly, then we know the feature was very important and vice versa, if the feature wasn't important then the performance does not fall.



f1	f2	f3	...	fn	y
2.29	3.47	2.55		3.17	0
2.86	2.38	0.72		3.37	0
0.95	0.44	0.08		1.61	0
1.28	0.48	0.10		3.12	1
0.74	1.32	1.41		3.42	1

Permutation Feature Importance

(Fisher, Rudin, and Dominici)

- **Input:** Trained model \hat{f} , feature matrix X , target vector y , error measure $L(y, \hat{f})$

$$e_{orig} = L(y, \hat{f})$$

- For each feature $j \in \{1, \dots, p\}$ do
 - Generate X_{perm} by permuting feature j
 - Estimate error $e_{perm} = L(Y, \hat{f}(X_{perm}))$
 - Compute feature importance $FI_j = \frac{e_{perm}}{e_{orig}}$ or $FI_j = e_{perm} - e_{orig}$



DIGITAL

- Sort feature by descending FI_j

f1	f2	f3	...	fn	y
2.29	3.47	2.55		3.17	0
2.86	2.38	0.72		3.37	0
0.95	0.44	0.08		1.61	0
1.28	0.48	0.10		3.12	1
0.74	1.32	1.41		3.42	1

Fisher, Aaron, Cynthia Rudin, and Francesca Dominici. "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously." <http://arxiv.org/abs/1801.01489> (2018).

Permutation Feature Importance

Penguin Sex Classification: Logistic Regression Models

- Trained 3 logistic regression models to predict penguin sex
- Used 2/3 of the data for training, 1/3 for feature importance evaluation
- Measured error using **log loss**

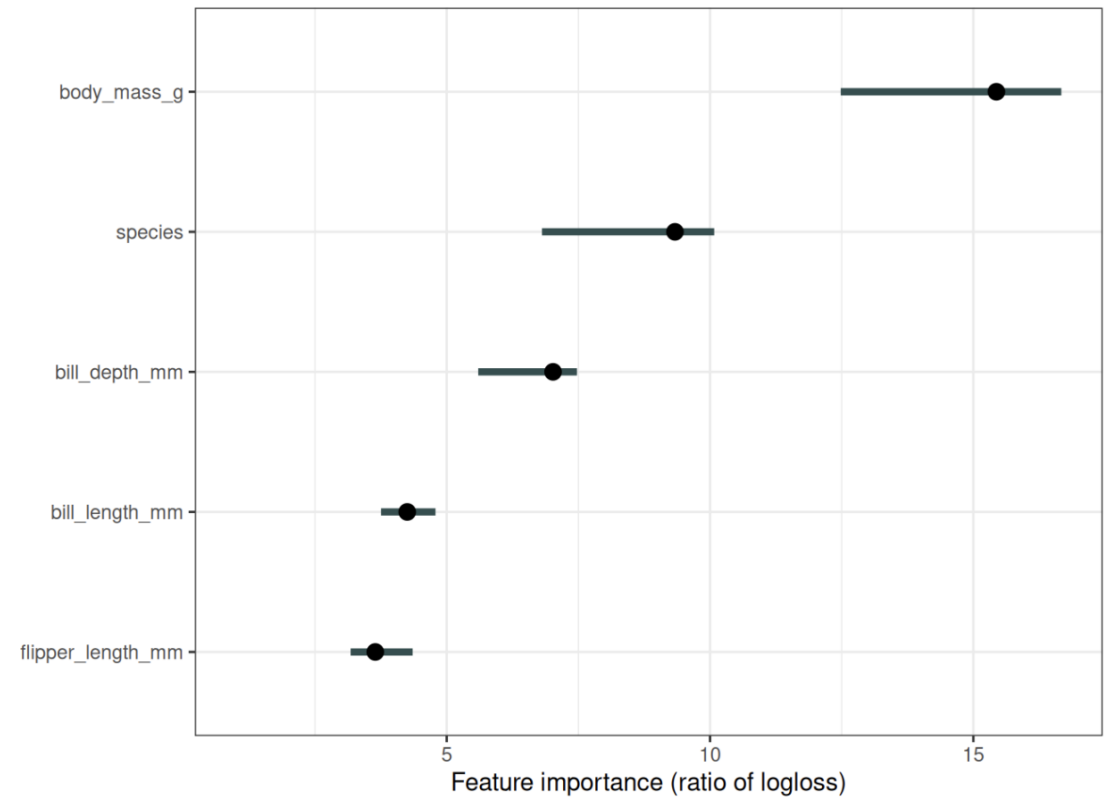


Figure: Permutation feature importance values for the penguin classification task. [Source](#)



Permutation Feature Importance

Model baseline accuracy = **95%**

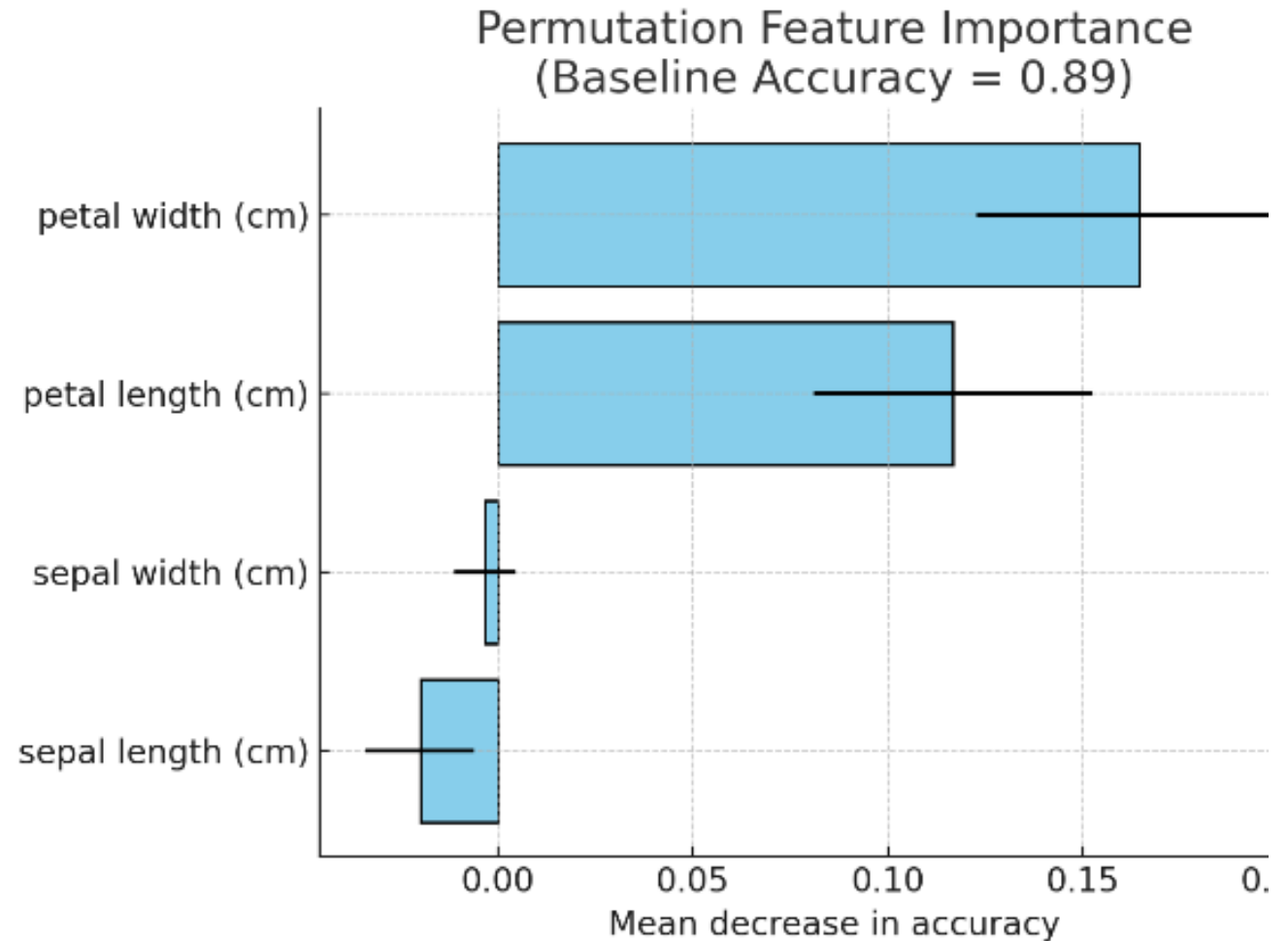
Shuffle **Petal Length** → accuracy = 60% → 🌟 most important

Shuffle **Petal Width** → accuracy = 65% → 🌟 very important

Shuffle **Sepal Length** → accuracy = 90% → moderate importance

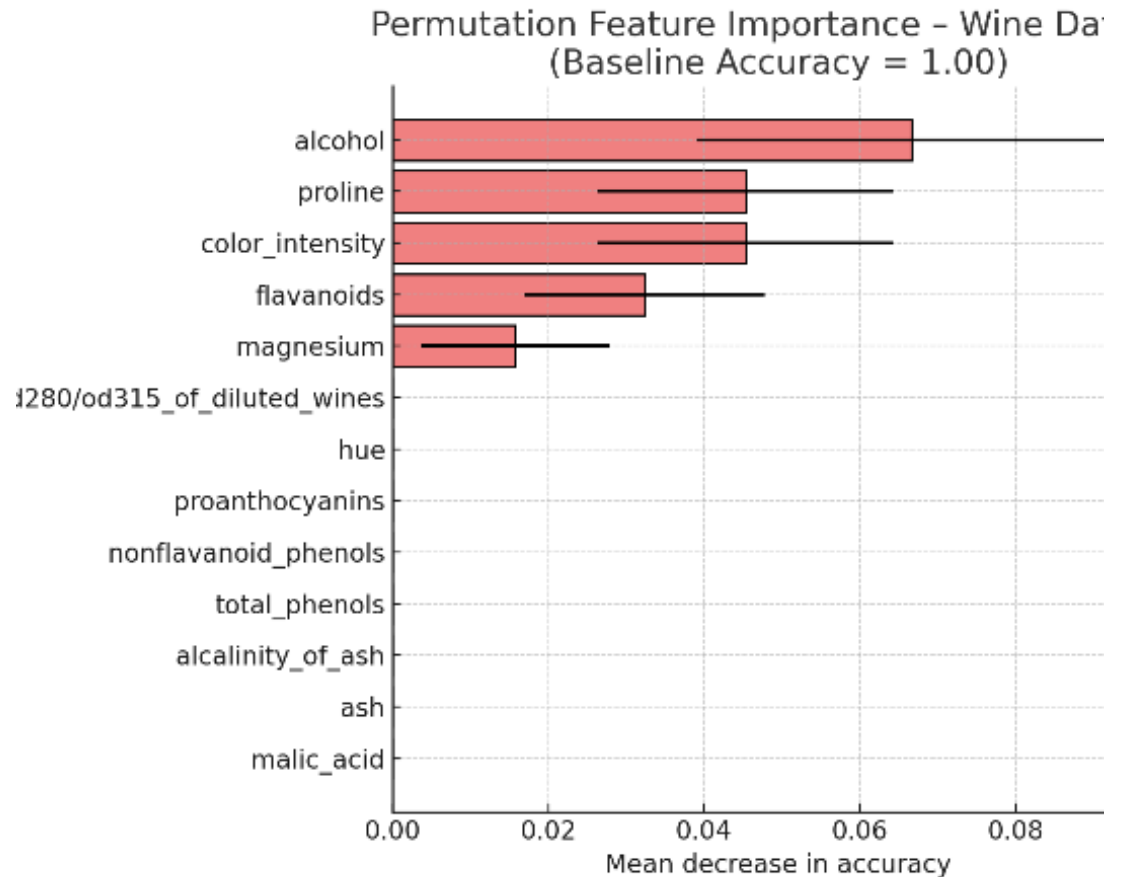
Shuffle **Sepal Width** → accuracy = 94% → low importance

👉 **Key Idea:** Importance = how much performance drops when a feature's information is destroyed.



Permutation Feature Importance

- **Samples:** 178 wines
- **Classes:** 3 (cultivars)
- **Features (13):** chemical analysis of wines
- Alcohol, Malic acid, Ash, Magnesium, Flavanoids, Proline, etc.



Permutation Feature Importance

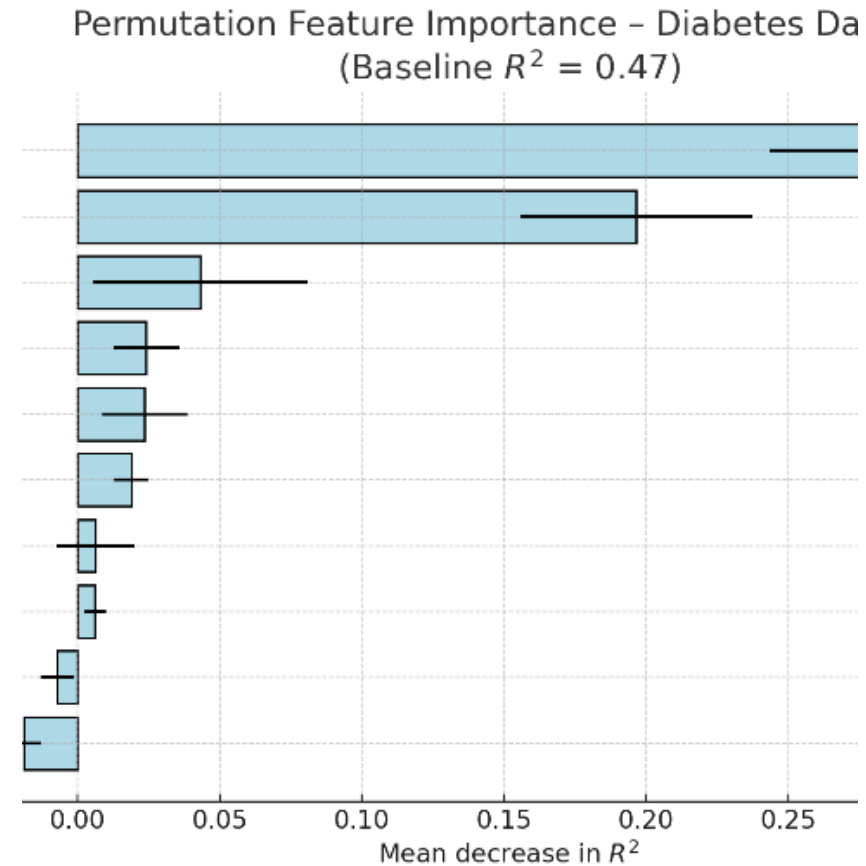
California Housing Dataset Overview

- **Samples:** ~20,000 houses in California
- **Target:** Median house price (continuous variable)
- **Features (8):**
 - MedInc – Median income in the district
 - HouseAge – Median house age
 - AveRooms – Average number of rooms per household
 - AveBedrms – Average number of bedrooms
 - Population – Population of the district
 - AveOccup – Average household size
 - Latitude, Longitude – Location



Permutation Feature Importance

- **Samples:** 442 patients
- **Features (10):** baseline variables such as
 - Age
 - Sex
 - Body Mass Index (BMI)
 - Blood Pressure
 - Blood serum measurements (6 biochemical markers)
- **Target:** Disease progression one year after baseline (continuous measure).



Permutation Feature Importance

- Nice Interpretation
- Comparable across different problems.
- Need access to the true outcome
- Can be biased by unrealistic data instances

Further reading: <https://christophm.github.io/interpretable-ml-book/feature-importance.html>

Another one: Wei, Pengfei, Zhenzhou Lu, and Jingwen Song. "Variable importance analysis: a comprehensive review." Reliability Engineering & System Safety 142 (2015): 399-432



Ceteris Paribus Plots

Ceteris paribus (CP) plots visualize how changes in a single feature change the prediction of a data point.



DIGITAL

Algorithm 1 Ceteris Paribus (CP) Profile Computation[Numerical]

Require: Data point $\mathbf{x}^{(i)}$, feature j , prediction model \hat{f}

- 1: Create an equidistant value grid z_1, z_2, \dots, z_K , where typically:
 $z_1 = \min(x_j)$ and $z_K = \max(x_j)$
 - 2: **for** each grid value $z_k \in \{z_1, \dots, z_K\}$ **do**
 - 3: Create new data point $\mathbf{x}_{x_j:=z_k}^{(i)}$ by replacing feature j with z_k
 - 4: Compute prediction $\hat{f}(\mathbf{x}_{x_j:=z_k}^{(i)})$
 - 5: **end for**
 - 6: Visualize the CP curve:
 - Plot line for data points $\{z_k, \hat{f}(\mathbf{x}_{x_j:=z_k}^{(i)})\}_{k=1}^K$
 - Plot dot for original data point $(x_j^{(i)}, \hat{f}(\mathbf{x}^{(i)}))$
-

Algorithm 1 Ceteris Paribus (CP) Profile for Categorical Feature

Require: Data point $\mathbf{x}^{(i)}$, categorical feature j , prediction model \hat{f}

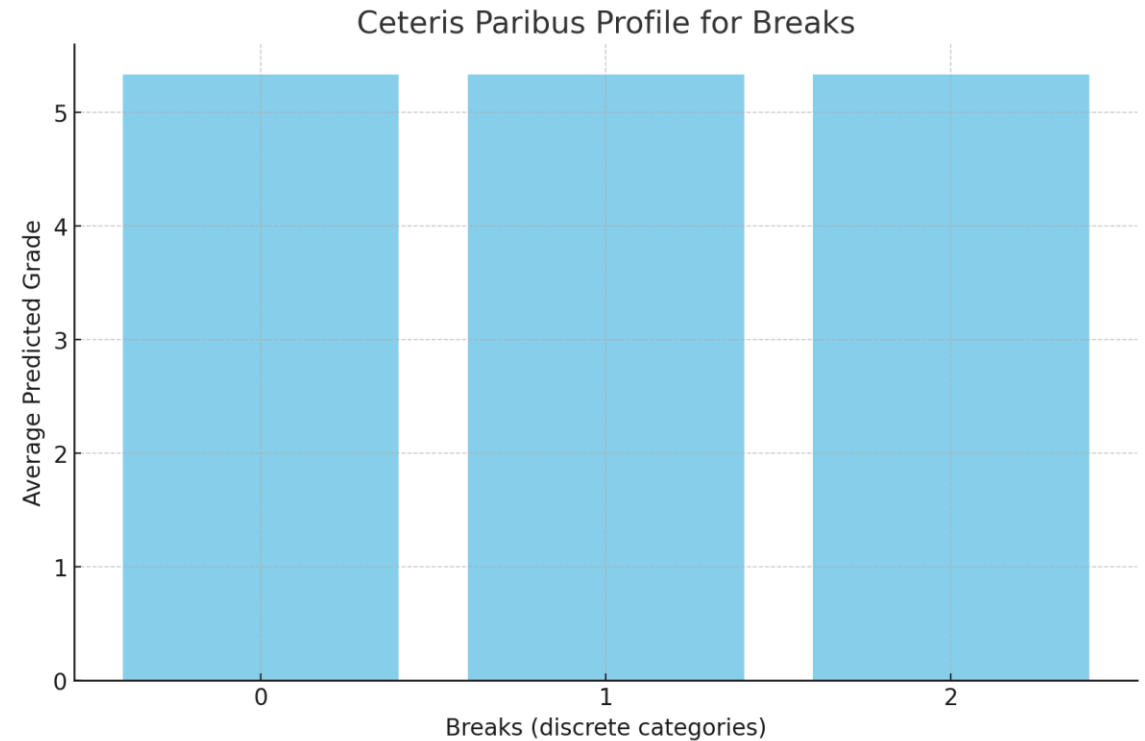
- 1: Create list of unique categories z_1, z_2, \dots, z_K
 - 2: **for** each category $z_k \in \{z_1, \dots, z_K\}$ **do**
 - 3: Create new data point $\mathbf{x}_{x_j:=z_k}^{(i)}$ by replacing feature j with z_k
 - 4: Compute prediction $\hat{f}(\mathbf{x}_{x_j:=z_k}^{(i)})$
 - 5: **end for**
 - 6: **Visualize:** Create bar plot or dot plot with categories on x-axis and predictions on y-axis
-

Ceteris Paribus Plots

Study hours (x1)	Breaks (x2)	Sleep(x3)	grade
1	2	7	5
2	2	6	6
3	1	7	7
4	1	6	8
5	0	7	9
6	0	5	9

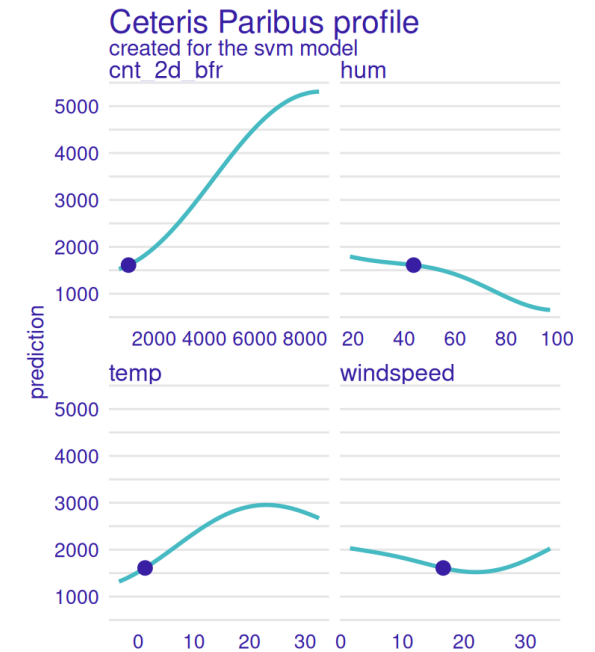
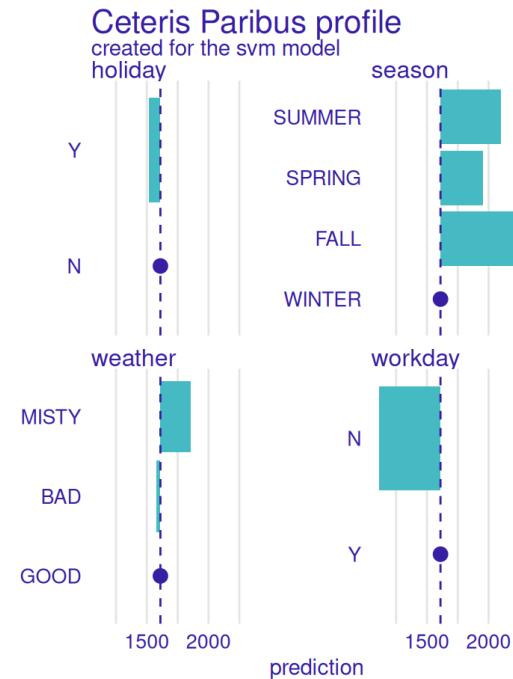
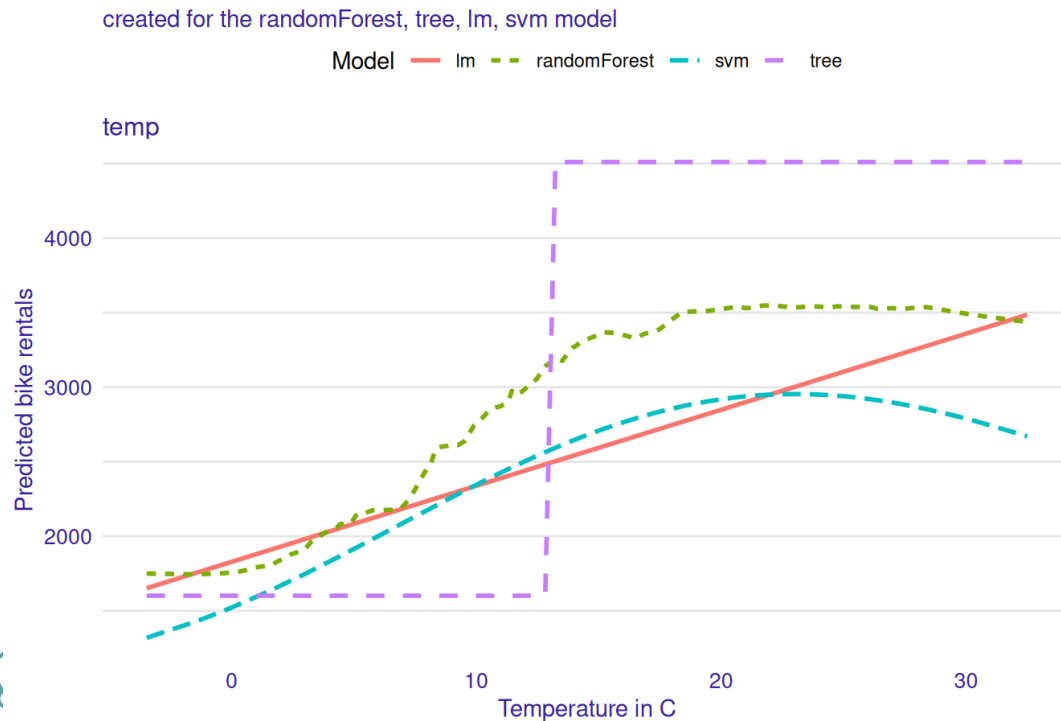


DIGITAL



Ceteris Paribus Plots

- Compare features.
- Compare models from different machine learning algorithms or with different hyperparameter settings.
- Compare class probabilities.
- Compare different data points.
- Subset the data (e.g., by a binary feature) and compare CP curves.



Individual Conditional Expectation (ICE)

Show **one line per instance**, tracking how its prediction changes as a feature varies.

Computation:

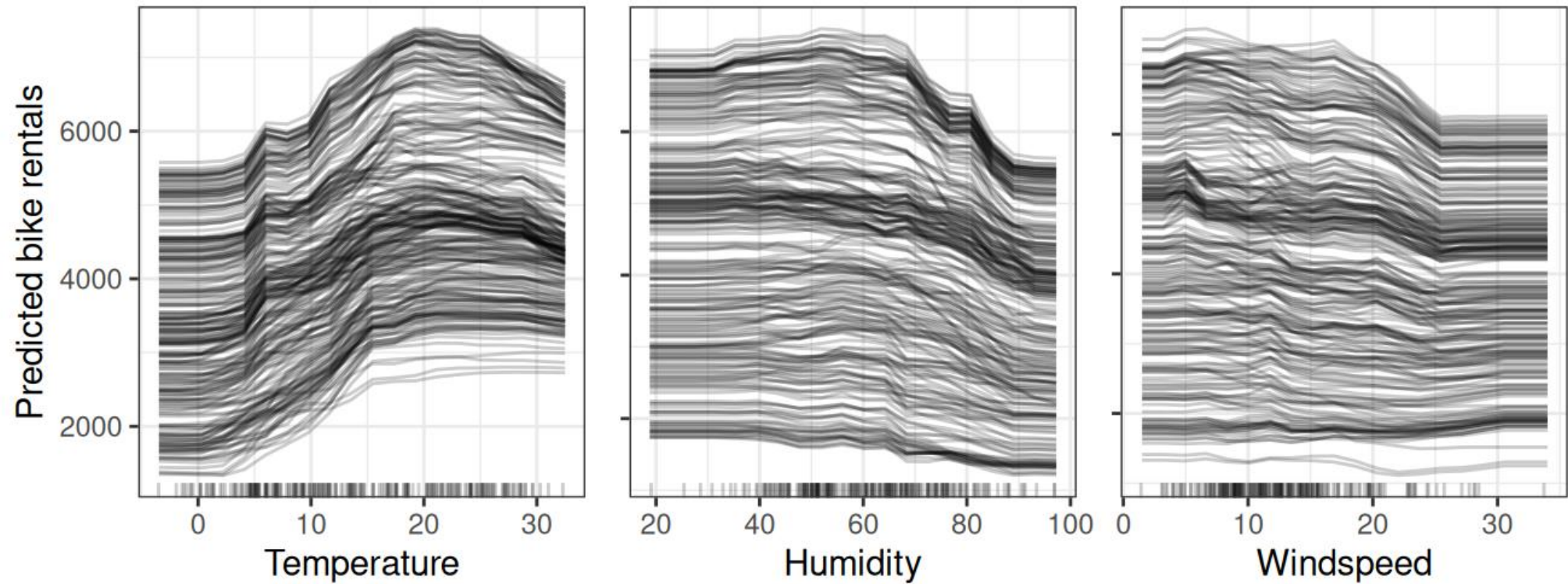
- Keep all other features fixed.
- Replace chosen feature with grid values.
- Predict with the model.

Result: Each line = one instance's ceteris paribus curve.

ICE = many CP curves shown together.



Individual Conditional Expectation (ICE)



Partial Dependence Plot

- Partial Dependence Plot (PDP), sketches the functional form of the relationship between an input feature and the target.
 - *show the average effect on predictions as the value of feature changes.*
 - Reveal if the relationship is **linear, monotonic, or complex**.
 - **Example:** For a **linear regression model**, PDPs will **always show a straight line**.
- **Assumption:** *the feature of interest are independent from the complement features*
 - this method is applied to a model which is already trained (can be used in conjunction with permutation importance)
 - use it to see "how" the predictions are changed by changes in a feature.

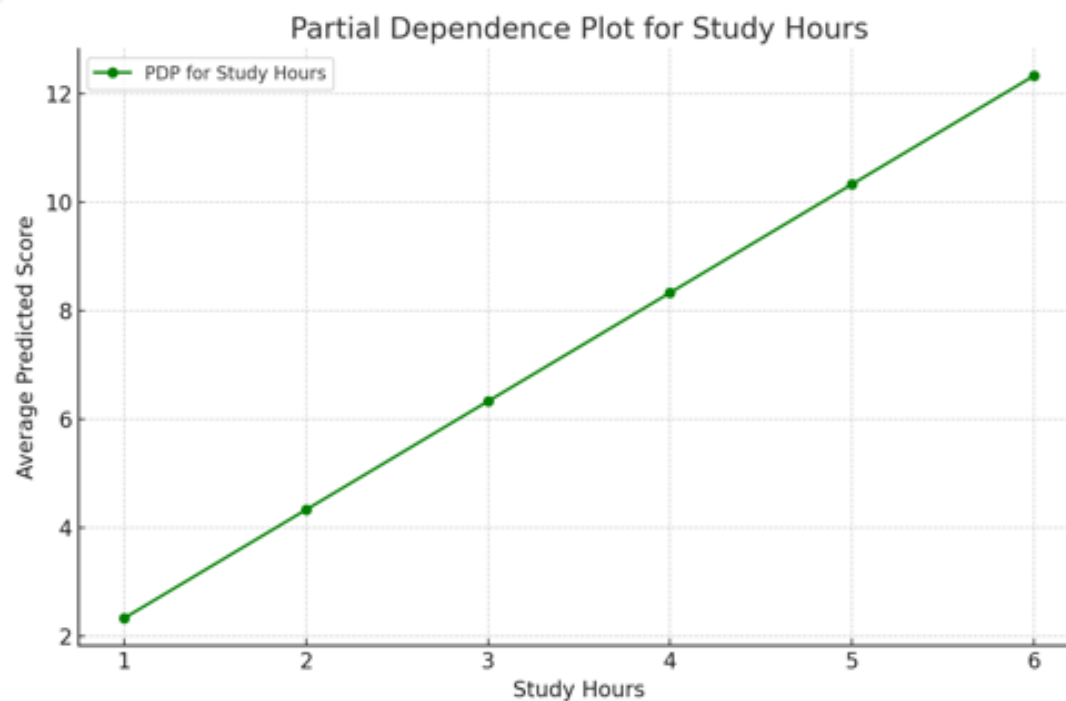


Study hours (x1)	Breaks (x2)	Sleep(x3)	grade
1	2	7	5
2	2	6	6
3	1	7	7
4	1	6	8
5	0	7	9
6	0	5	9

X1	X2	X3	Y_pred
1	2	7	5
1	2	6	4
1	1	7	3
1	1	6	2
1	0	7	1
1	0	5	-1
Average			14/6

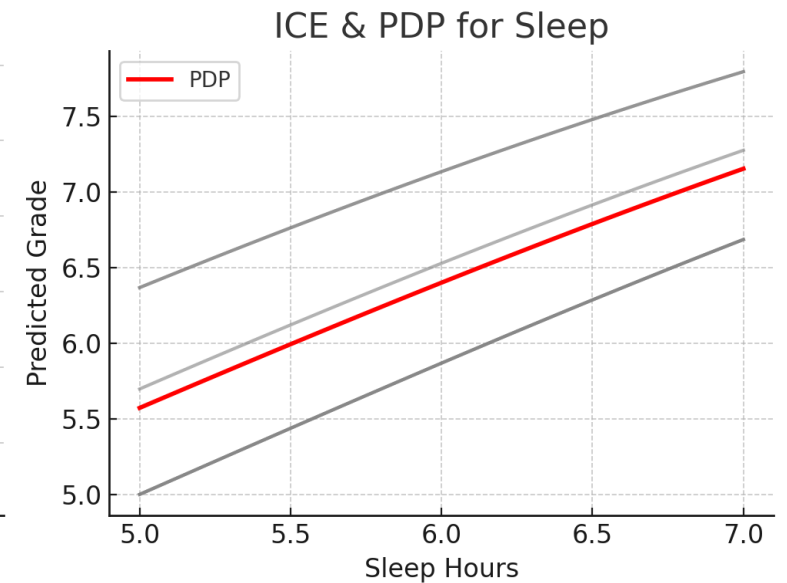
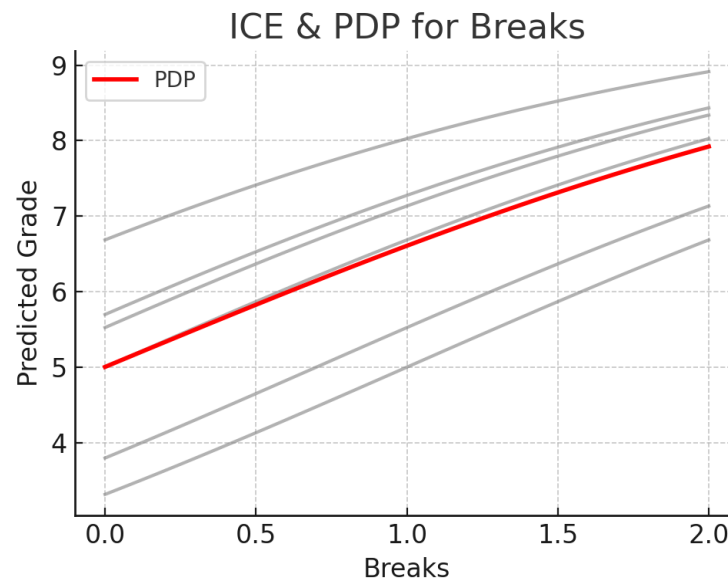
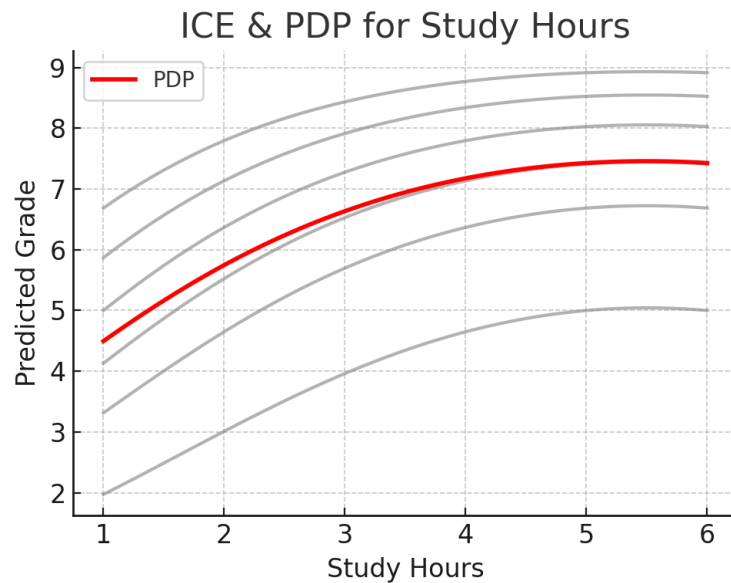
X1	X2	X3	Y_pred
2	2	7	7
2	2	6	6
2	1	7	5
2	1	6	4
2	0	7	3
2	0	5	1
Average			26/6

X1	Y(x1)
1	2.33
2	4.33
3	6.33
4	8.33
5	10.33
6	12.33

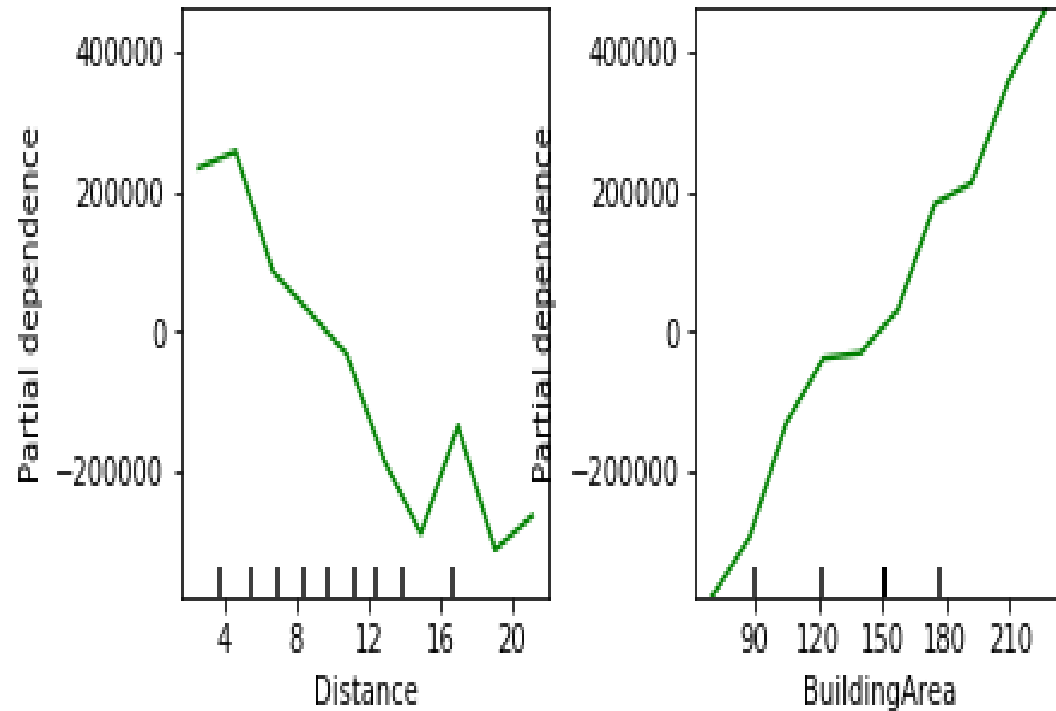


ICE & PDP

- ICE shows **individual variation** (gray curves).
PDP shows the **average relationship** (red).

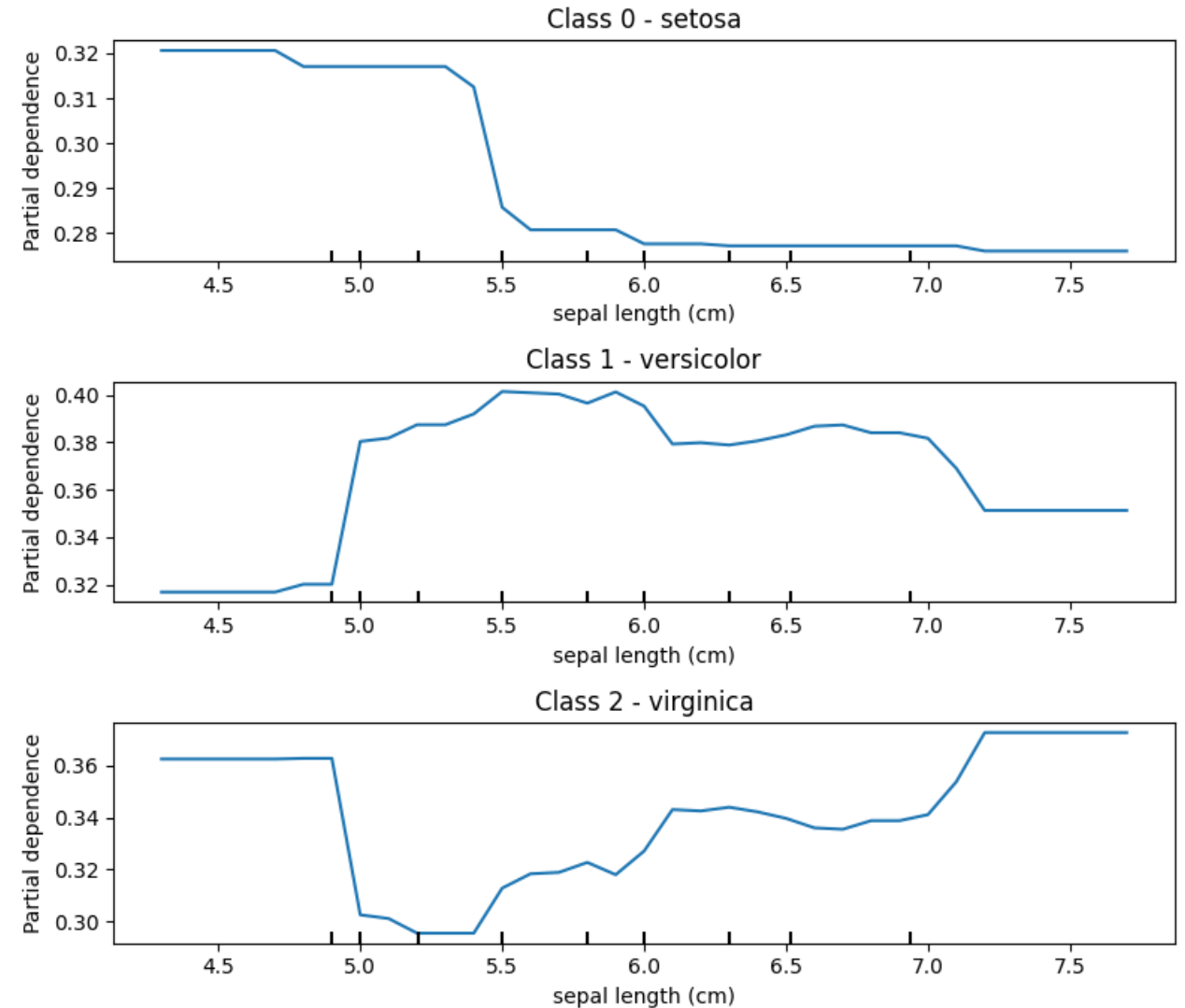


Partial Dependence Plot



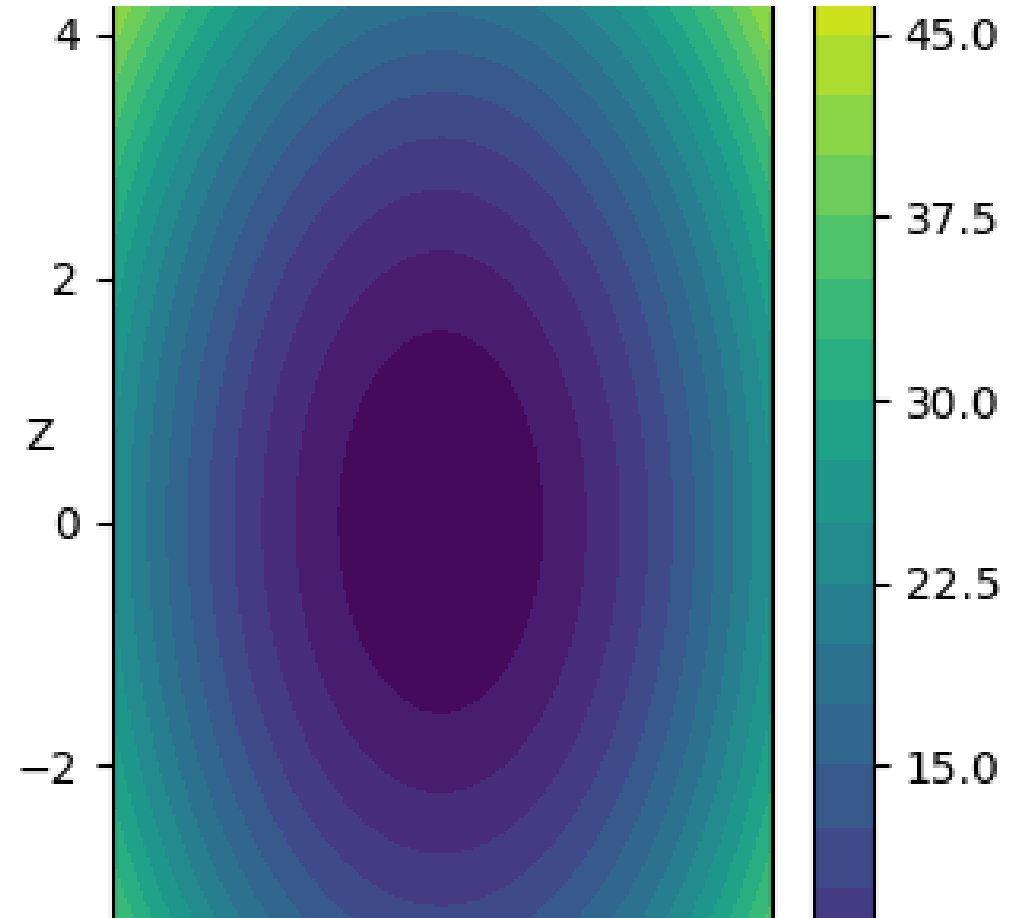
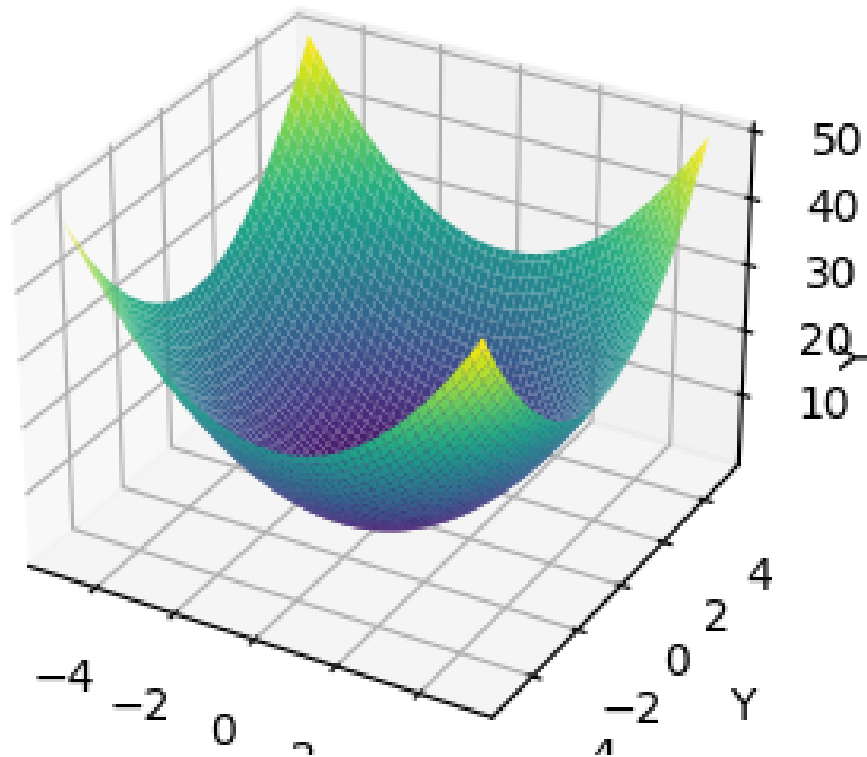
the relationship (according to our model)
between Price and a couple variables from the
Melbourne Housing dataset. [source](#)

DIGITAL



Contour Map

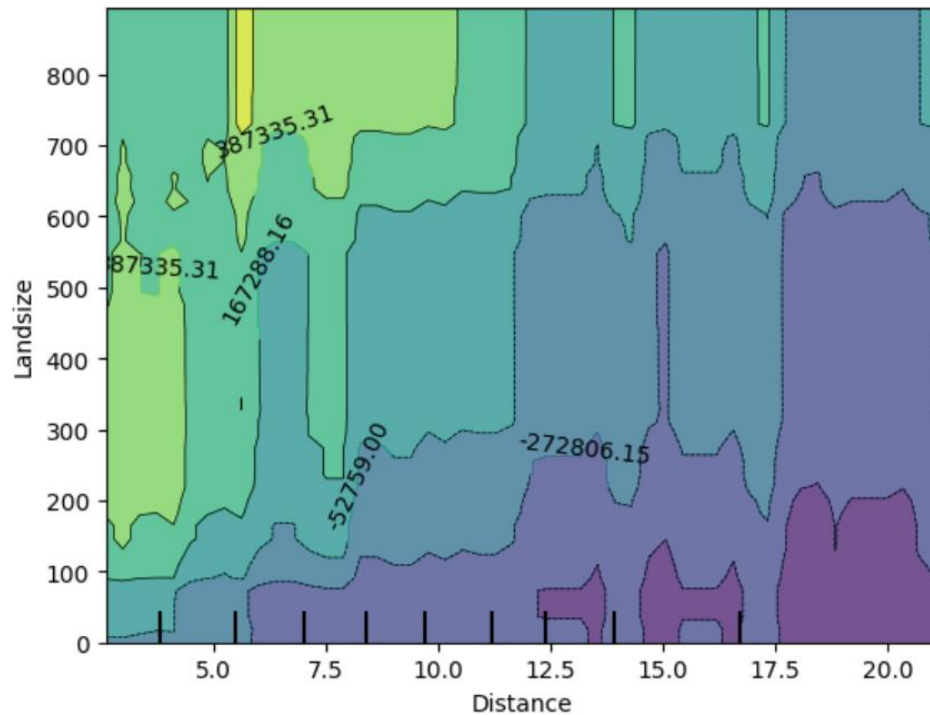
3D surface plot



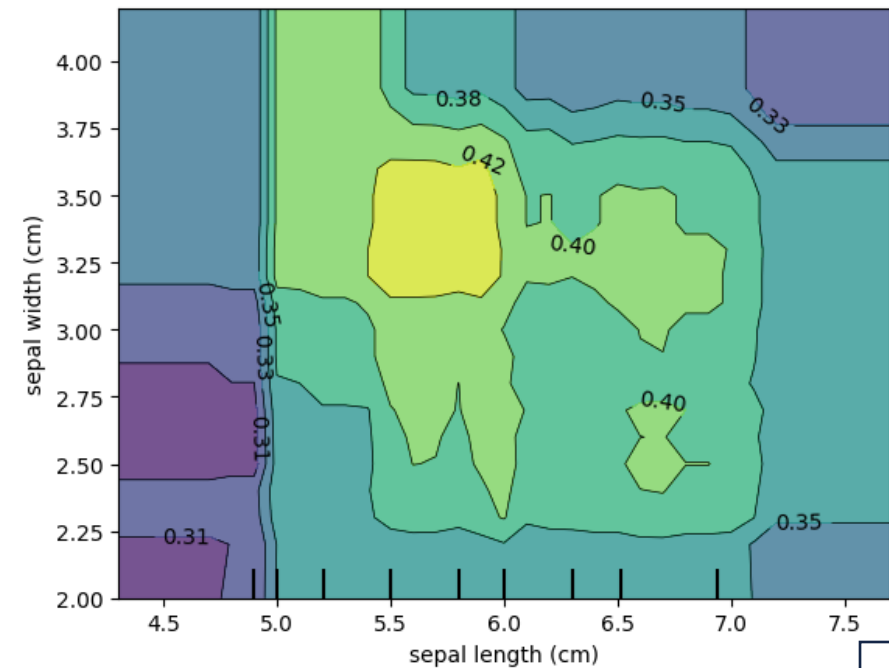
Partial Dependence Plot

- One-way PDPs tell us about the interaction between the target response and an input feature of interest
- Two-way PDPs show the interactions among the two features.

Two-way Partial Dependence Plot of Land Size and Distance



Two-way Partial Dependence Plot of Sepal Length and Sepal Width



[See also](#)



Partial Dependence Plot

Property	Assessment
Completeness	Interpretability achieved with agnostic method, completeness is low, limited possibility of anticipating model predictions (we can just look at goal scored as rough indicator)
Expressive power	Good in terms of getting evidence of the most important feature but on average and without details of feature interactions (or limited)
Translucency	Low, we don't have insight into model internals
Portability	High, the method doesn't rely on the ML model specs
Algorithmic complexity	Low, no need of complex methods to generate explanations
Comprehensibility	Good level of human understandable explanations



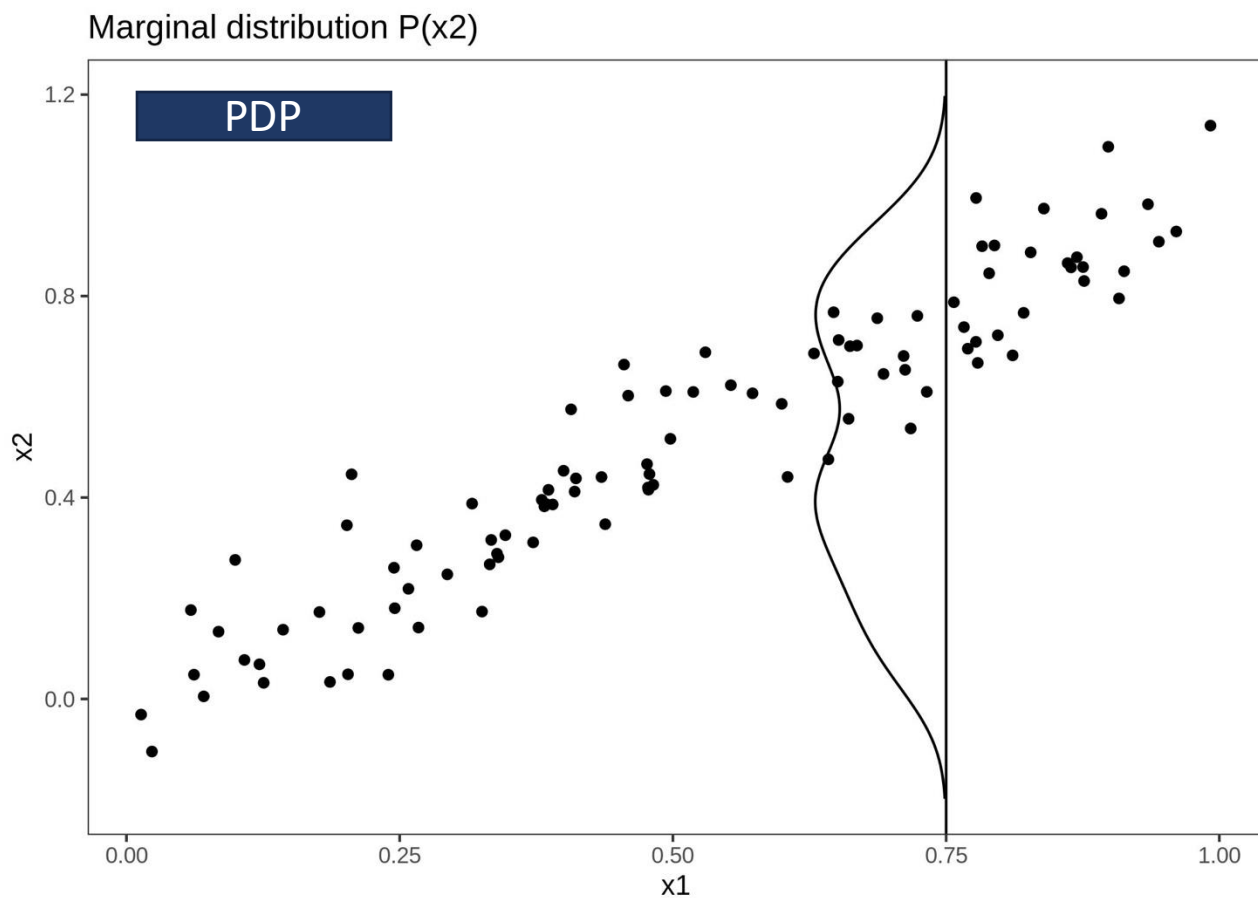
Partial Dependence Plot

- +Computation is intuitive
- +Interpretation is clear (Caution: Uncorrelated)
- +Causal interpretation
- maximum number of features
- Omitting the feature distribution can be misleading
- Assumption of independence
- Heterogeneous effects might be hidden



From PDP to Accumulated Local Effects

What will happen if we have 24 (an unrealistic value) here instead of 7?



Study hours (x1)	Breaks (x2)	Sleep(x3)	grade
1	2	7	5
2	2	6	6
3	1	7	7
4	1	6	8
5	0	7	9
6	0	5	9

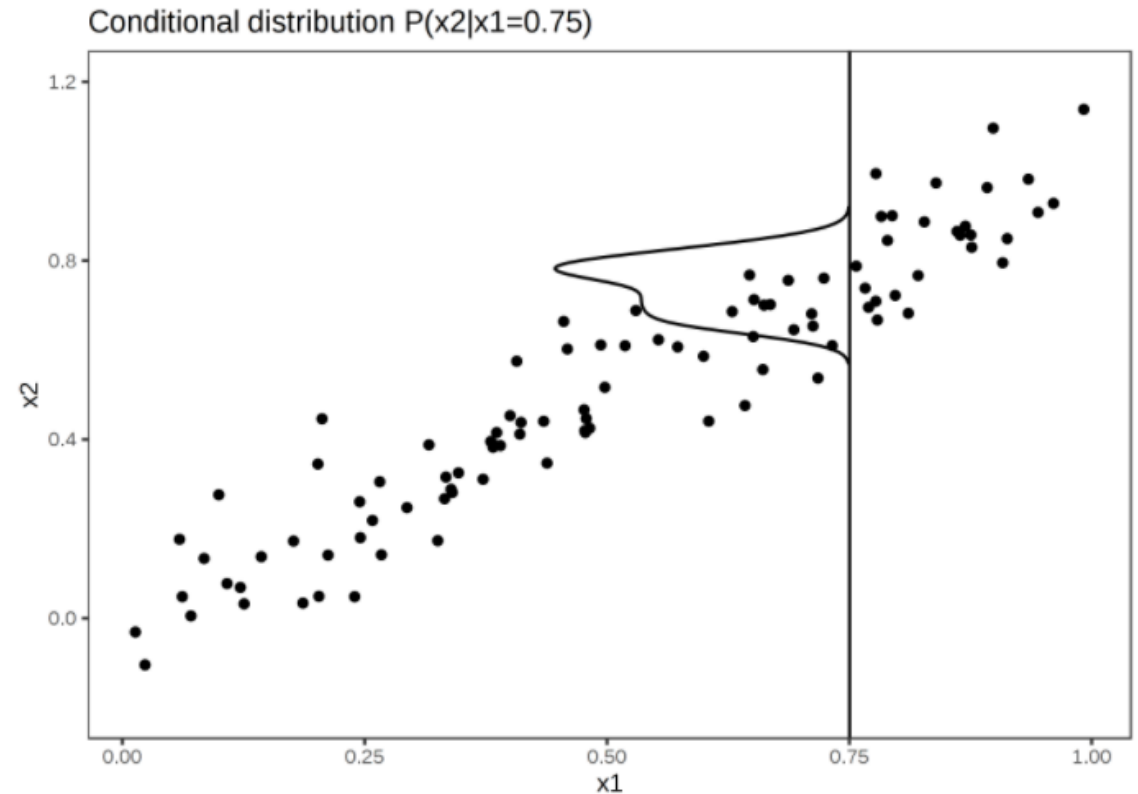
x1	Y(x1)
1	2.33
2	4.33
3	6.33
4	8.33
5	10.33
6	12.33

x1	x2	x3	Y_pred	x1	x2	x3	Y_pred
1	2	7	5	2	2	7	7
1	2	6	4	2	2	6	6
1	1	7	3	2	1	7	5
1	1	6	2	2	1	6	4
1	0	7	1	2	0	7	3
1	0	5	-1	2	0	5	1
Average			14/6	Average			26/6






M-Plot

- Averages predictions over the *conditional distribution* of other features.
- Respects correlations → only considers realistic combinations.
 - At 5 study hours → only averages predictions for students with *observed sleep patterns* (mostly 5–6 hours).
- mixes effects of correlated features → shows *combined effect*.
 - effect of study now partly includes effect of sleep (since they move together).



Accumulated Local Effects

- show how features impact predictions by accumulating the local effects of features across the data distribution.
- Break feature range into small intervals.
- Compute average change in prediction when feature increases within each interval.
- Accumulate these effects across the range.
- Advantages:
 -  Handles correlated features (unlike PDP).
 -  Does not confound effects (unlike M-Plot).
 -  Shows *local* feature effects, centered at 0.



Accumulated Local Effects

- Accumulate difference for each data point where x_{ij} falls within interval K :

$$\tilde{f}_{j,ALE}(x_j) = \sum_{k: z_{k-1,j} \leq x_{ij} \leq z_{k,j}} \Delta \tilde{f}_{i,k}$$

- Adjust ALE to have zero mean across the dataset

$$f_{j,ALE}(x_i) = \tilde{f}_{j,ALE}(x_i) - \frac{1}{N} \sum_{i=1}^N \tilde{f}_{j,ALE}(x_i) \quad N \text{ is the number of instances}$$



Accumulated Local Effects

Accumulate difference for each data point where x_{ij} falls within interval K :

$$\tilde{f}_{j,ALE}(x_j) = \sum_{k: z_{k-1,j} \leq x_{ij} \leq z_{k,j}} \Delta \tilde{f}_{i,k}$$

Adjust ALE to have zero mean across the dataset

$$f_{j,ALE}(x_i) = \tilde{f}_{j,ALE}(x_i) - \frac{1}{N} \sum_{i=1}^N \tilde{f}_{j,ALE}(x_i)$$

N is the number of instances



DIGITAL

Algorithm 1 Accumulated Local Effects (ALE) Plots

Require: Trained prediction model, model

Require: Feature index for ALE plot, feature_index

Require: Dataset containing features and outputs, data

Require: Number of intervals, num_intervals

Ensure: ALE plot of feature x_j

- 1: Calculate quantile bounds for the feature x_j over the specified number of intervals, num_intervals
- 2: Initialize arrays local_effects and all_effects to zeros with length equal to the number of data instances
- 3: **for** $k = 1$ to num_intervals **do**
- 4: Determine bounds $z_{k-1,j}$ and $z_{k,j}$ for the current interval
- 5: Create modified datasets data_lower and data_upper by replacing x_j in all instances with $z_{k-1,j}$ and $z_{k,j}$, respectively
- 6: Compute model predictions for both modified datasets: predictions_lower and predictions_upper
- 7: Calculate differences $\Delta \hat{f}_{i,k} = \hat{f}(z_{k,j}, x_{-j}) - \hat{f}(z_{k-1,j}, x_{-j})$
- 8: **for** each data instance i **do**
- 9: **if** $\text{data}[i, \text{feature_index}] \geq z_{k-1,j}$ and $\text{data}[i, \text{feature_index}] < z_{k,j}$ **then**
- 10: Accumulate effects: $\text{local_effects}[i] += \Delta \hat{f}_{i,k}$
- 11: **end if**
- 12: **end for**
- 13: **end for**
- 14: Calculate the mean of local_effects:

$$\text{mean_effect} = \frac{1}{N} \sum_{i=1}^N \text{local_effects}[i]$$

- 15: Adjust each element in local_effects by subtracting the mean effect:
 $\text{all_effects}[i] = \text{local_effects}[i] - \text{mean_effect}$
 - 16: Plot all_effects against feature x_j values to visualize the ALE plot
-

Accumulated Local Effects

Sample Data			
age	bmi	heart_disease	P of stroke
2	12	0	20
3	15	0	21
6	11	0	20
22	24	0	30
24	21	0	31
27	24	0	29
45	23	0	40
43	25	0	41
47	25	0	45
66	30	1	93
68	28	1	88
63	29	1	95

- **Data Type:**

- **Numerical Features:** ALE is calculated by dividing the feature into intervals, computing prediction differences for small changes within these intervals, and accumulating these to get the ALE curve.
- **Categorical Features:** Special methods like ordering categories based on similarity (using metrics like the Kolmogorov-Smirnov distance) are required since categorical data doesn't naturally fit into intervals.



Sample Data			
age	bmi	heart_disease	P of stroke
2	12	0	20
3	15	0	21
6	11	0	20
22	24	0	30
24	21	0	31
27	24	0	29
45	23	0	40
43	25	0	41
47	25	0	45
66	30	1	93
68	28	1	88
63	29	1	95

Age 3			
Age interval 2-6 (Lower)			
age	bmi	heart_disease	P of stroke
2	12	0	20
2	15	0	22
2	11	0	21
Average P			21

Age 3			
Age interval 2-6 (Upper)			
age	bmi	heart_disease	P of stroke
6	12	0	22
6	15	0	23
6	11	0	20
Average P			22

Difference	
2	
1	
-1	
0.67	Average Diff.

Age 24			
Age interval 22-27 (Lower)			
age	bmi	heart_disease	P of stroke
22	24	0	30
22	21	0	29
22	22	0	27
Average P			29

Age 24			
Age interval 22-27 (Upper)			
age	bmi	heart_disease	P of stroke
27	24	0	31
27	21	0	29
27	22	0	29
Average P			30

Difference	
1	
0	
2	
1	Average Diff.

Age 45			
Age interval 43-47 (Lower)			
age	bmi	heart_disease	P of stroke
43	23	0	40
43	25	0	42
43	25	0	44
Average P			42

Age 45			
Age interval 43-47 (Upper)			
age	bmi	heart_disease	P of stroke
47	23	0	42
47	25	0	44
47	25	0	45
Average P			44

Difference	
2	
2	
1	
1.67	Average Diff.

Age 66			
Age interval 63-68 (Lower)			
age	bmi	heart_disease	P of stroke
63	30	1	93
63	28	1	87
63	29	1	94
Average P			91

Age 66			
Age interval 63-68 (Upper)			
age	bmi	heart_disease	P of stroke
68	30	1	96
68	28	1	90
68	29	1	95
Average P			94

Difference	
3	
3	
1	
2.33	Average Diff.

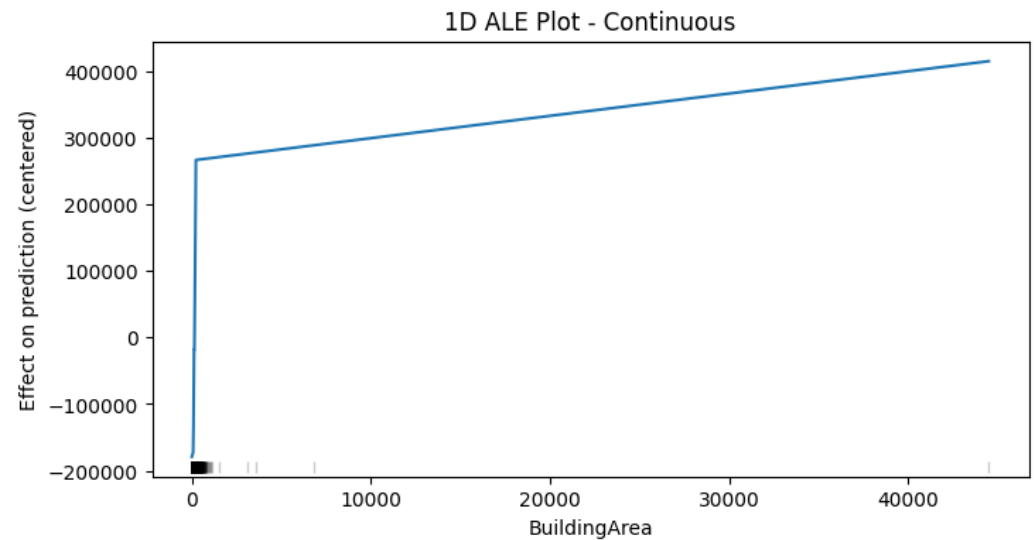
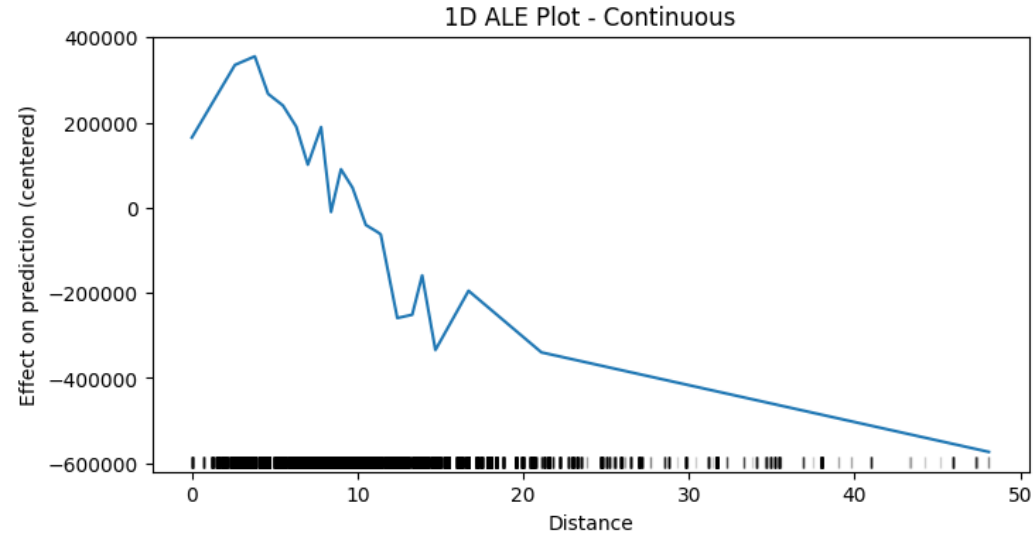
5.67	Accum Diff.
------	-------------

0.5	Accum Diff / N
-----	----------------

*Average Prediction is rounded to the nearest integer value



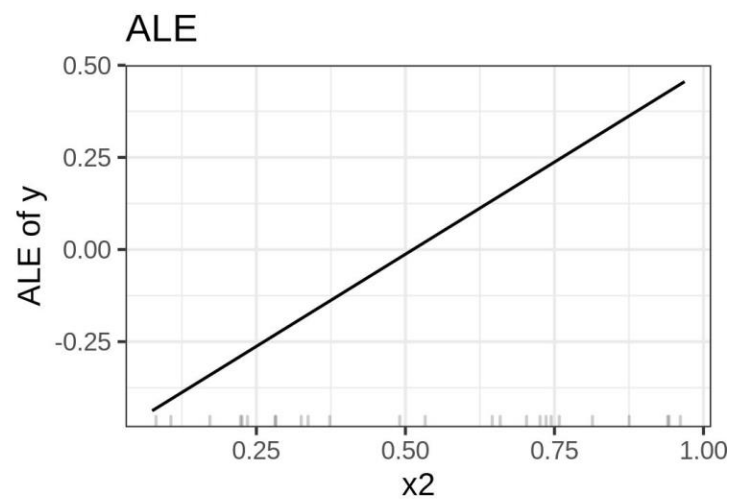
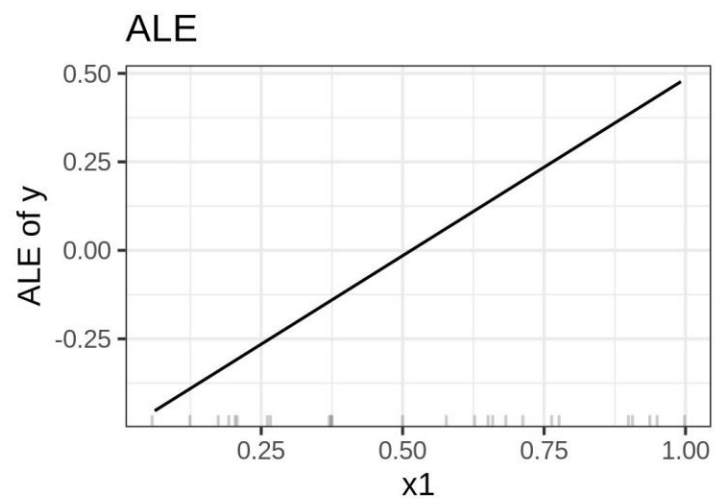
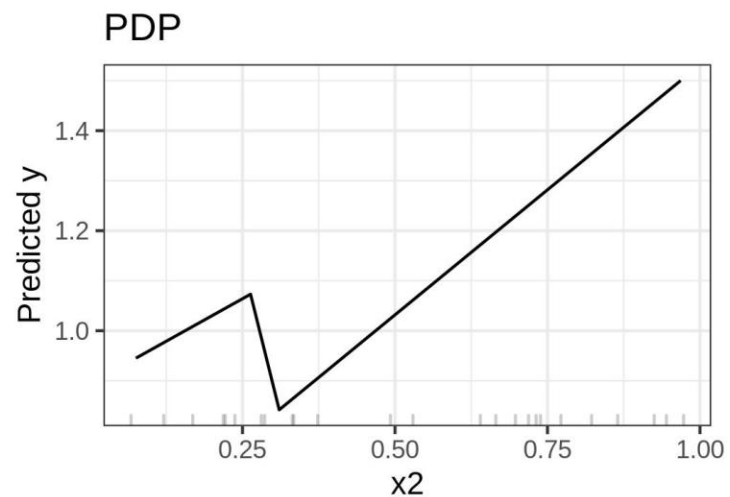
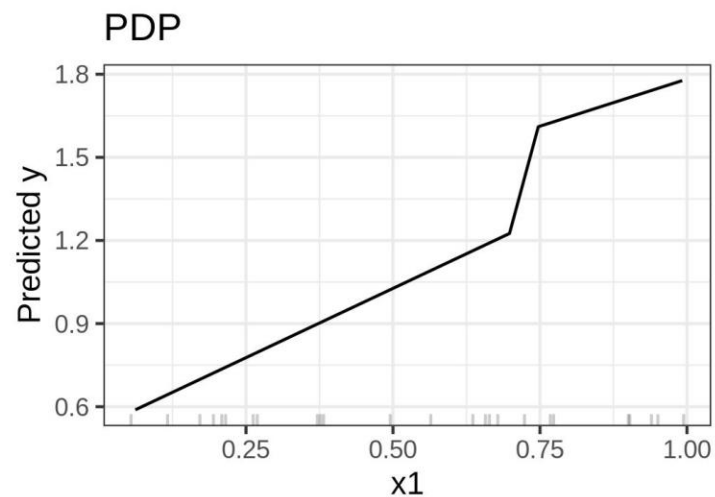
ALE Example



- **Limitations**

- **Computational Complexity:** ALE plots require significant computational resources, particularly with large datasets or many feature intervals.
- **Interpretation Challenges:** Interpreting the results of ALE plots can be difficult, especially in complex, high-dimensional models.





PDP vs ALE

SOURCE:
[HTTPS://CHRISTOPHM.GITHUB.IO/INTERPRETABLE-ML-BOOK/ALE.HTML](https://christophm.github.io/interpretable-ml-book/aale.html)

Interpretability and Understandability

- 📌 The debate on AI explainability is not new – it dates back to expert systems in the 1980s.
- 📌 In recent years, discussions have broadened: interpretability, frameworks, and input from social sciences.
- 📌 There is no single, universal definition of “*explanation*” in AI.
- 📌 Most approaches focus on making results **human-understandable**, rather than defining explanation formally.

Reference:

Hamon, Ronan, Henrik Junklewitz, and Ignacio Sanchez. *Robustness and explainability of artificial intelligence*. Publications Office of the European Union 207 (2020).







When is Interpretability Needed?

Not all ML systems require interpretability.

-  Example: movie recommendations, ad placement.

Interpretability is **critical in high-stakes domains**:

-  Healthcare (diagnosis, treatment decisions)
-  Autonomous driving (safety-critical decisions)
-  Finance (credit scoring, fraud detection)
-  Legal & regulatory settings (compliance, fairness)

 **Message:** Use interpretability when ML decisions directly affect people's lives, safety, or rights.

[It is a legal requirement!!!](#)





DIGITAL



**Funded by
the European Union**

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Horizon Europe: Marie Skłodowska-Curie Actions. Neither the European Union nor the granting authority can be held responsible for them.



DIGITAL

This project has received funding from the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101119635