# Post-modeling Explainability

Explainable and Trustworthy AI

Eliana Pastor

# Stages of Explainability

- Explainability involves the entire AI development pipeline

**Pre-modelling explainability**

Before building the model

- Data exploration
- Data selection
- Feature engineering

**Explainable modeling**

Build inherently interpretable models

- Manage the accuracy and interpretability trade-off

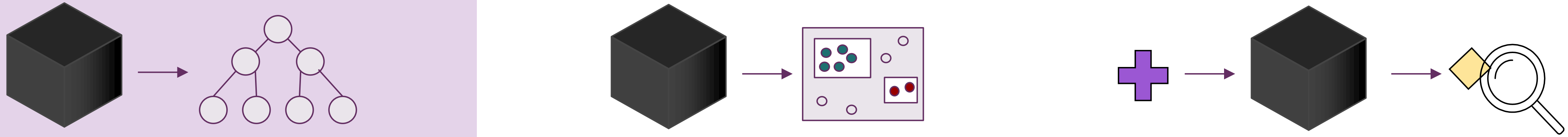
**Post-modelling explainability**

After model development

- Explaining predictions and behavior of trained models

# Scope of Explainability

- *What do we explain?*



**Global**

**Subgroup**

**Individual/local**

How the model globally works

How the model behaves in data subgroups

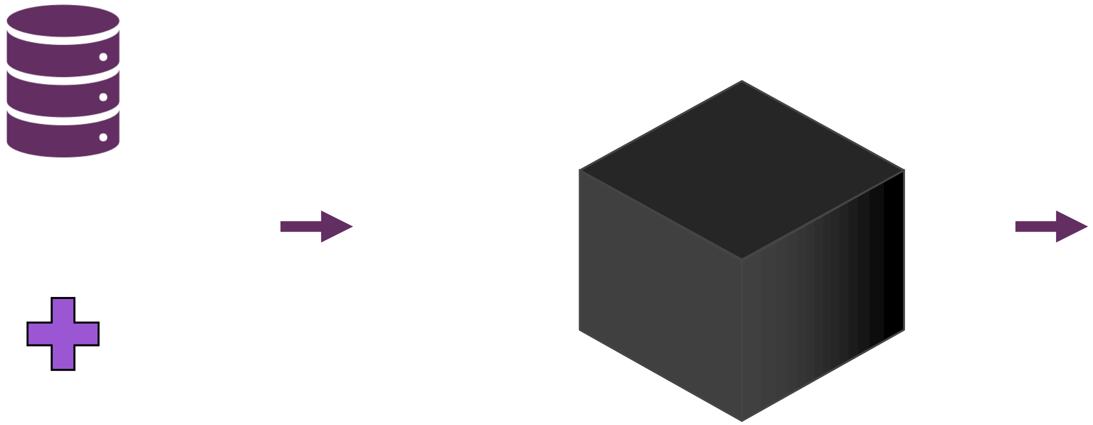
Explaining the reasons behind individual predictions

# Generalizability of Explainability

- **Model dependent solutions**
  - Only applicable for specific models
    - e.g., specific approaches for explaining SVM, approaches for explaining a specific neural network
  - Relies on the model structure/properties

- **Model agnostic solutions**
  - Applicable to any model
  - Relies on the model as an oracle (model predictions, output probabilities)

# Model agnostic solution



**Output**
Prediction
Prediction probabilities
…

# Advantages of model agnostic solutions

- **Model Flexibility/Compatibility**
  - Explain complex and high-performing model
  - Model agnostic methods can be used across different frameworks and libraries
- **Explanation Flexibility**
  - Adopt the explanation representation/format more suitable for target users and domains
- **Representation Flexibility**
  - The representation used for the explanations (e.g., patches of the image, set of words) can differ from the ones used by the models (e.g., pixels, embeddings)
- **Lower Cost to Switch**
  - We can change the underlying model while preserving the explanation representation
- **Model comparison**
  - Easier to compare models if the explanation representation is the same

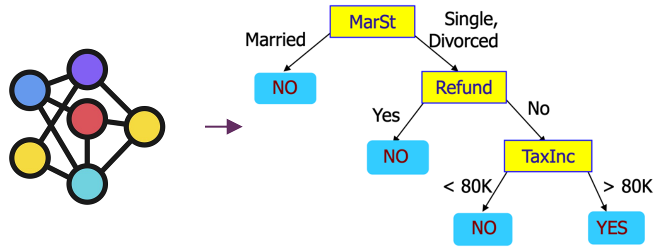
Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Model-agnostic interpretability of machine learning." arXiv preprint (2016).

# Scope of Explainability – Global explanations

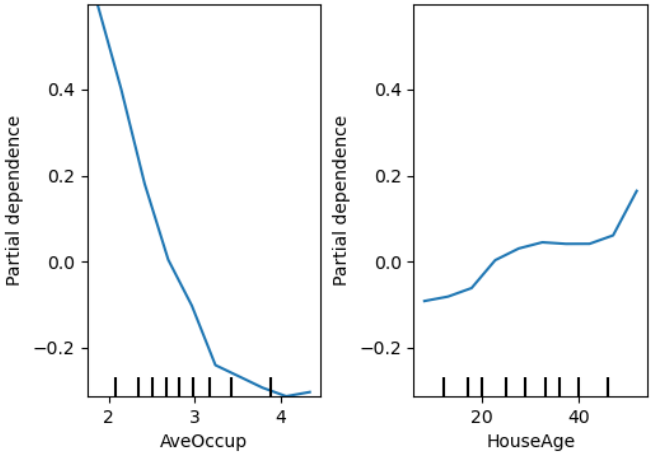
Global methods describe the overall behavior of model

Explain how the model works in general

**Interpretable global surrogate models**

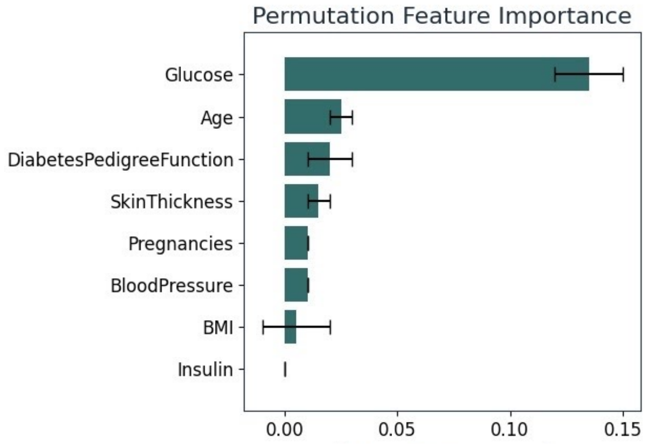



**Partial dependence plots**




Dependence between the target response and an input feature of interest

**Permutation feature importance.**



# Global surrogate model

- Interpretable surrogate or proxy for a complex (black-box) model
  - Trained to approximate the predictions of the black box model
  - Simplified and understandable representation

Goal

- Approximate the black box prediction function f with a surrogate model prediction function g. The surrogate model g should closely mimic the behavior of f, under the constraint that g is interpretable
  - g: decision tree, logistic regression, linear regression, rules

# Global surrogate model – Steps

- **Training data** X. The dataset can be the same as the one used to train the black-box model f or a new dataset reflecting its distribution
- **Labeling**. For dataset X, get the predictions of the black box model F
- **Interpretable model g**. Choose an interpretable model type that best suits the problem domain and requirements.
- **Surrogate training**. Train the interpretable model using the dataset X and the predictions of f
  - The interpretable model learns to approximate the behavior of the f
- **Evaluation.** Measure how well the surrogate model replicates the predictions of the black-box model using appropriate evaluation metrics.
  - Mean squared error (MSE), accuracy, or AUC-ROC
- **Interpretation**. Interpret the surrogate model to gain insights into its decision-making

https://christophm.github.io/interpretable-ml-book/global.html

# Global surrogate model

- Multiple variations and optimization proposed, e.g.,:
- **TREPAN**
  - Use **tree** as interpretable model
  - Consider fidelity to the original model in the tree construction process
  - Best first expansion
    - Prioritize the nodes with the greatest potential to increase the fidelity of the extracted tree to the model
      - Evaluation of node n = reach(n) x (1 - fidelity(n))
        - **reach**(n) is the estimated fraction of instances that reach n when passed through the tree
        - **fidelity**(n) is the estimated fidelity of the tree to the network for those instances

# Advantages of Global surrogate models

- Provide a simplified representation of complex models
- Different forms of explanations, depending on the interpretable model adopted
  - Can enable both global and local explainability
- Easy to build
- Model agnostic, in terms of both
  - Model to explain
  - Surrogate model adopted
    - Flexibility on the choice of interpretable model g

# Limitations of Global surrogate models

- **Approximation**. The surrogate model is an approximation of the complex model, and there might be cases where it fails to capture its complexity
- **Oversimplification**. Simplifying a complex model inherently involves some level of abstraction, and there's a risk of oversimplifying the decision boundaries,
- **Global behavior.** Global surrogate models provide an overall view of the model's behavior, but they may not capture local nuances or specific decision boundaries of the complex model
- **Data dependence.** The quality of the surrogate model heavily relies on the quality and representativeness of the training data used
- **Interpretability.** The surrogate model can be still difficult to interpret

# Permutation feature importance

Estimate the importance of features for a model.

- It evaluate the impact of permuting (randomly shuffling) the values of individual features on the model's performance.
- By measuring how much the model's performance decreases after permuting a specific feature, one can infer the importance of that feature in making accurate predictions.
- The higher the change, the more the feature is importance

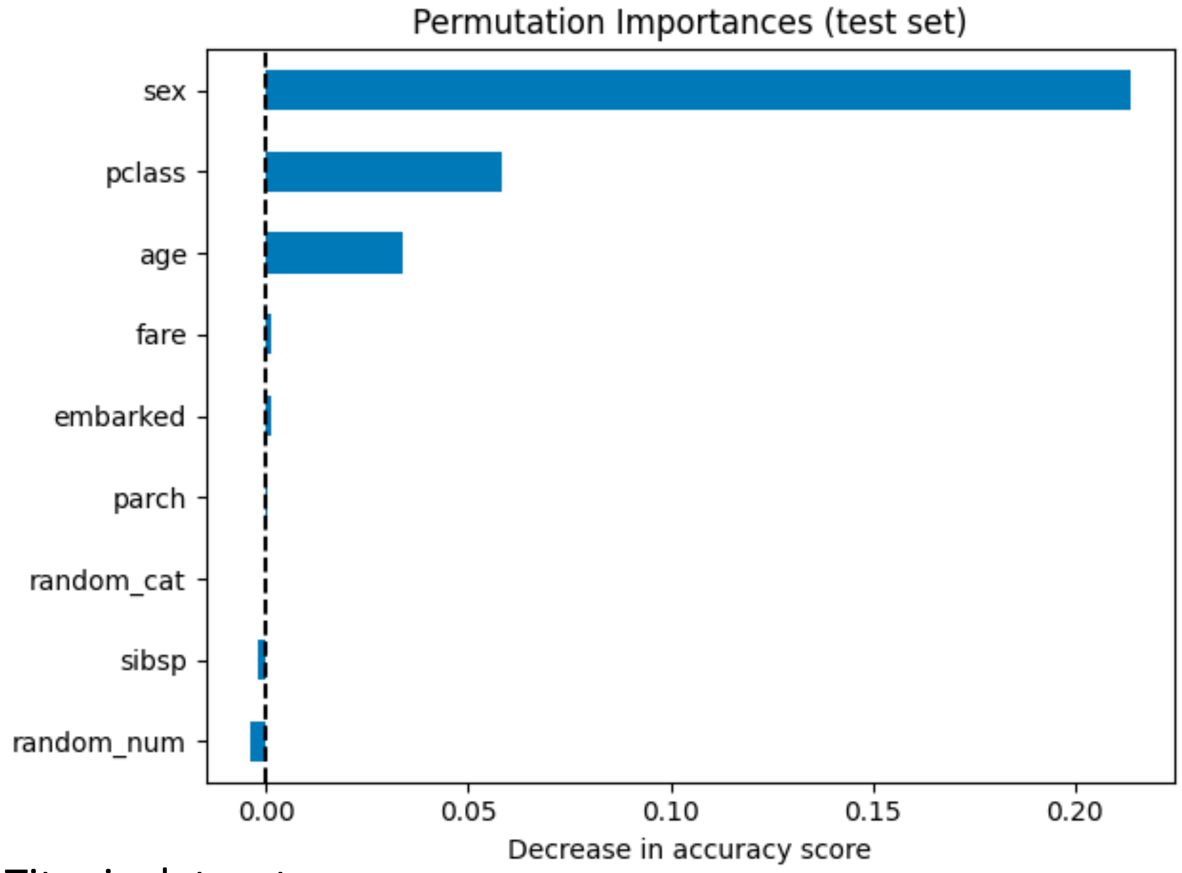
# Permutation feature importance

- Compute the reference score, e.g., accuracy of the model on dataset D
- For each feature
  - **Permute the feature**. Randomly shuffle the values of the feature across D
  - **Evaluate model performance**. Apply the model on the dataset with the permuted feature and record the performance metric.
  - **Compute the importance Score**. The importance score for the feature is the difference (or the ratio) between the original performance metric (the reference) and the performance metric after permuting the feature.
- **Rank features**. Rank the features based on their importance scores. The higher the drop in performance when a feature is permuted, the more the feature is important

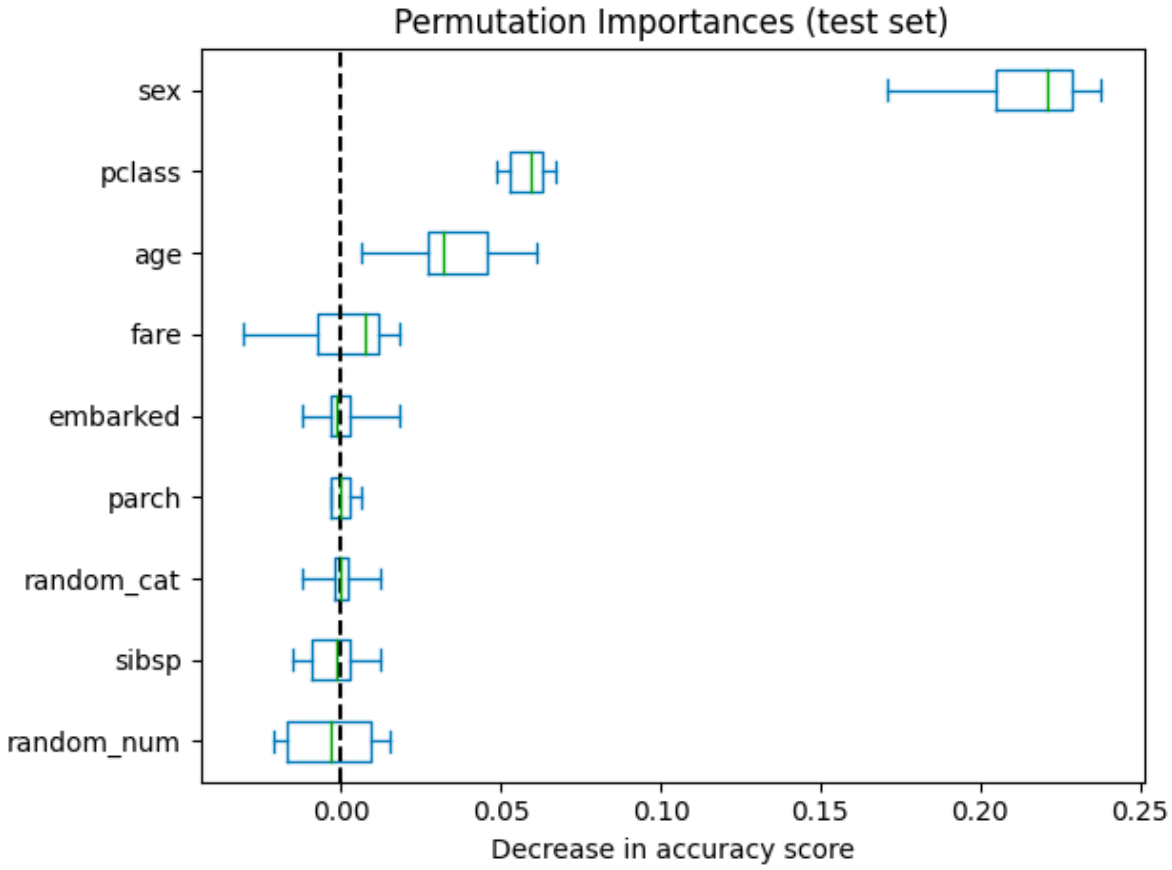
- Typically, the permutations and the scores are computed N times, to account for the randomness of the process, improving stability of the results

# Permutation feature importance

Bar plot (mean)




Box plot




Titanic dataset

# Advantages of Permutation feature importance

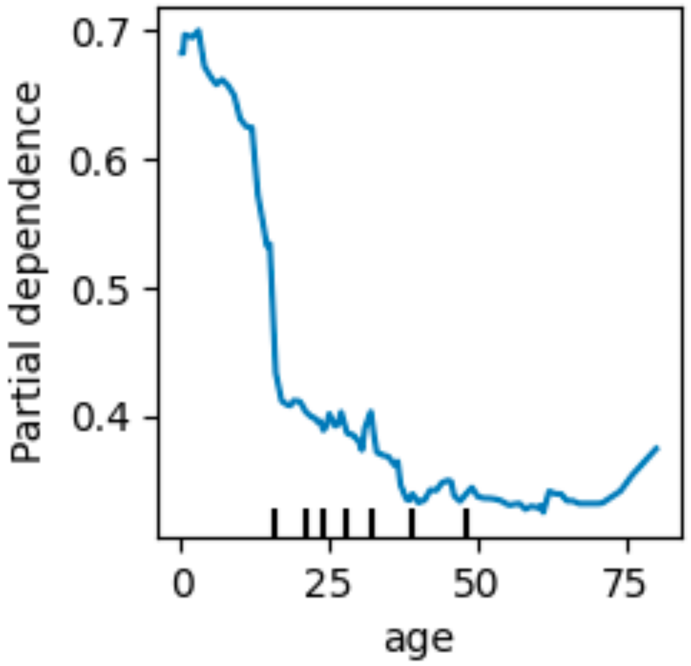
- Model-Agnostic
- Intuitive interpretation of feature importance
- Provide compressed, global insight into the model behavior
- Simple Implementation
- No Assumption of Linearity in the relationship between features and the target. It can capture complex, non-linear relationships in the data.
- Performance ratio (compared to the performance difference) enable to compare importance across different models and problems.
- Does not require retraining the model

# Limitations of Permutation feature importance

- **Feature Independence Assumption**. If features are correlated,
  - it can be biased by unrealistic data instances.
    - e.g., person of 1.8m and 20kg
  - The importance of correlated features decrease (shared), as when permuting one, the model still has access to the other
    - Risk in the interpretation and if used for feature selections
- **Linked to the model performance**. Other measures (e.g., model variance explained by the features) can be of interest
- **Require the ground truth**
- **Randomicity.** Depends on feature shuffling, randomness to the measurement
  - Repeating the process stabilizes the measure, but increases the computational time

# Partial dependence plots

- Partial Dependence Plots (PDPs) are a **visualization** tool used to understand the relationship between a model predictions and specific input variables.
  - It show the dependence between the target outcome and a set of input features of interest, marginalizing over the values of all other input features
- PDPs reveal how changes in individual features influence the target response as a function of the input features of interest.
- Typically, we analyze one (or two) feature at a time, due to the limits of human perception




Titanic dataset – class 1

# Partial dependence plots - Definition

- $X_S$ = features of interest
- $X_C$ = other features ($X_S$ complement)

Partial dependence is computed by marginalizing the output over the other features C, so that the function shows the relationship between the features in set S and the outcome.

The partial dependence of model $f$ at a point $x_S$ is:

$$pd_{x_S}(x_S) = \mathbb{E}_{X_C}[f(x_S, X_C)] = \int f(x_S, x_C)dP(x_C)$$

where $f(x_S, x_C)$ is the model outcome (e.g., prediction probability) for a sample whose values are defined by $x_S$ for the features in $X_S$, and by $x_C$ for the features in $X_C$. $dP(x_C)$ marginal distribution. By marginalizing over the other features, we get a function that depends only on features in S.

We compute $pd_{x_S}(x_S)$ for different $x_S$ and we plot it

# Partial dependence plots - Computation

- $x_S$ = feature value(s) for $X_S$, features for which the partial dependence function is plotted, typically 1 or 2

The partial dependence of $x_S$ is computed as an average over the data X:

$$pd_{x_s}(x_S) \approx \frac{1}{n}\sum_{i=1}^{n} f(x_s, x_C^{(i)})$$

- $x_C^{(i)}$ is the value of the i-th sample for the features in $X_C$.
- $n$ is the number of instances in the dataset
- $x_S, x_C$ represents total feature space

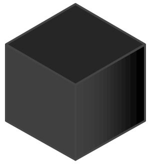
The partial function tells us for given value(s) of features S what the average marginal effect on the prediction is.

# Partial dependence plots - Computation

Example

| age | pclass | sex |
|---|---|---|
| 40 | 1 | female |
| 35 | 2 | male |
| 50 | 2 | female |
| 20 | 3 | male |

**pd(Age=10)**

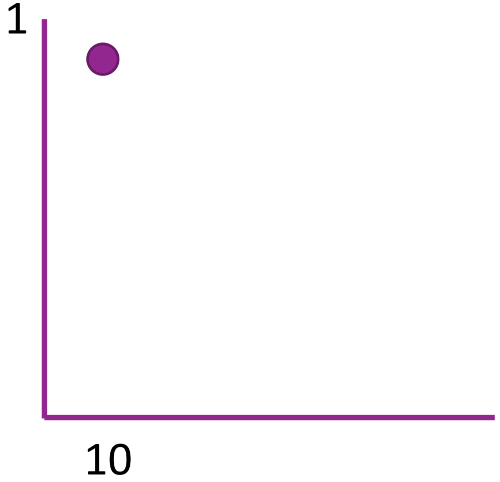


| age | pclass | sex |
|---|---|---|
| 10 | 1 | female |
| 10 | 2 | male |
| 10 | 2 | female |
| 10 | 3 | male |

**P(class=1)**

| p |
|---|
| 0.99 |
| 0.95 |
| 0.95 |
| 0.9 |

$\rightarrow \sum \rightarrow$ 0.9475

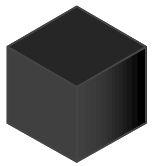


# Partial dependence plots - Computation

Example

| age | pclass | sex |
|---|---|---|
| 40 | 1 | female |
| 35 | 2 | male |
| 50 | 2 | female |
| 20 | 3 | male |

**pd(Age=20)**

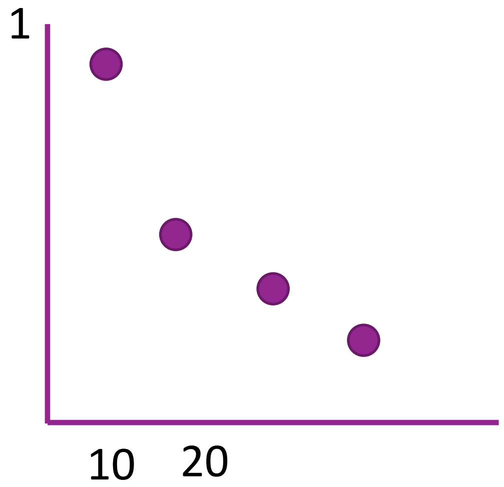


| age | pclass | sex |
|---|---|---|
| 20 | 1 | female |
| 20 | 2 | male |
| 20 | 2 | female |
| 20 | 3 | male |

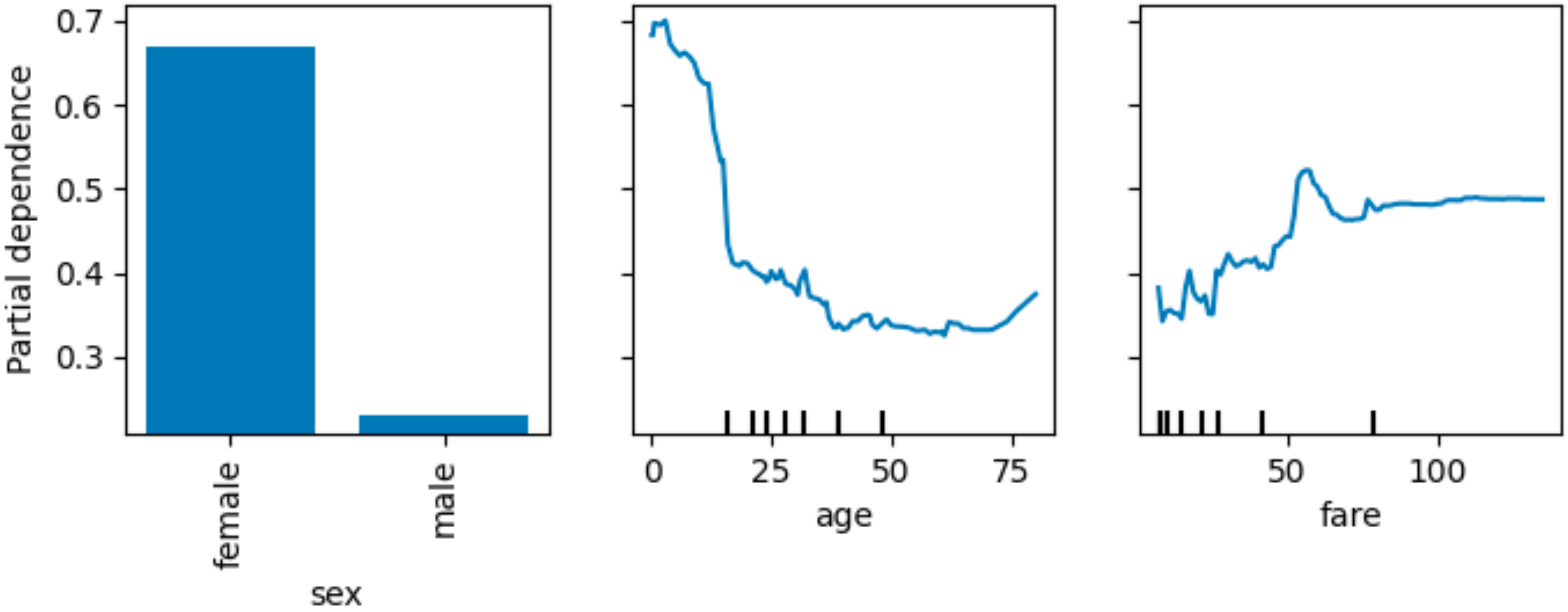
**P(class=1)**

| p |
|---|
| 0.9 |
| 0.4 |
| 0.7 |
| 0.1 |

$\rightarrow \sum \rightarrow$ 0.525



# Partial dependence plots




Titanic dataset – class 1

# Advantages of Partial dependence plots

- The computation of partial dependence plots is intuitive
  - The PDP at a feature value $x_S$ is the average prediction if we force all data points to assume that feature value

- Explanation in visualization form, easy to inspect
  - The PDP shows how the average prediction in your dataset changes when the feature S is changed.

- Easy to implement

# Limitations of Partial dependence plots

- **Independence Assumption**
  - PDPs assume independence between the inspect feature and others, not correlated features
  - If correlated features, we may create of unrealistic data

- Typically analyze **one feature at a time**

- PDP typically do not show the feature distribution. The risk is to overinterpret regions with almost no data

- The average marginal effect may hide heterogeneous effects
  - E.g., counterbalances of positive and negative effect

# References


- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Model-agnostic interpretability of machine learning." arXiv preprint (2016).
- Molnar, Christoph. *Interpretable machine learning* https://christophm.github.io/interpretable-ml-book/
- Craven, Mark, and Jude Shavlik. "Extracting tree-structured representations of trained networks." *Advances in neural information processing systems* 8 (1995)
- https://scikit-learn.org/stable/modules/partial_dependence.html#partial-dependence
- T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning, Second Edition, Section 10.13.2, Springer, 2009.
- Breiman, Leo."Random Forests." Machine Learning 45 (1). Springer: 5-32 (2001).
- 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).