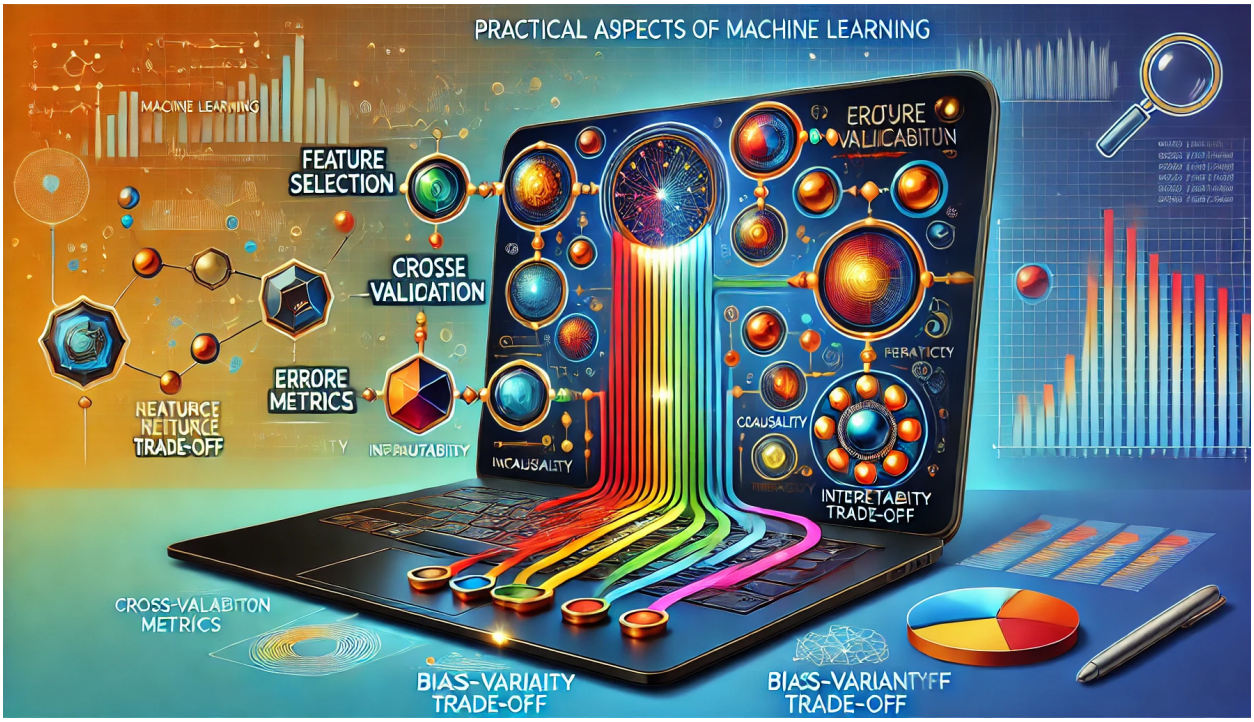
ML in Practice

Mark Asch - ML-PREP

2025



Practical ML

Program

- Method choice: which method should I use for my problem?

  $\Rightarrow$ Bias-Variance tradeoff.
  $\Rightarrow$ Interpretability and explainability.

- EDA (Exploratory Data Analysis): how to use unsupervised learning as an initial phase?

- Model tuning and validation: how can I be sure of the accuracy and robustness of my model?

- Evaluation metrics and predictive power: which metrics measure the predictive power best?

  $\Rightarrow$ Dangers of R^2 and p-values.

- Causality vs Correlation: how to distinguish between them?

$\Rightarrow$ Shapley values

- Feature selection, feature importance and feature engineering.
- PINN (Physics Informed Neural Networks): when and how should they be incorporated?
- Ethics and bias.

PRINCIPLES

The 3 Principles of Coding

1. Coding is Understanding
   (a) if you don't code, you don't/won't understand (coding helps to understand, good code enables good understanding)
   (b) if you understand, then you can code (coding requires perfect undertanding)

2. Coding is Learning
   (a) if you learn well, then you can code well
   (b) if you code well, then you'll learn (from/with the code)

3. Coding is Tennis
   (a) you cannot code by just watching
   (b) if you code, then you're a player (in the game of life/science/research/industry/etc.), otherwise you will remain a spectator...

3 Principles of Coding

- Coding is Understanding
- Coding is Learning
- Coding is Tennis

PYTHON

Basic Python Installation

- We recommend the use of the conda environment.
- If not already installed on your laptop, please download a version (mini or full) from the conda website—see the links on the CSU course software webpage.

Conda is an open-source package management system and environment management system that runs on Windows, macOS, and Linux. Conda quickly installs, runs, and updates packages and their dependencies. Conda easily creates, saves, loads, and switches between environments on your local computer.

Python/Conda Environments - Machine Learning

Note

For the duration of the training, we will require a number of specific python packages. For this, we will create and use tailor-made conda environments.

```
conda create -n prep_ml python=3
conda activate prep_ml
conda install jupyterlab numpy matplotlib pandas
conda install scikit-learn
conda install pytorch torchvision -c pytorch
pip install islp
.
.
jupyter notebook
.
.
conda deactivate
conda env list
```

Test Python Installation

Note

Before starting any of the labs, we must test our python environment.

- Launch a jupyter notebook.
- Test pytorch:

```
import torch
x = torch.rand(5, 3)
print(x)
```

- Test sklearn:

```
import sklearn
print(sklearn.__version__)
```

Writing python code

- Here, we will use Jupyter Notebooks, which are very good for communicating results and research.

- Real coders, however, use far more sophisticated code editing applications, such as:

  ⇒ Visual Studio - developed by Microsoft, but freely available. Open-source version available (see References below).
  ⇒ PyCharm - for professional developers.
  ⇒ Spyder - comes with the anaconda bundle.

- Note that chatGPT and other generative transformers, can be used as a coding assistant - but **beware**, the code generated should be thoroughly debugged and tested before use! There are very often serious errors and bugs in the codes generated, except in the simplest cases...

ML FRAMEWORK

Mathematical Formulation

- Suppose that we have:

  $\Rightarrow$ a response variable (to explain), Y,
  $\Rightarrow$ p explanatory[1] variables, $X = (X_1, X_2, \ldots, X_p)$,
  $\Rightarrow$ n samples of data, giving an $(n \times p)$ matrix,

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & & \\ \vdots & & \ddots & \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

  $\Rightarrow$ a relationship between Y and X of the form

$$Y = f(X) + \epsilon$$

  $\Rightarrow$ where
    $\rightarrow$ f is an unknown function of (columns) $X_1, X_2, \ldots, X_p$

[1] Also called: features, attributes, covariates

→ ϵ is a random error term, independent of X, and with zero mean

- ML is then an ensemble of approaches for estimating the unknown function f with the objectives of

⇒ Prediction: $\hat{Y} = \hat{f}(X)$ where $\hat{f}$ is an estimation for f and $\hat{Y}$ is the resulting prediction

⇒ Inference: to understand how Y varies as a function of X (correlations, importances, linearity, etc.)

Initial Categories

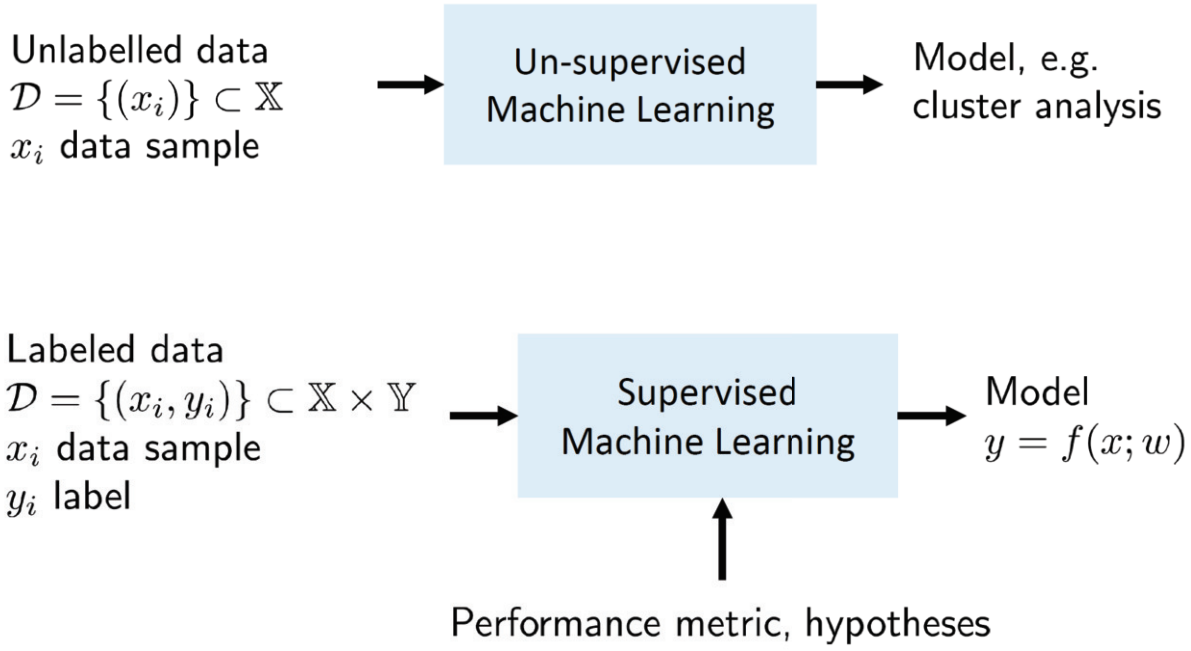
See: Overview - general, Overview - methods

1. Supervised vs. Unsupervised.

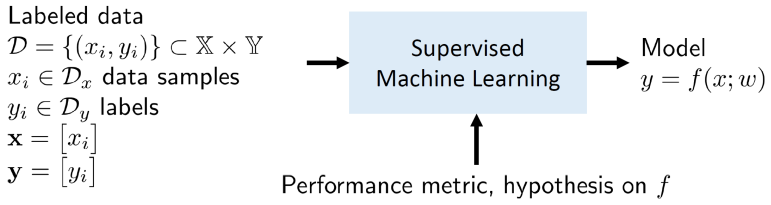
2. Regression vs. Classification.

3. Mixtures of the above?

Supervised vs. Unsupervised

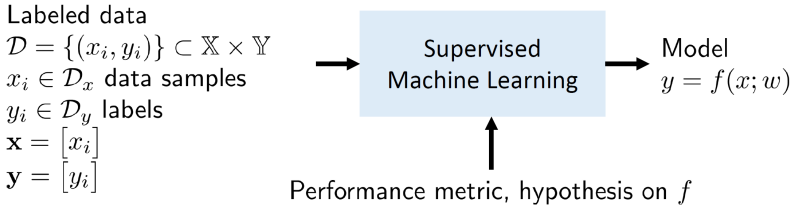


Regression vs. Classification



Model purpose – Regression

- The model f shall map $x \mapsto y$ and approximate an unknown function $\hat{f} : \mathbb{X} \to \mathbb{Y}$
- $y_i \in \mathbb{Y} \subseteq \mathbb{R}^{n_y}$
- Examples: data-driven modeling, energy forecasting, ...



Model purpose – Classification

- The model f shall map $x \mapsto y$ and approximate an unknown function $\hat{f} : \mathbb{X} \to \mathbb{Y}$
- $y_i \in \mathbb{Y} \subseteq \mathbb{N}^{n_y}$
- Examples: spam filter, fraud detection, fault detection, ...

The only difference is the space over which the response Y is defined!

METHOD CHOICE

Classes of ML Methods

- supervised learning
- unsupervised learning
- reinforcement learning (AlphaGo, AlphaFold)
- self-supervised learning (LLMs, ChatGPT)

Table of ML Methods

Class	Model	Task
Supervised	Linear regression	R
	CART (trees)	R, C
	SVM	R, C
	NN	R, C
	k-NN	C
	Naive Bayes	C
Unsupervised	k-means	clustering
	hierarchical	clustering
	PCA	patterns

- R = regression, C = classification
- CART = Classification and Regression Trees
- SVM = Support Vector Machine
- NN = Neural Network
- PCA = Principal Component Analysis

Methods - presentations

Lectures and code examples for all the above methods are available on the CMU-IMU-2023 course website.

How to choose a model?

- Once the tuning (hyper)parameters have been determined/chosen (see: cross-validation below), we still must choose between several models

  $\Rightarrow$ the choice will largely depend on the data characteristics and the type of questions we ask

- But, predicting which model will be the most pertinent, is in general quite difficult...

- A recommended scheme for finalizing the choice is as follows:

1. Begin with a few models that are the least interpretable and the most flexible. These models have a strong chance of producing better precision.

   (a) SVM.
   (b) Trees with boosting.

(c) Random Forests (RF).

2. Study simpler models, that are less opaque.

   (a) Linear models.
   (b) (GAM/GLM)
   (c) Naive Bayes.
   (d) k-NN.
   (e) Logistic Regression.
   (f) Regression Splines (MARS).

3. Use, if possible, the simplest model (see: bias-variance trade-off) that approximates reasonably well the performance of the more complex models.

Remark 1. In ML, the algorithm learns the parameters of the model, usually by optimization of a error-loss score. The method itself is defined by hyperparameters. These hyperparameters control the learning process. They have default values, but need to be tuned for each new dataset.

EDA

Recall: EDA

See: Overview - general

Remark 2. In addition to plots and summary statistics, unsupervised learning techniques often play an important role in identifying structure in the data.

CROSS-VALIDATION and TUNING

Recall: CV and Tuning

See: Resample,, elementary train-test tutorial, a bit more advanced CV tutorial

- Take-home lessons:

  ⇒ Basic, train-test split is not recommended since a single split may not be representative, and the resulting score may not be a reliable estimate of the predictive power on unseen data (i.e. in production.)

  ⇒ Simple, k-fold CV is better, but is rarely sufficient to ensure model reliability.

  ⇒ Nested CV with inner loop to select hyperparameters, outer loop to evaluate the best model, is probably the best option.

  ⇒ Repeated k-fold CV.

  → A single run of the k-fold cross-validation procedure may result in a noisy estimate of model performance.

  → Different splits of the data may result in very different results.

→ The mean result is expected to be a more accurate estimate of the true unknown underlying mean performance of the model on the dataset, as calculated using the standard error.

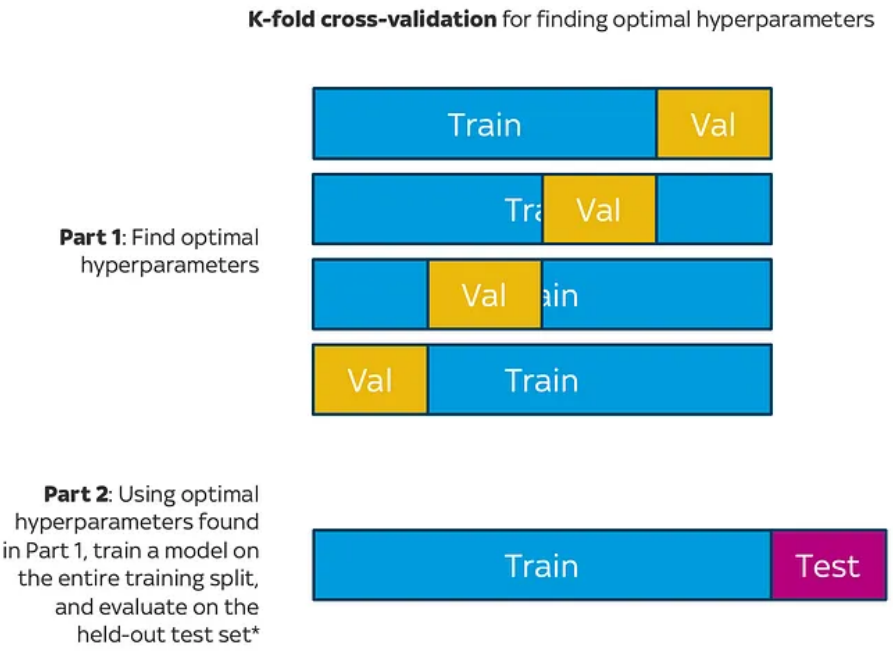
→ Usual numbers of repeats include 3, 5, and 10.

⇒ Time-series data requires special splitting, that respects the temporality (autocorrelation). For this one should use the scikit function TimeSeriesSplit

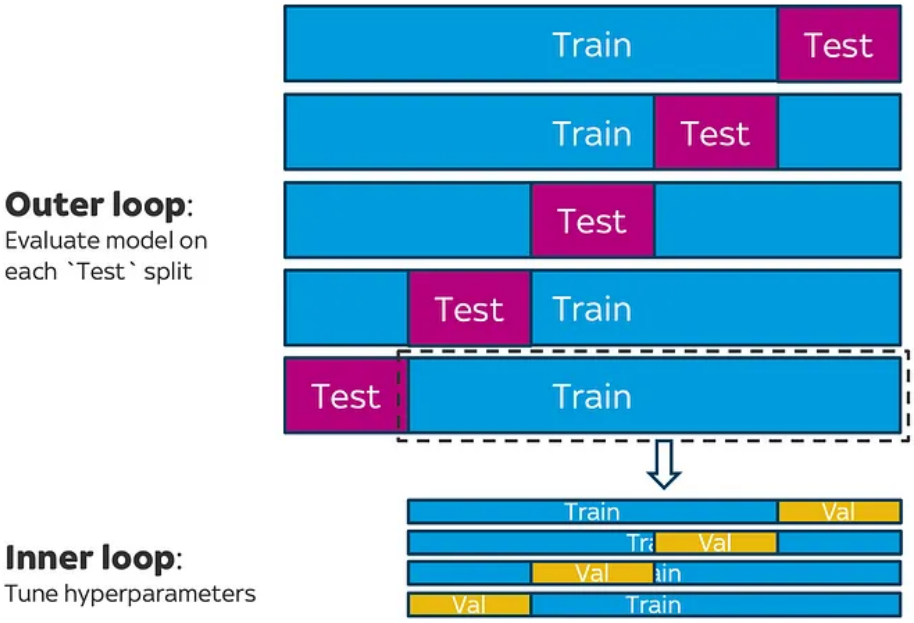
Nested CV

Single split:

k-fold CV:



Nested k-fold CV:



See Example notebook.

Resampling: Take-Home Lessons

No resampling method is uniformly better than another and the choice of a particular method should consider several aspects.

- If the sample size is small, repeated (5- or) 10-fold cross-validation is recommended for several reasons:

  $\Rightarrow$ the bias and variance properties are good and,
  $\Rightarrow$ given the sample size, the computational costs are not large.

- If the goal is to choose between models, as opposed to getting the best indicator of performance, it is advisable to use a bootstrap method since these have very low variance.

- For large sample sizes, the differences between resampling methods become less pronounced, and computational efficiency becomes the critical factor. Here,

simple 10-fold cross-validation should provide acceptable variance and low bias, and is relatively cheap to compute.

Finalized Model

- A final machine learning model is a model that you will use to make predictions on new data.
- You finalize a model by applying the chosen, tuned machine learning procedure on all of your data. That's it.
- See Example notebook.

EVALUATION METRICS

Loss Functions and Metrics

We need to distinguish between metrics for the 2 following cases. See: linear regression, model selection.

- For a given model: MSE, R^2, etc.
- For comparing different models: AIC, BIC, etc.

MSE in Practice

See Overview - general

- This is the oldest (Gauss, Legendre, in the 1800's...) error (or loss) metric.
  - $\Rightarrow$ MSE, or RMSE, is adapted to optimization errors (is this the best possible fit?), and should be used exclusively for this.
  - $\Rightarrow$ MSE, or RMSE, is not adapted to evaluation of a ML model (how well is it doing?), since it is highly sensitive to outliers.

- For model performance evaluation, one should prefer the Median Absolute Deviation (MAD), or the Mean Absolute Error (MAE), though the former is more robust to outliers. In other words, the choice of metric depends on the characteristics of your data.

- To avoid orders of magnitude in errors, due to measurement units, and to obtain relative measures, the MAD and MAE can be recalculated as percentages.

R^2 in Practice

- This very widely-used regression metric has a number of shortcomings, and some people say that it should not be used at all, because:

  $\Rightarrow$ R^2 relies on data leakage
  $\Rightarrow$ R^2 decreases with increasing noise level
  $\Rightarrow$ R^2 increases with increasing model complexity

- One can use R^2 for the following instances:

  $\Rightarrow$ determine whether a change in explanatory variables improves the model fit
  $\Rightarrow$ compare models with different subsets of explanatory variables
  $\Rightarrow$ detecting co-linearity

- Possible solutions

  $\Rightarrow$ do not use it,

$\Rightarrow$ use an out-of-sample version where, for test performance evaluation, the TSS is calculated over the training samples only,

$$\mathrm{TSS}_{\mathrm{oos}} = \sum_{i=1}^{n} (y_i - \bar{y}_{\mathrm{train}})^2$$

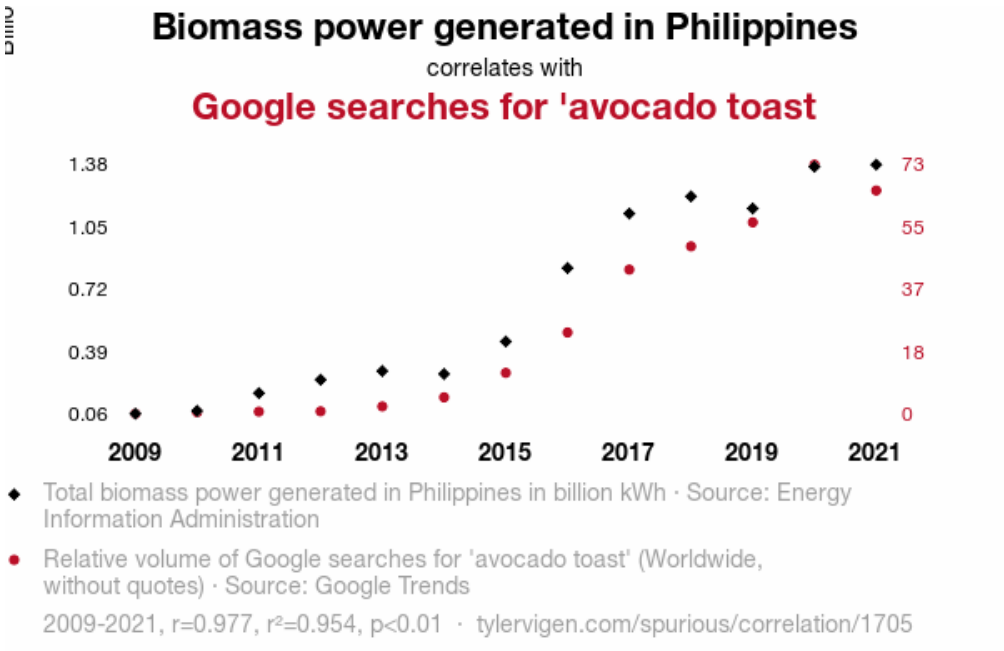
Classification Metrics

- There are a number of potential pitfalls in measuring the performance of classification models.
- The scikit tutorial is an excellent basis for illustrating several of these problems.
- See also: Overview - general.

CAUSALITY
and
CORRELATION

Principles

- Correlation can be misleading.



- Causality can be difficult to evaluate.

  ⇒ Confounder analysis.
  ⇒ Partial correlation.
  ⇒ Shapley values.

FEATURES

Features

- Feature selection: see model selection.
- Feature engineering: transformed values can be more meaningful.
- Feature importance: use ranking provided by ensemble-based methods.

  ⇒ The scikit tutorial is an excellent basis for illustrating these aspects.

  ⇒ Shapley values are powerful indicators of feature importance.

Shapley Values

Definition

Shapley values quantify the significance of individual input features' contribution to the model's predicted values.

- Analysis of explainability through SHAP regression values aims to evaluate the contribution of input variables (often called "input features") to the predictions made by a machine learning predictor model.

- The contribution of that input feature to a prediction is calculated mathematically through the construction of a so-called "explanation model".

- The explanation model evaluates the predictions of a machine learning system as the sum of the contributions of each input feature and the mean predicted value.

- Mathematically, the explanation model can be stated as

$$y = \bar{y} + \sum_i \phi_i,$$

where

⇒ y is an individual prediction,
⇒ $\bar{y}$ is the average predicted value across all predictions, and
⇒ ϕ_i is the contribution of input feature i to the prediction (also known as the "SHAP regression value" or "SHAP value").

- Input variables with larger magnitude SHAP values are interpreted as contributing more to a specific prediction than those with a smaller magnitude SHAP values.

- For a given case,

⇒ positive SHAP values indicate a specific feature contributes toward increasing the final predicted value y, and
⇒ negative SHAP values indicate a contribution toward decreasing the prediction.

- These SHAP values, ϕ_i, are calculated following a game theoretic approach to assess prediction contributions.

- The SHAP framework has several key desirable properties:

  $\Rightarrow$ The sum of the contributions accurately reproduces the predicted value.

  $\Rightarrow$ The contributions of input features that are not present in the machine learning model are assigned values of 0.

  $\Rightarrow$ SHAP is a machine learning model agnostic technique and can be applied to a wide class of prediction methods.

- Despite these key advantages, there are several potential deficiencies to the application of SHAP analysis to error characterization in the Earth Sciences.

  $\Rightarrow$ SHAP analysis cannot directly yield causal insights.

  $\Rightarrow$ Direct calculation of SHAP values is very computationally expensive.

  $\rightarrow$ As such, care must be taken to properly interpret the SHAP values resulting from any particular analysis.

→ It is important to note that to improve computational performance, common implementations of SHAP analysis in existing machine learning libraries

Warnings on SHAP for Causality

- Flexible predictive models like XGBoost or LightGBM are powerful tools for solving prediction problems. However, they are not inherently causal models, so interpreting them with SHAP will fail to accurately answer causal questions in many common situations.

- Unless features in a model are the result of experimental variation, applying SHAP to predictive models without considering confounding is generally not an appropriate tool to measure causal impacts used to inform policy.

- SHAP and other interpretability tools can be useful for causal inference, and SHAP is integrated into many causal inference packages, but those use cases are explicitly causal in nature.

- To that end, using the same data we would collect for prediction problems and using causal inference methods like double ML that are particularly designed to return

causal effects is often a good approach for informing policy.

- In other situations, only an experiment or other source of randomization can really answer what if questions. Causal inference always requires us to make important assumptions.

- The main point is that the assumptions we make by interpreting a normal predictive model as causal are often unrealistic.

PINN

Physics Informed Neural Networks

See: Basics of NNs, PINNs

Remark 3. This approach, though seemingly very elegant and powerful, has serious problems of cost, robustness and precision. It often requires a lot of tuning to work. PINNs should be employed with great caution.

ETHICS

Moral, Ethics and Bias

See: Ethics

References

1. Please consult the list provided on the CMU-IMU-2023 website:
   CODE REFERENCES

2. Scikit-learn MOOC website

3. VSCODIUM for code editing Site, Downloads

4. G. James, D. Witten, T. Hastie, R. Tibshirani, J. Taylor. *An Introduction to Statistical Learning*. Springer. 2023.
   ISL Site