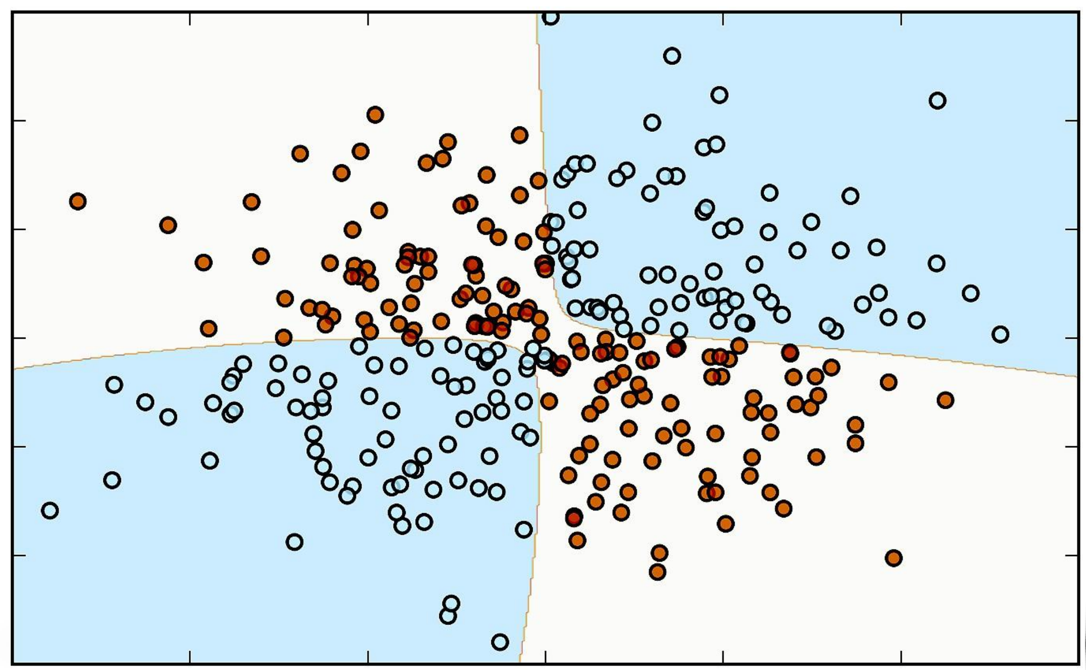
Teaching Demo

Introduction of Classification (Supervised Learning): Big Pictures and Practicum



Ludy Hasby Aulia

Lab Assistant Candidate
PJJ Computer Science - Binus University

June. 16, 25

Our Agenda

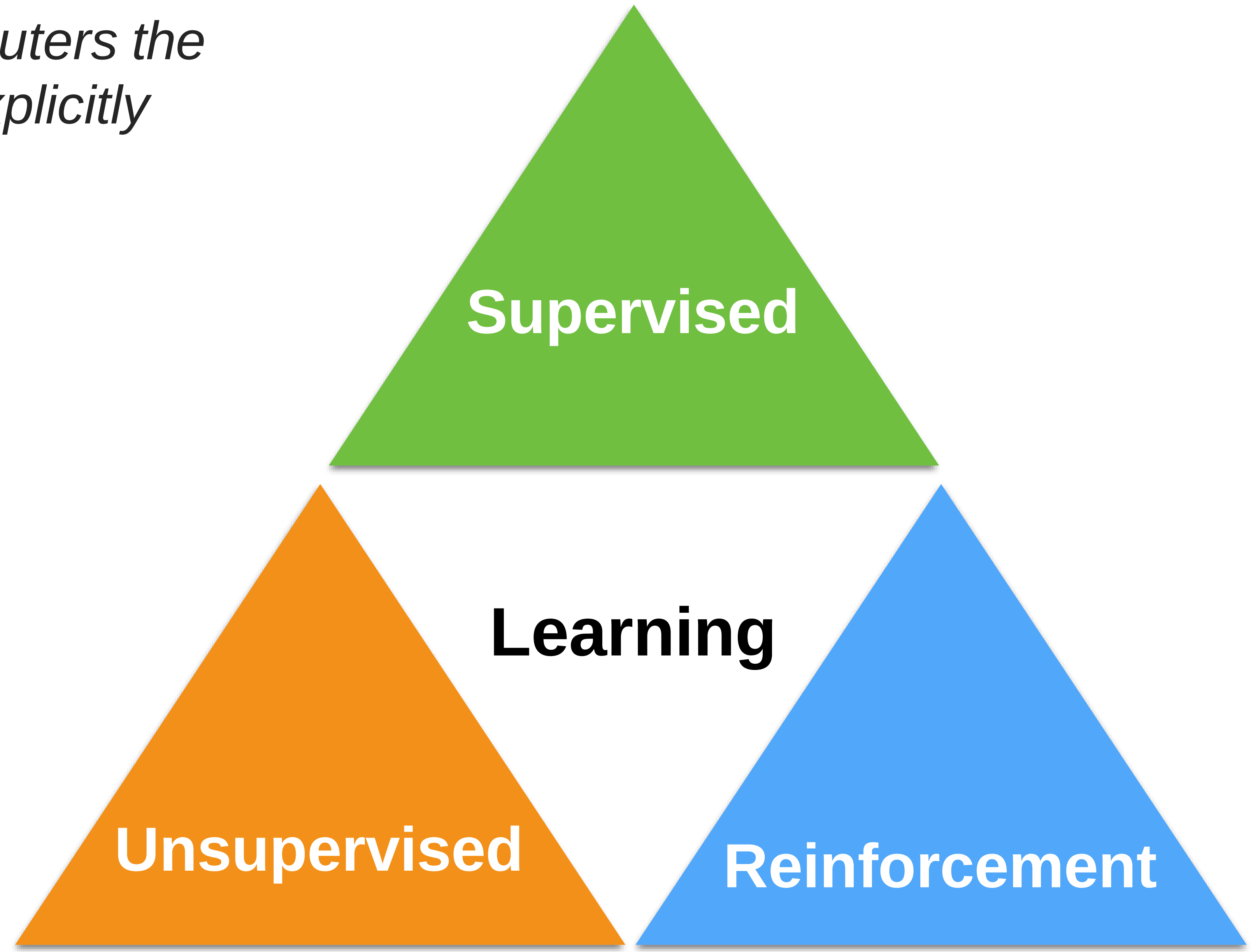
- **Algorithm in big picture**
- **Example Case**
- **Model Building Demo**

Machine Learning?

"Field of study that gives computers the ability to learn without being explicitly programmed."

(Arthur Samuel, 1959)

- Labeled data
- Direct feedback
- Predict outcome/future

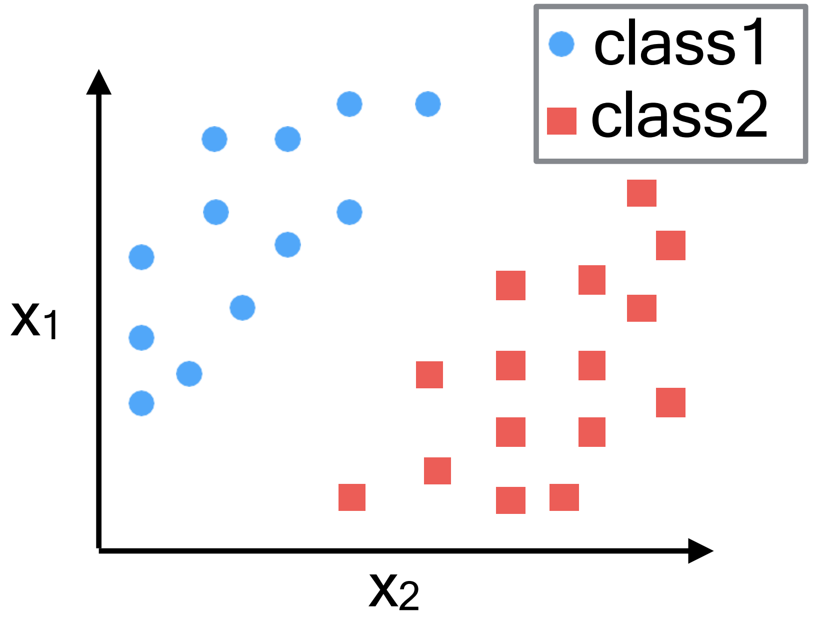


- No labels
- No feedback
- "Find hidden structure"

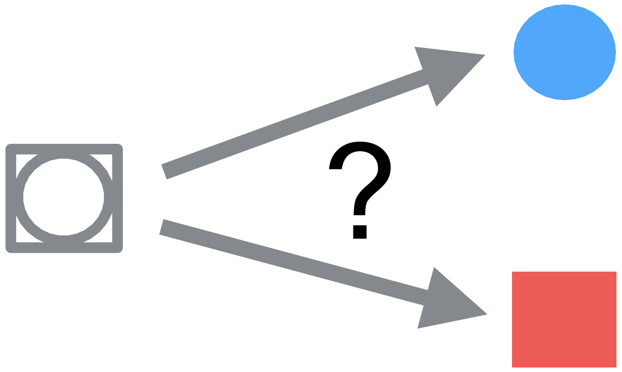
- Decision process
- Reward system
- Learn series of actions

Classification

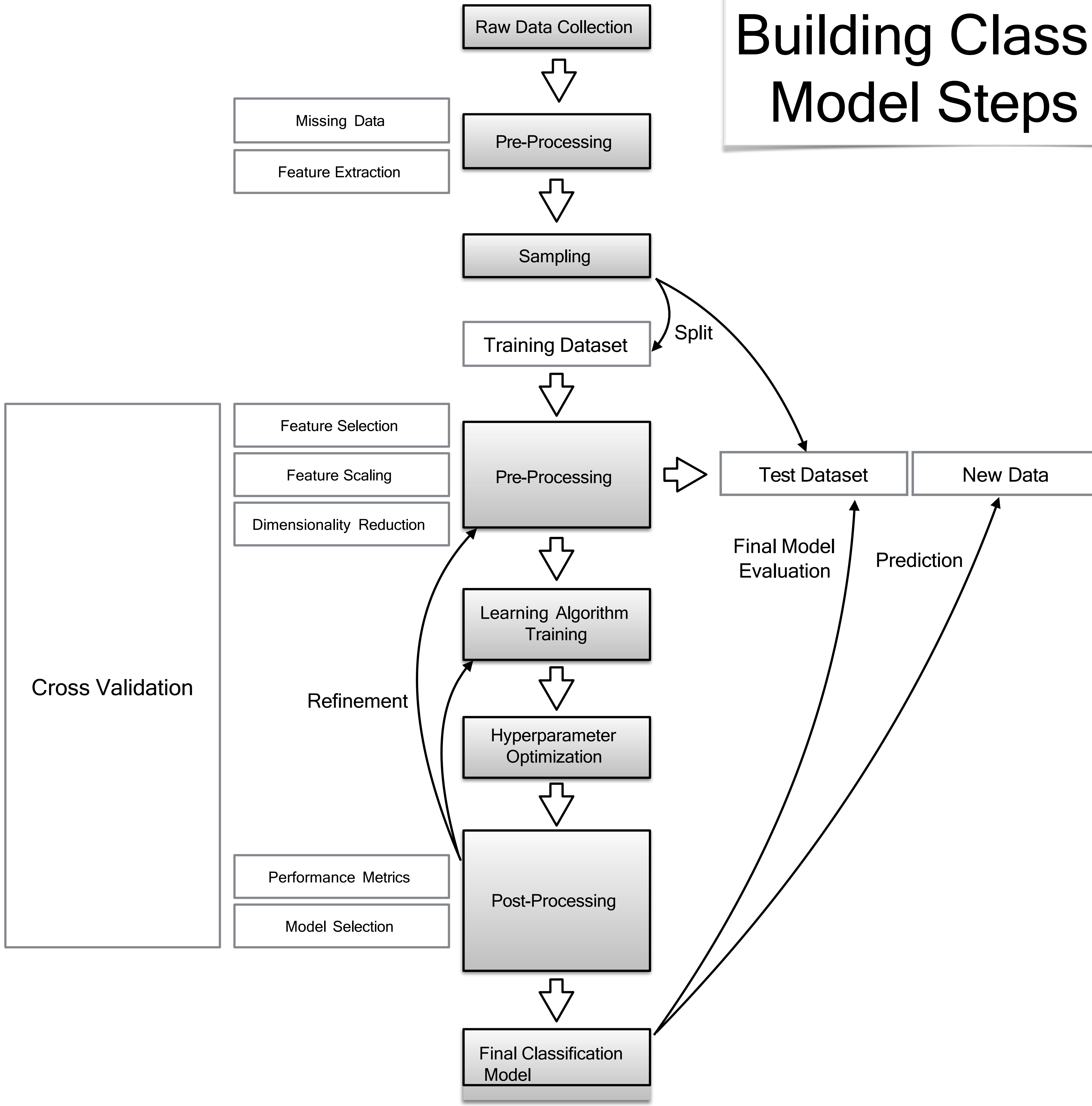
1) Learn from training data



2) Map unseen (new) data

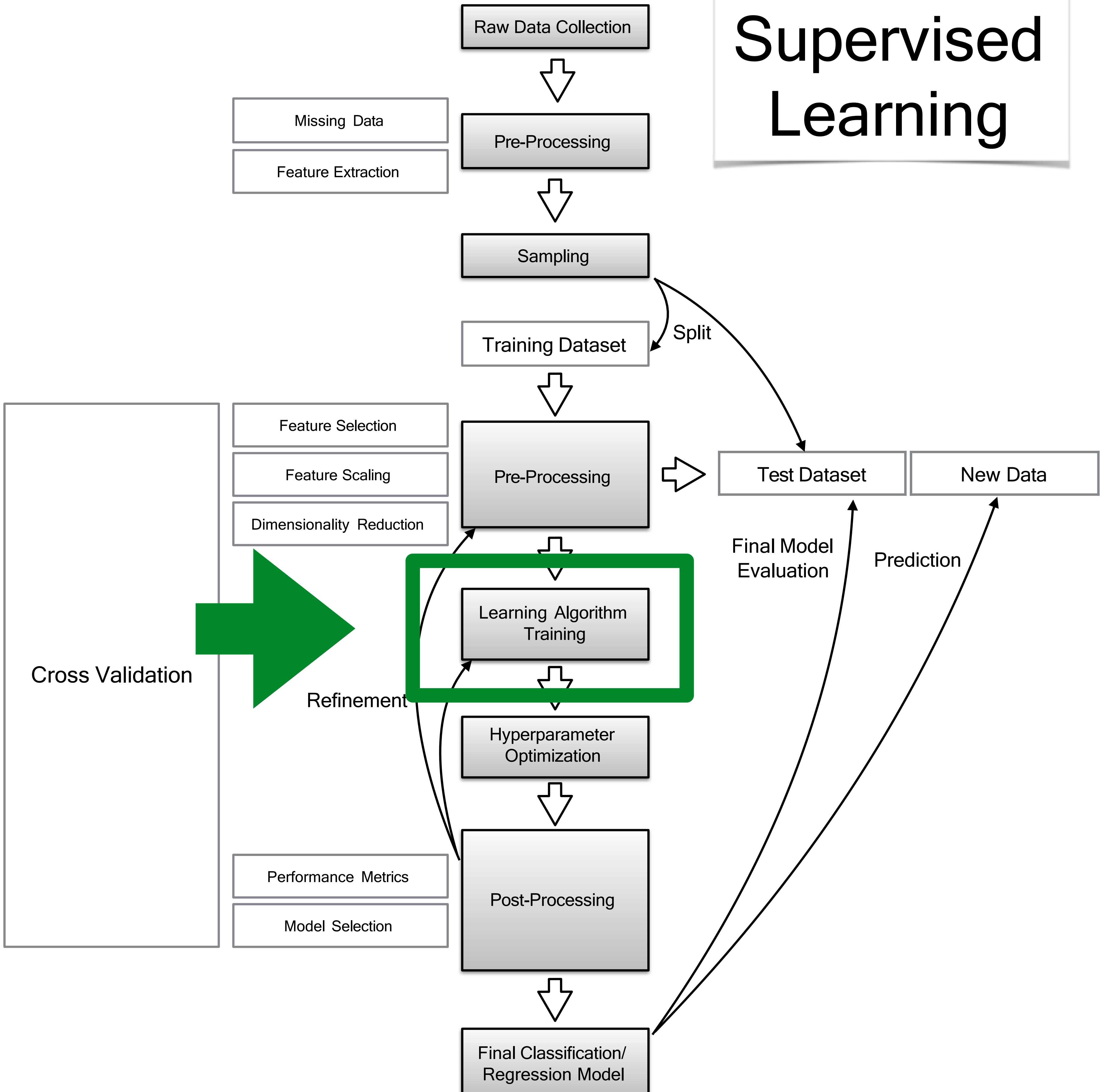


Building Classification Model Steps



Sebastian Raschka 2014

Supervised Learning



A Few Common Classifiers

Perceptron

Naive Bayes

Decision Tree

K-Nearest Neighbor

Logistic Regression

Artificial Neural Network / Deep Learning

Support Vector Machine

Ensemble Methods: Random Forest, Bagging, AdaBoost

A Few Common Classifiers

Perceptron

Naive Bayes

Decision Tree

K-Nearest Neighbor

Logistic Regression

Artificial Neural Network / Deep Learning

Support Vector Machine

Ensemble Methods: Random Forest, Bagging, AdaBoost

Logistic Regression

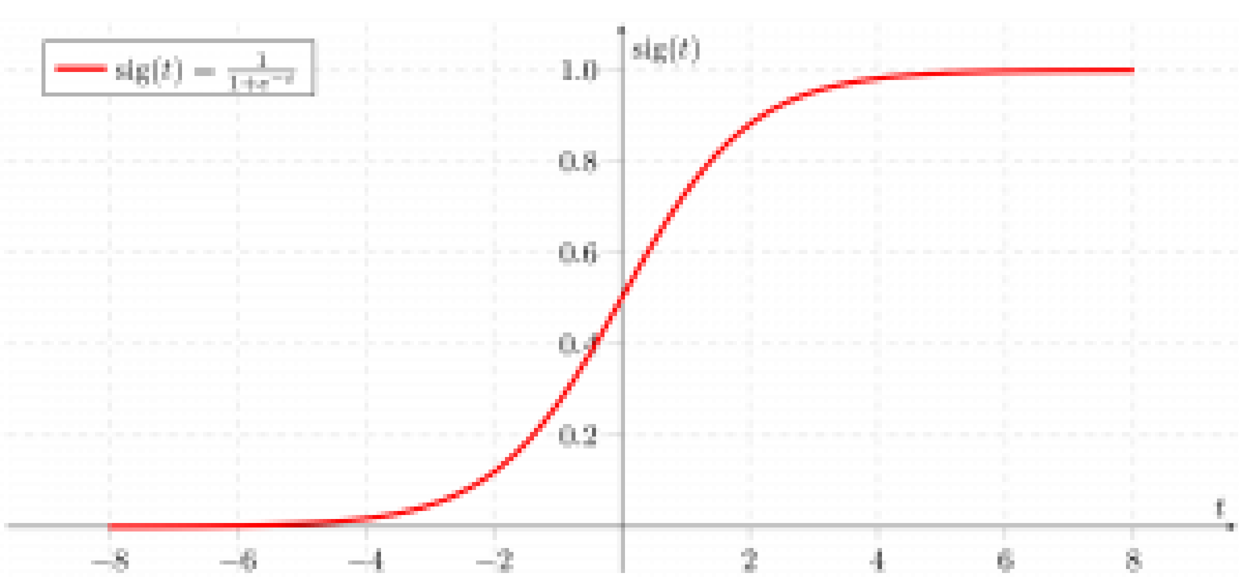
Binomial

Step 1. Calculate and Predict MLR

$$\mathbf{z} = b + w.X$$

Step 2. Convert to Sigmoid Function

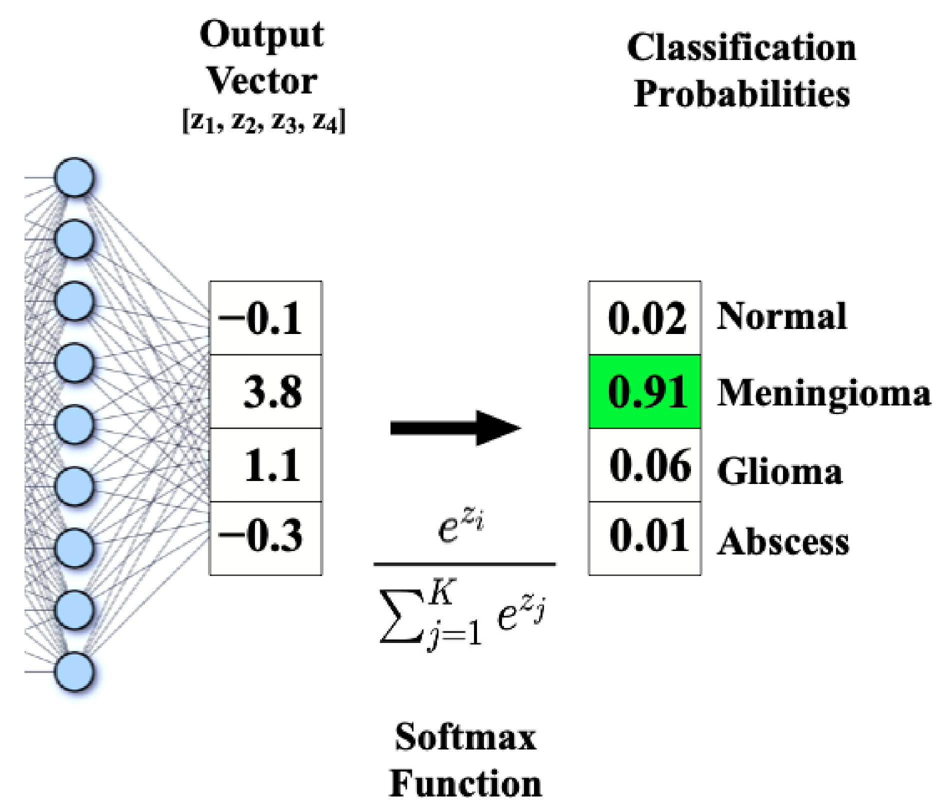
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



Multinomial

Use Softmax Function

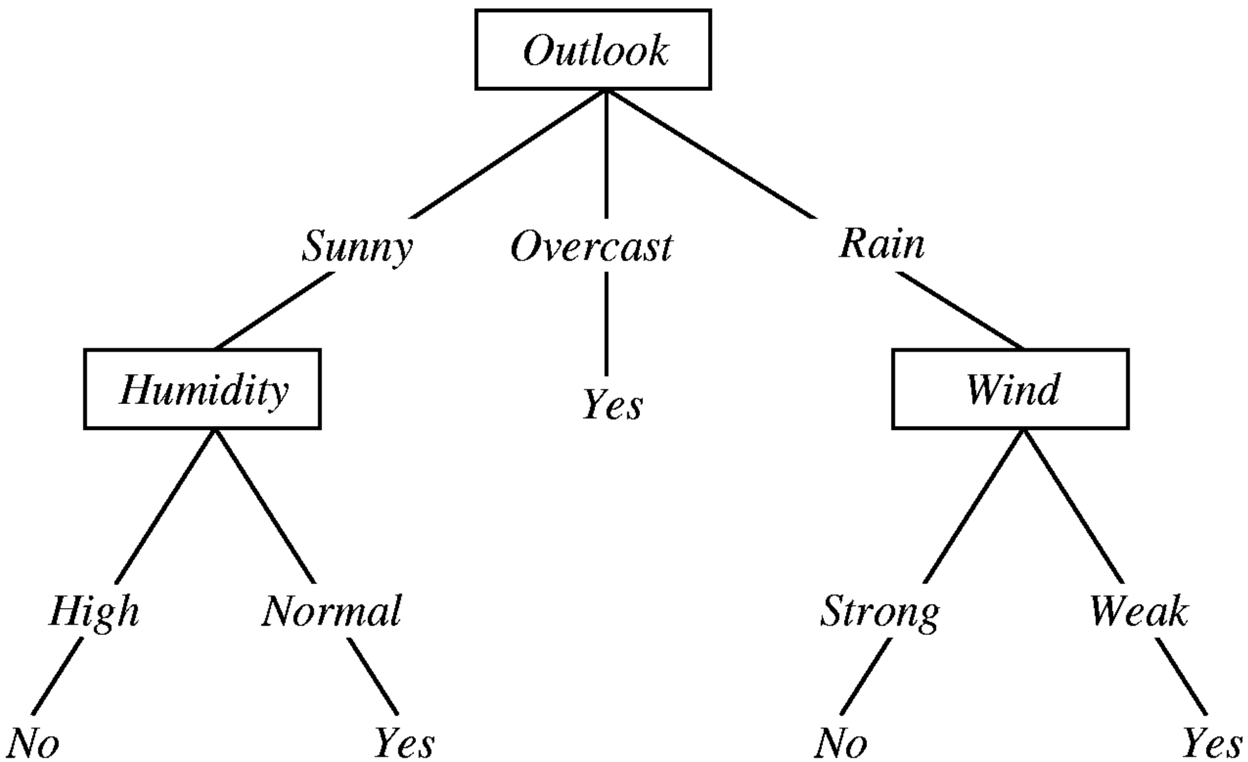
$$softmax(z) = \frac{e^{z}}{\sum_{j=1}^{K} e^{z_j}} = \mathrm{P}(\mathrm{Y} = \mathrm{c} \mid \mathrm{X})$$



Read More

https://www.geeksforgeeks.org/machine-learning/understanding-logistic-regression/

Decision Tree



Read More

https://www.geeksforgeeks.org/machine-learning/decision-tree-algorithms/

Algorithm ID3

Step 1. Select an attribute as the root (Highest Gains)

$$\boldsymbol{Entropy(S)} = \sum_{i=1}^{n} -pi \ * \log_2 pi$$

$$\boldsymbol{Gain(S, A)}$$

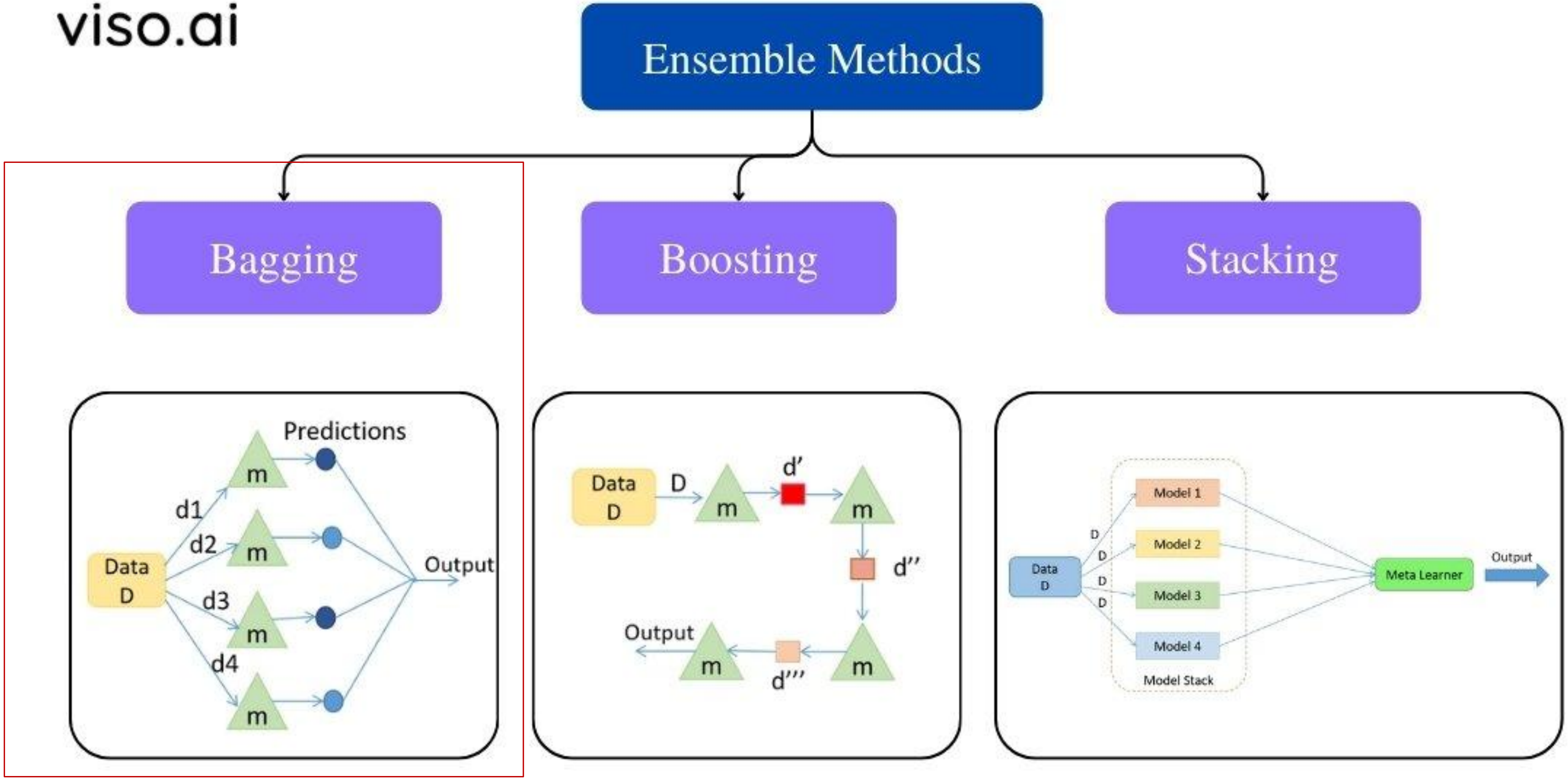
$$= Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$

Step 2. Create Branch Each Value

Step 3. Repeat the process until each branch has the same class

Random Forest

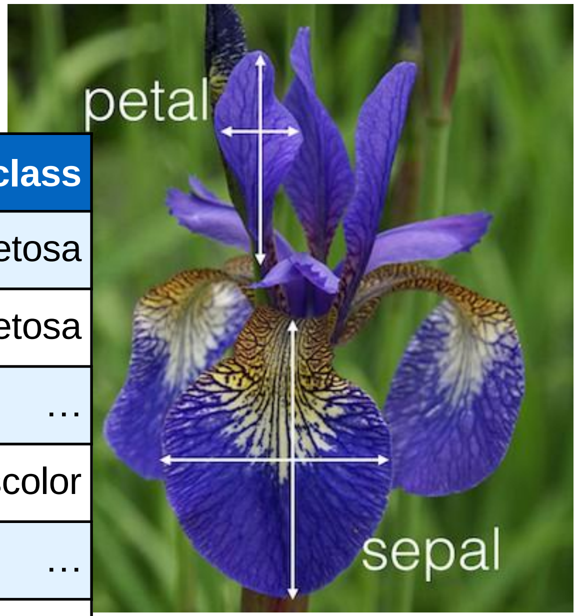
viso.ai



Classification Case

IRIS

https://archive.ics.uci.edu/ml/datasets/Iris



Instances (samples, observations)

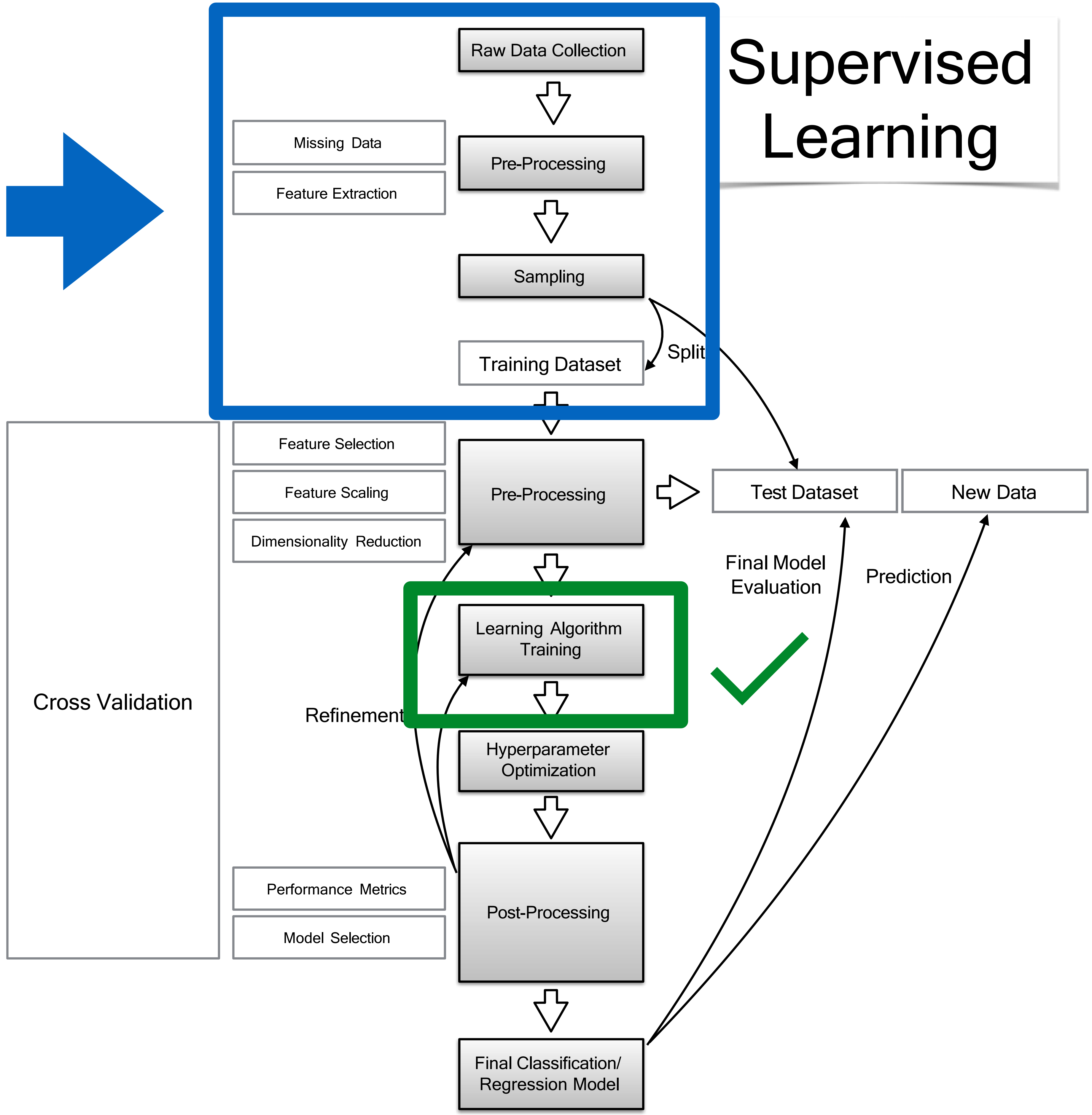
	sepal_length	sepal_width	petal_length	petal_width	class
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
…	…	…	…	…	…
50	6.4	3.2	4.5	1.5	veriscolor
…	…	…	…	…	…
150	5.9	3.0	5.1	1.8	virginica

Features (attributes, dimensions)

Classes (targets)

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ x_{31} & x_{32} & \cdots & x_{3D} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix}$$

$$\mathbf{y} = [y_1, y_2, y_3, \cdots y_N]$$


Supervised Learning
Raw Data Collection
Missing Data
Feature Extraction
Pre-Processing
Sampling
Split
Training Dataset
Feature Selection
Feature Scaling
Dimensionality Reduction
Pre-Processing
Test Dataset
New Data
Final Model Evaluation
Prediction
Cross Validation
Refinement
Learning Algorithm Training
Hyperparameter Optimization
Performance Metrics
Model Selection
Post-Processing
Final Classification/ Regression Model

Missing Values:

- Remove features (columns)
- Remove samples (rows)
- Imputation (mean, nearest neighbor, …)

Feature Scaling:

e.g., *standardization:*

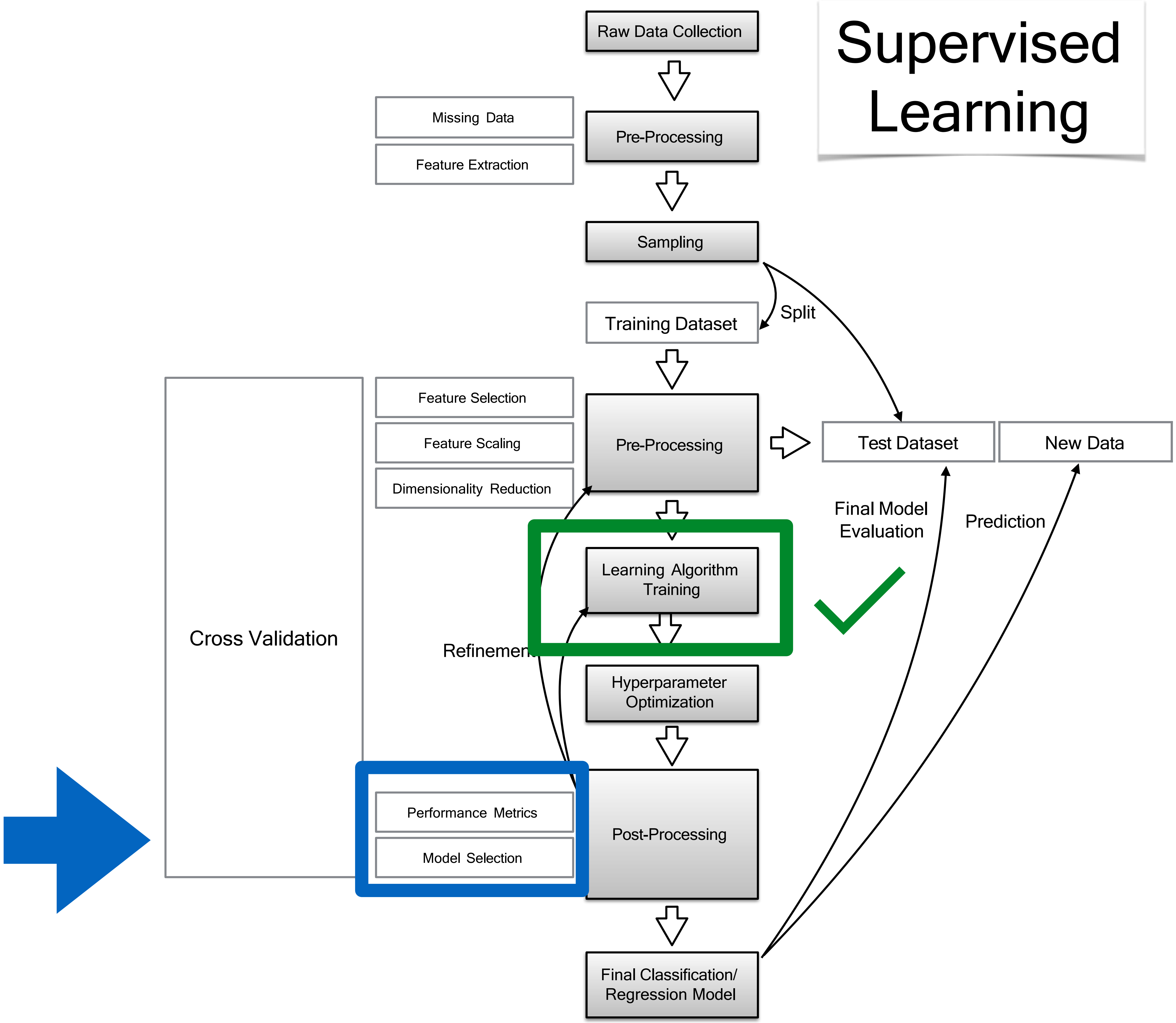
$$z = \frac{x_{ik} - \mu_k}{\sigma_k}$$

(use same parameters for the test/new data!)

- Faster convergence (gradient descent)
- Distances on same scale (k-NN with Euclidean distance)
- Mean centering for free
- Normal distributed data
- Numerical stability by avoiding small weights

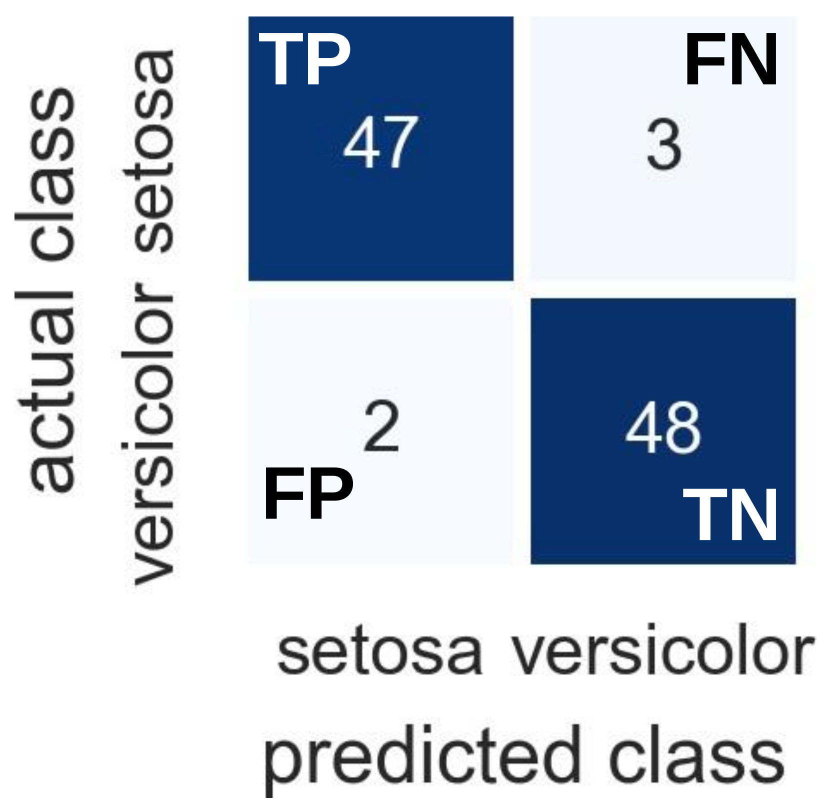
Sampling:

- Random split into training and validation sets
- Typically 60/40, 70/30, 80/20
- Don't use validation set until the very end! (overfitting)


Supervised Learning
Raw Data Collection
Missing Data
Feature Extraction
Pre-Processing
Sampling
Split
Training Dataset
Test Dataset
New Data
Cross Validation
Feature Selection
Feature Scaling
Dimensionality Reduction
Pre-Processing
Learning Algorithm Training
Refinement
Hyperparameter Optimization
Performance Metrics
Model Selection
Post-Processing
Final Model Evaluation
Prediction
Final Classification/ Regression Model

Error Metrics

here: “setosa” = “positive”



[Linear SVM on sepal/petal lengths]

“micro” and “macro” averaging for multi-class

$$\textit{Accuracy} = \frac{TP + TN}{FP + FN + TP + TN} = 1 - \textit{Error}$$

$$\textit{False Positive Rate} = \frac{FP}{N}$$

$$\textit{True Positive Rate}\ \textit{(Recall)} = \frac{TP}{P}$$

$$\textit{Precision} = \frac{TP}{TP + FP}$$

Demo Time