Introduction to Machine Learning

Dr. Elham Barezi
Lead Research AI Scientist

June 24th 2025

PURDUE UNIVERSITY

Outline

1. Definitions:
    a. AI, ML, DL
    b. ML categories
2. ML WorkFlow
    a. Data Preprocessing
    b. Data Partitioning
    c. Training
        i. cost function
        ii. optimization method selection
        iii. under/Overfitting or Bias/Variance Tradeoff
            1. regulating the loss function
            2. ensemble Learning
    d. Validation and test
        i. evaluation metric selection
3. Basic ML Models
4. Deep Learning

Outline

1. Definitions:
    a. AI, ML, DL
    b. ML categories
2. ML WorkFlow
    a. Data Preprocessing
    b. Data Partitioning
    c. Training
        i. cost function
        ii. optimization method selection
        iii. under/Overfitting or Bias/Variance Tradeoff
            1. regulating the loss function
            2. ensemble Learning
    d. Validation and test
        i. evaluation metric selection
3. Basic ML Models
4. Deep Learning

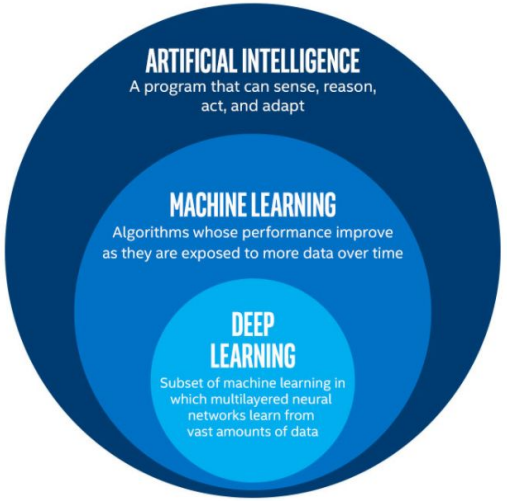
Some Definitions

Artificial intelligence (AI) is technology that enables computers and machines to simulate human learning, comprehension, problem solving, decision making, creativity and autonomy:

Machine learning, rule-based, symbolic AI, planning, Genetic Algorithms & Evolutionary Computation

Machine learning is a pathway to artificial intelligence, which uses algorithms to automatically learn insights and recognize patterns from data, make increasingly better decisions: supervised, unsupervised, reinforcement learning

Deep learning is an advanced method of machine learning. Deep learning models use large neural networks — networks that function like a human brain to logically analyze data — to learn complex patterns and make predictions.



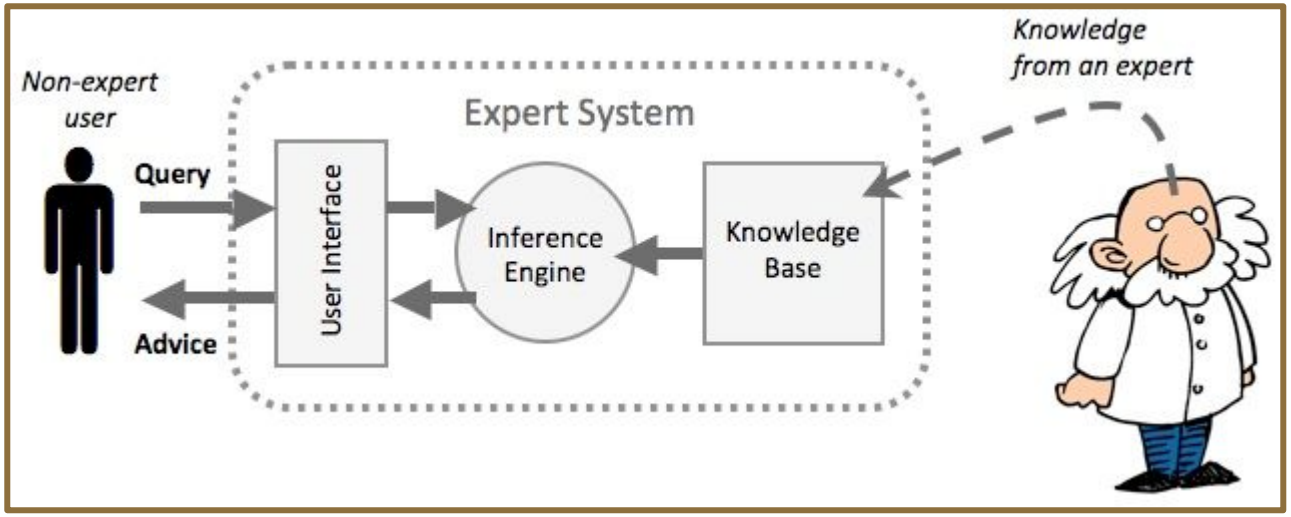
source:https://medium.com/@sidharthgn/but-what-is-the-difference-between-ai-machine-learning-deep-learning-2dd39ad55c40

PURDUE UNIVERSITY

Example of AI That Is Not Machine Learning:

Expert Systems

A **medical diagnosis expert system**, like **MYCIN (developed in the 1970s)**, is an **AI system that does not use ML** but instead relies on **if-then rules** to diagnose diseases and recommend treatments.



Why It's AI but Not ML?

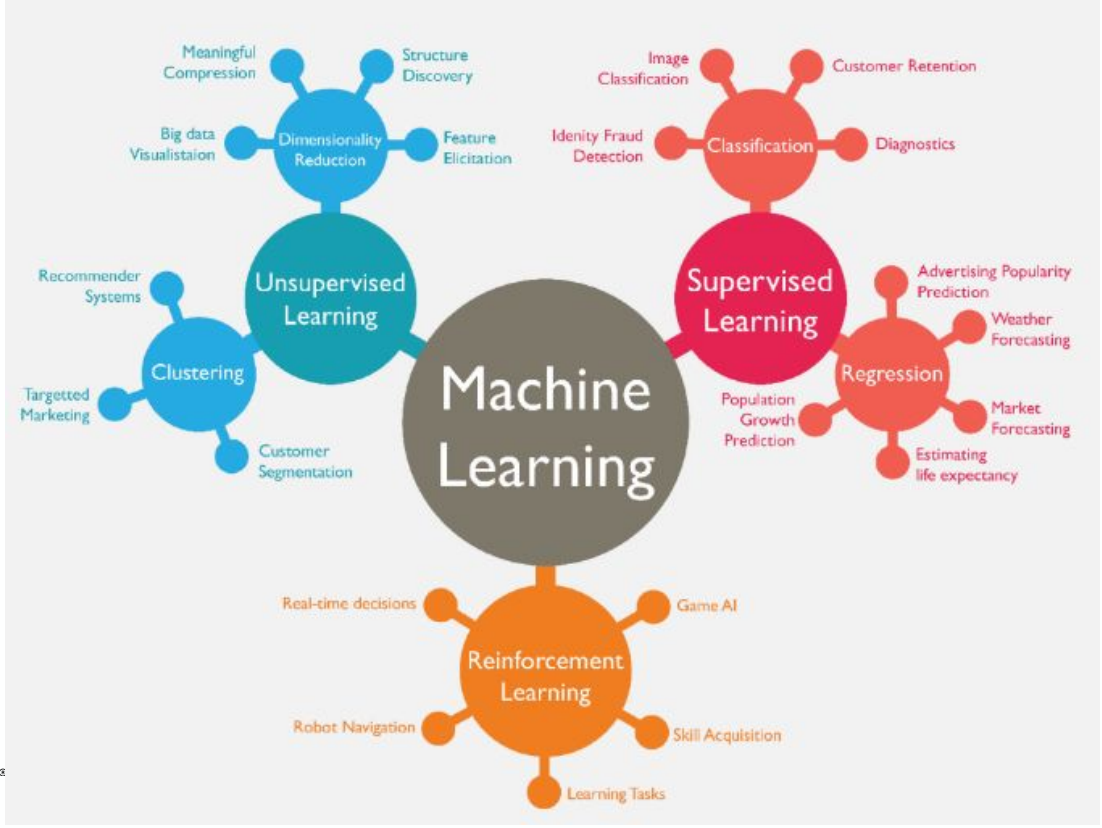
- **AI Aspect**: It **simulates human decision-making** using logic and reasoning.
- **Not ML**: It **does not learn from data**—all knowledge is predefined. new rules must be **manually added** by experts.

In contrast, a **machine learning-based diagnosis system** would analyze patient data, detect patterns, and improve over time!

photo source: https://balasubramanyamlanka.com/expert-system/

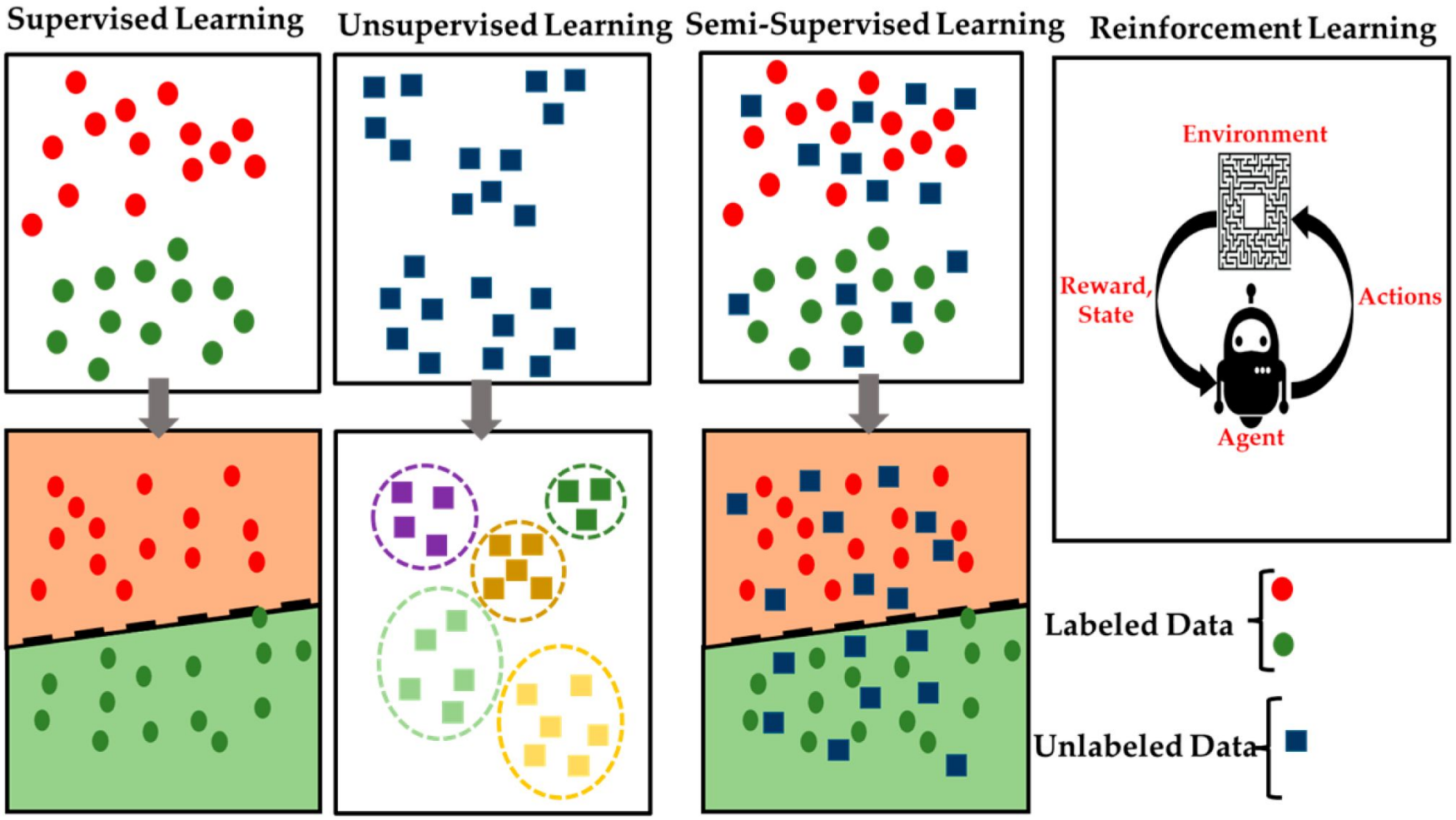
Machine Learning Categories

Supervised Learning	UnSupervised Learning	Reinforcement Learning
• Labeled data • Direct feedback • Predict outcome/future	• No labels/targets • No feedback • Find hidden structure in data	• Decision process • Reward system • Learn series of actions



https://www.digitalvidya.com/blog/semi-supervised-learning/

Machine Learning Categories

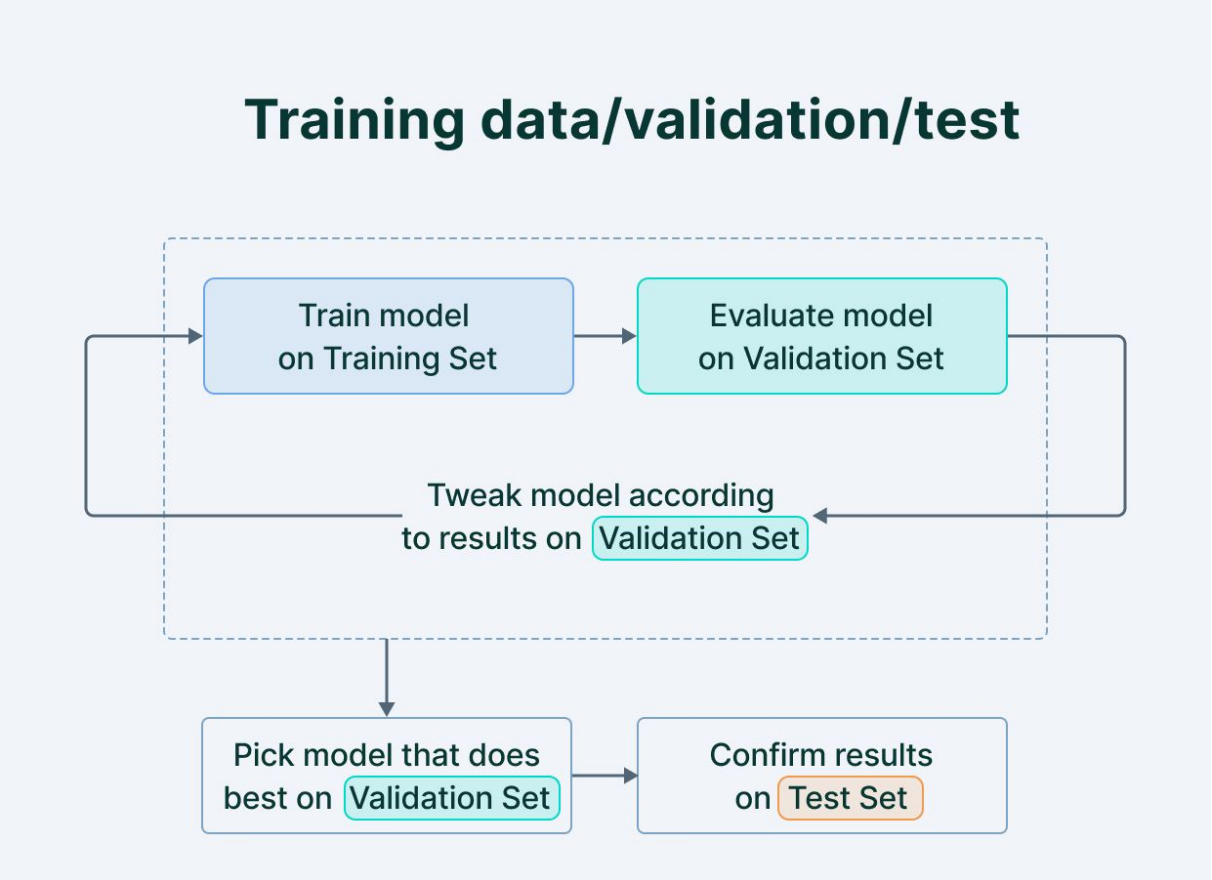


source: https://www.mdpi.com/1996-1073/16/16/5972

Outline

1. Definitions: AI, ML, DL
    a. What is AI, but not ML
    b. ML categories
2. ML WorkFlow
    a. Data Preprocessing
    b. Data Partitioning
    c. Training
        i. cost function
        ii. optimization method selection
        iii. under/Overfitting or Bias/Variance Tradeoff
            1. regulating the loss function
            2. ensemble Learning
    d. Validation and test
        i. evaluation metric selection
3. Basic ML Models
4. Deep Learning

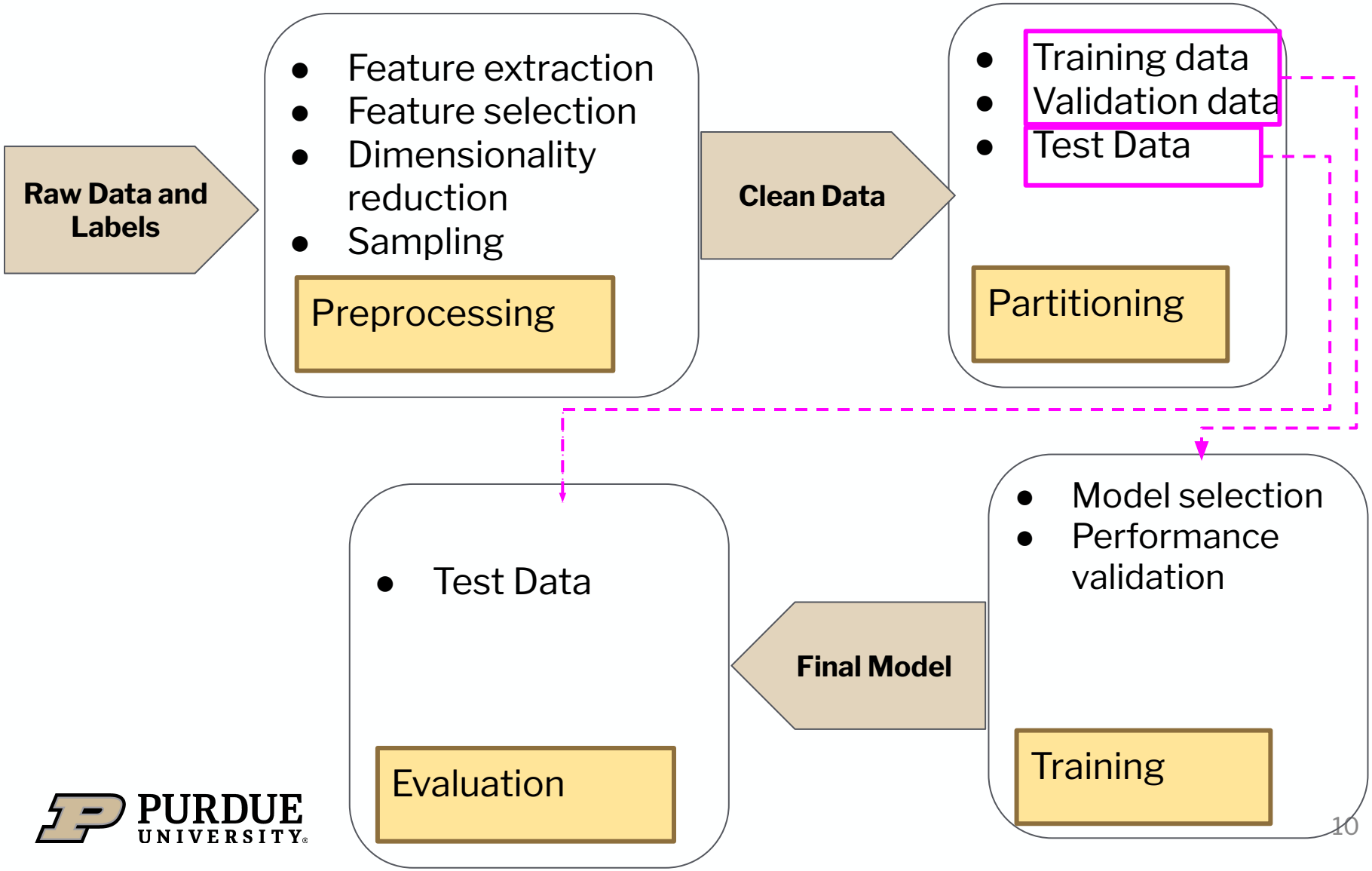
Machine Learning Paradigm



https://willmbennett.medium.com/clean-up-your-ml-process-using-helper-functions-f09a5920bd2e

PURDUE UNIVERSITY

Machine Learning Workflow



Outline

1. Definitions: AI, ML, DL
    a. What is AI, but not ML
    b. ML categories
2. ML WorkFlow
    a. Data Preprocessing
    b. Data Partitioning
    c. Training
        i. cost function
        ii. optimization method selection
        iii. under/Overfitting or Bias/Variance Tradeoff
            1. regulating the loss function
            2. ensemble Learning
    d. Validation and test
        i. evaluation metric selection
3. Basic ML Models
4. Deep Learning

Features and Labels

- Feature extraction: The process of converting the raw data to useful features.
  - **Images**: color, texture, and shape.
  - **Text**: word frequency (BOW), TF-IDF, …
  - **Speech**: pitch, energy, frequency, MFCC, …

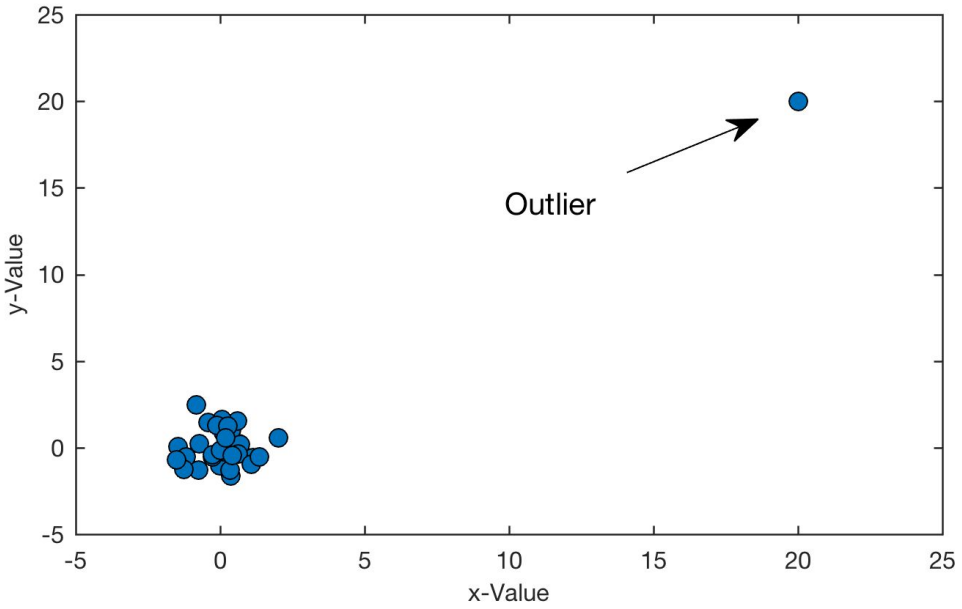
- Data Cleaning, Data normalization, feature selection,
- Handling Imbalanced Data
- Handling outlier

Animal	Ear Shape (Feature 1)	Nose Shape (Feature 2)	Tail Length (Feature 3)	Label
Cat	0 (Round)	1 (Pointed)	1 (Short)	Cat
Dog	1 (Pointy)	1 (Pointed)	2 (Medium)	Dog
Horse	1 (Pointy)	0 (Flat)	3 (Long)	Horse

Outlier Cleaning

An **outlier** is a data point that is **significantly different** from the rest of the data.

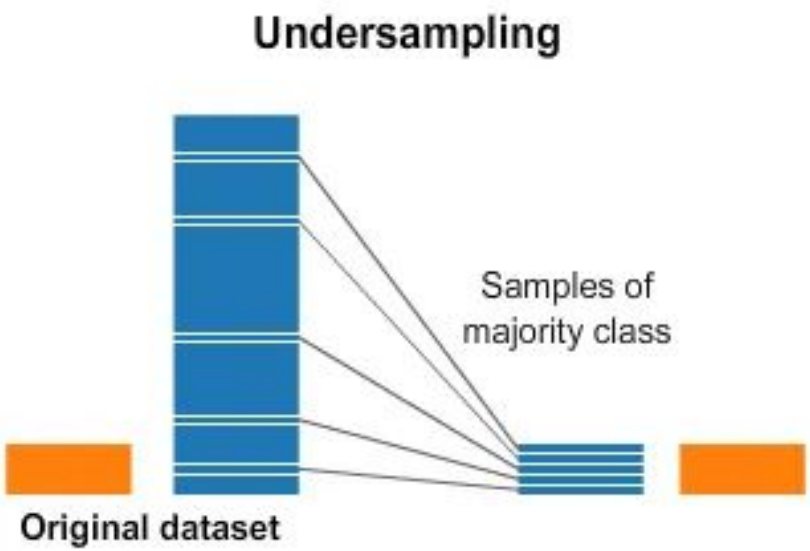
An outlier can be recognized through visual or statistical methods (Z-score, scatter plot, ..)

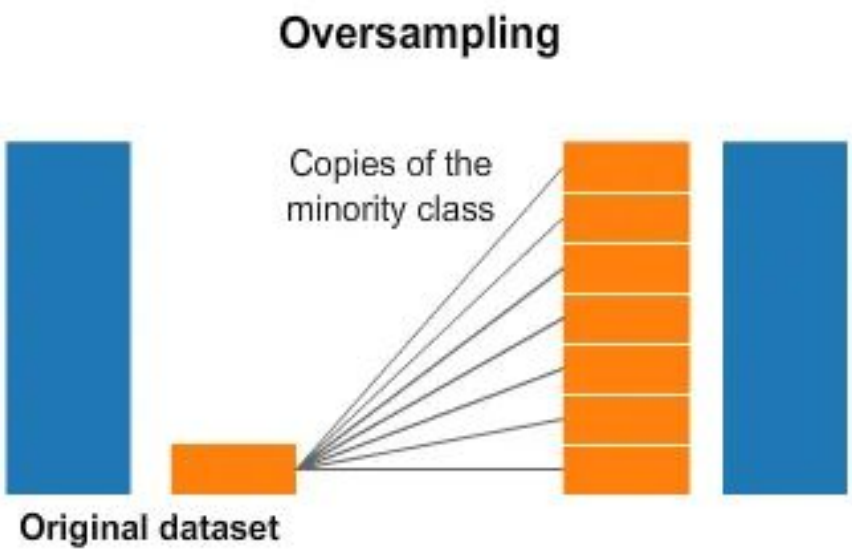


PURDUE UNIVERSITY

Imbalanced Data

- When the distribution of classes in a dataset is skewed, with one or more classes having significantly fewer samples than others,
- it can lead to trained models that make biased predictions and show poor overall performance.





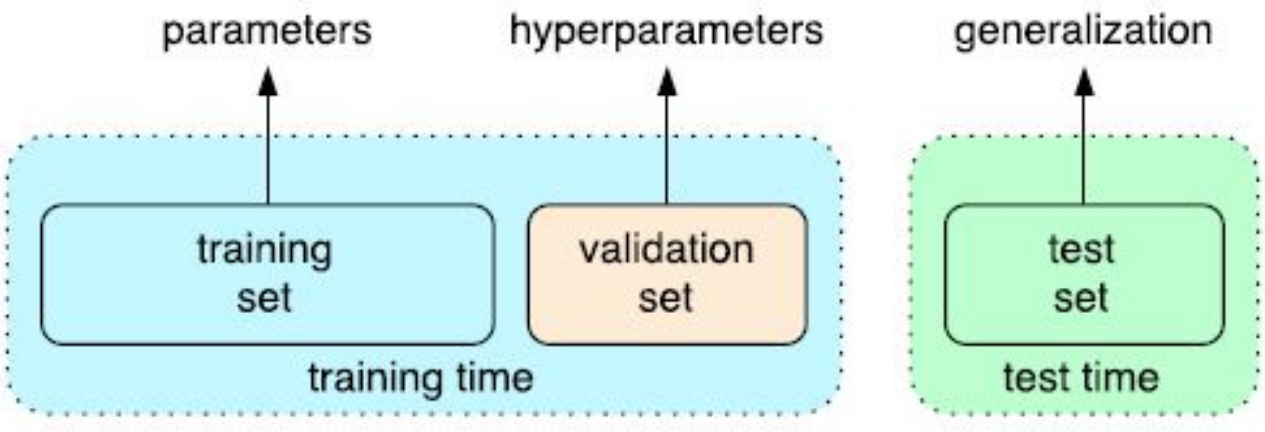
https://medium.com/@jainvidip/the-imbalance-dilemma-when-your-data-is-more-biased-than-a-reality-show-6bae25ddf78a

Outline

1. Definitions: AI, ML, DL
    a. What is AI, but not ML
    b. ML categories
2. ML WorkFlow
    a. Data Preprocessing
    b. Data Partitioning
    c. Training
        i. cost function
        ii. optimization method selection
        iii. under/Overfitting or Bias/Variance Tradeoff
            1. regulating the loss function
            2. ensemble Learning
    d. Validation and test
        i. evaluation metric selection
3. Basic ML Models
4. Deep Learning

Data Partitioning

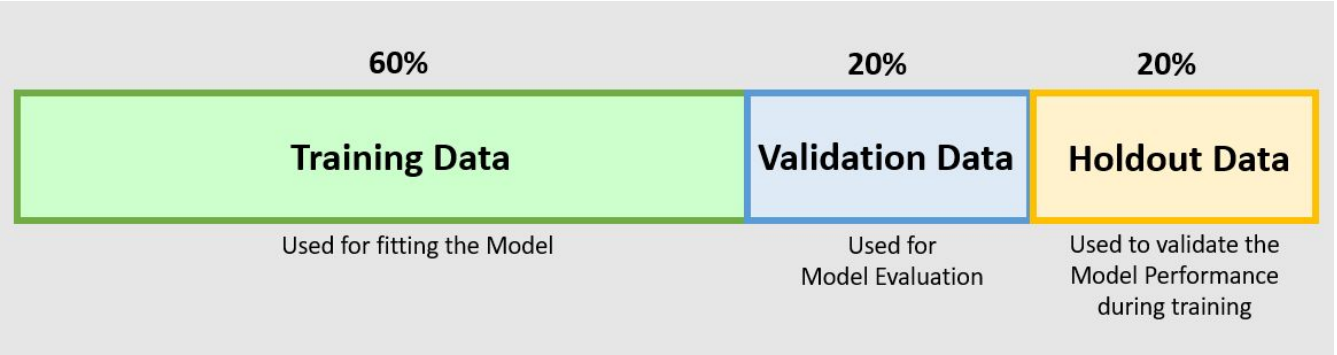
- To increase the predictive performance by tweaking the learning algorithm and selecting the best performing model from a given hypothesis space.
    - **Parameters** are **learned by the model** from data during training.
        - Example: Weights in a neural network, coefficients in linear regression.
    - **Hyperparameters** are **set by the user** before training.
        - Example: Learning rate, number of trees in a random forest, number of layers in a neural network.
- Test Data: To estimate the **generalization** performance of our model on future (unseen) data.



1. Holdout Method

if your dataset is very large

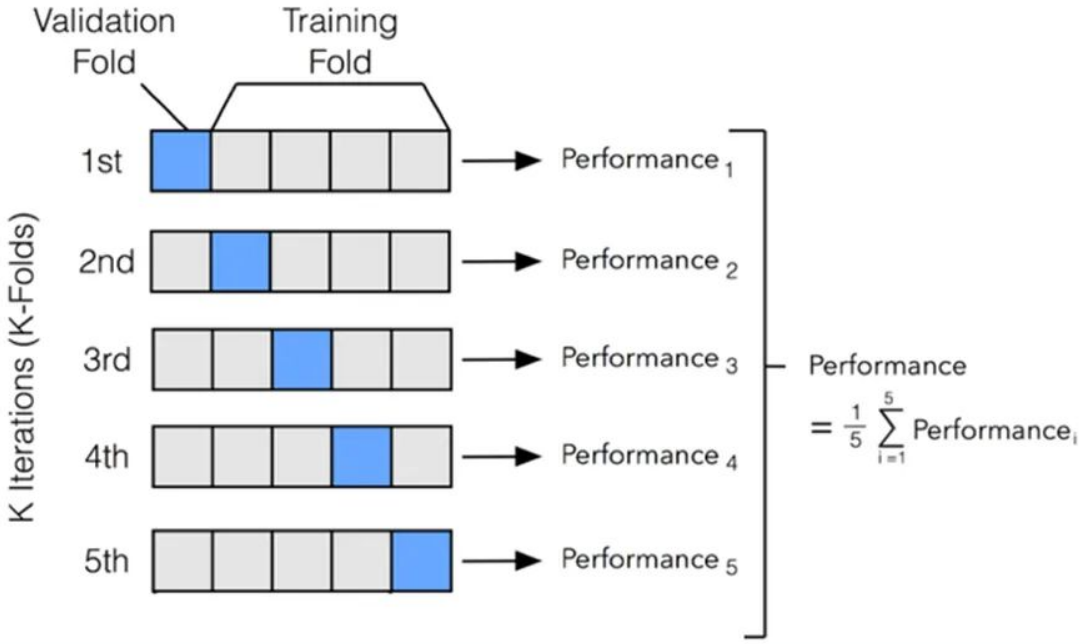
- **Splitting the data to train, test and validation set**
- **Use when:** you have **lots of data** and want a fast approach
- **Risk:** sensitive to how the split is made



K-fold Cross-Validation

if your dataset is not large

- **In K-Fold Cross-Validation** Data is split into *K* folds, and the model is trained *K* times, each time using a different fold as the validation set.
- That way, **we reduce the impact of partition randomness on the results.**



PURDUE UNIVERSITY

Leave-One-Out Cross-Validation

For small datasets

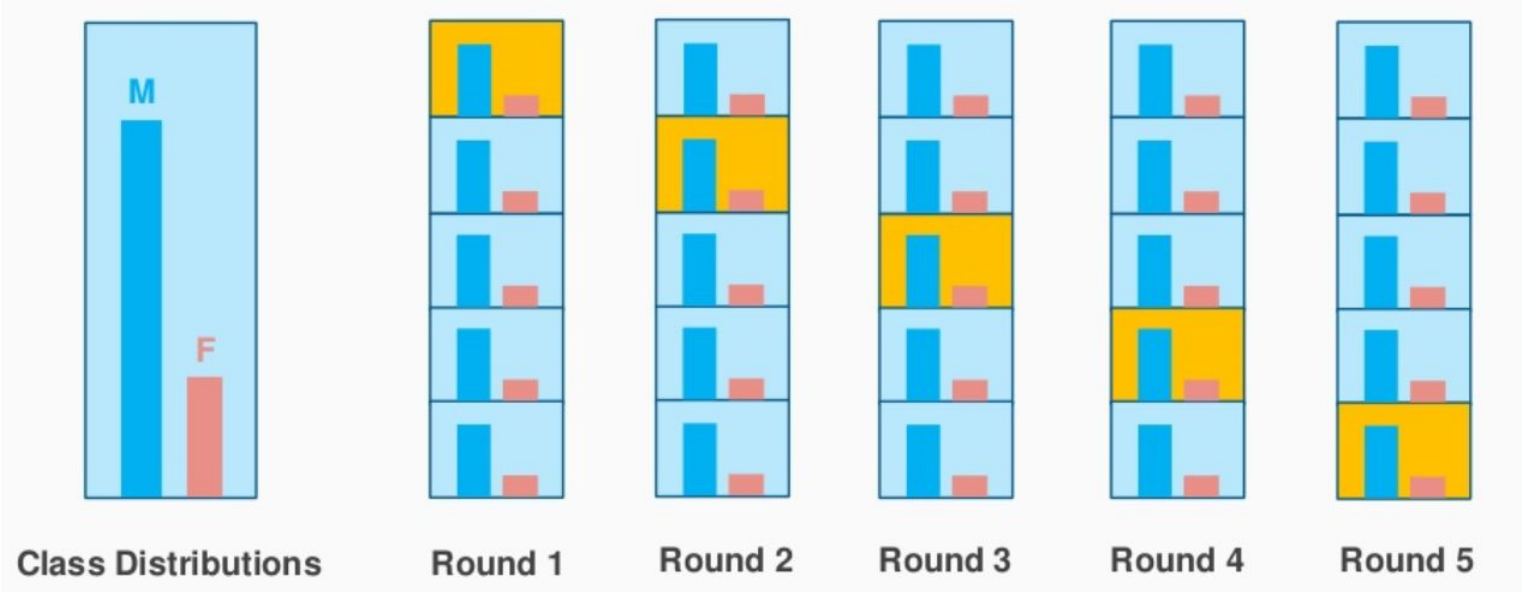
- **Leave-One-Out (LOO) Cross-Validation:** Each sample is used as a validation set once, and all others as training data. Computationally expensive but unbiased.
  a. **LOO is an extreme case of k-fold where K=N.**
  b. When the size is small, LOO is more appropriate since it will use more training samples in each iteration. That will enable our model to learn better representations.
  c. Very computationally expensive

PURDUE UNIVERSITY

Stratified K-Fold Cross-Validation

For imbalance data

- **Stratified K-Fold:** Same as K-Fold but ensures but Ensures class distribution is preserved **in each fold** (useful for imbalanced datasets).
- Use for: classification with imbalanced labels



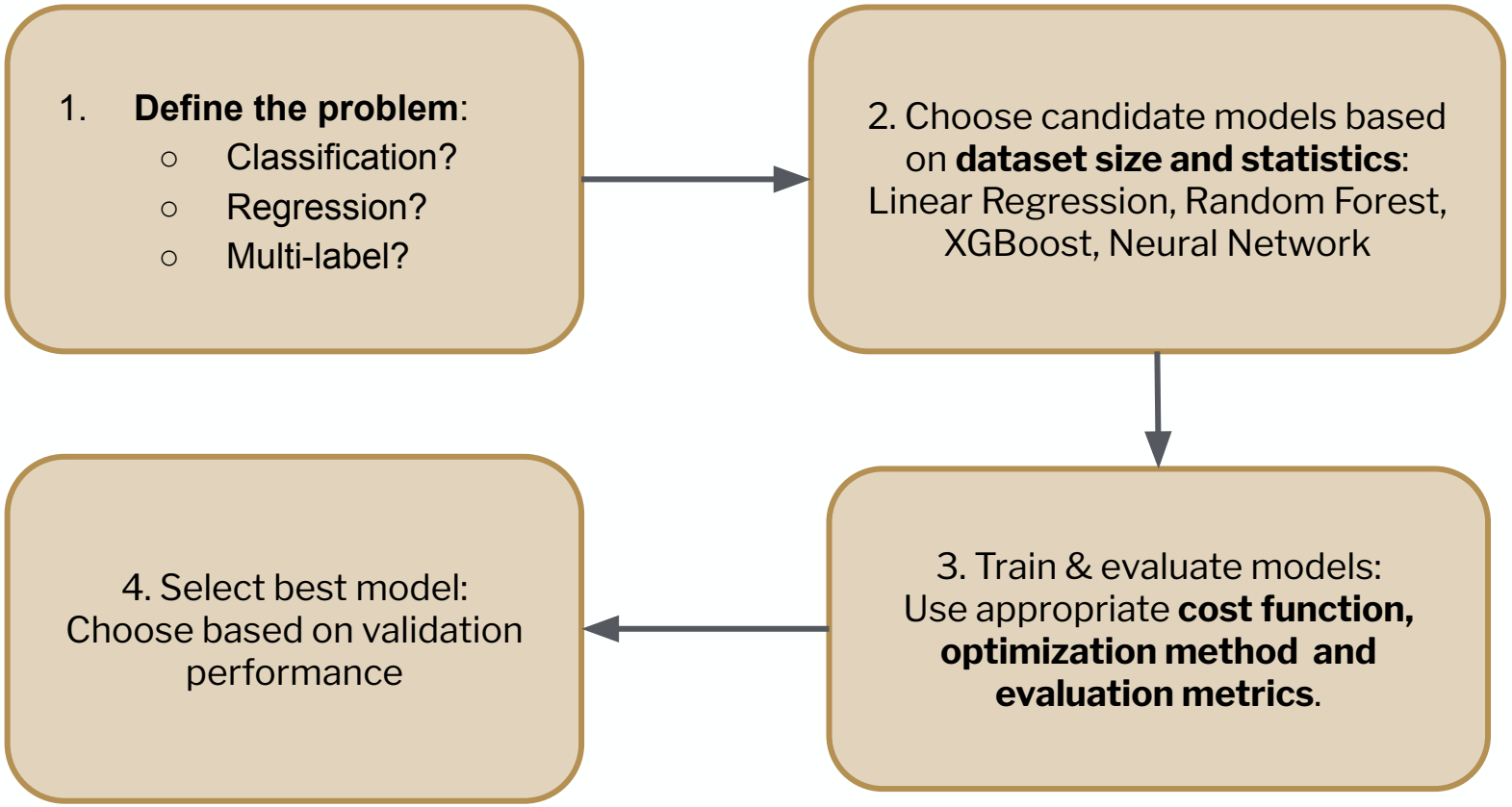
PURDUE UNIVERSITY

Outline

1. Definitions: AI, ML, DL
    a. What is AI, but not ML
    b. ML categories
2. ML WorkFlow
    a. Data Partitioning
    b. Training
        i. cost function
        ii. optimization method selection
        iii. under/Overfitting or Bias/Variance Tradeoff
            1. regulating the loss function
            2. ensemble Learning
    c. Validation and test
        i. evaluation metric selection
3. Basic ML Models
4. Deep Learning

Model Selection

Model selection is the process of **choosing the best-performing model** from a set of candidate **algorithms or configurations** based on performance on **validation data**.



Outline

1. Definitions: AI, ML, DL
    a. What is AI, but not ML
    b. ML categories
2. ML WorkFlow
    a. Data Partitioning
    b. Training
        i. Objective/cost /loss function
        ii. Optimization method
        iii. under/Overfitting or Bias/Variance Tradeoff
            1. regulating the loss function
            2. ensemble Learning
    c. Validation and test
        i. evaluation metric selection
3. Basic ML Models
4. Deep Learning

Objective/Cost Function

- An **objective function (cost function)** is a mathematical function that quantifies the **difference between predicted outputs and true target values**.
- It is the function that a learning algorithm seeks to **minimize** (or sometimes maximize) during training.
  - Maximize the posterior probabilities (e.g., naive Bayes)
  - Maximize the total reward/value function (reinforcement learning)
  - Maximize information gain/minimize child node impurities (decision tree classification)
  - Minimize a mean squared error cost (or loss) function (linear regression, adaptive linear neurons, ...)
  - Maximize log-likelihood or minimize cross-entropy loss (or cost) function
  - Minimize hinge loss (support vector machine)

Objective Functions

- $\mathcal{X} \subseteq \mathbb{R}^d$ be the **feature space** (input domain)
- $\mathcal{Y} \subseteq \mathbb{R}^k$ be the **target space** (labels or outputs)
- $f_\theta : \mathcal{X} \to \mathcal{Y}$ be a **parametric model** with parameters $\theta \in \Theta$
- $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ be a training dataset
- $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ be a **loss function** comparing predicted output $\hat{y}_i = f_\theta(x_i)$ with true target y_i

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(f_\theta(x_i), y_i)$$

Cost/Loss Functions

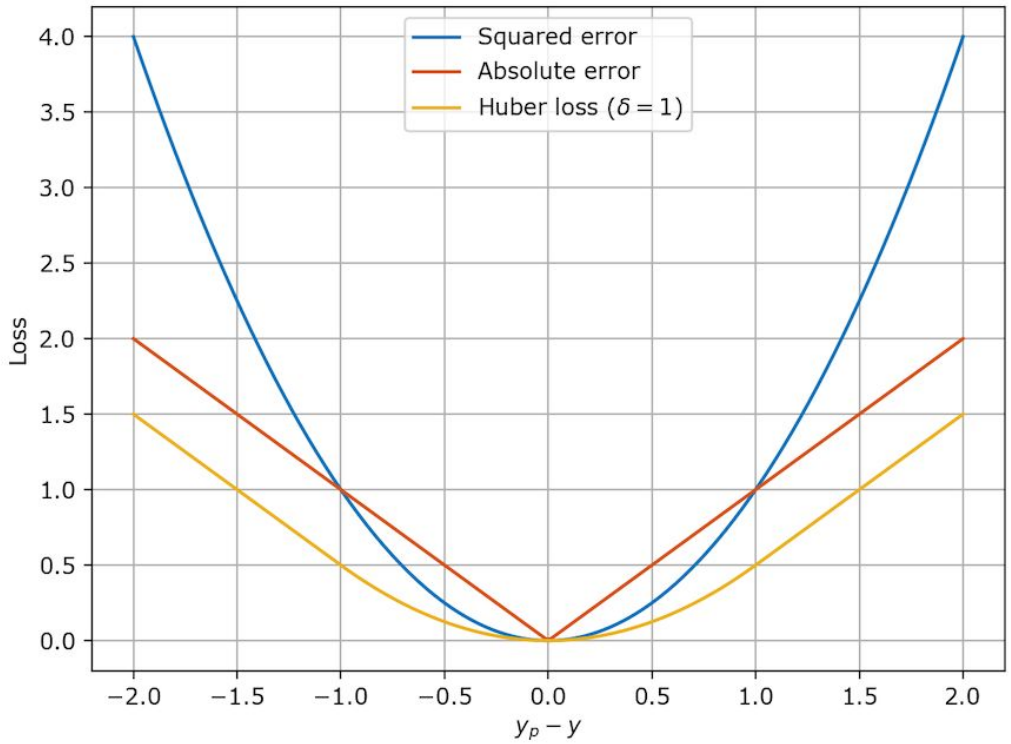
Loss Function	Applicability to Classification	Applicability to Regression	Sensitivity to Outliers
Mean Squared Error (MSE)	✖	✔	High
Mean Absolute Error (MAE)	✖	✔	Low
Cross-Entropy	✔	✖	Medium
Hinge Loss	✔	✖	Low
Huber Loss	✖	✔	Medium
Log Loss	✔	✖	Medium

Regression loss

$$\mathcal{L}_{\mathcal{MAE}} = \frac{1}{N}\sum_{i=1}^{N}|y_i - f(x_i)|$$

$$\mathcal{L}_{\mathcal{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2$$

$$\mathcal{L}_{\text{Huberloss}} = \begin{cases} \frac{1}{2}(y_i - f(x_i))^2 & : \ |y_i - f(x_i)| \leq \delta \\ \delta(|y_i - f(x_i)| - \frac{1}{2}\delta) & : otherwise \end{cases}$$



Classification Loss

$$\mathcal{L}_{\text{zero-one}}(x, y) = \begin{cases} 0, & \text{if } f(x) = y \\ 1, & \text{if } f(x) \neq y \end{cases}$$

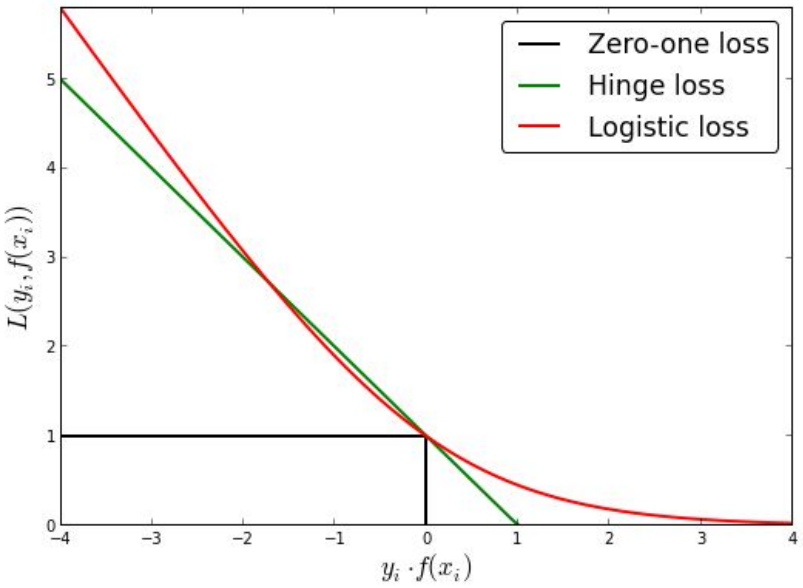
$$\mathcal{L}_{\text{Hinge}} = max(0, 1 - (f(x) \cdot y))$$

$$\mathcal{L}_{\mathcal{BCE}} = \frac{1}{N} \sum_{i=1}^{N} y_i \cdot log(p(x_i)) + (1 - y_i) \cdot log(1 - p(x_i))$$

$$\mathcal{L}_{\mathcal{CE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij}.log(f(x_{ij}))$$

$$N : samples; M : classes$$

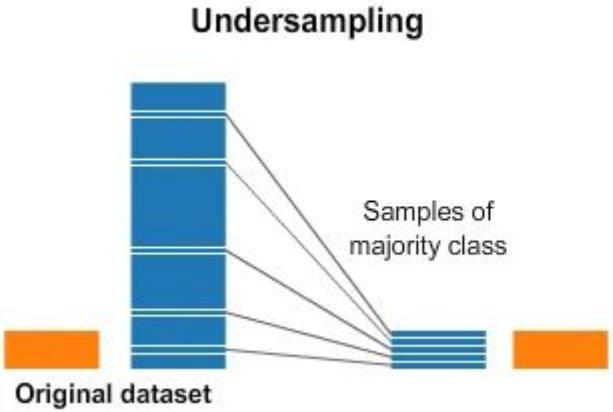
- **o-1 loss** only checks sign.
- **Hinge Loss** is based on the decision margin and works with raw output values,
- **BCE Loss** operates on probabilities and is often used in models that output probabilities (e.g., logistic regression or neural networks with sigmoid activations).

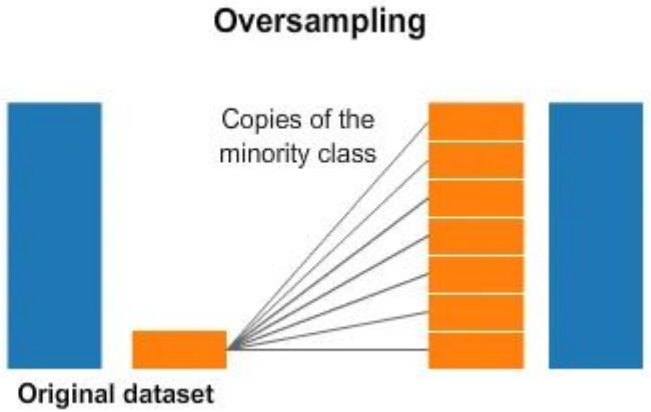


Risk Minimization

Loss is not the only factor

- **Class Imbalance:** Model favors majority class, ignoring rare high-risk cases
- Issues with Loss functions:
  - **False Negatives:** Higher real-world costs (e.g., missing a fraud or cancer diagnosis)
  - **Bias in Decisions:** Unfair treatment in finance, hiring, healthcare, etc.
- **Mitigation:** Re-weighting, resampling, cost-sensitive training





Risk Minimization

In cost-sensitive methods it is more interesting to recognize the positive instances rather than the negative ones.

- For example, in medical domain
  - the cost of misclassifying a non cancerous patient is limited to additional medical tests,
  - while the cost of misdiagnosis of will be fatal as potentially cancerous patients will be considered healthy.

$$L = -\frac{1}{N}\sum[\alpha y \log(\hat{y}) + \beta(1-y)\log(1-\hat{y})]$$

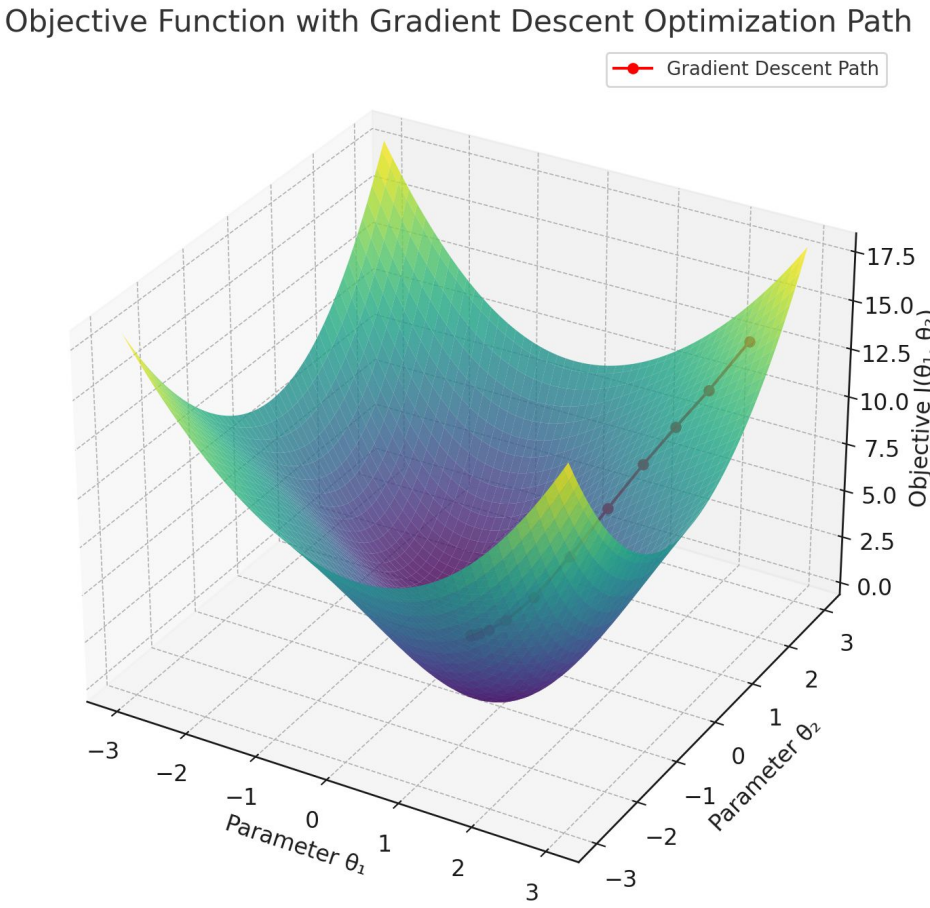
Outline

1. Definitions: AI, ML, DL
    a. What is AI, but not ML
    b. ML categories
2. ML WorkFlow
    a. Data Preprocessing
    b. Data Partitioning
    c. Training
        i. cost function
        ii. optimization method selection
        iii. under/Overfitting or Bias/Variance Tradeoff
            1. regulating the loss function
            2. ensemble Learning
    d. Validation and test
        i. evaluation metric selection
3. Basic ML Models
4. Deep Learning

PURDUE UNIVERSITY

Optimization Methods

- Unconstrained convex optimization
- Constrained convex optimization
- Nonconvex optimization
- Constrained nonconvex optimization
- Combinatorial search,
- greedy search



PURDUE UNIVERSITY

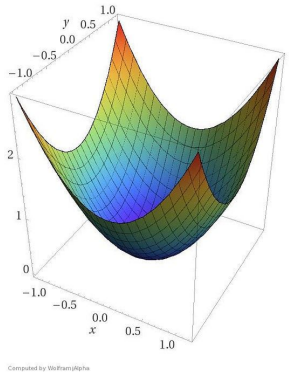
Convex vs non-Convex Function

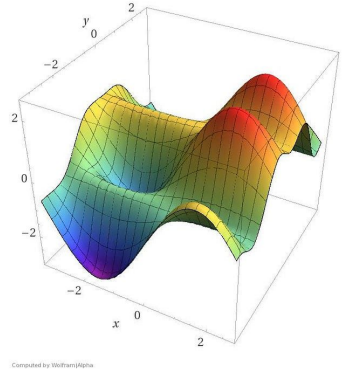
A function f(x) is **convex** if it satisfies:

- **Second Derivative Test**: $f''(x) \geq 0$
- **Jensen's Inequality**: $f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2).$
- **Hessian Test**: For multivariable functions, the Hessian matrix must be **positive semi-definite** (all eigenvalues ≥0)

$$H = \nabla^2 f(x)$$

Convex	non-Convex
Have a **single global minimum**.	Contain **multiple local minima and saddle points**, making optimization harder.
Solvable using **Gradient Descent, Newton's Method, and Interior Point Methods**.	Optimized using **heuristics, SGD variants, …**
Example: **Linear Regression, Support Vector Machines (SVMs)**.	Example: **Deep Neural Networks, Reinforcement Learning**.



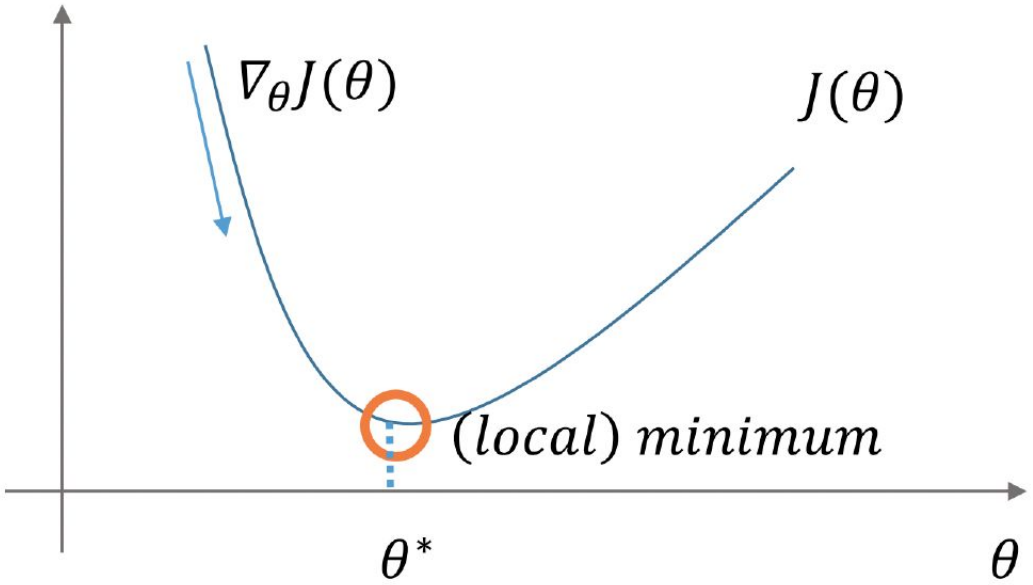


PURDUE UNIVERSITY

Gradient Descent

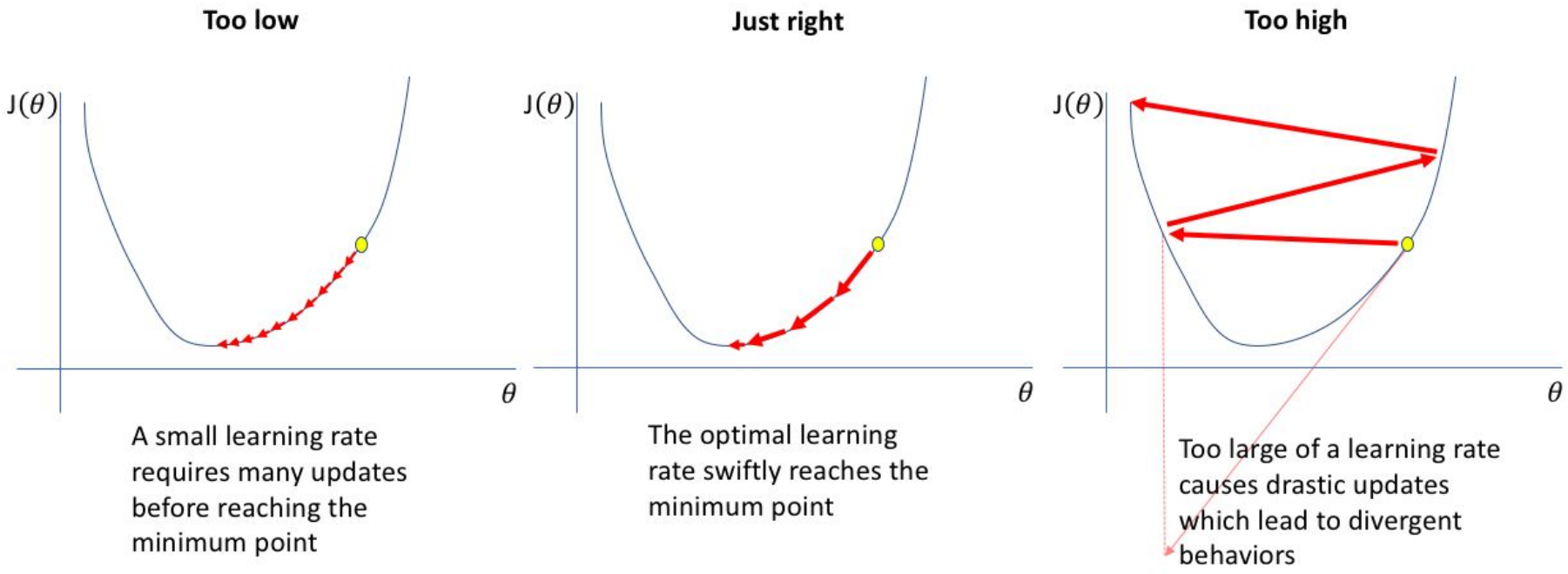
Gradient descent is a way to minimize an objective function J(θ).

- $\theta \in \mathbb{R}^d$: model parameters
- η : learning rate
- $\nabla_\theta J(\theta)$: gradient of the objective function with regard to the parameters
- It updates the parameters in **opposite direction of gradien** $\theta = \theta - \eta \cdot \nabla_\theta J(\theta)$



Gradient Descent

Learning Rate



https://www.jeremyjordan.me/nn-learning-rate/

Gradient Descent Variations

	Method	Pros	Cons
(Batch) Gradient Descent $\theta = \theta - \eta \cdot \nabla_\theta J(\theta)$	Computes gradient with the **entire dataset**.	- Guaranteed to converge to **global minimum for convex** error surfaces - Guaranteed to converge to a l**ocal minimum for non-convex** surfaces.	- Very slow. - computationally expensive - Intractable for datasets that do not fit in memory. - No online learning.
SGD $\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}; y^{(i)})$	Computes update for **each example** x(i)y(i).	- Much faster than batch gradient descent. - Allows online learning.	- High **variance** updates.
Mini-Batch SGD $\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)})$	Performs update for **every mini-batch** of n examples	- Reduces variance of updates.	- Mini-batch size is a hyperparameter.

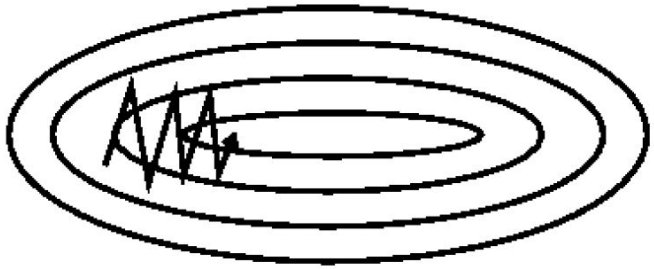
Momentum

- *SGD has trouble navigating ravines.*
- *Momentum [Qian, 1999] helps SGD to accelerate.*
  - *Adds a fraction γ of the update vector of the past step vt-1 to current update vector vt .*

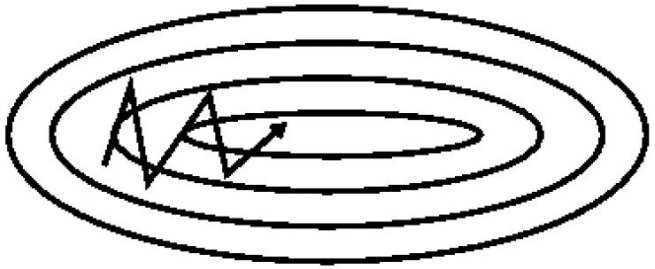
$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$

$$\theta = \theta - v_t$$

  - **Reduces updates** for dimensions whose gradients **change directions.**
  - **Increases updates** for dimensions whose gradients **point in the same directions**.



(a) SGD without momentum



(b) SGD with momentum

Figure: Source: Genevieve B. Orr

Adagrad

- Previous methods: Same learning rate η for all parameters θ .
- Adagrad [Duchi et al., 2011] adapts the learning rate to the parameters (large updates for infrequent parameters, small updates for frequent parameters).
- Adagrad divides the learning rate by the square root of the sum of squares of historic gradients.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$$

Gt ∈ R^{2*2} : diagonal matrix where each diagonal element i ; i is the sum of the squares of the gradients w.r.t. θi up to time step t.

Pros:

Well-suited for dealing with sparse data.

Significantly improves robustness of SGD.

Lesser need to manually tune learning rate.

Cons: Accumulates squared gradients in denominator.

Causes the learning rate to shrink and become in nitesimally small.

Adam

- Adaptive Moment Estimation (Adam) [Kingma and Ba, 2015] also stores running average of past squared gradients vt like Adadelta and RMSprop.
- Like Momentum, stores running average of past gradients mt .

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

- mt : rst moment (mean) of gradients
- vt : second moment (uncentered variance) of gradients
- β_1 , β_2: decay rates

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

PURDUE UNIVERSITY

Outline

1. Definitions: AI, ML, DL
    a. What is AI, but not ML
    b. ML categories
2. ML WorkFlow
    a. Data Preprocessing
    b. Data Partitioning
    c. Training
        i. cost function
        ii. optimization method selection
        iii. under/Overfitting or Bias/Variance Tradeoff
            1. regulating the loss function
            2. ensemble Learning
    d. Validation and test
        i. evaluation metric selection
3. Basic ML Models
4. Deep Learning

Bias-Variance Trade-off

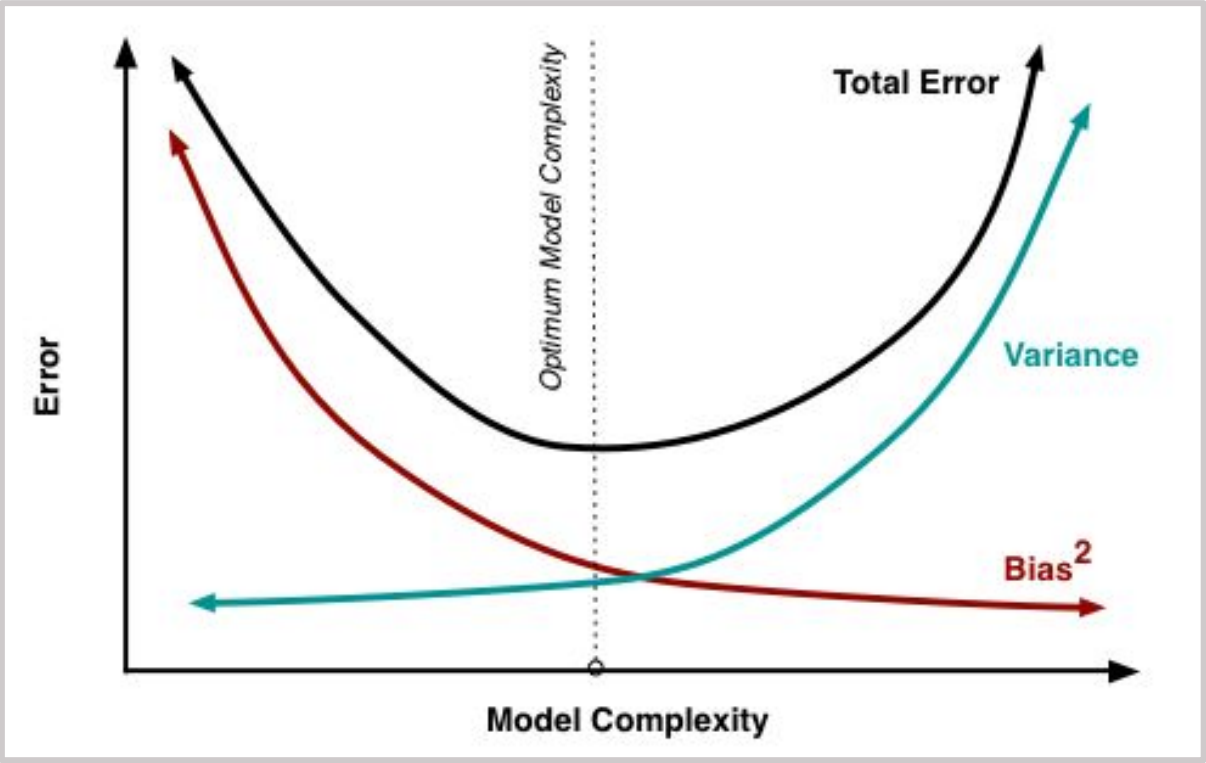
$$\mathbb{E}_{D,\epsilon}\left[\left(\hat{f}_D(x) - y\right)^2\right] = \underbrace{\mathbb{E}_D\left[\left(\hat{f}_D(x) - \mathbb{E}_D[\hat{f}_D(x)]\right)^2\right]}_{\text{Variance}(x)} + \underbrace{\left(\mathbb{E}_D[\hat{f}_D(x)] - f(x)\right)^2}_{\text{Bias}^2(x)} + \underbrace{\mathbb{E}_\epsilon\left[\epsilon^2\right]}_{\text{Noise}(x)}$$

- $y = f(x) + \epsilon$: true function plus noise.
- $\hat{f}_D(x)$: prediction from model trained on dataset D.
- $\epsilon \sim \mathcal{N}(0, \sigma^2)$: noise, assumed to have zero mean.

Term	Meaning
Variance	How much predictions vary across different training datasets (Overfitting and Model sensitivity)
Bias²	How far the average prediction is from the true function (Underfitting and Model assumption)
Noise	Irreducible error caused by inherent randomness in the data distribution and feature representation. an aspect of data, not model.

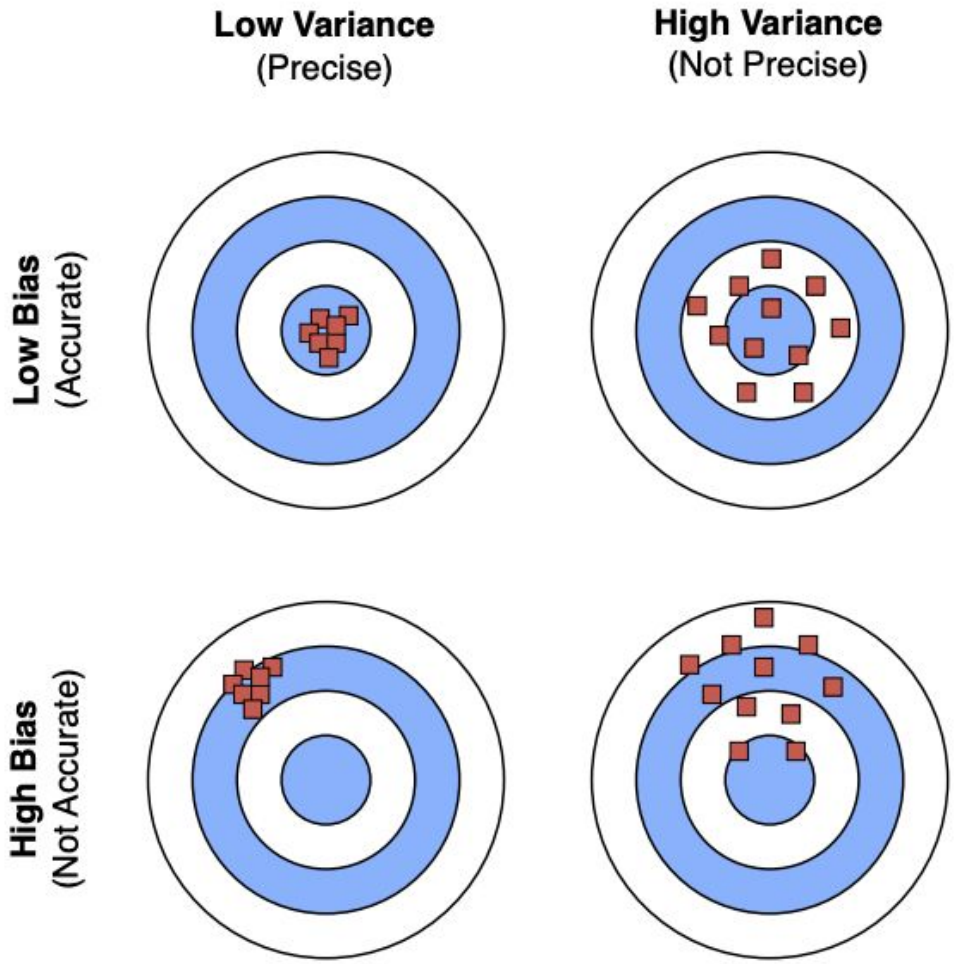
Under/Overfitting

More complex models overfit while the simpler models underfit.



source:https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote12.html

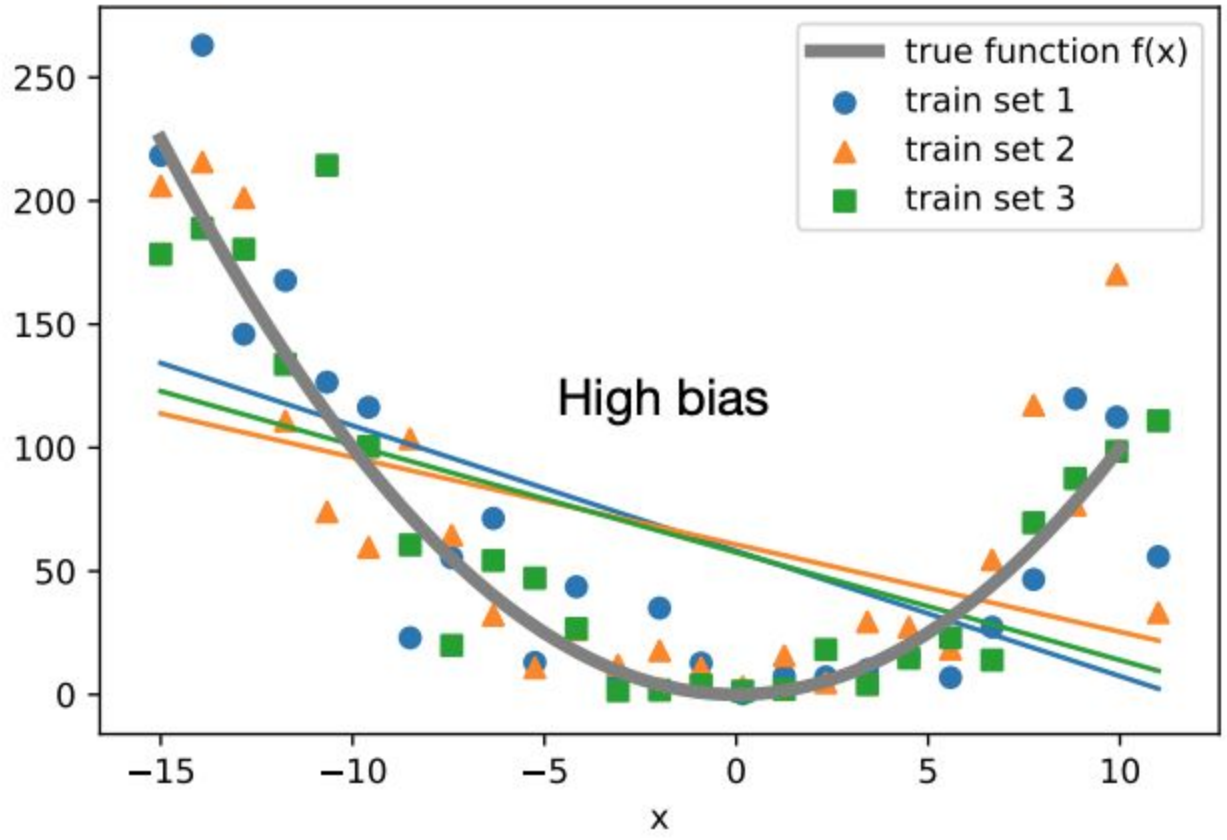
Bias-Variance Trade-off



source:
https://sebastianraschka.com/pdf/lecture-notes/stat479fs18/08_eval-intro_notes.pdf

High Bias Issue

Change of training data, doesn't help the model simplicity and weakness.

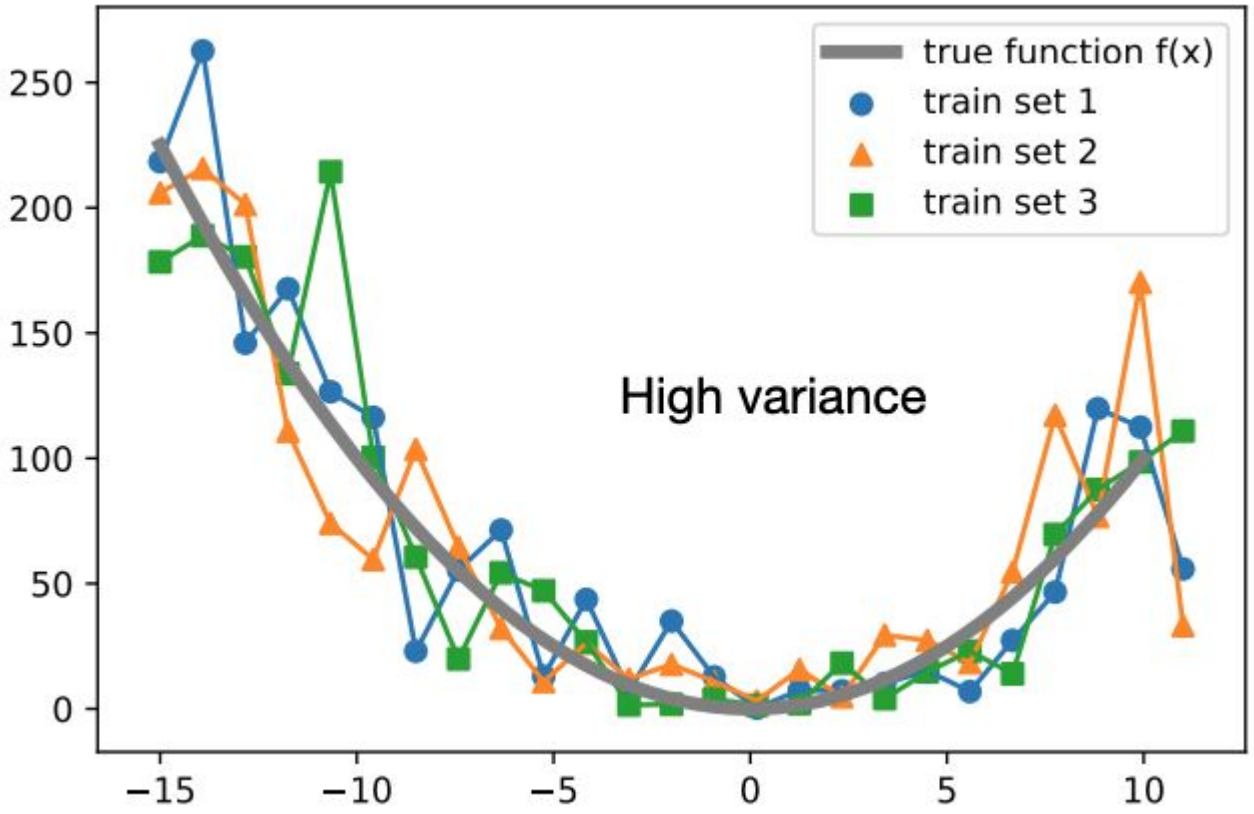


https://sebastianraschka.com/pdf/lecture-notes/stat479fs18/08_eval-intro_notes.pdf

PURDUE UNIVERSITY

High Variance Issue

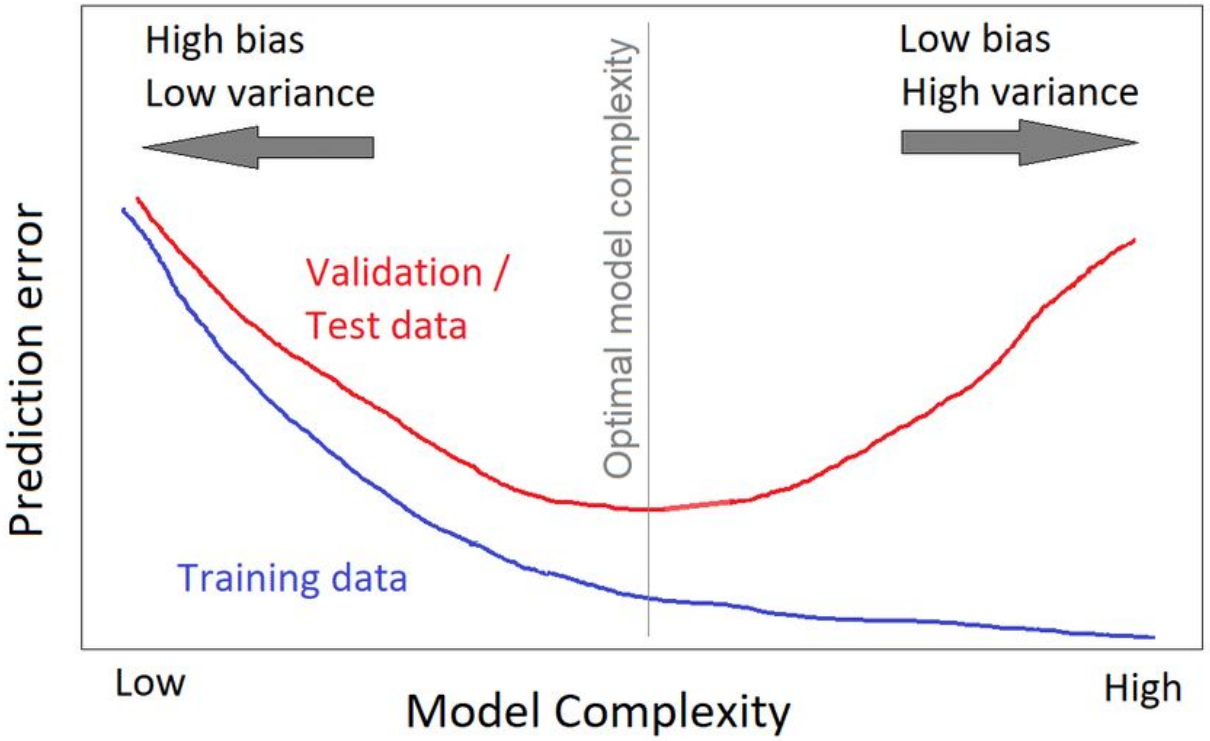
Model is very sensitive to any change of training data.



https://sebastianraschka.com/pdf/lecture-notes/stat479fs18/08_eval-intro_notes.pdf

PURDUE UNIVERSITY

Issue	Symptoms	Causes
Underfitting	- High error on both training and validation sets -Poor learning	- Model too simple - Inadequate training time - Features not informative
Overfitting	Training error is much lower than validation error	- Model too complex - Training too long - Lack of regularization



source:https://www.cs.cornell.edu/courses/cs4780/2024sp/lectures/pdfs/lecturenote11.pdf

Regularization

Regularization is a technique used to **prevent overfitting** by adding constraints to a machine learning model.

1. **L1 Regularization (Lasso Regression)**
   - Adds the absolute value of weights (**L1 norm**) as a penalty.
   - Encourages sparsity by shrinking some weights to zero (feature selection).
2. **L2 Regularization (Ridge Regression)**
   - Adds the squared value of weights (**L2 norm**) as a penalty.
   - Prevents large weights, reducing model complexity while keeping all features.
3. **Elastic Net**
   - Combines L1 and L2 regularization for a balance of sparsity and smoothness.
4. **Dropout & Early Stopping**
   - **Dropout:** Randomly ignores some parameters during training.
   - **Early Stopping:** Stops training when validation performance stops improving.

Regularization

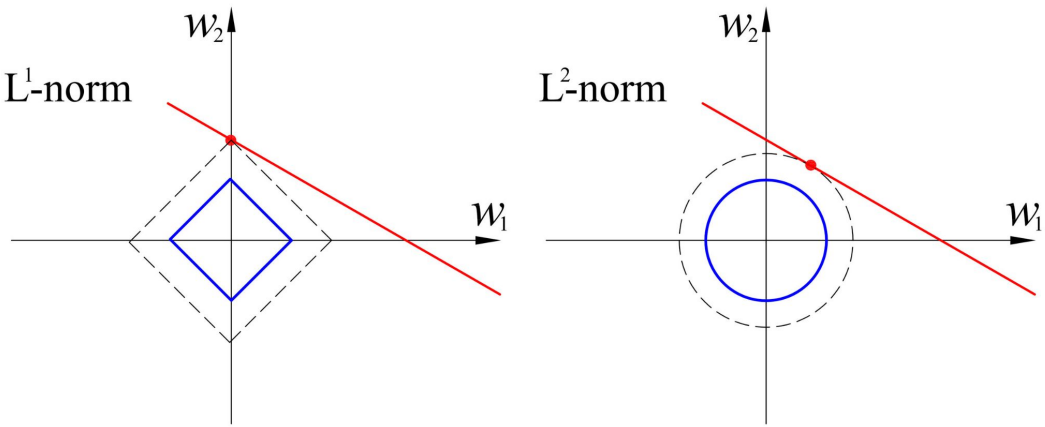
L1 Regularization

$$\text{Modified loss function} = \text{Loss function} + \lambda \sum_{i=1}^{n} |W_i|$$

L2 Regularization

$$\text{Modified loss function} = \text{Loss function} + \lambda \sum_{i=1}^{n} W_i^2$$

L1 prefers some sparser values for W, and L2 prefers smaller values for W.



PURDUE UNIVERSITY

Ensemble Learning

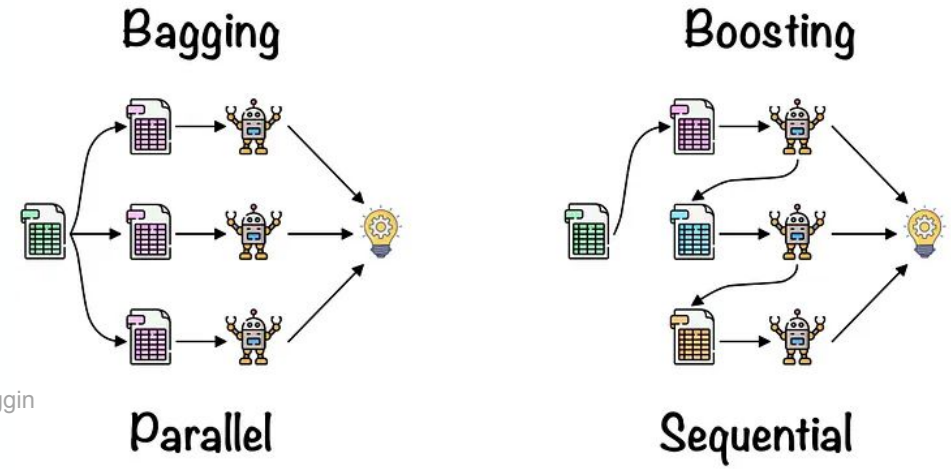
- Ensemble Learning **combines multiple models** to improve performance, reduce overfitting, and enhance generalization.
- Instead of relying on a single model, it aggregates predictions from multiple models to achieve **higher accuracy and robustness**.

Bagging

- Trains multiple **independent** models on different random subsets of data and averages predictions.
- Reduce **variance** and prevent overfitting.
- Models are trained in **parallel** and independently.

Boosting

- Trains models **sequentially**, where each model corrects the errors of the previous one.
- Reduce **bias** and improve weak learners.
- Models are trained **sequentially**, each learning from previous mistakes.



PURDUE UNIVERSITY

source:https://medium.com/@roshmitadey/bagging-v-s-boosting-be765c970fd1

Outline

1. Definitions: AI, ML, DL
    a. What is AI, but not ML
    b. ML categories
2. ML WorkFlow
    a. Data Preprocessing
    b. Data Partitioning
    c. Training
        i. cost function
        ii. optimization method selection
        iii. under/Overfitting or Bias/Variance Tradeoff
            1. regulating the loss function
            2. ensemble Learning
    d. Validation and test
        i. evaluation metric selection
3. Basic ML Models
4. Deep Learning

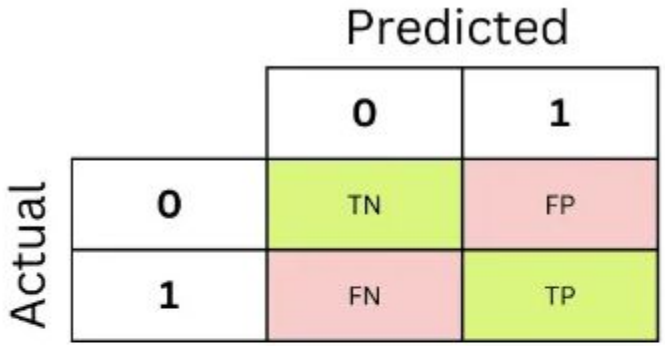
Training/Validation/Test measures

Cost/Objective/Loss Function	Validation Metric
• Used for Model training (optimization) • Guides the optimizer to update weights and Learns model parameters using training data	• Used for Model selection validation set • Used for final performance reporting on test set

• **Examples:**
  ○ Classification: `Cross-Entropy Loss`
  ○ Regression: `Mean Squared Error (MSE)`
  ○ Ranking: `Hinge loss`, `Triplet loss`
	• **Examples:**
	  ○ Classification: `Accuracy`, `Precision`, `Recall`, `F1 Score`, `AUC`
	  ○ Regression: `MAE`, `RMSE`
	  ○ Ranking: `NDCG`, `MAP`, `MRR`

• Differentiable, used for training (e.g. cross-entropy).	Human-readable evaluation (e.g. accuracy, F1-score).

Validation Metrics: Confusion Matrix



- **Class imbalance** → prefer **Precision/Recall** over Accuracy
- Avoid false negatives → Use **Recall**
- Avoid False Positive → Use **Precision**

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

$$Recall = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

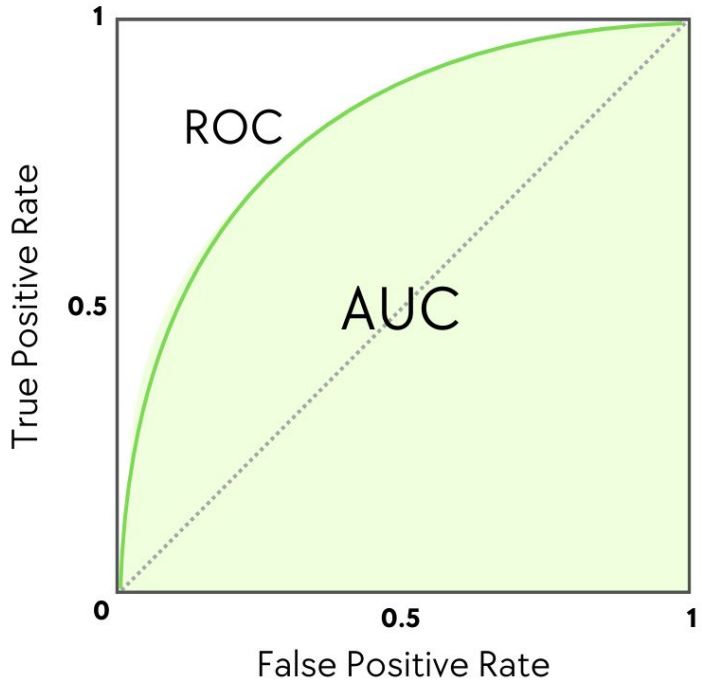
Validation Metrics

Receiver Operating Characteristic (ROC): Curve plotting TPR vs FPR across all thresholds

Area Under Curve (AUC): Area under the ROC curve, summarizing model's discrimination power

$$\text{True Positive Rate} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative (FN)}}$$

$$\text{False Positive Rate} = \frac{\text{False Positive (FP)}}{\text{False Positive (FP)} + \text{True Negative (TN)}}$$



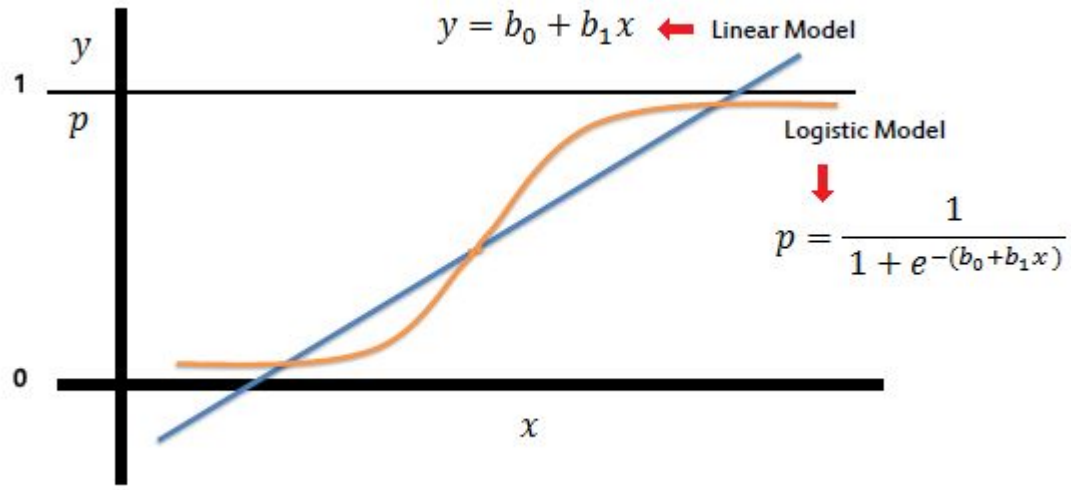
PURDUE UNIVERSITY

Outline

1. Definitions: AI, ML, DL
    a. What is AI, but not ML
    b. ML categories
2. ML WorkFlow
    a. Data Preprocessing
    b. Data Partitioning
    c. Training
        i. cost function
        ii. optimization method selection
        iii. under/Overfitting or Bias/Variance Tradeoff
            1. regulating the loss function
            2. ensemble Learning
    d. Validation and test
        i. evaluation metric selection
3. Basic ML Models
4. Deep Learning

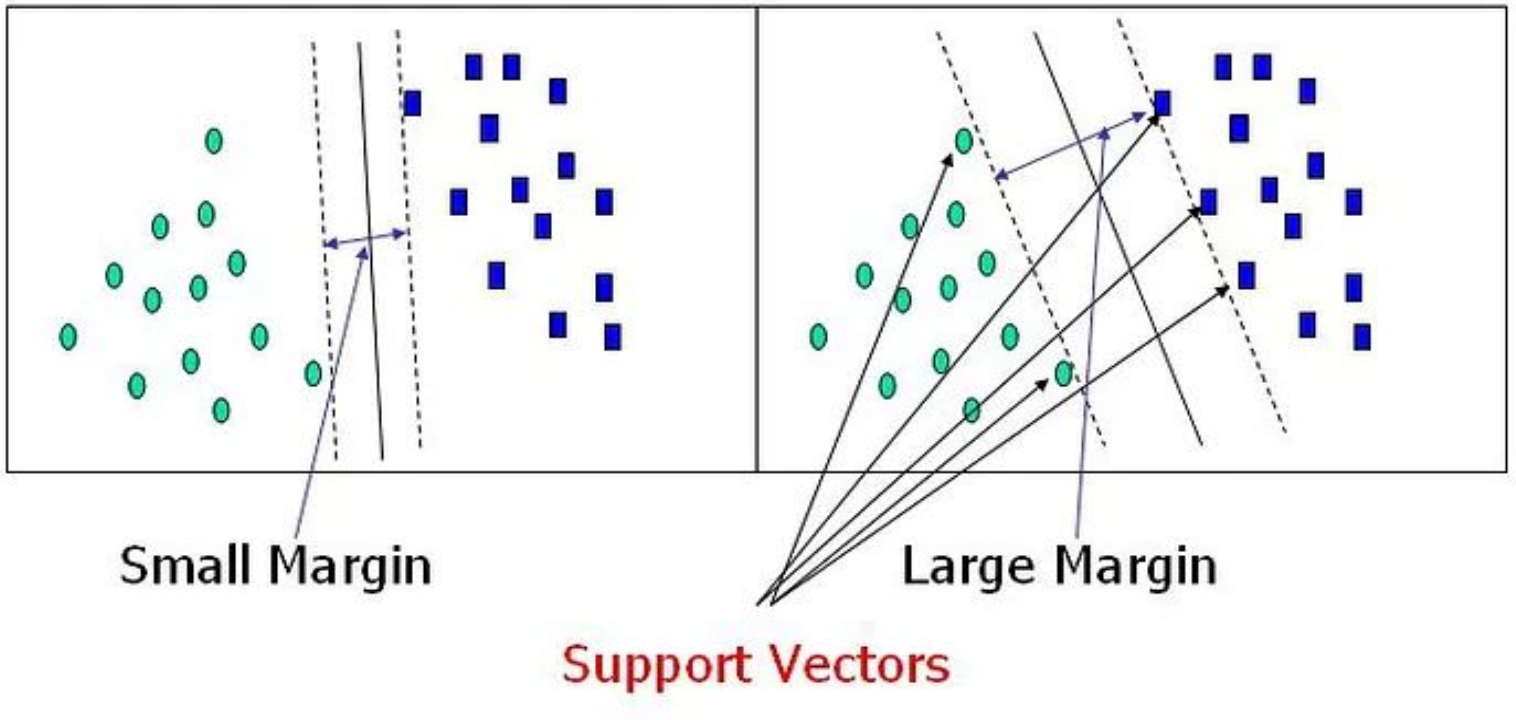
Linear Regression vs Logistic regression

- **Linear Regression (blue line)** produces continuous values, which can extend beyond 0 and 1.
- **Logistic Regression (orange sigmoid curve)** maps inputs to probabilities between **0 and 1**, making it suitable for classification.
- This shows why **logistic regression is ideal for binary classification**, whereas linear regression is used for continuous prediction



Support Vector Machine(SVM)

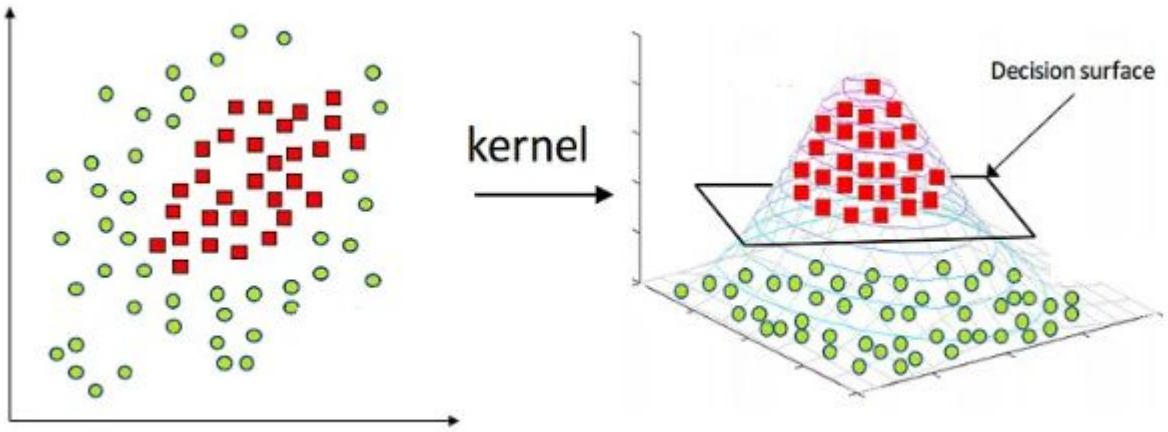
- It **Maximizes Margin** by finding the optimal hyperplane that maximizes the margin between classes.
- **Support Vectors** are the Data points closest to the decision boundary that influence the model.



PURDUE UNIVERSITY

Support Vector Machine(SVM)

- **Kernel Trick** Extends SVM to handle non-linearly separable data.
- The **kernel trick** transforms input features into a **higher-dimensional space**, making SVM more powerful for complex decision boundaries
  - Polynomial Kernel,
  - RBF Kernel,
  - …

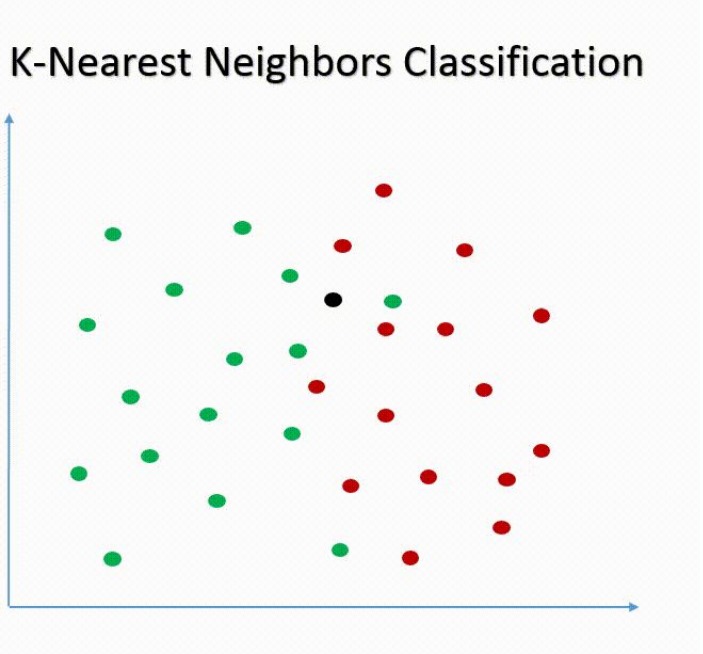


K Nearest Neighbour (KNN)

Lazy Learning Algorithms

- **Lazy Learning Algorithm**:
  - No training phase;
  - it stores the entire dataset and classifies new data based on similarity.
- **Choice of K Matters**: A small K may lead to overfitting, while a large K smooths the decision boundary.
- We can use various **distance metrics**,

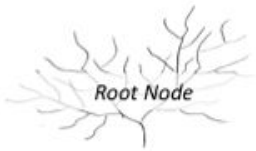
  including Euclidean, Manhattan, cosine similarity,
- **For regression**, use average of the K neighbours labels,
- **For classification,** use majority voting.

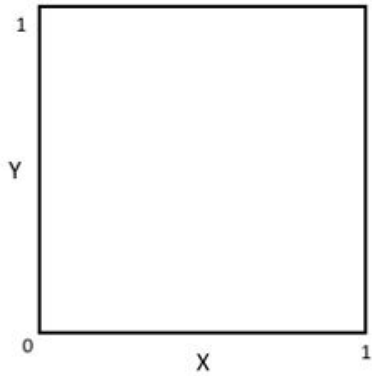


https://machinelearningknowledge.ai/

Decision Tree

- A flowchart-like structure that splits data into branches based on features.
  - Each internal node: a decision on a feature **(X and Y).**
  - Each leaf node: a Prediction **(Grey or Green)**.
- In each step Identify the feature that splits data points into two groups that are most homogeneous (maximizing information gain/minimizing entropy).
- Stop partitioning of the samples in each node when:
  - Data points at the leaf are all of the same predicted category/value
  - The leaf contains less than five data points
  - Further branching does not improve homogeneity beyond a minimum threshold



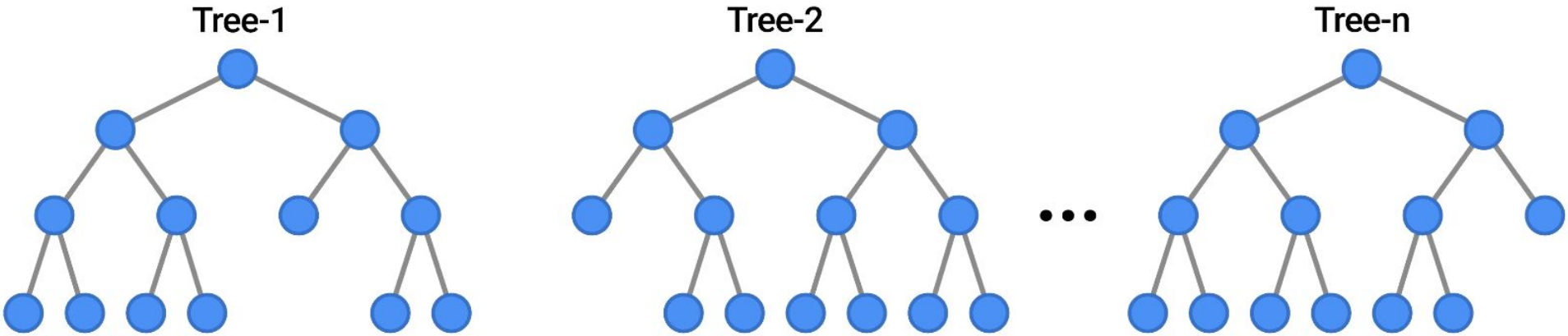


PURDUE UNIVERSITY

https://algobeans.com/2016/07/27/decision-trees-tutorial/

Random Forests

- Random Forest is an ensemble over several decision trees.
- Final decision using majority voting.



https://medium.com/@hakanyuceturk/from-roots-to-leaves-exploring-decision-tree-algorithms-random-forests-and-related-techniques-in-c7492ba4260b

Outline

1. Definitions: AI, ML, DL
    a. What is AI, but not ML
    b. ML categories
2. ML WorkFlow
    a. Data Preprocessing
    b. Data Partitioning
    c. Training
        i. cost function
        ii. optimization method selection
        iii. under/Overfitting or Bias/Variance Tradeoff
            1. regulating the loss function
            2. ensemble Learning
    d. Validation and test
        i. evaluation metric selection
3. Basic ML Models
4. Deep Learning

What is a Neuron?

Input: It is the set of features, For example, the input in object detection can be an array of pixel values pertaining to an image.

Weight : Its main function is to give importance to those features that contribute more towards the learning.

Bias: like as a constant in a linear function.

Transfer function: it combines multiple inputs into one output value using a simple summation of all the inputs.

Activation Function: It introduces non-linearity in the working of perceptrons. Without this, the output would just be a linear combination of input values.

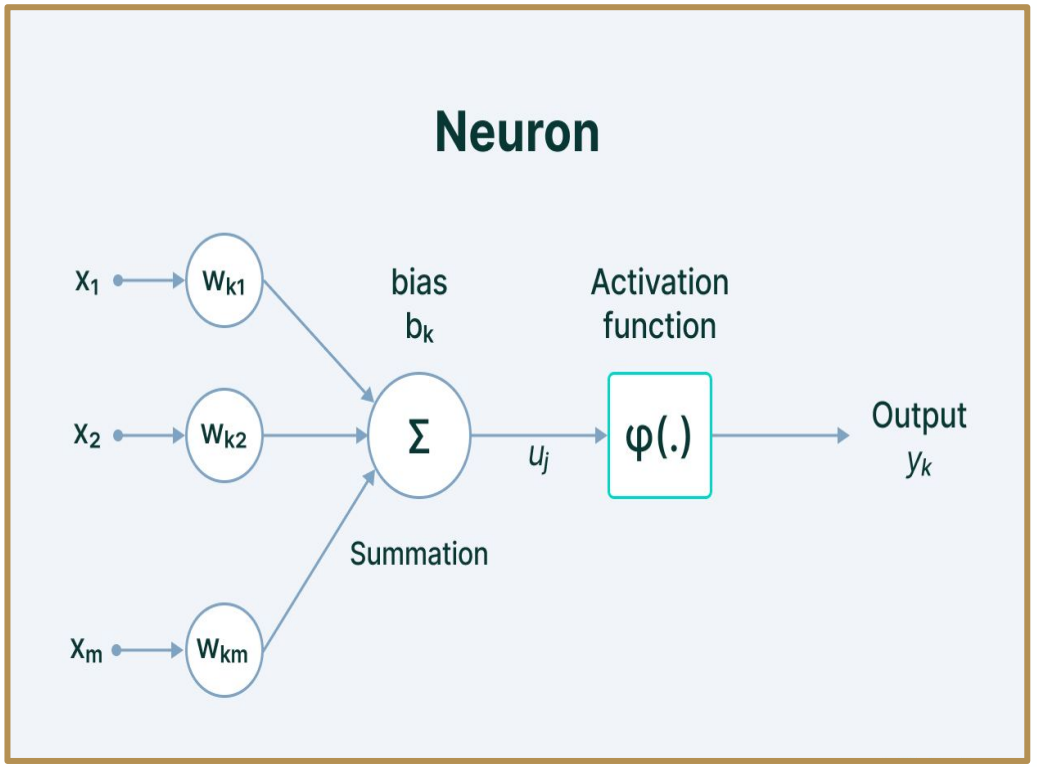
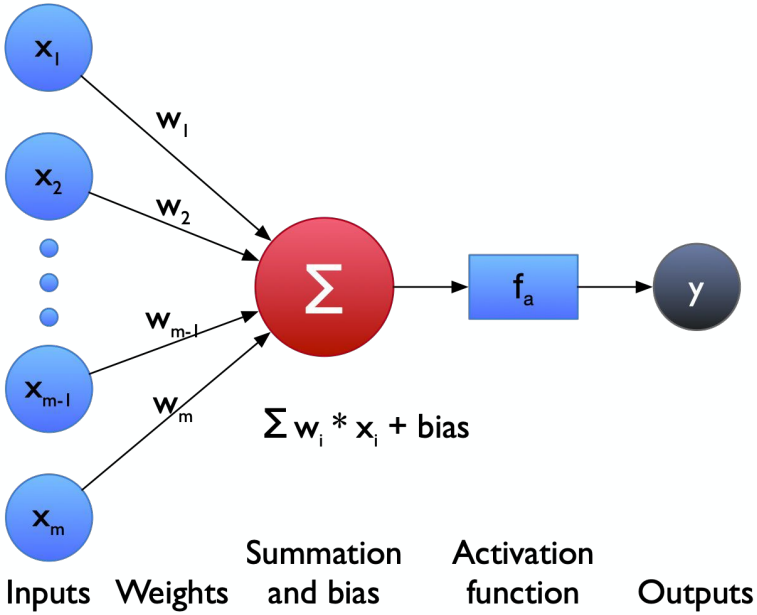
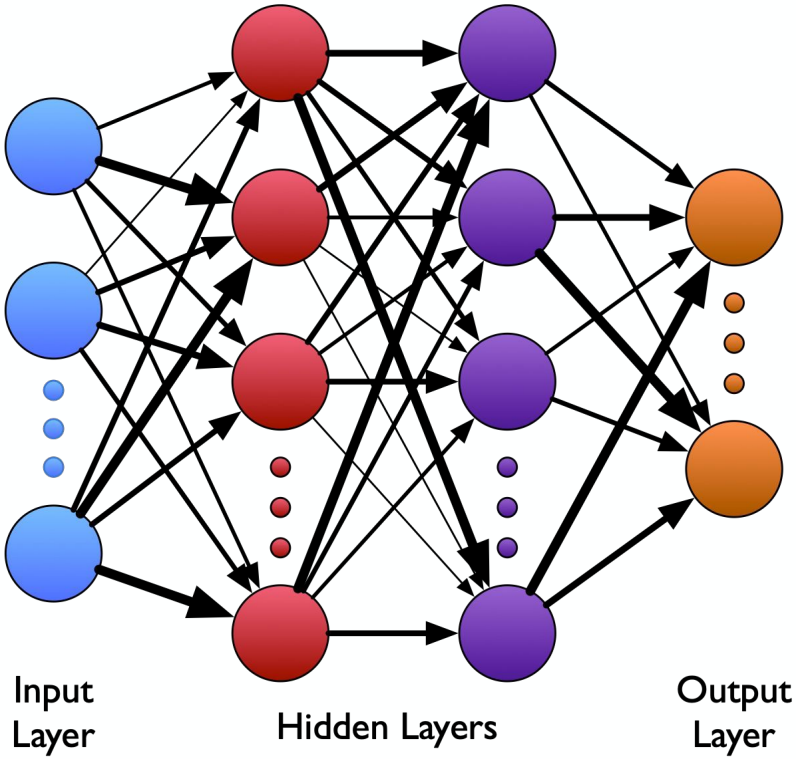


Photo taken from [2]

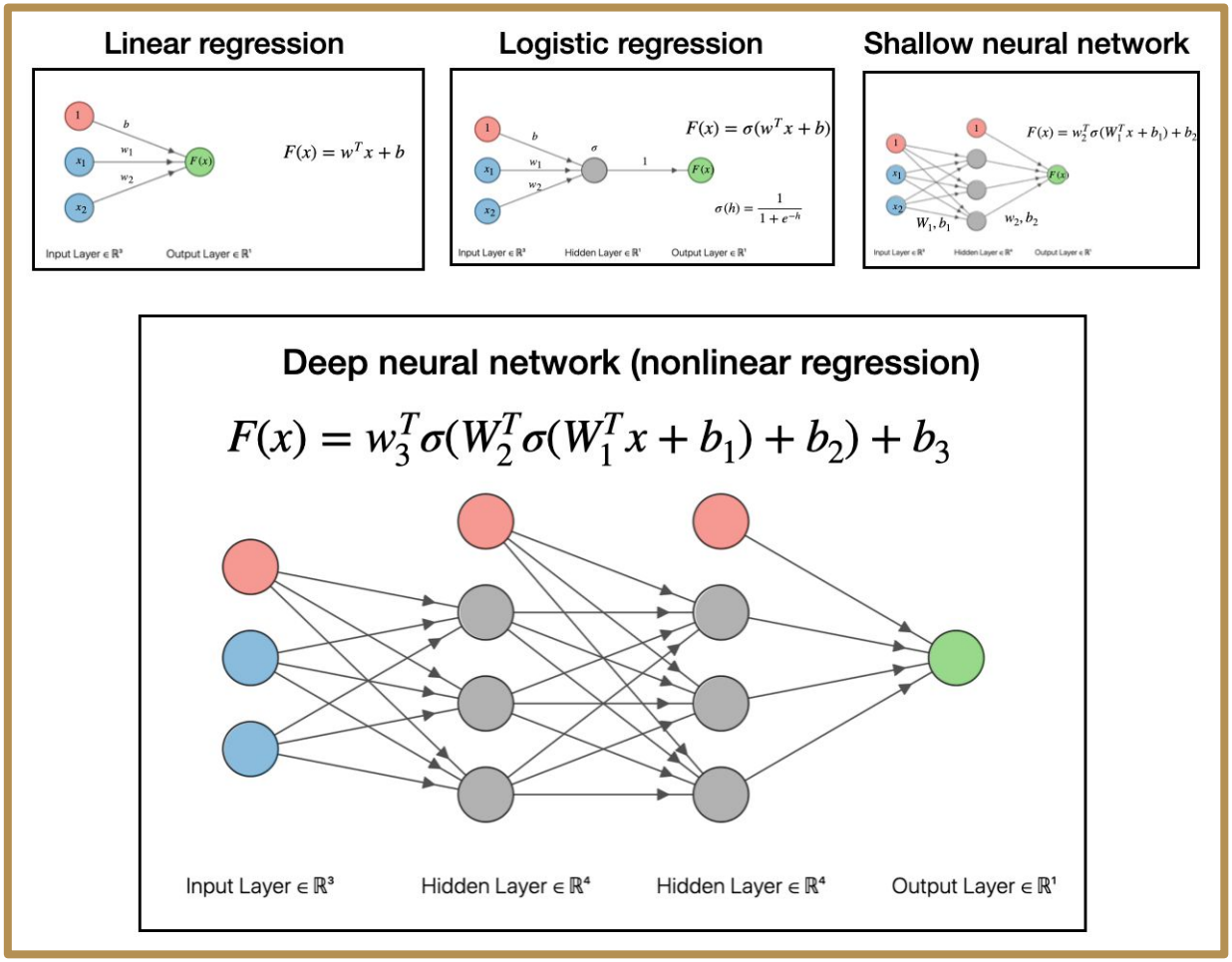
PURDUE UNIVERSITY

From Neuron to Deep Neural Network





Neural Network



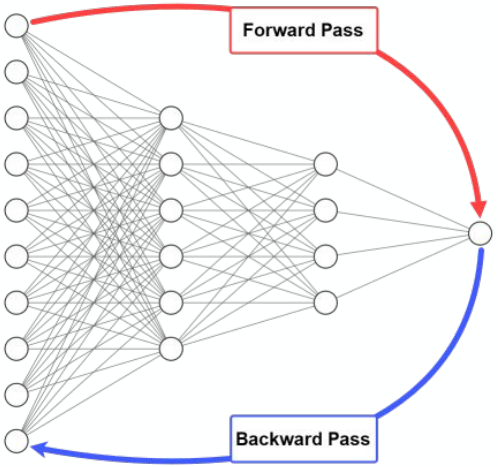
PURDUE UNIVERSITY

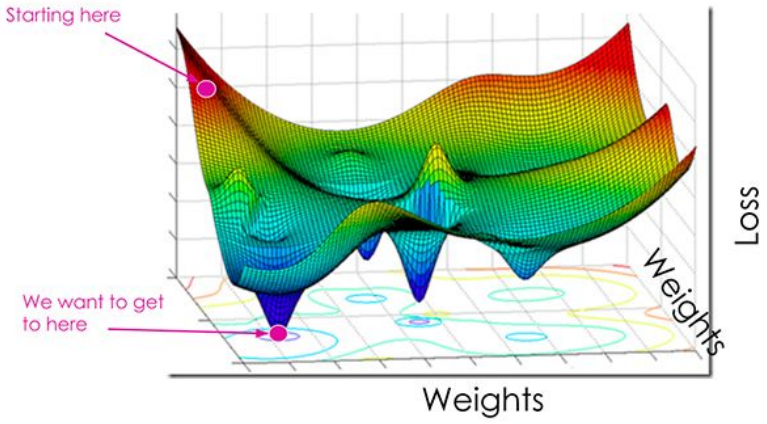
Krishna, S.T. and Kalluri, H.K., 2019. Deep learning and transfer learning approaches for image classification. International Journal of Recent Technology and Engineering (IJRTE), 7(5S4), pp.427-432.

Deep Learning and Backpropagation

A deep network has a huge parameter space, so that:

- Needs More training data
- Prone to overfitting and less generalizable
- Needs special initialization and optimization methods to avoid vanishing/exploding gradient
- Needs strong hardware for training and inference





Deep Learning History

Year	Contributor	Contribution
300 BC	Aristotle	introduced Associationism, started the history of humans attempt to understand brain.
1873	Alexander Bain	introduced Neural Groupings as the earliest models of neural network, inspired Hebbian Learning Rule.
1943	McCulloch & Pitts	introduced MCP Model, which is considered as the ancestor of Artificial Neural Model
1949	Donald Hebb	Considered as the father of neural networks, introduced Hebbian Learning Rule, which lays the foundation of modern neural network.
1956	**John McCarthy**	Together with Minsky held Dartmouth Conference named "**Artificial Intelligence**".
1958	Frank Rosenblatt	Introduced the **first perceptron**, which highly resembles modern perceptron.
1969	Minsky & Papert	They proved that single-layer perceptrons couldn't solve non-linearly separable problems like XOR, limiting their power.
1974	Paul Werbos	Introduced **Backpropagation**
1980	Kunihiko Fukushima	Introduced Neocogitron, which inspired **Convolutional Neural Network**
1982	**Minsky at "The Society of Mind"**	Warned that early AI methods, including expert systems, were not as powerful or general as their proponents claimed.

Deep Learning History

Year	Contributor	Contribution
1986	Michael I. Jordan	Defined and introduced **Recurrent Neural Network**
	Hinton & Rumelhart	**Backpropagation for MLP:** This solved **Minsky & Papert's critique** that perceptrons were too limited.
1989	**Yann Lecun**	Introduced CNNs for **handwritten digit recognition**
1997	Hochreiter & Schmidhuber	Introduced **LSTM**, solved the problem of **vanishing gradient** in recurrent neural networks
1999	Nvidia	Developed the world's first GPU
2006	**Geoffrey Hinton**	Introduced Deep Belief Networks, also introduced **layer-wise pretraining** technique, **opened current deep learning era**.
2006		Researchers started implementing deep learning models on GPUs.
2012	**Geoffrey Hinton**	Introduced Dropout, to avoid **overfitting** and improving **generalization**.
2017		The **Transformer** model replaced CNNs and RNNs in NLP tasks.
2020		**Vision Transformers (ViTs)** challenged CNN dominance in vision tasks.

Deep Learning and LLM Lecture Series

To learn more on deep learning and Large Language Models (LLMs), please refer to our course: *https://www.rcac.purdue.edu/training/deep-learning-series*

PURDUE UNIVERSITY

Thank You

PURDUE UNIVERSITY