Introduction to Machine Learning: Modeling, Training and Evaluation

Evan Misshula

June 10, 2025

g

End to end process

Recall ML workflow is a sequence of steps to build and deploy a model that solves a problem using data.

The pipeline

Ingestion & Preprocessing	Analysis	Modeling	Deployment
Definition	EDA	Selection	Tuning
Data Collection	Feature Engineering	Training	Deployment
Cleaning		Evaluation	Monitoring

ML Workflow Graph

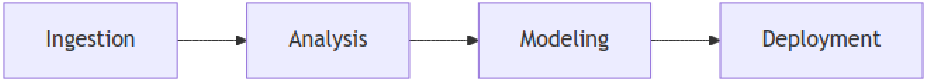
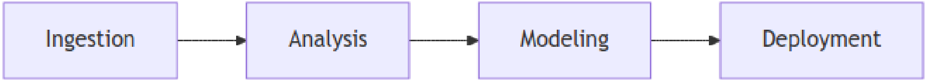


Figure: ML workflow steps rendered as a flowchart



What is Model Training?

- Model training is the process of estimating parameters θ of a model $f_\theta(x)$ using data $\{(x_i, y_i)\}_{i=1}^n$.
- Typically achieved by minimizing a loss function:

$$\hat{\theta} = \arg\min_\theta \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f_\theta(x_i), y_i) \tag{1}$$

- Common loss functions:
  - Squared error loss (regression): $\mathcal{L}(\hat{y}, y) = (\hat{y} - y)^2$
  - Cross-entropy loss (classification):

$$\mathcal{L}(\hat{y}, y) = -\sum_{c} \mathbb{1}_{\{y=c\}} \log \hat{p}_c \mathbb{1}_{\{x=1\}} \tag{2}$$

Cross-Entropy Loss (Classification)

The cross-entropy loss measures how well a predicted probability distribution $\hat{p}$ matches the true label y.
For a multiclass classification problem:

$$\mathcal{L}(\hat{y}, y) = -\sum_{c=1}^{C} \mathbb{1}_{\{y=c\}} \log \hat{p}_c \tag{3}$$

- $\hat{p}_c$: predicted probability for class c
- y: true class label
- Only the log probability of the true class contributes to the loss.

Binary Cross-Entropy Example

$$\mathcal{L}(\hat{y}, y) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \tag{4}$$

Interpretation

- Penalizes confident wrong predictions heavily.
- Encourages models to predict probabilities that reflect the actual distribution.

Training vs Generalization

- Empirical risk (training error):

$$\hat{R}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f_\theta(x_i), y_i) \tag{5}$$

- Expected risk (true/generalization error):

$$R(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathcal{L}(f_\theta(x), y)\right] \tag{6}$$

- Generalization gap: $R(\theta) - \hat{R}(\theta)$
- Overfitting: small $\hat{R}$, large R

What Is Expected Risk?

The expected risk or generalization error is the average loss over the true data distribution $\mathcal{D}$:

$$R(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathcal{L}(f_\theta(x), y)\right] \tag{7}$$

- θ: model parameters
- $f_\theta(x)$: model prediction
- $\mathcal{L}$: loss function (e.g., squared error)
- $\mathcal{D}$: unknown true distribution of the data

Why It Matters

- It tells us how well the model will perform on new data.
- Since $\mathcal{D}$ is unknown, we estimate it using validation or test sets.

Empirical vs Expected Risk

Risk Type	Expression	Description
Empirical Risk	$\hat{R}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f_\theta(x_i), y_i)$	Error on training data
Expected Risk	$R(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(f_\theta(x), y)]$	Error on all data

- Goal: Minimize expected risk while avoiding overfitting.
- Overfitting in practical terms means complicating the model so that it lowers the Emperical Risk without lowering the Expected Risk

Evaluation Metrics

- Regression:
  - Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

  - R^2 score:

$$R^2 = 1 - \frac{\sum_i(\hat{y}_i - y_i)^2}{\sum_i(y_i - \bar{y})^2}$$

- Classification:
  - Accuracy: $\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\{\hat{y}_i = y_i\}}$
  - Precision: $\frac{\text{TP}}{\text{TP}+\text{FP}}$
  - Recall: $\frac{\text{TP}}{\text{TP}+\text{FN}}$
  - F1 score: harmonic mean of precision and recall

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Overview of Classification Metrics

Different tasks prioritize different types of error.

Metric	Measures	Use Case
Accuracy	Overall correctness	Balanced datasets
Precision	True positives among predicted pos	False positives are costly (e.g., spam)
Recall	True positives among actual pos	False negatives are costly (e.g., disease)
F1 Score	Harmonic mean of P and R	Imbalanced data, cost for FP and FN
ROC AUC	Probabilistic ranking	Model comparison, threshold tuning

Definition and Intuition

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

- Correct predictions / Total predictions
- Best for balanced datasets

Precision

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

- How many predicted positives are truly positive?
- High precision = few false positives

Recall

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

- How many actual positives were correctly predicted?
- High recall $=$ few false negatives

Balancing Precision and Recall

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Harmonic mean of precision and recall
- Use when both types of error matter
- Good for imbalanced datasets

Receiver Operating Characteristic

- Plot of True Positive Rate vs. False Positive Rate at various thresholds
- Area Under Curve (AUC) ranges from 0.5 (random) to 1.0 (perfect)

$$\text{TPR} = \frac{TP}{TP + FN}, \quad \text{FPR} = \frac{FP}{FP + TN}$$

- AUC is threshold-independent
- Use when you want to compare classifiers

Choosing the Right Metric

- Accuracy: for balanced classes
- Precision: when false positives are costly
- Recall: when false negatives are costly
- F1: when both matter, especially in imbalanced data
- AUC: for ranking models across thresholds

Cross-Validation

- Cross-validation estimates generalization error by partitioning data.
- k-fold CV:
  - Split data into k disjoint subsets.
  - For each $i = 1, \ldots, k$:
    - Train on $k - 1$ folds
    - Evaluate on fold i
  - Average the evaluation metrics.

Bias-Variance Tradeoff

- Expected prediction error at point x:

$$\mathbb{E}[(f(x)-y)^2] = \underbrace{[\mathbb{E}(f(x))-y]^2}_{\text{Bias}^2} + \underbrace{\mathbb{E}[(f(x)-\mathbb{E}(f(x)))^2]}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{Irreducible error}}$$

- Simple models: low variance, high bias
- Complex models: low bias, high variance

Model Selection

- Choose the best model using a validation set or cross-validation.
- Avoid tuning hyperparameters using the test set.
- Balance:
  - Training error
  - Generalization performance
  - Computational cost

Hyperparameter Tuning

Some model settings are not learned from data but must be specified manually these are hyperparameters.

Model	Hyperparameter Examples
k-NN	Number of neighbors k
Decision Tree	Max depth, min samples per leaf
Lasso/Ridge	Regularization strength α
Neural Network	Learning rate, batch size

Why Tune Hyperparameters?

- Improve generalization
- Prevent overfitting
- Optimize computational efficiency

Best Practices

- Use a validation set or cross-validation to evaluate each setting.
- Never use the test set for tuning it must simulate unseen data.

Trade-offs

- Training error vs validation error
- Model complexity vs performance
- Runtime vs accuracy

Tools

- Grid search, random search, or Bayesian optimization

Summary Training and Evaluation

- Training minimizes empirical loss.
- Evaluation uses test or validation data.
- Use metrics appropriate for the task.
- Cross-validation provides robust error estimates.
- The bias-variance tradeoff is fundamental in choosing models.