Introduction to Machine Learning Modeling, Training and Evaluation

Evan Misshula

June 8, 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

:BEAMER$_{env}$: block :END:

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

ML Workflow Graph

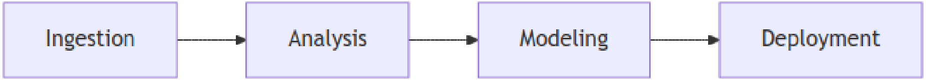
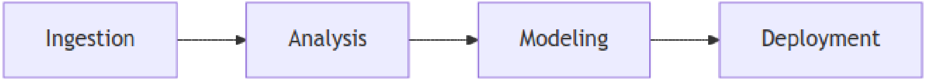


Figure: ML workflow steps rendered as a flowchart



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):

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

Training vs Generalization

- Empirical risk (training error):

$$\hat{R}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f_\theta(x_i), y_i) \tag{3}$$

- Expected risk (true/generalization error):

$$R(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}(f_\theta(x), y)\right] \tag{4}$$

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

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

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

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

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.