Model Validation

Recap

We have discussed

- Linear regression, polynomial regression, lasso regression,
- kNN, loess, spline regression
- Decision trees, boosting, random forests
- Hierarchical clustering, K-means
- PCA, tSNE

Today: How to choose the *best* model (or to tune the hyperparameters)?

Model validation

- **<u>Other names</u>**: model evaluation, model criticism
- Goal: evaluate how well a model describe a dataset
- Criteria (to convince others and *ourselves* our model makes sense)
    - Subjective criteria: domain knowledge, common practice, interpretability
    - **Objective/quantitative criteria**: numeric metrics
- Methods (to ensure the criteria are evaluated properly)
    - Validation with new data
    - Validation with existing data
        - **Split-sample validation, cross-validation**
        - Residual analysis (e.g., residual plots)

Evaluation metrics

We will take a brief glimpse at the metrics to use for

- Supervised learning with numeric output
- Supervised learning with categorical output
- Unsupervised learning with numeric output
- Unsupervised learning with categorical output

Supervised Learning with Numeric Output

Metric / loss	What it measures	When to use	Pros / Cons
MSE / RMSE	Quadratic loss; heavily penalises large errors	Default for least-squares, Gaussian noise	Smooth, differentiable; sensitive to outliers
MAE	Mean Absolute error	Heavy-tailed noise, median regression	Robust to outliers; but gradient is constant (harder for some optimisers)
MAPE / SMAPE	Relative % error	Retail demand, finance	Scale-free; explodes near 0
^{2} / Adj-^{2}	Fraction of variance explained	Linear models, presentation	Intuitive; can be inflated by heteroskedasticity
Log-likelihood	Average log-density under predictive distribution	Probabilistic regression, Bayesian models	Proper scoring rule; needs predictive density
Rank-based metrics	Order rather than value	Recommenders, credit risk	Invariant to monotone transforms

Supervised Learning with Categorical Output

Metric / loss	What it measures	When to use	Pros / Cons
Accuracy / Misclassicification rate	Proportion of correct classifications	Balanced classes, quick sanity check	Ignores class imbalance, probability calibration
Precision / Recall	Confusion-matrix slices	Imbalanced classes, information retrieval	Focus on positives
Specificity, Sensitivity	True-neg / true-pos rates	Medical screening	Complements recall
AUC-ROC	Ranking quality as threshold varies	Class-imbalance, ranking tasks	Threshold-free; AU-PR preferred for rare positives
Likelihood ratio / Deviance	Model comparison (nested GLMs)	Classic stats	Links to chi-square tests

Confusion matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Unsupervised Learning with Numeric Output

Metric	What / How	Typical use-case	Pros / Cons
Within-cluster SSE (Inertia, WCSS)	Sum of squared errors	K-means elbow method	Fast; decreases monotonically—needs elbow or gap statistic
Silhouette coefficient	internal vs. nearest-other-cluster distance	Any clustering with distance metric	Needs pairwise distances
Reconstruction error (MSE, MAE)	PCA, auto-encoder	Dim. reduction, anomaly detection	Comparable to supervised losses
Explained variance ratio	Variance captured by first PCs	PCA	Easy to interpret; linear only
Log-likelihood, AIC/BIC	Density fit of GMM, KDE, normalizing flows	Model selection	Penalises over-fit by param count

Unsupervised Learning with Categorical Output

Metric	What / How	Typical use-case	Pros / Cons
Purity & Entropy	For each cluster, majority-class proportion / entropy; average across clusters	Quick external check when ground-truth labels available	Simple; ignores cluster size balance
Adjusted Rand Index (ARI)	Pair-wise agreement corrected for chance	External cluster validation	Sensitive to the number of clusters
Normalized Mutual Information (NMI)	MI between cluster labels & truth	External validation (large K)	Symmetric; handles many classes

Thoughts?

- With so many metrics, should we consider evaluation measures of model evaluation measures?
- How do we evaluate complex output? (1) Code. (2) Image. (3) Text.
- ...

Validation Methods

Recall that

- Validation with new data
- Validation with existing data
    - **Split-sample validation, cross-validation**
    - Residual analysis (e.g., residual plots)

Next: why do we need to employ proper validation methods?

Model Complexity

- Complexity of a model
  $\approx$
  number of parameters in the model
    - More parameters
      $\rightarrow$
      more flexible to learn an unknown mechanism
- Too many paramters
  $\rightarrow$
  the model learns the (noninformative, irreproducible, ungeneralizable) patterns of noise
- **Numeric example with polynomial regression**
- **Illustration by MLU**

Double Descent

- Occurs when model capacity grows beyond the point where it can *exactly* interpolate the training data
- Highly over-parameterised models (deep neural nets, large ensembles, high-degree kernels) often defies the U-shape curve
- Double-dip (hence double descent) phenomon
- **Illustration by MLU**

Validation Methods

- Observation: error on new dataset explodes when model is too complex
  Solution: penalize model complexity
- Information Criteria:
    - Akaike Information Criterion
    - Bayesian Information Criterion

Validation Methods

- Observation: Validation with new data is the gold standard, but infeasible in practice
  Solution: create "new" data from existing data
- Validation with existing data using data spliting
    - **Split-sample validation**
    - **Cross-validation** (resampling method)
    - **Bootstrap out-of-bag errors** (resampling method, too)