MACHINE LEARNING QUICK REFERENCE: BEST PRACTICES

Topic	Common Challenges	Suggested Best Practice
Data Preparation		
Data collection	• Biased data • Incomplete data • The curse of dimensionality • Sparsity	• Take time to understand the business problem and its context • Enrich the data • Dimension-reduction techniques • Change representation of data (e.g., COO)
“Untidy” data	• Value ranges as columns • Multiple variables in the same column • Variables in both rows and columns	Restructure the data to be “tidy” by using the melt and cast process
Outliers	• Out-of-range numeric values and unknown categorical values in score data • Undue influence on squared loss functions (e.g., regression, GBM, *k*-means)	• Robust methods (e.g., Huber loss function) • Discretization (binning) • Winsorizing
Sparse target variables	• Low primary event occurrence rate • Overwhelming preponderance of zero or missing values in target	• Proportional oversampling • Inverse prior probabilities • Mixture models
Variables of disparate magnitudes	• Misleading variable importance • Distance measure imbalance • Gradient dominance	Standardization
High-cardinality variables	• Overfitting • Unknown categorical values in holdout data	• Discretization (binning) • Weight of evidence • Leave-one-out event rate
Missing data	• Information loss • Bias	• Discretization (binning) • Imputation • Tree-based modeling techniques
Strong multicollinearity	Unstable parameter estimates	• Regularization • Dimension reduction
Training		
Overfitting	High-variance and low-bias models that fail to generalize well	• Regularization • Noise injection • Partitioning or cross validation
Hyperparameter tuning	Combinatorial explosion of hyperparameters in conventional algorithms (e.g., deep neural networks, super learners)	• Local search optimization, including genetic algorithms • Grid search, random search
Ensemble models	• Single models that fail to provide adequate accuracy • High-variance and low-bias models that fail to generalize well	• Established ensemble methods (e.g., bagging, boosting, stacking) • Custom or manual combinations of predictions
Model Interpretation	Large number of parameters, rules, or other complexity obscures model interpretation	• Variable selection by regularization (e.g., L1) • Surrogate models • Partial dependency plots, variable importance measures
Computational resource exploitation	• Single-threaded algorithm implementations • Heavy reliance on interpreted languages	• Train many single-threaded models in parallel • Hardware acceleration (e.g. SSD, GPU) • Low-level, native libraries • Distributed computing, when appropriate
Deployment		
Model deployment	Trained model logic must be transferred from a development environment to an operational computing system to assist in organizational decision-making processes	• Portable scoring code or scoring executables • In-database scoring • Web service scoring
Model decay	• Business problem or market conditions have changed since the model was created • New observations fall outside domain of training data	• Monitor models for decreasing accuracy • Update/retrain models regularly • Champion-challenger tests • Online updates

https://github.com/sassoftware/enlighten-apply/tree/master/ML_tables