Best practices for supervised learning

Fabrice Rossi

28 novembre 2025

Table des matières

1 Data preparation

An excellent book on the subject : Feature Engineering and Selection: A Practical Approach for Predictive Models.

1.1 Numerical data

— centring and reduction strongly recommended (especially with neural networks)
— transformation of the target variable
  — for example, predicting log of the target rather than the target
  — on a case-by-case basis
— **never** quantise numerical data excepted for (generalised) linear models

1.2 Nominal data

— full disjunctive coding
  — a.k.a. *one hot encoding*
  — very classical solution (automatic in R)
  — useless for some models when they are well programmed (trees), but this is not always the case !

— **never** represent modalities by *ad hoc* numerical values
— large number of modalities
  — potentially problematic (especially with disjunctive coding)
  — various solutions
    — grouping of rare modalities
    — hash codes
    — supervised groupings (beware of data leakage !)
    — external expert based representation (e.g. postcodes -> geographical coordinates)
— specific solutions for ordinal data (polynomial contrasts)
— see Encoding Categorical Data for implementations of various solutions with tidymodels in R

2 Building a predictive model

2.1 Quality criteria

The choice of the loss function is not trivial :
— must be linked to business indicators
— may be constrained by the library used : it is rare to be able to change the loss function beyond the basic choices (one of the most flexible libraries is XGboost)
— try to specify asymmetric costs directly to the training if the application requires it (in supervised classification)
— be careful to adapt the loss function to the problem and the model
  — regression versus supervised classification
  — but also : quantile regression (pinball loss), survival analysis, etc.
  — gradient boosting in classification : logistic loss (for example)

2.2 Which models ?

— for classical tabular data (without specific structure) :
  — state of the art in performance
    — random forest
    — gradient boosting (XGboost, CatBoost, Lightgbm)
  — reference models
    — generalised linear model
    — basic tree
— for specific data such as text and images : deep neural networks (pre-trained or with very large data)

2.3 Model selection

2.3.1 Basic principles

It is imperative to :
— select the meta-parameters of the model and the model itself by a statistically valid procedure (e.g. cross-validation)
— estimate the future performance of the model by a statistically valid procedure (e.g. learning/testing)
— **nothing** can be chosen on the basis of estimated future performance

2.3.2 Data leakage

Resampling methods are based on a fundamental principle : a model is **never** evaluated on the data used to build it. A data leak is a violation of this principle. Examples :

— centring and reduction
— PCA and similar models
— selection of variables by a filter method (e.g. mutual information or correlation)
— supervised coding of nominal variables (e.g. target encoding)
— imputation of missing data (both simple average/median imputation and advanced such as k-NN based one)

2.3.3 Coping with the effects of random splits

Resampling methods are generally very sensitive to random draws. Best practices :
— make the whole processing chain fully reproducible (set.seed in R and similar approaches)
— use the same cross-validation blocks to compare models (or the same source of randomness in general)
— estimate the sensitivity of the results to the random splits (in particular the learning/test split)

2.3.4 Parameter grids

Ideally, one wishes to explore as many combinations of the meta-parameters of the model under study as possible. Realistically, this means selecting a reasonable set of configurations. There are no well-established rules for the selection of these configurations :

— one avoids keeping a configuration "on the edge" as the optimal configuration : if, for example, a k-nearest neighbour is ideal for k = 11, but one has not tested k = 13 (or k = 9), then one cannot guarantee that the selected configuration is really the optimal one (if only locally)

— Various techniques more advanced than the use of a regular grid are available : incomplete (and pseudo-random) grids, Bayesian search, etc.

2.3.5 Special cases : the out-of-bag estimate and the leave-one-out

Some models have valid and efficient estimators of their future performance (the out-of-bag estimator for random forests and the leave-one-out estimator for linear models).
— this poses a comparison problem :
  — how to compare a linear regression loo with a random forest oob ?
  — how to compare a gradient boosting with cross-validation and a random forest oob ?
  — more generally, how can different estimators be compared ?
— in theory, no direct comparison is possible
  — different bias and variance
  — poorly controlled random effects
— in practice we suggest
  — to select the meta-parameters of a given model with its native estimator
  — to calculate the performance estimate by cross-validation of this model for the selected meta-parameters
— to compare the models with each other at least from this estimation on the same blocks

2.4 Analysis of the results

— full analysis of errors
  — do not report only the empirical risk (average of errors)
  — study the maximum errors
  — study the wrongly predicated cases (at the level of explanatory variables : do they display a typical behaviour ?)
— alternative quality criteria
  — the area under the ROC curve is very useful when deployment conditions may be variable (change in relative error costs, change in a priori class probabilities)

    - the confusion matrix and associated quantities (e.g. precision and recall) is essential in supervised classification
    - study the scores in classification, not only the accuracy
    - investigate the calibration of the probabilities in the classification setting
- importance and relevance of variables
    - use integrated solutions to analyse the importance of variables
    - beware of 'nonsense' (non-causal) variables

2.5 Planning for deployment

- deployment model ?
    - in general, individual forecasts will be made (one new observation at a time)
    - for monitoring purposes one may have to make a block of forecasts at once
- missing data
    - if data are missing during learning, they will probably be missing in production
    - the imputation method chosen must be compatible with the deployment model (generally individual forecasts)

3 Monitoring a deployed model

The fundamental difficulty of the deployment (apart from the IT aspects which should not be neglected) lies in the absence of an immediate correction signal : in general, we do not know quickly whether our forecast was correct. The data is even censored : in some cases (credit granting) we will never know if we were wrong on one of the classes.

It is often interesting to keep an alternative model during model building and to compare its behaviour with the chosen model :

- consensus on predictions
- specific study of cases of disagreement

3.1 Short term

Short-term monitoring is done without feedback on predictions. The distribution of new data is monitored :

- effect of centring and reduction learned in the learning phase
- comparison of the distributions of the explanatory variables
- distribution of predicted targets

3.2 Long term

Possible when there is feedback on the predictions :

- monitoring the quality of predictions (compared to what was expected)
- possible changes in the distribution of targets
- specific study of error cases (compared to learning error cases)