# 18 Feature Selection Overview

Contents

## 18.1 Models with Built-In Feature Selection

Many models that can be accessed using `caret` ’s `train` function produce prediction equations that do not necessarily use all the predictors. These models are thought to have built-in feature selection: `ada` , `AdaBag` , `AdaBoost.M1` , `adaboost` , `bagEarth` , `bagEarthGCV` , `bagFDA` , `bagFDAGCV` , `bartMachine` , `blasso` , `BstLm` , `bstSm` , `C5.0` , `C5.0Cost` , `C5.0Rules` , `C5.0Tree` , `cforest` , `chaid` , `ctree` , `ctree2` , `cubist` , `deepboost` , `earth` , `enet` , `evtree` , `extraTrees` , `fda` , `gamboost` , `gbm_h2o` , `gbm` , `gcvEarth` , `glmnet_h2o` , `glmnet` , `glmStepAIC` , `J48` , `JRip` , `lars` , `lars2` , `lasso` , `LMT` , `LogitBoost` , `M5` , `M5Rules` , `msaenet` , `nodeHarvest` , `OneR` , `ordinalNet` , `ORFlog` , `ORFpls` , `ORFridge` , `ORFsvm` , `pam` , `parRF` , `PART` , `penalized` , `PenalizedLDA` , `qrf` , `ranger` , `Rborist` , `relaxo` , `rf` , `rFerns` , `rfRules` , `rotationForest` , `rotationForestCp` , `rpart` , `rpart1SE` , `rpart2` , `rpartCost` , `rpartScore` , `rqlasso` , `rqnc` , `RRF` , `RRFglobal` , `sdwd` , `smda` , `sparseLDA` , `spikeslab` , `wsrf` , `xgbDART` , `xgbLinear` , `xgbTree` . Many of the functions have an ancillary method called `predictors` that returns a vector indicating which predictors were used in the final model.

In many cases, using these models with built-in feature selection will be more efficient than algorithms where the search routine for the right predictors is external to the model. Built-in feature selection typically couples the predictor search algorithm with the parameter estimation and are usually optimized with a single objective function (e.g. error rates or likelihood).

## 18.2 Feature Selection Methods

Apart from models with built-in feature selection, most approaches for reducing the number of predictors can be placed into two main categories. Using the terminology of John, Kohavi, and Pfleger (1994):

- *Wrapper* methods evaluate multiple models using procedures that add and/or remove predictors to find the optimal combination that maximizes model performance. In essence, wrapper methods are search algorithms that treat the predictors as the inputs and utilize model performance as the output to be optimized. **caret** has wrapper methods based on recursive feature elimination, genetic algorithms, and simulated annealing.
- *Filter* methods evaluate the relevance of the predictors outside of the predictive models and subsequently model only the predictors that pass some criterion. For example, for classification problems, each predictor could be individually evaluated to check if there is a plausible relationship between it and the observed classes. Only predictors with important relationships would then be included in a classification model. Saeys, Inza, and Larranaga (2007) surveys filter methods. **caret** has a general framework for using univariate filters.

Both approaches have advantages and drawbacks. Filter methods are usually more computationally efficient than wrapper methods, but the selection criterion is not directly related to the effectiveness of the model. Also, most filter methods evaluate each predictor separately and, consequently, redundant (i.e. highly-correlated) predictors may be selected and important interactions between variables will not be able to be quantified. The downside of the wrapper method is that many models are evaluated (which may also require parameter tuning) and thus an increase in computation time. There is also an increased risk of over-fitting with wrappers.

## 18.3 External Validation

It is important to realize that feature selection is part of the model building process and, as such, should be externally validated. Just as parameter tuning can result in over-fitting, feature selection can over-fit to the predictors (especially when search wrappers are used). In each of the **caret** functions for feature selection, the selection process is included in any resampling loops. See

See Ambroise and McLachlan (2002) for a demonstration of this issue.