A Recommended preprocessing

It has been said previously that the type of preprocessing is dependent on the type of model being fit. For example, models that use distance functions or dot products should have all of their predictors on the same scale so that distance is measured appropriately.

To learn more about each of these models, and others that might be available, a linked list can be found at `tidymodels.org/find/parsnip` .

This Appendix provides recommendations for baseline levels of preprocessing that are needed for various model functions. In the table below, the preprocessing methods are categorized as:

- **dummy**: Do qualitative predictors require a numeric encoding (e.g. via dummy variables or other methods).
- **zv**: Should columns with a single unique value be removed?
- **impute**: If some predictors are missing, should they be estimated via imputation?
- **decorrelate**: If there are correlated predictors, should this correlation be mitigated? This might mean filtering out predictors, using principal component analysis, or a model-based technique (e.g. regularization).
- **normalize**: Should predictors be centered and scaled?
- **transform**: Is it helpful to transform predictors to be more symmetric?

The information in the table below is not exhaustive and somewhat depends on the implementation. For example, as noted below the table, some models may not require a particular preprocessing operation but the implementation may require it. In the table, ✓ indicates that the method is required for the model and × indicates that it is not. The ◌ symbol means that the model *may* be helped by the technique but it is not required.

model	dummy	zv	impute	decorrelate	normalize	transform
bag_mars()	✓	×	✓	○	×	○
bag_tree()	×	×	×	○[1]	×	×
boost_tree()	×+	○	✓+	○[1]	×	×
C5_rules()	×	×	×	×	×	×
cubist_rules()	×	×	×	×	×	×
decision_tree()	×	×	×	○[1]	×	×
discrim_flexible()	✓	✓	✓	✓	×	○
discrim_linear()	✓	✓	✓	✓	×	○
discrim_regularized()	✓	✓	✓	✓	×	○
linear_reg()	✓	✓	✓	✓	×	○
logistic_reg()	✓	✓	✓	✓	×	○
mars()	✓	×	✓	○	×	○
mlp()	✓	✓	✓	✓	✓	✓
multinom_reg()	✓	✓	✓	✓	×	○
naive_Bayes()	×	✓	✓	○[1]	×	×
nearest_neighbor()	✓	✓	✓	○	✓	✓
pls()	✓	✓	✓	×	✓	✓
poisson_reg()	✓	✓	✓	✓	×	○
rand_forest()	×	○	✓+	○[1]	×	×
rule_fit()	✓	×	✓	○[1]	×	×
svm_*()	✓	✓	✓	✓	✓	✓

Footnotes:

1. Decorrelating predictors may not help improve performance. However, fewer correlated predictors can improve the estimation of variance importance scores (see Fig. 11.4 of Kuhn and Johnson (2020)). Essentially, the selection of highly correlated predictors is almost random.

The notation of $^+$ means that the answer depends on the implementation. Specifically:

- *Theoretically*, any tree-based model does not require imputation. However, many tree ensemble implementations require imputation.
- While tree-based boosting methods generally do not require the creation of dummy variables, models using the `xgboost` engine do.

REFERENCES

Kuhn, M, and K Johnson. 2020. *Feature Engineering and Selection: A Practical Approach for Predictive Models*. CRC Press.