(DA-Lab)
MS. SHWETA TIWARI
April 19, 2022

DA-Lab
DATA ANALYTICS-LAB_

By SHWETA TIWARI

Get Your Data Ready For Machine Learning in R with Pre-Processing

Preparing data is required to get the best results from machine learning algorithms.

In this post you will discover how to transform your data in order to best expose its structure to machine learning algorithms in R using the caret package.

You will work through 8 popular and powerful data transforms with recipes that you can study or copy and paste into your current or next machine learning project.

Let's get started.

Need For Data Pre-Processing

You want to get the best accuracy from machine learning algorithms on your datasets.

Some machine learning algorithms require the data to be in a specific form. Whereas other algorithms can perform better if the data is prepared in a specific way, but not always. Finally, your raw data may not be in the best format to best expose the underlying structure and relationships to the predicted variables.

It is important to prepare your data in such a way that it gives various different machine learning algorithms the best chance on your problem.

You need to pre-process your raw data as part of your machine learning project.

Data Pre-Processing Methods

It is hard to know which data-preprocessing methods to use.

You can use rules of thumb such as:

- Instance based methods are more effective if the input attributes have the same scale.
- Regression methods can work better of the input attributes are standardized.

These are heuristics, but not hard and fast laws of machine learning, because sometimes you can get better results if you ignore them.

You should try a range of data transforms with a range of different machine learning algorithms. This will help you discover both good representations for your data and algorithms that are better at exploiting the structure that those representations expose.

It is a good idea to spot check a number of transforms both in isolation as well as combinations of transforms.

In the next section you will discover how you can apply data transforms in order to prepare your data in R using the caret package.

Data Pre-Processing With Caret in R

The caret package in R provides a number of useful data transforms.

These transforms can be used in two ways.

- **Standalone**: Transforms can be modeled from training data and applied to multiple datasets. The model of the transform is prepared using the *prePrecess()* function and applied to a dataset using the *predict()* function.
- **Training**: Transforms can prepared and applied automatically during model evaluation. Transforms applied during training are prepared using the *prePrecess()* and passed to the *train()* function via the preProcess argument.

A number of data preprocessing examples are presented in this section. They are presented using the standalone method, but you can just as easily use the prepared preprocessed model during model training.

All of the preprocessing examples in this section are for numerical data. Note that the preprocessing functions will skip over non-numeric data without raising an error.

You can learn more about the data transforms provided by the caret package by reading the help for the preProcess function by typing ?preProcess and by reading the Caret Pre-Processing page.

The data transforms presented are more likely to be useful for algorithms such as regression algorithms, instance-based methods (like kNN and LVQ), support vector machines and neural networks. They are less likely to be useful for tree and rule based methods.

Summary of Transform Methods

Below is a quick summary of all of the transform methods supported in the method argument of the *prePrecess()* function in caret.

- "*BoxCox*": apply a Box–Cox transform, values must be non-zero and positive.
- "*YeoJohnson*": apply a Yeo-Johnson transform, like a BoxCox, but values can be negative.
- "*expoTrans*": apply a power transform like BoxCox and YeoJohnson.
- "*zv*": remove attributes with a zero variance (all the same value).
- "*nzv*": remove attributes with a near zero variance (close to the same value).
- "*center*": subtract mean from values.
- "*scale*": divide values by standard deviation.
- "*range*": normalize values.

- "*pca*": transform data to the principal components.
- "*ica*": transform data to the independent components.
- "*spatialSign*": project data onto a unit circle.

The following sections will demonstrate some of the more popular methods.

1. Scale

The scale transform calculates the standard deviation for an attribute and divides each value by that standard deviation.

2. Center

The center transform calculates the mean for an attribute and subtracts it from each value.

3. Standardize

Combining the scale and center transforms will standardize your data. Attributes will have a mean value of 0 and a standard deviation of 1.

4. Normalize

Data values can be scaled into the range of [0, 1] which is called normalization.

5. Box-Cox Transform

When an attribute has a Gaussian-like distribution but is shifted, this is called a skew. The distribution of an attribute can be shifted to reduce the skew and make it more Gaussian. The BoxCox transform can perform this operation (assumes all values are positive).

6. Yeo-Johnson Transform

Another power-transform like the Box-Cox transform, but it supports raw values that are equal to zero and negative.

7. Principal Component Analysis

Transform the data to the principal components. The transform keeps components above the variance threshold (default=0.95) or the number of components can be specified (pcaComp). The result is attributes that are uncorrelated, useful for algorithms like linear and generalised linear regression.

8. Independent Component Analysis

Transform the data to the independent components. Unlike PCA, ICA retains those components that are independent. You must specify the number of desired independent components with the *n.comp* argument. Useful for algorithms such as naive bayes.

Tips For Data Transforms

Below are some tips for getting the most out of data transforms.

- **Actually Use Them**. You are a step ahead if you are thinking about and using data transforms to prepare your data. It is an easy step to forget or skip over and often has a huge impact on the accuracy of your final models.
- **Use a Variety**. Try a number of different data transforms on your data with a suite of different machine learning algorithms.
- **Review a Summary**. It is a good idea to summarize your data before and after a transform to understand the effect it had. The *summary()* function can be very useful.
- **Visualize Data**. It is also a good idea to visualize the distribution of your data before and after to get a spatial intuition for the effect of the transform.

Summary

In this section you discovered 8 data preprocessing methods that you can use on your data in R via the caret package:

- Data scaling
- Data centering
- Data standardization
- Data normalization

- The Box-Cox Transform
- The Yeo-Johnson Transform
- PCA Transform
- ICA Transform