For quite a while we have been teaching estimating variable re-encodings on the exact same data they are later *naively* using to train a model on, leads to an undesirable nested model bias. The vtreat package (both the R and Python Version) both incorporate a cross-frame method that allows one to use all the training data both to build learn variable re-encodings and to correctly train a subsequent model (for an example please see our recent [PyData LA talk](http://www.win-vector.com/blog/2019/12/pydata-los-angeles-2019-talk-preparing-messy-real-world-data-for-supervised-machine-learning/)).

The next version of vtreat will warn the user if they have improperly used the same data for both vtreat impact code inference and downstream modeling. So in addition to us warning you not to do this, the package now also checks and warns against this situation. vtreat has had methods for avoiding nested model bias for vary long time, we are now adding new warnings to confirm users are using them.

**Demonstrate the Warning**

One way to design variable treatments for binomial classification problems in vtreat is to design a cross-frame experiment.

# For this example we want vtreat version 1.5.1 or newer

# remotes::install\_github("WinVector/vtreat")

library(vtreat)

packageVersion("vtreat")

## [1] '1.5.1'

...

transform\_design = vtreat::mkCrossFrameCExperiment(

# data to learn transform from

dframe = training\_data,

# columns to transform

varlist = setdiff(colnames(training\_data), c('y', 'yc')),

# outcome variable

outcomename = 'yc',

# outcome of interest

outcometarget = TRUE

)

Once we have that we can pull the data transform and correct cross-validated training frame off the returned object as follows.

transform <- transform\_design$treatments

train\_prepared <- transform\_design$crossFrame

train\_prepared is prepared in the correct way to use the same training data for inferring the impact-coded variables, using the returned $crossFrame from mkCrossFrameCExperiment().

We prepare new test or application data as follows.

test\_prepared <- prepare(transform, test\_data)

The issue is: for training data we should not call prepare(), but instead use the cross-frame that is produces during transform design.

The point is we should not do the following:

train\_prepared\_wrong <- prepare(transform, training\_data)

## Warning in prepare.treatmentplan(transform, training\_data):

## possibly called prepare() on same data frame as designTreatments\*()/

## mkCrossFrame\*Experiment(), this can lead to over-fit. To avoid this, please use

## mkCrossFrame\*Experiment$crossFrame.

Notice we now get a warning that we should not have done this, and in doing so we may have a nested model bias data leak.

And that is the new nested model bias warning feature.