Auxiliary Uses of Decision Trees:

# Optimal Binning of Nominal Variables with Many Levels Using Decision Trees in SAS E-Guide

## Introduction

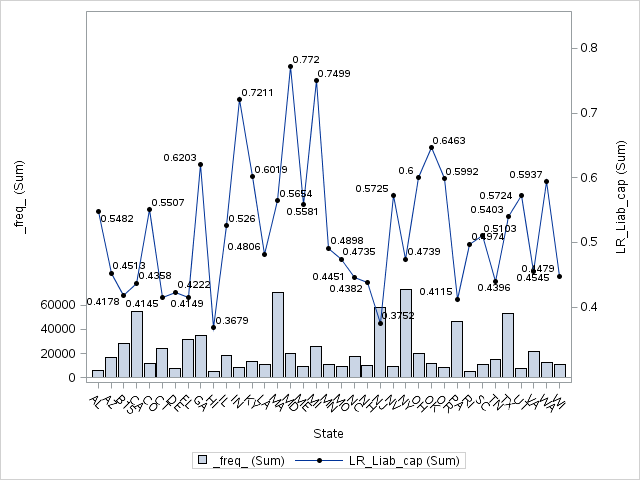
## A persistent issue in data mining is dealing with highly dimensional data. Paradoxically adding more data (in terms of numbers of variables) does not necessarily improve the ability to develop a reliably predictive model, as this also increases the dimensions of the input vector. This problem has become known as the curse of dimensionality. Reducing the number of dimensions can in some situations help the modeler develop a more reliable result.

## Highly dimensional data is frequently encountered, very often in dealing with nominal values that are used as inputs to models. A nominal value has one of a finite number of discrete values, such as a product code or a region code. Typically, character variables used as inputs in data mining are nominal variables. Numeric variables may be nominal as well; a classic example being a zip code.

## Identifying factor levels can be arbitrary, judgmental, or optimality driven. Decision trees are transparent, intuitive, non-parametric, and robust to influential values, outliers, and missing values; therefore, they can be used to find optimal bins, or factor levels.

## Example

The target variable in our model is the Loss Ratio. A typical risk factor is the State. In the following graph we can see the relationship between the Loss Ratio and the different states:



This variable has too many level in the context of the current model. Therefore, we would like to collapse the different states in only few levels. We can use decision trees to accomplish this task in an optimal way.

## Implementation Using the HPSPLIT Procedure

We can use the new HPSPLIT procedure in SAS STAT 13.2 (THE current version of SAS) to create decision trees in SAS Enterprise Guide 7.1 (the current version of SAS Enterprise Guide). Therefore, we don’t need to leave the SAS Enterprise Guide context in order to create our decision trees in SAS Enterprise Miner.

When we use a decision tree to create an optimal binning for a categorical variable with many levels, several other decisions must also be made during this process.

* How many leaves should have the decision tree?
* What is the maximum tree depth?
* What should the level of significance for factor splits be?
* What splitting criterion should be used?
* Should there be a minimum number of members in a node (policies or claims) to create a split?
* How Specifies how to handle missing values in an input variable?

The answers to the above question are the following:

* The level are conditioned by the number of observations in our dataset. In a typical context of data mining with several million of observation a subjective criteria many times applied is 16.
* The maxims tree depth should be one. Here we are using a decision tree such an auxiliary tool in our modeling process, therefore, we would like only to collapse the variable in few levels.
* The significant level for this kind of exercises usually is large, that is 0.1
* We should prevent post-pruning.
* The splitting criteria in the context of this example with at continuous target should be Variance or F-Test.
* The number of minimum observations in a node should be a credible exposure.
* The HPSPLIT is able to handle the missing values based on three options:
  + BRANCH requests that missing values be assigned to their own branch.
  + POPULARITY requests that missing values be assigned to the most popular, or largest, branch.
  + SIMILARITY requests that missing values be assigned to the branch they are most similar to (using the chisquare or F test criterion)

An implementation of the above criteria using the HPSPLIT procedure in the context of our above example is the following SAS code:

ods graphics on;

**proc** **hpsplit** data=Auto\_DS\_BT maxdepth=**1** maxbranch=**9** leafsize=**5000** alpha=**0.1**;

criterion ftest ; /\*default for interval target VARIANCE or FTEST \*/

prune none;

target LR\_liab\_cap / level = int;

input STATE / level = nom ;

output nodestats=stat;

**run**;

title 'Tree visualization';

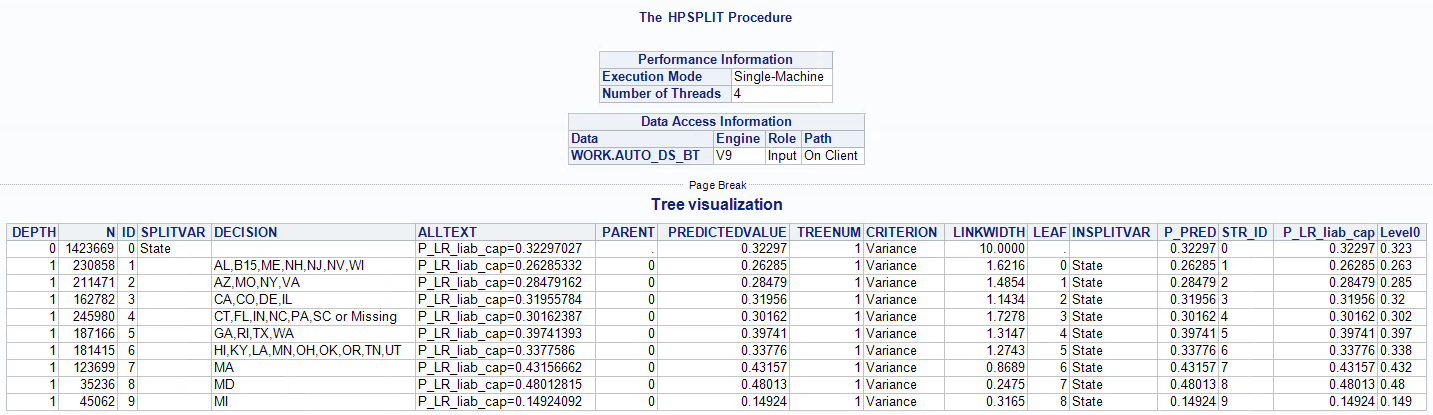
**proc** **print** data=stat noobs;

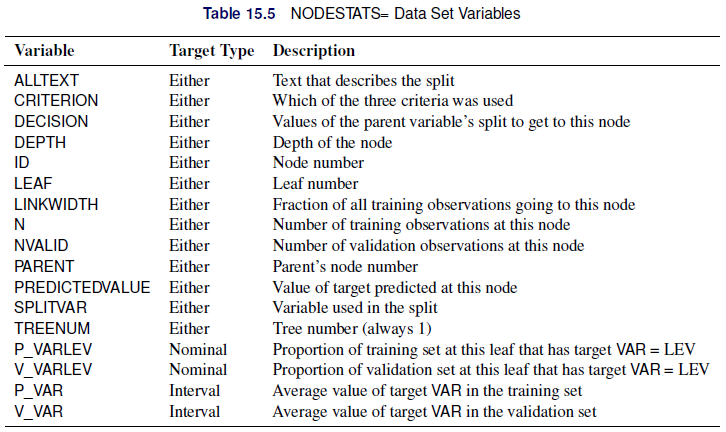
**run**;

Where:

* The dataset is Auto\_DS\_BT
* The maximum depth of the tree to be grown equals one
* The MAXBRANCH option specifies the maximum number of children per node in the tree. PROC HPSPLIT tries to create this number of children unless it is impossible (for example, if a split variable does not have enough levels). The default is the number of target levels. The maximum number of branches in our example is 9
* The minimum number of polices in each and every leaf is 5000. This option specifies the minimum number of observations that a split must contain in the training data set in order for the split to be considered. By default, LEAFSIZE=1.
* The split criterion used is the F-Test. By default, CRITERION = variance
* The alpha option specify the maximum p-value for a split to be considered, in this case 0.1. By default is 0.3. This option is only considered if you specify the FTEST criterion
* The prune option is not used
* The target variable is LR\_Liab\_cap. We used the level = int option to indicate that the target variable is continuous
* The explanatory variable is STATE. We used the level = nom option to indicate that the explanatory variable is nominal
* The output of the decision tree calculation is the SAS dataset called STAT
* The option for missing handle is not implemented in the current STAT product. Therefore, by default the criteria is POPULARITY.

Finally, I used the PRINT procedure to visualize the output:





Thankfully to the decision tree we have a derived new variable that groups the States in an optimal way:

if STATE in('AL','B15','ME','NH','NJ','NV','WI') then STATE\_TREE = '02:AL,B15,ME,NH,NJ,NV,WI';

else if STATE in('AZ','MO', 'NY','VA') then STATE\_TREE = '03:AZ,MO,NY,VA';

else if STATE in('CA','CO', 'DE','IL') then STATE\_TREE = '05:CA,CO,DE,IL';

else if STATE in('CT','FL', 'IN','NC','PA','SC','') then STATE\_TREE = '04:CT,FL,IN,NC,PA,SC,Miss';

else if STATE in('GA','RI', 'TX','WA') then STATE\_TREE = '07:GA,RI,TX,WA';

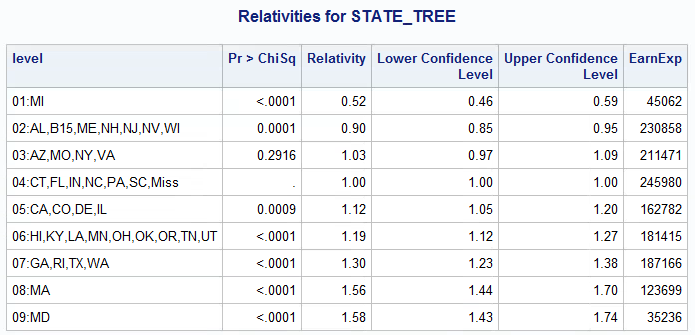
else if STATE in('HI','KY', 'LA','MN','OH','OK','OR','TN','UT') then STATE\_TREE = '06:HI,KY,LA,MN,OH,OK,OR,TN,UT';

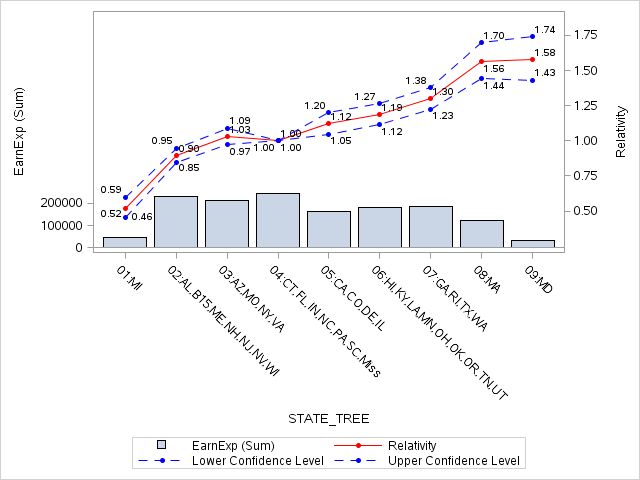
else if STATE in('MA') then STATE\_TREE = '08:MA';

else if STATE in('MD') then STATE\_TREE = '09:MD';

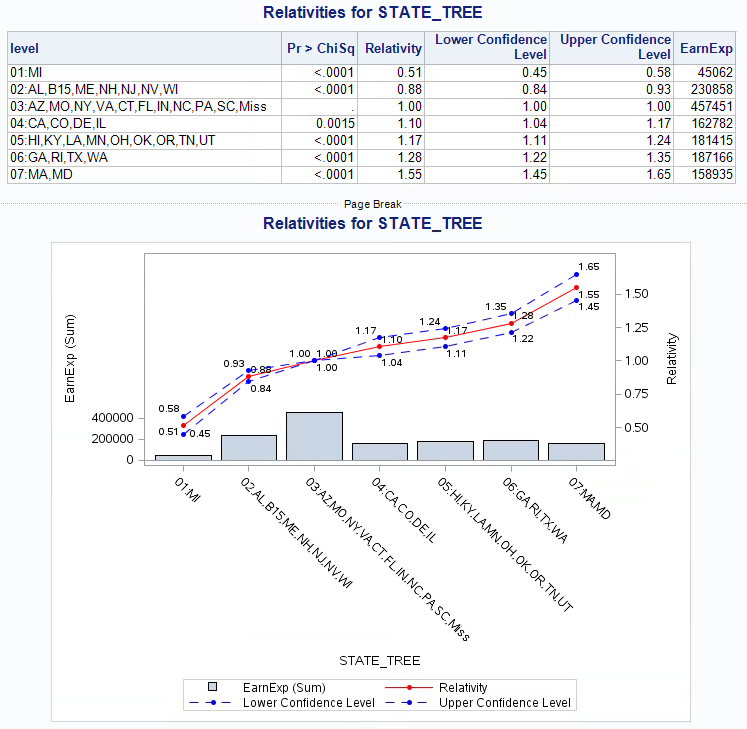
else if STATE in('MI') then STATE\_TREE = '01:MI';

The relativities for the STATE\_TREE variable in a multivariate model are (training dataset):





In a multivariable context some of the levels should be re-grouped again, such as 03 and 04 or 08 and 09



## Summary

* The new HPSPLIT can be used as an auxiliary tool during the modeling process creating an optimal binning of nominal variable with too many levels.
* Identifying factor levels can be arbitrary, judgmental, or optimality driven. Decision trees are transparent, intuitive, non-parametric, and robust to influential values, outliers, and missing values.
* The process is very quickly and you don’t need to leave the SAS E-Guide context to use SAS Enterprise Miner

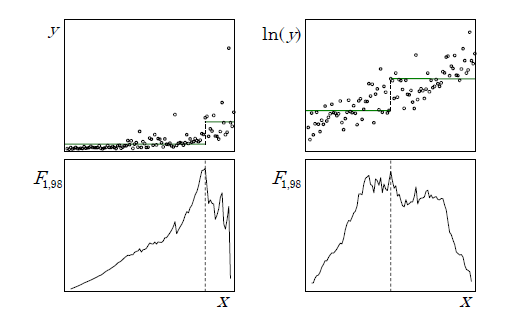
## Appendix: Regression Trees, Partition Criteria and Heteroscedasticity

There are two splitting criteria used in regression tress:

* The first criterion is based on impurity function of a node. When the target distribution is continuous, the sample variance is the obvious measure of impurity.
* The second criterion is based on a statistical test for one-way ANOVA, the F-Test.

Some differences between the two above criteria:

* The F test is better than (sample) variance reduction as it has P-value adjustment number of branches.
* F test is relatively robust to departures from normality assumption
* However, F test is sensitive to departures from non-constant variance



The F test has many optimal properties when the distribution of the target is independently and normally distributed with constant variance. The F test is relatively robust to departures from the normality assumption. However, variance heterogeneity (heteroscedasticity) can have disastrous effects. For example, the F test is too liberal (overstates the significance of the effect) when small nodes have larger variance. Consider the common case of a nonnegative target with variance increasing with the mean. Using the F test as the splitting criterion will tend to favor small splits of the largest values. Decision trees are usually regarded as robust and nonparametric. However, regression trees are not robust to heteroscedasticity. Like classical regression models, finding a suitable variance stabilizing transformation can improve the model. For example, apply a logarithm or a squared root transformation to the target variable. A typical context where a logarithm transformation is useful happens when the severity is the target variable.

## References

* De Ville, Barry, and Padraic Neville. 2013. Decision Trees for Analytics Using SAS® Enterprise Miner. Cary, NC: SAS Institute Inc.
* [The Anti-Curse: Creating an Extension to SAS® Enterprise Miner™ Using PROC ARBORETUM Andrew Cathie, SAS Institute (NZ) Ltd, Auckland, New Zealand](http://support.sas.com/resources/papers/proceedings11/155-2011.pdf)
* Decision Tree Modeling Course Notes was developed by William J.E. Potts and revised by Lorne Rothman. Technical review was provided by Bob Lucas and Michael J. Patetta. 2006 by SAS Institute Inc. Chapter 4 – Auxiliary Uses of Trees. Section 4.3. - Collapsing Levels
* [SAS/STAT® 13.2 User’s Guide High-Performance Procedures - Chapter 15. The HPSPLIT Procedure](http://support.sas.com/documentation/cdl/en/stathpug/67524/PDF/default/stathpug.pdf)
* SAS Enterprise Miner 13.2 High Performance Data Mining Nodes Help