Auxiliary Uses of Decision Trees:

Optimal Binning of Continuous Variables Using Decision Trees in SAS E-Guide

Introduction

An interval or continuous variable has infinite number of discrete values, such as the operator age or vehicle length. Also, these continuous variables, very frequently, have missing observations, outliers, repeated observations etc. that makes inefficient try to use these variables as a continuous variable during the modeling process.

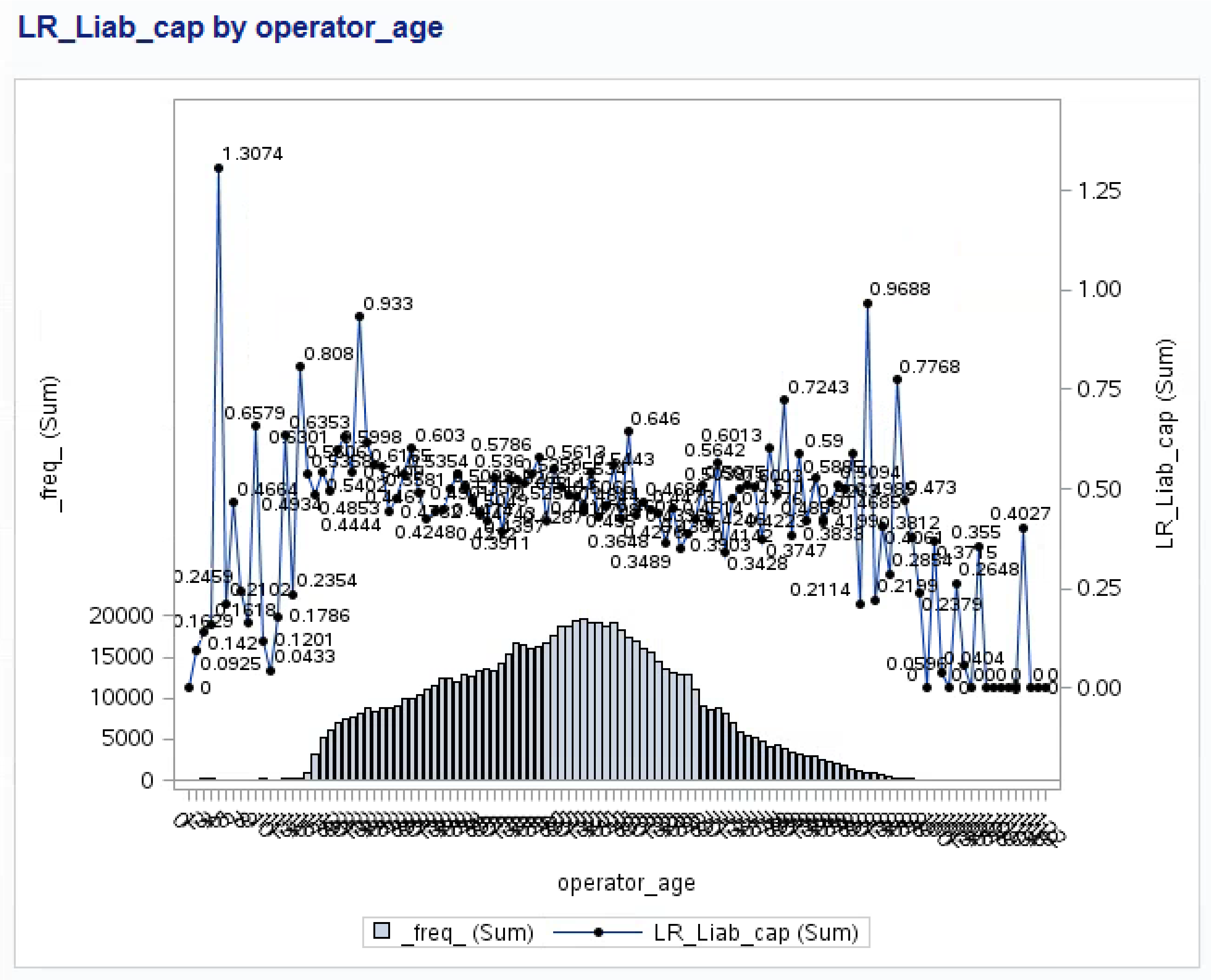
It is very frequent also, that several variables are sparsely populated (greater than90 percent missing values), so we need to address missing values. Normally, we would use imputation methodology to replace missing values with statistically generated values. However, we theorize that such a high number of imputed missing values could skew the distribution of the original variables and bias our predictions. Instead, we address missing values by optimal binning. We consider dropping the highly missing variables, but, given the high number of missing values, we realized these might have a significant effect in a predictive model.

For numeric variables with missing values, we decide to create "optimal bins." Optimal bins are created by converting a numeric variable into a categorical variable in such a way that its relationship with the target is maximized in terms of a purity measure, such as chi-square, Gini coefficient, F-Test, variance reduction, etc.

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 operator age. In the following graph we can see the relationship between the Loss Ratio and the operator age:



In the above plot, the operator age variable has been discretized in many levels. It is not easy to find a clear relationship.

We would like to see a clear pattern using an optimal criterion. 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 continuous variable, 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 as starting point 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 chi-square 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:

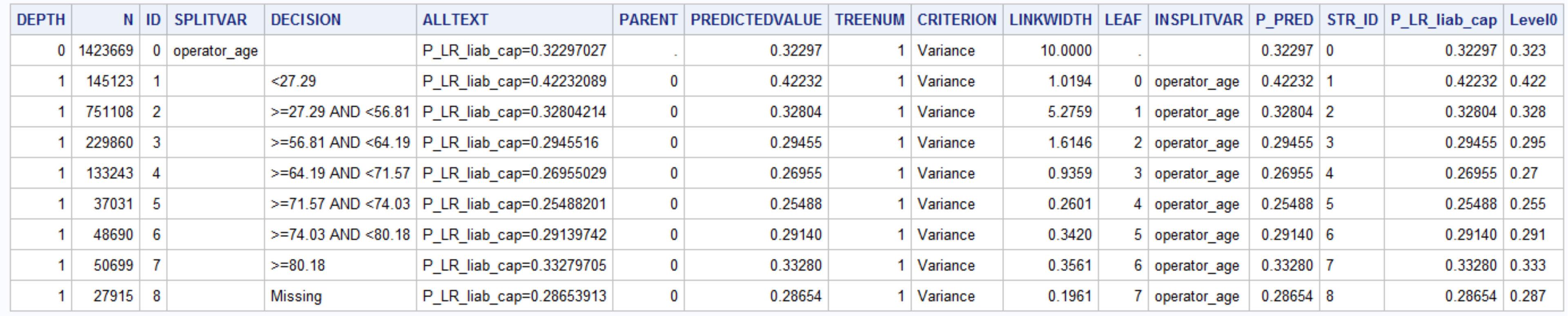
```
ods graphics on;
proc hpsplit data=Auto_DS_BT maxdepth=1 missing = branch /*intervalbins= 10*/ maxbranch=8
leafsize=5000 alpha=0.1;
criterion variance; /*default for interval target VARIANCE or FTEST */
prune none;
target LR_liab_cap / level = int;
input operator_age / level = int;
output nodestats=stat;
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, a credible amount. 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 OPERATOR_AGE. We used the level = int option to indicate that the explanatory variable is interval
- 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. In the example we selected the BRANCH method in order to create an specific group for the missing observations

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



DEPTH	N	ID	SPLITVAR	DECISION	ALLTEXT	PARENT	PREDICTEDVALUE	TREENUM	CRITERION	LINKWIDTH	LEAF	INSPLITVAR	P_PRED	STR_ID	P_LR_liab_cap	Level0
0	1423669	0	operator_age		P_LR_liab_cap=0.32297027	.	0.32297	1	Variance	10.0000	.		0.32297	0	0.32297	0.323
1	145123	1		<27.29	P_LR_liab_cap=0.42232089	0	0.42232	1	Variance	1.0194	0	operator_age	0.42232	1	0.42232	0.422
1	751108	2		>=27.29 AND <56.81	P_LR_liab_cap=0.32804214	0	0.32804	1	Variance	5.2759	1	operator_age	0.32804	2	0.32804	0.328
1	229860	3		>=56.81 AND <64.19	P_LR_liab_cap=0.2945516	0	0.29455	1	Variance	1.6146	2	operator_age	0.29455	3	0.29455	0.295
1	133243	4		>=64.19 AND <71.57	P_LR_liab_cap=0.26955029	0	0.26955	1	Variance	0.9359	3	operator_age	0.26955	4	0.26955	0.27
1	37031	5		>=71.57 AND <74.03	P_LR_liab_cap=0.25488201	0	0.25488	1	Variance	0.2601	4	operator_age	0.25488	5	0.25488	0.255
1	48690	6		>=74.03 AND <80.18	P_LR_liab_cap=0.29139742	0	0.29140	1	Variance	0.3420	5	operator_age	0.29140	6	0.29140	0.291
1	50699	7		>=80.18	P_LR_liab_cap=0.33279705	0	0.33280	1	Variance	0.3561	6	operator_age	0.33280	7	0.33280	0.333
1	27915	8		Missing	P_LR_liab_cap=0.28653913	0	0.28654	1	Variance	0.1961	7	operator_age	0.28654	8	0.28654	0.287

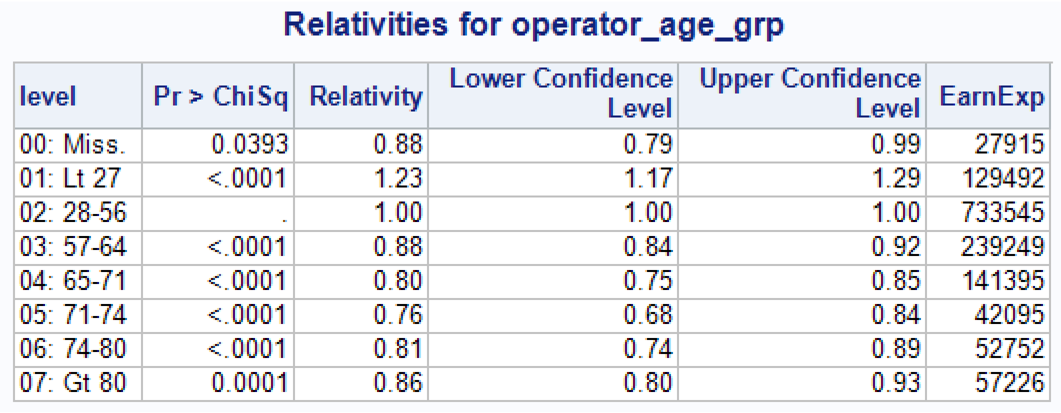
Table 15.5 NODESTATS= Data Set Variables

Variable	Target Type	Description
ALLTEXT	Either	Text that describes the split
CRITERION	Either	Which of the three criteria was used
DECISION	Either	Values of the parent variable's split to get to this node
DEPTH	Either	Depth of the node
ID	Either	Node number
LEAF	Either	Leaf number
LINKWIDTH	Either	Fraction of all training observations going to this node
N	Either	Number of training observations at this node
NVALID	Either	Number of validation observations at this node
PARENT	Either	Parent's node number
PREDICTEDVALUE	Either	Value of target predicted at this node
SPLITVAR	Either	Variable used in the split
TREENUM	Either	Tree number (always 1)
P_VARLEV	Nominal	Proportion of training set at this leaf that has target VAR = LEV
V_VARLEV	Nominal	Proportion of validation set at this leaf that has target VAR = LEV
P_VAR	Interval	Average value of target VAR in the training set
V_VAR	Interval	Average value of target VAR in the validation set

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

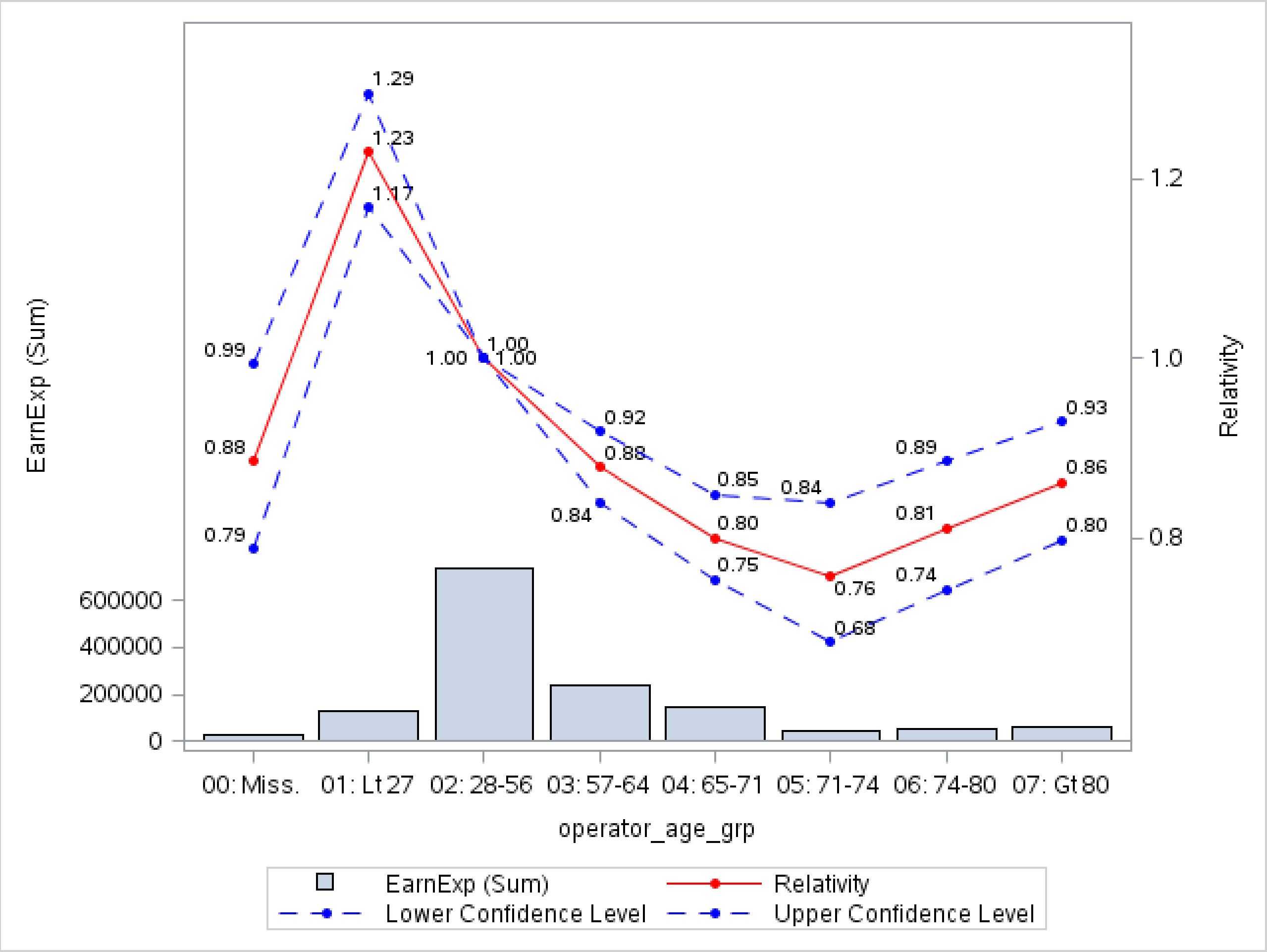
```
     if operator_age eq .  then operator_age_grp = "00: Miss.";
else if operator_age lt 27 then operator_age_grp = "01: Lt 27";
else if operator_age lt 56 then operator_age_grp = "02: 28-56";
else if operator_age lt 64 then operator_age_grp = "03: 57-64";
else if operator_age lt 71 then operator_age_grp = "04: 65-71";
else if operator_age lt 74 then operator_age_grp = "05: 71-74";
else if operator_age lt 80 then operator_age_grp = "06: 74-80";
else if operator_age ge 80 then operator_age_grp = "07: Gt 80";
```

The relativities for the OPERATOR_AGE_GRP variable in a multivariate model are (training dataset):

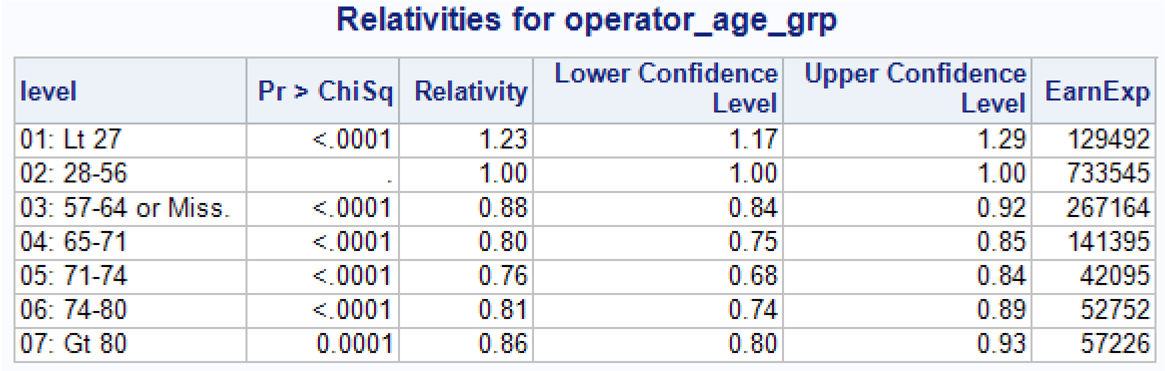


Relativities for operator_age_grp

level	Pr > ChiSq	Relativity	Lower Confidence Level	Upper Confidence Level	EarnExp
00: Miss.	0.0393	0.88	0.79	0.99	27915
01: Lt 27	<.0001	1.23	1.17	1.29	129492
02: 28-56	.	1.00	1.00	1.00	733545
03: 57-64	<.0001	0.88	0.84	0.92	239249
04: 65-71	<.0001	0.80	0.75	0.85	141395
05: 71-74	<.0001	0.76	0.68	0.84	42095
06: 74-80	<.0001	0.81	0.74	0.89	52752
07: Gt 80	0.0001	0.86	0.80	0.93	57226

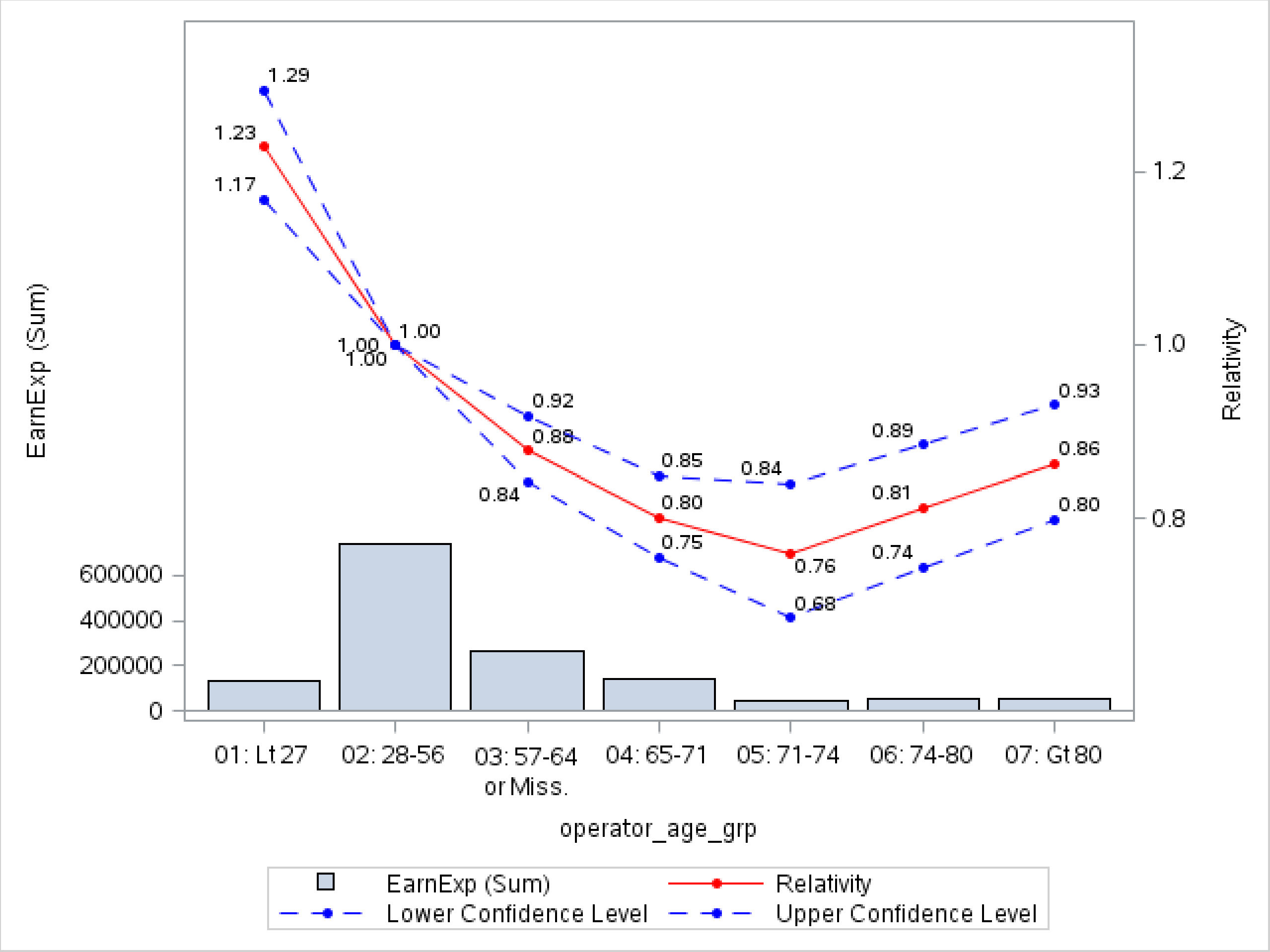


Now the characteristic 'U' shape relationship for the driver age arise clearly. Also, it is clear that the missing observations should be collapsed with the drivers between 57 and 65 years.



Relativities for operator_age_grp

level	Pr > ChiSq	Relativity	Lower Confidence Level	Upper Confidence Level	EarnExp
01: Lt 27	<.0001	1.23	1.17	1.29	129492
02: 28-56	.	1.00	1.00	1.00	733545
03: 57-64 or Miss.	<.0001	0.88	0.84	0.92	267164
04: 65-71	<.0001	0.80	0.75	0.85	141395
05: 71-74	<.0001	0.76	0.68	0.84	42095
06: 74-80	<.0001	0.81	0.74	0.89	52752
07: Gt 80	0.0001	0.86	0.80	0.93	57226



Summary

- The new HPSPLIT can be used as an auxiliary tool during the modeling process creating an optimal binning for interval or continuous variables.
- 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
- Because the HPSPLIT is a high performance (HP) procedure, the tree fitting takes only few seconds. Therefore, it is really easy to use a macro to apply this procedure to other variables.

Appendix: Regression Trees, Partition Criteria and Heteroscedasticity

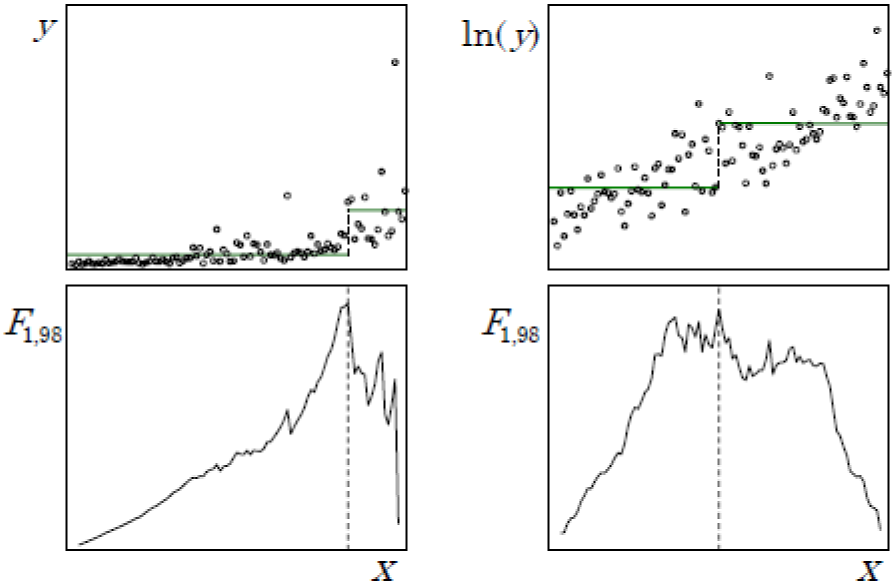
There are two splitting criteria used in regression tress:

- 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.

Appendix: The INTERVALBINS option in the HPSPLIT Statement

There is an option of the HPSPLIT statement called INTERVALIBINS that devoted to create a specific number of bins in an interval variable. Here we have a description of this option.

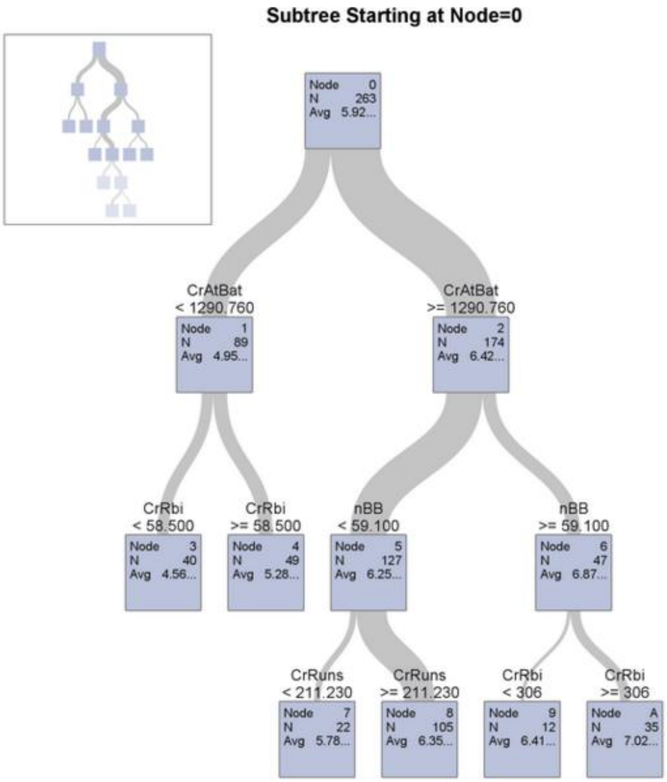
INTERVALBINS=number specifies the number of bins for interval variables. By default, INTERVALBINS=100.

This option only works if we don't use the MAXDEPTH=1 option and obviously the MAXBRANCH=8 option.

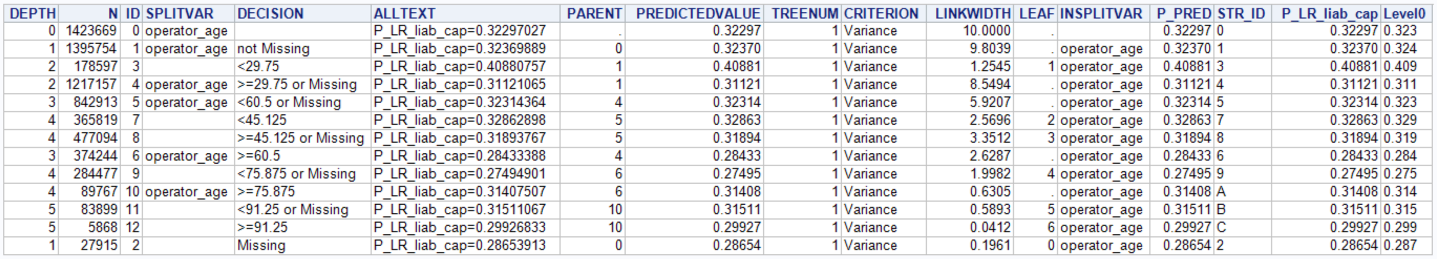
```
ods graphics on;
proc hpsplit data=Auto_DS_BT /*maxdepth=1*/ missing = branch intervalbins=8
/*maxbranch=8*/  leafsize=5000 alpha=0.1;
criterion variance; /*default for interval target VARIANCE or FTEST */
prune none;
target LR_liab_cap / level = int;
input operator_age / level = int;
output nodestats=stat;
run;

title 'Tree visualization';
proc print data=stat noobs;
run;
```

Without the MAXDEPTH option equal to one, we get also 8 levels, but, in this case, the decision tree has 4 depth levels. In our current version of SAS STAT 13.2, a graphical decision tree easy to interpret, is not implement. Only in the new SAS STAT 14.1 we can see in SAS E-Guide something like this:



Therefore, we get a relatively more complex table that is not obvious how interpret without experience with decision trees:



DEPTH	N	ID	SPLITVAR	DECISION	ALLTEXT	PARENT	PREDICTEDVALUE	TREENUM	CRITERION	LINKWIDTH	LEAF	INSPLITVAR	P_PRED	STR_ID	P_LR_liab_cap	Level0
0	1423669	0	operator_age		P_LR_liab_cap=0.32297027	.	0.32297	1	Variance	10.0000	.		0.32297	0	0.32297	0.323
1	1395754	1	operator_age	not Missing	P_LR_liab_cap=0.32369889	0	0.32370	1	Variance	9.8039	.	operator_age	0.32370	1	0.32370	0.324
2	178597	3		<29.75	P_LR_liab_cap=0.40880757	1	0.40881	1	Variance	1.2545	1	operator_age	0.40881	3	0.40881	0.409
2	1217157	4	operator_age	>=29.75 or Missing	P_LR_liab_cap=0.31121065	1	0.31121	1	Variance	8.5494	.	operator_age	0.31121	4	0.31121	0.311
3	842913	5	operator_age	<60.5 or Missing	P_LR_liab_cap=0.32314364	4	0.32314	1	Variance	5.9207	.	operator_age	0.32314	5	0.32314	0.323
4	365819	7		<45.125	P_LR_liab_cap=0.32862898	5	0.32863	1	Variance	2.5696	2	operator_age	0.32863	7	0.32863	0.329
4	477094	8		>=45.125 or Missing	P_LR_liab_cap=0.31893767	5	0.31894	1	Variance	3.3512	3	operator_age	0.31894	8	0.31894	0.319
3	374244	6	operator_age	>=60.5	P_LR_liab_cap=0.28433388	4	0.28433	1	Variance	2.6287	.	operator_age	0.28433	6	0.28433	0.284
4	284477	9		<75.875 or Missing	P_LR_liab_cap=0.27494901	6	0.27495	1	Variance	1.9982	4	operator_age	0.27495	9	0.27495	0.275
4	89767	10	operator_age	>=75.875	P_LR_liab_cap=0.31407507	6	0.31408	1	Variance	0.6305	.	operator_age	0.31408	A	0.31408	0.314
5	83899	11		<91.25 or Missing	P_LR_liab_cap=0.31511067	10	0.31511	1	Variance	0.5893	5	operator_age	0.31511	B	0.31511	0.315
5	5868	12		>=91.25	P_LR_liab_cap=0.29926833	10	0.29927	1	Variance	0.0412	6	operator_age	0.29927	C	0.29927	0.299
1	27915	2		Missing	P_LR_liab_cap=0.28653913	0	0.28654	1	Variance	0.1961	0	operator_age	0.28654	2	0.28654	0.287

Note that, the above tree shape graph and many other option are available at SAS Enterprise Miner. Here, we only try to show how to use SAS E-Guide to fit a basic decision tree instead to use SAS E-Miner.

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
- 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
- SAS Enterprise Miner 13.2 High Performance Data Mining Nodes Help
- Machine Learning With SAS® Enterprise Miner™ How a Team of SAS® Modelers Created and Determined a Champion Model to Predict Churn Using KDD Cup Data
- Ratemaking Using SAS® Enterprise Miner™: An Application Stud Billie Anderson, SAS Institute Inc., Cary, NC