Macros In Scoring

Jukui Yi , First Union National Bank , Charlotte, NC

ABSTRACT

In the banking industry, scores are commonly used to rank customers in a number of areas, including their propensity to purchase a new product, their profitability, and their probability of default. Scores are critical to the success of a company. Developing a good score can take weeks or, even months. After collecting the data, for most score builders, most of their time is devoted to studying which attributes have the best predictive power; and then deciding how to use the chosen attributes to create the scoring. After the scoring is done, its predictability and accuracy must be validated. Its robustness, that is, its sensitivity to changes in the data, must also be checked. This paper will demonstrate a number of time-saving approaches in scoring with utilization of SAS Macros at all levels. Although the examples are from the banking industry, the approach can be applied anywhere. Some familiarity with SAS/STAT and modeling concepts would be helpful. Relevant references are provided.

INTRODUCTION

In the banking industry, scoring devices are critical to the success of a company. This paper will summarize the basic steps in the development of a scoring device, ranging from the preparation of data to methods of building models. We will also show each score building method's advantages and disadvantages. Performance comparison of the various score models will be provided. We will demonstrate how to use SAS macros to save time on a scoring project.

DATA PREPARATION

To make a predictive score by data mining (AKA, predictive modeling) techniques, first we need to have a clean data.

Data cleaning(AKA, data cleansing) and data preparation is a process of ensuring that all values in a data set are correctly and meaningfully recorded. This process also includes missing data replacement and anomalous data treatment. Derivation is very important, i. e., the creation of new independent variables as functions of existing variables. The new fields are created in the expectation that they will be better predictors than the old ones. Another important consideration is to define the right type (character or numeric) for each variable.

Numeric type: continuous and its amplitude will be taken into account.
Character type: not continuous and each value is considered an independent group.

SCORING METHODS

After the data is collected and prepared, one needs to decide how to create a score and by what methods. The most frequently used methods in the modeling industry may be summarized into the following 6 categories:

1. ALL BASIC DUMMY VARIABLES

Make new dummy independent variables out of existing independent variables according to the relationship between the dependent variable and the independent variables. The final score is created based on such basic dummy variables only.

`See the "Sample Code Table".`

Advantage: easy to learn, easy to code and no interaction (can get decline-reason-code).

Disadvantage: Data-sensitive (can be very good on development data, but not good at validation data).

2. ADVANCED COMBINATIONS OF LINEAR SPLINES AND DUMMY VARIABLES

Make both new dummy independent variables and linear splines out of existing independent variables according to the relationship between the dependent variable and the independent variables. The final score is created based on such basic dummy variables and linear splines only.

`See the "Sample Code Table".`

Advantage: easy to code, less data sensitive and no interaction(can get decline-reason-code).

Disadvantage: need to sacrifice part of the information contained in the data.

3. COMBINATIONS OF POLYNOMIAL SPLINES (AKA, POWER LADDER) AND DUMMY VARIABLES

Make both new dummy independent variables and polynomial splines out of existing independent variables according to the relationship between the dependent variable and the independent variables.
The final score is created based on such basic dummy variables and polynomial splines only.
See the "Sample Code Table".

Advantage: may be very good for some special case and no interaction(can get decline-reason-code).

Disadvantage: Data sensitive(may be very good on development data, but not good at validation data).

4. TREE

Chi-square automatic interaction detection (CHAID) - A decision tree technique used for predictive classification of a data set, or prediction of a continuous value. It provides a set of rules that you can apply to a new (unclassified) data set to predict which records will have a given outcome. It segments a data set by using chi-square tests to create multi-branch splits.

Classification and regression trees (CART) - A decision tree technique that can be used for predictive classification of a data set, or prediction of a continuous value. It provides a set of rules that can be applied to a new (unclassified) data set to predict which records will have a given outcome. It segments a data set by creating two branch splits only.

Advantage: makes use of the power of interactions.

Disadvantage: Hard to get decline-reason-code, and the final result is too rough. CHAID has no good facility for dealing with missing or anomalous data.

CART is better at dealing with missing or anomalous data than CHAID since it segments a data set by creating two-branch splits only (according to proponents of this technique).

5. NEURAL NETWORKS.

Neural network - Non-linear predictive models that learn both structure and parameter values through training, which also superficially resemble biological neural networks in structure. It could be very complicated.

Advantage: It could be complicated enough to fit every event on the development data.

Disadvantage: It's a "black box" and sometimes the results are unexplainable.

6. SMOOTHING SPLINES (AKA, PIECE-WISE, OR CHAIN).

Using connected piece-wise curves (linear or non-linear)to fit an underlying response structure. (A bike chain comes to mind.) It is more flexible in capturing any type of intrinsic trend in the data.

The smoothness between “joints” is considered for continuous independent variables. The smoothing technology will also help to get rid of the abnormal turn-points.

Special rules:

(A) Separation of Missing and None-missing (Known / Unknown)
(B) Separation of Zero and None-zero (Have / Have Not)
(C) No individual dummy will cross zero.

Advantage: Very flexible, less data sensitive and no interaction(easy to get decline-reason-code).
Disadvantage: .

COMPARISON AMONG METHODS

The 2 examples on the next 3 pages are illustrations of how this technology works (how to fit the "chain" into each variable's intrinsic structure smoothly).

There are some other methods available, such as, clustering, topit, neural-fuzzy, artificial intelligence, classification tree, data visualization, fuzzy logic, Multivariate Adaptive Regression Splines (MARS), Knowledge discovery, etc. There are still non-technical issues to be considered, e.g., the Equal Credit Opportunity Act (ECOA). Model builders must obey ECOA to avoid using certain variables, e.g., in a credit solicitation score, one must avoid using sex, marital status, or even age.

We'll compare various techniques (The final scorecard can be easily coded by using any computer language) in cases 1,2,3 and 6. We are going to compare both the techniques and the performances.

The *Sample Code Table* on page 6 shows how to make the splines and dummy variables of cases 1,2 and 3. We'll use the codes on the data set from which they are made to create scores, then compare the performance of the scores.

Case 6 is a little more complicated, we'll show you how to use the SAS macros to get the 2 examples on the last 2 pages done and explain how this technique works.

Explanations of the 2 examples:
the top 2 boxes on the next 2 pages are 2 types of partition method used to partition each independent variables and the statistics(page 3 for attr0001 and page 4 for attr0005), and the charts are visual representations of the boxes above.
Partition by Population: sort the independent variable, then group the accounts by equal size.
Partition by Value: sort the independent variable, then group the accounts by equal length in the variable's value.

The 2 boxes on page 5 shows the Smoothing Splines' partitioning of the independent variables(attr0001 and attr0005) and the statistics.
The charts below the boxes are the visual representation of the 2 box.

CONCLUSION

Of all the different methods, each has its own advantages and disadvantages, the performing statistics are listed below:

Methods	Data Set	K. S. Difference	Concordant	Lowest 5% Contains % of Bad
All Basic Dummy	Development	47.09%	81.10%	24.70%
	Validation	44.70%		24.48%
Linear Splines and Dummy	Development	43.22%	78.80%	23.88%
	Validation	42.84%		22.73%
Polynomial Splines and Dummy	Development	45.38%	79.20%	25.54%
	Validation	40.72%		23.43%
Smoothing Splines	Development	45.93%	80.10%	24.38%
	Validation	44.83%		24.83%

As seen in the table above, overall, the Smoothing Splines method provides the best performance with more robust and more explainable score cards.

```
%macro zipsort(lib,dsn,byvar);
data _null_;if 0 then set &lib..&dsn nobs=num;
call symput('zcount',left(put(num,8.))); stop;run;%put &zcount;
%let yn= %eval( &zcount / 100000); %put &yn;
%if &yn >20 %then %let yn=20; %if &yn <4 %then %let yn=4;
%macro outdata(n);%do i=0 %to &n -1;zyzy&I %end;%mend;
%macro outdatao(n); %do i=0 %to &n -1;
%if &i <=&n -1 %then %do;if mod(_n_,&n )=&i then output zyzy&i;
%end; %else %do;if mod(_n_,&n )=&i then output zyzy&i;
%end;%end;%mend;
data %outdata(&yn);set &lib..&dsn;%outdatao(&yn);run;
%macro outdatas(n);%do i=0 %to &n -1;proc sort data=zyzy&i;
by &byvar; run;%end;%mend;
%outdatas(&yn);
data &lib..&dsn;set %outdata(&yn);by &byvar;run;
proc datasets lib=work nolist;delete %outdata(&yn);run;
%mend;

%zipsort(d,mod3,attr0005);
```

As an example, the above macro(zipsort) will help to sort large size data set. It not only speeds up the sorting time but also minimizes the sorting space(many people don't know why their data step failed when sorting a large size data set).

Note: Those who wish to obtain the SAS micros for the Smoothing Splines technology and try on their own computers may contact the author.

ACKNOWLEDGMENTS

I wish to thank Professor Robert Serfling at University of Texas at Dallas for his untiring encouragement and guidance over the years. His teaching has led to this article and much of my appreciation of statistical technology as a profession.

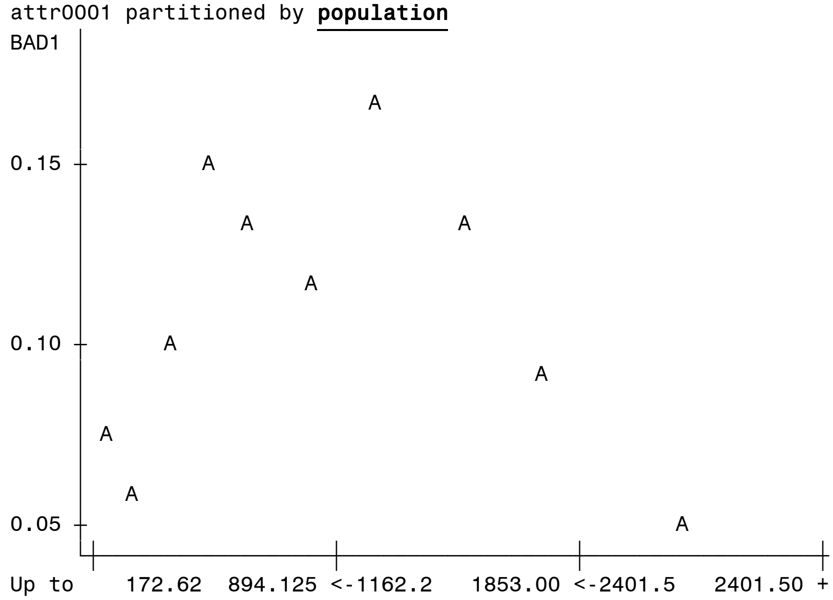
CONTACT INFORMATION

Author Name :Jukui Yi
Company: First Union National Bank
Address: 1525 W.W.T Harris Boulevard
City state ZIP: Charlotte, NC 28288-5427
Work Phone: (704) 427-0165
Fax: (704) 590-4028
Email: jukui.yi@firstunion.com
Web:

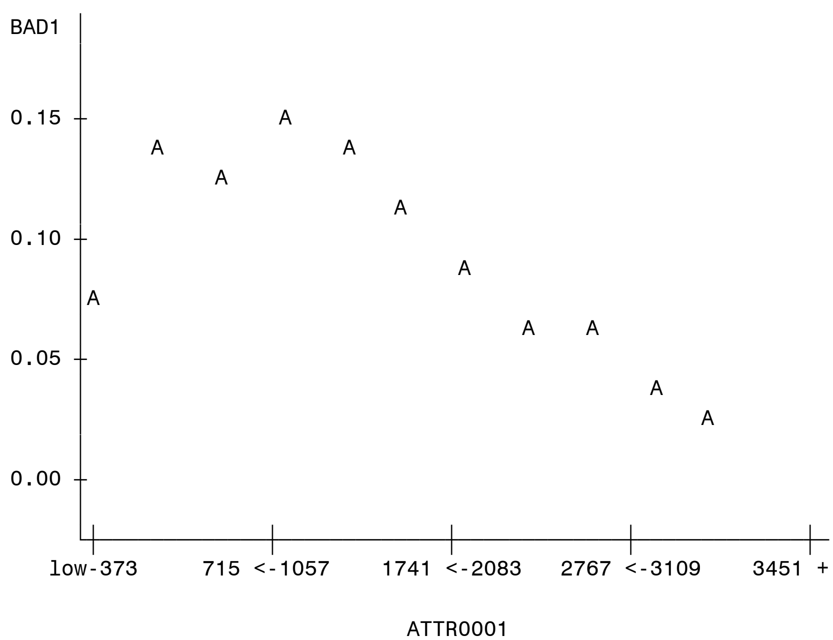
attr0001 partitioned by **population**				
Analysis Variable: BAD1				
ATTR0001	N Obs	N	Mean	Std Dev
Up to 172.62	600	549	0.0783	0.2689
172.625 <-320.43	599	548	0.0566	0.2312
320.438 <-467.43	599	555	0.0991	0.2991
467.438 <-653.12	600	550	0.1473	0.3547
653.125 <-894.12	599	552	0.1322	0.3391
894.125 <-1162.2	601	566	0.1184	0.3233
1162.25 <-1538.5	598	563	0.1652	0.3717
1538.50 <-1853.0	599	571	0.1296	0.3362
1853.00 <-2401.5	599	576	0.0955	0.2941
2401.50 +	599	591	0.0525	0.2231

attr0001 partioned by **value**				
Analysis Variable: BAD1				
ATTR0001	N Obs	N	Mean	Std Dev
low-373	1484	1360	0.0721	0.2587
373 <-715	1071	989	0.1335	0.3403
715 <-1057	833	773	0.1190	0.3240
1057 <-1399	580	545	0.1541	0.3614
1399 <-1741	660	630	0.1413	0.3486
1741 <-2083	469	449	0.1136	0.3177
2083 <-2425	314	301	0.0897	0.2862
2425 <-2767	226	221	0.0679	0.2521
2767 <-3109	138	136	0.0588	0.2362
3109 <-3451	96	96	0.0417	0.2009
3451 +	122	121	0.0248	0.1561

attr0001 partitioned by **population**



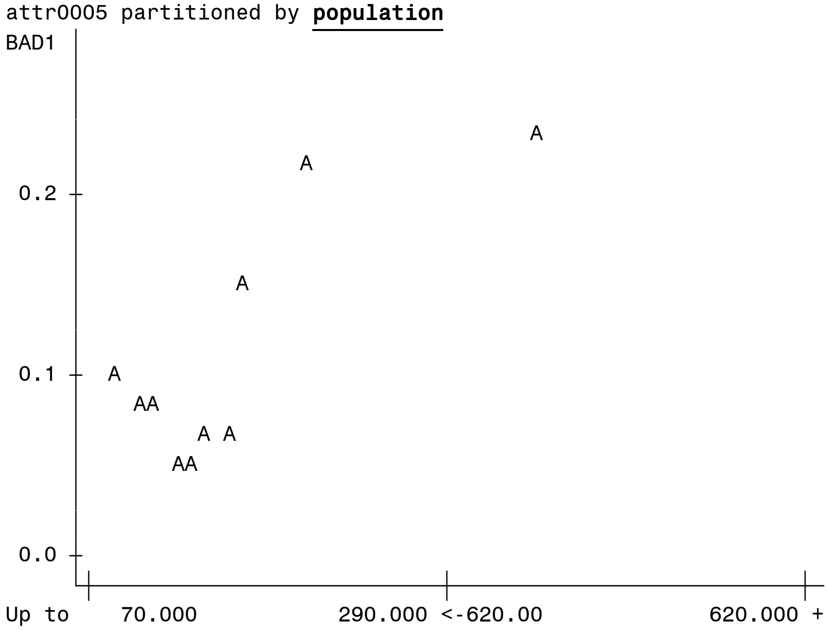
attr0001 partitioned by **value**



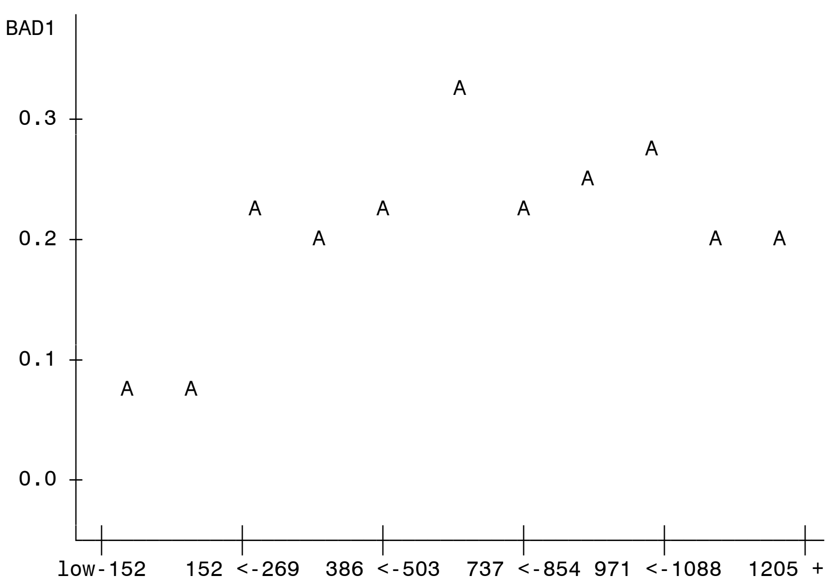
attr0005 partitioned by **population**				
Analysis Variable: BAD1				
ATTR0005	N Obs	N	Mean	Std Dev
Up to 70.000	711	667	0.0945	0.2927
70.0000 <-90.000	546	508	0.0886	0.2844
90.0000 <-120.00	638	608	0.0773	0.2673
120.000 <-140.00	643	611	0.0458	0.2093
140.000 <-160.00	562	537	0.0466	0.2109
160.000 <-185.00	627	591	0.0694	0.2543
185.000 <-210.00	549	524	0.0630	0.2432
210.000 <-290.00	520	487	0.1458	0.3533
290.000 <-620.00	605	541	0.2181	0.4133
620.000 +	592	547	0.2413	0.4283

attr0005 partioned by **value**				
Analysis Variable: BAD1				
ATTR0005	N Obs	N	Mean	Std Dev
low-152	2829	2673	0.0737	0.2613
152 <-269	1900	1801	0.0800	0.2713
269 <-386	200	177	0.2147	0.4118
386 <-503	261	227	0.2026	0.4029
503 <-620	211	196	0.2347	0.4249
620 <-737	131	120	0.3167	0.4671
737 <-854	130	117	0.2308	0.4231
854 <-971	93	89	0.2472	0.4338
971 <-1088	30	28	0.2857	0.4600
1088 <-1205	83	78	0.1923	0.3967
1205 +	125	115	0.1913	0.3950

attr0005 partitioned by **population**



attr0005 partitioned by **value**



ATTR0005

Attr0001 and attr0005 partitioned by **Smoothing Splines Technology**

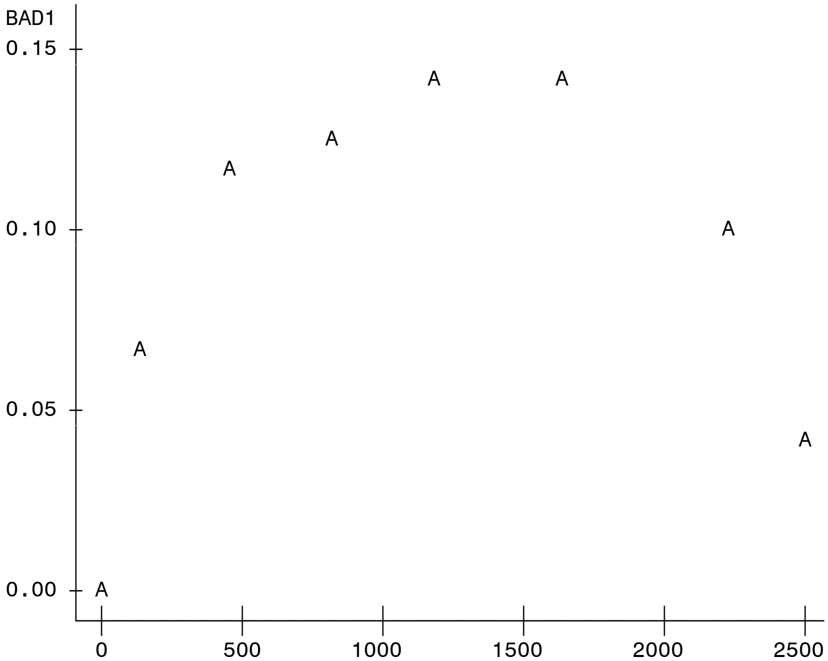
bad1 rates distribution
Analysis Variable: BAD1

ATTR0001	N Obs	N	Mean	Std Dev
Zero	1	1	0	.
1-308	1115	1019	0.0658	0.2480
309-614	1112	1025	0.1190	0.3240
615-1027	1100	1022	0.1243	0.3300
1028-1323	542	510	0.1431	0.3506
1324-1952	1073	1022	0.1380	0.3450
1953-2517	533	512	0.0977	0.2971
2518+	517	510	0.0451	0.2077

bad1 rates distribution
Analysis Variable: BAD1

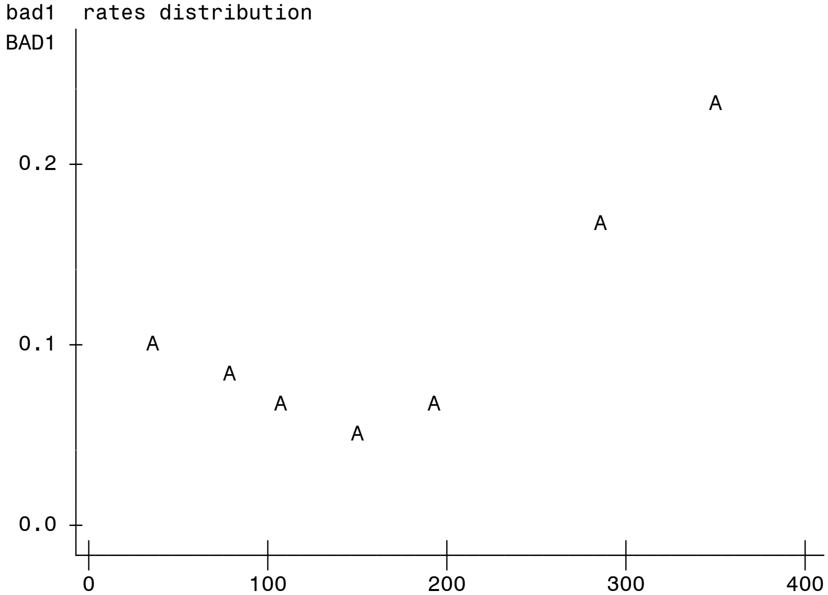
ATTR0005	N Obs	N	Mean	Std Dev
1--65	584	548	0.1004	0.3008
66-85	529	493	0.0852	0.2795
86-130	1067	1016	0.0719	0.2584
131-170	1154	1097	0.0465	0.2106
171-215	1023	972	0.0730	0.2603
216-350	516	473	0.1607	0.3676
351+	1120	1022	0.2299	0.4210

bad1 rates distribution



ATTR0001

bad1 rates distribution



ATTR0005

All Basic Dummy Variables	Advanced Combination Of Linear Splines and Dummy	Combination Of Polynomial Splines and Dummy
if 0 <= attr0001 <=100 then at001a01=1; else at001a01=0;	if attr0001<=1000 then bttr0001=attr0001; else bttr0001=0;	if attr0001 > 4000 then at001 = 4000 ; else
if 100 < attr0001 <=320 then at001a02=1; else at001a02=0;	if attr0001 >4000 then cttr0001=4000; else	if attr0001 < 170 then at001 = 170 ; else
if 465 < attr0001 <=895 then at001a03=1; else at001a03=0;	if attr0001 >1000 then cttr0001=attr0001; else cttr0001=0;	at001 = attr0001 ;
if 1160 < attr0001 <=1540 then at001a04=1; else at001a04=0;		at001s = at001 * at001 /100;
if 2400 < attr0001 then at001a05=1; else at001a05=0;	if attr0002<= -5 then bttr0002=-5; else	
if attr0002 <=-3.8 then at002a01=1; else at002a01=0;	if attr0002<= 0 then bttr0002=attr0002; else bttr0002=0;	if attr0002 > 6 then at002 = 6 ; else
if -3.8< attr0002 <=-3.6 then at002a02=1; else at002a02=0;	if attr0002 > 6 then cttr0002=6 ; else	if attr0002 < -4.5 then at002 = -4.5 ; else
if -3.6< attr0002 <=-2.95 then at002a03=1; else at002a03=0;	if attr0002 > 0 then cttr0002=attr0002; else cttr0002=0;	at002 = attr0002;
if -1.45< attr0002 < 0 then at002a04=1; else at002a04=0;		at002c = at002 * at002 *at002;
if 0<= attr0002 then at002a05=1; else at002a05=0;		
if attr0004 <= 55 then at004a01=1; else at004a01=0;	if attr0004<=100 then bttr0004=100; else	if attr0004 > 2000 then at004 = 2000 ; else
if 55 < attr0004 <= 115 then at004a02=1; else at004a02=0;	if attr0004>2000 then bttr0004=2000;else bttr0004=attr0004;	if attr0004 < 100 then at004 = 100 ; else
if 115 < attr0004 <= 180 then at004a03=1; else at004a03=0;		at004 = attr0004 ;
if 180 < attr0004 <= 280 then at004a04=1; else at004a04=0;	if attr0005<=150 then bttr0005=attr0005; else bttr0005=0;	at004s = at004 * at004 /1000;
if 540 < attr0004 <= 640 then at004a05=1; else at004a05=0;	if attr0005 >900 then cttr0005=900; else	
if 640 < attr0004 <= 815 then at004a06=1; else at004a06=0;	if attr0005 >150 then cttr0005=attr0005; else cttr0005=0;	if attr0005 <=150 then at005a=attr0005; else at005a=0;
if 815 < attr0004 <=1155 then at004a07=1; else at004a07=0;		if attr0005 > 150 then at005b=attr0005; else at005b=0;
if 1155 < attr0004 then at004a08=1; else at004a08=0;	if attr0007 >4.75 then bttr0007=1; else bttr0007=0;	if at005b>875 then at005b=875;
if 80 < attr0005 <= 120 then at005a01=1; else at005a01=0;	if 4<=attr0007 <=4.75	
if 120 < attr0005 <= 170 then at005a02=1; else at005a02=0;	or attr0007=0 then cttr0007=1; else cttr0007=0;	if attr0007 > 5.2 then at007 = 5.2 ; else
if 170 < attr0005 <= 225 then at005a03=1; else at005a03=0;		if attr0007 = 0 then at007 = 4 ; else
if 225 < attr0005 then at005a04=1; else at005a04=0;	if attr0008<=0 then bttr0008=0;	if attr0007 > 3.9 then at007 = attr0007; else at007 = 0;
if attr0007 = 0	if attr0008 <=2.25 then bttr0008=attr0008; else bttr0008=0;	
or 3.90 < attr0007 <= 4.75 then at007a01=1; else at007a01=0;	if 4.65>=attr0008 >2.25	if 1.6 <= attr0008 <= 3.15 then at008a01=1; else at008a01=0;
if 4.75 < attr0007 then at007a02=1; else at007a02=0;	then cttr0008=attr0008; else cttr0008=0;	if 3.15<= attr0008 <= 12 then at008a02=1; else at008a02=0;
if 2.1 < attr0008 <= 3.10 then at008a01=1; else at008a01=0;		
if 3.10 < attr0008 then at008a02=1; else at008a02=0;	if attr0033 = 'LL' then bttr0033=1; else bttr0033=0;	if attr0033 ='MM' then at033a01=1; else at033a01=0;
if attr0033 ='MM' then at033a01=1; else at033a01=0;	if attr0033 = 'MM' then cttr0033=1; else cttr0033=0;	if attr0033 ='SS' then at033a02=1; else at033a02=0;
if attr0033 ='SS' then at033a02=1; else at033a02=0;		
if attr0036 in ('N','Y') then at036a01=1; else at036a01=0;	if attr0036 in ('N','Y') then bttr0036=1; else bttr0036=0;	if attr0036 in ('N','Y') then at036a01=1; else at036a01=0;
Keep bad1	Keep bad1	Keep bad1
at001a01-at001a05 at002a01-at002a05 at004a01-at004a08	bttr0001 cttr0001 bttr0002 cttr0002 bttr0004	at001 at001s at002 at002c at004 at004s
at005a01-at005a04 at007a01-at007a02 at008a01-at008a02	bttr0005 cttr0005 bttr0007 cttr0007 bttr0008 cttr0008	at005a at005b at007 at008a01 -at008a02
at033a01-at033a02 at036a01 ;	bttr0033 cttr0033 bttr0036 ;	at033a01 -at033a02 at036a01 ;