CHAPTER 17
PREDICTIVE POWER AND VARIABLE REDUCTION II

17.1 INTRODUCTION

Chapter 12 presented the main scheme of variable reduction. In that scheme there are two main reduction mechanisms: (1) the reduction of candidate independent variables with respect to a dependent variable and (2) the reduction of a set of variables into a smaller set while keeping most of the information content. To pursue the latter mechanism, there are two methods for variable reduction, namely, principal component analysis and factor analysis (discussed in Chapters 15 and 16, respectively).

This chapter pursues the first mechanism of data reduction using some of the concepts and techniques explored in Chapters 13 and 14 for the definition of measures of association between variables.

The majority of data mining predictive models focus on two tasks: classification and estimation. Most classification models have binary dependent variables. In classification models where the dependent variable (DV) has more than two categories, it is common practice to map the DV into indicator variables and use these new variables to build different models with binary DVs. In the case of estimation models, the dependent variable is always continuous. Therefore, we focus our attention now on these two types of dependent variables: binary and continuous. However, we explore *all* types of independent variables (IVs).

17.2 DATA WITH BINARY DEPENDENT VARIABLES

17.2.1 NOTATION

The following notation will be adopted for the case of binary dependent variables.

y = The binary dependent variable y (i.e., taking values 0 and 1)

N = Total number of records in the dataset

N_1, N_2 = Total number of records with y in the two categories 1 and 2

N_k^j = Number of records with $y = k$ and IV equal to the category j

n_j = Total number of records with IV equal to category j in dataset

m = Number of categories in IV

17.2.2 Nominal Independent Variables

We present here three common measures of predictive power based on

- The Pearson Chi-squared statistic
- The Gini diversity measure
- The entropy variance

The Pearson Chi-squared statistic was defined in Equation 13.16 (see also Agresti [2002]). The notation follows the conventions of contingency tables. Using the simplified notation just listed, we write it as

$$X^2 = \sum_{j=1}^{m} \sum_{k=1}^{q} \frac{\left(N - \left(\frac{n_j N_k}{N}\right)\right)^2}{\left(\frac{n_j N_k}{N}\right)}, \tag{17.1}$$

where q is the number of categories of the dependent variable, which is 2 in the current case of binary DV.

The statistic follows a χ^2 distribution with $(m-1)(q-1)$ degrees of freedom and is produced by `PROC FREQ`. The implementation of this statistic was introduced in macro `PearChi()` in Section 13.4.3.

The Gini ratio is defined (see Breiman et al. 1998) as

$$G_r = 1 - \frac{G_o}{G_i}, \tag{17.2}$$

where

$$G_i = 1 - \frac{\sum_{k=1}^{q} N_k^2}{N^2}, \tag{17.3}$$

and

$$G_o = \sum_{j=1}^{m} \frac{n_j G_j}{N} \left(1 - \frac{\sum_{k=1}^{q} (N_k^j)^2}{(n_j)^2}\right). \tag{17.4}$$

Finally, the Entropy ratio is defined (see Ross 1993) as

$$E_r = 1 - \frac{E_o}{E_i}, \tag{17.5}$$

where

$$E_i = -\frac{1}{\ln q} \sum_{k=1}^{q} \frac{N_k}{N} \ln \frac{N_k}{N}, \tag{17.6}$$

and

$$E_o = \sum_{j=1}^{m} \frac{n_j}{N} \left(-\frac{1}{\ln q} \sum_{k=1}^{q} \frac{N_k^j}{N} \ln \left[\frac{N_k^j}{N} \right] \right). \tag{17.7}$$

The implementation of the Gini and Entropy ratios is straightforward. It involves finding the different counts in Equations 17.2 and 17.5. The following two macros are almost identical, with the exception of the part that actually substitutes in the formulas (see Table 17.1).

Table 17.1 Parameters of macro `GiniCatBDV()`.

Header	`GiniCatBDV(DSin, XVar, DV, M_Gr);`
Parameter	*Description*
`DSin`	Input dataset
`XVar`	The nominal independent variable
`DV`	The binary dependent variable name
`M_Gr`	Output Gini ratio

Step 1

Count the frequencies of cross-tabulating the variable `XVar` versus the dependent variable `DV` using `PROC FREQ`. Also count the number of categories in the independent variable `XVar` by using a second `TABLE` statement to generate the unique categories and then count them using `PROC SQL`.

```
proc freq data=&DSin noprint;
 table &XVar*&DV /out=Temp_freqs;
 table &XVar /out=Temp_cats;
 run;
proc sql noprint;
  /* Count the number of categories */
   %local m;
   select count(*) into : m from temp_cats;
```

Step 2

Calculate the frequencies N_1, N_2 for DV values of 0 and 1, respectively, and substitute in the expression for G_i.

```
/* frequencies of DV=1, DV-0 , N*/
   %local N0 N1 N;
   Select sum(Count) into :N0 from temp_freqs where DV=0;
   select sum(Count) into :N1 from temp_freqs where DV=1;
quit;
%let N=%eval(&N0+&N1);

/* Gi */
%local Gp;
%let Gp=%sysevalf(1 - (&N0*&N0+&N1*&N1 ) / (&N*&N) );
```

Step 3

Extract the unique values for the independent variable `XVar` to use them in the queries.

```
data _null_;
 set temp_cats;
  call symput('Cat_'|| left(_N_), &XVar );
run;
```

Step 4

Loop over the categories of `XVar`, extract the frequencies N_k^j, n_j, and substitute in the expressions for G_o and G_r.

```
proc sql noprint;
%local ss i Ghat NN0 NN1 NN;
%let ss=0;
%do i=1 %to &m;
  /* get n_o^i (NN0) , n_1^i (NN1) */
  select max(0,sum(count)) into :NN0
    from temp_freqs where DV=0 and &XVar="&&Cat_&i";
  select max(0,sum(count)) into :NN1
    from temp_freqs where DV=1 and &XVar="&&Cat_&i";
  %let NN=%eval(&NN1+&NN0);
  %let ss=%sysevalf(&ss+ (1-((&NN0 * &NN0)+
            (&NN1 * &NN1))/(&NN * &NN)) * &NN);
%end; /* end of variable loop */
quit;
%let Ghat=%sysevalf(&ss/&N);
%let &M_Gr=%sysevalf(1-&Ghat/&Gp);
```

Step 5
Clean the workspace and finish the macro.

```
proc datasets library=work;
delete temp_freqs temp_cats;
quit;
%mend;
```

The next macro is for the calculation of the Entropy ratio E_r, defined in Equation 17.5. Since the calculation of E_r requires the same quantities needed to calculate the Gini ratio G_r, the macro is almost identical to the macro `GiniCatBDV()`. Except in the calculation of E_r, it follows the same steps (see Table 17.2).

Table 17.2 Parameters of macro `EntCatBDV()`.

Header	`EntCatBDV(DSin, XVar, DV, M_Er);`
Parameter	*Description*
`DSin`	Input dataset
`XVar`	The nominal independent variable
`DV`	The binary dependent variable name
`M_Er`	Output Entropy ratio

Steps 1 and 2
Extract frequencies needed later in calculations and compute E_i.

```
proc freq data=&DSin noprint;
 tables &XVar*&DV /out=Temp_freqs;
 table &XVar /out=Temp_cats;
 run;
proc sql noprint;
  /* Count the number of categories */
   %local m;
   select count(*) into : m from temp_cats;

/* frequencies of DV=1, DV-0 , N*/
%local N0 N1 N;
Select sum(Count) into :N0 from temp_freqs where DV=0;
select sum(Count) into :N1 from temp_freqs where DV=1;
%let N=%eval(&N0+&N1);
/* Ei */
%local Ein;
```

```
%let Ein=%sysevalf(  -1* ( &N0 * %sysfunc(log(&N0/&N))
                          +&N1 * %sysfunc(log(&N1/&N)) )
                        /( &N * %sysfunc(log(2))       ) );
quit;
```

Step 3

Get the unique categories of `XVar`.

```
data _null_;
 set temp_cats;
  call symput('Cat_'|| left(_N_), &XVar );
run;
```

Step 4

Loop on the variables, extract the frequencies from the cross-tabulation results, and substitute in the final entropy ratio equation. The complex condition on testing zero frequencies is used to prevent numerical errors because the `ln(·)` function does not admit a zero argument.

```
proc sql noprint;
%local ss i Eout NN0 NN1 NN;
%let ss=0;
%do i=1 %to &m;
 /* get n_o^i (NN0) , n_1^i (NN1) */
  select max(sum(count),0) into :NN0
    from temp_freqs where DV=0 and &XVar="&&Cat_&i";
  select max(sum(count),0) into :NN1
    from temp_freqs where DV=1 and &XVar="&&Cat_&i";
  %let NN=%eval(&NN1+&NN0);
  %if(&NN0>0 and &NN1>0) %then
   %let  ss=%sysevalf(&ss- &NN*
                    (&NN0 * %sysfunc(log(&NN0/&NN))
                    + &NN1 * %sysfunc(log(&NN1/&NN)) )
                     /( &NN * %sysfunc(log(2)) ) );
  %else %if (&NN0=0)%then
   %let  ss=%sysevalf(&ss- &NN*
                    ( &NN1 * %sysfunc(log(&NN1/&NN)) )
                     /( &NN * %sysfunc(log(2)) ) );
   %else
   %let  ss=%sysevalf(&ss- &NN*
                    (  &NN0 * %sysfunc(log(&NN0/&NN)) )
                     /( &NN * %sysfunc(log(2)) ) );
%end; /* end of variable loop */
quit;
%let Eout=%sysevalf(&ss/&N);
%let &M_Er=%sysevalf(1-&Eout/&Ein);
```

Step 5
Clean the workspace and finish the macro.

```
proc datasets library=work;
delete temp_freqs temp_cats;
quit;
%mend;
```

17.2.3 Numeric Nominal Independent Variables

The macros `GiniCatBDV()` and `EntCatBDV()` assume that the nominal independent variable `XVar` is a string. In the case when `XVar` is numeric (e.g., binary 1/0), we need to modify these two macros such that the queries, which count the frequencies n_j and N_k^j, use numeric selection criteria. This would be achieved by modifying the `SELECT` statements of step 4 in both macros, as follows:

```
select max(sum(count),0) into :NN0
   from temp_freqs where DV=0 and &XVar=&&Cat_&i;
 select max(sum(count),0) into :NN1
   from temp_freqs where DV=1 and &XVar=&&Cat_&i;
```

It is possible to test the type of the variable `XVar` before executing the queries to select the appropriate form. However, to keep things simple, we create two dedicated macros to deal with numeric nominal variables, where the preceding queries are used. These macros are `GiniCatNBDV()` and `EntCatNBDV()`. (Note the *N* in the macro name.)

17.2.4 Ordinal Independent Variables

In the case of ordinal IVs, it is expected that the categories have been replaced by numeric scores. The Gini and entropy ratios do *not* take into account the order relationship between the categories of ordinal variables. However, the definitions given in Equations 17.2 and 17.5 could also be used to calculate G_r and E_r. Therefore, we use the macros `GiniCatNBDV()` and `EntCatNBDV()`. In addition, it is often better to keep the names of the macros applied to different variable types indicative of those types. Therefore, we make copies of these two macros under the names `GiniOrdBDV()` and `EntOrdBDV()` to stress their use with ordinal variables.

In addition to the Gini and entropy ratios, there are two correlation coefficients that could be used to evaluate the association with ordinal variables. These are the Pearson correlation coefficient and the Spearman correlation coefficient, r and r_s, defined in Equations 13.19 and 13.23, respectively.

In Section 13.6, we presented the macros `ContPear()` and `ContSpear()` for the calculation of r and r_s, respectively. They were wrappers for `PROC FREQ`. The following macro combines these two in one macro to calcualte both r and r_s (see Table 17.3).

Table 17.3 Parameters of macro `PearSpear()`.

Header	`PearSpear(DSin, XScore, YScore, M_R, M_RS);`
Parameter	*Description*
`DSin`	Input dataset
`XScore`	Name of the *X* variable scores
`YScore`	Name of the *Y* variable scores
`M_R`	The Pearson correlation coefficient
`M_RS`	The Spearman correlation coefficient

```
%macro PearSpear(DSin, XScore, YScore, M_R, M_RS);
/* Calculation of Pearson and Spearman correlation coefficients
   for ordinal variables using the scores given in XScore, YScore.
   The results are stored in M_R and M_RS */

proc freq data=&DSin noprint;
tables &XScore*&YScore / measures;
output measures out=temp_rs;
run;

data_NULL_;
 set temp_rs;
 call symput("rs", abs(_SCORR_));
 call symput("rr", abs(_PCORR_));
run;
%let &M_R=&rr;
%let &M_rs = &rs;
proc datasets nodetails;
 delete temp_rs;
quit;

%mend;
```

17.2.5 Continuous Independent Variables

To measure the predictive power of continuous IVs with binary dependent variables, we simply convert them to an ordinal scale using binning. We may use either equal-height or equal-width binning. Equal-width binning always allows the use of a simple linear scale to assign the ordinal scores for the resulting bins. This is not always the case with equal-height binning because the width of the bins in this case is not necessarily uniform. A simple and practical approach is to use the mid-point value of each bin as the score corresponding to that bin. However, in most cases, optimal binning (detailed in Section 10.3.3) results in better predictive power.

Once the continuous variables have been binned, they can be treated using the same macros for ordinal variables (i.e, `GiniOrdBDV()`, `EntOrdBDV()`).

Macro `VarCorr()` of Section 14.4, calculates the values of r and r_s with or without binning of the continuous IV. Therefore, we can use it directly to keep as much information in the variable distribution as possible.

We can also use the F-test, as presented in Section 14.3. In this case, we can use the macro `ContGrF()`, which computes both the Gini ratio, without the binning, and the F^*, defined in Equation 14.10, and its p-value.

17.3 Data with Continuous Dependent Variables

17.3.1 Nominal Independent Variables

This case is simply the mirror image of the case of Section 17.2.5. We use the macros `GiniOrdBDV()`, and `EntOrdBDV()`, with binning of the continuous DV, and the macro `PearSpear()`. Note that these macros will work properly when the IV is binary (1/0).

In all cases, the macro `ContGrF()` provides the entropy/Gini ratio and the F^* and its p-value.

17.3.2 Ordinal Independent Variables

In this case, the macro `VarCorr()`, of Section 14.4, provides the correlation coefficients r and r_s and the macro `ContGrF()` provides the entropy/Gini ratio and the F^* and its p-value. The only requirement is to assign the categories of the ordinal variable the appropriate scores.

17.3.3 Continuous Independent Variables

In this final case, the correlation coefficients r and r_s could be calculated using the `VarCorr()` macro. If one of the variables, preferably the IV, can be binned, then the macro `ContGrF()` can be used to calculate the entropy/Gini ratio and the F^* and its p-value. However, without binning, one can also calculate the F^* and its p-value from the original definition (Equation 14.9).

17.4 Variable Reduction Strategies

The last two sections provided several methods for the calculation of measures of the predictive power of potential independent variables. The next task is to eliminate the variables that do not show strong association with the dependent variable.

The challenge in this task arises from the fact that in almost all data mining models, all types of variables are present (i.e., nominal, ordinal, and continuous). It is not possible

to compare the measures used for different variables types. For example, the Gini ratio cannot be compared to the correlation coefficient r. Therefore, it is possible that when comparing two variables, the use of r suggests using one of them, while using G_r suggests using the other.

The general strategy is to attempt to evaluate the variables using more than one criterion to make sure not to miss a possible good predictor. We propose to implement *all* possible evaluation methods for each variable. Variables that appear to be good predictors using different methods should not be removed.

To facilitate the implementation of this strategy, we present the following macro, which wraps the methods used to assess nominal variables with binary dependent variable (see Table 17.4). The macro uses a list of nominal variables to test against a binary DV and outputs all the measures of associations in different datasets.

Table 17.4 Parameters of macro `PowerCatBDV()`.

Header	`PowerCatBDV(DSin, VarList, DV, ChiDS, GiniDS, EntDS);`
Parameter	*Description*
`DSin`	Input dataset
`VarList`	List of nominal independent variables to test
`DV`	Binary dependent variable name
`ChiDS`	Output dataset with the Pearson Chi-squared statistic
`GiniDS`	Output dataset with the Gini ratio
`EntDS`	Output dataset with the Entropy ratio

Step 1

Extract the variables from the input dataset and store them in macro variables.

```
%local i condition;
%let i=1;
%let condition = 0;
%do %until (&condition =1);
   %let Word=%scan(&VarList,&i);
   %if &Word =  %then %let condition =1;
            %else %do;
   %local Var&i;
    %let Var&i=&word;
   %let i = %Eval(&i+1);
      %end;
%end;
```

Step 2

Create the three output datasets to hold the results.

```
proc sql noprint;
 create table &ChiDS
        (VarName char(32), Chi2 num, PChi2 num);
 create table &GiniDS
        (VarName char(32), GiniRatio num);
 create table &EntDS
        (VarName char(32), EntropyRatio num);
quit;
```

Step 3

Loop over the variables and call the macros `PearChi()`, `GiniCatBDV()`, and `EntCatBDV()` to calculate the Chi-squared statistic, the Gini ratio, and the entropy ratio, respectively.

```
%local j Vx;
%do j=1 %to %EVAL(&i-1);
 %let Vx=&&Var&j;
 %let X2=;%let pvalue=;
 %PearChi(&DSin, &Vx, &DV, X2, pvalue);
 %let pvalue=%sysevalf(1-&pvalue);

%let Gr=;
%GiniCatBDV(&DSin, &Vx, &DV, Gr);

%let Er=;
%EntCatBDV(&DSin, &Vx, &DV, Er);
```

Step 4

Store the results in the output datasets.

```
 proc sql noprint;
  insert into &ChiDS values ("&&Var&j", &X2, &pvalue);
  insert into &GiniDS values("&&Var&j", &Gr);
  insert into &EntDS values("&&Var&j", &Er);
 quit;
%end;
```

Step 5

Sort the results datasets using the association measures in descending order.

```
proc sort data=&ChiDS;
by DESCENDING PchI2 Descending Chi2;
run;

proc sort data=&GiniDS;
by DESCENDING GiniRatio;
run;
proc sort data=&EntDS;
```

```
by DESCENDING EntropyRatio;
run;

%mend;
```

Examination of the three output datasets of macro `PowerCatBDV()` should reveal the most persistent predictive variables. To extend macro `PowerCatBDV()` to the cases of other independent variable types, we need to consider the following.

- Modification of step 3 to call the different macros for other IV and DV types
- Binning of continuous variables when needed
- Storing the results in different datasets to accommodate the nature of the measures being calculated

The macros `PowerOrdBDV()` and `PowerContBDV()`, listed in Section A.14, follow these steps to assess the predictive power of ordinal and continuous independent variables with binary DV. An additional macro, `PowerCatNBDV()`, is also provided, which is identical to `PowerCatBDV()`; however, it assumes that the independent categorical variables are all numeric.