J. Dairy Sci. 102:7948–7960
https://doi.org/10.3168/jds.2018-16176

SUPPLEMENTAL MATERIAL

SAS Macro: Model Validation

```
DATA example;
INPUT A$ B$ C$ Y X1 X2 X3 X4 X5 X6;
DATALINES;
A1    B1    C1    23    7     11    20    25    32    39
A1    B1    C2    16    9     14    15    26    31    39
A1    B2    C1    18    9     14    20    29    34    39
A1    B2    C2    23    7     15    17    26    31    39
A1    B3    C1    18    8     14    20    28    35    38
A1    B3    C1    .     8     .     20    28    35    38
A1    B3    C2    22    8     10    15    25    34    36
A1    B4    C1    16    10    13    19    29    33    40
A1    B4    .     16    .     13    19    .     33    40
A1    B4    C2    18    10    12    16    28    32    39
A2    B1    C1    16    8     11    19    30    35    40
A2    B1    C2    16    8     14    15    28    33    37
A2    B2    C1    17    7     14    19    30    33    35
A2    B2    C2    20    9     12    16    29    32    39
A2    B3    C1    28    9     11    20    25    33    38
A2    B3    C2    11    9     10    15    27    33    39
A2    .     C1    21    5     13    15    25    .     36
A2    B4    C1    21    5     13    15    25    35    36
A2    B4    C2    12    5     14    17    30    31    36
A3    B1    C1    17    5     14    20    25    35    35
A3    B1    C2    27    10    15    17    28    32    37
.     B1    .     17    5     14    .     25    35    35
A3    B2    C1    19    9     11    15    27    30    35
A3    B2    C2    29    6     13    15    28    35    35
A3    B3    C1    26    9     11    19    30    31    37
A3    B3    C2    19    10    14    19    27    32    39
A3    B4    C1    19    8     10    17    28    30    35
A3    B4    C2    10    9     11    20    28    34    36
A4    B1    C1    20    6     11    18    30    35    37
A4    B1    C2    25    7     14    20    28    33    35
A4    B2    C1    21    5     13    16    26    35    38
A4    B2    C2    17    9     10    20    27    31    35
A4    B3    C1    12    9     15    20    27    31    38
A4    B3    .     12    .     15    20    27    31    .
A4    B3    C2    10    10    14    15    30    33    35
A4    B4    C1    23    8     12    17    26    32    35
A4    B4    C2    13    9     13    16    30    33    35
A5    B1    C1    29    6     13    17    26    32    39
A5    B1    C2    26    10    14    20    30    33    37
A5    B2    C1    28    5     14    18    30    33    36
A5    B2    C2    28    5     12    19    26    35    40
A5    .     C1    17    7     .     20    27    33    40
A5    B3    C1    17    7     11    20    27    33    40
A5    B3    C2    12    10    12    18    28    32    35
A5    B4    C1    10    5     12    17    27    32    38
A5    B4    C2    20    7     11    19    26    32    35
;
```

J. Dairy Sci. 102:7948–7960
https://doi.org/10.3168/jds.2018-16176


```
ODS HTML CLOSE;
ODS HTML;
TITLE 'HPMixed model analysis';
TITLE2 'Observations that need to be thrown out because of missing
covariates';
PROC HPMIXED DATA = example;
  CLASS A B C; /*enter your classificatory variables*/
  MODEL Y = X1 X2 X3 X4 X5 X6 /*fit the fullest model*/
  /SOLUTION CL;
  RANDOM int /SUBJECT=A;/*enter first set of random effects here (if
necessary)*/
  RANDOM int /SUBJECT=B;/*enter second set of random effects here (if
necessary)*/
  RANDOM int /SUBJECT=C TYPE = un;/*enter third set of random effects here
(if necessary)*/
  OUTPUT OUT=output1 PRED=predm RESID=residm;
RUN;
TITLE; TITLE2; TITLE3;
DATA exampleclean (DROP=predm residm);
  SET output1;
  IF predm eq . THEN DELETE;
  IF Y eq . THEN DELETE; /* Y is the response variable*/
run;

DATA exampleclean;
  SET exampleclean;
  record = _n_;
RUN;

DATA _null;
  SET exampleclean;
  CALL symput('total',_n_)/* Number of records in cleaned-up data*/;
RUN;

***** Create the cross-validation folds for each several random partitions of
the data *****;

%MACRO cvfold(npartition,nfold,n_size);
  %DO partition = 1 %TO &npartition %BY 1;
        DATA temp;
            DO record = 1 TO &total;
            random_num = ranuni(-1);
           OUTPUT;
          END;
        RUN;
        PROC SORT DATA=temp;
              BY random_num;
        RUN;
        DATA partition&partition (DROP=counter random_num);
              SET temp;
              counter = _n_;
              fold&partition = floor(&nfold*(counter-1)/&n_size)+ 1;
        RUN;
        PROC SORT DATA=partition&partition;
              BY record;
        RUN;
```

J. Dairy Sci. 102:7948–7960
https://doi.org/10.3168/jds.2018-16176


```
  %END;
%MACRO append2;
  %DO i = 1 %TO &npartition ;
  partition&i
  %END;
%MEND append2;
DATA finalpartition;
  MERGE %append2;
  BY record;
RUN;
TITLE "Double check on balance of folds within each partition";
PROC FREQ DATA=finalpartition ;
  TABLE fold1-fold&npartition;
RUN;
%MEND cvfold;

***** Macro to conduct the cross-validation *****;

%MACRO cvrandom(npartition,nfold,datasource,model);
PROC DATASETS LIBRARY=work NOLIST;
  DELETE CVsummary&model;
RUN;
%DO partition = 1 %TO &npartition %BY 1;
  %DO fold = 1 %TO &nfold  %BY 1;
  DATA partition&partition.fold&fold;
     MERGE &datasource finalpartition(keep=fold&partition record);
       BY record;
     IF fold&partition =&fold THEN
      DO;
        partition = 'validation';
            Y = .; /* Y is the response variable*/
        END;
       ELSE partition = 'training';
  RUN;
  ODS EXCLUDE ALL;
  PROC HPMIXED DATA= partition&partition.fold&fold;
     ID record partition;
     CLASS A B C; /*enter your classificatory variables*/
     MODEL Y = %fixedeffects; /* Y is the response variable*/
     %randomeffects;
       OUTPUT OUT=p&partition&fold PRED=pred;
  RUN;
  ODS EXCLUDE NONE;
  DATA pcheck&partition&fold;
     UPDATE p&partition&fold &datasource;
     BY record;
  RUN;
  ODS EXCLUDE ALL;
  PROC CORR DATA=pcheck&partition&fold COV OUTP=COV&partition&fold ;
     WHERE partition = 'validation';
     VAR Y Pred; /* Y is the response variable*/
  RUN;
  ODS EXCLUDE NONE;
  ** Save key statistics;
  DATA null;
     SET cov&partition&fold;
```

J. Dairy Sci. 102:7948–7960
https://doi.org/10.3168/jds.2018-16176


```
        **variance for obs;
        IF _TYPE_="COV" and _NAME_ = "Y" THEN CALL symput('var1',Y); /* Y is
the response variable*/
        **variance for pred;
        IF _TYPE_="COV" and _NAME_ = "pred" THEN CALL symput('var2',pred);
        **covariance between obs and pred;
        IF _TYPE_="COV" and _NAME_ = "Y" THEN CALL symput('cov12',pred);
        **mean for obs;
        IF _TYPE_="MEAN"  THEN CALL symput('mean1',Y); /* Y is the response
variable*/
        **mean for pred;
        IF _TYPE_="MEAN"  THEN CALL symput('mean2',pred);
        **correlation between obs and pred;
        IF _TYPE_="CORR" and _NAME_ = "Y" THEN CALL symput('cor12',pred);
  RUN;
  DATA ccc&partition&fold;
     **compute CCC and correlation;
     ccc = 2*&cov12/(&var1+&var2+(&mean1-&mean2)**2);
       corr = &cor12;
       fold = &fold;
       partition = &partition;
  RUN;
  DATA SSEP&partition&fold;
     **compute MSEP and RMSEP and its components;
     SET pcheck&partition&fold ;
       WHERE partition = 'validation';
       msep = (Y-pred)**2;
       odev = (Y-&mean1);
       pdev = (pred-&mean2);
  RUN;
  PROC MEANS DATA=SSEP&partition&fold SUM N NOPRINT;
     VAR msep odev pdev;
       OUTPUT OUT=msep&partition&fold (drop=_type_ _freq_) SUM= N=
/AUTONAME;
  RUN;
  DATA msep&partition&fold (KEEP=msep rmsep ECT ECR ED fold partition);
     SET msep&partition&fold;
       msep = msep_Sum/msep_N;
       rmsep = sqrt(msep);
       s2p = (pdev_N-1)*&var2/pdev_N;
       s2o = (odev_N-1)*&var1/odev_N;
       **compute mean bias as % of RMSEP;
     ECT = (((&mean2-&mean1)**2)*100)/MSEP;
       **compute slope bias % of RMSEP;
       ECR = (((sqrt(s2p) - &cor12*sqrt(s2o))**2)*100)/MSEP;
       **compute random error % of RMSEP;
     ED = (((1-&cor12**2)*s2o)*100)/MSEP;
       fold = &fold;
       partition = &partition;
  RUN;
  DATA slopebias&partition&fold;
     SET pcheck&partition&fold;
       WHERE partition = 'validation';
       residual = Y-pred;
       **compute slope bias for residuals on centered predictions;
       pred_dev = pred-&mean2;
```

J. Dairy Sci. 102:7948–7960
https://doi.org/10.3168/jds.2018-16176


```
  RUN;
  ODS EXCLUDE ALL;
  PROC REG DATA=slopebias&partition&fold  ;
    MODEL residual=pred_dev;
      ODS OUTPUT ParameterEstimates = ParameterEstimates&Partition&Fold;
  RUN;
  ODS EXCLUDE NONE;
  PROC TRANSPOSE DATA=ParameterEstimates&Partition&Fold
OUT=slopebiasestimates&partition&fold ;
  RUN;
  DATA slopebiasestimates&partition&fold (KEEP = fold partition intercept_res
slope_res);
     SET slopebiasestimates&partition&fold;
     IF _NAME_ = "Estimate";
       fold = &fold;
       partition = &partition;
       intercept_res = COL1;
       slope_res = COL2;
  RUN;
  DATA catchall&partition&fold;
     MERGE ccc&partition&fold msep&partition&fold
slopebiasestimates&partition&fold;
       BY partition fold;
  RUN;
  PROC APPEND BASE=CVsummary&model DATA=catchall&partition&fold FORCE;
  RUN;
  PROC DATASETS LIBRARY=work NOLIST;
     DELETE pearson&partition&fold msep&partition&fold cov&partition&fold
partition&partition.fold&fold slopebiasestimates&partition&fold;
  RUN;
  %END;
%END;
  DATA CVsummary&model;
    SET CVsummary&model;
      model = &model;
    fixed = "%fixedeffects";
    random = "%randomeffects";
  RUN;
  ODS results ON;
  TITLE "Final summary statistics for Model &model";
  PROC PRINT DATA=CVsummary&model;
  RUN;
%MEND cvrandom;

***** Validation structure *****;

%LET npartition=2;  /* specify number of partitions */
%LET nfold = 5;     /* specify number of folds per partition */

%cvfold(&npartition,&nfold,&total);

** Model Description;

* Model 1;
%LET model = 1;
  %MACRO fixedeffects;
```

J. Dairy Sci. 102:7948–7960
https://doi.org/10.3168/jds.2018-16176


```
     X1 X2 /* define the fixed effects for Model 1*/
  %MEND fixedeffects;
  %MACRO randomeffects;
     RANDOM int /SUBJECT=A; /* define the random effects for Model 1*/
     RANDOM int /SUBJECT=B; /* define the random effects for Model 1*/
     RANDOM int /SUBJECT=C TYPE=un; /* define the random effects for Model
1*/
  %MEND randomeffects;
  %cvrandom(&npartition,&nfold,exampleclean,&model);

* Model 2;
%LET model = 2;
  %MACRO fixedeffects;
     X3 X4 /* define the fixed effects for Model 2*/
  %MEND fixedeffects;
  %MACRO randomeffects;
     RANDOM int /SUBJECT=A; /* define the random effects for Model 2*/
     RANDOM int /SUBJECT=B; /* define the random effects for Model 2*/
     RANDOM int /SUBJECT=C TYPE=un; /* define the random effects for Model
2*/
  %MEND randomeffects;
  %cvrandom(&npartition,&nfold,exampleclean,&model);

* Model 3;
%LET model = 3;
  %MACRO fixedeffects;
     X5 X6 /* define the fixed effects for Model 3*/
  %MEND fixedeffects;
  %MACRO randomeffects;
     RANDOM int /SUBJECT=A; /* define the random effects for model 3*/
     RANDOM int /SUBJECT=B; /* define the random effects for model 3*/
     RANDOM int /SUBJECT=C TYPE=un; /* define the random effects for model
3*/
  %MEND randomeffects;
  %cvrandom(&npartition,&nfold, exampleclean,&model);

** Models comparisons;

DATA CVsummary;
  SET CVsummary1 CVsummary2 CVsummary3;
  block = compress(partition||fold);
RUN;
%MACRO comparemodels(response);
ODS EXCLUDE ALL;
TITLE "Summary for comparing models for &response";
PROC MIXED DATA=CVsummary;
  CLASS block model;
  MODEL &response = model;
  LSMEANS model /diff;
  RANDOM block;
  ODS OUTPUT lsmeans=lsmeans&response;
  ODS OUTPUT diffs=diffs&response;
RUN;
ODS EXCLUDE NONE;
PROC PRINT DATA=lsmeans&response;
RUN;
```

J. Dairy Sci. 102:7948–7960
https://doi.org/10.3168/jds.2018-16176


```
PROC PRINT DATA=diffs&response;
RUN;
PROC DATASETS LIBRARY=work NOLIST;
  DELETE lsmeans&response diffs&response;
RUN;
%MEND comparemodels;
%comparemodels(corr)
%comparemodels(ccc)
%comparemodels(intercept_res)
%comparemodels(slope_res);
%comparemodels(msep);
%comparemodels(ECT);
%comparemodels(ECR);
%comparemodels(ED);
%comparemodels(rmsep);
QUIT;
```

Instructions on how to use the SAS macro

The provided code is a generic SAS macro for model validation. We intend to provide a starting code for researchers who want to perform model validation in SAS; adaptations in the code are needed to fit the specificity of each database and the researcher's goals. Given that, each researcher is in charge to develop their codes and can use as a starting point the proposed generic SAS code.

To use the generic SAS code for model evaluation, the user will need to enter the following information: observed, fixed, and random variables. In the generic SAS code, the observed variable is "X," the fixed variables are "Y1 to Y6," and the random variables are "A, B, and C." All random variables are classificatory variables. By default, the code allows comparison of 3 models using two partitions and five folds.

In the code, we included several comments to facilitate the adaptation of the generic SAS code to the case of each researcher. Please follow the comments and pay attention if your variables have a different name from the variables used in the example.

J. Dairy Sci. 102:7948–7960
https://doi.org/10.3168/jds.2018-16176

Finally, the output is composed of 4 parts: (1) HPMixed model analysis, (2) Double check on the balance of folds within each partitioning, (3) Final summary statistics, and (4) Summary for comparing models.

(1) HPMixed model analysis: check if the code is reading the database properly.

(2) Double check on the balance of folds within each partitioning; here you can check how the database is partitioned.

(3) Final summary statistics: here is an overview of the fit statistics parameters and the fixed and random effects included in the model. By default, the code allows three models, so in this part of the output will be generated three tables, one for each model. The fit statistic parameters are coded as follows:

ccc: concordance coefficient correlation;

corr: correlation between observed and predicted;

msep: mean square error of prediction;

rmsep: root mean square error of prediction;

ECT: mean bias (as % of MSEP, decomposition of the MSEP);

ECR: slope bias (as % of MSEP, decomposition of the MSEP);

ED: random error (as % of MSEP, decomposition of the MSEP);

intercept_res = mean bias;

J. Dairy Sci. 102:7948–7960
https://doi.org/10.3168/jds.2018-16176

slope_bias= slope bias.

(4) Summary for comparing models: in this part of the report each fit statistic parameter has two tables. The first table is testing if the estimated coefficient is different from zero, and in the second table is the comparison between the three models.