Running Stand-Alone SAS® Viya™ Code Using SAS® Enterprise Miner™

Process Flow Diagram:

Data:
The CS_ACCEPTS data set in the SAMPSIO SAS library is used to create the data source. The target is the binary variable **GB**, which indicates whether a new customer defaulted on a loan during a certain time window, typically one year. The remaining input variables are a set of demographics (age, number of children, region, income, and so on) and a summary of information from the credit bureau or the credit institution (number of loans with the bank, number of credit cards, type of credit product, and so on). The data also contain a frequency variable (_freq_) that counts of occurrence of each observation in each analysis.

Goal:
The goal is to use the SAS Viya Code node first to train a neural network algorithm and score input data, and then to evaluate the trained model by using scores.

Flow:
The SAS Viya Code node (titled “SAS Viya Standalone” in the flow diagram) includes a DATA step to upload the sample data into CAS, then two procedures in SAS® Visual Data Mining and Machine Learning: the NNET procedure to train a neural network model in order to predict the credit (good or bad) of a new customer, and the ASSESS procedure to evaluate the supervised model that is trained by the NNET procedure. The code is as follows:

```
data &em_caslib..cs_accepts2;
    set sampsio.cs_accepts;
run;
```

```
proc nnet data=&em_casLib..cs_accepts2;
      target gb/level=nominal;
      input location resid inc car / level=nominal;
      input age income / level=interval;
      train outmodel=&em_casLib..outmodel;
      architecture MLP;
      hidden 3;
      optimization alg=SGD;
      score out=&em_casLib..nnet_score copyvars=(_all_);
      code file="&em_file_scorecode";
      ods output ModelInfo=&em_user_modelinfo  OptIterHistory=&em_user_iterhist
                  ScoreInfo=&em_user_ScoreInfo;
  run;

  proc assess data=&em_casLib..nnet_score
              maxiters=5000  epsilon=0.0000001  nbins=20  ncuts=10;
      var P_GB0 ;
      target GB / event="0" level=nominal;
      fitstat pvar=P_GB1 /  pevent="1" ;
      ods output ROCInfo=&em_user_roc LIFTInfo=&em_user_lift FitStat=&em_user_fitstat;
  run;
```

The DATA= option in the PROC NNET statement names the input data table to use. Because no Input Data node precedes the SAS Viya Code node, this data table must be named explicitly, not by using a macro variable. The TARGET statement specifies the variable **GB** as a nominal response variable. INPUT statements define the input variables as categorical and continuous variables. The TRAIN statement is used to determine a set of network weights that best predict the targets in the training data. The OUTMODEL= option in the TRAIN statement stores the final model from training to the table represented by **&em_casLib..outmodel**. The ARCHITECTURE statement specifies the architecture of the neural network as a multilayer perceptron (MLP) architecture. The HIDDEN statement specifies three neurons in a hidden layer. The OPTIMIZATION statement requests that the model be trained by the stochastic gradient descent (SGD) algorithm optimization method. The OUT= option in the SCORE statement creates a new data table (**&em_casLib..nnet_score**) to contain the scored results. The CODE statement returns the SAS score code that can be used to score data that are similar to the input data.

The ASSESS procedure produces lift information because the target variable is nominal. The NBINS= option in the PROC ASSESS statement requests that 20 bins be used in the lift calculation, and the NCUTS= option requests that 10 cuts be used in the ROC (receiver operating characteristic) calculation. The MAXITERS= option specifies the maximum number of iterations for the percentile algorithm. The VAR statement requests that the variable **P_GB0** be analyzed in the model assessment. The TARGET statement specifies the variable **GB** as the nominal response variable. The FITSTAT statement requests that the following error metrics be reported for a nominal target: average square error, divisor of average square error, root average square error, mean consequential error, and multiclass log loss.

The %EM_VIYA_ASSESS macro is used to assess the models. The NAME= argument in this macro is used to specify a name for each model. The assessment includes the creation of lift, ROC, and fit statistics tables when applicable. The %EM_REPORT macro specifies the contents of a results window display that is created using a registered data table. The displayed contents (view) can be a data table view or a plot

view. The KEY= argument links the graph to the data set via the key that was registered previously using %EM_REGISTER. The VIEWTYPE= argument specifies the desired type of graph. The DESCRIPTION= argument specifies the text that is to appear in the title bar of the graph pane. The BLOCK= argument labels the block in the **View** menu. The AUTODISPLAY= argument requests that the graph be automatically displayed in the **Results** window.