Demo Steps: Creating a Gradient Boosting Model in SAS Studio

The following Tasks and Utilities generated code is saved on the server. The code can be opened by navigating to **Files (Home)** > **Courses** > **EVMLOPRC** > **SAS_Studio** > **Machine_Learning_Demo_SAS_Studio.sas**.

1. Navigate to SAS Studio by using the orange tool bar at the bottom of the Viya for Learners web page.

2. In the SAS Studio code editor, use a LIBNAME statement to create a caslib named **mycaslib**. Then click the **Libraries** tab and open **mycaslib**. Notice that the promoted table **HMEQ** is present.
```
libname mycaslib cas;
```
3. Use a DATA step to separate the training and validation cases based on the **_PartInd_** variable, where a value of 1 indicates training and a value of 0 indicates validation.
```
data mycaslib.train mycaslib.validate;
        set mycaslib.hmeq;
        if _PartInd_ = 1 then output mycaslib.train;
        else output mycaslib.validate;
run;
```
4. On the Tasks and Utilities tab, select **SAS Viya Supervised Learning** and open **Gradient Boosting**.

5. On the Data tab, enter **mycaslib.train** in the data field. Under the **Roles** field, select **Use a nominal target**. Add the nominal target **BAD** to the **target** field. Then add the appropriate imputed variables to the interval (**IMP_CLAGE**, **IMP_CLNO**, **IMP_DEBTINC**, **IMP_DELINQ**, **IMP_DEROG**, **IMP_LOAN**, **IMP_MORTDUE**, **IMP_NINQ**, **IMP_VALUE**, **IMP_YOJ**) and nominal (**IMP_JOB**, **IMP_REASON**) fields. Notice that the GRADBOOST procedure syntax is populated in the code window.

6. On the Output tab, select the **Save scoring model** check box and then select **All variables**. Specify the CAS table **mycaslib.gbt_model** and run the generated code.
```
proc gradboost data=MYCASLIB.TRAIN;
        target BAD / level=nominal;
        input IMP_CLAGE IMP_CLNO IMP_DEBTINC IMP_DELINQ IMP_DEROG
            IMP_LOAN IMP_MORTDUE IMP_NINQ IMP_VALUE
            IMP_YOJ / level=interval;
        input IMP_JOB IMP_REASON / level=nominal;
        savestate rstore=MYCASLIB.gbt_model;
        id _all_;
run;
```
7. On the Tasks and Utilities tab, select **SAS Viya Evaluate and Implement** and click **Scoring** to open it.

8. On the Data tab, enter **mycaslib.validate** in the **DATA** field. Then select **Use saved scoring model** in the **SCORING TYPE** field and specify the CAS table **mycaslib.GBT_MODEL**. Finally, in the **OUTPUT DATA** field, specify a new CAS table, **mycaslib.gbt_scored**, to save the scoring information. Run the generated code.
```
proc astore;
        score data=MYCASLIB.VALIDATE out=mycaslib.gbt_scored
           rstore=MYCASLIB.GBT_MODEL;
run;

proc contents data=mycaslib.gbt_scored;
run;
```
9. On the Tasks and Utilities tab, select **SAS Viya Evaluate and Implement** and click **Assess** to open it.

10. On the Data tab, specify the scored CAS table. In the **ROLES** field, select **Use a nominal targe**t and add the variable **BAD** in the **Target** field. Select **1** in the **Event level of target** field and add **P_BAD1** in the **Posterior probability of target event** field.

11. On the Options tab, clear the **Produce fit statistics** and **Lift chart** check boxes. Run the generated code.
```
proc assess data=MYCASLIB.GBT_SCORED nbins=10 ncuts=10;
        target BAD / event="1" level=nominal;
        input P_BAD1;
        ods output ROCInfo=WORK._roc_temp;
run;

data _null_;
        set WORK._roc_temp(obs=1);
        call symput('AUC', round(C, 0.01));
run;

proc sgplot data=WORK._roc_temp noautolegend aspect=1;
        title 'ROC Curve (Target = BAD, Event = 1)';
        xaxis label='False positive rate' values=(0 to 1 by 0.1);
        yaxis label='True positive rate' values=(0 to 1 by 0.1);
        lineparm x=0 y=0 slope=1 /
```

```
            transparency=.7 LINEATTRS=(Pattern=34);
        series x=fpr y=sensitivity;
        inset "AUC=&AUC"/position=bottomright border;
run;

proc delete data=WORK._roc_temp;
run;
```