# Solution: Using Tasks to Generate a Neural Network in SAS Studio from a Promoted Table

The following Tasks and Utilities generated code is saved on the server. To open the code, navigate to **Files (Home)** > **Courses** > **EVMLOPRC** > **SAS_Studio** > **Machine_Learning_Practice_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. Define a home directory path macro variable.

```
%let homedir=%sysget(HOME);
```

3. In the code editor, create a caslib named **mycaslib** and use the LIBNAME statement below. Then click the **Libraries** tab and open **mycaslib**. Notice that the promoted table **BANK** is present.

```
libname mycaslib cas;
```

4. Use the following 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.bank;
        if _PartInd_ = 1 then output mycaslib.train;
        else output mycaslib.validate;
run;
```

5. On the Tasks and Utilities tab, select **SAS Viya Supervises Learning** and open **Neural Network**.

6. On the Data tab, enter **mycaslib.train** in the data field. Under the **Roles** field, select **Use a nominal target**. Add the nominal target **B_TGT** to the **target** field.
   Then add the appropriate imputed variables to the **interval** field:
   - **IMP_demog_age**
   - **IMP_demog_homeval**
   - **IMP_demog_inc**
   - **IMP_rfm5**
   - **IMP_rfm6**
   - **IMP_rfm7**
   - **IMP_rfm8**
   - **IMP_rfm9**
   - **IMP_rfm10**
   - **IMP_rfm11**
   - **IMP_rfm12**

   and to the **nominal** field:
   - **IMP_cat_input1**
   - **IMP_cat_input2**
   - **IMP_demog_gen**
   - **IMP_demog_hos**

   Notice that the NNET procedure syntax is populated in the code window.

7. On the Options tab, for **Hidden Layer**s, change the number of hidden units to **100**.

8. On the Output tab, select **Save scoring code** and change the file to **nn_model.sas**. Run the generated code.

```
libname _tmpcas_ cas;

proc nnet data=MYCASLIB.BANK;
        target b_tgt / level=nominal;
        input IMP_demog_age IMP_demog_homeval IMP_demog_inc
           IMP_rfm10 IMP_rfm11 IMP_rfm12 IMP_rfm5 IMP_rfm6
           IMP_rfm7 IMP_rfm8 IMP_rfm9 / level=interval;
        input IMP_cat_input1 IMP_cat_input2 IMP_demog_gen
           IMP_demog_hos / level=nominal;
        hidden 100;
        train outmodel=_tmpcas_._Nnet_model_;
        optimization regL2=0.1;
        code file="&homedir./nn_model.sas";
run;
```

```
proc delete data=_tmpcas_._Nnet_model_;
run;

libname _tmpcas_;
```

9. On the Tasks and Utilities tab, select **SAS Viya Evaluate and Implement** and open **Scoring**.

10. On the Data tab, specify **mycaslib.validate** as the data table. Then select **Use scoring code** in the **Scoring Type** field. Change the file to **nn_model.sas**. Finally, in the **Output Data** field, specify a new CAS table, **mycaslib.nn_scored**, to save the scoring information. Run the generated code.

```
data mycaslib.nn_scored;
        set MYCASLIB.VALIDATE;
        %include "&homedir./nn_model.sas";
run;

proc contents data=mycaslib.nn_scored;
run;
```

11. On the Tasks and Utilities tab, select **SAS Viya Evaluate and Implement** and click **Assess** to open it.

12. On the DATA tab, specify the CAS table as **mycaslib.nn_scored**. Select **Use a nominal target** under **Roles** and add the target **b_tgt** to the **target** field. Change the **event level of target value** to **1** and the **Posterior probability of target event** value to **P_b_tgt1**.

13. On the OPTIONS tab, clear the **Produce fit statistics** and **Lift chart** check boxes. Run the generated code.

```
proc assess data=MYCASLIB.NN_SCORED nbins=10 ncuts=10;
        target b_tgt / event="1" level=nominal;
        input P_b_tgt1;
        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 = b_tgt, 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;
```

14. To drop a global table from the server in SAS Studio, use the following CASUTIL procedure:

```
proc casutil;
   droptable casdata="bank";
run;
```