Enterprise Miner Open Source Integration and R

What is R?

R is an open source programming language for statistical computing and machine learning. Being open source means that the source code is freely available and distributed. It is used a lot in academia and also in data science. There are over 10,000 packages available to perform specific tasks.

Want more help with a particular R problem?

Stack Overflow is a great place to ask specific questions (most of the time someone has already asked your question).
https://stackoverflow.com/

R-Bloggers is a great site with 'how to' tutorials.
https://www.r-bloggers.com/

How do you get R working in Enterprise Miner?

Installation and set up

Your administrator needs to install R (and the pmml package) on the server, tell Enterprise Miner where R is located and enable R to run in Enterprise Miner.
Patrick Hall has a good blog that takes you through the installation and common errors
https://communities.sas.com/t5/SAS-Communities-Library/The-Open-Source-Integration-node-installation-cheat-sheet/ta-p/223470

Confirming that R code will run in Enterprise Miner
You can check if SAS will execute R code by running the following command in a SAS code node:

```
proc options option=rlang; run;
```

Using the Open Source Integration node to run R scripts

The open source integration node will allow you to create and score R models via Enterprise Miner. You can find it under the Utilities tab. It will handle the data and results transfer for you.

What about Python?
Another programming languages used by data scientists is Python

You can run Python scripts in Enterprise Miner. Check out the SAS video https://youtu.be/GROwni8nw64

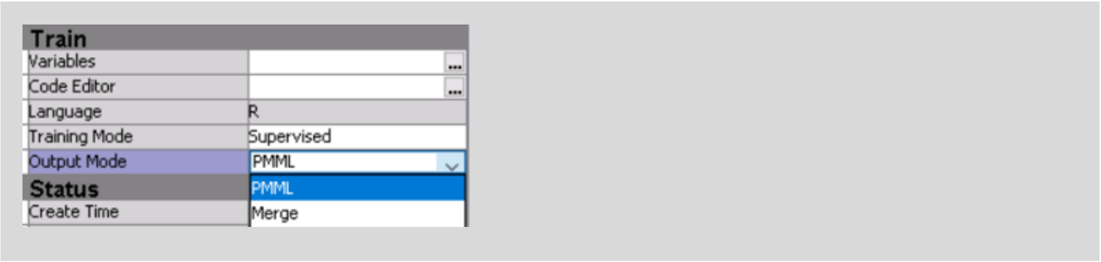
SAS also has a training course called **SAS Open Source Series: SAS and R and Python** (OSSSASR) that will give you step by step instructions for doing this in Enterprise Miner.

The key options to know

Asides from the Code Editor ellipsis, the two main options that are likely to require changes are the **Output Mode** in the **Open Source Integration** node and the **Mapping Editor** in the **Model Import** node.

Output Mode

The two main choices you are interested in here are **PMML** and **Merge**.



Train	
Variables	...
Code Editor	...
Language	R
Training Mode	Supervised
Output Mode	PMML
Status	PMML
Create Time	Merge

Most of the work is done for you with PMML. SAS score code is produced. But PMML only works with the following R packages:

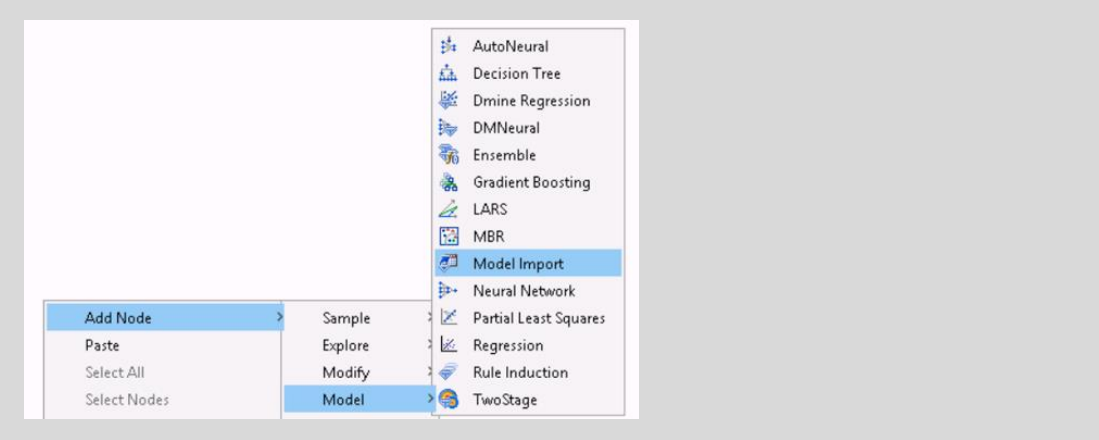
- base (linear models, generalized linear models, kmeans)
- rpart (decision trees)
- nnet (neural networks and multinomial log-linear models)

For a more complicated model like **xgboost** you will need to set the output mode to **Merge** mode.

With merge mode

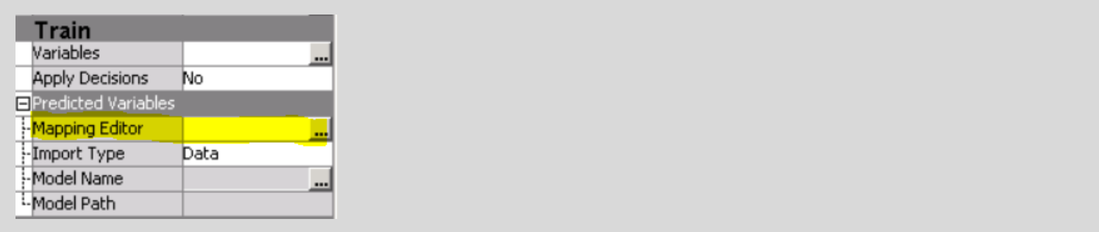
- There is no SAS score code. Make use of R's **predict()** function
- The **Model Import** node is required to compare the R model to the other models created in Enterprise Miner

The **Model Import** node can be found under the **Model** tab.



The Mapping Editor

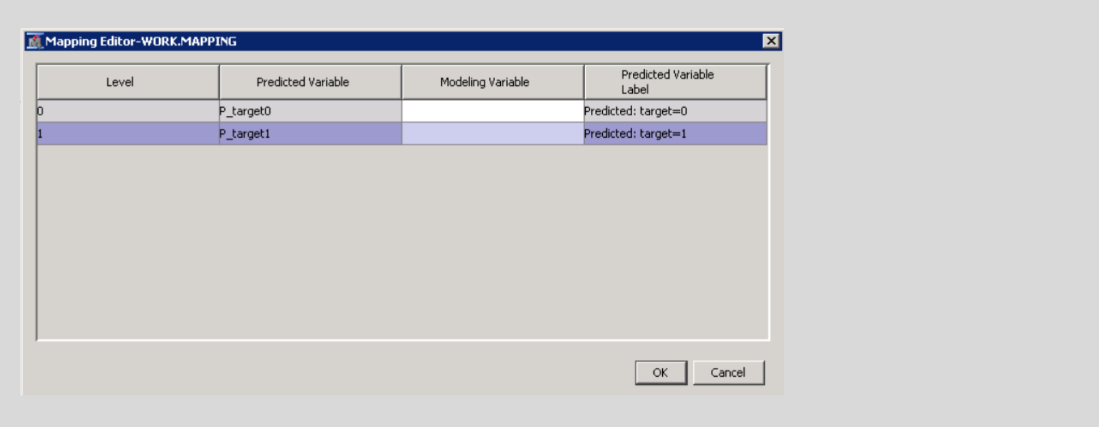
The main option to change in the **Model Import** node is the **Mapping Editor**. It maps the R target variables to the Enterprise Miner target variables. This allows you to do model comparisons. You can bring up the mapping editor by clicking on the ellipsis.



Train

Variables	...
Apply Decisions	No
⊟ Predicted Variables	
Mapping Editor	...
Import Type	Data
Model Name	...
Model Path	

You need to tell Enterprise Miner which variable represents the primary target and which one represents the secondary target. EMR_VAR1 represents level 0 and EMR_VAR2 represents level 1.



Mapping Editor-WORK.MAPPING

Level	Predicted Variable	Modeling Variable	Predicted Variable Label
0	P_target0		Predicted: target=0
1	P_target1		Predicted: target=1

The macro variables

Macro variables are great to use when passing data and results between SAS and R

- If you change your target or your input variables you don't need to rewrite the R code
- R is case sensitive. The macro variables will deal with this
- The target variable must be preceded by the letter r
- Some of the macro variables must be defined so that the score code can be generated

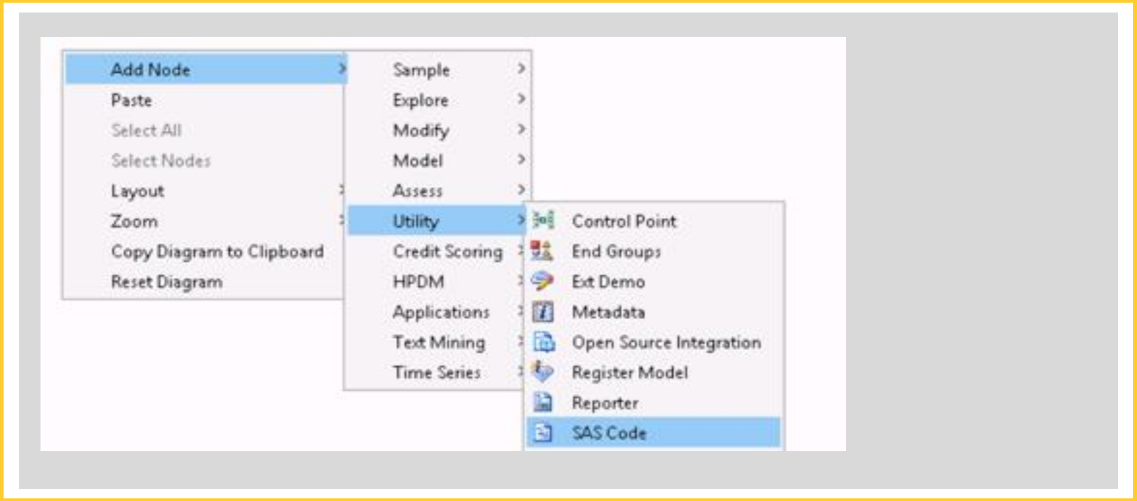
The following macro variables and their definitions are specified below

Macro variable	Definition
&EMR_MODEL	This is the name of the model object in R
&EMR_IMPORT_DATA	Name of the dataset to be used in R to build the model (usually it is the training dataset)
&EMR_CLASS_TARGET	Name of the target variable (must be categorical)
&EMR_NUM_TARGET	Name of the target variable (must be numeric)
&EMR_CLASS_INPUT	Categorical variables used as inputs to the predictive model
&EMR_NUM_INPUT	Numeric/continuous variables used as inputs to the predictive model
&EMR_EXPORT_TRAIN	Name of the dataset containing the scored training data to be exported from the Open Source Integration node
&EMR_EXPORT_VALIDATE	Name of the dataset containing the scored validation data to be exported from the Open Source Integration node
&EMR_IMPORT_VALIDATE	Name of the validation dataset to be used in R for predictions

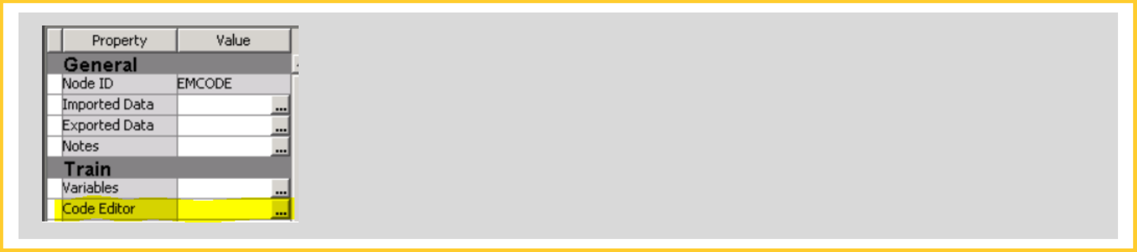
Exercises

Ex-1 Checking that R is enabled on Enterprise Miner

1. Start Enterprise Miner and open the project **How-Open-Source-Integration**.
2. Open the diagram called **Ex-1**.



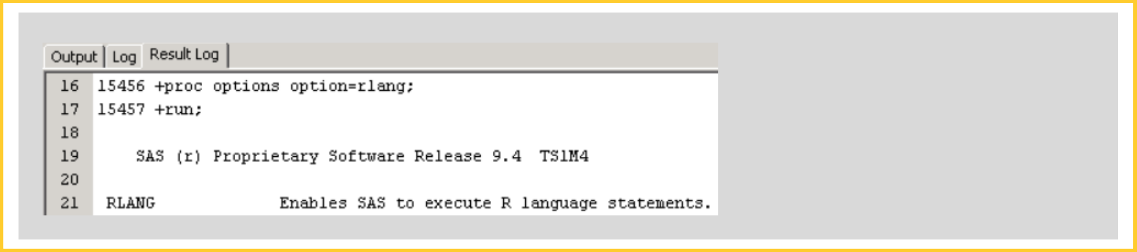
3. Right click on the white space and select: **Add Node > Utility > SAS Code**.
4. Click the code editor ellipsis:



Property	Value
General	
Node ID	EMCODE
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Code Editor	...

Type the following code into the editor:
proc options option=rlang; run;

5. Click on the result log tab in the training code window and confirm you have the following message:



```
15456 +proc options option=rlang;
15457 +run;

    SAS (r) Proprietary Software Release 9.4  TS1M4

 RLANG             Enables SAS to execute R language statements.
```

Note: if you encounter the message below it means that SAS cannot execute R scripts and your administrator would need to enable this.

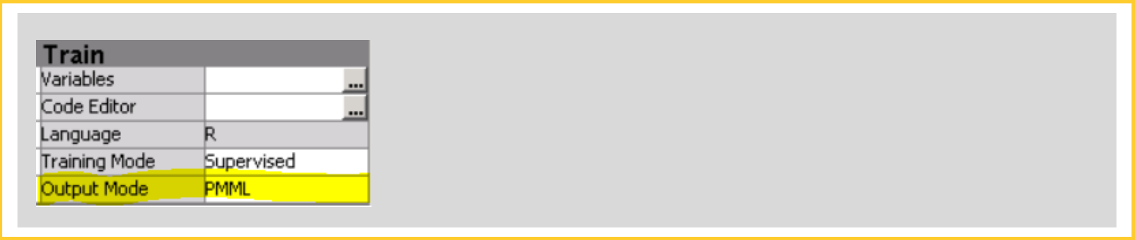
```
ERROR: The RLANG system option must be specified in the SAS configuration
file or on the SAS invocation command line to enable the submission of R
language statements.
```

Another note you may encounter in the log also indicates that R needs to be enabled

```
NORLANG              Disables SAS from executing R language statements
```

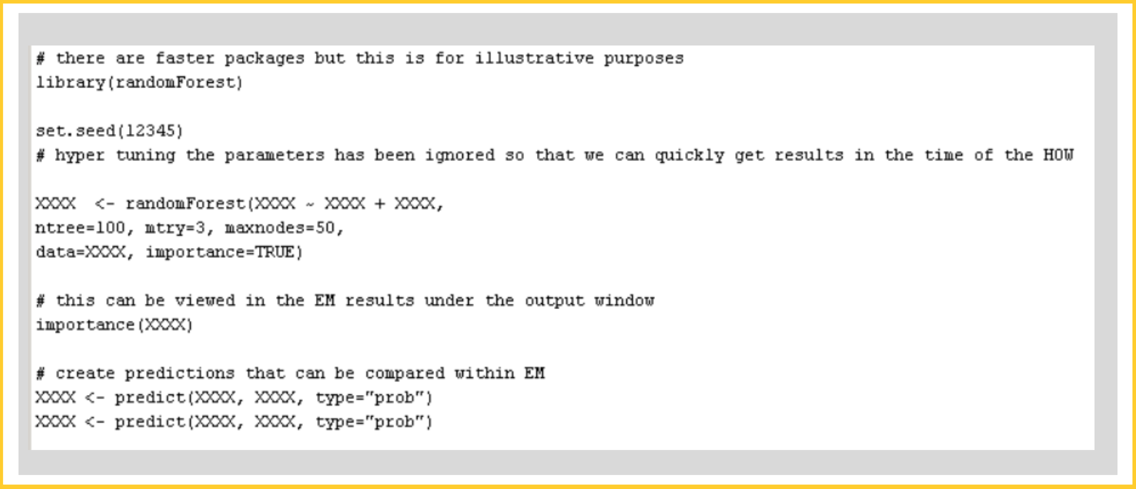
Ex-2 Running a predictive model in R via the Open Source Integration node

1. If you haven't already, Start Enterprise Miner and open the project called **HOW-Open-Source-Integration**.
2. Open the diagram **Ex-2-3**.
   Click on the Open Source Integration node that has been renamed to **R randomForest**. Change the output mode to **Merge**.



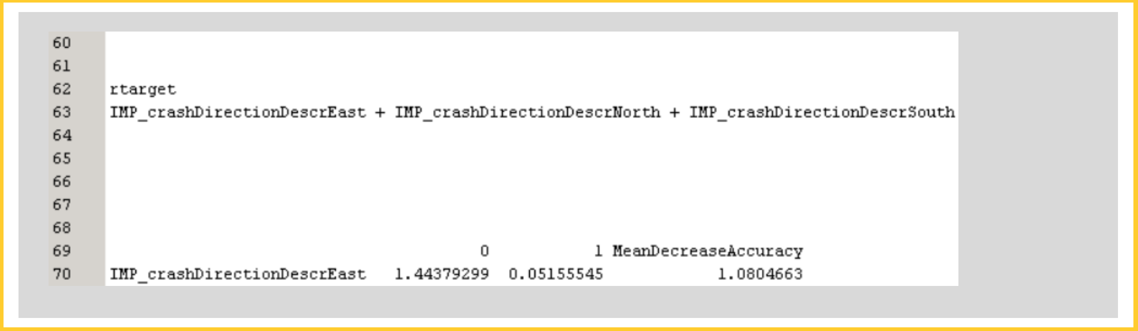
Train	
Variables	...
Code Editor	...
Language	R
Training Mode	Supervised
Output Mode	PMML

3. Click on the **Code Editor** ellipsis You will see the following code below.



```
# there are faster packages but this is for illustrative purposes
library(randomForest)

set.seed(12345)
# hyper tuning the parameters has been ignored so that we can quickly get results in the time of the HOW

XXXX  <- randomForest(XXXX ~ XXXX + XXXX,
ntree=100, mtry=3, maxnodes=50,
data=XXXX, importance=TRUE)

# this can be viewed in the EM results under the output window
importance(XXXX)

# create predictions that can be compared within EM
XXXX <- predict(XXXX, XXXX, type="prob")
XXXX <- predict(XXXX, XXXX, type="prob")
```

4. Replace all the XXXX with the appropriate macro variables. The arguments for randomForest that need macro variables are the target variable, categorical inputs, numeric inputs and the dataset name. The arguments for the predict function are the model followed by a dataset to score. Hint: there is a helper text file called **Ex-2-3-Helper.txt** if you are having trouble.
5. Click OK when you are finished.
6. Run the node by right clicking on it and selecting **Run**.
7. When it has finished view the **Results**.
8. Expand the **Output** window and go to line 60. Notice how the macro variables have resolved to the actual variables.



```
60
61
62   rtarget
63   IMP_crashDirectionDescrEast + IMP_crashDirectionDescrNorth + IMP_crashDirectionDescrSouth
64
65
66
67
68
69                                            0           1 MeanDecreaseAccuracy
70   IMP_crashDirectionDescrEast   1.44379299  0.05155545            1.0804663
```

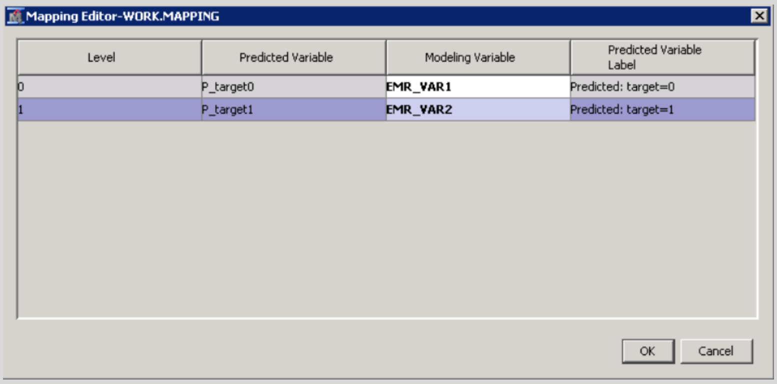
9. Close the results window

Ex-3 Bonus exercise: Model Comparison

Next, we need to make sure that this model can be used for model comparison in Enterprise Miner.

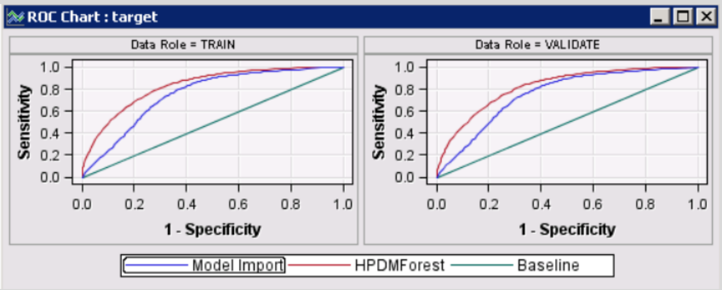
1. If you haven't already, Start Enterprise Miner and open the project called **HOW-Open-Source-Integration** and open the diagram **Ex-2-3**.

2. Click on the **Model Import** node. Then click on the Mapping Editor ellipsis. Specify the modelling variables as follows:



Level	Predicted Variable	Modeling Variable	Predicted Variable Label
0	P_target0	EMR_VAR1	Predicted: target=0
1	P_target1	EMR_VAR2	Predicted: target=1

3. Click OK.
4. Right click on the **Model Comparison** node and select Run.
5. When it finishes running view the results. Notice how the different pre-processing as well as the different implementations have resulted in different models.



Tips and Tricks

Watch out for missing values

R and SAS handle missing values differently

- R will either ignore them or encounter an error
- Some SAS models like trees can use records with missing values

Imputation will deal with missing values and ensure all records get used in R. Imputation can be done in SAS or R. For more info on imputation refer to the blog: How to Handle Missing Data by Alvira Swalin https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4

Write your scripts in an IDE like R Studio

Use an Integrated Development Environment (IDE) like R studio. They are designed to speed up the development process by autocompleting variable names, highlighting syntax for easy error spotting and so on.

You are less likely to make mistakes compared to writing directly into the Code Editor.

User a helper file for your macro variables

Having all the macro variables that you need saved in a helper file and the default ones prepopulated will save you a few typos and a few errors.

Get the admin team to help with package installation

R needs to be on the SAS server. It is also a good idea to have the administrator install all the relevant packages ahead of time. When you install packages in R you can get a prompt asking what distribution site (i.e. what mirror) you would like to download the package from. A dialog box will not pop up in Enterprise Miner. You will probably get an error if you do not explicitly specify the mirror.

References

Hall, P. (2014). *The Open Source Integration node installation cheat sheet*. Retrieved 25th January 2019 from https://communities.sas.com/t5/SAS-Communities-Library/The-Open-Source-Integration-node-installation-cheat-sheet/ta-p/223470

Swalin, A. (2018). *How to handle missing data*. Retrieved 8th February 2019 from https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4

Zhang, R. (2015). *How to execute a Python script in SAS Enterprise Miner.* Retrieved 25th February from https://youtu.be/GROwni8nw64