

MLOps for Pirates: R-SASCTL

Sophia Rowland

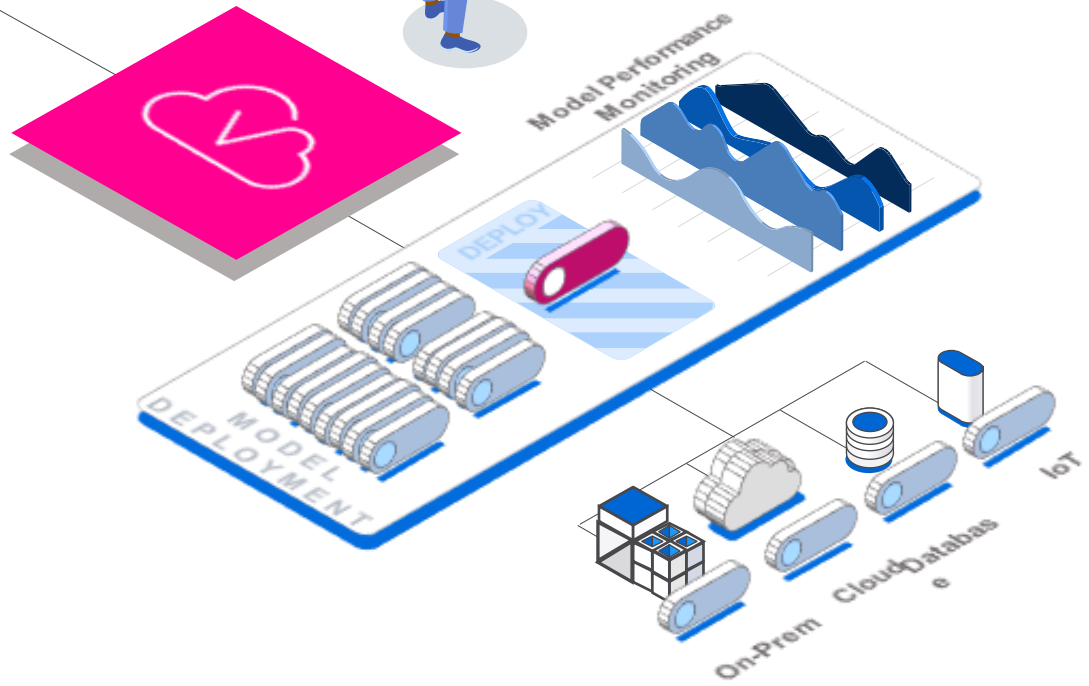
Sr. Product Manager, ModelOps & MLOps

SAS CTL

Bridging the Gap
From Open Modeling
To Enterprise AI & Analytics



MLOps Engineers
Manage SAS and
open source models



The Anatomy of a Model

The screenshot displays the SAS Model Manager interface for a model named "Decision Tree (R) (1.0)". The main window shows the score code for the model, which is a R script file named "hmeq_classtree_r_score.r". The code is as follows:

```
1 library(rpart)
2
3 rScoreFunction <- function(JOB, REASON, CLAGE, CLNO, DEBTINC, DELINQ, DEROG, NINQ, YOJ)
4 {
5   #output: EM_EVENTPROBABILITY, EM_CLASSIFICATION
6
7   if(!exists("myClassTree"))
8   {
9     assign("myClassTree", readRDS(file = paste(rdsPath, 'hmeq_classtree_r.rds', sep = "")), envir = .GlobalEnv)
10  }
11
12  # Threshold for the misclassification error
13  threshPredProb <- 0.199916001679966
14
15  # Recode JOB into a numeric factor
16  JOBtype <- 0
17  JOBtype[JOB == 'Mgr'] <- 1
18  JOBtype[JOB == 'Office'] <- 2
19  JOBtype[JOB == 'Other'] <- 3
20  JOBtype[JOB == 'ProfExe'] <- 4
21  JOBtype[JOB == 'Sales'] <- 5
22  JOBtype[JOB == 'Self'] <- 6
23
24  # Recode REASON into a numeric factor
25  REASONtype <- 0
26  REASONtype[REASON == 'DebtCon'] <- 1
27  REASONtype[REASON == 'HomImp'] <- 2
28
29  # Impute missing covariates by their means
30  CLAGE[is.na(CLAGE) | is.null(CLAGE)] <- 178.6067620442
31  CLNO[is.na(CLNO) | is.null(CLNO)] <- 21.21746724891
32  DEBTINC[is.na(DEBTINC) | is.null(DEBTINC)] <- 33.47332952233
33
34  # Impute missing covariates by their modes
35  DELINQ[is.na(DELINQ) | is.null(DELINQ)] <- 0.0
36  DEROG[is.na(DEROG) | is.null(DEROG)] <- 0.0
37  NINO[is.na(NINO) | is.null(NINO)] <- 0.0
```

The interface also shows a sidebar with various categories: SCORE CODE, SCORE RESOURCES, VARIABLES, PROPERTIES AND METADATA, and OTHER. The right-hand panel displays the Model Summary, including details like Created by, Modified by, Date modified, Displayed version, Latest published version, Score Code Type (R), Algorithm (Decision tree), Development Tool (R), Function (Classification), and Training Code Type (R).

The Anatomy of a Model

Score code & score resources enable scoring and publishing

SCORE CODE

- hmeq_classtree_r_score.r

SCORE RESOURCES

- DS2EP_PythonWrapper.sas
- hmeq_classtree_r_score.py
- hmeq_classtree_r.rds
- rPy2Wrapper.py

VARIABLES

- inputVar.json
- outputVar.json

PROPERTIES AND METADATA

- fileMetadata.json
- ModelProperties.json

OTHER

- dmcas_fitstat.json
- dmcas_lift.json
- dmcas_roc.json
- hmeq_classtree_train.r

```
1 library(rpart)
2
3 rScoreFunction <- function(JOB, REASON, CLAGE, CLNO, DEBTINC, DELINQ, DEROG, NINQ, YOJ)
4 {
5   #output: EM_EVENTPROBABILITY, EM_CLASSIFICATION
6
7   if (!exists("myClassTree"))
8   {
9     assign("myClassTree", readRDS(file = paste(rdsPath, 'hmeq_classtree_r.rds', sep = "")), envir = .GlobalEnv)
10  }
11
12  # Threshold for the misclassification error
13  threshPredProb <- 0.199916001679966
14
15  # Recode JOB into a numeric factor
16  JOBtype <- 0
17  JOBtype[JOB == 'Mgr'] <- 1
18  JOBtype[JOB == 'Office'] <- 2
19  JOBtype[JOB == 'Other'] <- 3
20  JOBtype[JOB == 'ProfExe'] <- 4
21  JOBtype[JOB == 'Sales'] <- 5
22  JOBtype[JOB == 'Self'] <- 6
23
24  # Recode REASON into a numeric factor
25  REASONtype <- 0
26  REASONtype[REASON == 'DebtCon'] <- 1
27  REASONtype[REASON == 'HomImp'] <- 2
28
29  # Impute missing covariates by their means
30  CLAGE[is.na(CLAGE) | is.null(CLAGE)] <- 178.6067620442
31  CLNO[is.na(CLNO) | is.null(CLNO)] <- 21.21746724891
32  DEBTINC[is.na(DEBTINC) | is.null(DEBTINC)] <- 33.47332952233
33
34  # Impute missing covariates by their modes
35  DELINQ[is.na(DELINQ) | is.null(DELINQ)] <- 0.0
36  DEROG[is.na(DEROG) | is.null(DEROG)] <- 0.0
37  NINO[is.na(NINO) | is.null(NINO)] <- 0.0
```

Model Summary

Created by: sorowl
Modified by: sorowl
Date modified: Aug 2, 2023 10:13 AM
Displayed version: 1.0
Latest published version: Not published

Score Code Type
R

Algorithm
Decision tree

Development Tool
R

Function
Classification

Training Code Type
R

The Anatomy of a Model

The screenshot displays the SAS Model Manager interface for a project named "Home Loan Default_Version 1 (1.0)". The main window shows the "Decision Tree (R) (1.0)" model configuration. A table in the top left lists input and output variables:

Name	Data Type	Input	Output	Length	Measurement	Description
DEROG	Decimal	✓		8	Interval	
NINQ	Decimal	✓		8	Interval	
JOB	Character	✓		7	Nominal	
REASON	Character	✓		7	Nominal	
CLAGE	Decimal	✓		8	Interval	
CLNO	Decimal	✓		8	Interval	
DEBTINC	Decimal	✓		8	Interval	
DELINQ	Decimal	✓		8	Interval	
YOJ	Decimal	✓		8	Interval	
EM_EVENTPROBABILITY	Decimal		✓	8	Interval	
EM_CLASSIFICATION	Character		✓	1	Nominal	

The central pane shows the R code for the model, titled "hmeq_classtree_r_score.r". The code includes library imports, function definitions, and data processing steps. A callout box highlights the "VARIABLES" section of the code, which defines input and output variables:

```
VARIABLES
inputVar=json
outputVar=json
```

The right-hand pane displays the "Model Summary" for the "Decision Tree" model, showing details such as the creator (sorowl), modification date (Aug 2, 2023 10:13 AM), and the score code type (R).

Input and output variables let us know what our models needs and generates

The Anatomy of a Model

The screenshot displays the SAS Model Manager interface for a model named "Decision Tree (R) (1.0)". The interface is divided into several sections:

- General Properties:** Located on the left, it shows details such as Name, Description, Created by, Date created, Date modified, Location, Project name, Project version, Function (Classification), Score code type (R), Training table, Training code type (R), and Algorithm (Decision tree).
- Model Summary:** Located on the right, it provides a high-level overview including Created by, Modified by, Date modified, Displayed version, Latest published version, Score Code Type (R), Algorithm (Decision tree), Development Tool (R), Function (Classification), and Training Code Type (R).
- Code Editor:** The central pane shows the R script for the model, titled "hmeq_classtree_r_score.r". The code includes library loading, function definition, variable recoding, and imputation steps.
- Resources and Files:** A sidebar on the left lists various files and resources associated with the model, including "fileMetadata.json" and "ModelProperties.json" which are highlighted in a blue box.

Properties help tie a model to a modeler and a file to a role

The Anatomy of a Model

The screenshot displays the SAS Model Manager interface for a model named "Decision Tree (R) (1.0)". The main area shows the R score code for the model, which includes library loading, function definition, and variable recoding. The right-hand panel provides a "Model Summary" with details such as the creator (sorowl), modification date (Aug 2, 2023 10:13 AM), and the algorithm used (Decision tree). Below the summary, the "Score Code Type" is identified as R, and the "Function" is listed as Classification. The "Development Tool" is also noted as R. The "Training Code Type" is also R. In the bottom right corner of the interface, the "Model Properties" section lists several JSON files: dmcas_fitstat.json, dmcas_lift.json, dmcas_roc.json, and hmeq_classtree_train.json. A blue callout box highlights these files, with a line pointing to the text "Diagnostic metrics help us compare models side-by-side".

```
1 library(rpart)
2
3 rScoreFunction <- function(JOB, REASON, CLAGE, CLNO, DEBTINC, DELINQ, DEROG, NINQ, YOJ)
4 {
5   #output: EM_EVENTPROBABILITY, EM_CLASSIFICATION
6
7   if (!exists("myClassTree"))
8   {
9     assign("myClassTree", readRDS(file = paste(rdsPath, 'hmeq_classtree_r.rds', sep = "")), envir = .GlobalEnv)
10  }
11
12  # Threshold for the misclassification error
13  threshPredProb <- 0.199916001679966
14
15  # Recode JOB into a numeric factor
16  JOBtype <- 0
17  JOBtype[JOB == 'Mgr'] <- 1
18  JOBtype[JOB == 'Office'] <- 2
19  JOBtype[JOB == 'Other'] <- 3
20  JOBtype[JOB == 'ProfExe'] <- 4
21  JOBtype[JOB == 'Sales'] <- 5
22  JOBtype[JOB == 'Self'] <- 6
23
24  # Recode REASON into a numeric factor
25  REASONtype <- 0
26  REASONtype[REASON == 'DebtCon'] <- 1
27  REASONtype[REASON == 'HomImp'] <- 2
28
29  # Impute missing covariates by their means
30  CLAGE[is.na(CLAGE) || is.null(CLAGE)] <- 178.6067620442
31  CLNO[is.na(CLNO) || is.null(CLNO)] <- 21.21746724891
32  DEBTINC[is.na(DEBTINC) || is.null(DEBTINC)] <- 33.47332952233
33
34  # Impute missing covariates by their modes
35  DELINQ[is.na(DELINQ) || is.null(DELINQ)] <- 0.0
36  DEROG[is.na(DEROG) || is.null(DEROG)] <- 0.0
37  NINO[is.na(NINO) || is.null(NINO)] <- 0.0
```

Diagnostic metrics help us compare models side-by-side

Demonstration

1

Get Started

- Install packages and dependencies
- Start session
- Import data

2

Prepare Data and Build Model

- Create training/testing/validation splits
- Train model

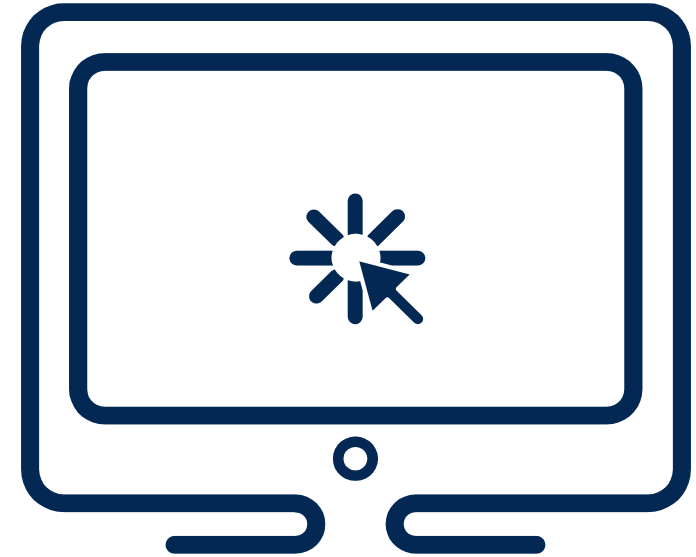
3

Register Model

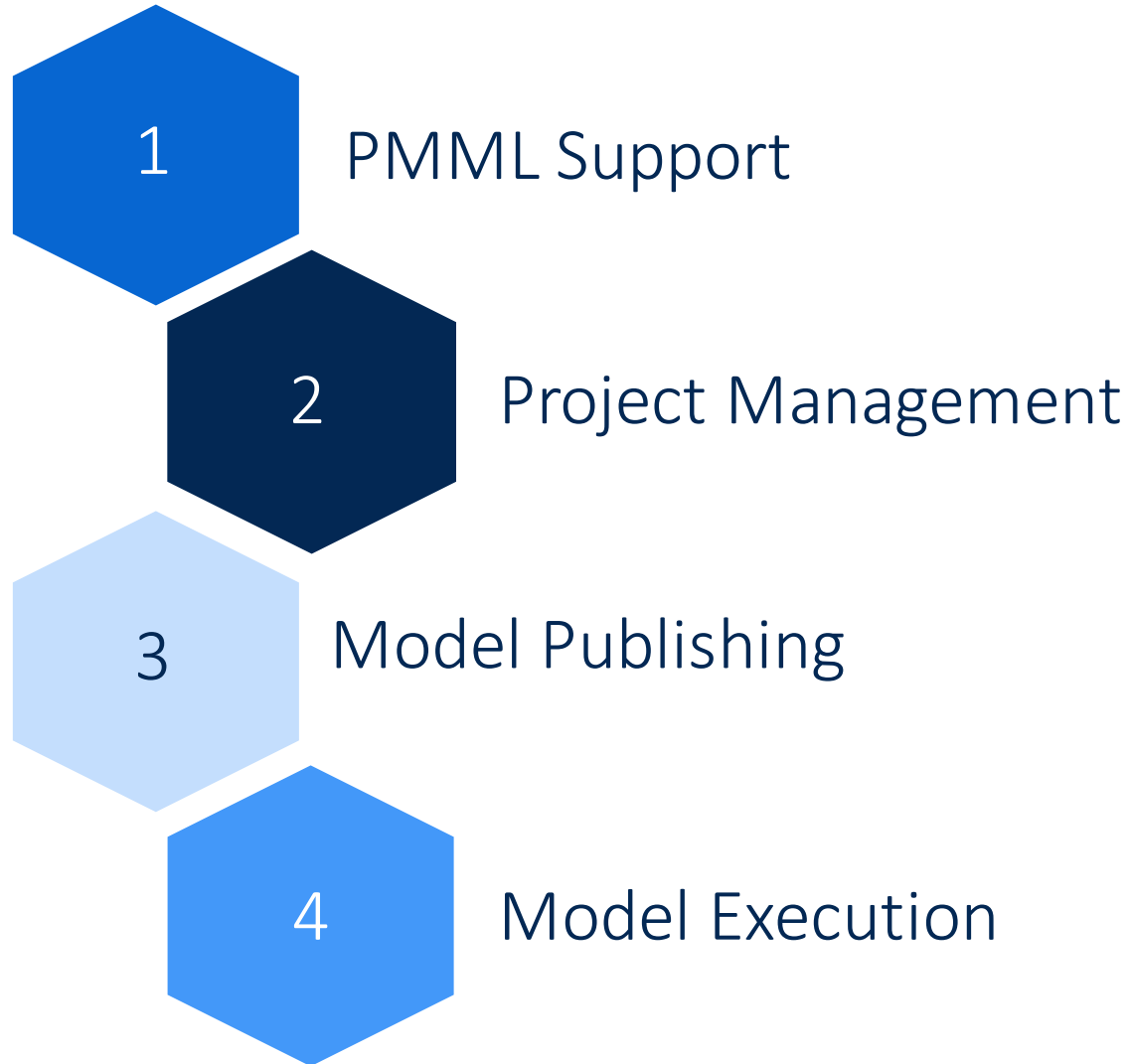
- Save model
- Generate score code
- Generate metadata (inputs, outputs, accuracy, properties).

4

Leverage Model inside SAS Model Manager



Beyond our Demonstration



Want to Learn More?

Check out the ModelOps booth and these free resources

R-sasctl GitHub



Sasctl Article



Thank you!

What remaining questions do you have?

eduardo.hellas@sas.com

