Python, R und SAS: Getrennte Sprachen, gemeinsame Prozessflüsse

Gerhard Svolba SAS Austria

(credits to David Weik, SAS Germany)

Links

- https://www.sas.com/de_at/webinars/ask-the-expert-serie-proc-python-wie-kann-ich-sas-und-python-kombinieren.html
- https://developer.sas.com/home.html
- https://blogs.sas.com/content/iml/2013/11/25/twelve-advantages-to-calling-r-from-the-sasiml-language.html
- https://blogs.sas.com/content/iml/2011/05/13/calling-r-from-sasiml-software.html

Agenda

- SAS und R
  - R-Studio Integration
  - Proc IML
- Open Source Nodes in SAS Model Studio
- SAS und Python
  - Modellverwaltung und Produktivstellung mit SAS Model Studio
  - Integration von mit dem Jupyter Notebook
  - Proc Python und SAS Studio

Dokumentation & Beispiele

developer.sas.com

SAS & R

CAS Bibliotheken in R Studio

SWAT Package im R Studio Standard

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

HMEQ

(Displaying up to 1,000 records)

	BAD	LOAN	MORTDUE	VALUE	REASON	JOB	YOJ	DEROG	DELINQ	CLAGE	NINQ	CLNO	DEBTINC
2	0	26800	46236	62711	DebtCon	Office	17.0	0	0	175.07506	1	22	33.059934
2100	0	26900	74982	126972	DebtCon	Office	0.0	0	0	315.81891	0	23	38.325990
3	0	26900	67144	92923	DebtCon	Other	16.0	0	0	89.11217	1	17	32.791478
4	0	26900	45763	73797	DebtCon	Other	23.0	NaN	0	291.59168	1	29	39.370858
5	0	27000	144901	178093	DebtCon	ProfExe	7.0	0	0	331.11397	0	34	40.566552
6	0	27000	NaN	38441	DebtCon	Other	8.0	1	NaN	85.17950	NaN	14	23.691176
7	0	27100	140305	237302	HomeImp	ProfExe	27.0	0	0	349.51427	2	18	37.420847
8	0	27100	118200	159739	DebtCon	ProfExe	6.0	0	1	117.15216	3	27	34.206008
9	0	27200	61000	104044	DebtCon	Other	16.0	0	1	141.98123	0	30	39.546508
10	0	27300	66878	93656	DebtCon	Office	11.0	0	0	153.79590	0	20	39.085006
11	0	27300	135823	185416	DebtCon	Mgr	1.0	NaN	NaN	135.35975	NaN	38	32.794976
12	0	27300	149950	177309	DebtCon	ProfExe	6.0	0	0	328.16812	0	35	39.159554

Environment | History | Connections | Tutorial

Help

dachsasviya41.ger.sas.com

- CASUSER(gerdaw)
- CPSAppData
- Formats
- ModelPerformanceData
- Models
- PostgresDB
- PricingResult
- Public
- Samples
- SystemData

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to File/Function Addins Project: (None)

HMEQ

(Displaying up to 1,000 records)

	BAD	LOAN	MORTDUE	VALUE	REASON	JOB	YOJ	DEROG	DELINQ	CLAGE	NINQ	CLNO	DEBTINC
2	0	26800	46236	62711	DebtCon	Office	17.0	0	0	175.07506	1	22	33.059934
2100	0	26900	74982	126972	DebtCon	Office	0.0	0	0	315.81891	0	23	38.325990
3	0	26900	67144	92923	DebtCon	Other	16.0	0	0	89.11217	1	17	32.791478
4	0	26900	45763	73797	DebtCon	Other	23.0	NaN	0	291.59168	1	29	39.370858
5	0	27000	144901	176093	DebtCon	ProfExe	7.0	0	0	331.11397	0	34	40.566552
6	0	27000	NaN	38441	DebtCon	Other	8.0	1	NaN	85.17950	NaN	14	23.691176
7	0	27100	140305	237302	HomeImp	ProfExe	27.0	0	0	349.51427	2	18	37.420847
8	0	27100	118200	159739	DebtCon	ProfExe	6.0	0	1	117.15216	3	27	34.206008
9	0	27200	61000	104044	DebtCon	Other	16.0	0	1	141.98123	0	30	39.546508
10	0	27300	66878	93656	DebtCon	Office	11.0	0	0	153.79590	0	20	39.085006
11	0	27300	135823	185416	DebtCon	Mgr	1.0	NaN	NaN	135.35975	NaN	38	32.794976
12	0	27300	149950	177309	DebtCon	ProfExe	6.0	0	0	328.16612	0	35	39.159554
13	0	27400	NaN	121625	DebtCon	Mgr	7.0	1	0	154.43785	0	27	33.373814
14	0	27400	41370	70528	DebtCon	Office	18.0	0	0	180.20434	1	21	33.495089
15	0	27500	150666	192346	HomeImp	Self	3.0	0	0	177.87147	0	17	39.978650
16	0	27600	54564	86568	DebtCon	Other	22.0	0	0	214.81574	1	27	35.223971
17	0	27600	150120	191972	HomeImp	Self	3.0	0	0	178.03849	1	16	41.807428
18	0	27600	124139	157940	DebtCon	Mgr	4.0	0	0	288.68654	0	24	38.226737
19	0	27700	63962	101676	DebtCon	Other	15.0	0	1	138.56941	0	30	37.384586

Showing 1 to 20 of 1,000 entries, 13 total columns

Console Terminal Jobs

R 4.0.3 · ~/

```
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[workspace loaded from ~/.RData]

> library(swat)
SWAT 1.6.3
> port <- 443
> server <- 'https://dachsasviya41.ger.sas.com/cas-shared-default-http'
> username <- readline(prompt="Enter your username: ")
Enter your username: gerdaw
> password <- getPass::getPass("Password: ")
> conn <- swat::CAS(server, port, username, password)
NOTE: Connecting to CAS and generating CAS action functions for loaded
      action sets...
NOTE: To generate the functions with signatures (for tab completion), set
```

Environment History Connections Tutorial

Help Refresh Connection Data

dachsasviya41.ger.sas.com CAS

- ModelPerformanceData
- Models
- PostgresDB
- PricingResult
- Public
  - CARS
  - CUST_CAMPAIGN_EN_CLEANSED
  - CUST_CAMPAIGN_EN_ORIGINAL
  - INSURANCE_CHURN
  - LIVEEO_MEDIUMFLOOD
  - NOSHOW-PROCPYTHON
  - OUTTABLE
  - POLLING-PROC-PYTHON
  - QA-PROC-PYTHON
  - SHOW-PROCPYTHON
  - SNACKS
  - SNACKS_LOOKUP

Files Plots Packages Help Viewer

Install Update

User Library

Name	Description	Version
askpass	Safe Password Entry for R, Git, and SSH	1.1
assertthat	Easy Pre and Post Assertions	0.2.1
bitops	Bitwise Operations	1.0-7
caTools	Tools: Moving Window Statistics, GIF, Base64, ROC AUC, etc	1.18.2
cli	Helpers for Developing Command Line Interfaces	3.2.0
crayon	Colored Terminal Output	1.5.1
curl	A Modern and Flexible Web Client for R	4.3.2
Deriv	Symbolic Differentiation	4.1.3
digest	Create Compact Hash Digests of R Objects	0.6.29
dplyr	A Grammar of Data Manipulation	1.0.8
ellipsis	Tools for Working with ...	0.3.2
fansi	ANSI Control Sequence Aware String Functions	1.0.3
generics	Common S3 Generics not Provided by Base R Methods Related to Model Fitting	0.1.2
getPass	Masked User Input	0.2-2
glue	Interpreted String Literals	1.6.2
gplots	Various R Programming Tools for Plotting Data	3.1.1
gtools	Various R Programming Tools	3.9.2
httr	Tools for Working with URLs and HTTP	1.4.2
jsonlite	A Simple and Robust JSON Parser and Generator for R	1.8.0
lifecycle	Manage the Life Cycle of your Package Functions	1.0.1
magrittr	A Forward-Pipe Operator for R	2.0.3
mime	Map Filenames to MIME Types	0.12
neuralnet	Training of Neural Networks	1.44.2
openssl	Toolkit for Encryption, Signatures and Certificates Based on OpenSSL	2.0.0

Daten von SAS nach R laden

```
dav_mortality <- defCasTable(conn, '_DAV_MORTALITYRATE_HMX', caslib = "public")
cas_DF = to.casDataFrame(dav_mortality)
dta = to.data.frame(cas_DF)
#dta <- read.csv(file="mortality.csv")

## Print 10 random observations from the mortality dataset for an assessment of the plausibility of the data
print(class(data))
sample_n(dta, 10)
```

```
[1] "data.frame"
```

Country	Year	Gender	Age	log_mortality
DEUTW	1994	Female	84	-2.326069
DEUTW	1997	Male	45	-5.688992
DEUTW	1971	Female	64	-4.191472
DNK	1988	Female	55	-4.824944
DEUTW	1996	Female	62	-4.897396
DEUTW	1978	Female	84	-2.006317
DNK	1984	Male	52	-4.809001
JPN	1970	Male	85	-1.563375
JPN	1987	Female	32	-7.671325
DEUTW	1987	Male	62	-3.961686

SAS Algorithmik in Python & R?!

SAS Visual Data Mining and Machine Learning Programming Guide

Data Science Pilot Action Set

CASL Lua Python R

Provides actions for automating data science workflows, including automatic machine learning pipeline exploration, execution and ranking.

```
# Find Best Decision Tree Configuration
tune_dt <- cas.autotune.tuneDecisionTree(conn,
            trainOptions=list(table = list(name = 'hmeq', where = '_PartInd_ = 0'),
            target = 'BAD',
            inputs = list('LOAN', 'MORTDUE', 'VALUE', 'REASON', 'JOB', 'YOJ', 'DEROG',
                          'DELINQ', 'CLAGE', 'NINQ', 'CLNO', 'DEBTINC'),
            nominals = list('BAD', 'REASON', 'JOB'),
            varImp = TRUE,
            casOut = list(name = 'tune_dt_model', replace = TRUE)))
```

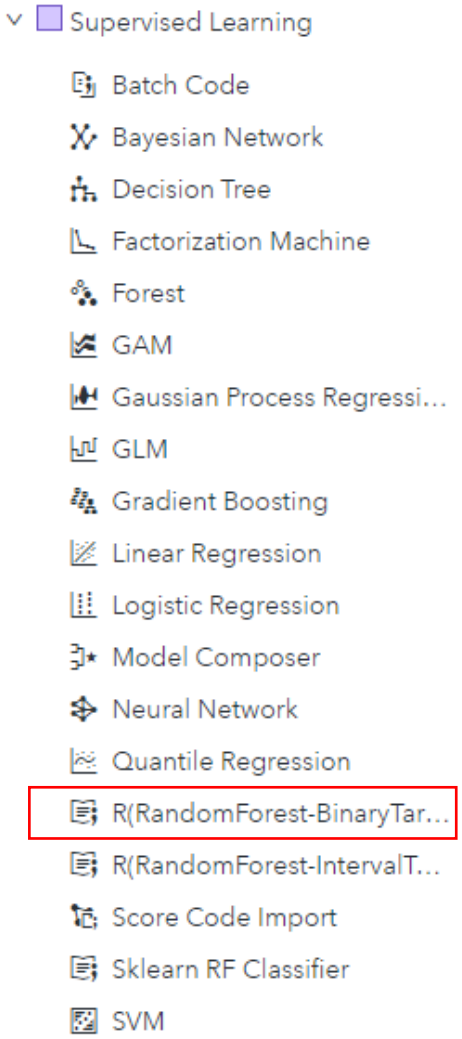
Proc IML

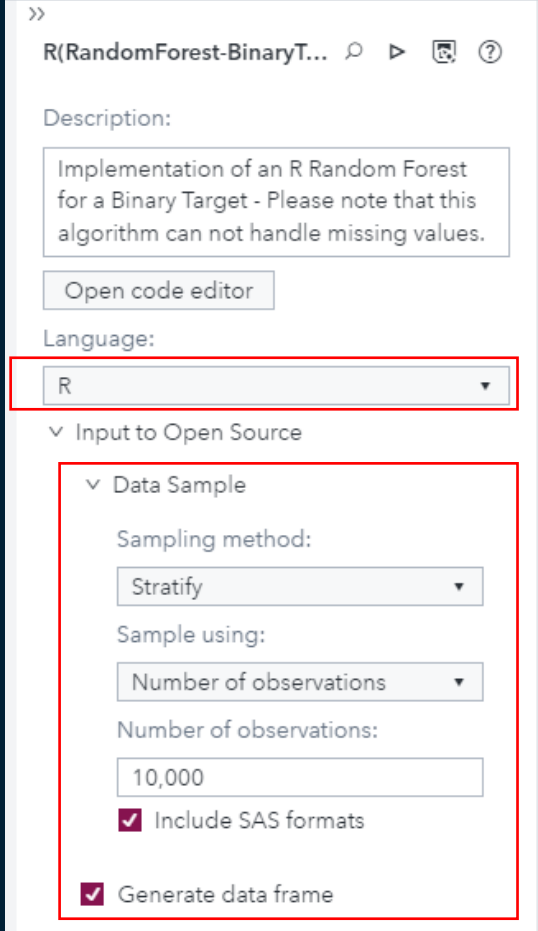
R Code mit SAS integrieren

```
proc iml;
call ExportDataSetToR("Sashelp.Class", "df" );
submit / R;
   names(df)
endsubmit;
```

Subroutine	R Source	SAS Destination
ImportDataSetFromR	R expression	SAS data set
ImportMatrixFromR	R expression	SAS/IML matrix
ImportTableFromR	R expression	SAS/IML table

Integration in SAS Model Studio





Open Source Code Editor

gerdaw-hmeq > R(RandomForest-BinaryTarget)

R Variables

Filter

- dm_class_input
- dm_classtarget_intovar
- dm_classtarget_level
- dm_dec_target
- dm_input
- dm_inputdf
- dm_interval_input
- dm_model
- dm_model_formula
- dm_nodedir
- dm_partition_train_val
- dm_partitionvar
- dm_predictionvar
- dm_scoreddf
- dm_traindf
- node_data.csv
- node_scored.csv

```r
library(randomForest)

# Fit RandomForest model w/ training data
dm_model <- randomForest(dm_model_formula, ntree=100, data=dm_traindf, importance=TRUE)

# Save MSE plot to PNG
png("rpt_forestMsePlot.png")
plot(dm_model, main='randomForest MSE Plot')
dev.off()

# Save VariableImportance to CSV
write.csv(importance(dm_model), file="rpt_forestIMP.csv", row.names=TRUE)

# Score full data
dm_scoreddf <- data.frame(predict(dm_model, dm_inputdf, type="prob"))
colnames(dm_scoreddf) <- c(dm_predictionvar[1], dm_predictionvar[2])
```

Live Demo:
Open Source in SAS Model Studio

SAS & Python

Open Source Modell durch SAS verwaltet

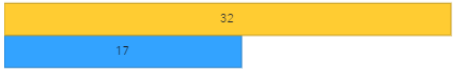
SAS® Model Manager - Manage Models

Models

Hide Dashboard

49

Total Number of Models



17 Published Models

Models per Score Code Type

Published Models per Destination

Search name | Advanced search | Add models | Compare

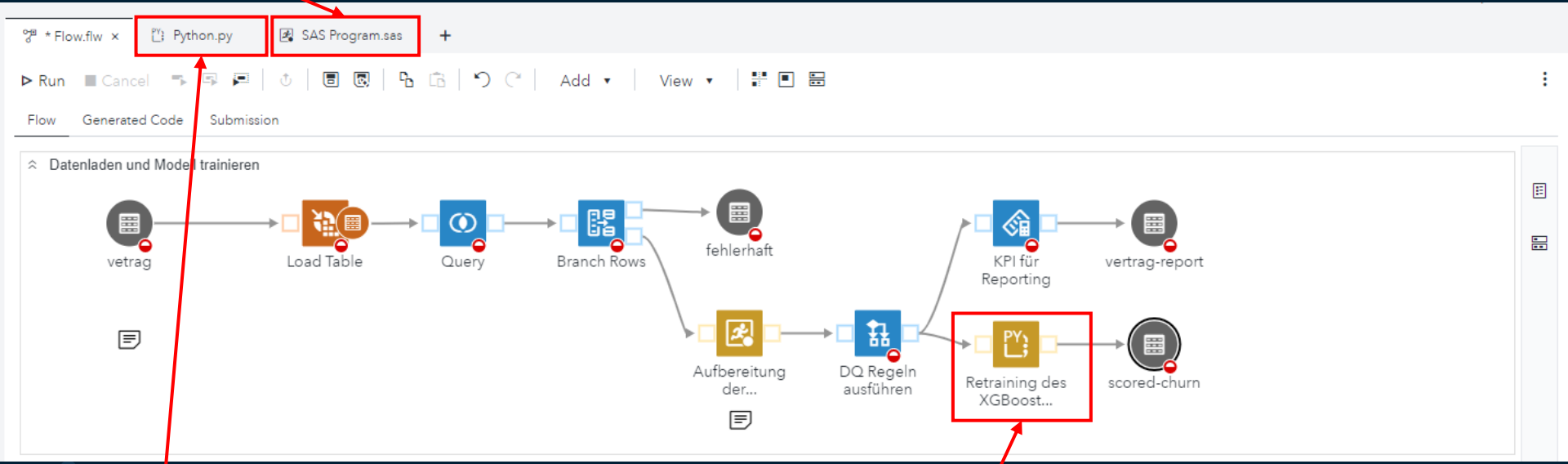
(10) gerdaw ×

Name	Role	Project (Version)	Model Attributes	Model Function	Modified By	Date Modified
HMEQ_XGBOOST		SAS-Python-Together, Version 1 (1.0)	Published, Python	Classification	G gerdaw	May 10, 2022 02:29 PM

SAS Studio

Kombination von visuellen Prozessflüssen mit Code-Editoren

SAS Code Editor



Python Code Editor

Python Programm als Schritt im Flow

SAS Studio

Der Python Code Editor

Python Code Editor

Daten direkt aus SAS als Dataframe

Dataframe zurück zu SAS

Syntax-Hilfe für Python

Ausgabe-Tabellen von Python anzeigen

Implizites Wrapping in Proc Python

SAS® Studio - Develop SAS Code

SAS Studio compute context

Start Page | * chainladder.py

Run | Cancel | Copy to My Snippets | Clear Log | Mar 29, 2022, 5:22:36 PM

Code

```
import pandas as pd
import chainladder as cl

#data = pd.read_csv('/sasdata/data/Insurance/Chainladder/clrd.csv')
data = SAS.sd2df('work.clrd')

# Create a triangle
triangle = cl.Triangle(
    data, origin='AccidentYear', development='DevelopmentYear',
    index=['GRNAME'], columns=['IncurLoss','CumPaidLoss','EarnedPremDIR'])

# Output
out_df = triangle.to_frame()
rc = SAS.df2sd(out_df, 'work.cl_triangle')
i
```

id
if
import
in
input
int
intern

Log | Output Data (1)

Errors (0) | Warnings (0) | Notes (27)

There are no messages.

```
proc python;
submit
NOTE: Python initialized.
Python 3.8.8 (default, Apr 13 2021, 19:58:26)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
!       ;

import pandas as pd
import chainladder as cl

#data = pd.read_csv('/sasdata/data/Insurance/Chainladder/clrd.csv')
data = SAS.sd2df('work.clrd')

# Create a triangle
triangle = cl.Triangle(
    data, origin='AccidentYear', development='DevelopmentYear',
    index=['GRNAME'], columns=['IncurLoss','CumPaidLoss','EarnedPremDIR'])

# Output
out_df = triangle.to_frame()
rc = SAS.df2sd(out_df, 'work.cl_triangle')

endsubmit;
quit;
```

Live Demo: Python Code in SAS Studio

Startseite | 00 Init.sas | 01 Python Hello World.py | 02 Python SAS Example.sas | 03 Python SAS Function.sas

Ausführen | Abbrechen | In Eigene Snippets kopieren | Code zu Ablauf | Log löschen | 08.09.2022, 10

Code

```
1  print("Hello World")
2  y = "Hi!"
3  def my_function():
4    print("Inside the my_script.py script")
5
6
7
8
```

Log

Fehler (0) | Warnungen (0) | Hinweise (10)

Es sind keine Meldungen vorhanden.

```
1    /* region: Generierte Präambel */
79
80   proc python;
81   submit
NOTE: Python initialized.
Python 3.8.8 (default, Apr 13 2021, 19:58:26)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
81 !       ;
```

SAS® Studio | GEL / SAS Studio Custom Steps

https://extviya4.emea.sas.com/SASStudio/main?locale=en_US&

SAS® Studio - Develop SAS Code

New | Options | View | Open | Save All

SAS Studio compute context

Explorer

- My Favorites
- Folder Shortcuts
- share-data
- SAS Content

Start Page

GET STARTED

- Program in SAS
- Build a flow
- Import data
- Query data

NEW Explore new features in SAS Studio

LEARN

Learn SAS Studio - videos, tutorials, and training

Learn SAS programming

STAY CONNECTED

Join the community

Request a feature

RECENTS

Insurance Churn.flw
/Users/gerdaw/My Folder/Flows
23.06.2022, 07:42:31

Von Open Source zu SAS und von SAS zu Open Source

Packages, Code Editoren & Prozeduren

Arbeiten mit dem Editor der Wahl

Konsistenter und Autorisierter Datenzugriff

Templates zur Wiederverwendung und Erweiterung der Funktionalität

Einheitliche Prozessflüsse für die Orchestrierung diverser Arbeitsweisen

8 wichtige Eigenschaften von SAS Viya

SAS Viya – Analytik All-in-one

Output und Ergebnisse

1 Analyseergebnisse in **Dashboards** aufbereiten und bereitstellen

2 **Big Data Analytics** – Mit NLP, Computer Vision, Machine/DeepLearning für neue Anforderungen gerüstet sein

SAS in Action

```
%let id = Destatis;
proc transpose;
proc report;
proc summary;
data _NULL_ ;
```

3 **SAS Programmierung** und bestehende Auswertungen weiterhin nutzen und ausbauen

4 Integration von SAS und **Open Source**

5 Erhöhung der Produktivität und Einbeziehen von unterschiedliche Benutzergruppen mit **NoCode/LowCode**

Governance

6 **Analyse-Code** strukturieren, dokumentieren, verwalten und wiederverwenden

7 Unterstützung des statistischen Lebenszyklus: **Data Governance, Model Governance**

8 **Statistische Geheimhaltung** durch Zellsperrung und CKM