Analytic Procedures in SAS Viya – Ein Überblick über das Methodenspektrum, das Paradigma, und die Möglichkeit „Altbekanntes“ weiter zu nutzen

Gerhard Svolba

Analytic Solutions Architect

SAS Austria

Twitter: gsvolba
https://github.com/gerhard1050
https://www.linkedin.com/in/gerhardsvolba/

Copyright © SAS Institute Inc. All rights reserved.

THE POWER TO KNOW.

This presentation shows you

- SAS Viya Procedures for supervised and unsupervised machine learning
- An Survival Analysis Example for Employee Headcount Analysis
- SAS Visual Frontends; and how they interact and generate SAS Code
- They layout of a SAS analytic procedure and how they generate CAS Actions
- An Articial Intelligence Example with Natural Language Processing and Object Detection in Realtime.

Copyright © SAS Institute Inc. All rights reserved.

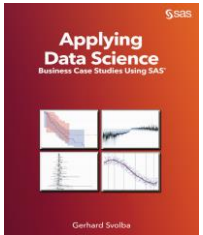
Using the PHSELECT procedure for time-to-event data

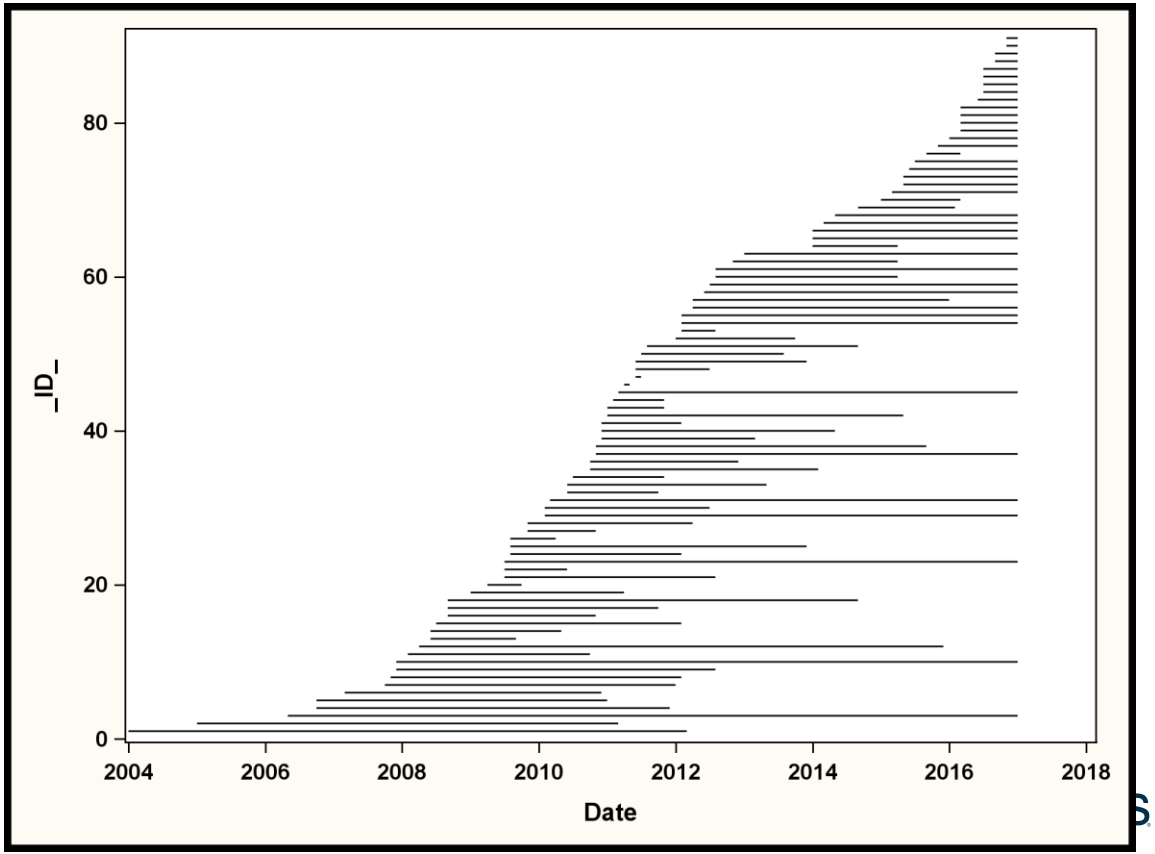
Copyright © SAS Institute Inc. All rights reserved.

Nicht zu allen Mitarbeitern haben wir ein „Ereignis-Datum“ (Glücklicherweise)

- Betrachten der Karrieren pro Mitarbeiter
  - Unterschiedliche Länge
  - Kündigung oder „zensiert“

- → KSFE 2018, Svolba, *„Kann ich die Verweildauer meiner Mitarbeiter analysieren und vorhersagen? Survival Analyse von SAS liefert die Antworten“*





Use PHSELECT to estimate a Cox-PropHazard Model

and output the score logic in a SAS program

```
PROC phselect DATA=casdata.Employees;
 CLASS department gender TechKnowHow / PARAM=effect REF=first;
 MODEL Duration*Status(1)= department gender TechKnowHow ;
 SELECTION method=lasso;
 CODE file="HeadcountScoreCode.sas"
           timepoint=6 12 18 24 36 48 60 showtime;
RUN;
```

Copyright © SAS Institute Inc. All rights reserved.

Use the score file to calculate the predicted survival for new employees

```
data casdata.employees_Scored;
 set casdata.employees_new;
 NewVarName = "Time_tmp";
 %inc "&c_path/020CAS/programs/HeadcountScoreCode.sas";
run;
```

Copyright © SAS Institute Inc. All rights reserved.

Structure Scored (WIDE) Data into a (LONG) Format for a line plot

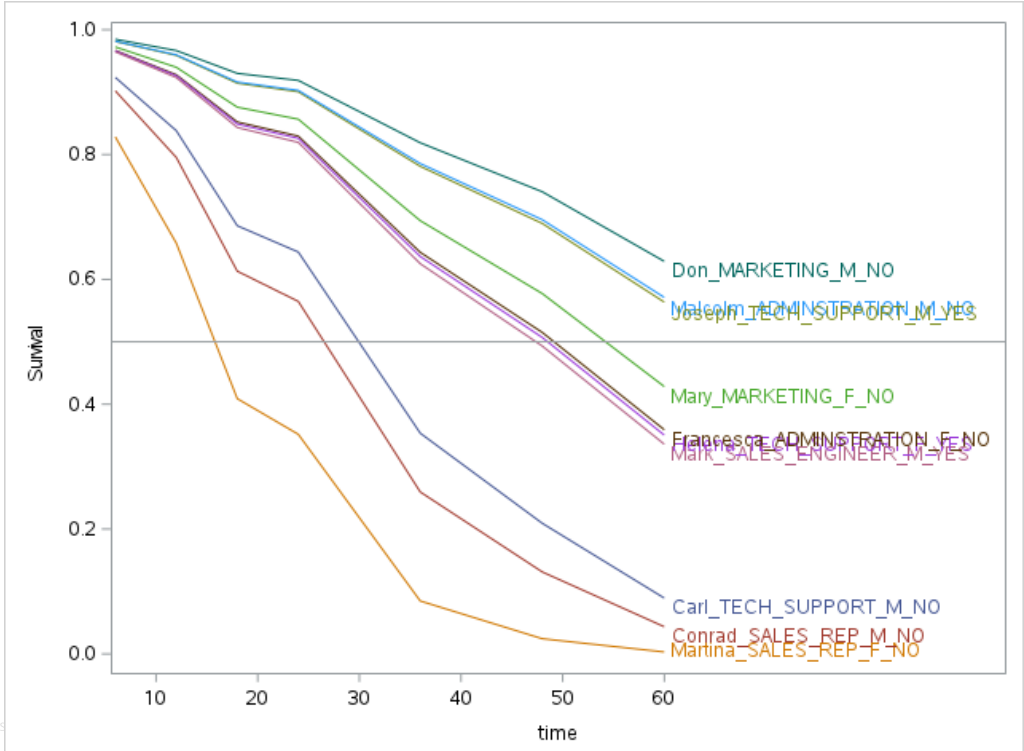
```
proc transpose data=casdata.employees_Scored
out=casdata.employees_Scored_tp;
 by  empno firstname department gender TechKnowHow;
 id NewVarName;
run;

data casdata.employees_Scored_tp;
 set casdata.employees_Scored_tp;
 Label=catx("_",Firstname,department,Gender,TechKnowHow);
 rename time_tmp = time;
 Survival = lag7(time_tmp);
 if substr(_name_,1,8)="Duration" then output;
run;
```

Copyright © SAS Institute Inc. All rights reserved.

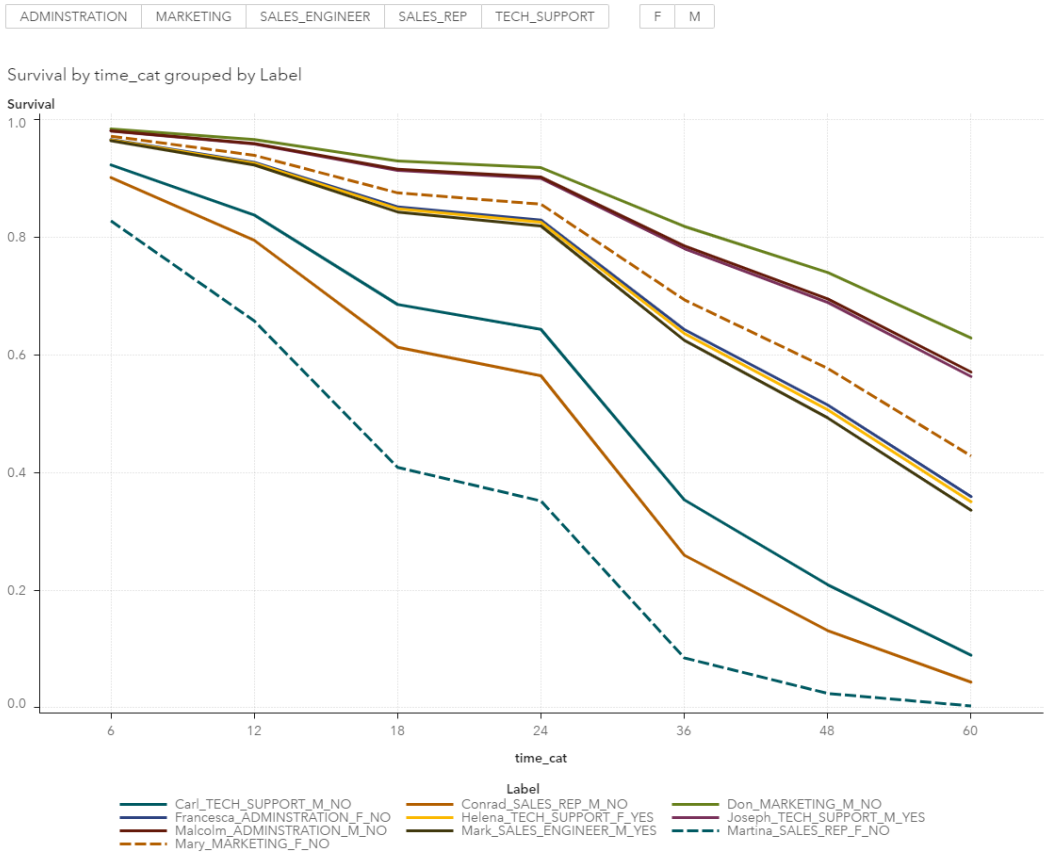
The SGPLOT procedure can run on CAS in-memory data

```
proc sgplot data=casdata.employees_Scored_tp;
  series x=Time y=survival /
      group=Label curvelabel;
  refline 0.5 / axis=y;
  yaxis min=0 max=1;
  xaxis min=0 max=60;
run;
```



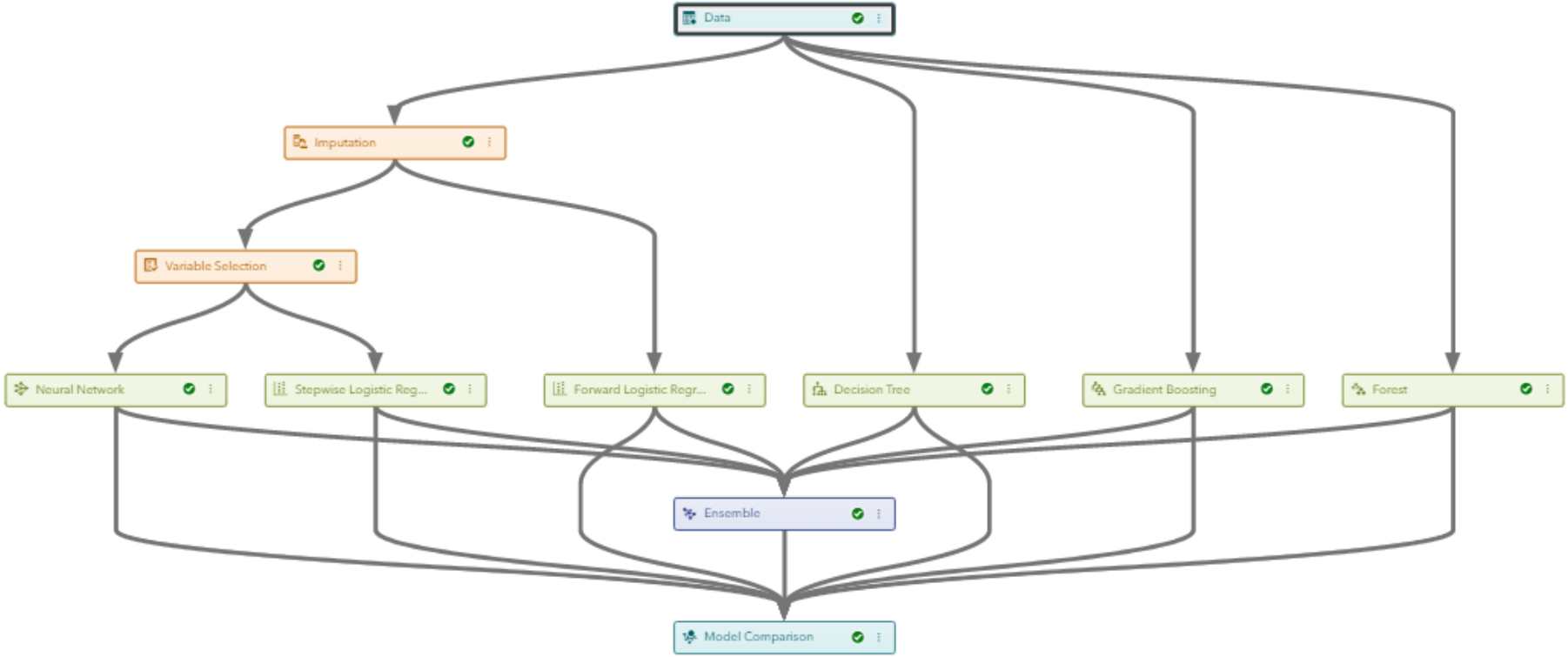
Copyright © SAS

Understand the content of your models: Displaying predictions in SAS Visual Analytics



Interactive Cutoff Analysis with SAS Visual Analytics

Pipelines im SAS Model Studio



Selected SAS Viya Analytic Procedures

Part 2 Statistics

The CORRELATION Procedure
The FREQTAB Procedure
The GAMMOD Procedure
The GENSELECT Procedure
The ICA Procedure
The KCLUS Procedure
The LMIXED Procedure
The LOGSELECT Procedure
The MBC Procedure
The MODELMATRIX Procedure
The NLMOD Procedure
The PCA Procedure
The PHSELECT Procedure
The PLSMOD Procedure
The QTRSELECT Procedure
The REGSELECT Procedure
The SPC Procedure
The TREESPLIT Procedure

SAS Visual Statistics

Part 3 Utility

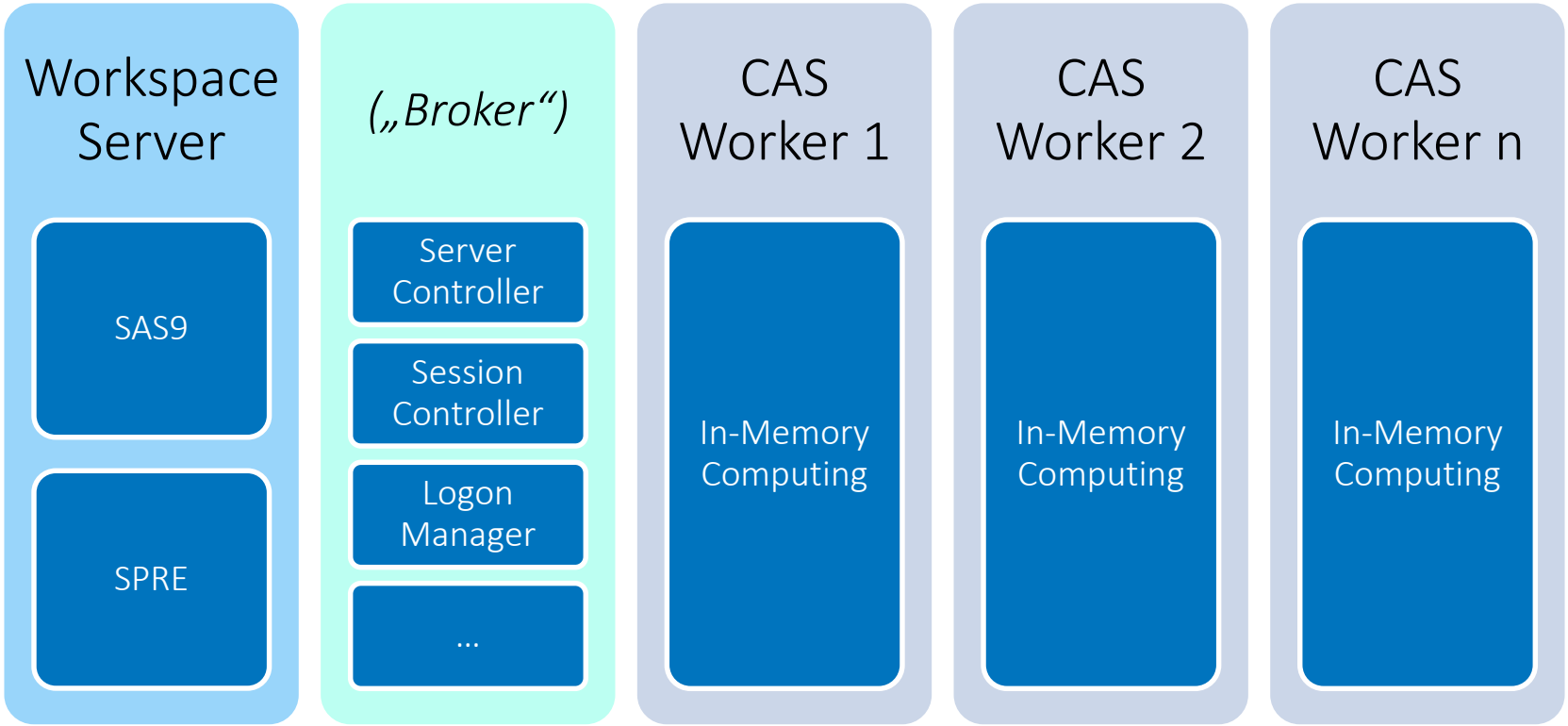
The ASSESS Procedure
The BINNING Procedure
The CARDINALITY Procedure
The PARTITION Procedure
The VARIMPUTE Procedure
The VARREDUCE Procedure

SAS Visual Data Mining and Machine Learning

The ASTORE Procedure
The BNET Procedure
The BOOLRULE Procedure
The FACTMAC Procedure
The FASTKNN Procedure
The FISM Procedure
The FOREST Procedure
The GMM Procedure
The GRADBOOST Procedure
The GVARCLUS Procedure
The MBANALYSIS Procedure
The MTLEARN Procedure
The MWPCA Procedure
The NNET Procedure
The RPCA Procedure
The SEMISUPLEARN Procedure
The SVDD Procedure
The SVMACHINE Procedure
The TEXTMINE Procedure
The TMSCORE Procedure
The TSNE Procedure

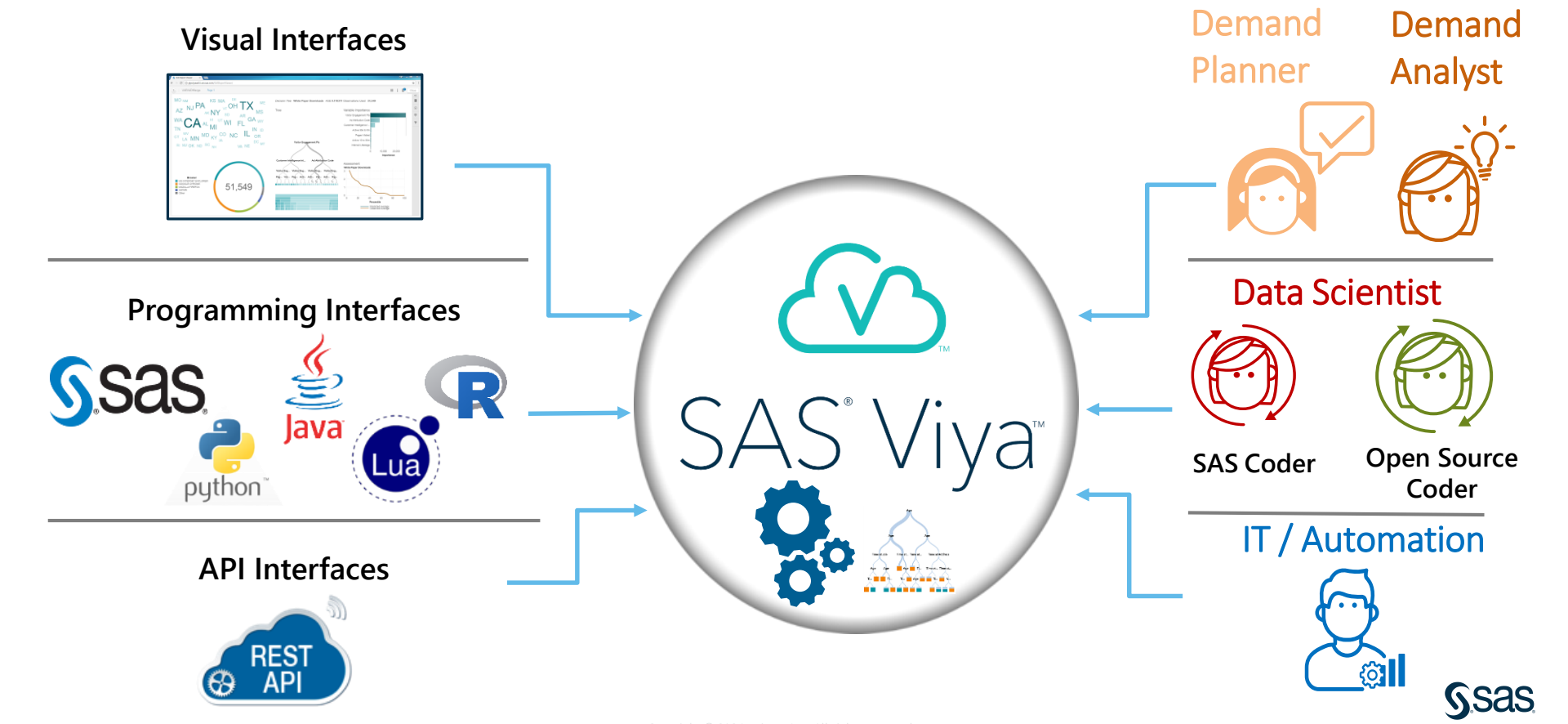
ght © SAS Institute Inc. All rights reserved.

Viya Architektur (schematisch, stark vereinfacht)



Copyright © SAS Institute Inc. All rights reserved.

Openness of the SAS Platform



Copyright © SAS Institute Inc. All rights reserved.

Multiple interfaces, single code base

Clients ask CAS to run “actions” on data

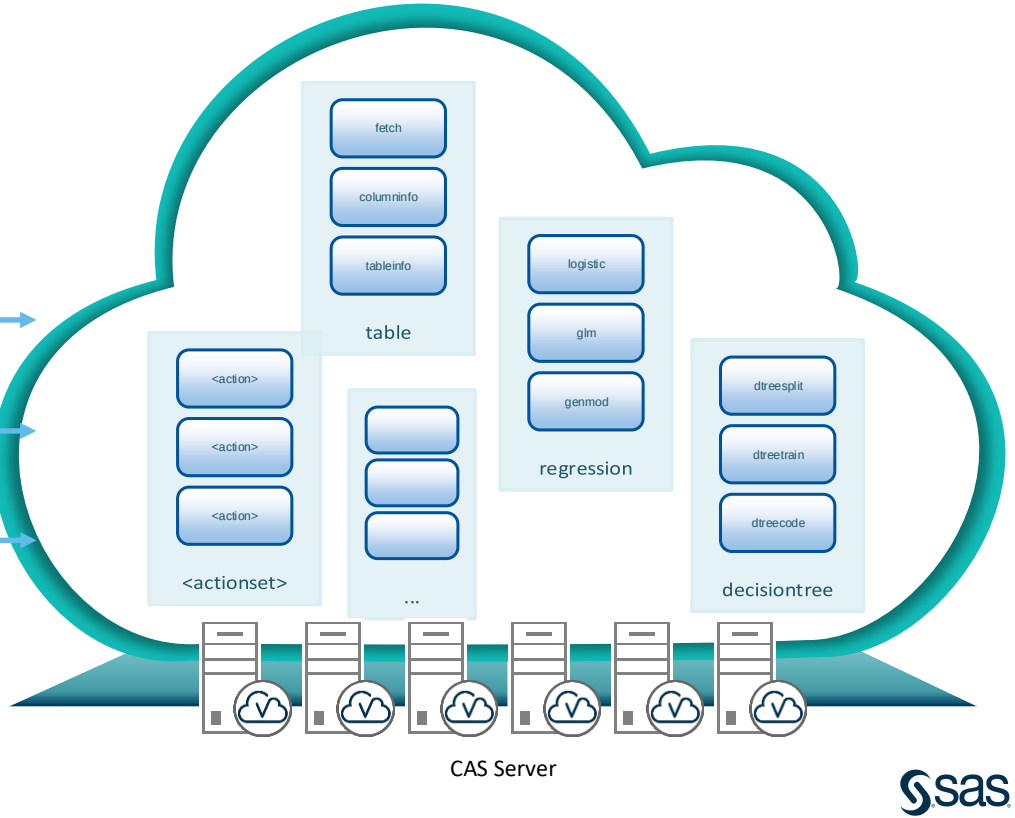
Visual Interfaces

Programming Interfaces

sas python Java Lua R

API Interfaces

REST API

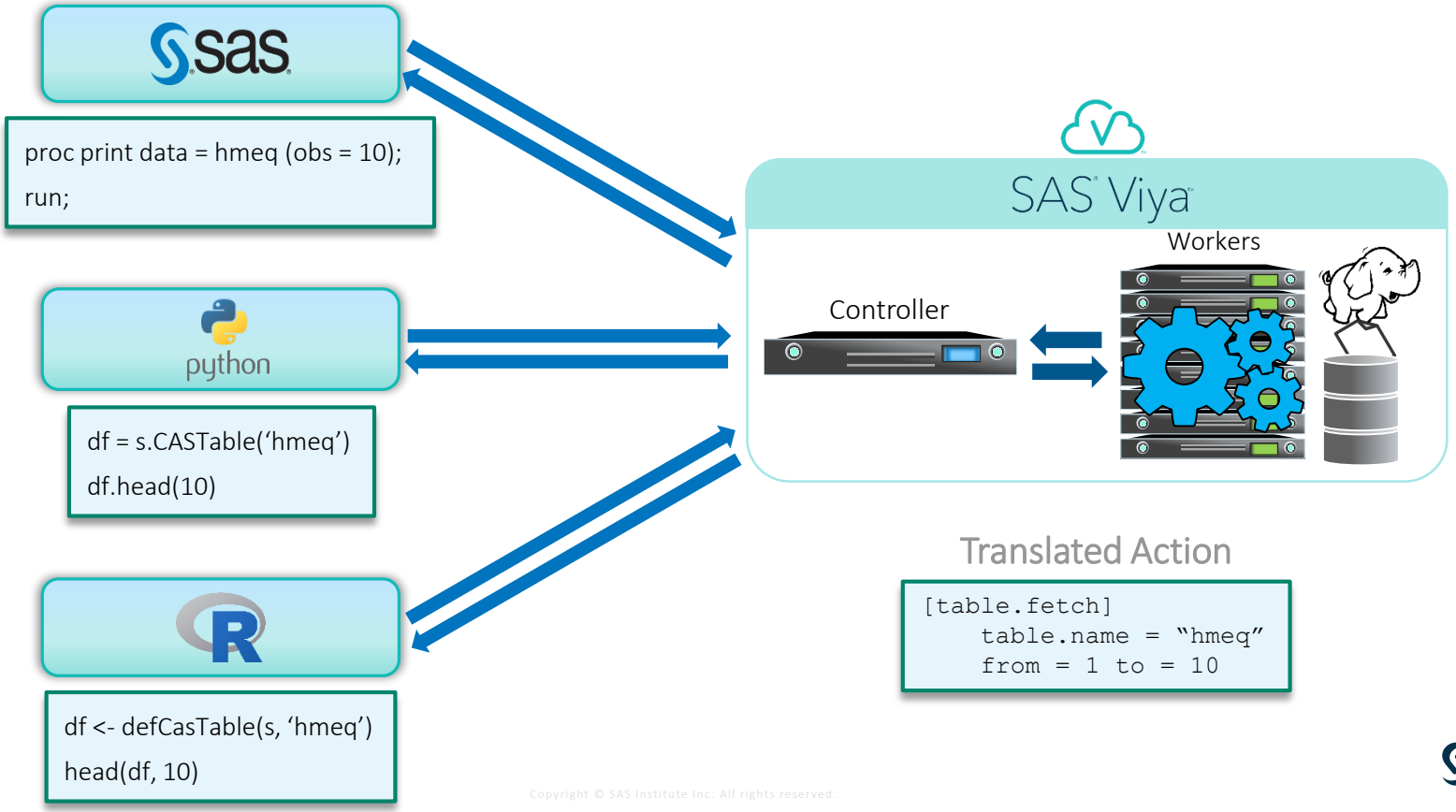


CAS Server

Copyright © SAS Institute Inc. All rights reserved.

SAS® Viya™: New Interface APIs

Different Languages – Same Power



Copyright © SAS Institute Inc. All rights reserved.

Selected SAS Viya Analytic Procedures

Part 2 Statistics

The CORRELATION Procedure
The FREQTAB Procedure
The GAMMOD Procedure
The GENSELECT Procedure
The ICA Procedure
The KCLUS Procedure
The LMIXED Procedure
The LOGSELECT Procedure
The MBC Procedure
The MODELMATRIX Procedure
The NLMOD Procedure
The PCA Procedure
The PHSELECT Procedure
The PLSMOD Procedure
The QTRSELECT Procedure
The REGSELECT Procedure
The SPC Procedure
The TREESPLIT Procedure

SAS Visual Statistics

Part 3 Utility

The ASSESS Procedure
The BINNING Procedure
The CARDINALITY Procedure
The PARTITION Procedure
The VARIMPUTE Procedure
The VARREDUCE Procedure

SAS Visual Data Mining and Machine Learning

The ASTORE Procedure
The BNET Procedure
The BOOLRULE Procedure
The FACTMAC Procedure
The FASTKNN Procedure
The FISM Procedure
The FOREST Procedure
The GMM Procedure
The GRADBOOST Procedure
The GVARCLUS Procedure
The MBANALYSIS Procedure
The MTLEARN Procedure
The MWPCA Procedure
The NNET Procedure
The RPCA Procedure
The SEMISUPLEARN Procedure
The SVDD Procedure
The SVMACHINE Procedure
The TEXTMINE Procedure
The TMSCORE Procedure
The TSNE Procedure

ght © SAS Institute Inc. All rights reserved.

Example Layout of a SAS Viya Machine Learning Procedure

Model Parameters

Partition Data

Perform Hyperparameter Tuning

Model Definition (Target, Inputs)

Generate Model Output Tables
(Fit, Variable Importance, …)

Save Model as an ASTORE

```
proc gradboost data=public.fraud_vsd
    earlystop(tolerance=0 stagnation=5)
    numBin=20 binmethod=BUCKET
    maxdepth=6
    maxbranch=2
    minleafsize=5
    assignmissing=USEINSEARCH minuseinsearch=1
    seed=12345
    printtarget
;
partition fraction (validate=0.7);
autotune useparameters=CUSTOM tuningparameters=(
    lasso(LB=0 UB=10 INIT=0)
    learningrate(LB=0.01 UB=1 INIT=0.1)
    ntrees(LB=20 UB=150 INIT=100)
    ridge(LB=0 UB=10 INIT=0)
    samplingrate(LB=0.1 UB=1 INIT=0.5)
    vars_to_try(LB=1 UB=19 INIT=19)
    )
    searchmethod=GA objective=KS maxtime=3600
    maxevals=50 maxiters=5 popsize=10
    targetevent='1'
;
target Fraud_Flag / level=nominal;
input age otherincome netincome / level=interval;
input gender marital_status education / level=nominal;
ods output
    VariableImportance    = &dm_lib..VarImp
    Fitstatistics         = &dm_data_outfit
    PredProbName = &dm_lib..PredProbName
    PredIntoName = &dm_lib..PredIntoName
    TunerResults = &dm_lib..tuneresults
    BestConfiguration = &dm_lib..tunebest(drop=name)
;
id 'accnbr'n 'age_emp_edu_peer_group'n 'ApplicationId'n '
savestate rstore=casdata.fraud_gradboost1;
run;
```

Copyright © SAS Institute Inc. All rights re

SAS Actions are generated automatically by the procedure

And can be reviewed with the `proc cas; history{first=-100}; run;` statement

```
proc cas;
  action builtins.loadActionSet / actionSet='decisionTree'; /* (SUCCESS) */
  action builtins.loadActionSet / actionSet='Sampling'; /* (SUCCESS) */
  action sampling.srs / table={name='FRAUD_VSD', caslib='public'}, sampPct=70, partInd=true, output={casOut={name='_data_',
   caslib='CASUSER(sasdemo01)', replace=true}, copyVars='ALL', partIndName='_Fraction_PartInd_'}; /* (SUCCESS) */
  action builtins.loadActionSet / actionSet='autotune'; /* (SUCCESS) */
  action autotune.tuneGradientBoostTree / tunerOptions={maxEvals=50, maxIters=5, maxTime=3600, popSize=10,
   userDefinedPartition=true, searchMethod='GA', objective='KS', targetEvent='1'}, useParameters='custom',
   tuningParameters={{namePath='nTree', lowerBound=20, upperBound=150, initValue=100}, {namePath='m', lowerBound=1,
   upperBound=19, initValue=19}, {namePath='learningRate', lowerBound=0.01, upperBound=1, initValue=0.1},
   {namePath='subSampleRate', lowerBound=0.1, upperBound=1, initValue=0.5}, {namePath='lasso', lowerBound=0, upperBound=10,
   initValue=0}, {namePath='ridge', lowerBound=0, upperBound=10, initValue=0}}, trainOptions={inputs={'age', 'OtherIncome',
   'NetIncome', 'Gender', 'Marital_Status', 'Education'}, table={name='_data_', casLib='casuser', where='_Fraction_PartInd_=0
   and Fraud_Flag NE .'}, casout={name='_model_', replace='TRUE', casLib='casuser'}, target='Fraud_Flag', nominals={'Gender',
   'Marital_Status', 'Education', 'Fraud_Flag'}, nbins=20, maxlevel=7, maxbranch=2, leafsize=5, missing='USEINSEARCH',
   minuseinsearch=1, ntree=100, seed=12345, binorder=true, varimp=true, mergebin=true, encodeName=true,
   saveState={name='FRAUD_GRADBOOST1', caslib='casdata', replace=true}, copyvars={'accnbr', 'age_emp_edu_peer_group',
   'ApplicationId', 'CCity', 'Company', 'CZipcode', 'Email', 'employer_date_peer_group', 'GivenName', 'NationalID', 'PCity',
   'PZipcode', 'Surname', 'Telephone'}, validTable={name='_data_', casLib='casuser', where='_Fraction_PartInd_=1 and Fraud_Flag
   NE .'}, earlyStop={stagnation=5}}, scoreOptions={table={name='_data_', casLib='casuser', where='_Fraction_PartInd_=1 and
   Fraud_Flag NE .'}, model={name='_model_', casLib='casuser'}, copyvars={'accnbr', 'age_emp_edu_peer_group', 'ApplicationId',
   'CCity', 'Company', 'CZipcode', 'Email', 'employer_date_peer_group', 'GivenName', 'NationalID', 'PCity', 'PZipcode',
   'Surname', 'Telephone'}, encodeName=true}; /* (WARNING) */
  action decisionTree.gbtreeScore / table={name='_data_', caslib='CASUSER(sasdemo01)', where='_Fraction_PartInd_=0 and
   Fraud_Flag NE .'}, modelTable={name='_model_', caslib='CASUSER(sasdemo01)'}, copyVars={'accnbr', 'age_emp_edu_peer_group',
   'ApplicationId', 'CCity', 'Company', 'CZipcode', 'Email', 'employer_date_peer_group', 'GivenName', 'NationalID', 'PCity',
   'PZipcode', 'Surname', 'Telephone'}, encodeName=true; /* (SUCCESS) */
  action decisionTree.gbtreeScore / table={name='_data_', caslib='CASUSER(sasdemo01)', where='_Fraction_PartInd_=1 and
   Fraud_Flag NE .'}, modelTable={name='_model_', caslib='CASUSER(sasdemo01)'}, copyVars={'accnbr', 'age_emp_edu_peer_group',
   'ApplicationId', 'CCity', 'Company', 'CZipcode', 'Email', 'employer_date_peer_group', 'GivenName', 'NationalID', 'PCity',
   'PZipcode', 'Surname', 'Telephone'}, encodeName=true; /* (SUCCESS) */
;
```

SAS Online Doc

[CAS Actions by Name]

Copyright © SAS Institute Inc. All rights reserved.

Autotuning mit SAS (Hyperparameter Tuning)

Autotune Action Set: Syntax

Provides actions to tune machine learning algorithm hyperparameters

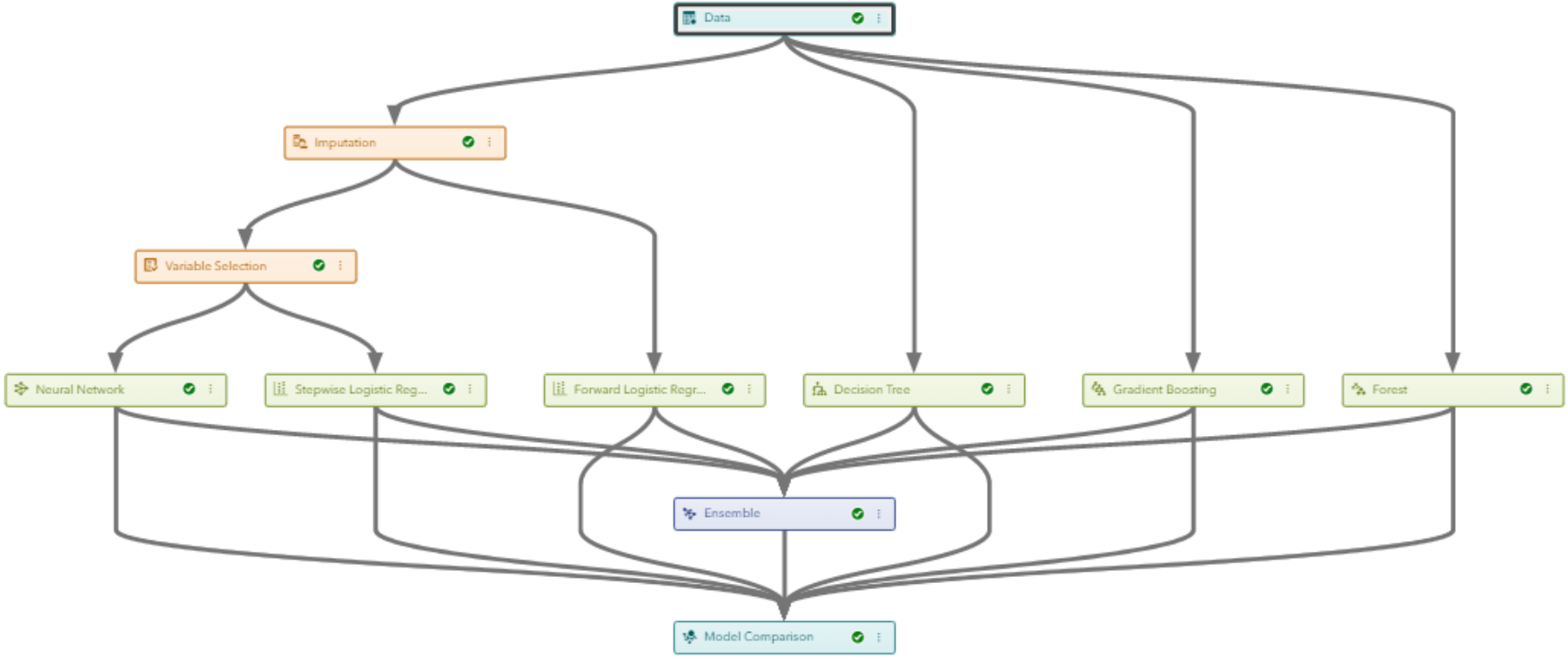
Syntax ▾ Details ▾ Examples ▾

Table of Actions

Action Name	Description
tuneBnet	Automatically adjusts Bayesian network classifier parameters to tune a model for minimum error
tuneDecisionTree	Automatically adjusts decision tree parameters to tune a model for minimum error
tuneFactMac	Automatically adjusts factorization machine parameters to tune a model for minimum error
tuneForest	Automatically adjusts forest parameters to tune a model for minimum error
tuneGradientBoostTree	Automatically adjusts gradient boosting tree parameters to tune a model for minimum error
tuneNeuralNet	Automatically adjusts neural network parameters to tune a model for minimum error
tuneSvm	Automatically adjusts support vector machine parameters to tune a model for minimum error

Copyright © SAS Institute Inc. All rights reserved.

Pipelines im SAS Model Studio



Selected SAS Viya Procedures that are called from SAS Model Studio

```
proc forest data=&dm_datalib..'DM_3G22ODBMI7MNHBPS6T
      minleafsize=5 seed=12345 loh=0 numbin=20 binmet
      assignmissing=USEINSEARCH minuseinsearch=1
      vote=PROBABILITY printtarget
   ;
   partition rolevar='_PartInd_'n (TRAIN='1' VALIDATE
   autotune useparameters=CUSTOM tuningparameters=(
      maxdepth(init=20 lb=1 ub=29)
```

```
proc treesplit data=&dm_datalib..'DM_3G22ODBMI7MNHBP
      nsurrogates=1 minleafsize=5 maxbranch=2 assignm
      minuseinsearch=1
      pruningtable
      seed=12345
      treeplot printtarget;
   autotune useparameters=CUSTOM tuningparameters=(
      maxdepth(LB=1 UB=19 INIT=10)
```

```
proc svmachine data=&dm_datalib..'_input_3KJWODXBTJW
      (&dm_data_caslib)
      maxiter = 25
      tolerance = 1.0E-6
      printtarget
   ;
   input %dm_interval_input / level = interval;
   input %dm_class_input
```

```
proc nnet data=&dm_datalib..'_input_49WCM2IT4WCG7C2S
      printtarget
      standardize=MIDRANGE;
   input %dm_interval_input / level=interval;
   input
      %dm_class_input
      / level=nominal;
   target 'Fraud Flag'n / level=nominal;
```

```
proc gradboost data=&dm_datalib..'DM_3G22ODBMI7MNHBP
      earlystop(tolerance=0 stagnation=5)
      numBin=20 binmethod=BUCKET
      maxdepth=6
      maxbranch=2
      minleafsize=5
      assignmissing=USEINSEARCH minuseinsearch=1
      seed=12345
```

```
proc genselect data=dmcaslib.'_input_9EQFF0IR82C1C6B
      tech=NRRIDG normalize=YES
   ;
   partition rolevar='_PartInd_'n (TRAIN='1' VALIDATE
   class
      %dm_class_input
      /
      order=formatted
```

Selected SAS Viya Analytic Procedures

SAS Visual Data Mining and Machine Learning

The ASTORE Procedure
The BNET Procedure
The BOOLRULE Procedure
The FACTMAC Procedure
The FASTKNN Procedure
The FISM Procedure
The FOREST Procedure
The GMM Procedure
The GRADBOOST Procedure
The GVARCLUS Procedure
The MBANALYSIS Procedure
The MTLEARN Procedure
The MWPCA Procedure
The NNET Procedure
The RPCA Procedure
The SEMISUPLEARN Procedure
The SVDD Procedure
The SVMACHINE Procedure
The TEXTMINE Procedure
The TMSCORE Procedure
The TSNE Procedure

SAS Visual Statistics

Part 2 Statistics

The CORRELATION Procedure
The FREQTAB Procedure
The GAMMOD Procedure
The GENSELECT Procedure
The ICA Procedure
The KCLUS Procedure
The LMIXED Procedure
The LOGSELECT Procedure
The MBC Procedure
The MODELMATRIX Procedure
The NLMOD Procedure
The PCA Procedure
The PHSELECT Procedure
The PLSMOD Procedure
The QTRSELECT Procedure
The REGSELECT Procedure
The SPC Procedure
The TREESPLIT Procedure

Part 3 Utility

The ASSESS Procedure
The BINNING Procedure
The CARDINALITY Procedure
The PARTITION Procedure
The VARIMPUTE Procedure
The VARREDUCE Procedure

ght © SAS Institute Inc. All rights reserved.

Copyright © SAS Institute Inc. All rights reserved.

Copyright © SAS Institute Inc. All rights reserved.

Robust Principal Component Analysis

RPCA Procedure in SAS Viya

Low Rank Matrix

Obs	index	X	Y
1	1	0.46412	0.46722
2	2	0.46443	0.46753
3	3	0.46434	0.46744
4	4	0.46490	0.46800
5	5	0.04435	0.04464
6	6	0.30894	0.31100
7	7	0.44700	0.44998
8	8	0.41900	0.42180

+

Sparse Matrix

Obs	index	X	Y
1	1	0.05788	0.04278
2	2	0.17757	0.11547
3	3	0.16366	0.07556
4	4	0.36110	0.40700
5	5	0.05665	-0.01364
6	6	0.00106	0.00000
7	7	0.00000	3.97102
8	8	0.00000	0.05920

=

Original Matrix

Obs	index	X	Y
1	1	0.522	0.510
2	2	0.642	0.583
3	3	0.628	0.543
4	4	0.826	0.875
5	5	0.101	0.031
6	6	0.310	0.311
7	7	0.447	4.421
8	8	0.419	0.481

Where:

Sparse Matrix

Obs	index	X	Y
1	1	0.05788	0.04278
2	2	0.17757	0.11547
3	3	0.16366	0.07556
4	4	0.36110	0.40700
5	5	0.05665	-0.01364
6	6	0.00106	0.00000
7	7	0.00000	3.97102
8	8	0.00000	0.05920

Noise

Anomalies

Copyright © SAS Institute Inc. All rights reserved.

Code Example

Daten:
- Maschinendaten

Coding Sprache:
- SAS Procedure
- CASL

Ziel:
- Auffinden von Anomalien

Robust Principle Component Analysis

```
proc rpca data=public.PHM08 method=alm lambdaweight=2
          outlowrank=casuser.low outsparse=casuser.sparse;
   id engine cycle;
   input X1-X24;
   svd method=eigen;
   where engine in (1,22,45,53,82,105,167,179);
run;
```

Augmented Lagrange Multiplier Method

In general, the augmented Lagrange method is used to solve nonlinear constrained optimization problems. In the case of PCP, an augmented Lagrange function is used to reformulate the PCP problem as the following nonlinear unconstrained optimization problem:

$$\text{minimize} \quad l(L,S,Y) = ||L||_* + \lambda||S||_1 + < Y, M-L-S > + \frac{\mu}{2}||M-L-S||_F^2$$

Candès et al. (2011) use the ALM method to find the solution to the preceding optimization problem. The basic idea is to update S, L, and Y iteratively. At iteration k, given L_k and Y_k, the first step is to find S_{k+1} by minimizing $l(L_k, S, Y_k)$. In the second step, L_{k+1} is obtained by the singular value thresholding operator, which minimizes $l(L, S_{k+1}, Y_k)$.[6] Next the Lagrange multiplier Y_{k+1} is updated. For more information, see Candès et al. (2011).

Copyright © SAS Institute Inc. All rights reserved.

Code Example

Daten:

- Maschinendaten

Coding Sprache:

- SAS Procedure
- CASL

Ziel:

- Auffinden von Anomalien

```
proc rpca data=public.PHM08 method=alm lambdaweight=2
          outlowrank=casuser.low outsparse=casuser.sparse;
   id engine cycle;
   input X1-X24;
   svd method=eigen;
   where engine in (1,22,45,53,82,105,167,179);
run;
```

Robust Principle Component Analysis

```
ods trace on;
 proc cas;
     loadactionset "tkrpca";
     action robustpca /
         table={caslib="public", name="PHM08",
                where="engine in (1,22,45,53,82,105,167,179)"}
         inputs={{name="X1"},{name="X2"},{name="X3"},
                 {name="X4"},{name="X5"},{name="X6"},
                 {name="X7"},{name="X8"},{name="X9"},
                 {name="X10"},{name="X11"},{name="X12"},
                 {name="X13"},{name="X14"},{name="X15"},
                 {name="X16"},{name="X17"},{name="X18"},
                 {name="X19"},{name="X20"},{name="X21"},
                 {name="X22"},{name="X23"},{name="X24"}}
         method="ALM"
         decomp="svd"
         lambdaweight=2
         svdmethod="EIGEN"
         outmat={lowrankmat={name="casllow" replace=True},
                 sparsemat={name="caslsparse" replace=True}}
         outsvd={svdleft={name="svdleft" replace=True},
                 svddiag={name="svddiag" replace=True},
                 svdright={name="svdright" replace=True}}
     ;
 run;
```

Copyright © SAS Institute Inc. All rights reserved.

Code Example

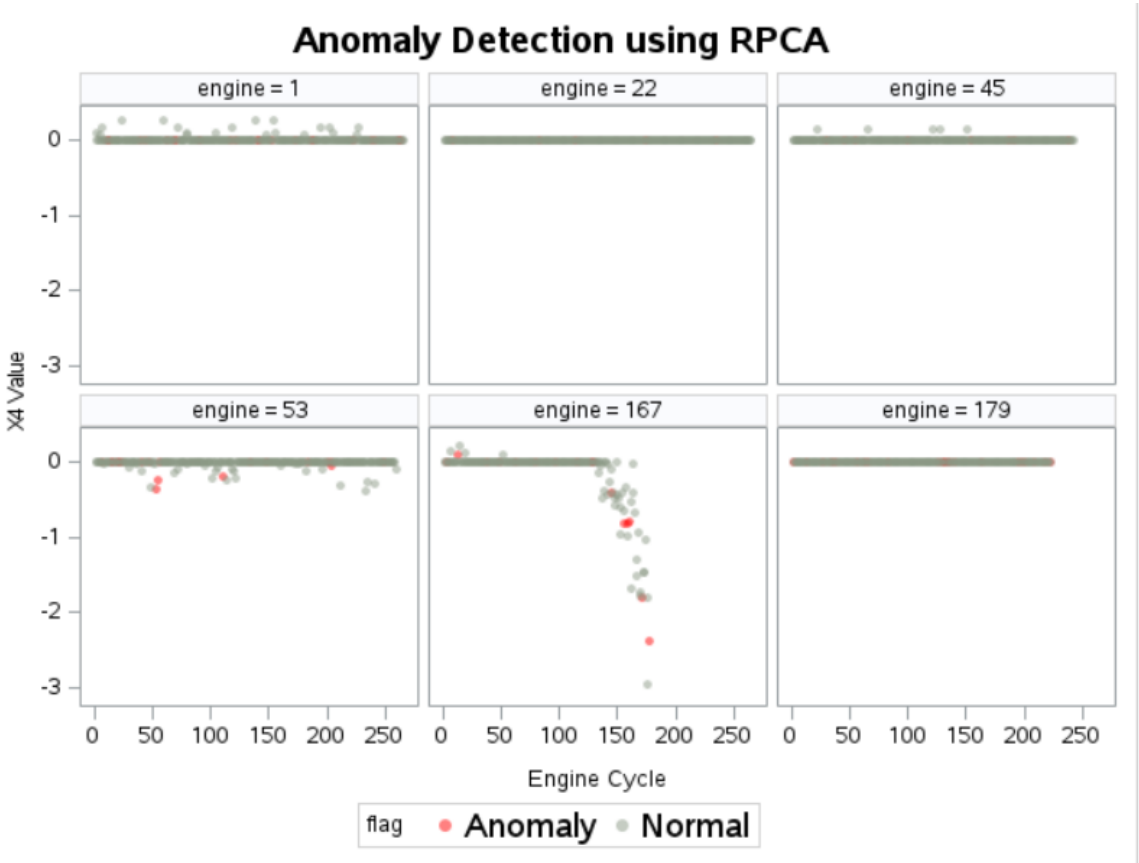
Daten:
- Maschinendaten

Coding Sprache:
- SAS Procedure
- CASL

Ziel:
- Auffinden von Anomalien

```
proc rpca data=public.PHM08 method=alm lambdaweight=2
          outlowrank=casuser.low outsparse=casuser.sparse;
   id engine cycle;
   input X1-X24;
   svd method=eigen;
   where engine in (1,22,45,53,82,105,167,179);
run;
```

Robust Principle Component Analysis



Copyright © SAS Institute Inc. All rights reserved.

SAS Solar Farm Analysis Demo

Copyright © SAS Institute Inc. All rights reserved.

Content of the Demo

- Training a **Natural Language Processing** (NLP) for the Chatbot with the SAS DLPY package
- **Time Series Forecasting** and Visual Data Analysis with SAS Visual Forecasting
- Training an **Object Recognition** model with the SAS DLPY package
- **Publishing this SAS model** the SAS Event Stream Processing engine implemented in a **Drone**
- (Interaction with the Drone via Voice Commands)

Copyright © SAS Institute Inc. All rights reserved.

DLPy - SAS Viya Deep Learning API for Python

- https://github.com/sassoftware/python-dlpy
- Overview
- DLPy is a high-level Python library for the SAS Deep learning features available in SAS Viya. DLPy is designed to provide an efficient way to apply deep learning methods to image, text, and audio data. DLPy APIs created following the Keras APIs with a touch of PyTorch flavor.

Copyright © SAS Institute Inc. All rights reserved.

DLPy - SAS Viya Deep Learning API for Python

An efficient way to apply deep learning methods to image, text, and audio data.

Additional Resources

•DLPy examples: https://github.com/sassoftware/python-dlpy/tree/master/examples

•DLPy API documentation sassoftware.github.io/python-dlpy.

•SAS SWAT for Python

•SAS ESPPy

•A series of Videos on DLPy examples:

- Introduction to the series
- Image classification using CNNs
- Object detection using TinyYOLOv2
- Import and export deep learning models with ONNX
- Text classification and text generation using RNNs
- Time series forecasting using RNNs

Copyright © SAS Institute Inc. All rights reserved.

This presentation shows you

- SAS Viya Procedures for supervised and unsupervised machine learning
- An Survival Analysis Example for Employee Headcount Analysis
- SAS Visual Frontends; and how they interact and generate SAS Code
- They layout of a SAS analytic procedure and how they generate CAS Actions
- An Articial Intelligence Example with Natural Language Processing and Object Detection in Realtime.

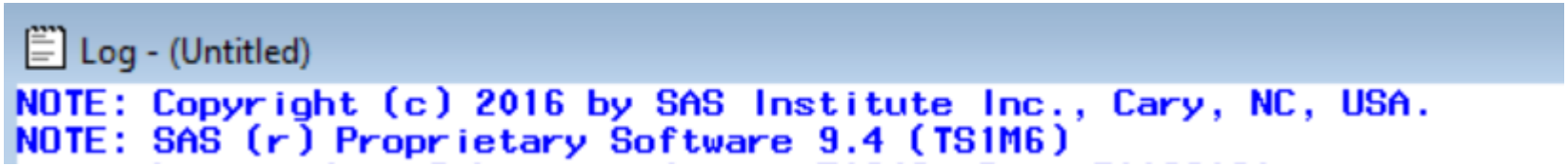
Copyright © SAS Institute Inc. All rights reserved.

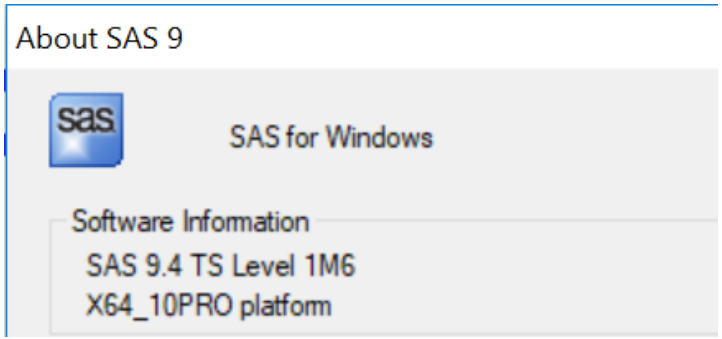
Additional Links

- SAS Online Documentation
  - https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=pgmsashome&docsetTarget=home.htm&locale=en
- SAS Viya 14-Day Test – Free Software
  - https://www.sas.com/de_de/trials.html
- SAS University Edition (SAS9)
  - https://www.sas.com/en_us/software/university-edition.html
- Paper SAS2184-2018 Parallel Programming with the DATA Step: Next Steps David Bultman and Jason Secosky, SAS Institute Inc., Cary, NC
  - https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/2184-2018.pdf

Copyright © SAS Institute Inc. All rights reserved.

Actual SAS 9.4 Version –> SAS9.4 M6





Copyright © SAS Institute Inc. All rights reserved.

SAS Bookstore → https://support.sas.com/en/books.html