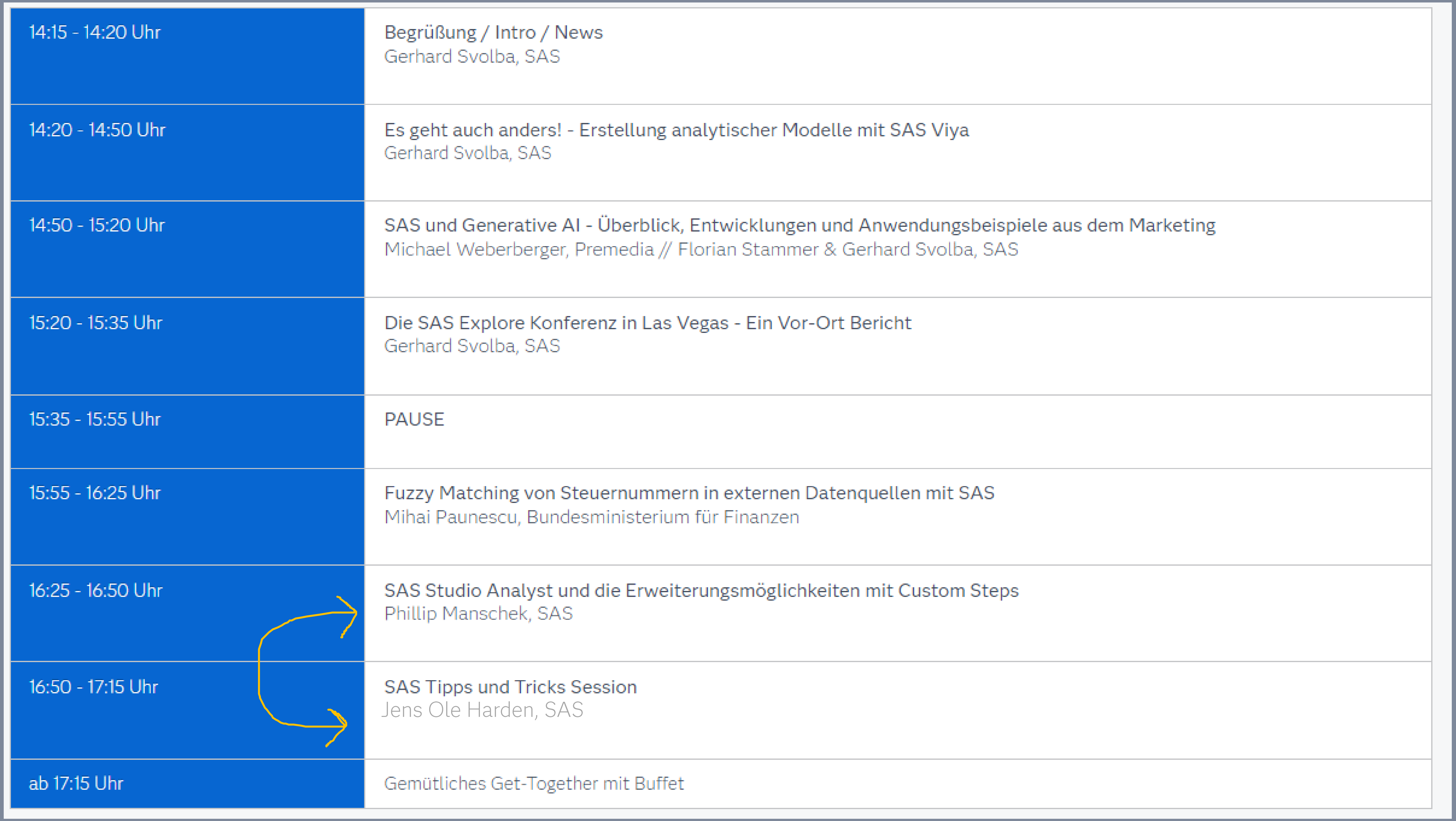
SAS Club 2023

Der Business Analytics Club für SAS User

Wien, SAS Office Trabrennstraße
19. Oktober 2023

Gerhard Svolba, Phillip Manschek, Jens-Ole Harden,
Michael Weberberger (Premedia), Florian Stammer

Agenda



14:15 - 14:20 Uhr	Begrüßung / Intro / News Gerhard Svolba, SAS
14:20 - 14:50 Uhr	Es geht auch anders! - Erstellung analytischer Modelle mit SAS Viya Gerhard Svolba, SAS
14:50 - 15:20 Uhr	SAS und Generative AI - Überblick, Entwicklungen und Anwendungsbeispiele aus dem Marketing Michael Weberberger, Premedia // Florian Stammer & Gerhard Svolba, SAS
15:20 - 15:35 Uhr	Die SAS Explore Konferenz in Las Vegas - Ein Vor-Ort Bericht Gerhard Svolba, SAS
15:35 - 15:55 Uhr	PAUSE
15:55 - 16:25 Uhr	Fuzzy Matching von Steuernummern in externen Datenquellen mit SAS Mihai Paunescu, Bundesministerium für Finanzen
16:25 - 16:50 Uhr	SAS Studio Analyst und die Erweiterungsmöglichkeiten mit Custom Steps Phillip Manschek, SAS
16:50 - 17:15 Uhr	SAS Tipps und Tricks Session Jens Ole Harden, SAS
ab 17:15 Uhr	Gemütliches Get-Together mit Buffet

Es geht auch anders! - Erstellung analytischer Modelle mit SAS Viya

Gerhard Svolba

SAS hat keine analytischen Modelle „ out-of-the-box“. Man muss die Modelle immer selbst entwickeln.

Nur SAS-Programmierer können analytische Modelle in SAS entwickeln.

Für automatisches Feature Engineering muss ich nach Open-Source wechseln.

Analytische Ergebnisse im SAS mögen zwar richtig sein. Graphisch kann man sie kaum dem Fachbereich präsentieren.

Ja, sie haben die Möglichkeit, Modelle in SAS zu entwickeln.
(Wenn sie wirklich Black-Box Modelle haben wollen, können wir das auch).

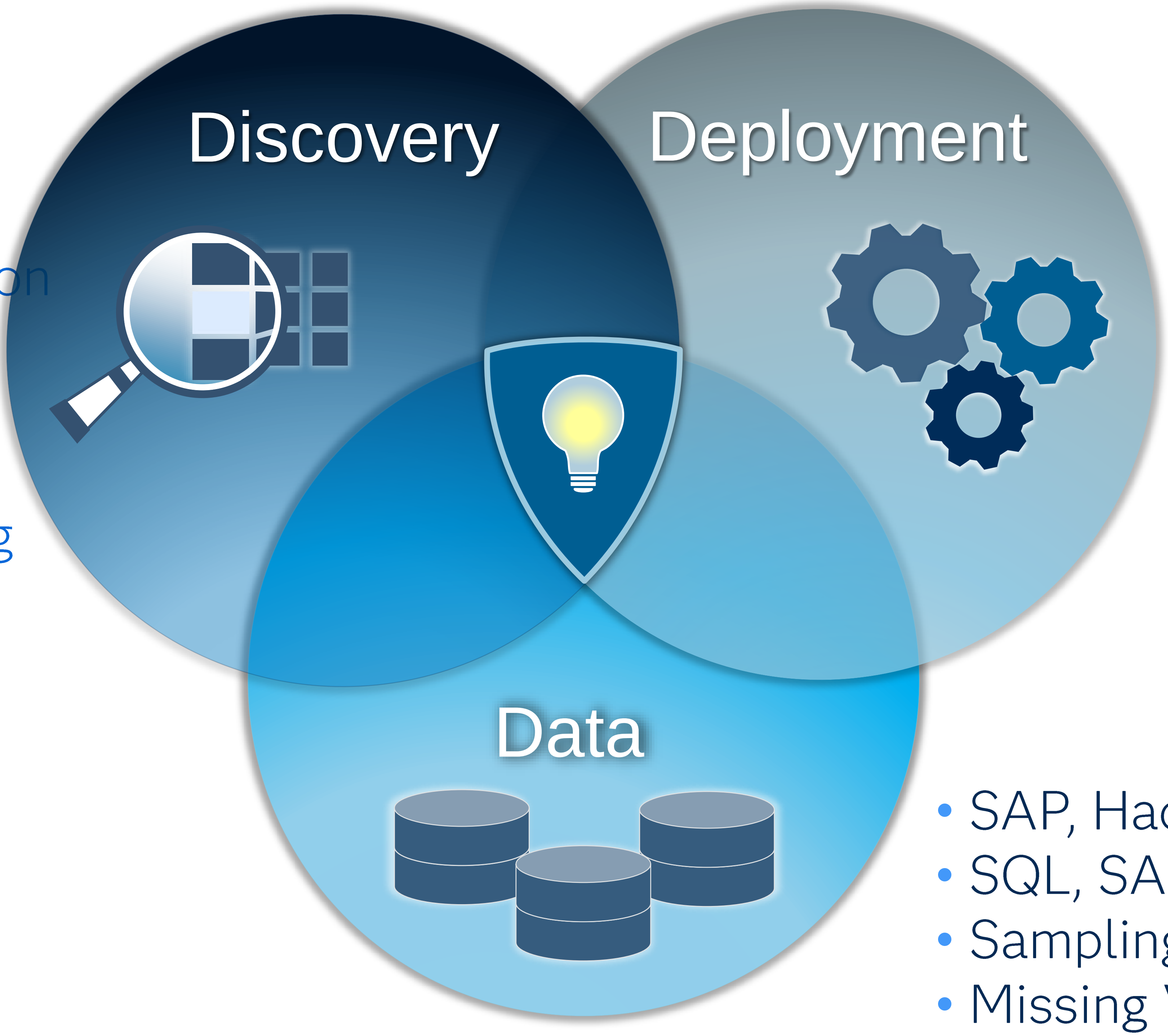
Ja, als SAS Programmierer haben Sie auch mit SAS Viya die Möglichkeit ihre Modelle mit SAS Code zu programmieren.
(das ist aber bei weitem nicht die einzige Möglichkeit SAS Modelle zu erstellen).

Ja, sie können Open Source Modelle und Methoden integrieren. Sie können auch in Open-Source Sprache (Python und R) mit dem SAS Server sprechen.
Sie müssen aber nicht: Moderne Machine Learning Methoden (Gradient Boosting, Random Forecast, NN, NLPs, Feature Selection, Feature Machines, …) sind integraler Teil von SAS Viya.

TD3_FinalModels_v1
Frequency of DemGender
DemGender
Frequency
Frequency of DemAffl
Frequency of DemAge
DemAge
Recommendations
Knoten-ID: 15
Anzahl: 823
DemGender: F
DemAge: ≥ 42 oder fehlend
DemAffl: ≥ 15
0: 163 (19.81%)
1: 660 (80.19%)

Data Mining und Machine Learning mit der SAS Platform

- Logistic Regression
- Linear Regression
- Generalized Linear Models
- Nonlinear Regression
- Ordinary Least Squares Regression
- Decision Trees
- Partial Least Squares Regression
- Quantile Regression
- K-means and K-modes Clustering
- Principal Component Analysis
- Random Forest
- Gradient Boosting
- Neural Networks
- Support Vector Machines
- Factorization Machines
- Network Analytics/Community Detection
- Text Mining
- Boolean Rules
- Auto-tuned Hyper-parameters



- Assess Supervised Models
- Model Management
- Deployment
- Periodic Validation
- Model-Retirement
- Retraining of Models

- SAP, Hadoop, Streaming, rel.DB, …
- SQL, SAS Datastep, Matrix
- Sampling and Partitioning
- Missing Value Imputation
- Variable Binning
- Variable Selection
- Transpose

Möglichkeiten der Interaktion mit der SAS Analytik Plattform

Graphische Benutzeroberfläche

Visuelle Oberfläche

Self-Service Analytik-Objekte

Integration mit Model Studio & Model Manager

Model Studio

Pipelines und Knoten, Feature-Engineerung, Optionen, Tuning, Open Source Integration, Integration mit dem Model Manager

Programmierung

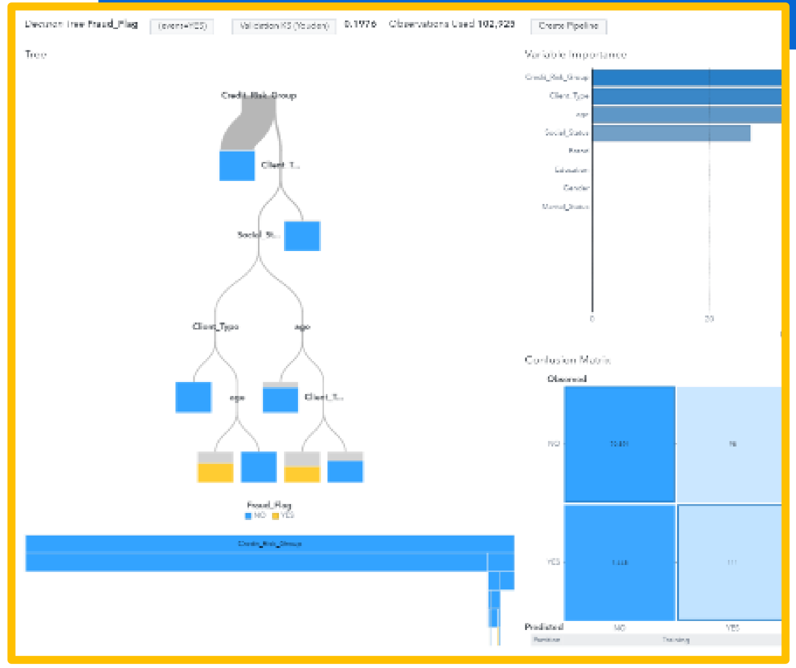
SAS

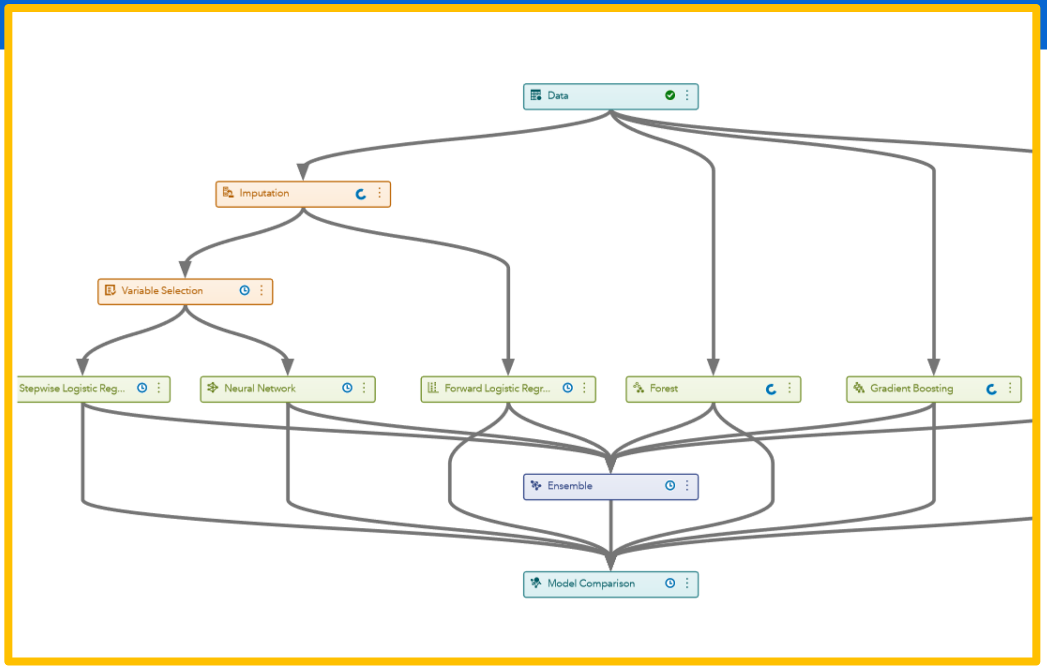
Volle Flexibilität bei der Programmierung in der SAS Language (Procedures, Actions, Functionen, …)

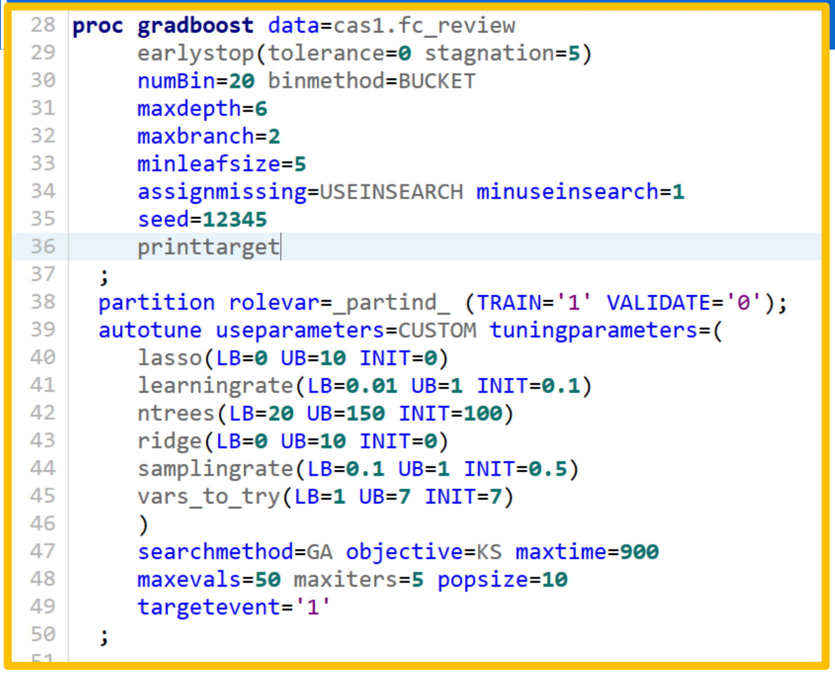
Open Source Integration

Open Source Sprache

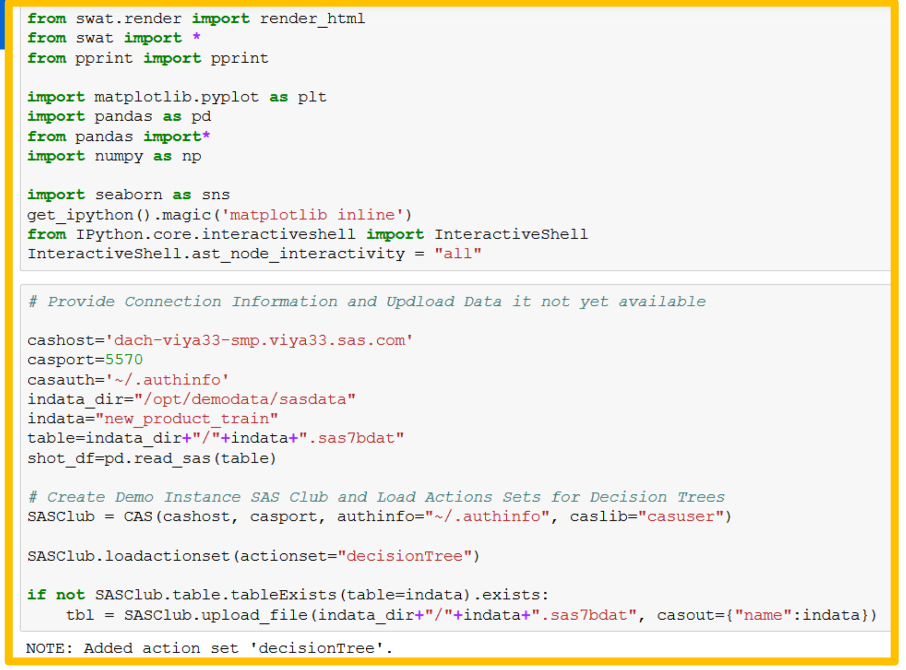
Interaktion mit der SAS Analytik-Plattform aus dem Jupyter-Notebook oder R-Studio heraus







```
proc gradboost data=cas1.fc_review
     earlystop(tolerance=0 stagnation=5)
     numBin=20 binmethod=BUCKET
     maxdepth=6
     maxbranch=2
     minleafsize=5
     assignmissing=USEINSEARCH minuseinsearch=1
     seed=12345
     printtarget
  ;
  partition rolevar=_partind_ (TRAIN='1' VALIDATE='0');
  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=7 INIT=7)
     )
     searchmethod=GA objective=KS maxtime=900
     maxevals=50 maxiters=5 popsize=10
     targetevent='1'
  ;
```

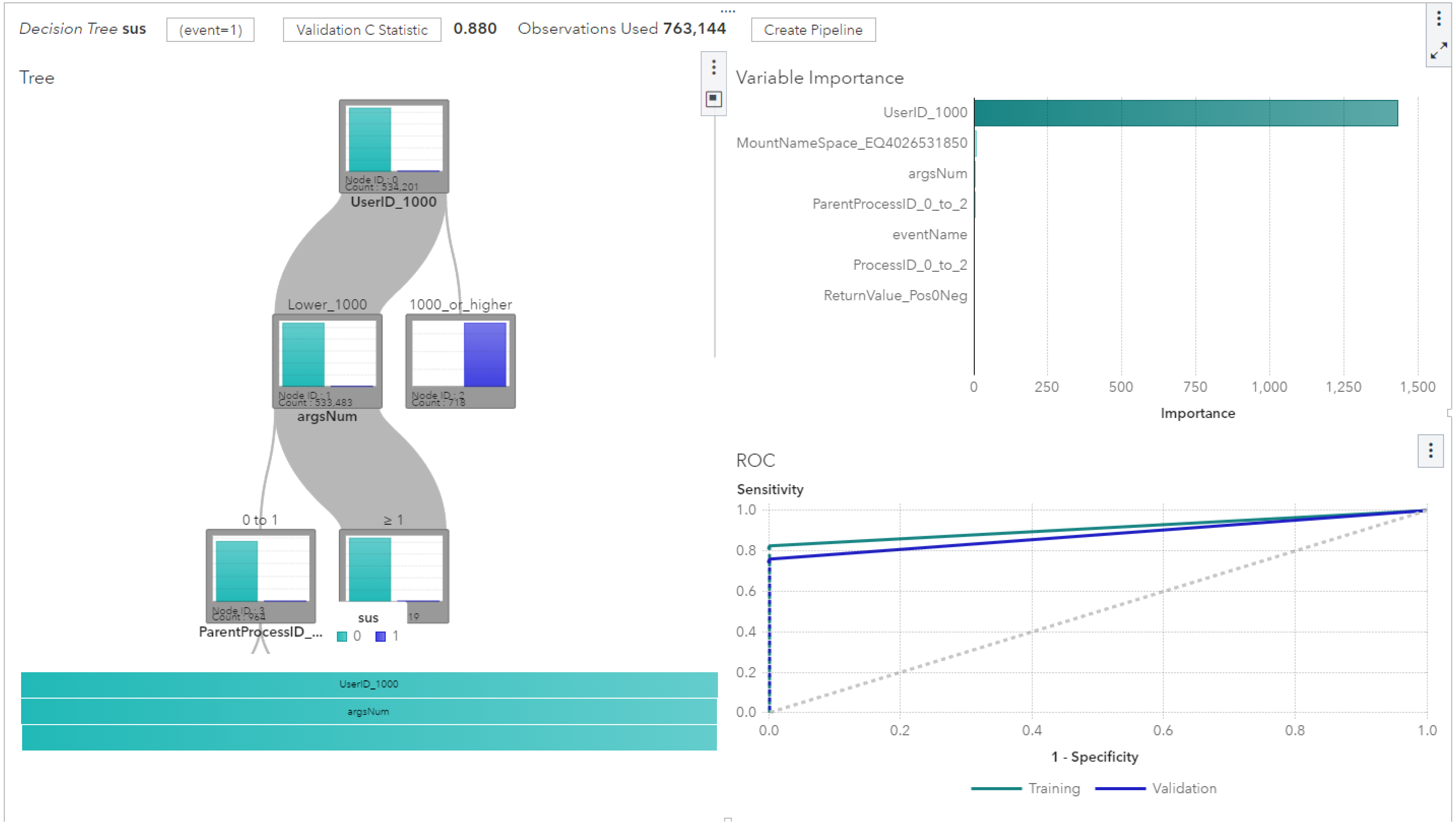


```
from swat.render import render_html
from swat import *
from pprint import pprint

import matplotlib.pyplot as plt
import pandas as pd
from pandas import*
import numpy as np

import seaborn as sns
get_ipython().magic('matplotlib inline')
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

```
# Provide Connection Information and Updload Data it not yet available

cashost='dach-viya33-smp.viya33.sas.com'
casport=5570
casauth='~/.authinfo'
indata_dir="/opt/demodata/sasdata"
indata="new_product_train"
table=indata_dir+"/"+indata+".sas7bdat"
shot_df=pd.read_sas(table)

# Create Demo Instance SAS Club and Load Actions Sets for Decision Trees
SASClub = CAS(cashost, casport, authinfo="~/.authinfo", caslib="casuser")

SASClub.loadactionset(actionset="decisionTree")

if not SASClub.table.tableExists(table=indata).exists:
    tbl = SASClub.upload_file(indata_dir+"/"+indata+".sas7bdat", casout={"name":indata})
```

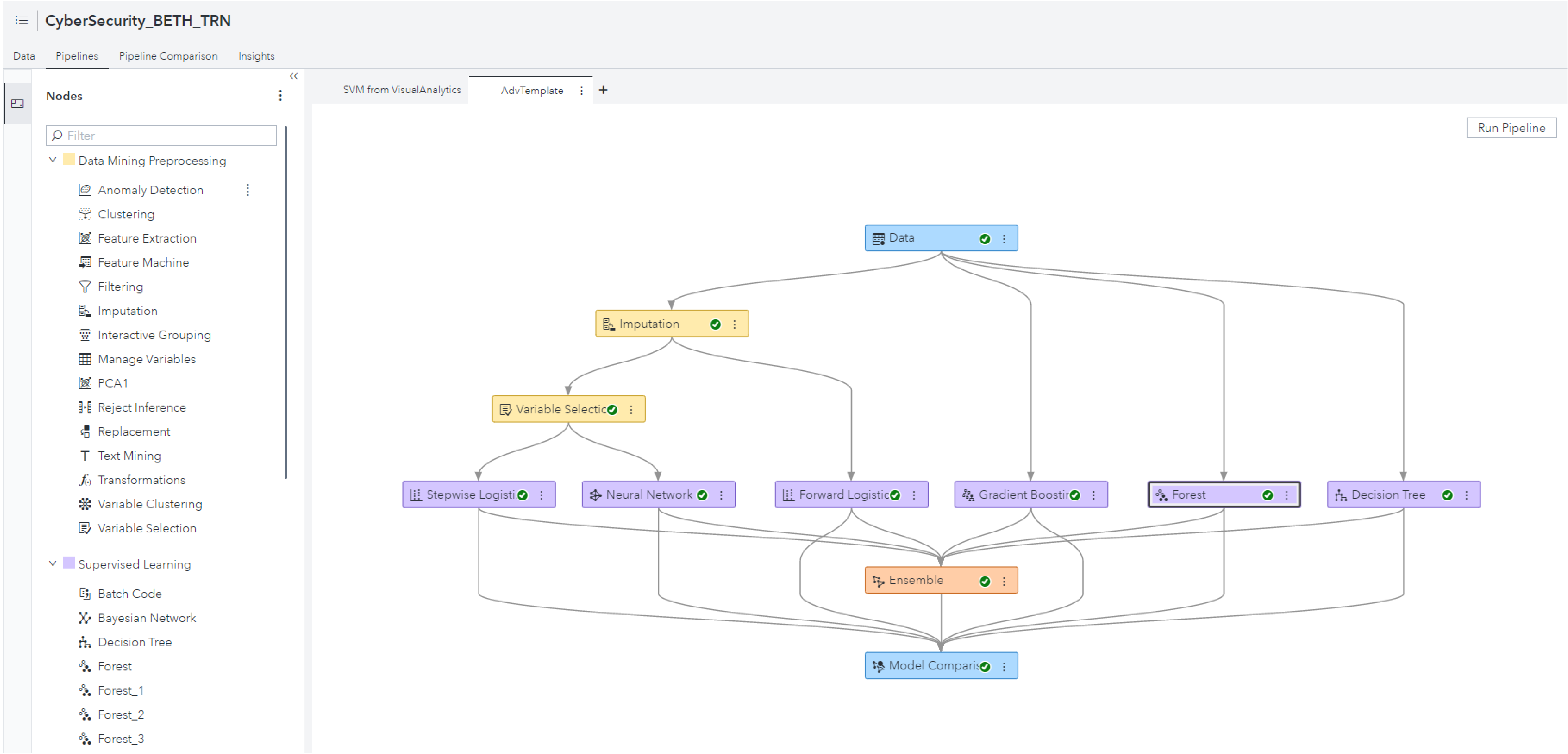
```
NOTE: Added action set 'decisionTree'.
```

Machine Learning Objects in SAS Visual Analytics



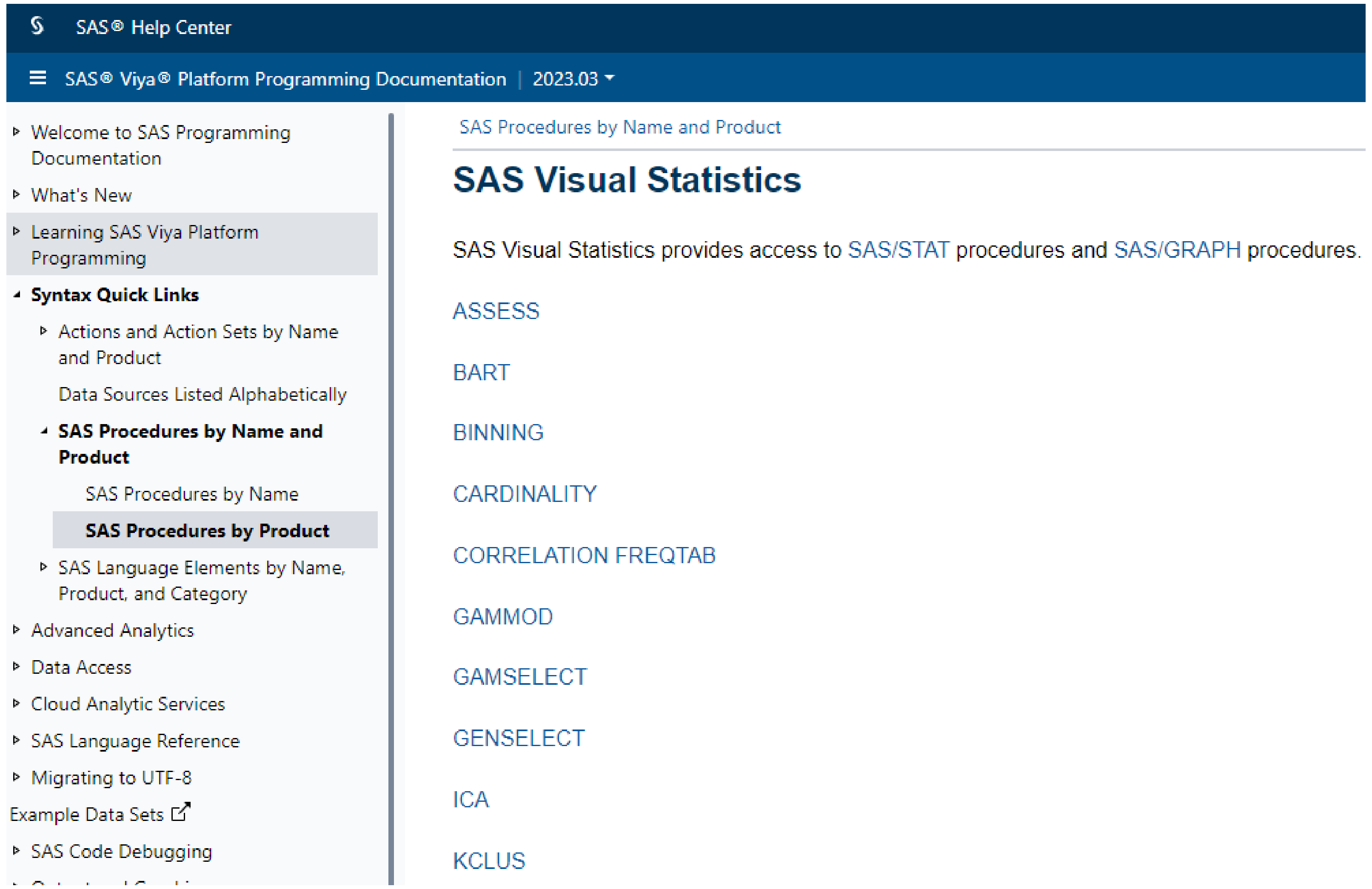
Decision Tree **sus** (event=1) Validation C Statistic **0.880** Observations Used **763,144** Create Pipeline

ML Pipelines in SAS Model Studio



Ausgewählte SAS Machine Learning Procedures in SAS Viya

- SAS Visual Statistics



SAS® Help Center

☰ SAS® Viya® Platform Programming Documentation | 2023.03 ▾

- Welcome to SAS Programming Documentation
- What's New
- Learning SAS Viya Platform Programming
- **Syntax Quick Links**
  - Actions and Action Sets by Name and Product
  - Data Sources Listed Alphabetically
  - **SAS Procedures by Name and Product**
    - SAS Procedures by Name
    - **SAS Procedures by Product**
  - SAS Language Elements by Name, Product, and Category
- Advanced Analytics
- Data Access
- Cloud Analytic Services
- SAS Language Reference
- Migrating to UTF-8
- Example Data Sets
- SAS Code Debugging

SAS Procedures by Name and Product

SAS Visual Statistics

SAS Visual Statistics provides access to SAS/STAT procedures and SAS/GRAPH procedures.

ASSESS

BART

BINNING

CARDINALITY

CORRELATION FREQTAB

GAMMOD

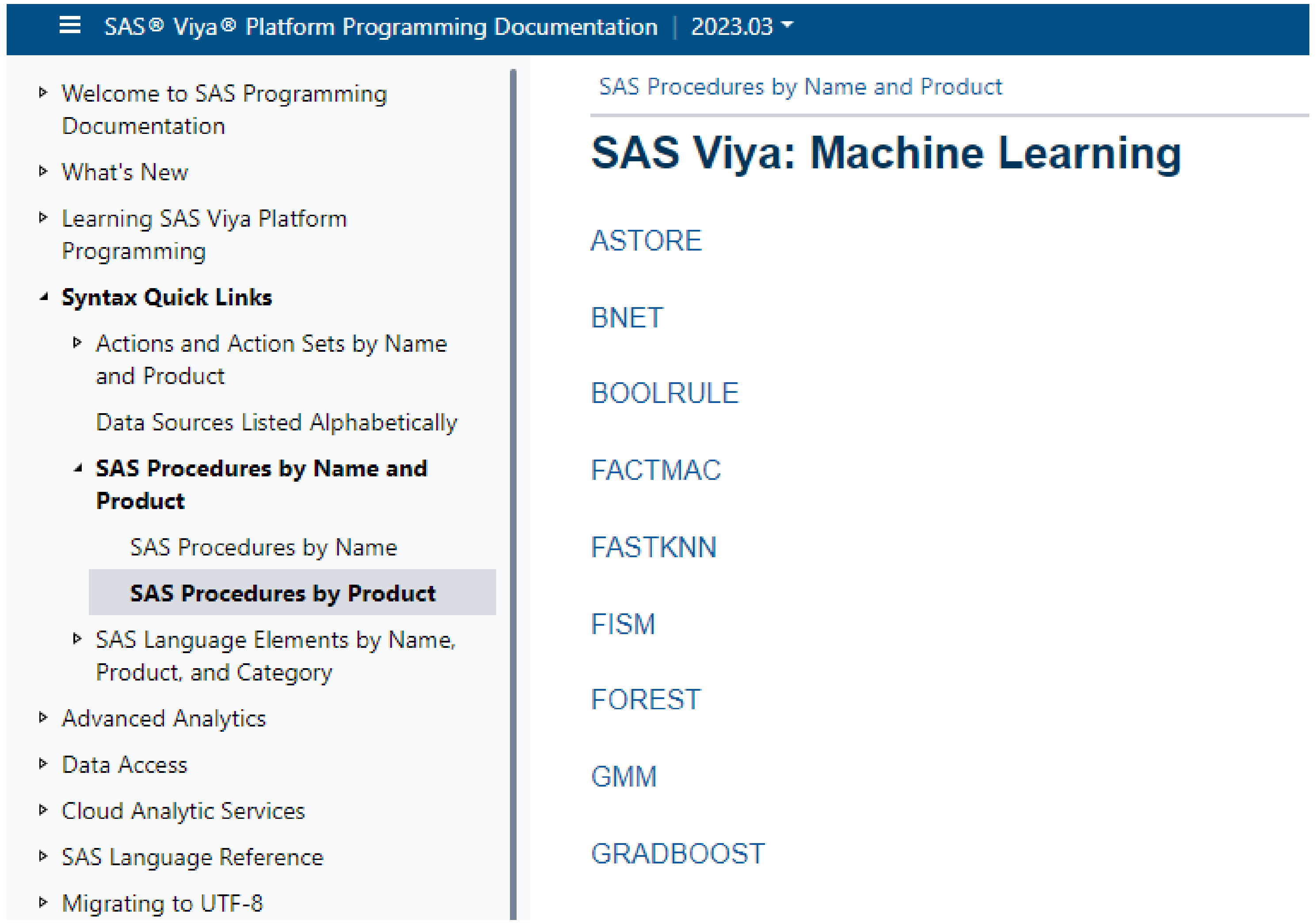
GAMSELECT

GENSELECT

ICA

KCLUS

- SAS Viya: Machine Learning



☰ SAS® Viya® Platform Programming Documentation | 2023.03 ▾

- Welcome to SAS Programming Documentation
- What's New
- Learning SAS Viya Platform Programming
- **Syntax Quick Links**
  - Actions and Action Sets by Name and Product
  - Data Sources Listed Alphabetically
  - **SAS Procedures by Name and Product**
    - SAS Procedures by Name
    - **SAS Procedures by Product**
  - SAS Language Elements by Name, Product, and Category
- Advanced Analytics
- Data Access
- Cloud Analytic Services
- SAS Language Reference
- Migrating to UTF-8

SAS Procedures by Name and Product

SAS Viya: Machine Learning

ASTORE

BNET

BOOLRULE

FACTMAC

FASTKNN

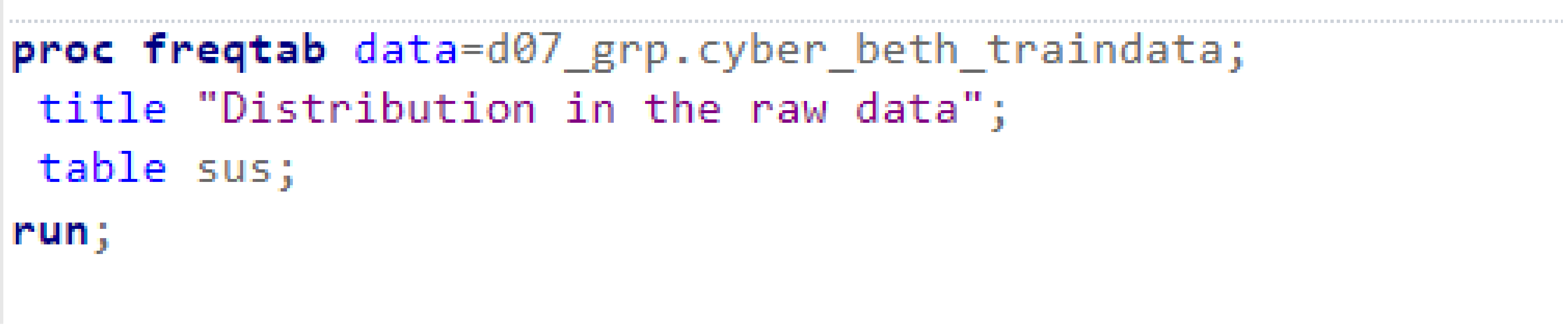
FISM

FOREST

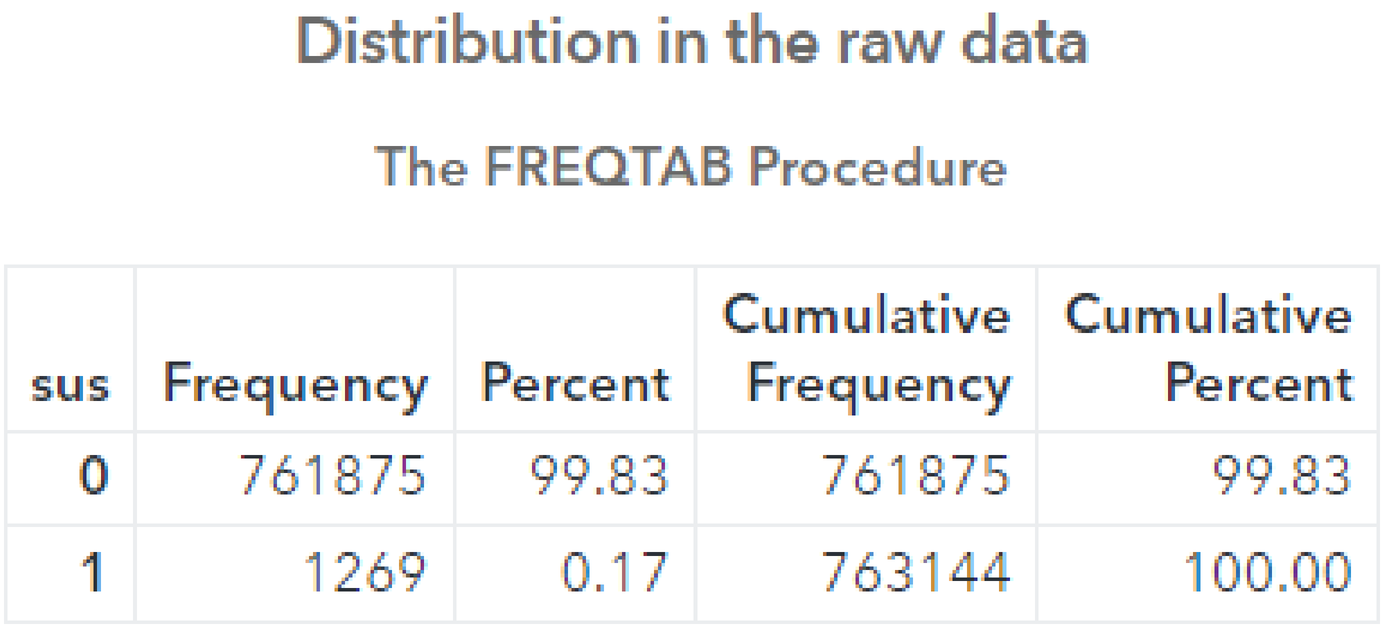
GMM

GRADBOOST

Oversampling the event from 0.17% to to 5% using PROC PARTITION



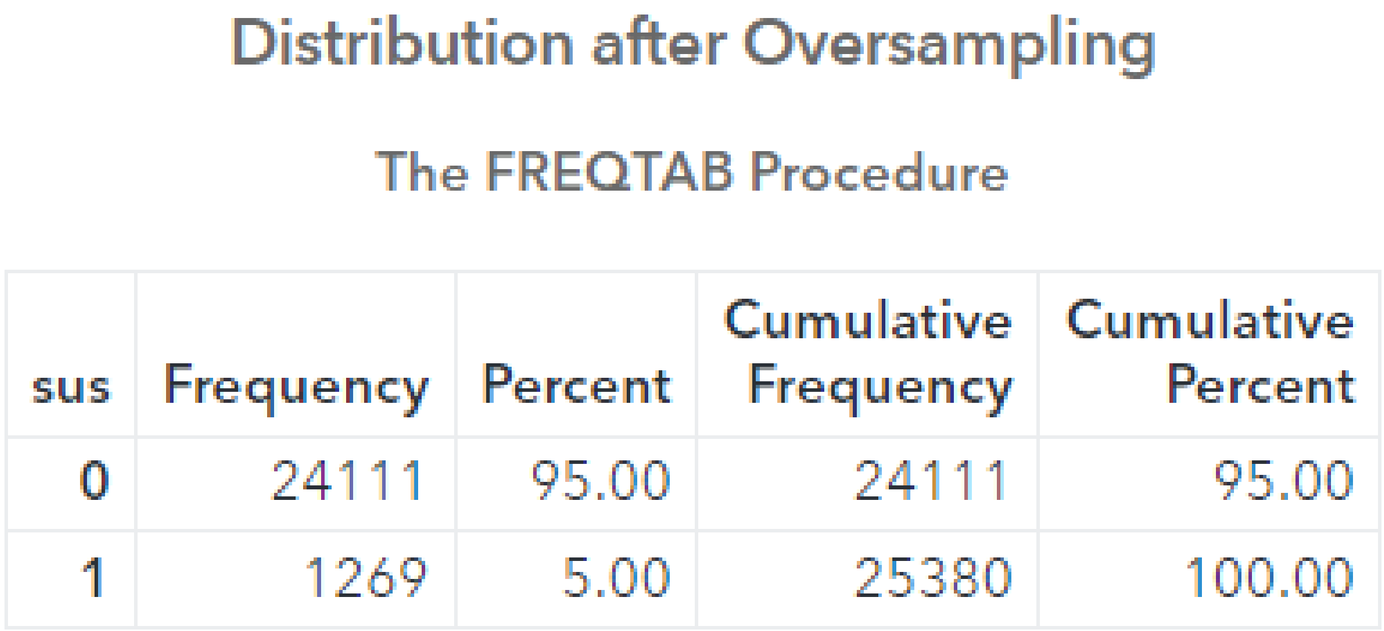
```
proc freqtab data=d07_grp.cyber_beth_traindata;
 title "Distribution in the raw data";
 table sus;
run;
```



Distribution in the raw data

The FREQTAB Procedure

sus	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	761875	99.83	761875	99.83
1	1269	0.17	763144	100.00

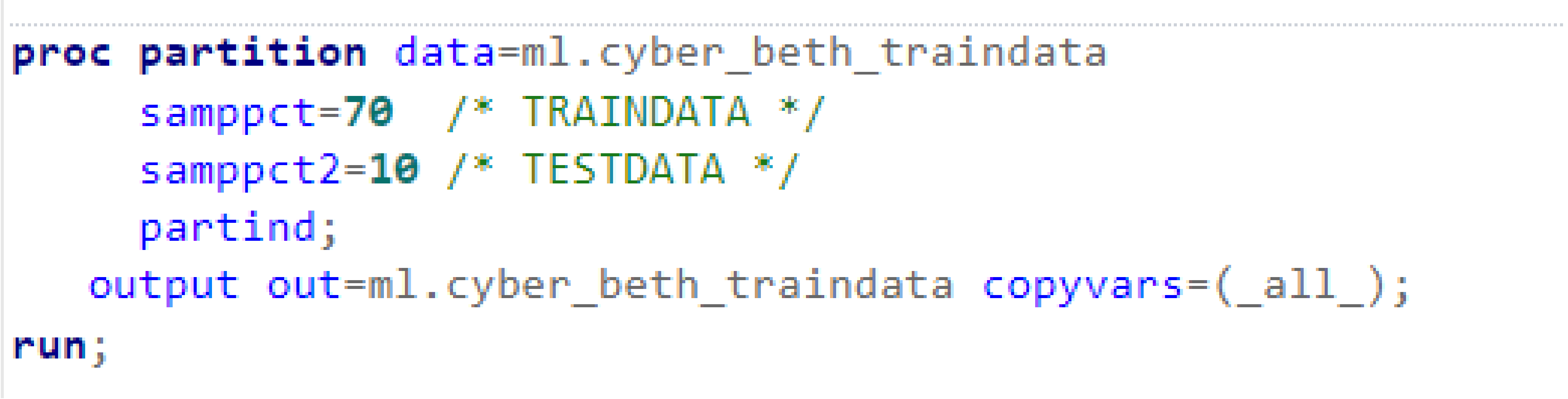


Distribution after Oversampling

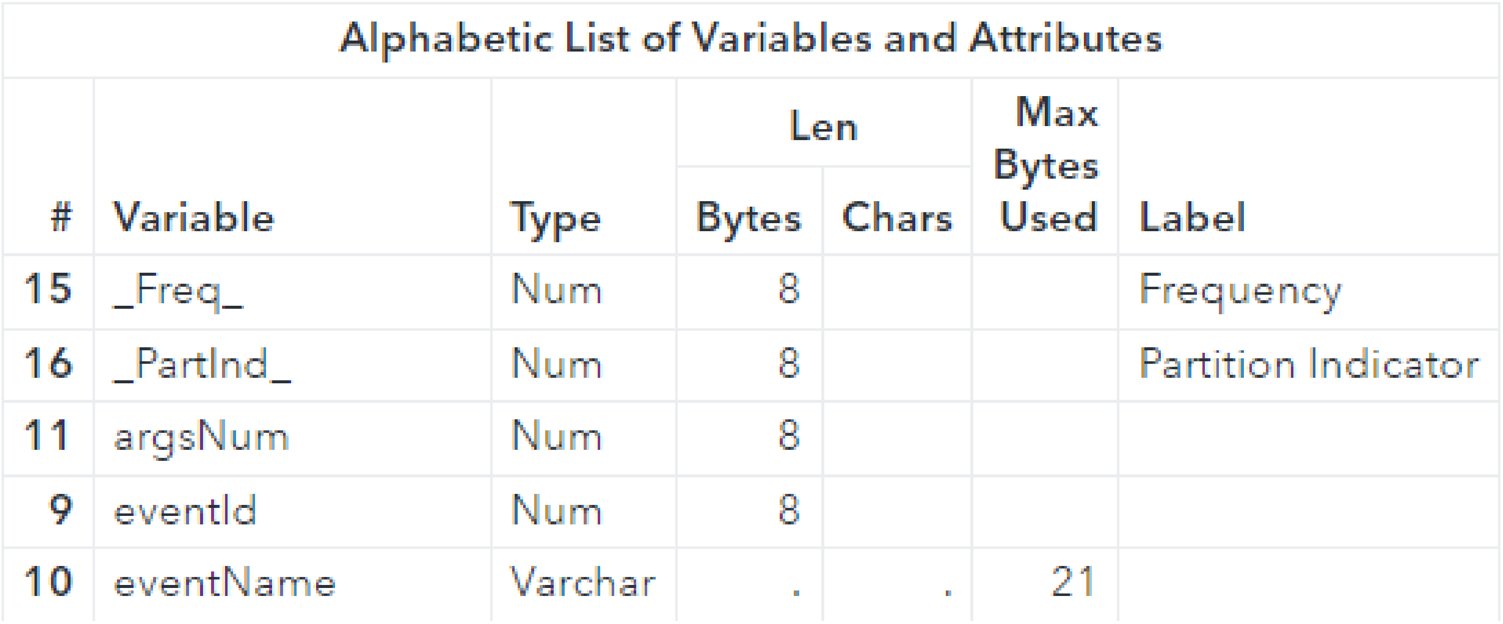
The FREQTAB Procedure

sus	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	24111	95.00	24111	95.00
1	1269	5.00	25380	100.00

Data Partition in 70/20/10 Splits using PROC PARTITION

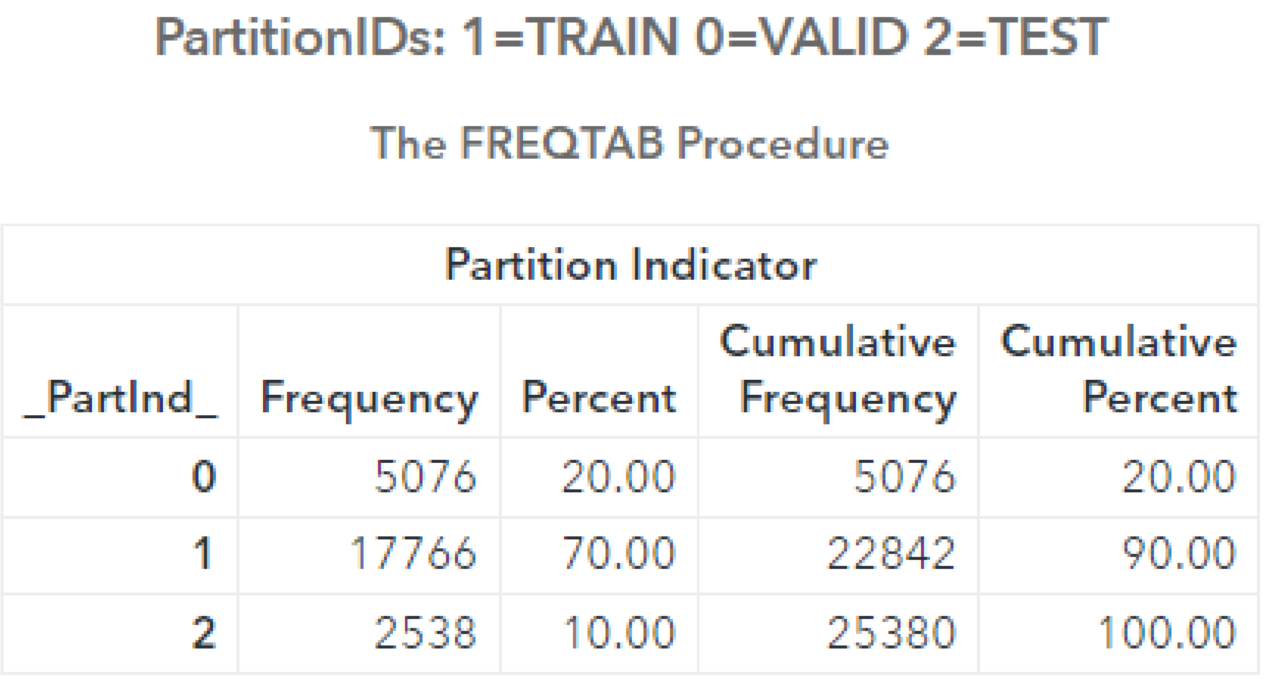


```
proc partition data=ml.cyber_beth_traindata
     samppct=70  /* TRAINDATA */
     samppct2=10 /* TESTDATA */
     partind;
   output out=ml.cyber_beth_traindata copyvars=(_all_);
run;
```



Alphabetic List of Variables and Attributes

#	Variable	Type	Len		Max Bytes Used	Label
			Bytes	Chars		
15	_Freq_	Num	8			Frequency
16	_PartInd_	Num	8			Partition Indicator
11	argsNum	Num	8			
9	eventId	Num	8			
10	eventName	Varchar	.	.	21	



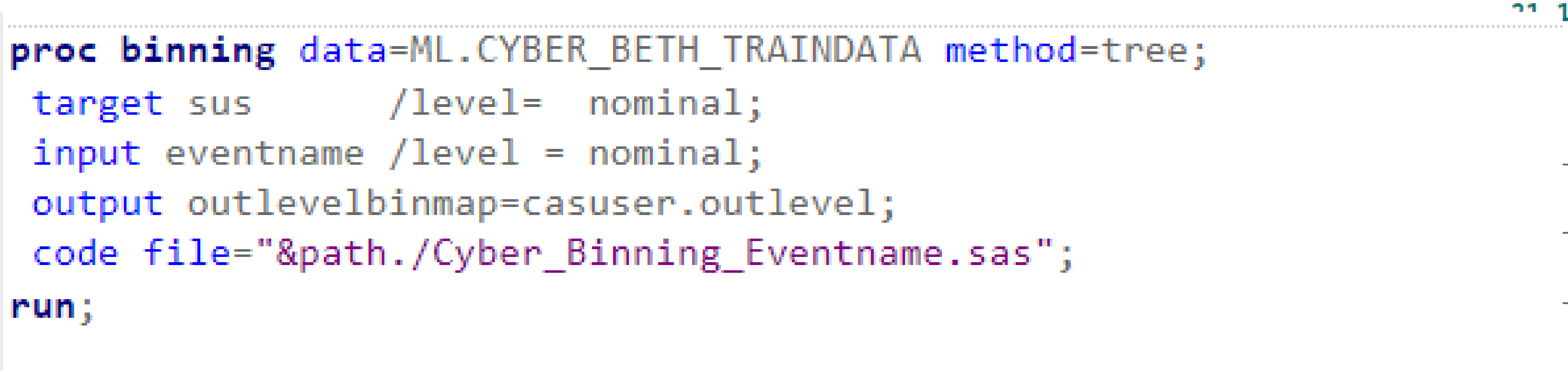
PartitionIDs: 1=TRAIN 0=VALID 2=TEST

The FREQTAB Procedure

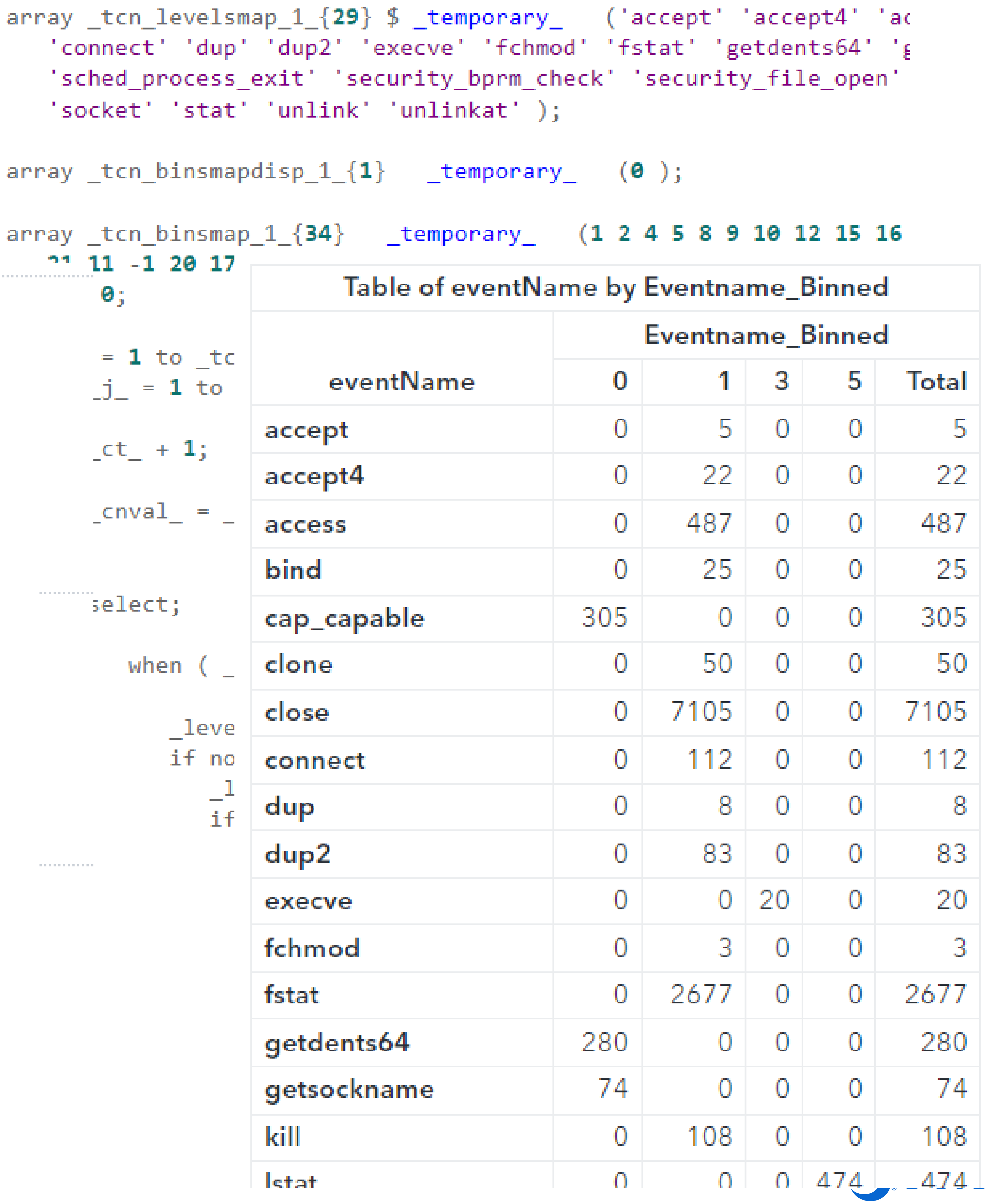
Partition Indicator

PartInd	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	5076	20.00	5076	20.00
1	17766	70.00	22842	90.00
2	2538	10.00	25380	100.00

Target-based BINNING



```
proc binning data=ML.CYBER_BETH_TRAINDATA method=tree;
 target sus      /level=  nominal;
 input eventname /level = nominal;
 output outlevelbinmap=casuser.outlevel;
 code file="&path./Cyber_Binning_Eventname.sas";
run;
```

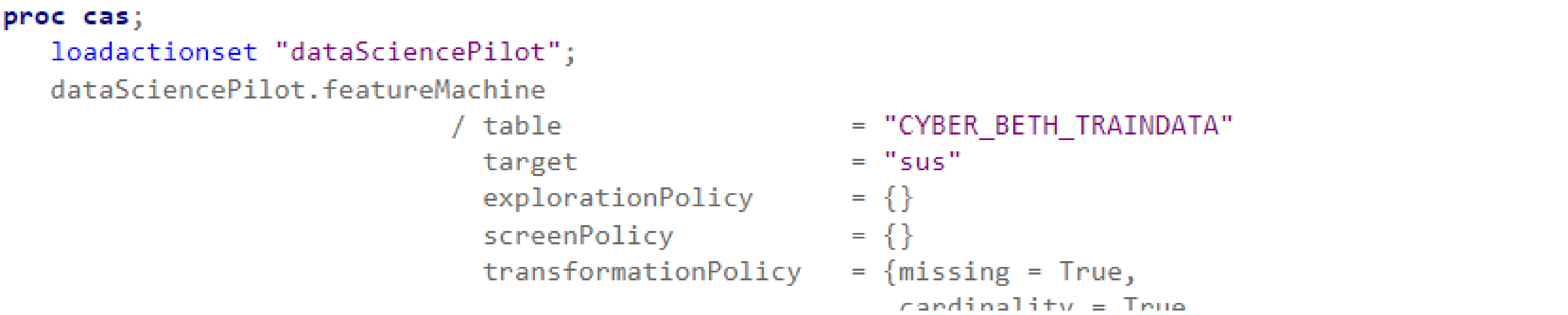


```
array _tcn_levelsmap_1_{29} $ _temporary_   ('accept' 'accept4' 'ac
   'connect' 'dup' 'dup2' 'execve' 'fchmod' 'fstat' 'getdents64' 'g
   'sched_process_exit' 'security_bprm_check' 'security_file_open'
   'socket' 'stat' 'unlink' 'unlinkat' );

array _tcn_binsmapdisp_1_{1}  _temporary_   (0 );

array _tcn_binsmap_1_{34}  _temporary_   (1 2 4 5 8 9 10 12 15 16
   21 11 -1 20 17
       0;

       = 1 to _tc
      _j_ = 1 to

      _ct_ + 1;

      _cnval_ = _


      select;

        when ( _

          _leve
          if no
           _l
           if
```

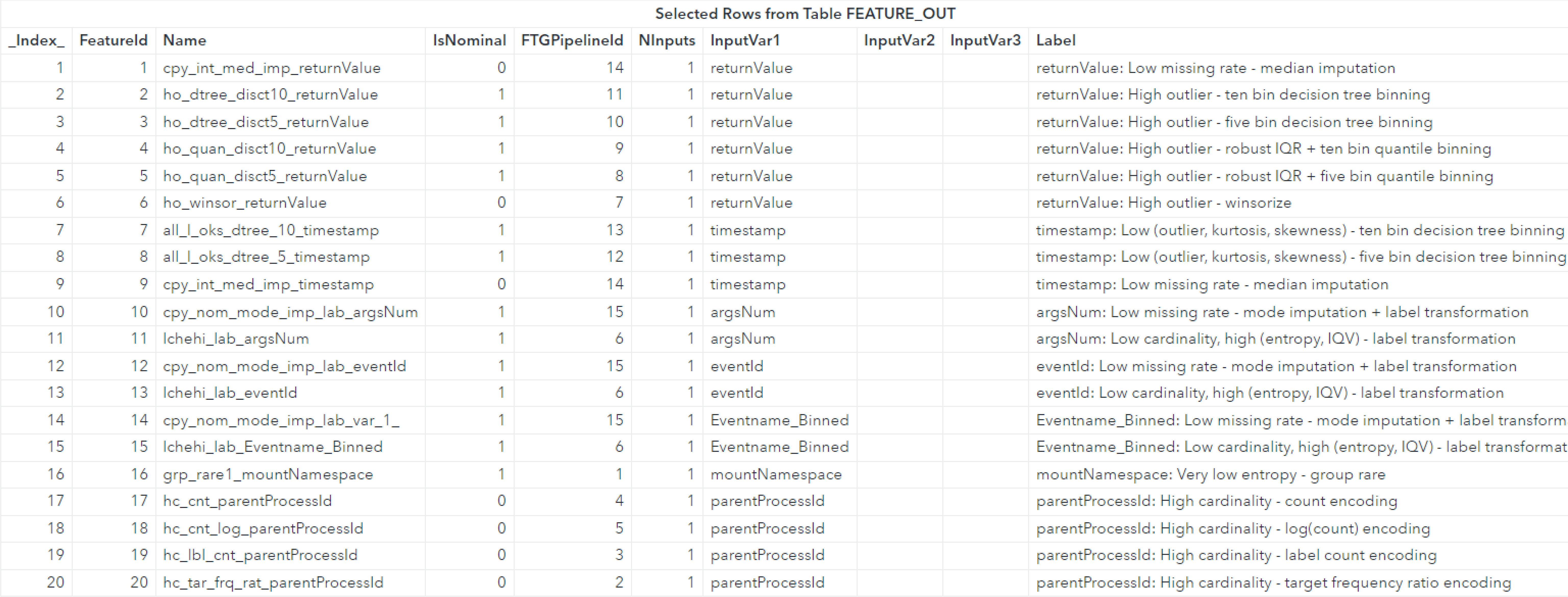
Table of eventName by Eventname_Binned

	Eventname_Binned				
eventName	0	1	3	5	Total
accept	0	5	0	0	5
accept4	0	22	0	0	22
access	0	487	0	0	487
bind	0	25	0	0	25
cap_capable	305	0	0	0	305
clone	0	50	0	0	50
close	0	7105	0	0	7105
connect	0	112	0	0	112
dup	0	8	0	0	8
dup2	0	83	0	0	83
execve	0	0	20	0	20
fchmod	0	3	0	0	3
fstat	0	2677	0	0	2677
getdents64	280	0	0	0	280
getsockname	74	0	0	0	74
kill	0	108	0	0	108
lstat	0	0	0	474	474

Feature Generation and Transformation mit dem Action-Set DataSciencePilot



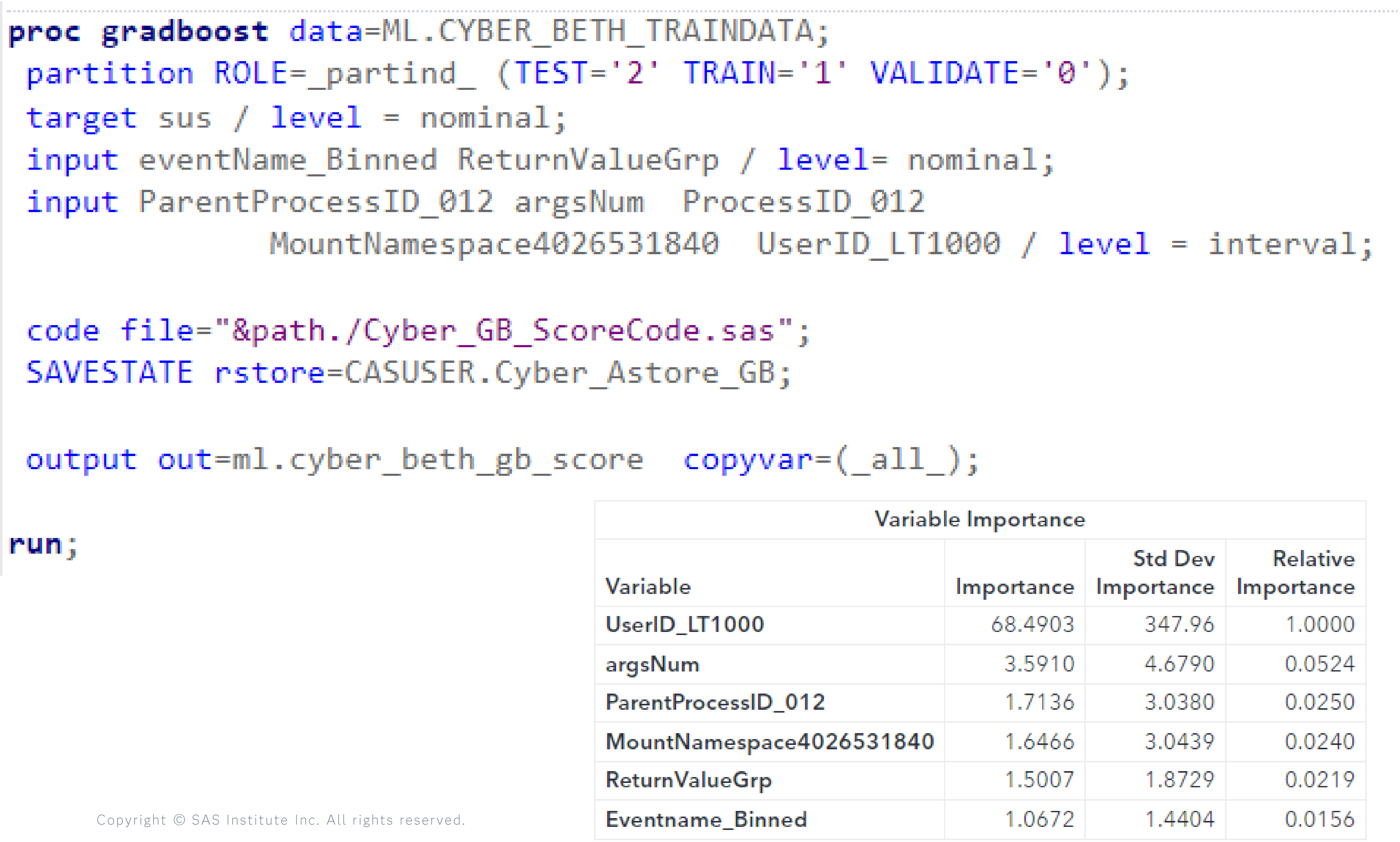
```
proc cas;
   loadactionset "dataSciencePilot";
   dataSciencePilot.featureMachine
                              / table                  = "CYBER_BETH_TRAINDATA"
                                target                 = "sus"
                                explorationPolicy      = {}
                                screenPolicy           = {}
                                transformationPolicy   = {missing = True,
                                                          cardinality = True

                                tra

                                fea

                                ca

                                sav

                                ;
```



Selected Rows from Table FEATURE_OUT

Index	FeatureId	Name	IsNominal	FTGPipelineId	NInputs	InputVar1	InputVar2	InputVar3	Label
1	1	cpy_int_med_imp_returnValue	0	14	1	returnValue			returnValue: Low missing rate - median imputation
2	2	ho_dtree_disct10_returnValue	1	11	1	returnValue			returnValue: High outlier - ten bin decision tree binning
3	3	ho_dtree_disct5_returnValue	1	10	1	returnValue			returnValue: High outlier - five bin decision tree binning
4	4	ho_quan_disct10_returnValue	1	9	1	returnValue			returnValue: High outlier - robust IQR + ten bin quantile binning
5	5	ho_quan_disct5_returnValue	1	8	1	returnValue			returnValue: High outlier - robust IQR + five bin quantile binning
6	6	ho_winsor_returnValue	0	7	1	returnValue			returnValue: High outlier - winsorize
7	7	all_l_oks_dtree_10_timestamp	1	13	1	timestamp			timestamp: Low (outlier, kurtosis, skewness) - ten bin decision tree binning
8	8	all_l_oks_dtree_5_timestamp	1	12	1	timestamp			timestamp: Low (outlier, kurtosis, skewness) - five bin decision tree binning
9	9	cpy_int_med_imp_timestamp	0	14	1	timestamp			timestamp: Low missing rate - median imputation
10	10	cpy_nom_mode_imp_lab_argsNum	1	15	1	argsNum			argsNum: Low missing rate - mode imputation + label transformation
11	11	lchehi_lab_argsNum	1	6	1	argsNum			argsNum: Low cardinality, high (entropy, IQV) - label transformation
12	12	cpy_nom_mode_imp_lab_eventId	1	15	1	eventId			eventId: Low missing rate - mode imputation + label transformation
13	13	lchehi_lab_eventId	1	6	1	eventId			eventId: Low cardinality, high (entropy, IQV) - label transformation
14	14	cpy_nom_mode_imp_lab_var_1_	1	15	1	Eventname_Binned			Eventname_Binned: Low missing rate - mode imputation + label transform
15	15	lchehi_lab_Eventname_Binned	1	6	1	Eventname_Binned			Eventname_Binned: Low cardinality, high (entropy, IQV) - label transformat
16	16	grp_rare1_mountNamespace	1	1	1	mountNamespace			mountNamespace: Very low entropy - group rare
17	17	hc_cnt_parentProcessId	0	4	1	parentProcessId			parentProcessId: High cardinality - count encoding
18	18	hc_cnt_log_parentProcessId	0	5	1	parentProcessId			parentProcessId: High cardinality - log(count) encoding
19	19	hc_lbl_cnt_parentProcessId	0	3	1	parentProcessId			parentProcessId: High cardinality - label count encoding
20	20	hc_tar_frq_rat_parentProcessId	0	2	1	parentProcessId			parentProcessId: High cardinality - target frequency ratio encoding

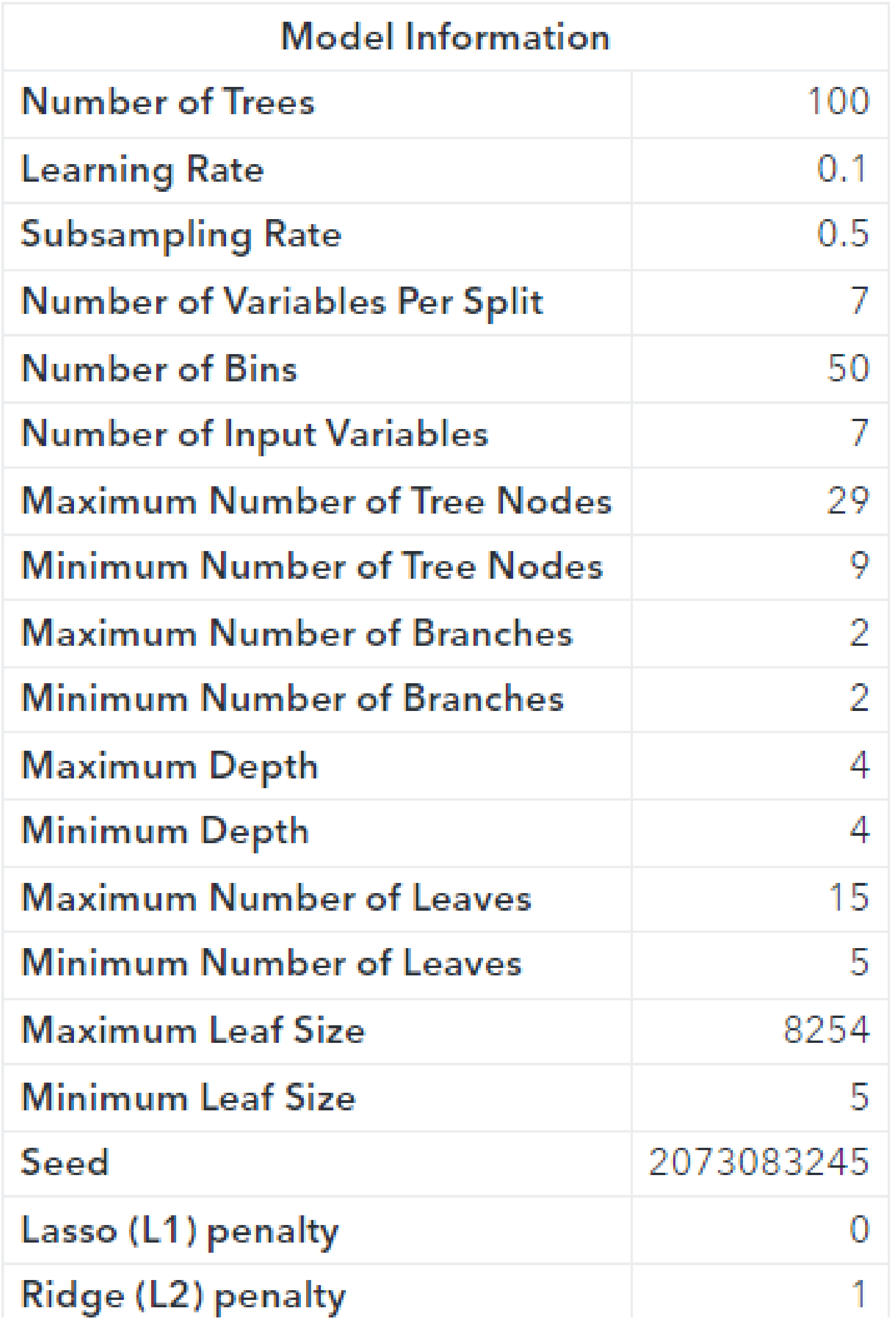
PROC GRADBOOST for gradient boosting models



```
proc gradboost data=ML.CYBER_BETH_TRAINDATA;
 partition ROLE=_partind_ (TEST='2' TRAIN='1' VALIDATE='0');
 target sus / level = nominal;
 input eventName_Binned ReturnValueGrp / level= nominal;
 input ParentProcessID_012 argsNum  ProcessID_012
              MountNamespace4026531840  UserID_LT1000 / level = interval;

 code file="&path./Cyber_GB_ScoreCode.sas";
 SAVESTATE rstore=CASUSER.Cyber_Astore_GB;

 output out=ml.cyber_beth_gb_score  copyvar=(_all_);

run;
```

Variable Importance

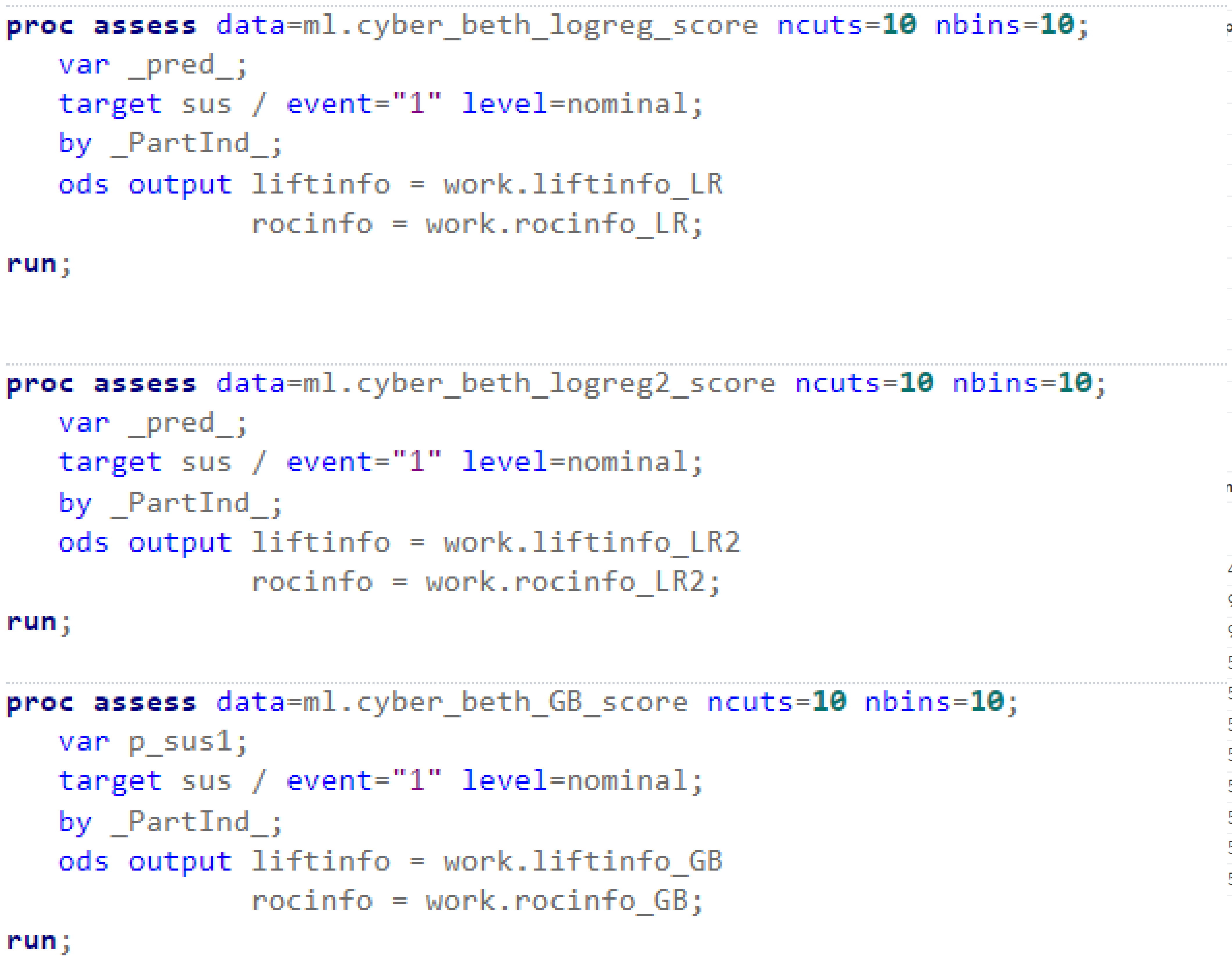
Variable	Importance	Std Dev Importance	Relative Importance
UserID_LT1000	68.4903	347.96	1.0000
argsNum	3.5910	4.6790	0.0524
ParentProcessID_012	1.7136	3.0380	0.0250
MountNamespace4026531840	1.6466	3.0439	0.0240
ReturnValueGrp	1.5007	1.8729	0.0219
Eventname_Binned	1.0672	1.4404	0.0156



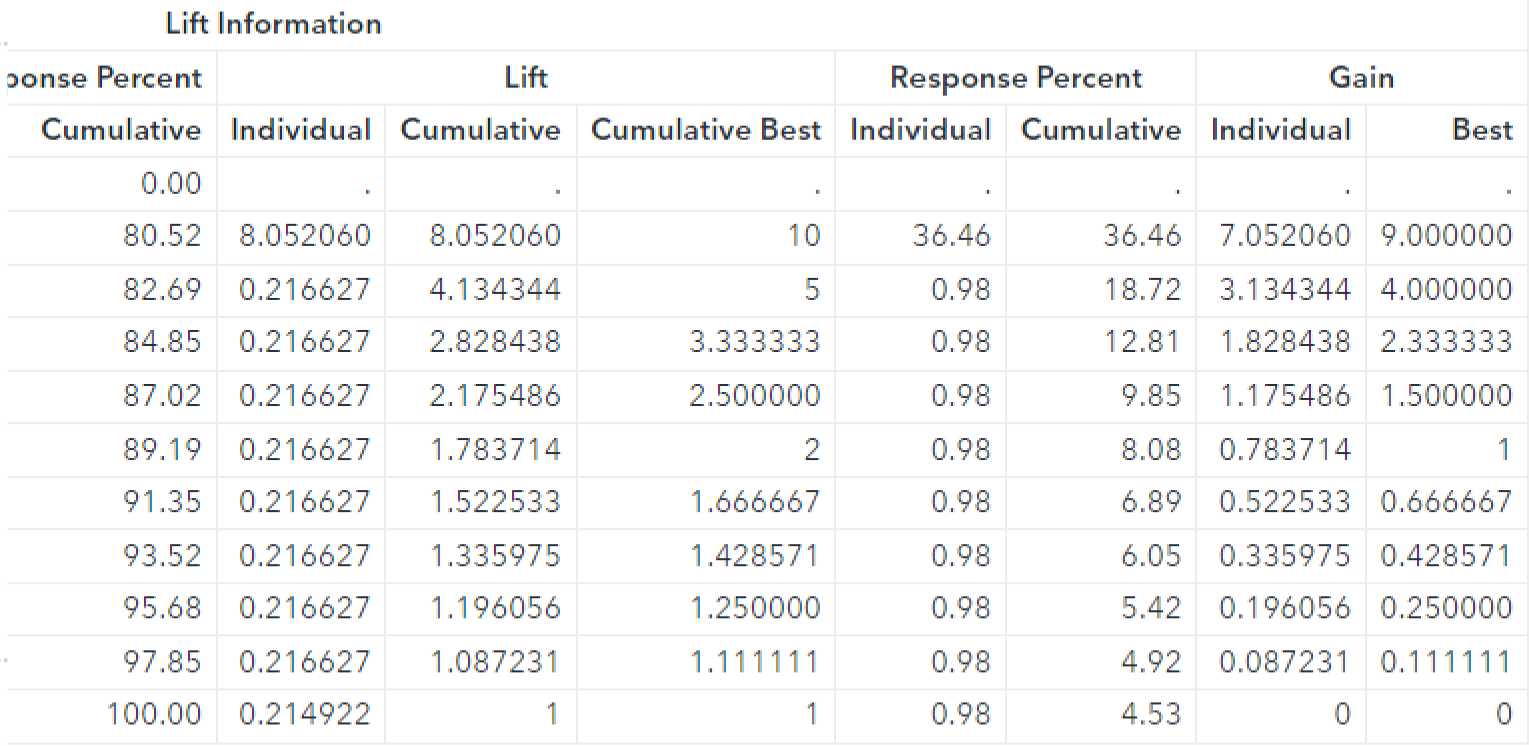
Model Information

Number of Trees	100
Learning Rate	0.1
Subsampling Rate	0.5
Number of Variables Per Split	7
Number of Bins	50
Number of Input Variables	7
Maximum Number of Tree Nodes	29
Minimum Number of Tree Nodes	9
Maximum Number of Branches	2
Minimum Number of Branches	2
Maximum Depth	4
Minimum Depth	4
Maximum Number of Leaves	15
Minimum Number of Leaves	5
Maximum Leaf Size	8254
Minimum Leaf Size	5
Seed	2073083245
Lasso (L1) penalty	0
Ridge (L2) penalty	1

Using PROC ASSESS to calculate lift, ROC and more

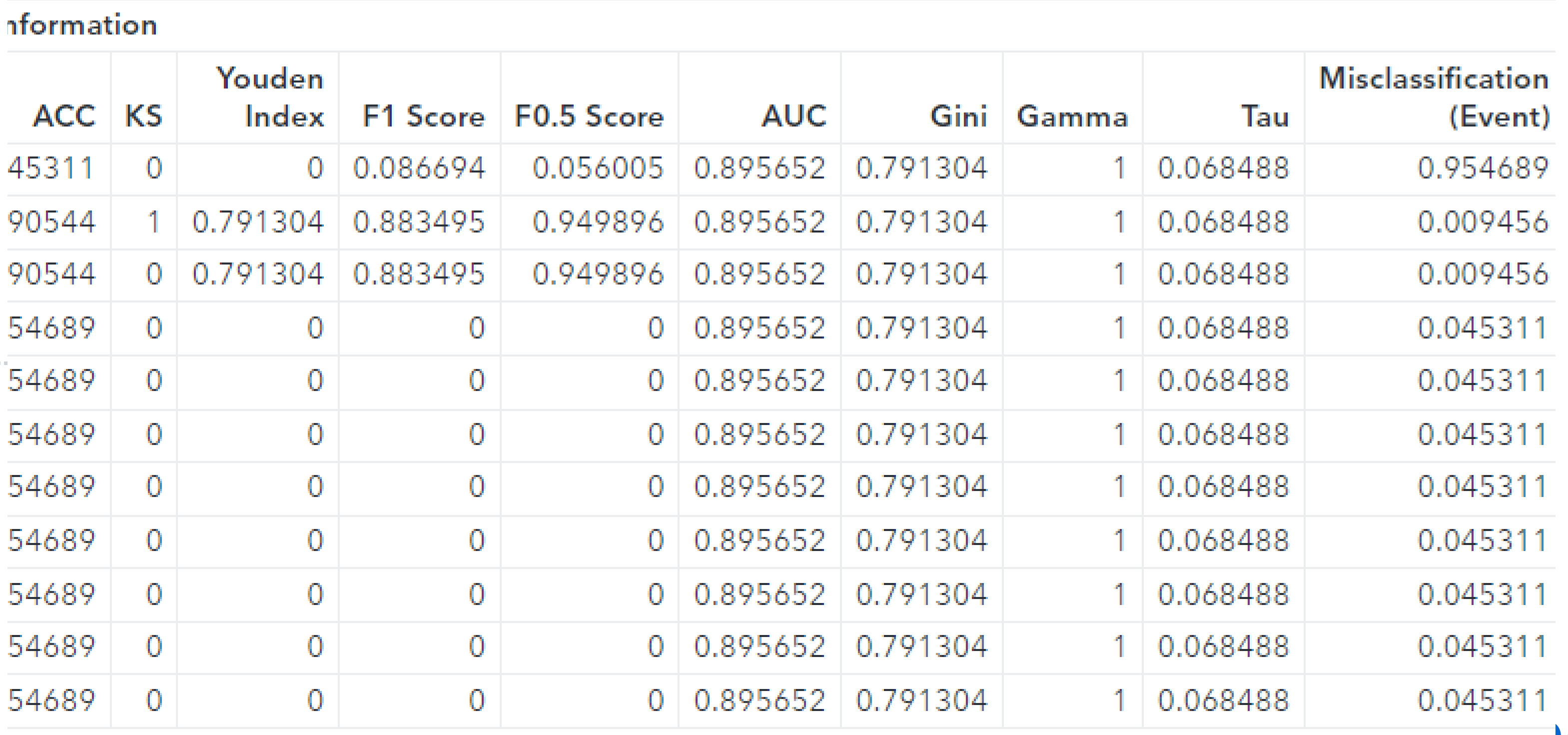


```
proc assess data=ml.cyber_beth_logreg_score ncuts=10 nbins=10;
   var _pred_;
   target sus / event="1" level=nominal;
   by _PartInd_;
   ods output liftinfo = work.liftinfo_LR
              rocinfo = work.rocinfo_LR;
run;

proc assess data=ml.cyber_beth_logreg2_score ncuts=10 nbins=10;
   var _pred_;
   target sus / event="1" level=nominal;
   by _PartInd_;
   ods output liftinfo = work.liftinfo_LR2
              rocinfo = work.rocinfo_LR2;
run;

proc assess data=ml.cyber_beth_GB_score ncuts=10 nbins=10;
   var p_sus1;
   target sus / event="1" level=nominal;
   by _PartInd_;
   ods output liftinfo = work.liftinfo_GB
              rocinfo = work.rocinfo_GB;
run;
```



Lift Information

ponse Percent	Lift			Response Percent		Gain	
Cumulative	Individual	Cumulative	Cumulative Best	Individual	Cumulative	Individual	Best
0.00	.	.	.	.	.	.	.
80.52	8.052060	8.052060	10	36.46	36.46	7.052060	9.000000
82.69	0.216627	4.134344	5	0.98	18.72	3.134344	4.000000
84.85	0.216627	2.828438	3.333333	0.98	12.81	1.828438	2.333333
87.02	0.216627	2.175486	2.500000	0.98	9.85	1.175486	1.500000
89.19	0.216627	1.783714	2	0.98	8.08	0.783714	1
91.35	0.216627	1.522533	1.666667	0.98	6.89	0.522533	0.666667
93.52	0.216627	1.335975	1.428571	0.98	6.05	0.335975	0.428571
95.68	0.216627	1.196056	1.250000	0.98	5.42	0.196056	0.250000
97.85	0.216627	1.087231	1.111111	0.98	4.92	0.087231	0.111111
100.00	0.214922	1	1	0.98	4.53	0	0



nformation

ACC	KS	Youden Index	F1 Score	F0.5 Score	AUC	Gini	Gamma	Tau	Misclassification (Event)
45311	0	0	0.086694	0.056005	0.895652	0.791304	1	0.068488	0.954689
90544	1	0.791304	0.883495	0.949896	0.895652	0.791304	1	0.068488	0.009456
90544	0	0.791304	0.883495	0.949896	0.895652	0.791304	1	0.068488	0.009456
54689	0	0	0	0	0.895652	0.791304	1	0.068488	0.045311
54689	0	0	0	0	0.895652	0.791304	1	0.068488	0.045311
54689	0	0	0	0	0.895652	0.791304	1	0.068488	0.045311
54689	0	0	0	0	0.895652	0.791304	1	0.068488	0.045311
54689	0	0	0	0	0.895652	0.791304	1	0.068488	0.045311
54689	0	0	0	0	0.895652	0.791304	1	0.068488	0.045311
54689	0	0	0	0	0.895652	0.791304	1	0.068488	0.045311
54689	0	0	0	0	0.895652	0.791304	1	0.068488	0.045311