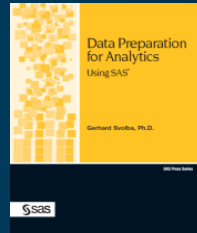


Wie bringe ich 4 unterschiedliche Analytik-Benutzergruppen an einen Tisch? – Die Offenheit von SAS Viya ermöglicht eine Analyseplattform für unterschiedliche Benutzertypen

Gerhard Svolba, SAS Austria

Mannheim, 2. März 2018 - KSFE 2018



<https://github.com/gerhard1050/>



SAS Forum Germany 2017



SAS Club 2017

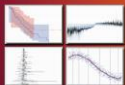


SAS Programmer,
25 years DATA; SET; RUN;

Python Guy with
Converse Shoes

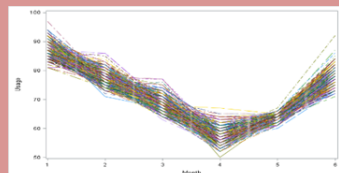
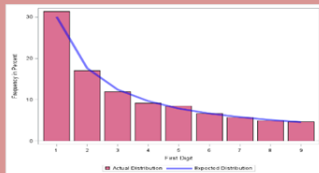
Point&Click
Business User





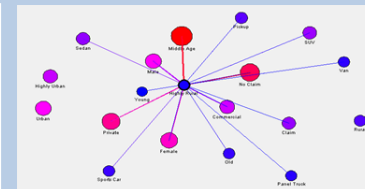
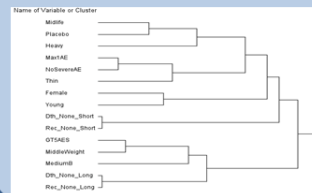
Checking the Alignment with Predefined Pattern

Which customers show a behaviour which is far from what you expected?



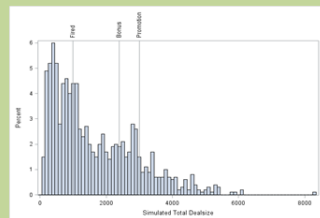
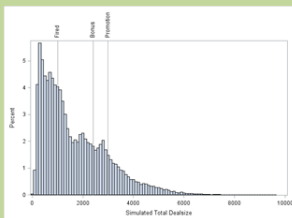
Listen to Your Data – Discover Unknown Relationships

Can your data tell you stories, even if you don't ask them?



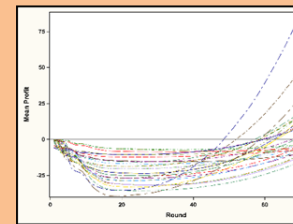
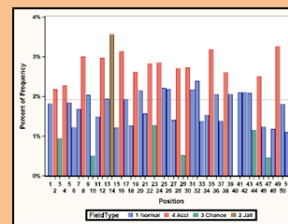
Using Monte Carlo Simulations to Understand the Outcome Distribution

Will the Sales Manager keep his job (when you look at his sales pipeline)?



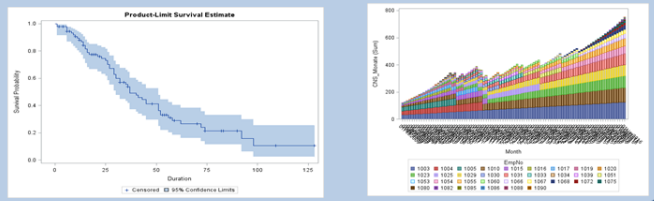
Studying Complex Systems – Simulate the Monopoly® Board Game

How can you simulate complex environments to get insight in the most frequent processes?



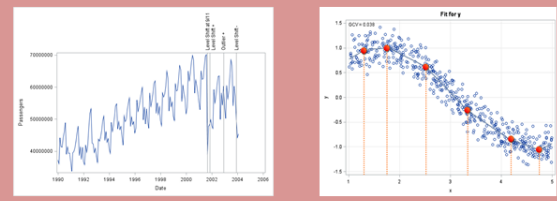
Performing Headcount Survival Analysis for Employee Retention

Can you make assumptions about the average length of time intervals, even if most of the endpoints have not yet been observed?



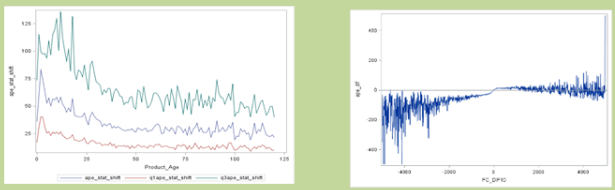
Detecting Outliers and Structural Changes in Longitudinal Data

Can you automatically detect events and changes in the course of your data over time?



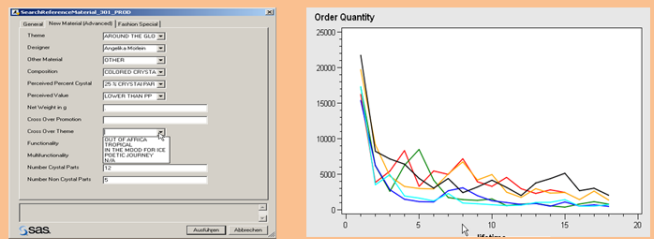
Explaining Deviations and Forecast Errors

Do the demand planners really improve forecast accuracy with their manual overwrites?



Forecasting the Demand for New Products

Can you assess the expected demand of products that are introduced right now?

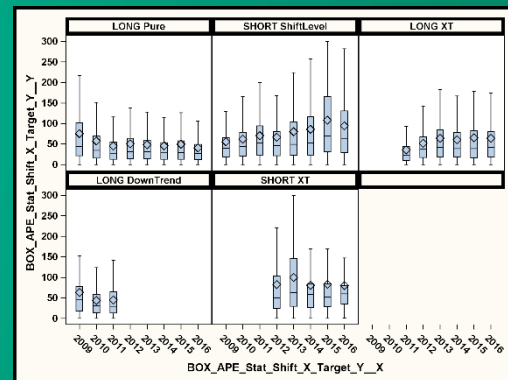
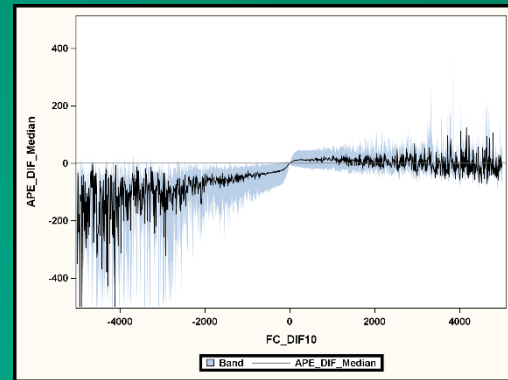


Data Science in Action: #4

Explaining Forecast Errors and Deviations

*Do the demand planners really improve
forecast accuracy with their manual
overwrites?*

Linear Regression
Quantile Regression
Descriptive Statistics



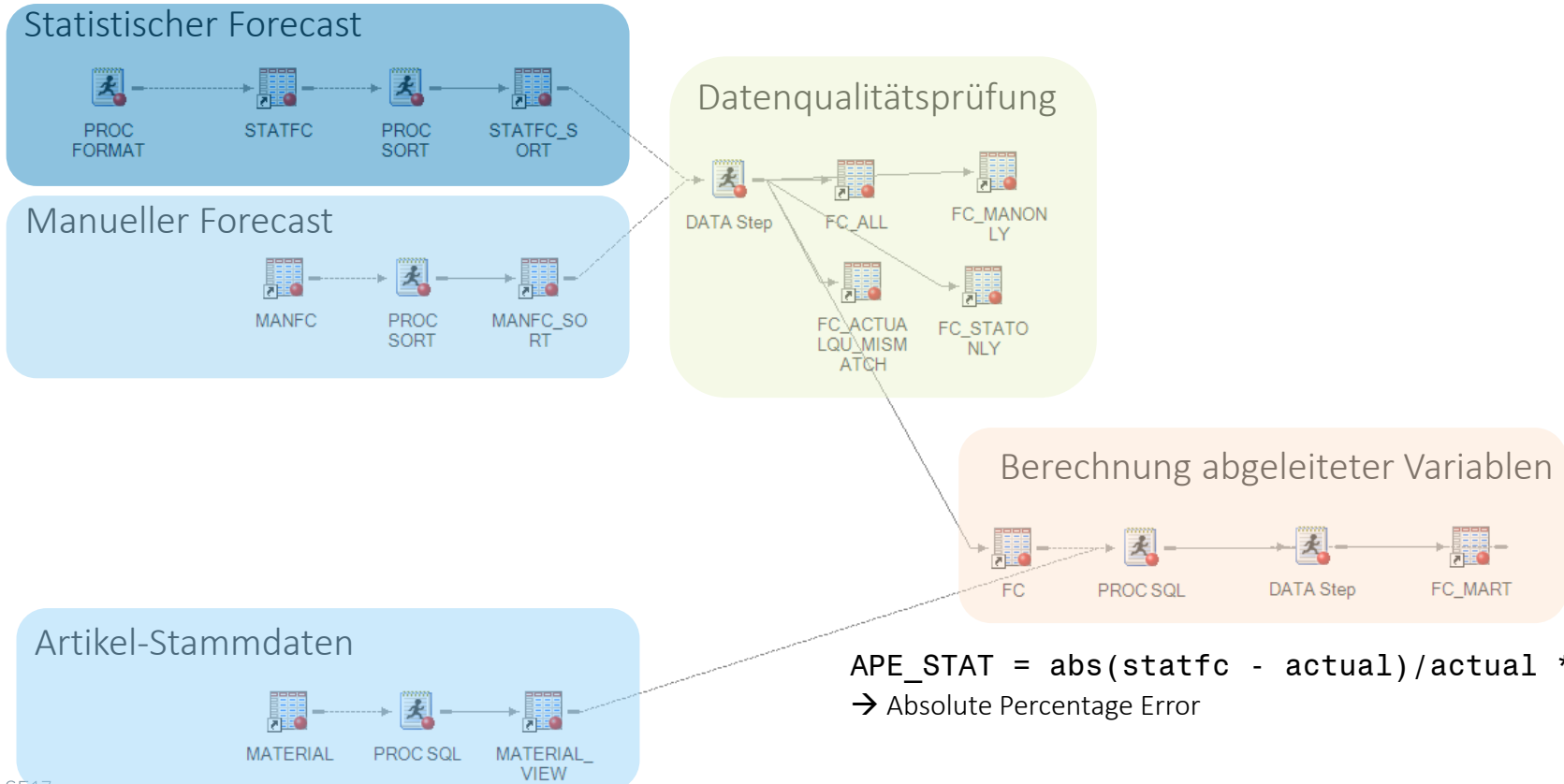
„Was bisher geschah“ (1)

Fachliche Fragestellung im Unternehmen

- Monatliche Nachfrage wird mittels Zeitreihenmodellen vorhergesagt.
- Produkte mit langer und kurzer Absatzhistorie (langjährige Artikel sowie Fashion)
- Analyse des Vorhersage-Fehlers
 - Welcher Vorhersage-Fehler wird von 50%, 75% meiner Vorhersagen nicht überschritten?
 - Welchen Faktoren führen zu geringen Vorhersagefehlern?
 - Gibt es zeitliche Trends und saisonale Muster im Vorhersagefehler?
- Faktoren
 - Stammdaten: PRODUCT_AGE, PRICE_INDEX, LAUNCH_MONTH, PRODUCT_GROUP
 - Stat. Forecasting: MODEL, LEAD_TIME, TARGET_YEAR, TARGET_CALENDAR_MONTH

„Was bisher geschah“ (2)

Datenaufbereitung



$$\text{APE_STAT} = \text{abs}(\text{statfc} - \text{actual}) / \text{actual} * 100$$

→ Absolute Percentage Error

4 different user roles analyse a business question

One Integrated Solution for Different User Types

Business Analyst
Gerhard Svolba
SAS Visual Analytics

New-to-SAS Statistician
Gernot Engel
SAS Studio Tasks

SAS Data Scientist
Franz Helmreich
SAS Studio Program

Open Source Data Scientist
Matthias Svolba
Python
(Jupyter Notebook)

IT and Application Mngt.



Opening the SAS Analytic Platform via Different Interfaces

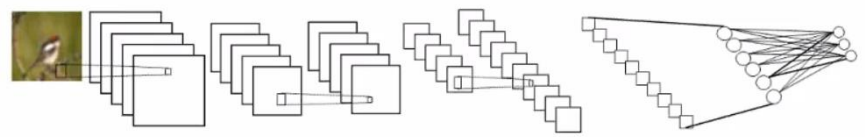


Start the SAS Cloud Analytic Server
Load the DeepLearning Action Set

```

jupyter cif_10_tech_exchng_demo Last Checkpoint: 30 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Help | Python
s = sw.CAS('cas01.unx.sas.com', 11775)
s.sessionprop.setsessopt(caslib='CASUSER', timeout=3.1536E7)
s.loadactionset('deepLearn')
    
```

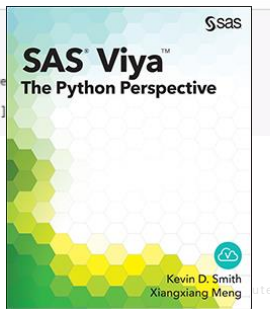
Define the Network Architecture



```

In [ ]: s.createModel(model=dict(name='convNet', replace=True), type='CNN')
s.addLayer(model='convNet', name='data', type='input',
            inputOpts=dict(nchannels=3, width=24, height=24, scale=1))
s.addLayer(model='convNet', name='conv1', type='convolution',
            convOpts=dict(nfilters=32, width=5, height=5, stride=1, init='msra2'), srcLayers=['data'])
s.addLayer(model='convNet', name='pool1', type='pooling',
            poolingOpts=dict(width=2, height=2, stride=2, pool='max'), srcLayers=['conv1'])
s.addLayer(model='convNet', name='conv2', type='convolution',
    
```

Define the Network Layers



Use the CAS Random Forest
Display the Results in R-Studio

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins
> dat1 = cas.read.csv(caslib, file = "//sashq/root/u/sasyqi/titanic_train.csv",
+                    casOut = list(replace = TRUE))
NOTE: Cloud Analytic Services made the uploaded file available as table TITANIC_TRAIN in caslib CASUSER(sasyqi)
> rfout = cas.decisionTree.forestTrain(
+   dat2[dat2$'_Pair_Ind_' == 1,]
    
```

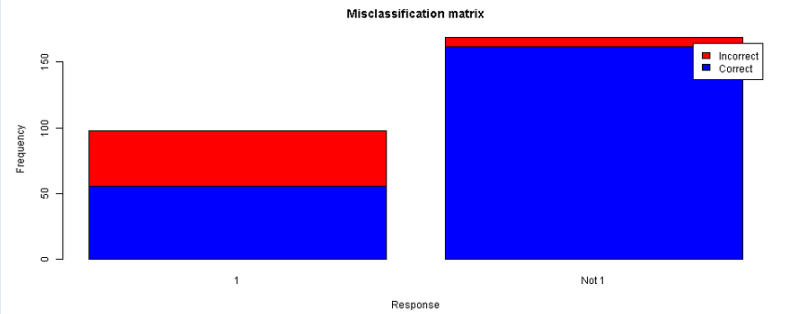
Shiny

Choose a cutstep value

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

0.63

Use R Applications

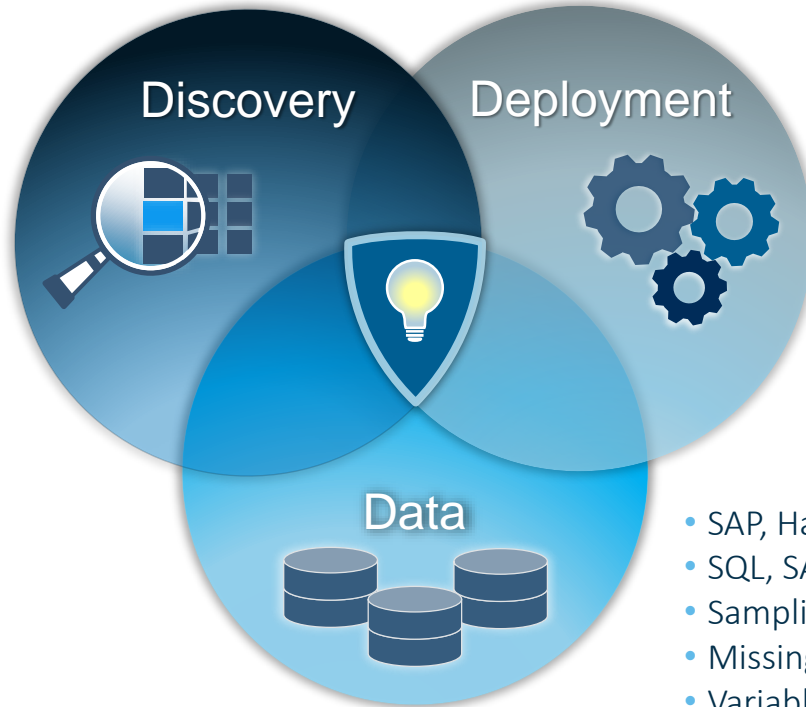


```

[1] "The statistics values for cutoff 0.63 is"
_Index_ Column_ Events_ Cutoff_ TP_ FP_ FN_ TN_ Sensitivity_ Specificity_
    
```

Data Mining und Machine Learning mit der SAS Analytic Plattform

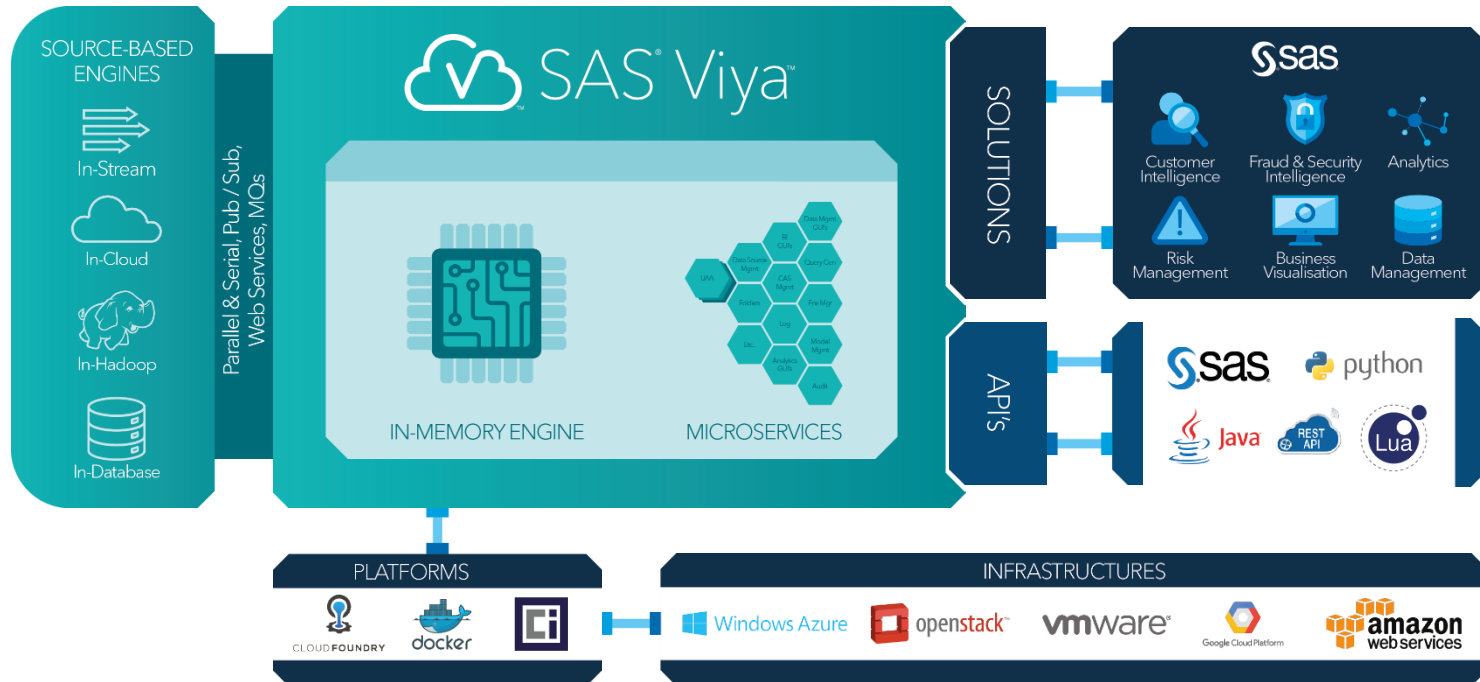
- Logistic Regression
- Linear Regression
- Generalized Linear Models
- Nonlinear Regression
- Ordinary Least Squares Regression
- Decision Trees
- Partial Least Squares Regression
- Quantile Regression
- K-means and K-modes Clustering
- Principal Component Analysis
- Random Forest
- Gradient Boosting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



- Assess Supervised Models
- Modellverwaltung
- Deployment
- Laufende Validierung
- Modell-Retirement
- Retraining

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

Überblick über die SAS Analytic Plattform

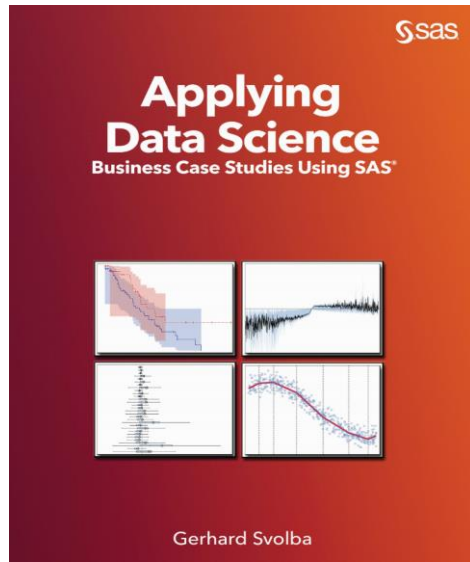


More Information

Gerhard Svolba – Principal Analytic Solutions Architect

sastools.by.gerhard@gmx.net

<https://github.com/gerhard1050/>



- Applying Data Science – Business Case Studies Using SAS, SAS Press 2017
- Eight Case Studies showing how Data Science and Analytics can be applied to provide insight into your data and improve your business decisions
- [http://www.sascommunity.org/wiki/Applying_Data_Science - Business Case Studies Using SAS](http://www.sascommunity.org/wiki/Applying_Data_Science_-_Business_Case_Studies_Using_SAS)

SAS Viya Technical Primer

Deep Learning Toolkit

Image Processing Toolkit

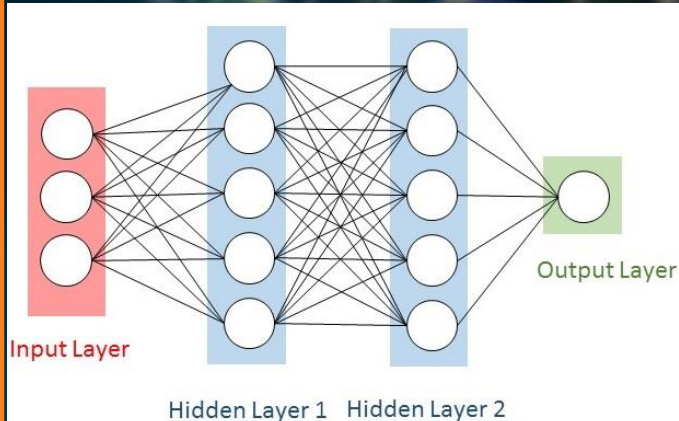
Natural Language Toolkit

Deep Learning Toolkit

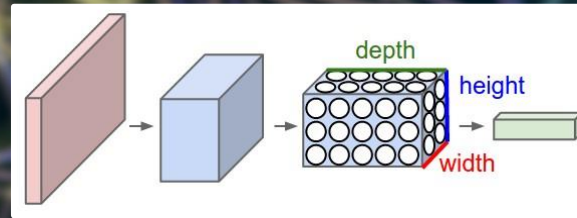
Ships with VDMML license as CAS actions

CAS Action, but also built into MS pipeline for Neural Nets with more than 5 layers

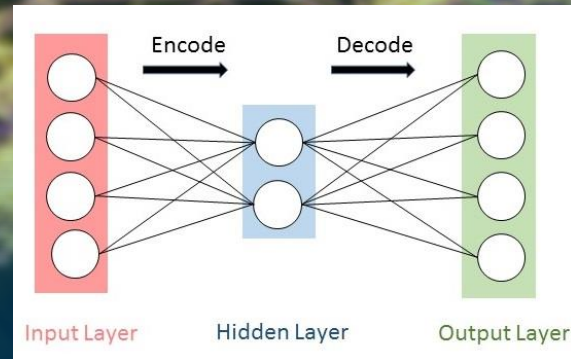
DEEP FORWARD



CONVOLUTIONAL



AUTOENCODERS



RECURRENT

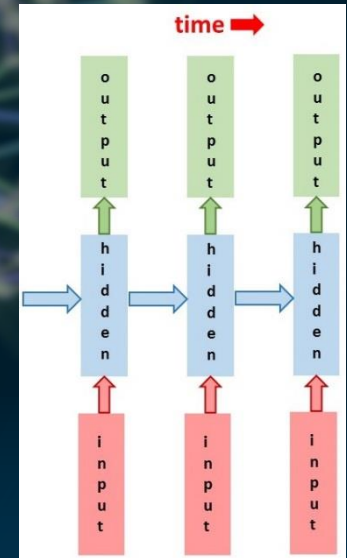


Image Processing Toolkit

Ships with VDMML license as CAS actions



Image Action Set (Image)

- **LoadImages** – loads images from a path
- **SaveImages** – writes images to a table
- **CompareImages** – compares two sets of images
- **ProcessImages** – performs core functions

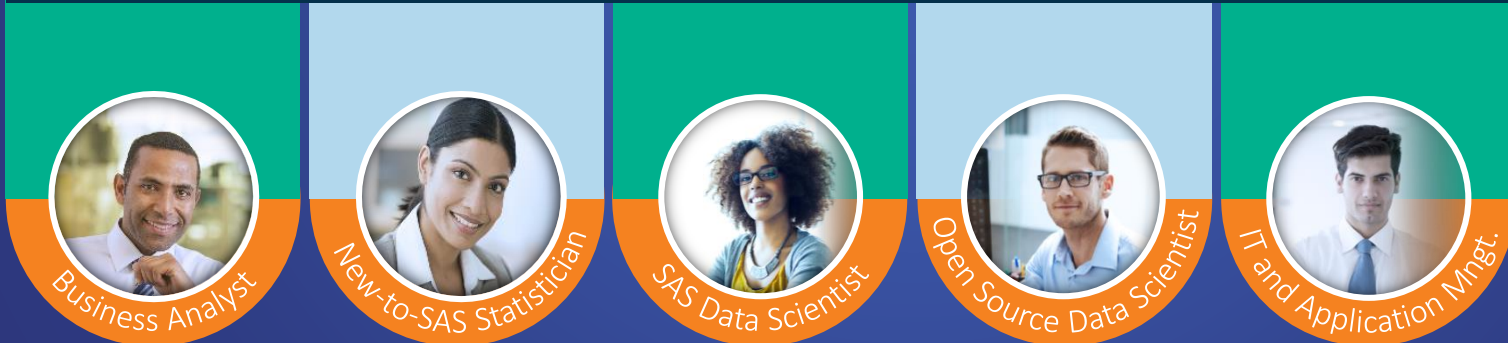
Biomedical Action Set (bioMedImage)

- **buildSurface** – This action can process an image table containing biomedical image data and generate 3D surfaces required to visualize the images.
- **Biomedical extensions to loadImages & saveImages** – Both of these actions are being extended to support loading and saving biomedical images (ex. DICOM images).

Analyse der NBA 1997 Daten durch 4 verschiedene Benutzer-Rollen

Offenheit der SAS Analytic Plattform für unterschiedliche Zugriffsarten

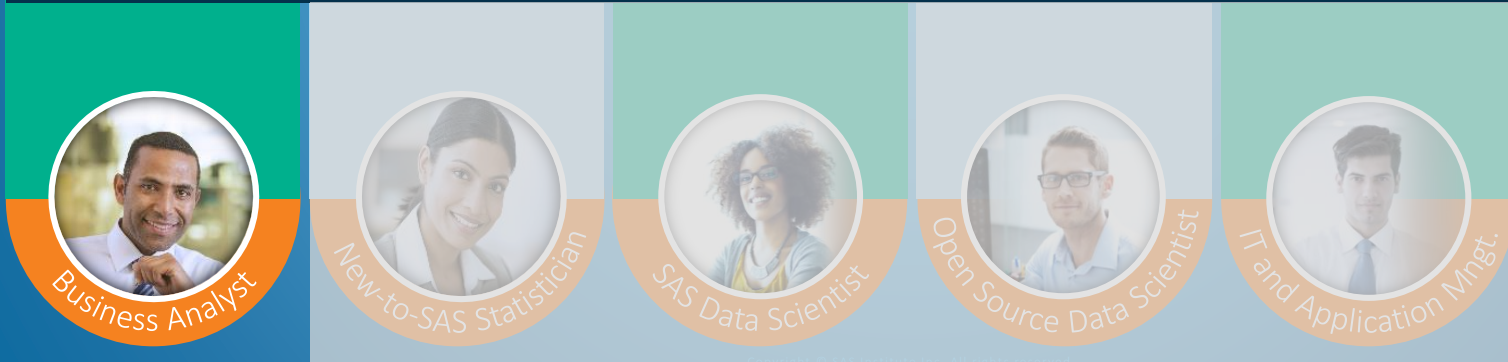
One Integrated Solution for Different User Types



Analyse der NBA 1997 Daten durch 4 verschiedene Benutzer-Rollen

SAS Visual Analytics und SAS Visual Statistics für den Business Analyst

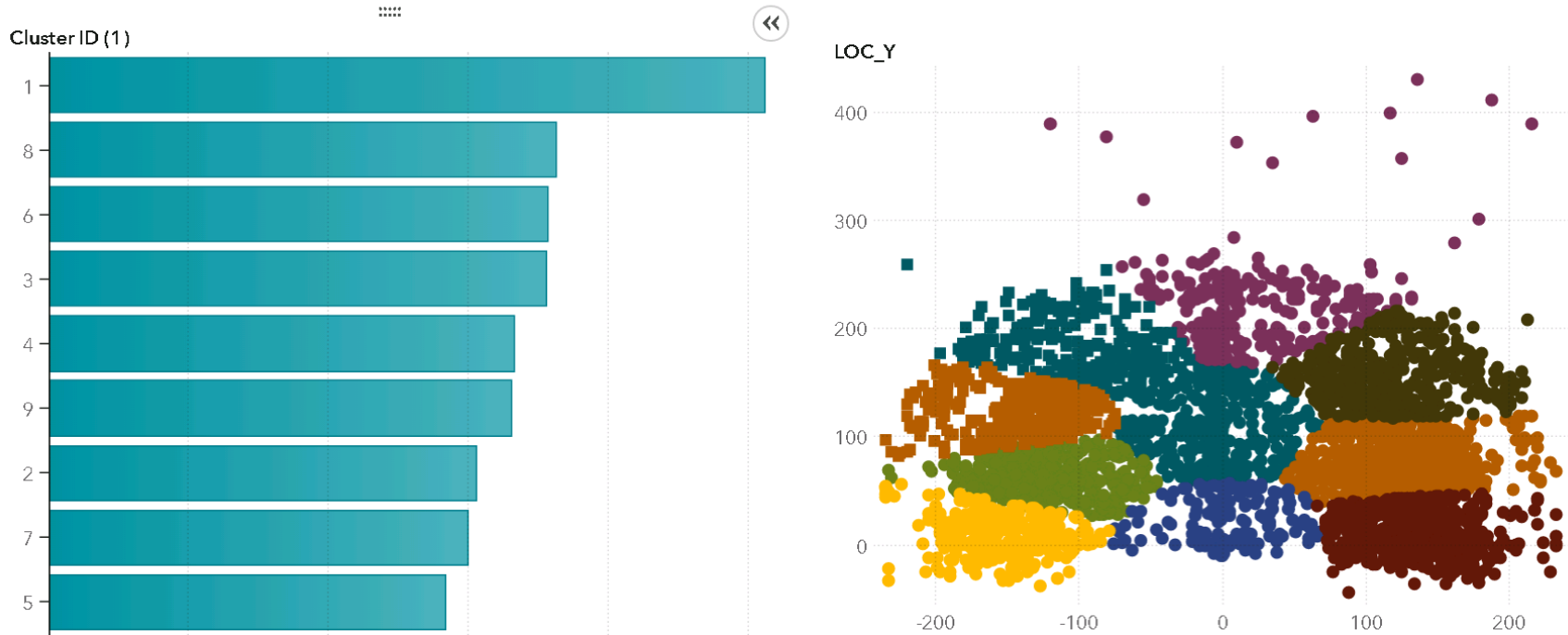
One Integrated Solution for Different User Types



SAS Visual Statistics für den Business Analyst

Point&Click Zugriff auf Machine Learning Methoden

Drop a data item or control to create a page prompt



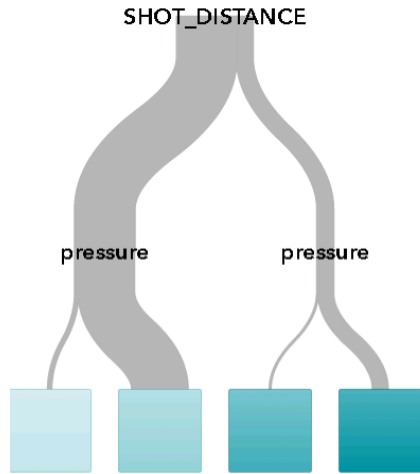
SAS Visual Statistics für den Business Analyst

Point&Click Zugriff auf Machine Learning Methoden

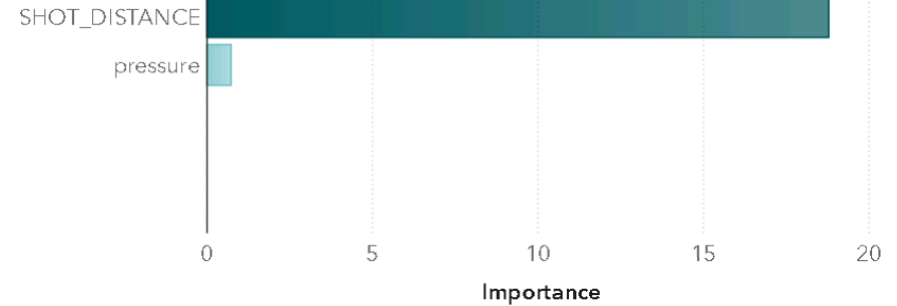
Drop a data item or control to create a page prompt

Decision Tree **SHOT_MADE_FLAG** ASE 0.245162 Observations Used 4,809

Tree



Variable Importance



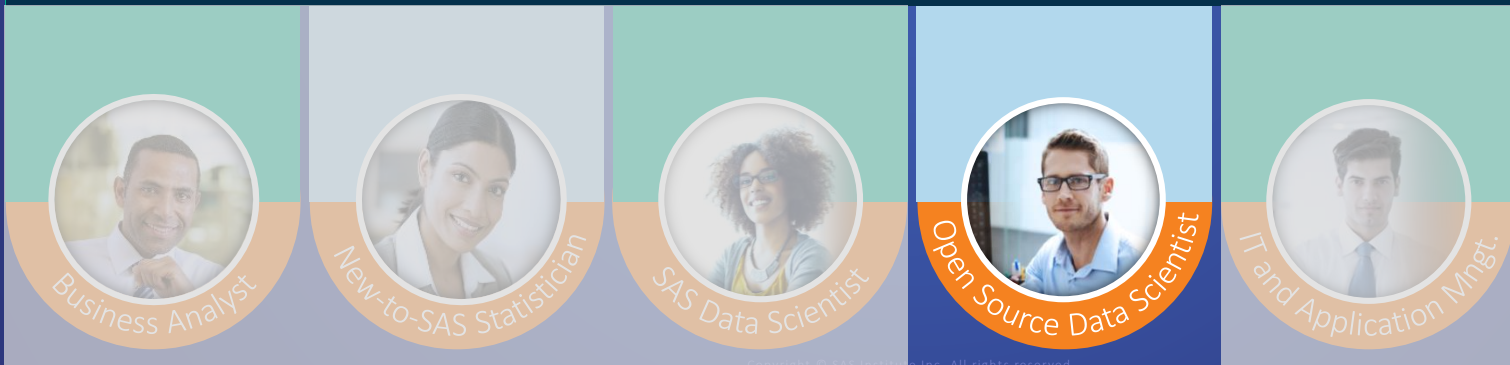
Assessment



Analyse der NBA 1997 Daten durch 4 verschiedene Benutzer-Rollen

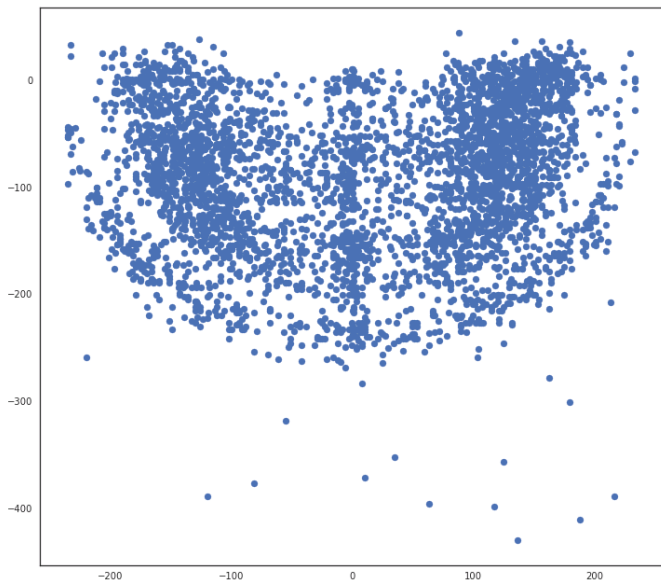
SAS-Python Integration für den Open Source Data Scientist

One Integrated Solution for Different User Types

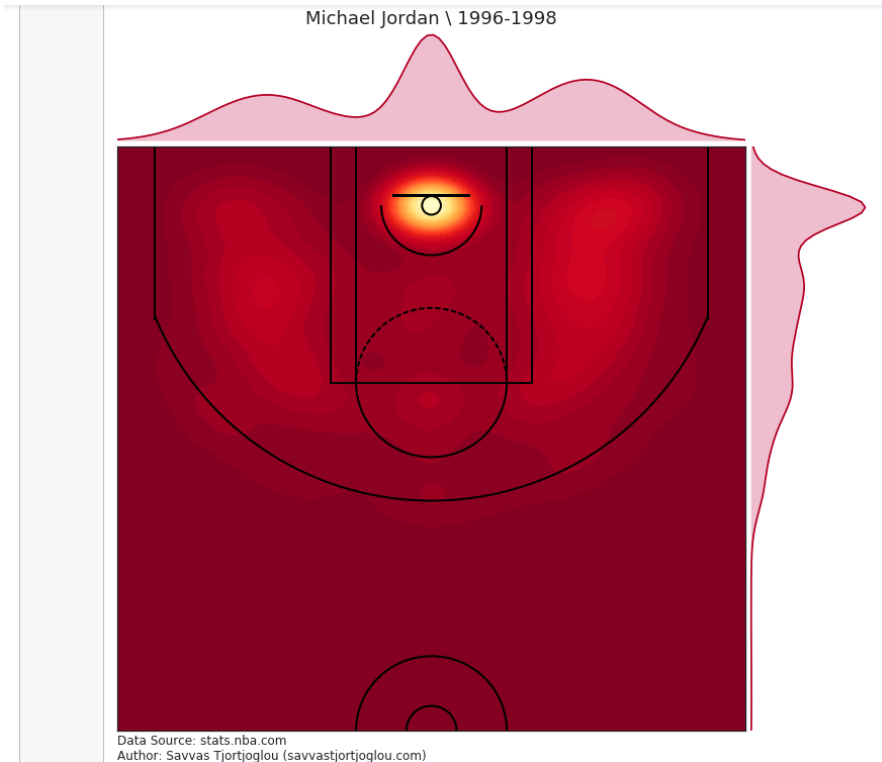


How good was Jordan under pressure?

Some Graphs benefiting from the open source community



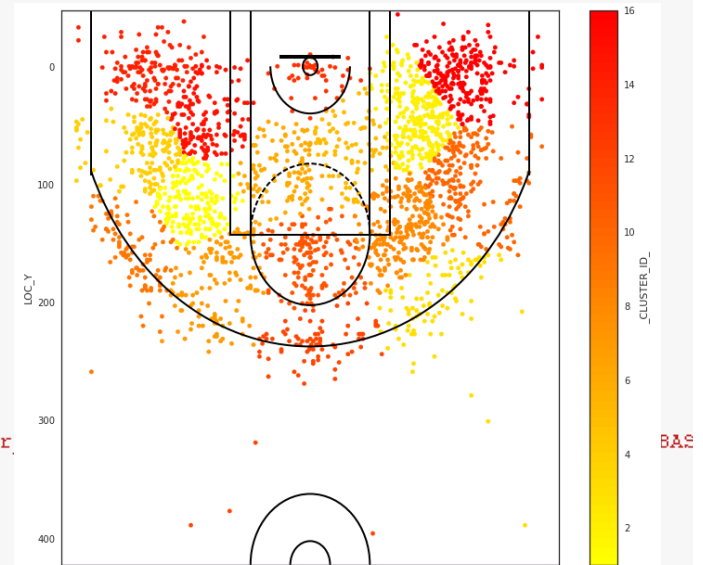
<http://savvastjortjoglou.com/nba-shot-sharts.html>



How good was Jordan under pressure?

Clustering the Court into Shot Zones using CAS action set

```
clust=sess.clustering.kClus(  
  table={  
    "name": "jordan_mining"  
  },  
  inputs={"LOC_X", "LOC_Y_MINUS"},  
  nClusters=30,  
  maxIters=10,  
  distanceNom="RELATIVEFREQ",  
  estimateNClusters={  
    "method": "ABC",  
    "B": 10,  
    "minClusters": 15,  
    "criterion": "ALL",  
    "align": "PCA"  
  },  
  kPrototypeParams={  
    "method": "USERGAMMA",  
    "value": 10  
  },  
  output={"CasOut": {"name": "kClusOutputScore", "replace": True},  
          "copyVars": {"LOC_X", "LOC_Y", "LOC_Y_MINUS", "Made", "Lead_Player"},  
          /  
  display={"names": {"Modelinfo", "ClusterSumIntNom"}}  
}
```



How good was Jordan under pressure?

Using CAS Regression to investigate performance under pressure

Logistic

```
In [69]: lr = sess.regression.logistic(
  table={"name": "kClusOutputScore"},
  classVars=[{"vars": {"_CLUSTER_ID_", "Lead_Player_Before", "Homegame", "ACTION_TYPE", "end_of_game", "Overtime", "close_game",
  model={
    "depVars": [{"name": "Made", "options": {"event": "1"}}],
    "effects": [{"vars": {"SHOT_DISTANCE", "_CLUSTER_ID_", "ACTION
  },

  outputTables={"names": "parameterestimates"}
)
sess.dataStep.runCode(
  code="""data round; set parameterestimates(keep=Parameter DF
do i = 1 to dim(_nums);
  _nums{i} = round(_nums{i}, .001);
end;
drop i;
run;""")
)
sess.fetch(table="round")
```

Out[69]: § Fetch

Selected Rows from Table ROUND

	Parameter	DF	Estimate	StdErr	ChiSq	ProbChiSq
0	Intercept	1.0	14.322	94.876	0.023	0.880
1	pressure 1	1.0	-0.385	0.188	4.193	0.041
2	pressure 0	0.0	0.000	NaN	NaN	NaN
3	Overtime Regular	1.0	0.382	0.370	1.069	0.301
4	Overtime Overtime	0.0	0.000	NaN	NaN	NaN
5	SHOT_DISTANCE	1.0	-0.035	0.014	6.500	0.011
6	ACTION_TYPE Tip Shot	1.0	-13.135	94.875	0.019	0.890
7	ACTION_TYPE Slam Dunk Shot	1.0	0.027	128.564	0.000	1.000
8	ACTION_TYPE Running Jump Shot	1.0	-12.634	94.876	0.018	0.894
9	ACTION_TYPE Layup Shot	1.0	-13.064	94.875	0.019	0.890
10	ACTION_TYPE Jump Shot	1.0	-14.106	94.875	0.022	0.882
11	ACTION_TYPE Hook Shot	1.0	-12.552	94.882	0.017	0.895
12	ACTION_TYPE Dunk Shot	1.0	-10.540	94.875	0.012	0.912
13	ACTION_TYPE Driving Layup Shot	1.0	-11.510	94.875	0.015	0.903
14	ACTION_TYPE Driving Dunk Shot	0.0	0.000	NaN	NaN	NaN

Analyse der NBA 1997 Daten durch 4 verschiedene Benutzer-Rollen

SAS Procedures für den SAS Data Scientist

One Integrated Solution for Different User Types



Business Analyst



New-to-SAS Statistician



SAS Data Scientist

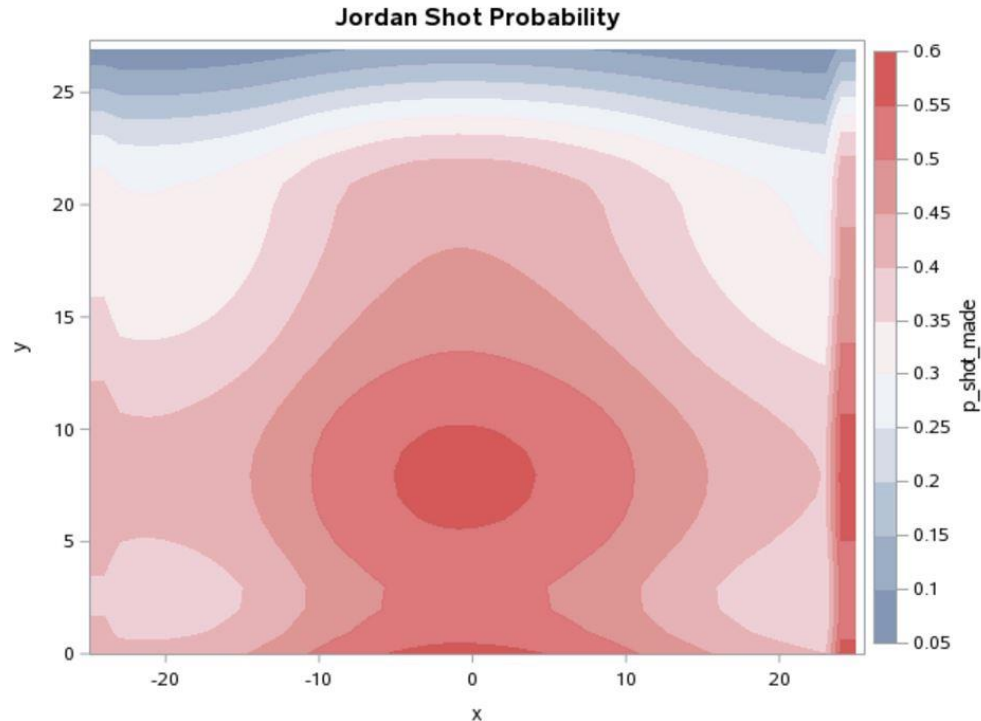


Open Source Data Scientist



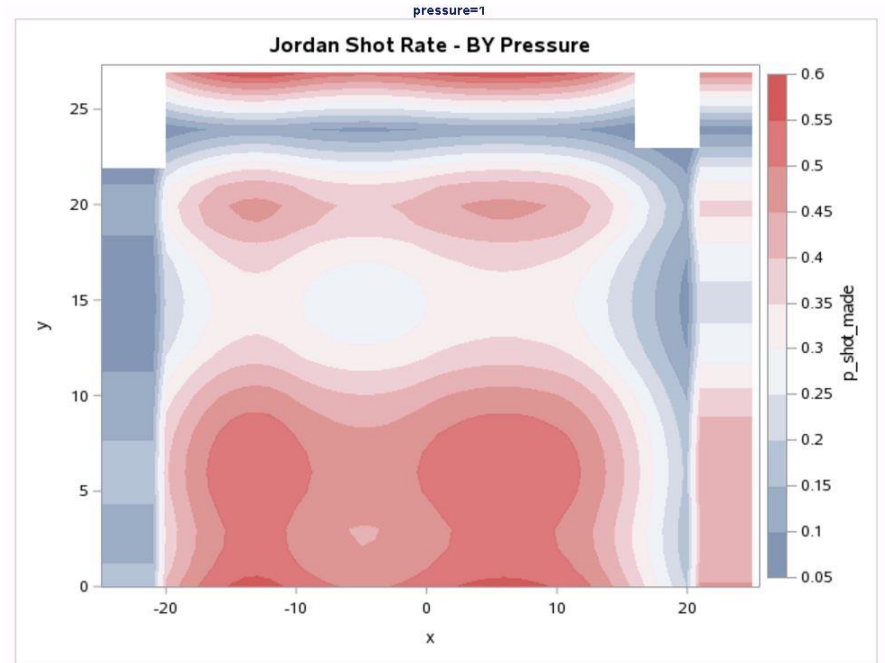
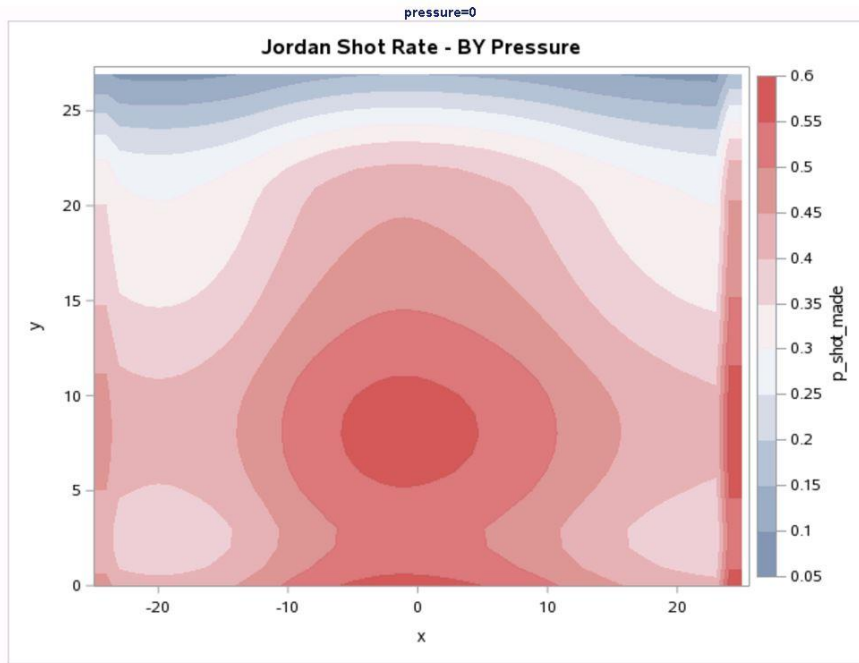
IT and Application Mngt.

Vorhersage der Treffer-Wahrscheinlichkeit von Jordan im $[-25,25] \times [0,27]$ Grid



```
proc logselect data=sfdcas.Jordan;  
where Shot_Distance <= 30;  
effect spl = spline(X Y / degree=2);  
model Shot_Made(event='1') = spl ;  
output out=sfdcas.Jordan_pred  
pred=p copyvars=(x y shot_distance  
shot_made);  
Code file='/opt/sasinside/  
DemoData/SFD/JordanPred_0.sas';  
run;
```

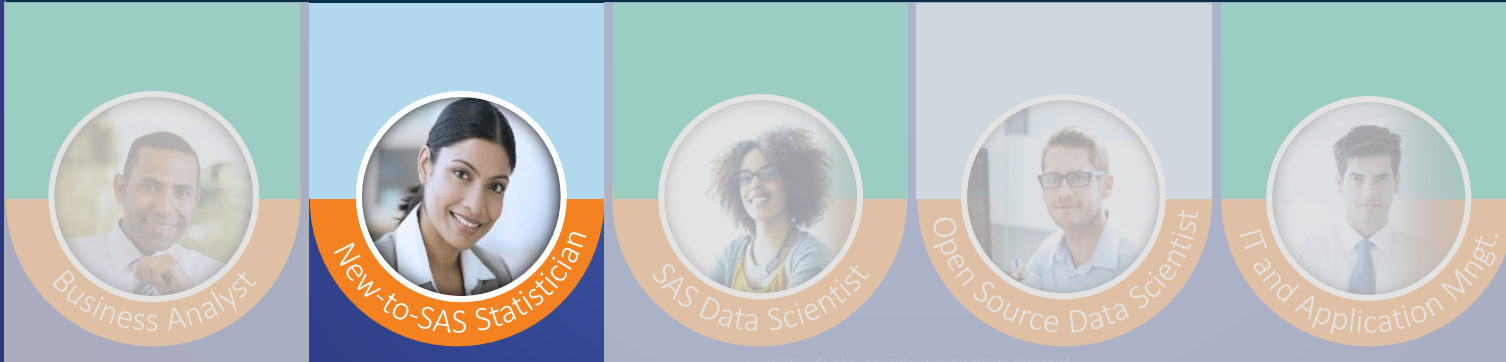
Vorhersage der Treffer-Wahrscheinlichkeit von Jordan getrennt nach Pressure ja/nein



Analyse der NBA 1997 Daten durch 4 verschiedene Benutzer-Rollen

Data Mining/Machine Learnings Tasks im SAS Studio für den „New-to-SAS“ Statistician

One Integrated Solution for Different User Types



Vordefinierte Tasks

SAS Studio

Code Generierung mit Tasks

The screenshot displays the SAS Studio interface. On the left, a 'Tasks' pane is open, showing various options for data analysis and visualization. A blue callout box labeled 'Optionen' points to this pane. The main window shows the 'Options' tab for a task, with various settings like 'Anzahl Bäume' (100), 'Maximale Baumtiefe' (20), and 'Mindestanzahl Beobachtungen je Blatt' (5). A blue callout box labeled 'Code Generierung' points to the code editor on the right, which contains SAS code for generating a forest plot and performing a classification task. The code includes comments in German and SAS procedures like `proc forest` and `proc sgplot`.

Vordefinierte Tasks

SAS Studio

Code Generierung mit Tasks

The screenshot displays the SAS Studio interface. On the left, a tree view shows 'Tasks' under 'Utilities'. A blue callout points to this area with the text 'Vordefinierte Tasks'. The main workspace is divided into 'METHODEN' and 'PLOTS' sections. The 'METHODEN' section contains various configuration options for a task, such as 'Anzahl Bäume' (set to 1,000), 'Maximale Baumtiefe' (set to 20), and 'Teilungskriterium' (set to 'Informationsgewinnquote (Standard)'). A blue callout points to these options with the text 'Optionen'. The 'PLOTS' section shows two plots: 'Fehlklassifikationen nach Anzahl der Bäume' and 'Variablenbedeutung'. The first plot is a line graph showing 'Fehlklassifikationsanteil' (y-axis, 0.30 to 0.45) versus 'Anzahl Bäume' (x-axis, 0 to 1000). It features three lines: 'Training' (solid blue), 'OOB' (dashed red), and 'Validierung' (dotted green). The 'OOB' line starts at approximately 0.45 and stabilizes around 0.43. The 'Training' line starts at approximately 0.35 and stabilizes around 0.32. The 'Validierung' line starts at approximately 0.42 and stabilizes around 0.41. A blue callout points to this plot with the text 'Ergebnisse'. The second plot is a horizontal bar chart titled 'Variablenbedeutung' showing the importance of variables like 'ACTION_TYPE', 'Margin_Player_Before', 'SHOT_ZONE_AREA', and 'SHOT_DISTANCE'. The 'ACTION_TYPE' variable has the highest importance, followed by 'Margin_Player_Before'.

Vordefinierte Tasks

SAS Studio

Code Generierung mit Tasks

The screenshot displays the SAS Studio interface with the 'Tasks' pane on the left, the 'Einstellungen' (Settings) pane in the center, and the 'ERGEBNISSE' (Results) pane on the right. A blue callout box points to the 'Auto-Tune' options in the settings, and another points to the 'Best Configuration' and 'Tuner Results' tables in the results pane.

Auto-Tune Options (Aktivierung der Autotuning Funktionalität):

- Auto-Tune für Anzahl Bäume
 - Startwert: 100
 - Untergrenze: 20
 - Obergrenze: 150
- Auto-Tune für Baumtiefe
 - Startwert: 20
 - Untergrenze: 1
 - Obergrenze: 29
- Auto-Tune für Bootstrap-Stichprobenanteil
 - Startwert: 0,6
 - Untergrenze: 0,1
 - Obergrenze: 0,9
- Auto-Tune für Anzahl der zur Teilung eines Knotens berücksichtigten Eingaben
 - Startwert: 5
 - Untergrenze: 1
 - Obergrenze: 100

Teilungskriterium: Informationsgewinnquote (Standard)

Mindestanzahl Beobachtungen je Blatt: 5

Maximale Anzahl Zweige je Knoten: 2

Variablenbedeutungsmethoden: Gini (Standard)

Methode zur Berechnung vorhergesagter nominaler Zielausprägungen: Wahrscheinlichkeit (Standard)

Best Configuration:

Parameter	Value
Evaluierung	12
Number of Trees	80
Number of Variables to Try	5
Bootstrap	0,795207
Maximum Tree Levels	14
Fehlklassifizierungsfehler Prozent	40,08

Tuner Results (Default and Best Configurations):

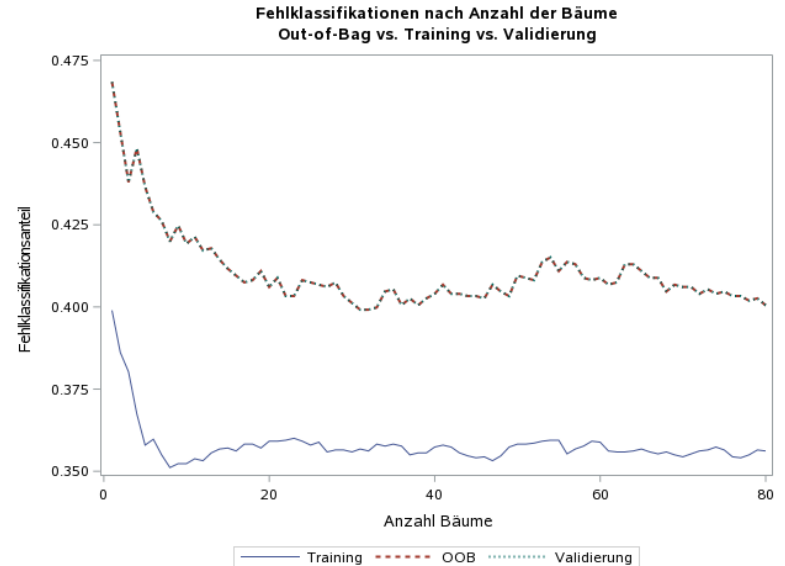
Evaluation	Maximum Tree Levels	Number of Trees	Number of Variables to Try	Bootstrap	Misclassification Error Percentage
0	21	100	14	0,600000	41,30
12	14	80	5	0,795207	40,08
45	14	80	5	0,791425	40,28
39	12	72	7	0,725718	40,54
36	12	70	7	0,705657	40,61
3	2	20	14	0,277778	40,88
17	5	34	12	0,403607	40,88
22	4	22	13	0,335016	40,88
24	5	34	12	0,394741	40,88
32	2	20	14	0,218958	40,88
33	6	33	12	0,415225	40,88

Ergebnisse

SAS Studio

Code Generierung mit Tasks

Tuner Results					
Default and Best Configurations					
Evaluation	Maximum Tree Levels	Number of Trees	Number of Variables to Try	Bootstrap	Misclassification Error Percentage
0	21	100	14	0.600000	41.30
12	14	80	5	0.795267	40.06
45	14	80	5	0.791425	40.26
39	12	72	7	0.725718	40.54
30	12	70	7	0.705657	40.61
3	2	20	14	0.277778	40.68
17	5	34	12	0.403667	40.68
22	4	22	13	0.335616	40.68
24	5	34	12	0.394741	40.68
32	2	20	14	0.218658	40.68
33	6	33	12	0.415225	40.68



SAS Studio

Code Generierung mit Tasks

