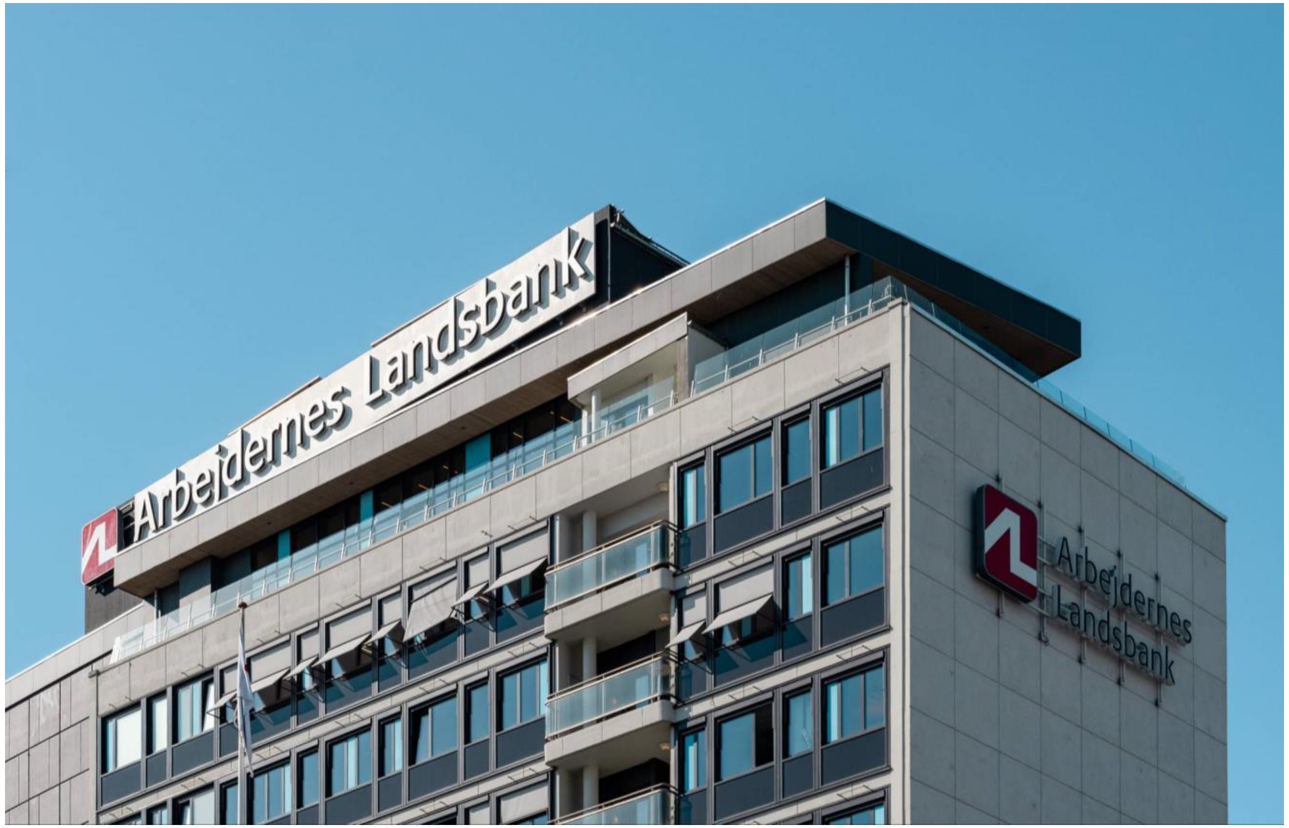
Arbejdernes Landsbank

Development and management of credit models in a new perspective

Michelle Brøndum Skinbjerg, Data Analyst

Agenda

- Introduction
- Journey
- Implementation



Introduction

Introduction

NAME AND TITLE

Michelle Brøndum Skinbjerg
Data Analyst at Arbejdernes Landsbank

BACKGROUND

Master of Science (MSc) in Mathematics-Economics from the University of Copenhagen

EXPERIENCE

- Model development, including credit risk models – especially LGD/EAD
- Rating models

Journey

New workflows are created during implementation

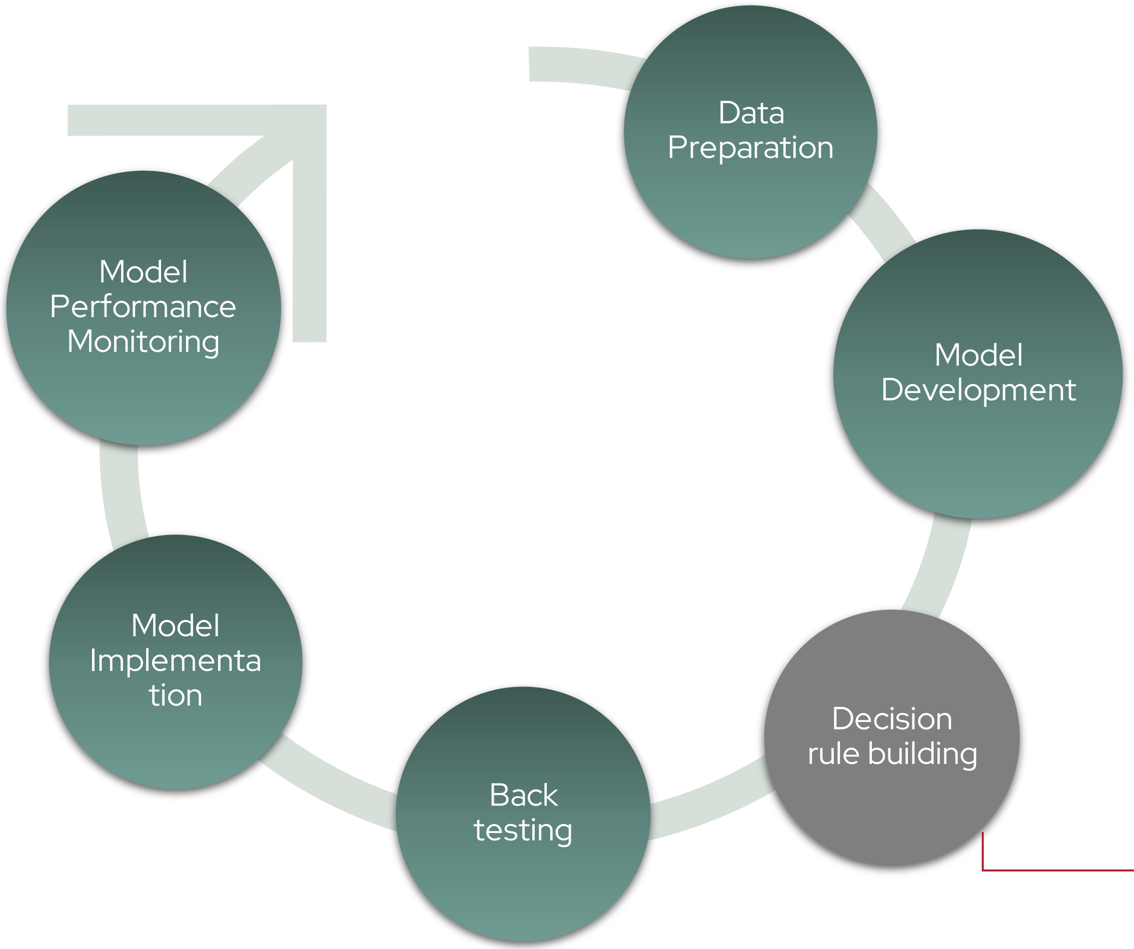
What is SAS® Risk Modeling?

SAS Risk Modeling is a next generation unified platform for the development and deployment of risk models.

SAS Risk Modeling offers an integrated infrastructure to handle data, build models using traditional statistical methods, advance machine learning techniques, back test the models, and quickly implement those models.

SAS Risk Modeling enables any institution that deals with risks to develop effective risk models for various use cases and to track these risks more accurately

Model lifecycle in SAS® Risk Modeling

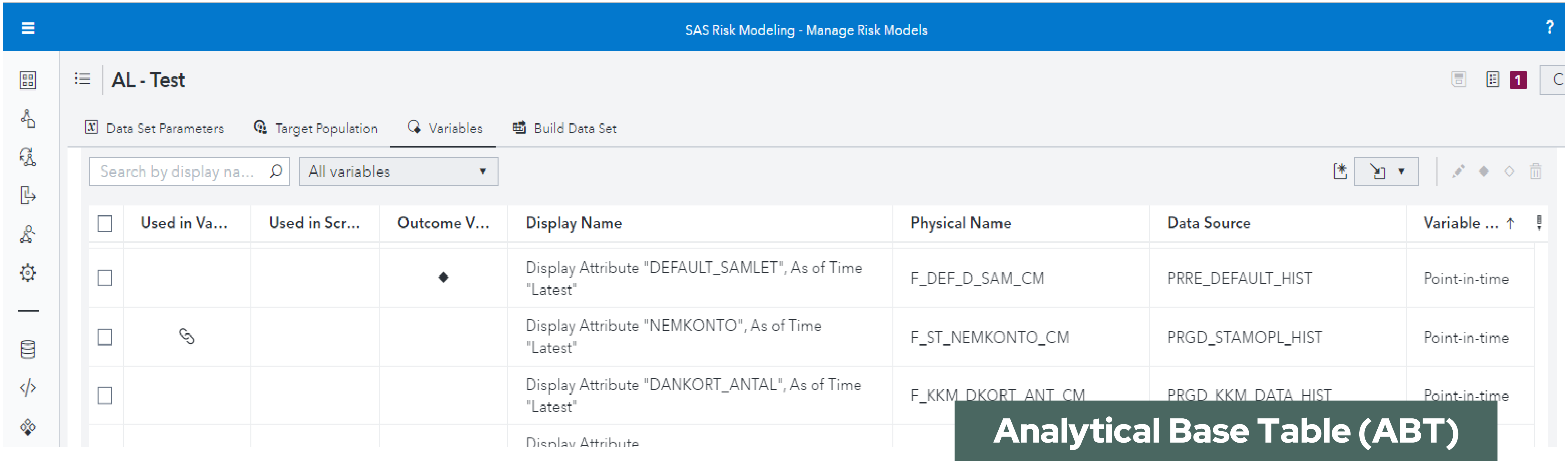


The decision-making process can be activated in both

- Batch
- Realtime

SAS® Risk Modeling

Data Preparation



SAS Risk Modeling - Manage Risk Models

AL - Test

Data Set Parameters | Target Population | Variables | Build Data Set

Search by display na... | All variables

□	Used in Va...	Used in Scr...	Outcome V...	Display Name	Physical Name	Data Source	Variable ... ↑
□			◆	Display Attribute "DEFAULT_SAMLET", As of Time "Latest"	F_DEF_D_SAM_CM	PRRE_DEFAULT_HIST	Point-in-time
□	🔗			Display Attribute "NEMKONTO", As of Time "Latest"	F_ST_NEMKONTO_CM	PRGD_STAMOPL_HIST	Point-in-time
□				Display Attribute "DANKORT_ANTAL", As of Time "Latest"	F_KKM_DKORT_ANT_CM	PRGD_KKM_DATA_HIST	Point-in-time
				Display Attribute			

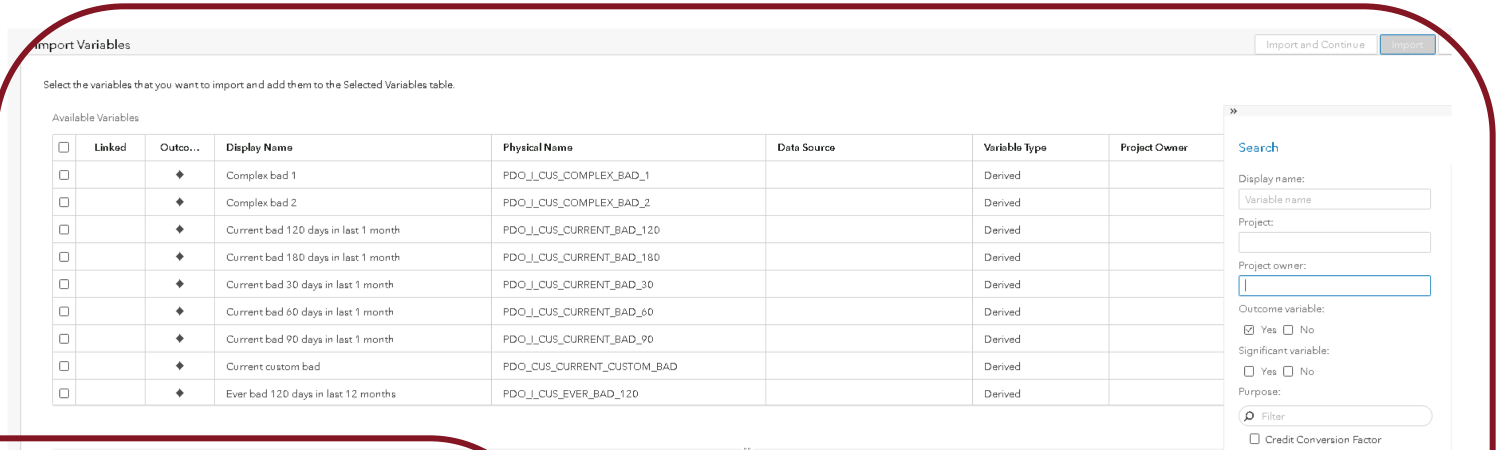
Analytical Base Table (ABT)

- Defining project
- Target population
- Modeling data set and variables
- Building the modeling dataset

Data Preparation

SAS® Risk Modeling

2



Import Variables

Select the variables that you want to import and add them to the Selected Variables table.

Available Variables

□	Linked	Outco...	Display Name	Physical Name	Data Source	Variable Type	Project Owner
□		◆	Complex bad 1	PDO_I_CUS_COMPLEX_BAD_1		Derived	
□		◆	Complex bad 2	PDO_I_CUS_COMPLEX_BAD_2		Derived	
□		◆	Current bad 120 days in last 1 month	PDO_I_CUS_CURRENT_BAD_120		Derived	
□		◆	Current bad 180 days in last 1 month	PDO_I_CUS_CURRENT_BAD_180		Derived	
□		◆	Current bad 30 days in last 1 month	PDO_I_CUS_CURRENT_BAD_30		Derived	
□		◆	Current bad 60 days in last 1 month	PDO_I_CUS_CURRENT_BAD_60		Derived	
□		◆	Current bad 90 days in last 1 month	PDO_I_CUS_CURRENT_BAD_90		Derived	
□		◆	Current custom bad	PDO_CUS_CURRENT_CUSTOM_BAD		Derived	
□		◆	Ever bad 120 days in last 12 months	PDO_I_CUS_EVER_BAD_120		Derived	

Name	Description	Data Source	Variable Type

No items are available.

Display name: Variable name

Project:

Project owner:

Outcome variable: ☑ Yes □ No

Significant variable: □ Yes □ No

Purpose: Filter

□ Credit Conversion Factor

□ Loss Given Default

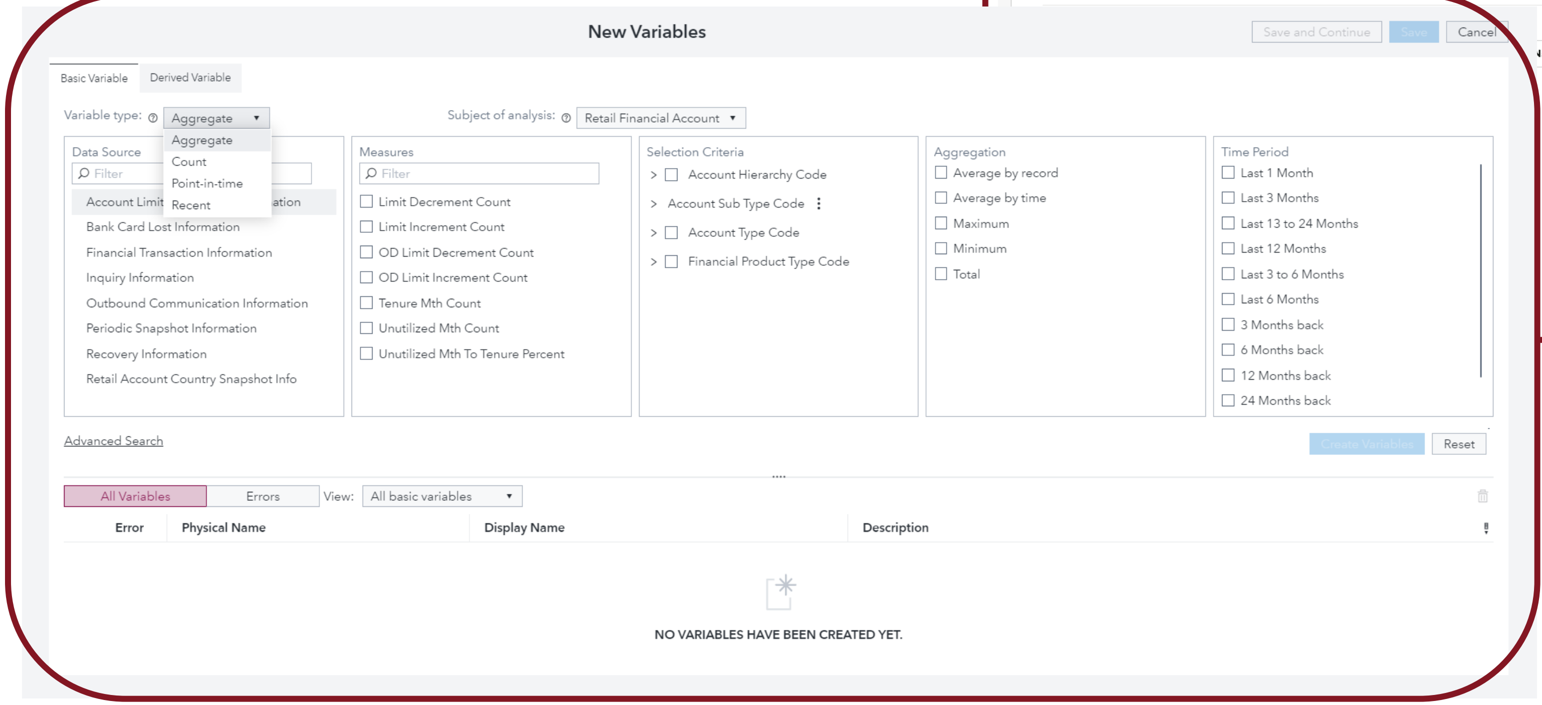
☑ Probability of Default

Subject of analysis: Filter

□ Retail Customer

Search Reset All

1



New Variables

Variable type: Aggregate

Subject of analysis: Retail Financial Account

Data Source
- Account Limit ...ation
- Bank Card Lost Information
- Financial Transaction Information
- Inquiry Information
- Outbound Communication Information
- Periodic Snapshot Information
- Recovery Information
- Retail Account Country Snapshot Info

Measures
- □ Limit Decrement Count
- □ Limit Increment Count
- □ OD Limit Decrement Count
- □ OD Limit Increment Count
- □ Tenure Mth Count
- □ Unutilized Mth Count
- □ Unutilized Mth To Tenure Percent

Selection Criteria
- > □ Account Hierarchy Code
- > Account Sub Type Code
- > □ Account Type Code
- > □ Financial Product Type Code

Aggregation
- □ Average by record
- □ Average by time
- □ Maximum
- □ Minimum
- □ Total

Time Period
- □ Last 1 Month
- □ Last 3 Months
- □ Last 13 to 24 Months
- □ Last 12 Months
- □ Last 3 to 6 Months
- □ Last 6 Months
- □ 3 Months back
- □ 6 Months back
- □ 12 Months back
- □ 24 Months back

View: All basic variables

Error	Physical Name	Display Name	Description

NO VARIABLES HAVE BEEN CREATED YET.

Core elements

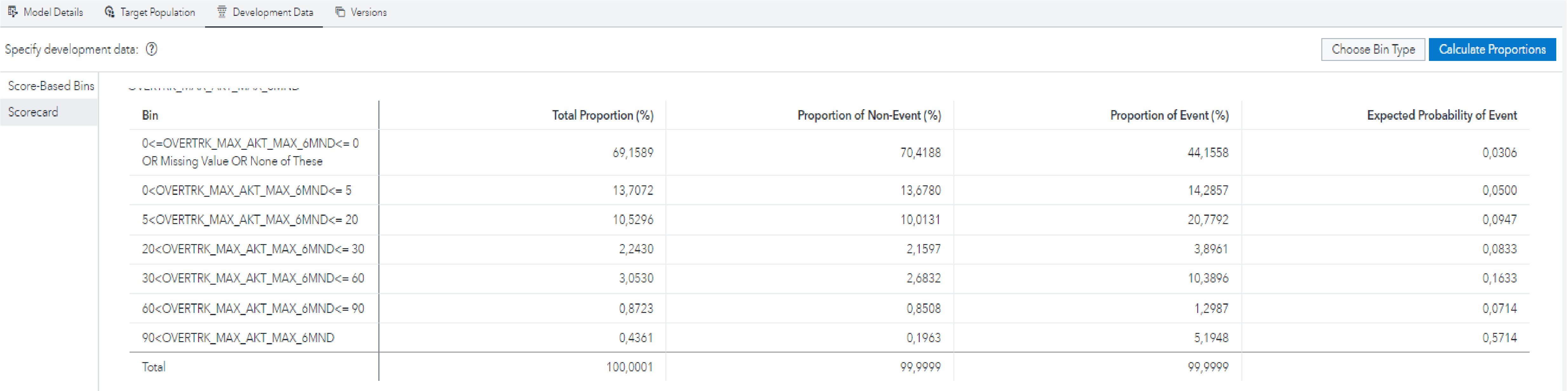
Create derived variables through point-and-click options and across dimensions:

- Aggregation is available immediately, saving time
- Reduces programming errors
- ABTs can be saved and shared, saving time and increasing productivity

Model development

- Develop a statistical or Machine learning model in SAS VDMML/Python
- Import the model in SAS Risk Modeling or create a user defined model in SAS Risk Modeling using the model specification workspace
- Define bins and proportions
- Calculate development data statistics
- Create model specification version

Model development



OVERTRK_MAX_AKT_MAX_6MND

Bin	Total Proportion (%)	Proportion of Non-Event (%)	Proportion of Event (%)	Expected Probability of Event
0<=OVERTRK_MAX_AKT_MAX_6MND<= 0 OR Missing Value OR None of These	69,1589	70,4188	44,1558	0,0306
0<OVERTRK_MAX_AKT_MAX_6MND<= 5	13,7072	13,6780	14,2857	0,0500
5<OVERTRK_MAX_AKT_MAX_6MND<= 20	10,5296	10,0131	20,7792	0,0947
20<OVERTRK_MAX_AKT_MAX_6MND<= 30	2,2430	2,1597	3,8961	0,0833
30<OVERTRK_MAX_AKT_MAX_6MND<= 60	3,0530	2,6832	10,3896	0,1633
60<OVERTRK_MAX_AKT_MAX_6MND<= 90	0,8723	0,8508	1,2987	0,0714
90<OVERTRK_MAX_AKT_MAX_6MND	0,4361	0,1963	5,1948	0,5714
Total	100,0001	99,9999	99,9999	

Model Implementation

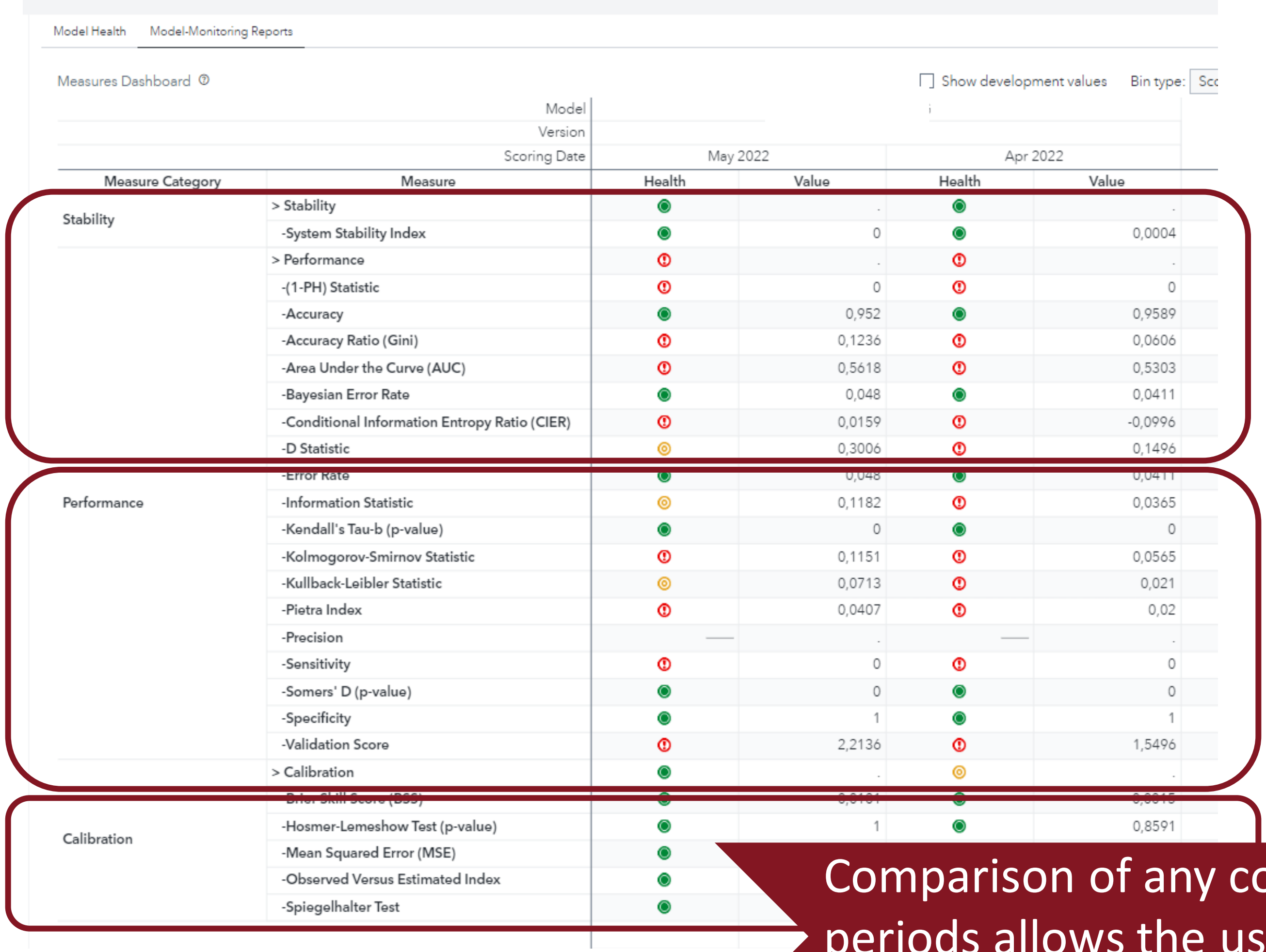
- Model Implementation means we are creating a deployed code that can be used for Scoring, Actual calculations and On-going model monitoring calculations.
- This deployed code can be used for Scoring the new through-the-door population.
- Model back testing is performed based on historical data.
- The model has to be implemented for making business decisions.
- Usually, it is done in batch mode.

Back Testing

- One does not want to wait for actual scoring results to validate model.
- Back testing allows you to validate the model immediately before we move it to production.
- Back testing is performed on historical data to compare scored/predicted Vs Actual results of model.

SAS® Risk Modeling

Back Testing



Model Health | Model-Monitoring Reports

Measures Dashboard ⓘ

□ Show development values   Bin type: Sco

		Model				
		Version				
		Scoring Date	May 2022		Apr 2022	
Measure Category	Measure		Health	Value	Health	Value
Stability	> Stability		🟢	.	🟢	.
	-System Stability Index		🟢	0	🟢	0,0004
	> Performance		ⓘ	.	ⓘ	.
	-(1-PH) Statistic		ⓘ	0	ⓘ	0
	-Accuracy		🟢	0,952	🟢	0,9589
	-Accuracy Ratio (Gini)		ⓘ	0,1236	ⓘ	0,0606
	-Area Under the Curve (AUC)		ⓘ	0,5618	ⓘ	0,5303
	-Bayesian Error Rate		🟢	0,048	🟢	0,0411
	-Conditional Information Entropy Ratio (CIER)		ⓘ	0,0159	ⓘ	-0,0996
	-D Statistic		◎	0,3006	ⓘ	0,1496
	-Error Rate		🟢	0,048	🟢	0,0411
Performance	-Information Statistic		◎	0,1182	ⓘ	0,0365
	-Kendall's Tau-b (p-value)		🟢	0	🟢	0
	-Kolmogorov-Smirnov Statistic		ⓘ	0,1151	ⓘ	0,0565
	-Kullback-Leibler Statistic		◎	0,0713	ⓘ	0,021
	-Pietra Index		ⓘ	0,0407	ⓘ	0,02
	-Precision		—	.	—	.
	-Sensitivity		ⓘ	0	ⓘ	0
	-Somers' D (p-value)		🟢	0	🟢	0
	-Specificity		🟢	1	🟢	1
	-Validation Score		ⓘ	2,2136	ⓘ	1,5496
	> Calibration		🟢	.	◎	.
	-Brier Skill Score (BSS)		🟢	[illegible]	🟢	[illegible]
Calibration	-Hosmer-Lemeshow Test (p-value)		🟢	1	🟢	0,8591
	-Mean Squared Error (MSE)		🟢			
	-Observed Versus Estimated Index		🟢			
	-Spiegelhalter Test		🟢			

Comparison of any combination of time periods allows the user to monitor seasonal effects for better decisions

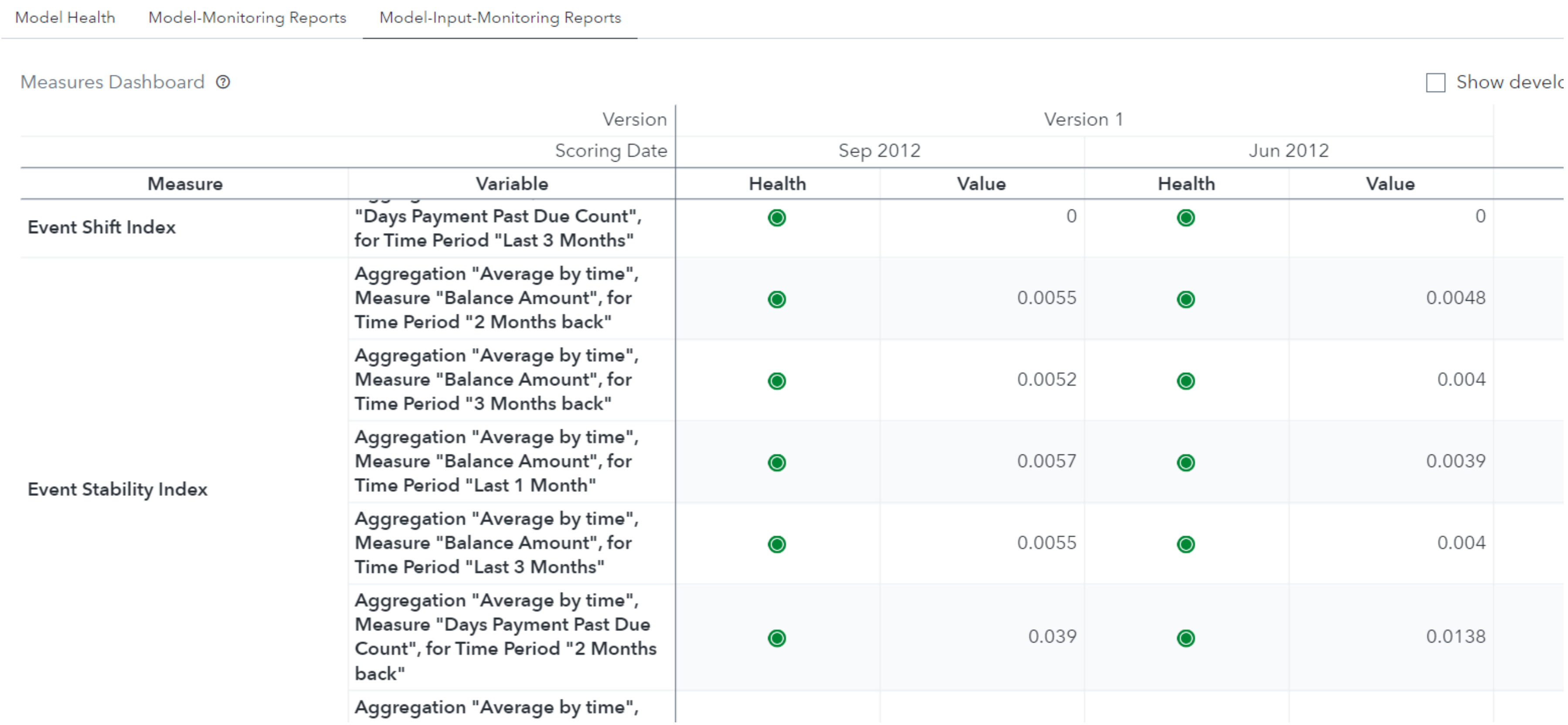
Model Monitering

SAS Risk Modeling includes an extensive set of reports for monitoring the performance of predictive scoring models; classified as:

- Model health reports
  - These reports show the overall health of the predictive scoring models through model health indicators.
- Model-monitoring reports
  - These reports monitor the performance of the predictive scoring models that score data in SAS Credit Scoring.
- Model-input-monitoring reports
  - These reports monitor the input variables on which the predictive scoring models are based.

SAS® Risk Modeling

Model Monitering



	Version	Version 1			
	Scoring Date	Sep 2012		Jun 2012	
Measure	Variable	Health	Value	Health	Value
Event Shift Index	"Days Payment Past Due Count", for Time Period "Last 3 Months"	●	0	●	0
Event Stability Index	Aggregation "Average by time", Measure "Balance Amount", for Time Period "2 Months back"	●	0.0055	●	0.0048
	Aggregation "Average by time", Measure "Balance Amount", for Time Period "3 Months back"	●	0.0052	●	0.004
	Aggregation "Average by time", Measure "Balance Amount", for Time Period "Last 1 Month"	●	0.0057	●	0.0039
	Aggregation "Average by time", Measure "Balance Amount", for Time Period "Last 3 Months"	●	0.0055	●	0.004
	Aggregation "Average by time", Measure "Days Payment Past Due Count", for Time Period "2 Months back"	●	0.039	●	0.0138
	Aggregation "Average by time",				

The above report is created by executing a few SAS jobs in SAS® Studio

```
%dabt_build_scr_abt_wrapper(m_model_id=95,m_scoring_as_of_dt  = 31MAY2022,  m_perform_scoring_flg  = Y , m_populate_arm_flg  = Y);
%dabt_build_act_abt_wrapper(m_model_id=95,  m_scoring_as_of_dt =31MAY2022  , m_populate_arm_flg  = Y);
%csbmva_ong_mdl_performance_run(model_id=95);
```

SAS® Intelligent Decisioning

SAS® Intelligent Decisioning

Improve decision-making process

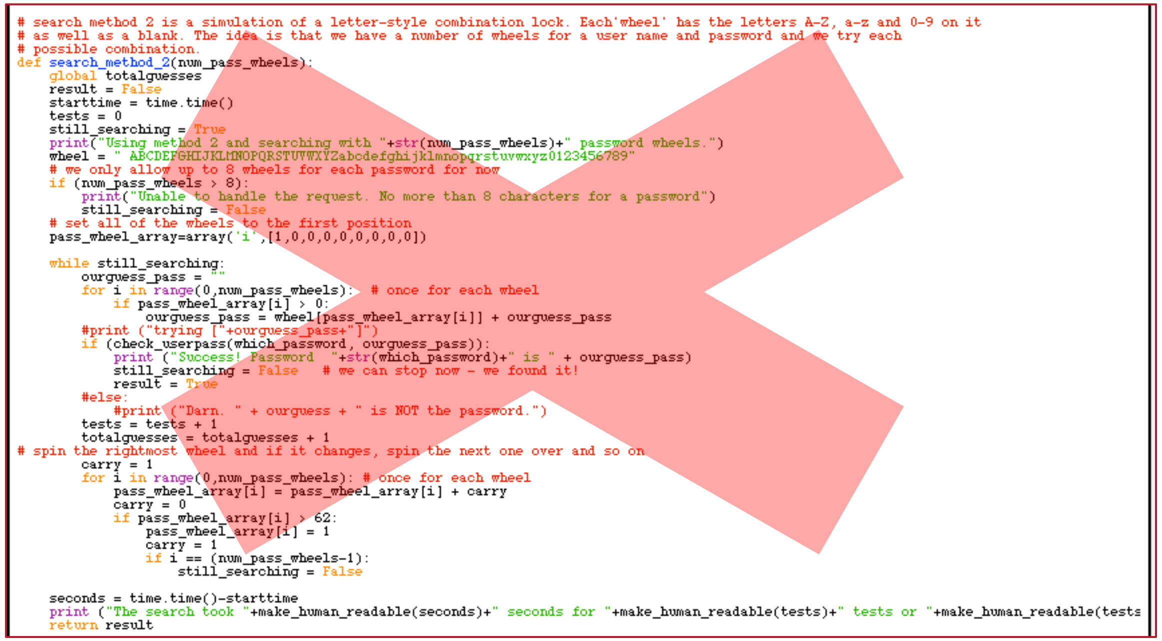
More transparent

More adaptable

More visual

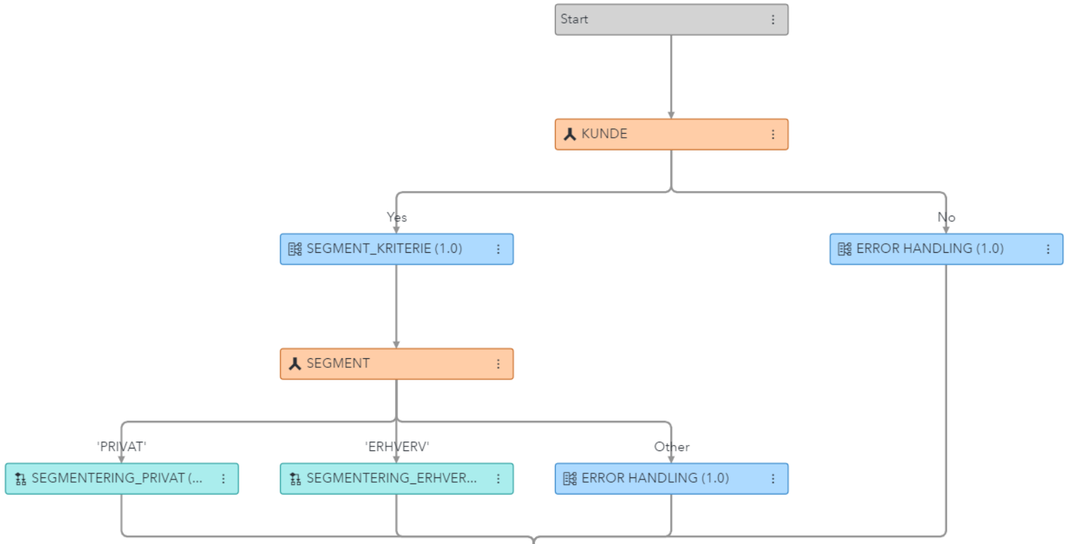
More consistent

Replace this:



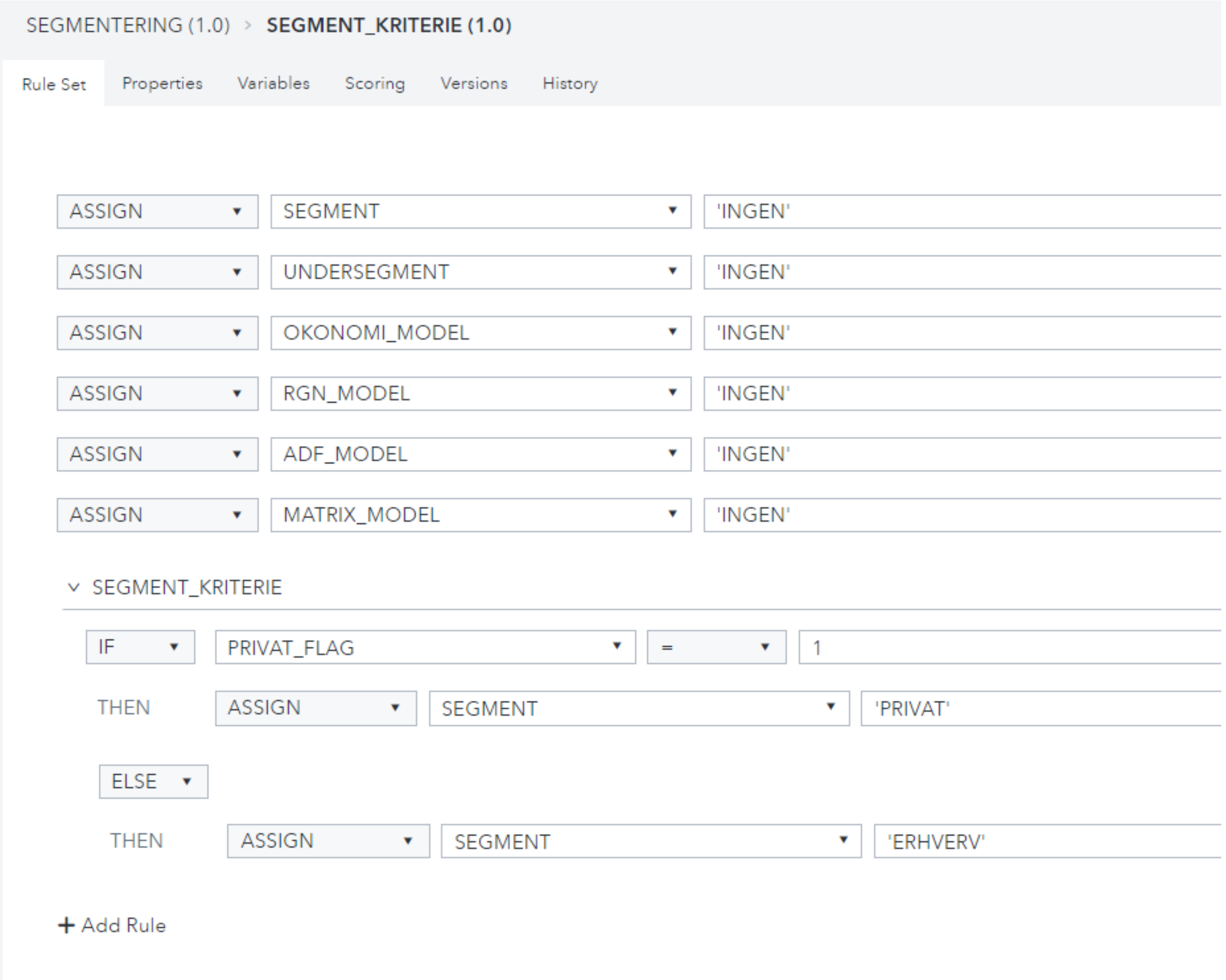
```
# search method 2 is a simulation of a letter-style combination lock. Each'wheel' has the letters A-Z, a-z and 0-9 on it
# as well as a blank. The idea is that we have a number of wheels for a user name and password and we try each
# possible combination.
def search_method_2(num_pass_wheels):
    global totalguesses
    result = False
    starttime = time.time()
    tests = 0
    still_searching = True
    print("Using method 2 and searching with "+str(num_pass_wheels)+" password wheels.")
    wheel = " ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"
    # we only allow up to 8 wheels for each password for now
    if (num_pass_wheels > 8):
        print("Unable to handle the request. No more than 8 characters for a password")
        still_searching = False
    # set all of the wheels to the first position
    pass_wheel_array=array('i',[1,0,0,0,0,0,0,0,0])

    while still_searching:
        ourguess_pass = ""
        for i in range(0,num_pass_wheels):  # once for each wheel
            if pass_wheel_array[i] > 0:
                ourguess_pass = wheel[pass_wheel_array[i]] + ourguess_pass
        #print ("trying ["+ourguess_pass+"]")
        if (check_userpass(which_password, ourguess_pass)):
            print ("Success! Password  "+str(which_password)+" is " + ourguess_pass)
            still_searching = False   # we can stop now - we found it!
            result = True
        #else:
            #print ("Darn. " + ourguess + " is NOT the password.")
        tests = tests + 1
        totalguesses = totalguesses + 1
# spin the rightmost wheel and if it changes, spin the next one over and so on
        carry = 1
        for i in range(0,num_pass_wheels): # once for each wheel
            pass_wheel_array[i] = pass_wheel_array[i] + carry
            carry = 0
            if pass_wheel_array[i] > 62:
                pass_wheel_array[i] = 1
                carry = 1
                if i == (num_pass_wheels-1):
                    still_searching = False

    seconds = time.time()-starttime
    print ("The search took "+make_human_readable(seconds)+" seconds for "+make_human_readable(tests)+" tests or "+make_human_readable(tests
    return result
```

By this:



- Automate decision-making
- Use information to make better decisions
- Apply in real time or batch

SAS® Intelligent Decisioning



SEGMENTERING (1.0) > **SEGMENT_KRITERIE (1.0)**

ASSIGN ▾ SEGMENT ▾ 'INGEN'

ASSIGN ▾ UNDERSEGMENT ▾ 'INGEN'

ASSIGN ▾ OKONOMI_MODEL ▾ 'INGEN'

ASSIGN ▾ RGN_MODEL ▾ 'INGEN'

ASSIGN ▾ ADF_MODEL ▾ 'INGEN'

ASSIGN ▾ MATRIX_MODEL ▾ 'INGEN'

∨ SEGMENT_KRITERIE

IF ▾ PRIVAT_FLAG ▾ = ▾ 1

THEN ASSIGN ▾ SEGMENT ▾ 'PRIVAT'

ELSE ▾

THEN ASSIGN ▾ SEGMENT ▾ 'ERHVERV'

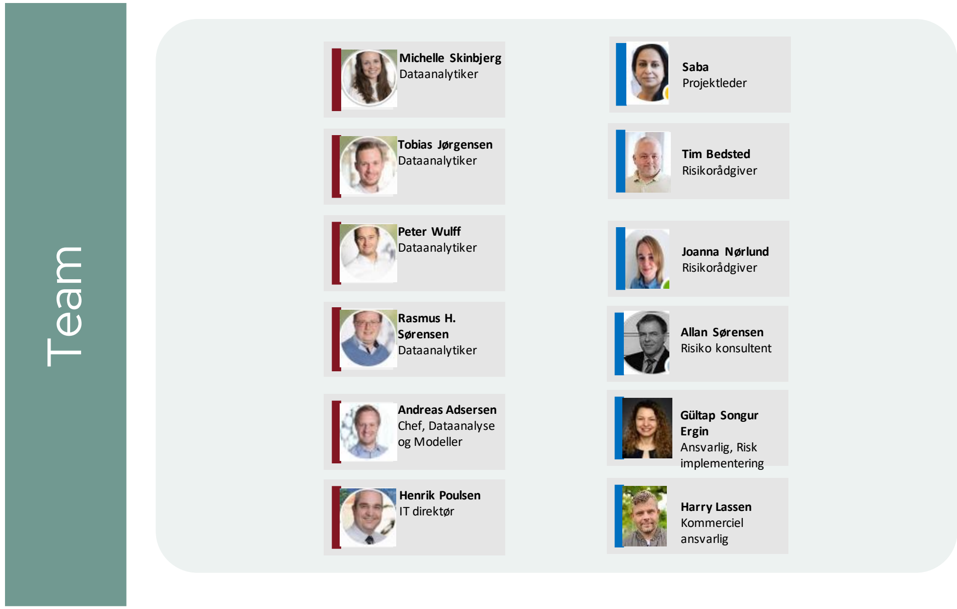
Implementation

An implementation that ensures quick start-up and training of model developers

The implementation team

The crucial thing for the team was to get full support when we encountered problems and that they were solved quickly

One team!



Michelle Skinbjerg
Dataanalytiker

Tobias Jørgensen
Dataanalytiker

Peter Wulff
Dataanalytiker

Rasmus H. Sørensen
Dataanalytiker

Andreas Adsersen
Chef, Dataanalyse og Modeller

Henrik Poulsen
IT direktør

Saba
Projektleder

Tim Bedsted
Risikorådgiver

Joanna Nørlund
Risikorådgiver

Allan Sørensen
Risiko konsultent

Gültap Songur Ergin
Ansvarlig, Risk implementering

Harry Lassen
Kommerciel ansvarlig

🟥 **AL employees** 🟦 **SAS Team**

Crucial for the analyst

1. One team, same goal and common approach to achieving the goal
2. AL has been responsible for a large part of the implementation
3. Coaching and sparring of SAS along the way
4. Open, experience-based dialogue that allows for continuous calibration

The implementation process

Gains for the model developer

- End-to-end process of the model lifecycle to develop, validate, implement and monitor risk models
- Cloud Analytic Services (CAS) can efficiently model and score millions of accounts quickly due to its parallel and distributed architecture
- A user-friendly graphical interface

Challenges and handling

- A new platform
- Only one person can be associated with an ABT and user-defined scorecards
- A user-defined scorecard must be locked (and thus cannot be changed) to move forward to Risk Management Cycle