Paper 2751-2018

# SAS® Credit Scoring for Banking – An Integrated Solution from Data Capture to Insight

Ewa Nybakk, Capgemini Norway

## ABSTRACT

The banking sector experiences increased demands related to risk assessment because of the Basel Capital Requirements. Credit modeling and scoring is an important component of estimating the capital requirement, and banks face various challenges and needs related to this modeling. SAS® Credit Scoring for Banking is an integrated solution that enables detailed analysis and improved prediction of credit risk with these challenges and needs in mind.

This paper is based on experiences gained from implementing SAS® Credit Scoring for Banking for a series of banks. We discuss the solution architecture in the context of challenges and needs related to credit modeling. The architecture overview highlights the SAS® software included in SAS® Credit Scoring for Banking and the stages at which the different software comes into play. We consider the benefits of SAS® Credit Scoring for Banking and assess its scalability with respect to additional data sources and models.

## INTRODUCTION

### RISK MANAGEMENT AND CREDIT SCORING

Following the 2008 financial crisis, banks face tougher regulation to their risk management practices. Demonstrating solid risk management practices will reduce capital requirements and enable higher investment ratios. Credit Scoring is an important element of risk management. Based on statistical models, banks assess their existing portfolio of client and accounts and estimate credit scores. Good statistical models are therefore directly profitable!

Credit Scoring activity is related to challenges. The credit analyst, who creates scoring models, often requires assistance from IT. Collecting data for various modeling and reporting purposes is a time-consuming task, meaning that the data is in danger of being outdated by the time the analyst reports on it.

SAS® Institute developed SAS® Credit Scoring for Banking with these, and other credit modeling challenges and needs, in mind.

# SAS® CREDIT SCORING FOR BANKING

## SOLUTION OVERVIEW

SAS® Credit Scoring for Banking is an integrated solution. This means that the solution contains all necessary components for credit scoring: Data extraction and aggregation, variable creation, model development and deployment to model reporting.

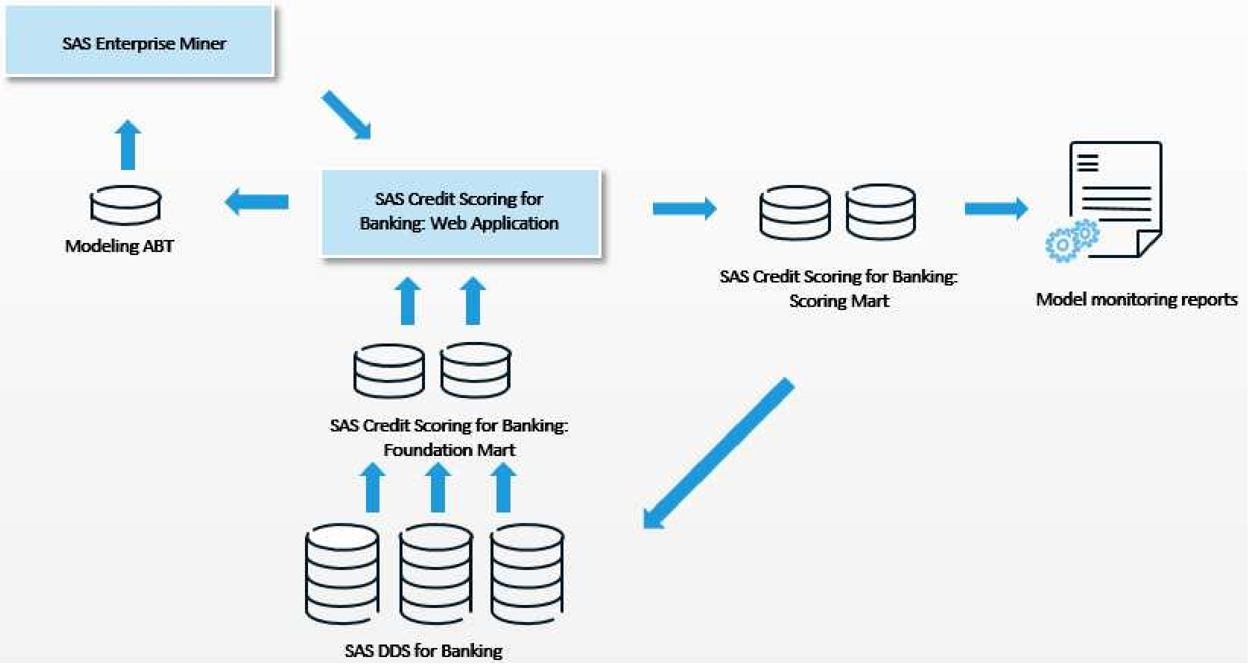

SAS Enterprise Miner
Modeling ABT
SAS Credit Scoring for Banking: Web Application
SAS Credit Scoring for Banking: Scoring Mart
Model monitoring reports
SAS Credit Scoring for Banking: Foundation Mart
SAS DDS for Banking

**Figure 1  High-level architecture overview**

Figure 1 displays a high-level architecture overview of SAS® Credit Scoring for Banking. Extract Transform Load (ETL) jobs extract relevant data for credit scoring from the data warehouse and load the data into the Foundation Mart.

The solution connects the data from the Foundation Mart using Information Maps. The credit analyst logs on to the web application to transform, aggregate and select model input in a modeling Analytical Base Table (ABT). The modeling ABT is the set of variables the credit analyst wants to use to create a scoring model.

After the ABT is defined, the credit analyst is ready to create the scoring model. The analyst can use SAS® Enterprise Miner to create statistical models with data mining, or create user-defined scorecards directly in the web application. The analyst can define scorecard details for a user-defined scorecard based on judgement or copy external scorecard details. You typically specify details by judgement when you do not have sufficient historical data to create a model.

When the modeling phase is finished and the model is ready for use, the analyst logs back onto the web application. Here she registers the model and deploys it. In the web application the analyst can choose to make the model a champion model or a challenger. The deployed model is stored as a physical file in SAS® BASE code, in a folder with the modeling ID that was assigned in the model registration process.

A scoring job, which is stored as a macro program, invokes the scoring process. Credit Scoring for Banking stores the results in a scoring ABT in the Scoring Mart and relevant model information is stored in the Detail Data Store through a set of ETL jobs.

## SAS® SOFTWARE COMPONENTS

SAS® Credit Scoring for Banking uses various components from the SAS® portfolio:

- SAS® Detailed Data Store for Banking – Data warehouse
- SAS® Information Map Studio - Connecting data
- SAS® DI Studio - ETL Foundation Mart and Scoring Mart jobs
- SAS® Macro language and SQL procedures - for solution-specific functions
- SAS® Enterprise Miner - for model development

# CREDIT SCORING FOR BANKING - MODIFICATIONS

## THE PROJECT

Capgemini Norway and SAS® Institute Norway has worked together to implement SAS® Credit Scoring for Banking for three banks.

The objective is to provide the banks with an improved analytics infrastructure to enable better risk management.

From August 2015 to date, we implemented the following components:

- SAS® Detail Data Store for Banking
- SAS® Regulatory Risk Management
- Application Score Models
- A data mart for Customer Relationship Management
- SAS® Credit Scoring for Banking

The project:

- Split the delivery into modules, with the implementation of the Detail Data Store being the largest module.
- Implemented the same solution for all three banks, with small customization between the banks.
- Implemented, tested and delivered the different modules in parallel, where we delivered a module one bank at a time. This method was beneficial for a large project with several deliveries for different banks. Rather than completing all modules for one bank before moving on to the next, we achieved a more continuous delivery.

## SAS® CREDIT SCORING FOR BANKING – PROJECT SCOPE

We implemented of SAS® Credit Scoring for Banking in three stages:

1. Ensured that the implementation meets functional requirements.
2. Expanded the Foundation Mart for further detailed data modeling.
3. Implemented two existing credit models in the solution.

The existing models are behavior score models for Probability of Default; one for Corporate Customers, and one for Retail Customers. The models, developed by a third party prior to the project, were available in documentation form.

There are several benefits to including existing models. With existing models available in Credit Scoring for Banking, the banks can:

- Use their existing model as a champion model, a baseline to measure new models.
- Take advantage of standard reporting and detailed data from the scoring mart on existing models, without disruption of operations.
- Import variables from existing models into new models.

To include the existing models in the solution, it was necessary to modify several steps of the default solution, from data extraction to model registration.

## Data Extraction

The banks' existing score models require some data sources that are not a part of the Credit Scoring for Banking Foundation Mart. After the data sources were included in the Detail Data Store, it was necessary to include these additional data sources (tables, columns) in the Foundation Mart.

We achieved this by:

1. Modifying existing and adding new ETL jobs in the Foundation Mart. The ETL jobs in the Foundation Mart consist of standard Data Integration transformations, and transformations with user written code. To ensure that additional columns were loaded properly in the Foundation Mart we mapped the data sources into the standard transformations, and modified the user written transformations. Knowledge of the SQL procedure syntax and macro functions was crucial in this step. Figure 2 and Figure 3 show examples of modification.

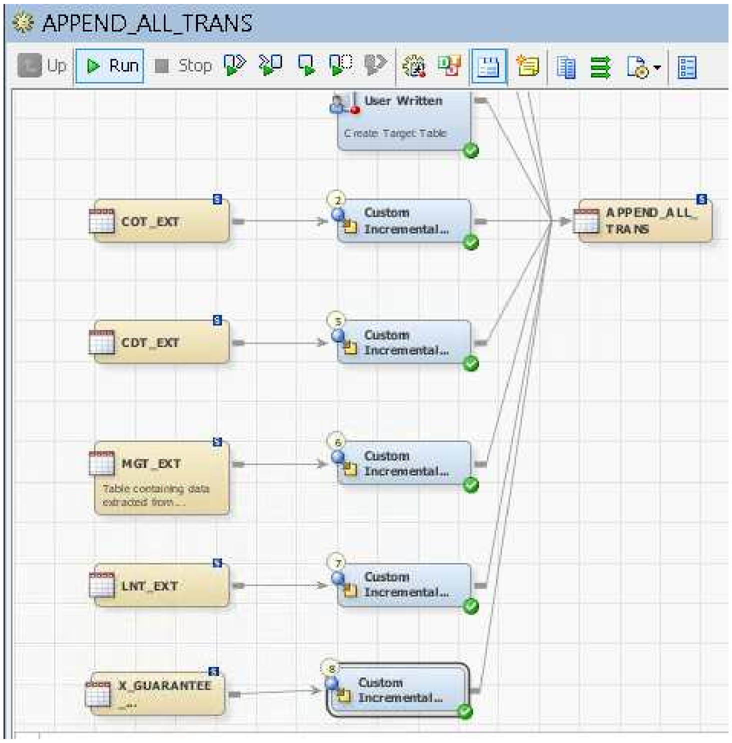
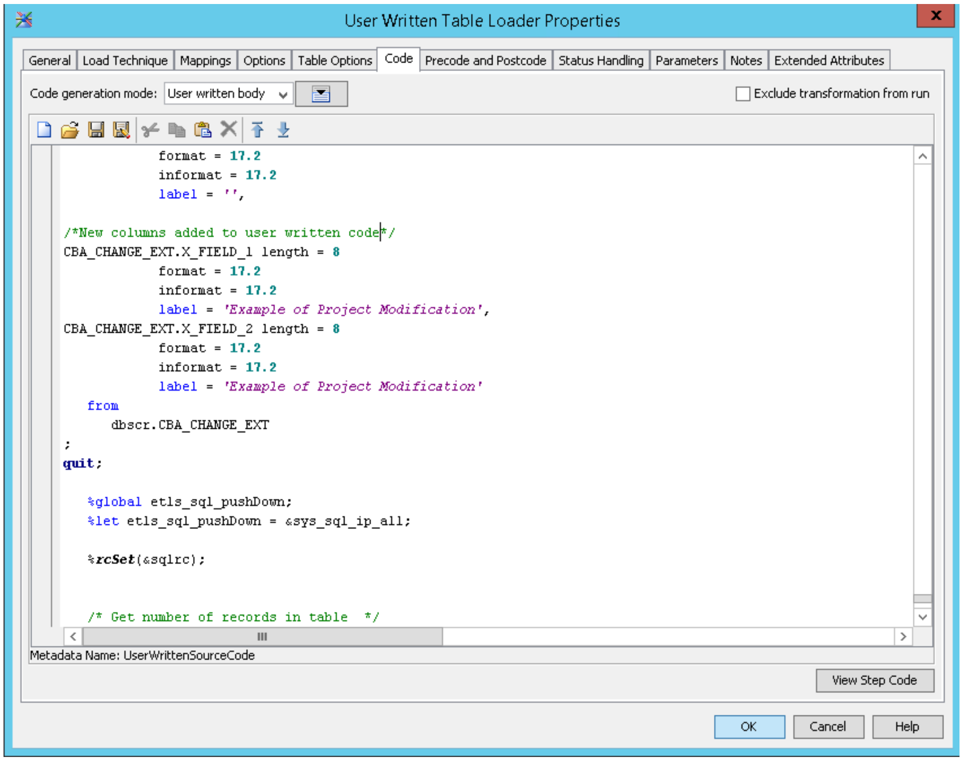


**Figure 2 Example - Modification of standard ETL job**



Code generation mode: User written body

☐ Exclude transformation from run

```
            format = 17.2
            informat = 17.2
            label = '',

/*New columns added to user written code*/
CBA_CHANGE_EXT.X_FIELD_1 length = 8
            format = 17.2
            informat = 17.2
            label = 'Example of Project Modification',
CBA_CHANGE_EXT.X_FIELD_2 length = 8
            format = 17.2
            informat = 17.2
            label = 'Example of Project Modification'
   from
      dbscr.CBA_CHANGE_EXT
;
quit;

   %global etls_sql_pushDown;
   %let etls_sql_pushDown = &sys_sql_ip_all;

   %rcSet(&sqlrc);


   /* Get number of records in table  */
```

Metadata Name: UserWrittenSourceCode

**Figure 3 – Modification of user written code in ETL job**

2. The tables in the Foundation Mart are interrelated (e.g. an account is related to a customer), but in order to use the data together it is necessary to logically combine them. That is, apply a set of code instructions that specify how the data is connected. Credit Scoring for Banking's Web Application uses Information Maps, a business metadata layer, to query the Foundation Mart when it defines input variables. The credit analyst is shielded from the complex data in the Foundation Mart, but able to build variables for modeling (e.g. summarize a customer's total balance for all active accounts in the last month). In order to build input variables using new data sources, we modified existing Information Maps and added new Information Maps in Information Map Studio, shown in Figure 4.

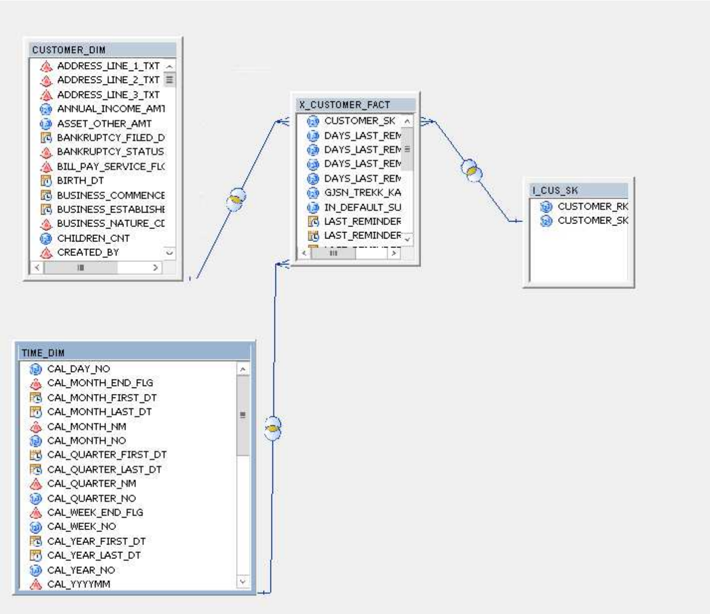
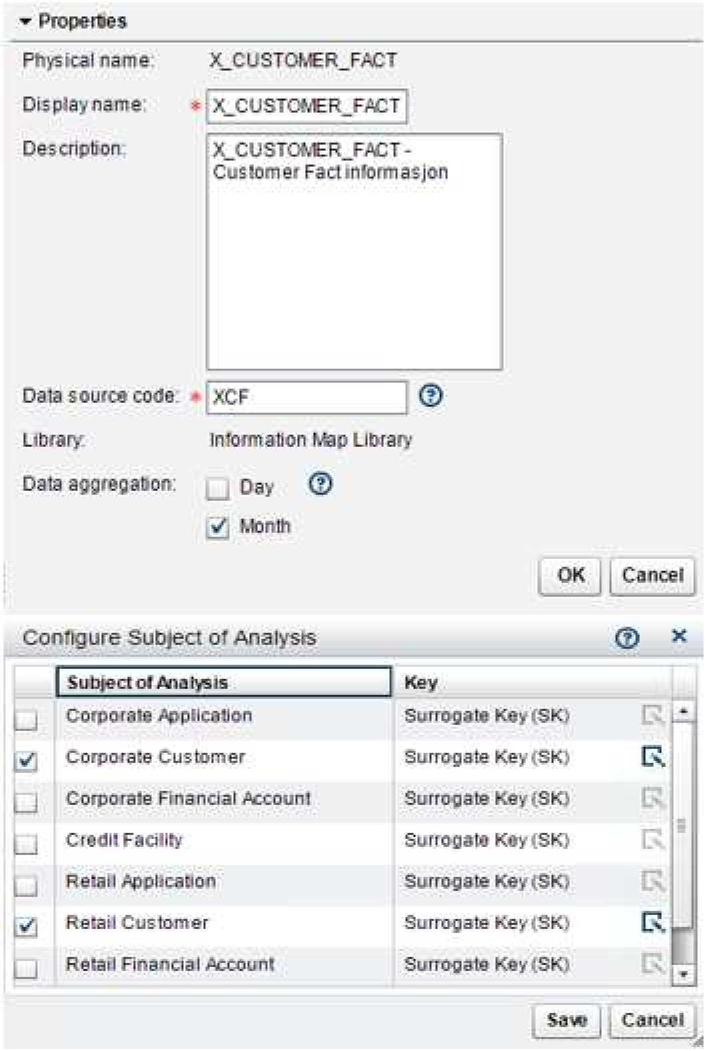


**Figure 4 Example – New data source information map in SAS Information Map Studio. The circles indicate the column that connects the tables.**

3. When we had loaded the additional data sources in the Foundation Mart and defined the Information Maps that connected the data together, we made the data available for variable creation. That is, we added and configured the data sources in the solution's web application, as shown in Figure 5.



Physical name: X_CUSTOMER_FACT

Display name: * X_CUSTOMER_FACT

Description: X_CUSTOMER_FACT - Customer Fact informasjon

Data source code: * XCF

Library: Information Map Library

Data aggregation: □ Day ☑ Month

| | Subject of Analysis | Key |
|---|---|---|
| □ | Corporate Application | Surrogate Key (SK) |
| ☑ | Corporate Customer | Surrogate Key (SK) |
| □ | Corporate Financial Account | Surrogate Key (SK) |
| □ | Credit Facility | Surrogate Key (SK) |
| □ | Retail Application | Surrogate Key (SK) |
| ☑ | Retail Customer | Surrogate Key (SK) |
| □ | Retail Financial Account | Surrogate Key (SK) |

**Figure 5 Example - Configuration of a new data source in Credit Scoring for Banking's Web Application**

## Variable Creation

We were ready to define the existing model's input when we had performed Steps 1-3 in the Data Extraction section. For each model, we defined a modeling ABT in the web application, with the model's relevant input variables. The modeling ABT was then built to create a physical file. We tested and troubleshot the modeling ABT by looking at the calculated variable results and analyzing the ABT building log. Analyzing the ABT building log requires an understanding of the SQL procedure syntax, and the data sources you have used to define the model variables.

## Model Inclusion

We coded the existing models in SAS® BASE prior to including them in Credit Scoring for Banking. Model results and input variables from the existing data warehouse were available. By combining the SAS® models with the existing input variables, we could validate the model results against results from an existing model for the customer portfolio. This approach reduced the risk of error in model implementation. This meant that the main task of including the existing models was deriving the model input variables in the web solution.

To include a model in SAS® Credit Scoring for Banking, it is necessary to register the model in the web application.

We were ready to register the model in SAS Credit Scoring for Banking when:

1. The necessary data sources were included in the Foundation Mart.
2. The modeling ABT was defined.

There are three alternate ways of including existing models in the solution:

1. Create a user defined model.
2. Create dummy models in Enterprise Miner.
3. Import external models using the import macro %aa_model_register.

In the project, we included the models with alternative 2 and alternative 3.

When you create models in Enterprise Miner, Enterprise Miner saves the model code as a .sas-file. Including the existing model in SAS® Credit Scoring for Banking by alternative 2 implies replacing the dummy Enterprise Miner code with the developed BASE code. With this method, it is important that the model variables are significant variables in the dummy model. The web solution will only connect significant variables to a scoring template.

Through trial and error, we found that alternative 3 was the most appropriate solution for the project. We implemented the models in parallel with project activities related to the data warehouse, meaning that some data sources were scarce. In this phase, it was challenging to ensure that all input variables for the existing model were statistically significant in the simple dummy model.

The model code was registered in Credit Scoring for Banking using the %aa_model_register() macro. The macro input parameters are listed in Table 1, while an example of the macro run is shown in Figure 6.

| Parameter | Description |
|---|---|
| *modelname* | The model name that is shown in the Credit Scoring for Banking web application |
| *modeldesc* | The description that is shown in the Credit Scoring for Banking web application. Included within %nrbquote() |
| *register* | Y/N Flag, indicating whether to register the model |
| *mrPath* | Points to the metadata folder for the credit analyst who wants to register the model. Included within %nrbquote() |
| *spk* | Y/N flag, indicating whether a .spk file should be created |
| *spkfolder* | Points to the physical path where the .spk is saved. Depends on the value of the *spk* parameter |
| *data* | Refers to the model's modeling ABT. Libname must be included |
| *target* | Name of the target variable in the model |
| *level* | Level of the target variable. Binary/Nominal |
| *miningalgorithm* | Algorithm used for modeling. Examples: Linear Regression, Logistic Regression |
| *scorecodefile* | Points to the physical location where the model score code is located. A .sas file |

**Table 1 - %aa_model_register – import parameters**

```
%aa_model_register(modelname=credit_balance,
modeldesc=%nrbquote(PD Model, Retail Customers),
register=Y,
mrPath=%NRBQUOTE(/User Folders/sasuser/My Folder/),
spk=N,
spkfolder=c:\temp\,
data=cs_mdl.pd_model, /*Name of the modeling ABT*/
target=TGT_VAR_DFT, /*Name of the target variable*/
level= Binary, /*Level of the target variable*/
miningalgorithm=regression,
scorecodefile=C:\Users\sasdemo\score_code.sas);
```

**Figure 6 – Credit Scoring import macro, example**

## LESSONS LEARNED

We implemented SAS® Credit Scoring for Banking in parallel with the Detail Data Store. The main benefit of parallelization was the reduced time to value for our customer: The different project modules were available to the customer as we delivered them, rather than having to wait a long time to access all elements of the project at once.

The main points are summarized below:

1. A general observation: SAS® Credit Scoring for Banking uses different components of the SAS® portfolio. For modification of the standard solution, troubleshooting and error handling, it is a great advantage to be familiar with these components beforehand, with an emphasis on SQL procedures.
2. SAS® Credit Scoring for Banking has a set of design principles to which the data sources must conform. These principles are important to keep in mind during implementation of the Detail Data Store and in modification of the Credit Scoring solution. We were able to adjust the logic in the Detail Data Store because we implemented the Detail Data Store in parallel with Credit Scoring for Banking, but it was a time consuming task. If Credit Scoring for Banking is implemented on top of an existing data warehouse, this task is even greater.
3. In general, there are many ways to include new data sources for modeling. One should allow for a thorough analysis and testing phase when using the data for modeling. This was a challenging and time consuming part of the project.
4. For inclusion of existing models, it is important that the model variables are significant variables in the dummy model.
5. You cannot alter model variables once you register a model for scoring. This requires registration of a new model.

Learning points 3, 4 and 5 proved to be the most time consuming and challenging parts of implementing existing models in the solution. Scarce documentation for the existing models meant that it was unclear how to derive variables, and we needed to work through this using trial and error.

If we discovered that the variable definition was wrong, it was in most cases possible to redefine the variables through the same data source. In some cases, we needed to make the data available through a different source. With a data warehouse in an implementation and testing phase, it was challenging to get significant model variables (i.e. missing data). This required a workaround, where we modified the modeling ABT.

In order to test the scoring results of the existing models we were required to register a model for scoring, even though the model input variables were in development. When testing proved the need to change an input variable, we needed to redo points 4 and 5 and, in the worst cases, point 3 as well.

## SAS® CREDIT SCORING FOR BANKING - BENEFITS

We find the benefits of implementing SAS® Credit Scoring for Banking in all layers of the business:

1. The IT Department: Less labor-intensive data management related to credit scoring. Work task reduced from repeatedly developing in-house tools for data preparation to monitoring scheduled jobs in the solution's Foundation Mart and Scoring Mart.
2. The Credit Analyst: SAS® Credit Scoring for Banking adds flexibility to the credit analyst's work tasks. Modeling data is readily available through a user-friendly graphical interface, and the analyst can easily create segmented models for increased accuracy. The ability to share variables and models with other analysts, combined with the solutions standard reporting on statistical variables, provides increased insight in the bank's credit models and the underlying data.
3. The Bank: The solution provides Basel II compliant back testing and model validation reports. With in-house modeling, the bank retains the modeling IP, knowledge and best practices. A reduction in time to value provides a basis for decision-making on fresh data. With SAS Credit Scoring for Banking, small banks can maintain an analytics platform usually reserved for large enterprises. The ability to verify the model accuracy and its strengths and weaknesses reduces the model risk. The increased insight in credit models, with lower model risk, allows for better business strategies.
4. The Customer: SAS® Credit Scoring for Banking also benefits the customers. Accurate credit models, split into segments for better prediction, provides better customer products. For some customers, an increase in model accuracy may be the reason their loan is granted.

## CONCLUSION

We have seen how SAS® Credit Scoring for Banking meets the needs and challenges faced by banking institutions, and its main technical components. For the credit analyst, there is a great benefit in creating models in a user-friendly graphical interface, where relevant data is available. The ability to share model variables and validate models provides a better understanding of the customer portfolio. For the bank, we have seen how a reduced duration of the analytical life cycle strengthens the basis for decision making, providing better business strategies and products. This directly benefits the customer.

In addition, we have seen how a project has modified the standard Credit Scoring for Banking solution to include existing scoring models, and lessons learned through this work. There are many benefits to including existing models, but it is not necessarily a straightforward task. Particular effort should be made in the following two steps:

1. Ensuring that the Detail Data Store meets the solution's design principles.
2. Testing and verification of model input variables prior to model registration.

## RECOMMENDED READING

- *Base SAS® Procedures Guide*
- *SAS® For Dummies®*
- *SAS® Credit Scoring for Banking – Admin Guide*
- *SAS® Credit Scoring for Banking – User Guide*
- *SAS® Credit Scoring for Banking – Data Reference*

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Ewa Nybakk
Capgemini Norway
Ewa.nybakk@capgemini.com