

Applying models with scoring

S. Bayerl¹, T. Bollinger², Chr. Schommer¹

¹*IBM Deutschland GmbH – SW BI Solution Consulting & Services*

²*IBM Deutschland Entwicklung GmbH – Data Management Development*

Abstract

"Scoring", in general, is defined as the usage of mining models - based on historical data - for classification or segmentation of new items. For example: if the historical data consist of classified customers, then we can use the model for the prediction of the behaviour of a new customer. Scoring offers novel ways to exploit the power of data mining models in everyday business activities, and proliferate mining applications to users who are not educated in mining.

In this paper, we present a) the generic scoring process b) its technical implementation, and c) an example of how scoring can be integrated in a real application.

The **generic process** consists of three steps: The mining models are learned first, then they are transferred into the application database, and finally the models are applied to the data loaded in that database. Arguments for the necessity of such a mining improvement are collected.

IBM DB2 Intelligent Miner Scoring (IM Scoring) is the first **technical implementation** of scoring. It is based on the emerging open-standard for mining models (*Predictive Model Markup Language - PMML*), and the mining extensions for SQL. Implementation issues are discussed, as well as problems that come along with its integration into operational applications.

The article closes with the description of a **sample application**, the integration of scoring into a call center environment. A discussion of the scoring method concludes this article.

1 Introduction

“Scoring” in general is defined as the classification (“rating”) or segmentation (“grouping”) of new items by the usage of mining models based on historical data. If we assume that the historical data consist of classified customers then we can use the model based on these customers for the prediction of the behavior of a new customer. The scoring result entails immediately a suggestion how to treat the new customer: It is reliable to argue that the new customer has similar properties or requests as the corresponding group of customers in which the new customer fits. Figure 1 explains the scoring process very roughly: Given a model of a data mining product, the model can be exported to a different environment where it can be applied to different data sets.

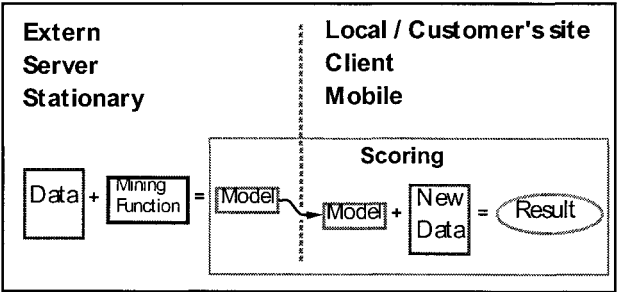


Figure 1: Scoring Process

1.1 Application scenarios

A lot of application scenarios are suited for scoring. We have collected here some typical ones:

Area/Branch	Application
Banking	In a call center run proven model for the selection of gold cards or additional accounts
Telco	Run model for churn regularly for customers who have to renew their contract
Retail	During the visit of a customer run the model for similar customers groups for a user-oriented offering
Insurance	Use model for the selection of new insurance products during a consultancy
Manufacturing	Run model describing the mistakenly manufactured product to test the new products
Health Care	Use the evaluated data of all earlier patients to select the most reasonable treatment
E-Commerce	Select adequate products for users visiting your online shop by a model having evaluated similar users

In **Banking** you may run a mining model in a call center for answering the question which products, e.g. which particular credit cards, would be interesting for your new customer. The information which will be given by the customer online is the key for the suggestion of the best suitable product. By running these data against the model a highly plausible selection will be provided by the system.

A well known problem in the area of **Telecommunication** is the retention of customers. In Telco the "disloyal" customers are called "churners" and they can be predicted by looking at the behaviour of the existing customers. It is very important to watch the behaviour of customers who are going to renew their contracts and who are likely to churn. Through scoring this information is generated in an event-driven manner and nearly effortless.

New marketing facilities in **retail** as customer-profiled product offerings, e.g. displayed on the shopping trolley, help to increase the revenue and satisfaction of the customers.

The diversity of products within the **insurance** field is high. Even the expert may not be able to offer a customer the optimal contract, moreover if time is the question. In this case a huge data set containing the contracts of all former customers will support the decision. Building a model over these data and using scoring enables the agent to score the most reasonable and acceptable contract.

In **manufacturing** the following deployment of scoring is useful: A comparison of the features of the produced item with similarities of rejected items indicates whether the item must be deselected or tested in detail. Standard test environments detect only items that overrun certain threshold values, but not items with valid values, but irregular combinations of values.

In **Health Care** scoring can support capacity planning by looking at the capacity model derived by mining. Or, scoring can give indications for the best treatment of a patient: The model built with evaluated data of all earlier patients leads to the most reasonable treatment. The huge amounts of information condensed by a scoring model plus their fast accessibility are the reasons why scoring in health care can pay in a short time [4].

We should not forget the tremendous possibilities of **e-commerce** and **web-applications** where scoring fits smoothly into the business request of fast information delivery. For example, you may use the buying model of former online customers for special on-line offerings of new customers. The click stream behavior plus the model implies a rating for profitability and a suggestion of typical other interests and products. That information can be used to channel the selection of displayed pages.

1.2 Prerequisites

This paper assumes a basic understanding in data mining as it could be acquired by [1], or by reading the introductory chapters of [2].

The application has been implemented with DB2 Intelligent Miner for Data for the modeling and DB2 Intelligent Miner Scoring for the scoring part. A short

overview of the product can be found in [<http://www.ibm.com/software/data/iminer/fordata/>]. Developers will find technical details of DB2 Intelligent Miner Scoring in [3].

2 The generic scoring process

The scoring process can be seen as a generic procedure not dependent on special database management systems or mining functions (see Figure 2):

First a training model on the historic data is generated. This part is done by a data mining expert who knows how to adapt the selected algorithm for your business problem and your particular data. The model is exported in Predictive Model Markup Language (PMML) format based on Extended Markup Language, short XML, into a flat file.

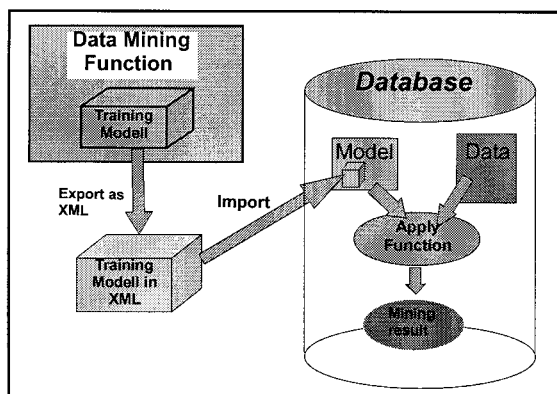


Figure 2: The Generic Scoring Process

Then the model can be imported into a database table by several parties which will apply it to their data. The application of the model (“Apply Function”) will be shown in more detail in Figure 3.

How will the model be applied? Let us assume that the model used in our example is a clustering or segmentation model whose active fields are “age” and “salary”. The records of the data table to which the model is applied are transformed by a scoring function, called “DM_app!Data” here, to an XML string containing field name/value pairs.

The column names do not have to be identical to the ones used in the model. This is because the field names are specified explicitly in the XML representation of the record.

Another scoring function, called “DM_applyClusModel” here, applies the clustering model to the XML string representation of the input record and computes the cluster id and additional information.

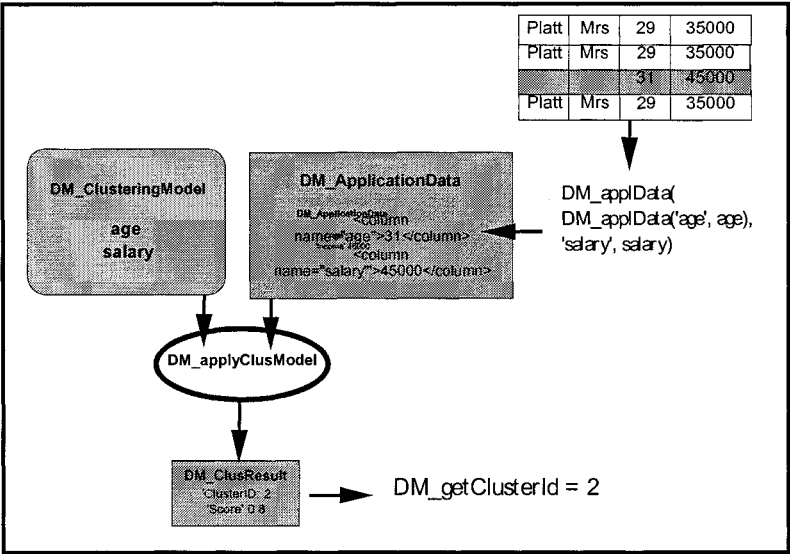


Figure 3: Apply process

In our case it is the cluster with number 2, maybe a cluster in which a high percentage of customers with gold cards could be found.

3 IBM DB2 Intelligent Miner Scoring - An implementation of Scoring

IBM DB2 Intelligent Miner Scoring (IM Scoring) provides a set of user defined functions (UDFs) that allow to perform all the operations of the scoring process. In particular it allows to apply models directly in the database through SQL statements. Providing an SQL interface has the advantage that it facilitates the integration into applications, as it does not depend on the programming language the application has been written in. Nearly every important programming language provides an interface to SQL (for instance JDBC for Java)

In the following we will briefly discuss how the scoring process with IM Scoring looks like (see Figure 4).

First you have to export the models from DB2 Intelligent Miner for Data (IM for Data). You can export them in the IM for Data or the PMML format. If your model is in the IM-for-Data format IM Scoring provides tools to convert them to PMML. Once the models are exported they can be imported into the database. For the import IM Scoring provides a set of UDFs as well.

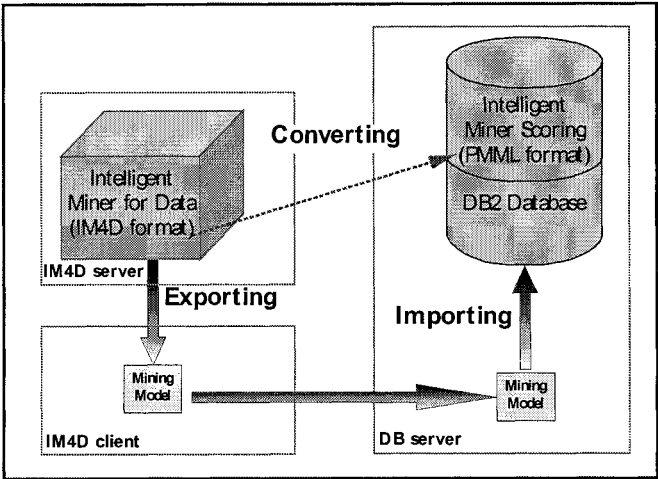


Figure 4 : Applying a model

The import process itself can be described as follows: An IM for Data model is converted to the PMML format. Then, the PMML model is parsed by DB2 and the model header constructed by the usage of the parsed information. The next step is the conversion of the PMML model to Unicode format (UTF-16 Big Endian). The model header is appended. And finally, the model is stored as a binary large object in the database (see Figure 5).

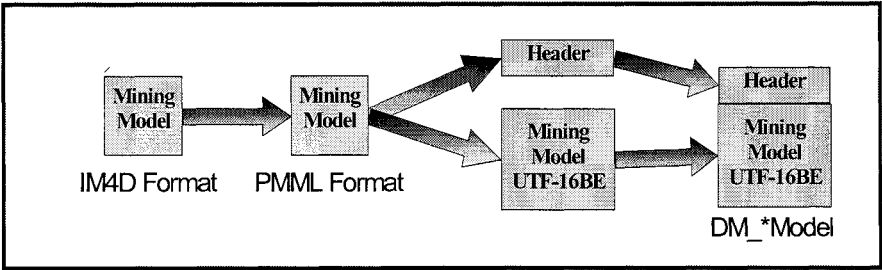


Figure 5: Import Process

Once the models are imported and stored in a table we can apply them to data that resides in the database or comes from the application. IM Scoring provides UDFs for creating the XML-representation of the input records where the models are applied to. For each mining function (like clustering or classification) specific apply-functions are available. As the result of the application of a model is a structured value, other UDFs allow to access the values of its specific attributes like cluster id and score for clustering result values.

The following example shows that the SQL statements behind the scoring process are quite simple.

```
INSERT INTO IDMMX.ClusterModels values
( 'DemoBanking',
IDMMX.DM_impClusFile('/tmp/demoBanking.mdl') );

SELECT
    d.name, d.age,
    IDMMX.DM_getClusterId(
        IDMMX.DM_applyClusModel( cm.model,
        IDMMX.DM_applData(
            IDMMX.DM_applData('age',d.age),
            'salary', d.salary)))
FROM ClusterModels cm, MyData d
WHERE cm.modelname='DemoBanking';
```

The first SQL statement imports the clustering result file `"/tmp/demoBanking.mdl"` into the `IDMMX.ClusterModels` table under the name "DemoBanking". The table `IDMMX.ClusterModels` contains two columns, one for the name of the model and one for the model itself. The user-defined function `IDMMX.DM_impClusFile` performs this import. `IDMMX` is the schema for all user-defined functions and types defined by IM Scoring.

The second SQL statement applies this model to the records of the `MyData` table and returns the cluster ID for each record. The XML string representing the input record is built with the `IDMMX.DM_applData` function. The SQL constants `'age'` and `'salary'` are the names of the active fields in the model; `d.age` and `d.salary` are the values for these fields retrieved from the `MyData` table. As the field names are represented explicitly in these XML-strings the column names in the data table do not have to match the names of the fields in the models. Furthermore, instead of values coming from a table we could have used SQL constants. This means that the data for the input record does not have to be stored in the database.

The `IDMMX.DM_applyClusModel` UDF takes this XML string and the clustering model `"cm.model"` as an argument and returns a clustering apply result. The UDF `IDMMX.DM_getClusterId` is applied to this result and it returns the cluster ID.

4 Sample application: Using scoring in a call center

What can a scoring application look like? Let us assume you are working in a call center of a bank, and you want to offer a customer a particular credit card, say a gold card. The question is when such an offering is acceptable or plausible. In a browser interface (see Figure 6), you may see a text field where you can type in the customer ID, if the respective customer is known to you. Or, if not,

you may select immediately "o.k." and feed the system with some of the facts later on.



Figure 6: Call Center – Enter customer ID

If the customer's data are available they will be displayed, otherwise you may select or specify features of the customer by yourself (see Figure 7).

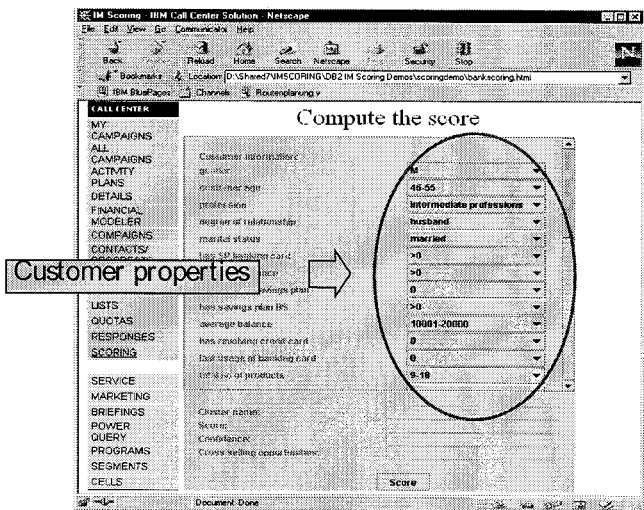


Figure 7: Call Center – Properties of the customer

Now you start the scoring run by pushing the “Score”-button.

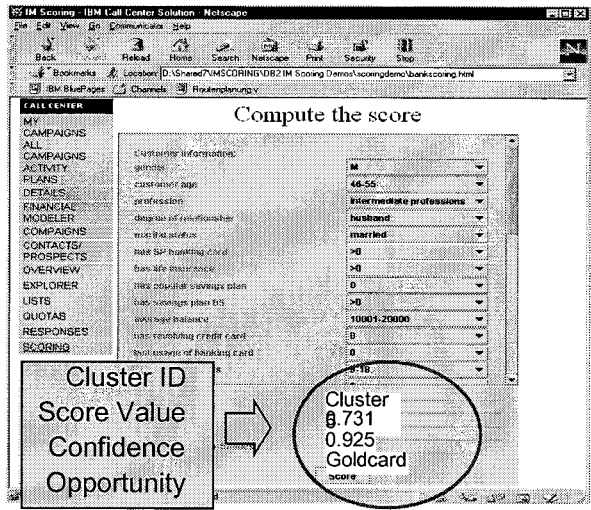


Figure 8: Call Center – The scoring result

This triggers the application of the model to the customer's data. The cluster id where the customer fits in best is returned. Based on this cluster id the call center application computes the products to offer to this customer. In this case it is a gold card (see Figure 8).

In case the customer provides new information, the call center agent can change it in this window and re-compute the score. This makes it possible for the call center agent to react on the most recent changes.

This sample application has been realized with DB2 Intelligent Miner Scoring. It is written in Java as an applet. The database access is realized through JDBC.

5 Discussion

The standard way for applying models to data is to run a model on a table and store the scores in an additional column. This is what you can do with standard mining tools like DB2 Intelligent Miner for Data. With scoring tools, like DB2 Intelligent Miner Scoring, you are much more flexible. You can apply models to single records dynamically, which is very important in scenarios where the information changes frequently and the exploitation of new information is important. Scoring can be integrated in operational applications easily with the SQL interface. The update of models is facilitated as well, as you need to exchange the models only and not the calculated score values, which can be huge amounts of data.

6 Conclusion

Scoring supports model deployment so that the real scoring procedure can be undertaken without having installed the mining product. Moreover, scoring facilitates the usage so that even a non-expert in data mining can manage the deployment task. End users can take advantage of the scoring results without having to be educated in data mining and trained in the mining product.

References

- [1] Jan H. Witten, Eibe Frank, *Data Mining*, Morgan Kaufmann: San Francisco, California 2000
- [2] Michael J.A. Berry, Gordon S. Linoff, *Mastering Data Mining*, Wiley& Sons: New York et al., 2000, Part 1 and 2.
- [3] IBM DB2 Intelligent Miner Scoring, Administration and Programming for DB2, Version 7.1, 2002.
- [4] Schommer, C.: *PASIWEL: Patient Scoring in a wireless environment*, Proceedings of the 6th World Multi-Conference on Systemic, Cybernetics and Informatics, SCI 2002, July 14 - 18, 2002, Orlando, Florida, USA.