Fast and Easy Delivery of Data Mining Insights to Reporting Systems

Ruben Pulido, Christoph Sieb

rpulido@de.ibm.com, christoph.sieb@de.ibm.com

Abstract: During the last decade data mining and predictive tools have evolved to a mature level. Many companies and organizations use those techniques to derive advanced insight from their company data. Today, mining experts analyze the data and manually deliver the results to a rather small number of decision makers. To make data mining insights available to a broader audience, manual delivery is not sufficient anymore. Suitable delivery vehicles are reporting systems, which are well known to business users. However, in most cases there is no straight forward delivery path. Much manual work and knowledge is necessary to make the insights consumable by reporting systems.

In this work we propose a system that allows fast and easy delivery of data mining insights to report servers. The system automatically analyzes and transforms mining results, generates reports and deploys them to the report server.

1 Introduction

Data mining is nowadays widely spread across enterprises, as a technique to derive valuable insight from large amounts of business data. This insight is then used to support the decision making process. In the last decades, research has mostly focused on generating data mining results rather than delivering them to a broad audience in an easy consummable way. For instance, in the retail sector, a mining based customer segmentation, combined with a customized product affinity analysis allows fine grained decision support even for front line employees. The employee can not only focus on the suitable customer group but also on the products and product goups she is responsible for. Most important however, is that employees are provided with this information in an easy consumable way. In most companies the enterprise reporting system represents the most appropriate way delivering those insights. Employees can easily access reports through their web browser and are familiar with the system interaction.

However, bringing this insight from the data-mining-expert-tool to the reporting front end, is not straight forward. Reporting systems usually access data stored in relational tables. Data mining results, however, are very different from flat table structures. They are mostly stored in hierarchical ways in large documents (e.g. standardized PMML [PMM] format. Since most of today's BI systems cannot directly consume data mining models, the mining insight must first be transformed to a form that is consummable by these systems.

In the following section we show related approaches. In section 3, we illustrate, how the

proposed deployment system is used to deliver insightful data mining reports. In section 4 we describe the uderlying architecture of the system and in section 5 we shortly outline products which are suitable in the context of the proposed deployment system.

2 Related approaches

Few vendors provide dedicated report design tools in which a report designer can manually create mining insight reports. Vendors, not providing dedicated tools force the user to transform mining insight into a form consumable by the BI system. Further, deep mining knowledge is required to create mining reports with rather general report tools. The creation of such reports is a tedious task and changes in the underlying data result in long lasting manual changes. Further, the task of transforming the mining insight, creating the reports and meta information requires deep knowledge of the involved tools. All those requirements lead to several people accros the enterprise beeing involved.

Known solutions are based on exporting images generated within the mining tool. However, this is a very static and non-interactive way. Further, this solution does not provide automatic deployment of the mining insight. Few tools allow visualizing standardized mining models natively. This approach is however not very flexible since it restricts the visualization of data mining insight to predefined graphics [IBM]. Microstrategy also provides basic native PMML support. However, the reports are assumed to be designed by a report designer [Mic], and not automatically generated by the system.

3 User interaction

The system we propose allows fast and easy delivery of data mining insights into reporting systems. The steps required to make those insights available in the reporting front end are described in this section.

A data mining expert first selects the most appropriate data mining method to solve the business problem he is facing. He then locates the business data available and explores its structure. After preparing the data to be used as input for a data mining method, the data mining expert can choose among an unlimited combination of available agorithms and settings. This is normally an iterative process, in which the data mining expert performs several data transformations and creates multiple data mining models and tests them. The data mining expert then visualizes the data mining models using visualization capabilities of the data mining tool, as shown in figure 1. Once the data mining expert is satisfied with one of the created data mining models, this model is ready to be put into production. It is ready to be integrated as part of an automated business process, or it can be deployed to a reporting system where employees can access it using the BI front end tool.

With the system presented in this article no additional people with reporting skills are needed to create the report metamodel and report specifications. Rather, the data mining

Figure 1: Visualization of a clustering model using the data mining tool

expert can directly deploy the data mining insight by following the steps, illustrated in figure 2.

(a) Enter the url where the reporting server can be accessed.

(b) Enter the credentials needed to authenticate against the reporting server.

(c) Specify the location (i.e. database, schema and table names) on the relational database in which the data mining model is to be extracted.

(d) Specify the location on the reporting system, on which the report meta model and the report specifications will be generated.

The system then automatically analyzes and transforms the mining results, generates the report metamodel and report specifications and deploys both to the reporting system.

Figure 3 shows one of the reports that are made available through the BI reporting system. In this example the user of the reporting system can visually analyze the content of the data mining insight, describing the different clusters in which the customers of the company have been segmented. Each row of the report represents a certain cluster, and each column represents a certain field or attribute (e.g: age, gender, average balance...). In each bar chart the distribution for a field of the customers belonging to a certain cluster is overlapped to the expected average distribution of all customers in the company. E.g. cluster 3 represents the elderly people who are rather female and have rather bank cards.

The system presented in this work does not only allow visualizing static data mining insight, which resides on the database prior to report execution. It is also capable of generating and deploying reports which allow the end user to invoke data mining directly from

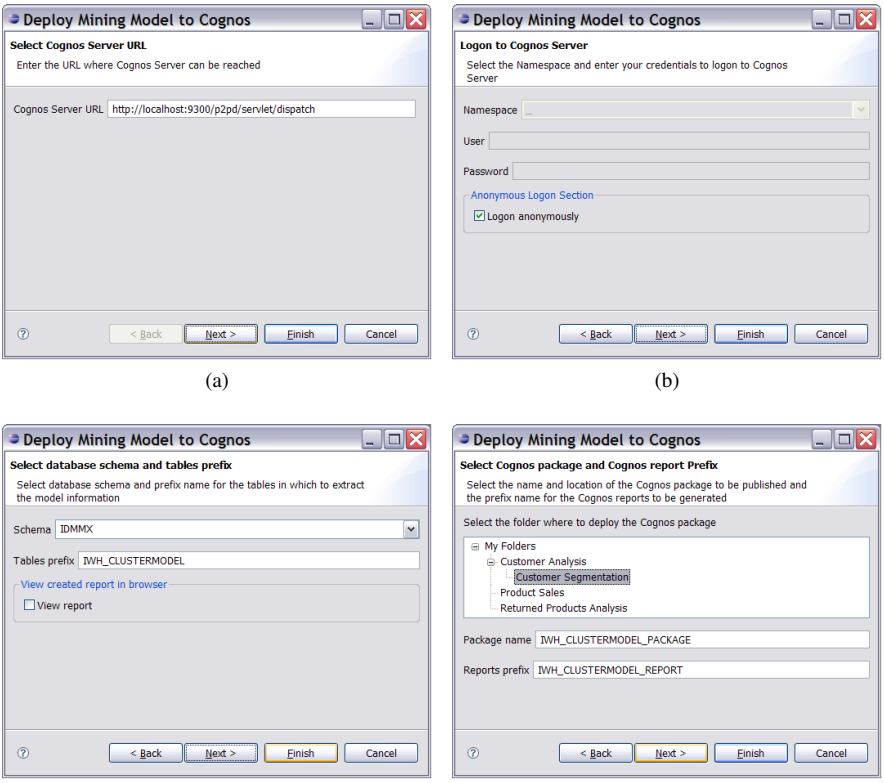

Figure 2: Steps to deploy data mining insight into reporting tool: (a) Enter the url for the reporting server; (b) Enter the credentials to authenticate against the reporting server; (c) Specify the location on the relational database in which the data mining model is to be extracted; and, (d) Specify the location on the reporting system, on which the reporting meta model and the report specifications will be generated.

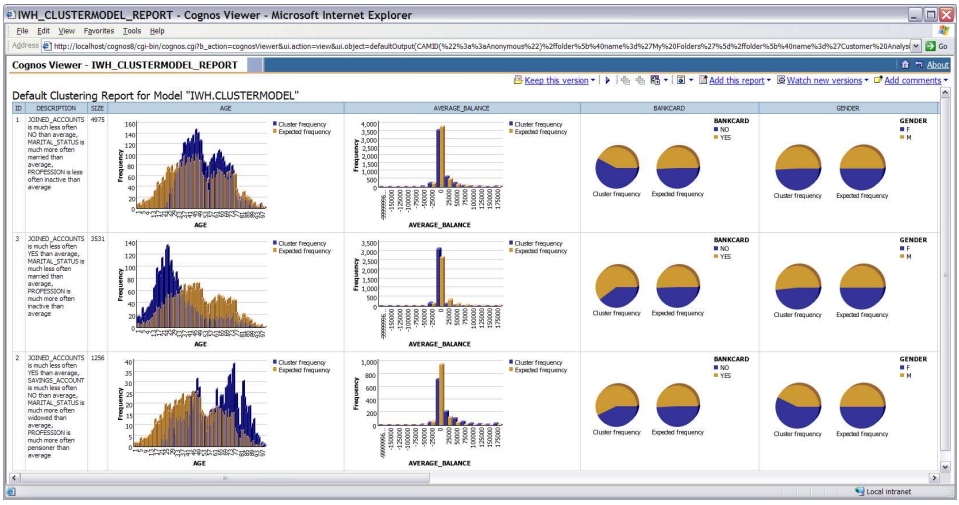

Figure 3: BI report presenting data mining insights for a clustering model

the reporting front end. E.g. the user can select the subset of data to be used for mining or he can configure simple parameters of the data mining algorithm, prior to execution.

Figure 4 shows how the user interacts with a report that allows dynamic invocation of association rule mining on a dedicated cluster selected by the user. The user simply performs the following steps:

(a) Select a certain cluster of customers, for which he wants to compute association rules.

(b) Enter a value for the minimum support of the rules to be computed.

(c) After submitting the report, the association rules are calculated by invoking the In-Database-Mining functionality.

(d) The calculated rules fulfilling the support threshold set by the user are then presented in the report front end.

The generation and deployment of such a report is made in a similar way as described in figure 2.

4 Architecture

The proposed deployment system links two fundamental BI structures. The Warehouse, based on a relational database that provides In-Database-Mining functionality and the BI system. The general functionality of the deployment system comprises:

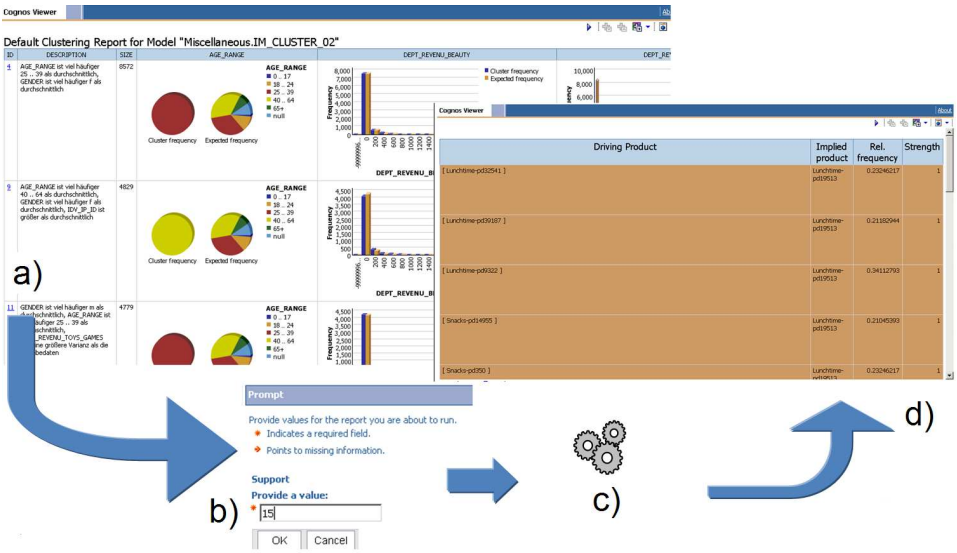

Figure 4: BI report invoking association rules data mining on demand on a subset of the data selected by the user

1. Translating the predictive insight into a common format uderstandable by the BI system
2. Generating meta information and report specifications for the BI system that are suitable to visualize mining / predictive insight in an appropriate and understandable format
3. Deploying the meta information and reports to the BI system.

There exist two fundamental approaches. A static approach in which a mining expert creates mining models that need to be delivered to the business user. And a dynamic approach, in which the mining expert just defines the process to create the mining model, which can then be invoked dynamically by the user through the BI system. The next chapters describe both approaches in detail.

4.1 Static mining content

The static approach is based on a **Mining Model** (see Figure 5)[1] created by a mining expert (including **Data Preparation** and **Modeling** steps). The deployment system automatically creates a table representation from the **Mining Model** and extracts it to the database as a

[1] For the whole paragraph, the bold terms refer to Figure 5

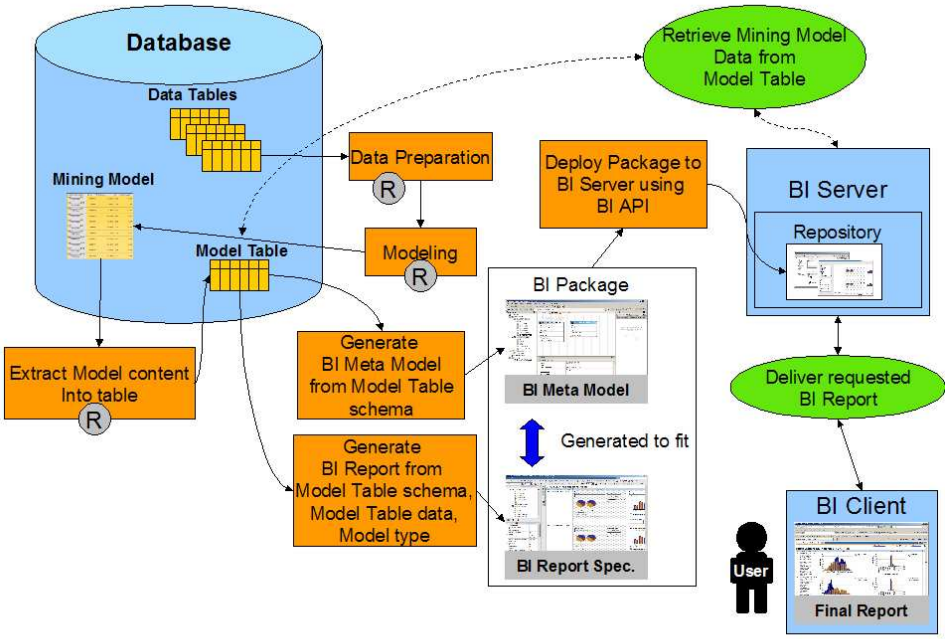

Figure 5: System architecture for deployment of static mining insights

Model Table. This representation is done in a fashion such that the mining insight can be accessed by the **BI Server**.

In a second step, the deployment system generates the necessary meta information (**BI Meta Model**) and **BI Report Specifications** required by the BI system. This meta information and reports are dynamically created based on the content of the **Mining Model** and the layout and data of the **Model Table** containing the mining insight.

Finally, the reports are automatically deployed to the **BI Server**. The deployment system uses the **BI Server's** API interfaces to deploy the generated **BI Meta Model** and **BI Report Specifications** without manual user interaction. It also triggers creation of the actual report from its specification within the **BI Server**. Then, the **User** can access the mining / predictive insight like any other report using the **BI Client**. The **BI Server** retrieves the mining insight directly from the Model Table.

Even though the report specification is static, the actual content can be updated. In more detail, the information contained in the **Model Table** can be updated by re-executing the steps marked with an R in Figure 5. E.g. the re-execution may result in a different clustering reflected by different information in the **Model Table**. The report specification remains the same but the content is retrieved from the updated **Model Table**. This update process

can also be incorporated into automatic business processes to keep the mining content up to date.

Generation of the BI-Meta Model:

The **BI Meta Model** generation is based on templates. The templates contain basic structures for the meta model according to the underlying mining method (Clustering, Classification, Association Mining,...). The deployment system first analyzes the layout of the **Model Table**. From this analysis, the meta model objects are derived. Second, the deployment system analyzes the actual data to derive meta model object types.

Generation of the Report specifications:

The generation of the **BI Report Specifications** is also based on templates. The templates contain basic structures for the report specification depending on the mining model type. During generation, the deployment system analyzes the **Model Table** content. The data of the model table is e.g. analyzed for the number of features and their relevance for each cluster. The report can then be restricted to those relevant features. Furthermore, the formating of the reports and charts is optimized based on the data of the model table. For each model type may exist several report specifications that are linked with each other. E.g. detail reports for dedicated charts or drill through reports as explained in the next paragraph.

Drill through aspect: Often users need to know details from the underlying data from which the data mining model was generated. The deployment system automatically incorporates so called Drill through data into the reports, allowing for better understanding of the mining model. Most useful is drill through data representing typical records. E.g. For clustering, typical records are those which best represent the characteristics of a certain cluster. The deployment system automatically detects such records and incorporates them into the reports.

4.2 Dynamic mining content

The deployment system also generates and deploys dynamic reports which invoke mining at the time the report consumer interacts with the BI system. This allows the user to customize the mining-reports by passing parameters and other settings.

The deployment system automatically creates a **Stored Procedure** (see Figure 6)[2] from the **Data Preparation** and **Modeling** steps which are defined by a mining expert. The **Extraction** step is also performed within the **Stored Procedure**. The **BI Meta Model** is now based on the stored procedure instead of the **Model Table** allowing for dynamic interaction. The stored procedure can be invoked by the BI Server passing parameters entered by the **User** using the **BI Client**. The dynamically created mining insight is then retrieved by the **BI Server** from the **Stored Procedure** result set.

Generation of the Stored Procedure:

[2]For the whole paragraph, the bold terms refer to Figure 6

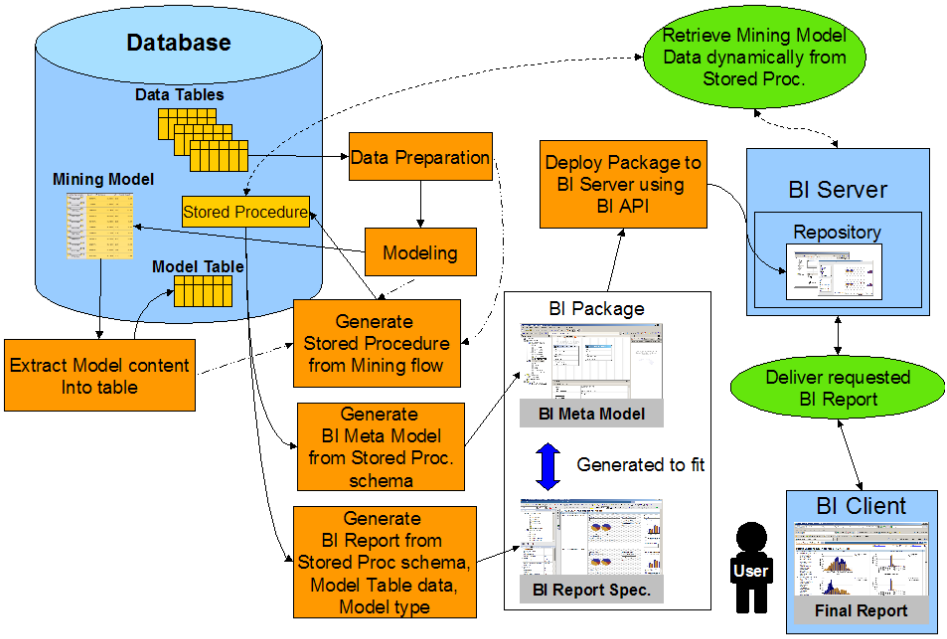

Figure 6: System architecture for deployment of dynamic mining insights

The stored procedure generation is based on the data preparation and mining steps defined by a mining expert (see Figure 6). Often, those steps are defined as data preparation and/or mining flows. The deployment system converts the flow into SQL statements which are placed into the stored procedure. Further, the deployment system incorporates parameters defined by the user. Those parameters are defined as input for the stored procedure and are placed at the proper positions within the SQL statements. E.g. a user could define the maximum number of clusters. As described above, the user simply invokes the report, then the **BI Server** invokes the **Stored Procedure** passing the parameters. The complex flow is completely transparent for the user. The **Stored Procedure** returns data in the same format as the **Model Table**.

5 Implementation

The realized deployment system is based on IBM InfoSphere Warehouse including DB2 with In-Database-Mining. The BI system used is IBM Cognos BI 8. Data preparation and modeling is performed using the Eclipse based InfoSphere Warehouse tool ”Design Studio”. Design Studio allows dfining data and mining flows which can be translated

into SQL and directly executed within DB2. The deployment system is implemented as an Eclipse Plugin for Design Studio. To deploy mining reports to the BI Server the deployment system uses the Web Services API of Cognos BI 8.

The general architecture is based on standardized technologies like relational databases, stored procedures and SQL. Therefore, the deployment system could also be realized for other relational databases with In-Database-Mining like Oracle, Microsoft SQL Server or Teradata. Further, many BI systems provide public APIs to interact with the BI Server like Microstrategy or Business Objects.

6 Summary

Besides the analytical task to generate high quality predictive models the delivery to the right people in time is an important step when turning predictive insight into action. To reach a broad audience the company's BI system is the right vehicle to present this advanced insight in a way the users are used to.

Due to the set up of todays BI environments the process of creating predictive insight and delivering it to the consumers is a complex process requiring several experts in different roles.

The presented system demonstrates how this complex task can be performed in an intelligent, automated fashion to accelerate the delivery process (from hours or days to seconds) and drastically reduce the expert knowledge required.

In this context, not only static mining insight is delivered but the system also allows users to dynamically invoke predictive analytics directly from the BI front-end without the need of deep statistics or mining skills.

As the proposed architecture is based on common standards like relational databases, stored procedures and SQL which are implemented by most vendors, an adoption of the proposed system can be easily realized for most BI environments and product combinations.

References

[IBM] IBM. Creating Web applications with Miningblox. https://publib.boulder.ibm.com /infocenter/db2luw/v9r7/topic/com.ibm.im.blox.doc/miningblox_overview.html.

[Mic] Microstrategy. Using the Microstrategy Platform to Distribute Data Mining and Predictive Analytics to the masses. http://bias.csr.unibo.it/lbaldacc/DataMiningWhitePaper.pdf.

[PMM] Predictive Model Markup Language. http://www.dmg.org/.