
The Excel Data Mining Add-in. Applications in Audit and Financial Reports

Daniel HOMOCIANU,
"Alexandru Ioan Cuza" University of Iași,
E-mail: daniel.homocianu@feaa.uaic.ro

Dinu AIRINEI,
"Alexandru Ioan Cuza" University of Iași,
E-mail: adinu@uaic.ro

Abstract

Performance reasons in decision making based on business data usually requires a good management of multiple data formats and also processing speed, flexibility, portability, automation, power of suggestion and ease of use. The paper comes with theoretical ideas and practical examples in favor of using the Excel Data Mining Add-in's for the aforementioned reasons. Most of the examples include figures linked to video scenarios constructed by the authors and part of an interactive on-line list with eighteen pieces. Together they contribute to understanding most of the requirements to fulfill in order to have valid examples and useful results.

Keywords: business and financial data, spreadsheets, Data Mining (DM), examples

JEL Classification: C61, D81, D83, M42

To cite this article:

Homocianu, D. and Airinei, D. (2017), The Excel Data Mining Add-in. Applications in audit and financial reports, *Audit Financiar*, vol. XV, no. 3(147)/2017, pp. 451-468, DOI: 10.20869/AUDITF/2017/147/451

To link to this article:

<http://dx.doi.org/10.20869/AUDITF/2017/147/451>

Received: 17.03.2017

Revised: 13.04.2017

Accepted: 20.04.2017

Introduction

This paper starts from some techniques used by most Data Mining tools when dealing with large data from databases and presents the advantages of using spreadsheets as client applications (msdn.microsoft.com/...dn282385.aspx). The last ones are so familiar to the end users and have an interface that integrates programming or scripting languages for office applications such as VBA meaning Visual Basic for Applications for Microsoft Excel (msdn.microsoft.com/.../ee814737.aspx), and Google Apps Script for Google Sheets (developers.google.com/.../sheets), many functions and advanced facilities for processing, analysis, representation and simulation all based on the interactivity and dynamics principles with great impact on users' ability to perceive, interpret, understand and manage complex information in different cases.

The concept of Data Mining essentially means the supervised identification of undiscovered patterns and hidden relationships in huge data sets (searchsqlserver.techtarget.com). Inmon which is a well-known guru in data warehousing (computerweekly.com) gave one of the most concise definitions of a Data Mining (Inmon and Linstedt, 2014) namely analysis of large quantities of data to find patterns such as groups of records, unusual records and dependencies. The Data Mining initiatives usually come from marketing and retail sales departments and are suited for organizations having very large databases (Airinei, 2002).

This concept is closely related to that of data oriented Decision Support Systems (DSS) and Business Intelligence (BI) – especially the one for strategic purposes (dssresources.com/...id=174) that requires huge amounts of data (bi-insider.com). Although the BI term is known as a set of concepts and methods for improving decision-making emerging in the 90's (Howard Dresner from the Gartner Group - dssresources.com/.../dsshistory.html), the evidences from the specialty literature indicate approaches from 15 years earlier (Cleland and King, 1975; Pearce, 1976) containing clear references to BI, business planners and managers and decision making.

As concluded by Dan Power (dssresources.com/...id=199), Data Mining tools include: case-based reasoning, data visualization (mostly graphs, trees, and clusters), fuzzy queries and analyzes, genetic algorithms, and neural networks.

Starting a few years ago we are witnessing implementations of this concept and related models not only in applications dedicated to database and data

warehouse management systems, but in modules of spreadsheet applications that are working with these above as suggested even from this paper's title. This seems obvious when thinking that such dedicated products allowed the construction of DM structures and models starting simply from one table (usually as aggregation of many others from a database).

The applicability of the theoretical and practical elements of this article in auditing, especially the one of performance (Fraser, 1998) and financial reports is justified starting from a specific need to valorize the existing data structures (often data records in tables and tables in databases) and get rapidly and at minimum cost reports able to present clear information on causality related to effectiveness (actual / estimated results compared to those proposed) and efficiency (consumed resources compared to achieved / estimated results).

The concrete examples in this article support certain conclusions drawn from the literature review, namely: the utility of approaching the audit engagements by using data mining techniques (Vintilescu Belciug et al., 2010) as a complement to traditional methods of risk analysis and intervention on site, the consecration of existence of possible areas of integration between data mining and audit processes grouped by stages (Sirikulvadhana, 2002) such as: planning, execution, documentation and completion) or by specific examples (Wang and Yang, 2009) as neural networks for: risk assessment, finding errors and fraud, determining the going concern of a company, evaluating financial distress, and making bankruptcy predictions and decision trees for: analysis of bankruptcy, bank failure, and credit risk), the advance of the latest software tools that implement data mining algorithms and the fact that many users considered them until recently being not very friendly (Chersan et al., 2013) and requiring technical skills advanced enough.

The paper also aims to eliminate some confusions on using time stamp values (e.g. calendar dates, parts of it or replacement values) when operating on time series containing business data (e.g. sales amount recognized as a factor of direct influence for the level of certain financial indicators such as the operating income).

1. The research methodology

The source data for the examples presented in this paper come from two Microsoft samples databases. The first set of examples was created starting from an Access database file called foodmart (sites.google.com/.../supp4excel2datamining) originally available on the

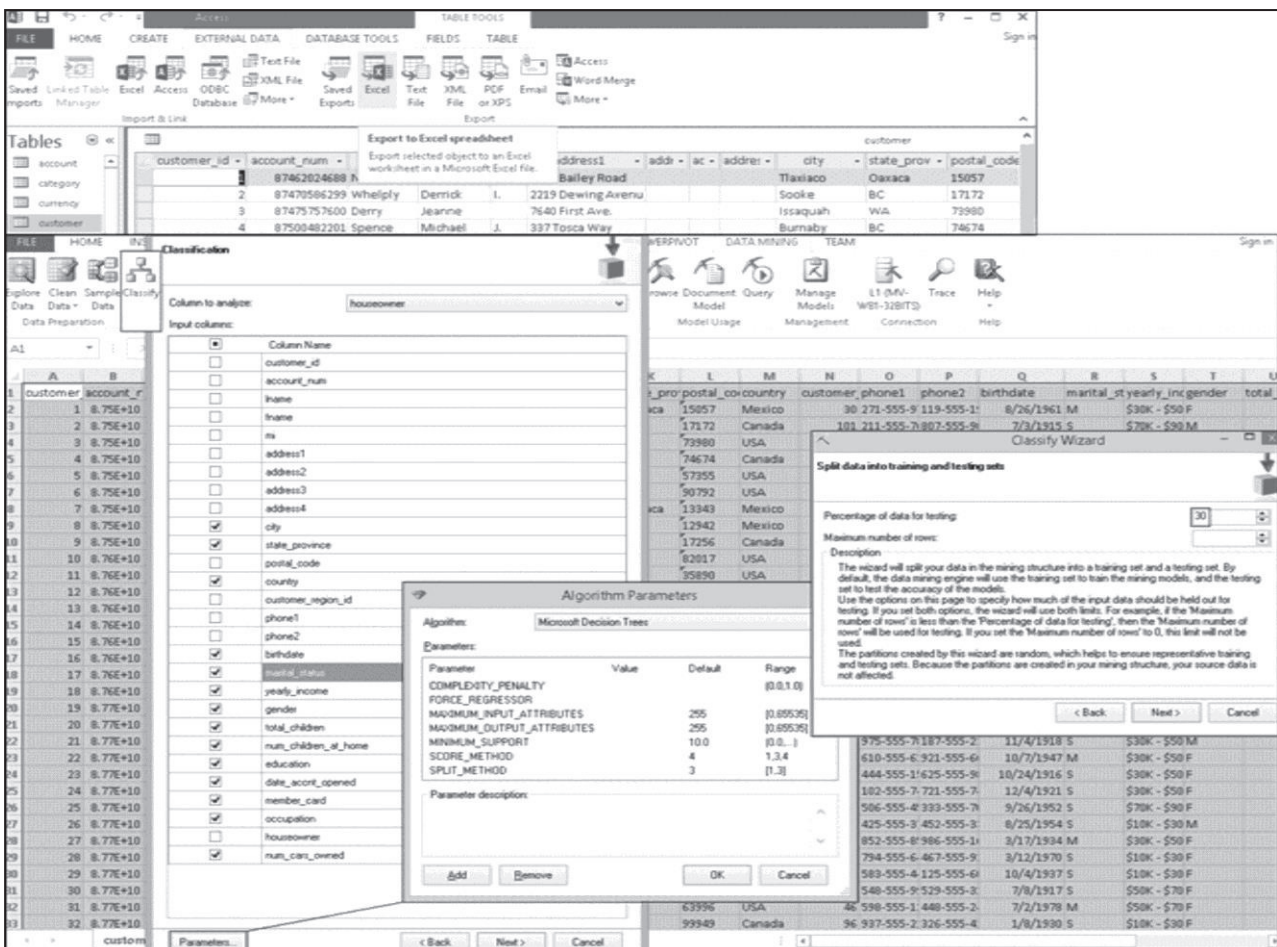
installation CD of a previous version of Microsoft (MS) SQL Server. The second one is from a MS SQL Server sample database called „AdventureWorksDW2012_Data.mdf” already installed and prepared for use inside a Windows 8.1 32 bits virtual machine (y2u.be/Xs2SWtBqdzi) that we have used for this article. This machine benefited from the Microsoft Imagine / formerly Dream Spark educational software license for all applications installed inside and it was optimized for Oracle Virtual Box. In fact SQL Server 2012 (or 2008) is a prerequisite for the installation of the Excel Data Mining add-in which is detailed in the second video tutorial (playlist mentioned below). Although they serve for building the examples and related video support materials (tutorials – the playlist created by the authors and available at goo.gl/JDDtFp), such data only have a

guide purpose in this research with high applicative nature, the similarities to reality being merely coincidental.

2. From intuitive patterns to deep analysis starting from simple sources of data as tables

The first example we have chosen was meant to classify by generating a decision tree where the estimated variable was a categorical one with two possible values (house_owner: Yes or No – Y or N) depending on some other fields (see Figure no. 1) containing information about customers (an export to Excel from the customer table in the foodmart database).

Figure no. 1. Example of export followed by the use of classification option of the Excel Data Mining add-in and the configuration of the input fields

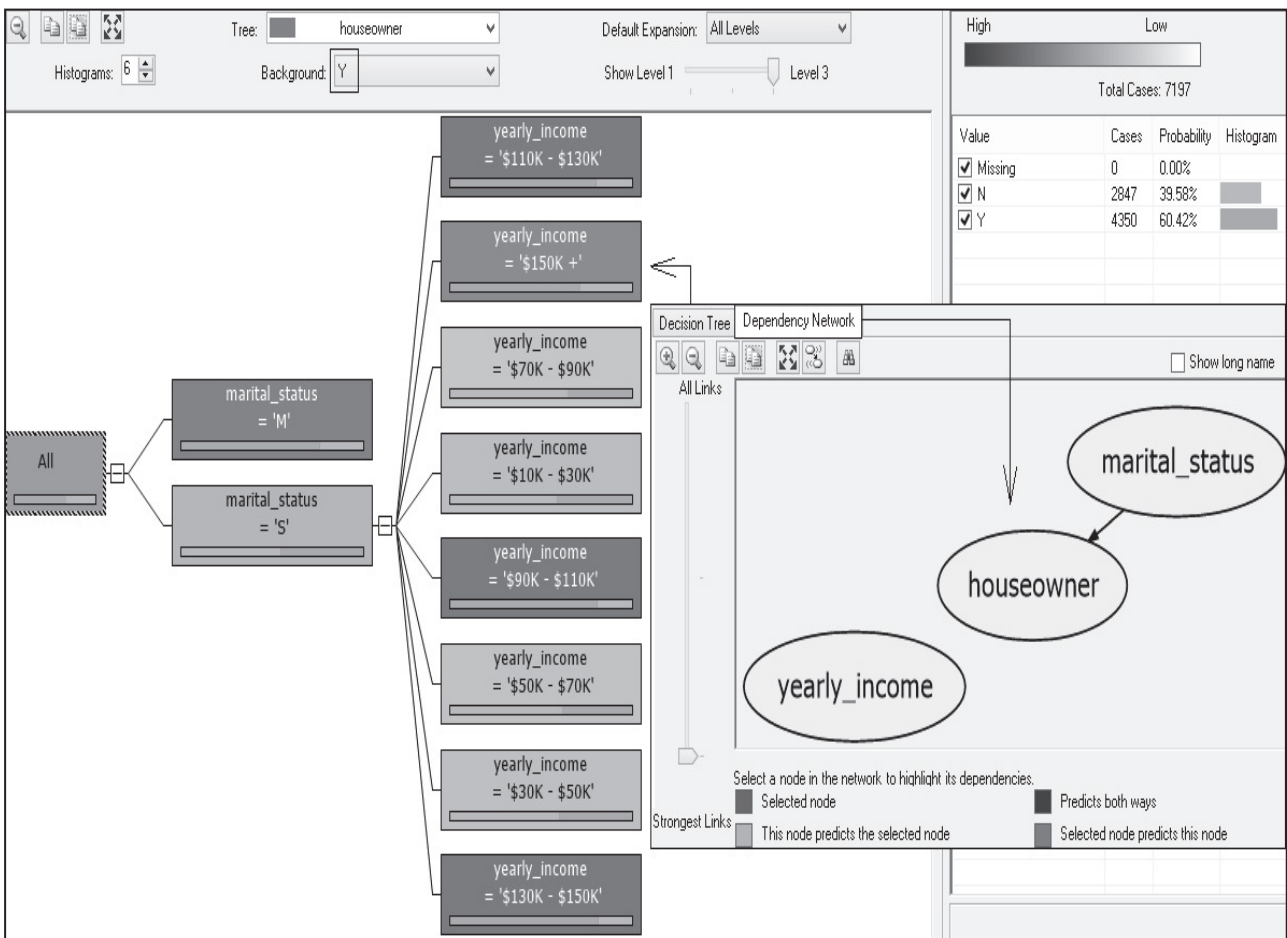


Source: The video tutorial created by the authors: y2u.be/Nx9xqCX1DjY

The results (Figure no. 2) of this classification above by using default settings (Microsoft Decision Trees algorithm and 30 percent of data for testing) consist in: (1) a decision tree and (2) a dependency network that indicate the most important variables that influence the

houseowner value, namely marital_status (married or single – M or S) and yearly_income (eight thresholds: '\$10K - \$30K', '\$30K - \$50K', '\$50K - \$70K', '\$70K - \$90K', '\$90K - \$110K', '\$110K - \$130K', '\$130K - \$150K', '\$150K +'), in this order of importance.

Figure no. 2. Example of result of classification starting from data in a table with customers and made by using the Microsoft Decision Trees algorithm

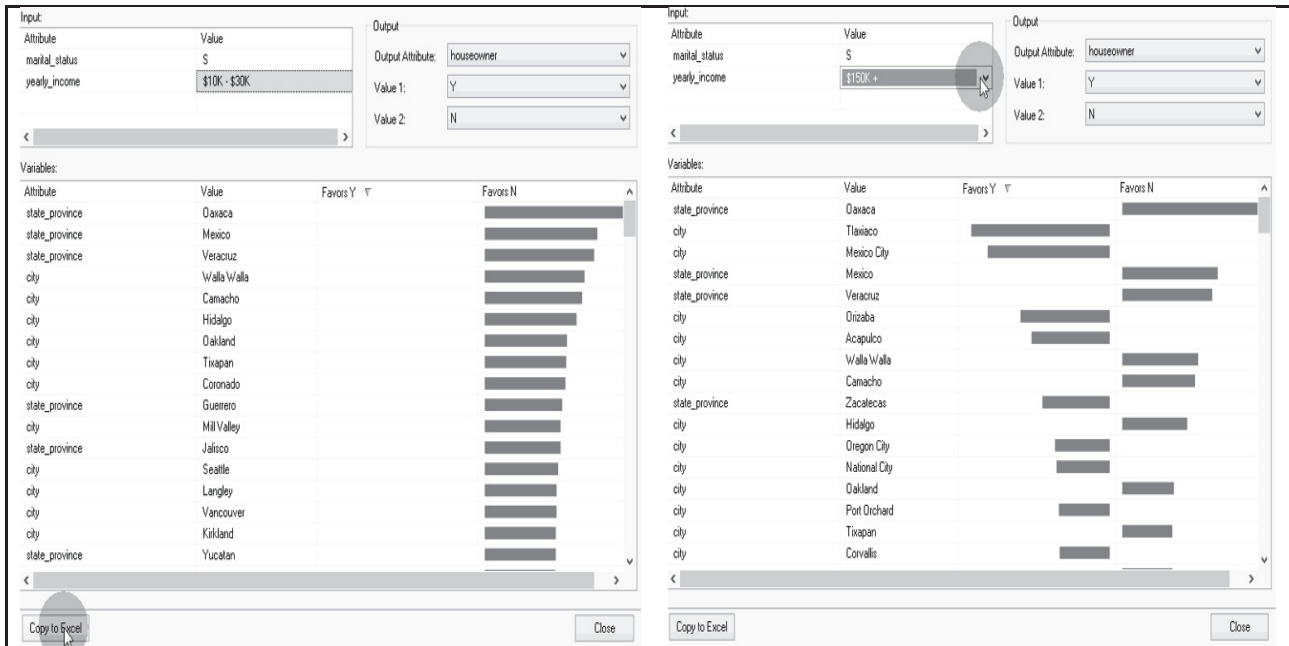


Source: The video tutorial created by the authors: youtu.be/Nx9xqCX1DjY

As seen above (left side of Figure no. 2) the branches that indicate a higher probability for Yes (Y – houseowner) are darker, the rest of them being colored with a lighter shade. We can also observe that the houseowner as a variable depends essentially on the marital_status (right side of Figure no. 2 – the slide bar on Strongest Links) and then on the yearly_income (the slide bar on All Links). And

that can also be deduced directly from the decision tree in which the node closest to the root expresses a test (inf.ucv.ro) corresponding to the marital_status attribute. When clicking on marital_status='M' (terminal node) we have got a probability more than 74% in all ten tests we have done in the same configuration (input columns, column to analyze, algorithm, percentage of data for testing).

Figure no. 3. Examples of discriminative analysis after applying the logistic regression (profs.info.uaic.ro) for the same conditions above and specifying those two already identified major impact input variables and some of their values



Source: The video tutorial created by the authors: [y2u.be/-6jzQuyTjlo](https://www.youtube.com/watch?v=y2u.be/-6jzQuyTjlo)

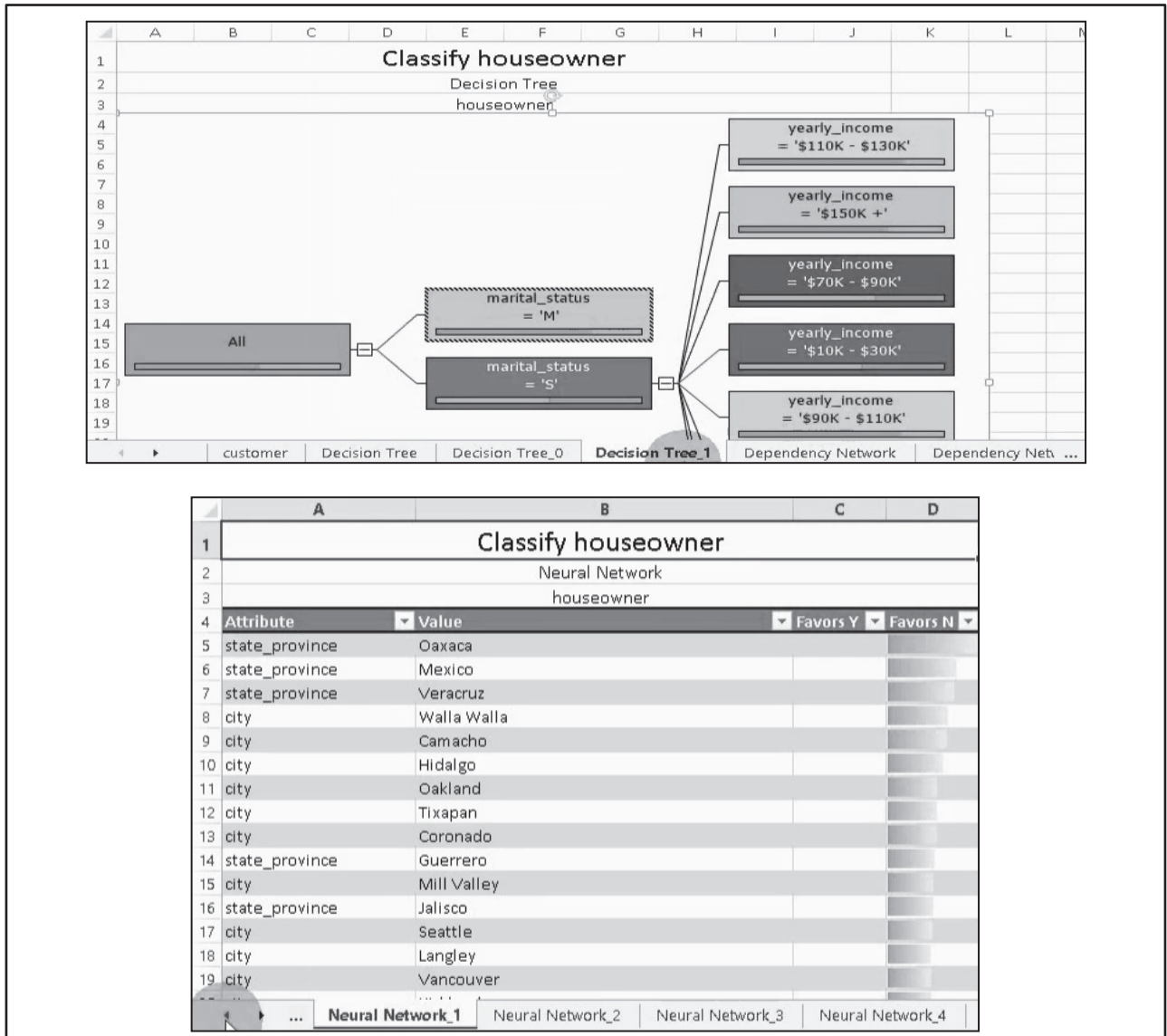
In the previous images (Figure no. 3) we tried to show how we have predicted probabilities that the customers fall into those two categories of the binary response (onlinecourses.science.psu.edu): house owner or not, depending on some explanatory variables and their values. We have done the discriminative analysis partially captured above (Figure no. 3) starting from another algorithm, namely Logistic Regression implemented by Microsoft using a variation of the Neural Network algorithm (msdn.microsoft.com/.../ms174828.aspx) which is easier to train.

The "Copy to Excel" functionality helped us to send the results back to Excel as new sheets containing screen shots (left side of Figure no. 4 for decision

trees) or most importantly data sets with visual effects usually involving conditional formatting done automatically (e.g. discriminative analysis based on logistic regression - right side of Figure no. 4).

Based on the results described above (Figures no. 1-4) one can develop similar examples to address also the problem of customer classification (corresponding to the acceptance / maintaining phase of the audit approach) into one of two categories: acceptable / unacceptable, starting from a validated log of such decisions, in tabular format and including many other descriptive attributes (geography, industry, average number of employees, turnover, evolutions of certain indicators, amount of fee, etc.).

Figure no. 4. Example of results of the “Copy to Excel” functionality



Source: The video tutorials created by the authors: [y2u.be/Nx9xqCX1DjY](https://www.youtube.com/watch?v=Nx9xqCX1DjY) and [y2u.be/6jzQuyTjlo](https://www.youtube.com/watch?v=6jzQuyTjlo)

3. Cumulating historical data and using descriptive fields from many tables of a database

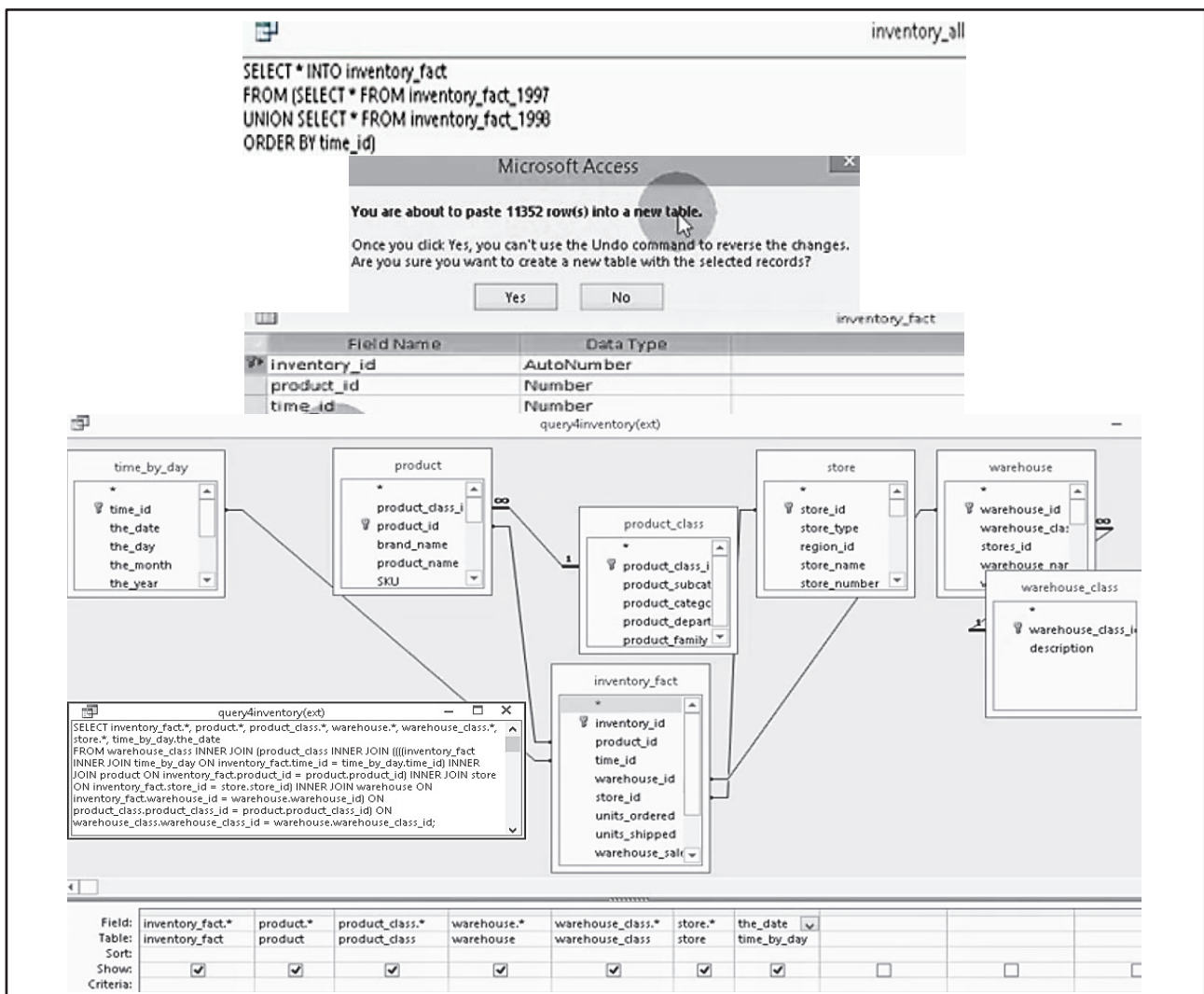
The dynamic and interactive reports responding to many information needs that we are so familiar with as well as the older static ones as snapshots of information at precise moments and generating more questions than answers (Rasmussen et al., 2002) may use both current

and historical data. The 1st category is represented by data from Transaction Processing Systems (TPS) and commonly referring to the current year while the second essentially means data involving a larger period as time reference. The proportion of using those two categories essentially depends on decisional needs (at operational, tactical or strategic level). For minimizing the redundancy and dependency of data or because of storage space and write speed needs (deshpande.mit.edu) the schema of a traditional

relational data source is usually thought as many tables obtained by applying the principles of normalization (w3schools.in). Moreover, because of further performance reasons (read, respectively write speed needs) historical data must be separated from current data. Both categories essentially include records from transaction tables (e.g. expenses, sales, exams, etc.) the difference being made by the value of the time stamp. That explains why those tables loaded only with historical data are being renamed with a time indication, archived and separated from the rest of the transactional system in order to improve its operational (current)

performance. When needing large amounts of historical data for analyzes based on ad-hoc queries the systems must do vice versa by aggregating into a single table (source for a fact table in a data warehouse) all the records from the historical archives of the transaction tables (of the same type as the resulting one). In most cases, that generates the advantage of an increased potential to identify patterns. But it also comes with difficulties related to putting data together in a common and consistent format especially when the applications and the structure of the data source have also changed in time.

Figure no. 5. Gathering both transactional and descriptive data by using two MS Access SQL queries in cascade

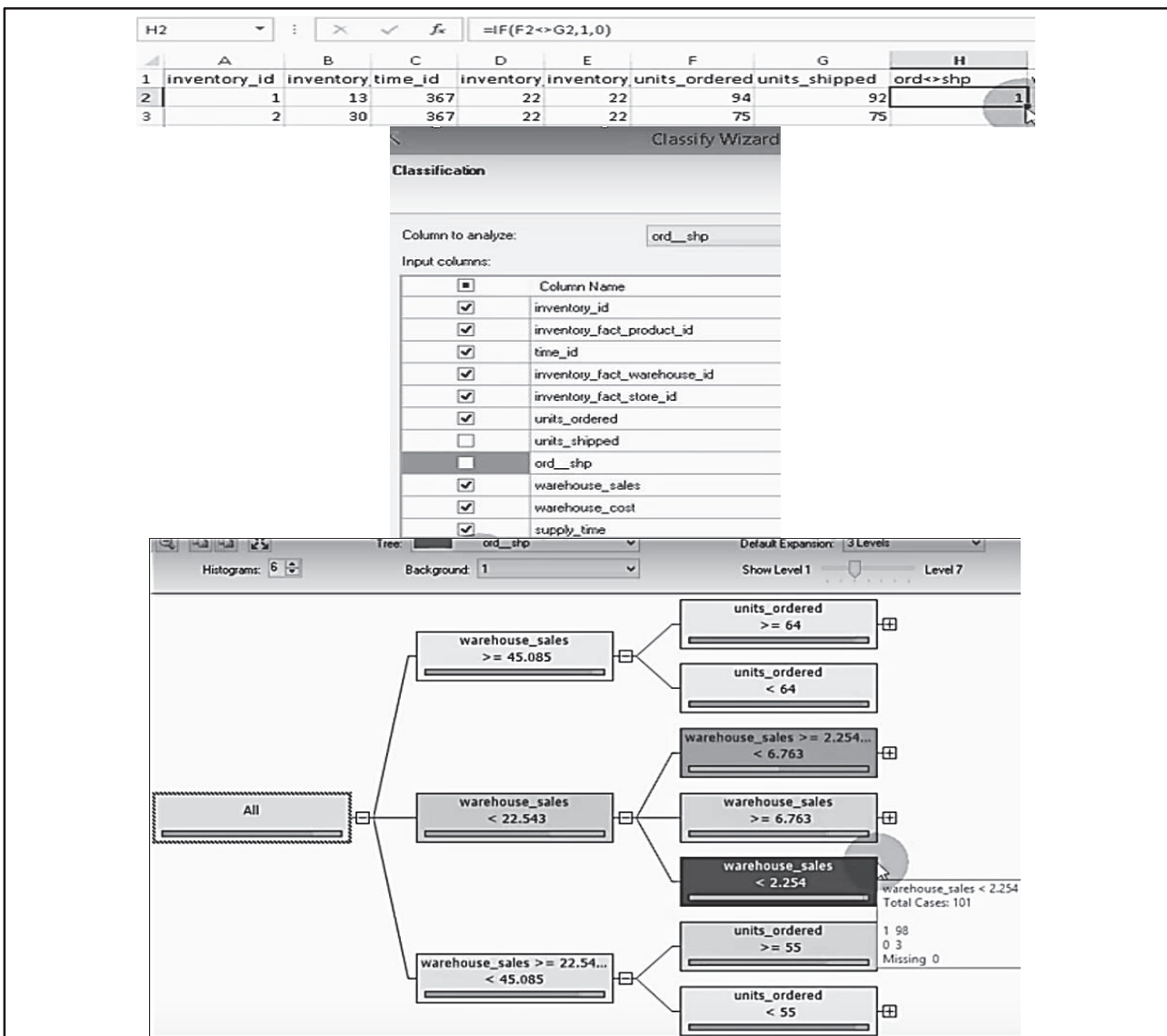


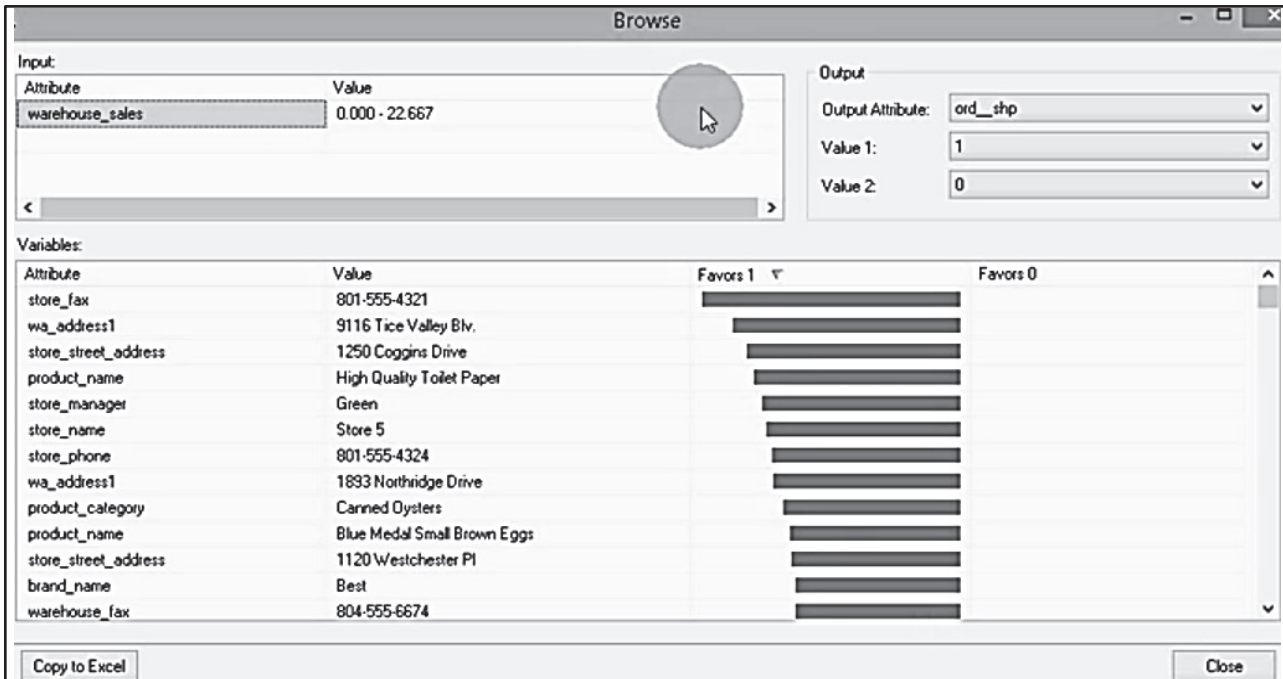
Source: The video tutorial created by the authors: [y2u.be/kTuYLuv3Eo](https://www.youtube.com/watch?v=y2u.be/kTuYLuv3Eo)

The **Figure no. 5** is presenting an example of inventory data gathering (applications including freight audit) in two major steps corresponding to two SQL queries in Microsoft Access: 1st - based on cumulating (UNION clause) the records from two transaction tables of the same type and corresponding to just two years (1997 and 1998) and adding an necessary id column (inventory_id with values generated automatically – AutoNumber type) in the resulting persistent table (INTO clause); 2nd - based on temporarily retrieving values of

descriptive fields from all the tables related or suitable for a relation (**Figure no. 5** – INNER JOIN clause) with the one resulting from the 1st query above, namely *inventory_fact*. In this case the resulting tabular data consisting in the second set of just 11352 records won't get into a persistent table of the database (a kind of de-normalization - searchoracle.techtarget.com) otherwise needed to save time at the expense of storage space and it will serve for external export (Excel) just after executing / running the query itself.

Figure no. 6. Results of consecutively using 2 Data Mining models - derived target field with only 2 values





Source: The video tutorials created by the authors: y2u.be/4nOMMRoC2BU and y2u.be/wce_aoTTsbw

Moreover, for speed of design reasons we have chosen all source fields without selecting them explicitly but indicating that by using the most flexible wildcard character, asterisk / “*”, after the table name (Figure no. 5), both SQL and design mode (techrepublic.com). For the same reasons above the new derived column needed for analysis (Figure no. 6 - output attribute for both models: classification-top and logistic regression-bottom) was then defined directly in Excel by using the IF function (top of Figure no. 6, ord<>shp as 1 or 0 meaning that units ordered and units shipped are different compared to each other or equal).

4. Association rules for identifying behavioral patterns

In the theory and practice of data warehouses and multidimensional modeling the examples below reminds of the “snowflake” schema meaning that the source for a dimension (perspective of analysis based on descriptive columns organized in hierarchies) is not represented by just a single table but many related ones (in one-to-many relations: e.g. product category, product subcategory, and product – Figure no. 7) able to support the analysis with more than just one descriptive

field per dimension. In order to be able to apply the association rules algorithm in this case below we have also needed repetitive values for the SalesOrderNumber field to be associated to different product categories / subcategories / names.

The main reason for gathering those descriptive data residing in multiple tables from the database in the example above (Figure no. 7) is to determine association rules type “If I buy the product X, I will buy the product Y too.” in the purchasing behavior (FactInternetSales source table) and the most important dependencies (Figure no. 8).

From the results in Figures no. 7 and 8 we can understand why the applications of the algorithms able to identify association rules can contribute to audit and fraud detection and prevention. As example, if the set of inputs would have attributes such as: Claim identifier, Insurance type, Name of the insurance product, Name of the insured person, Insurer, Name of the examiner agent and Solution (total or partial loss and reject) and the algorithm would identify “IF Casco insurance, Insured person X and Examiner agent Y THEN total loss” as association with high probability and importance, it would not necessarily mean a fraud alarm but it would worth at least the effort to investigate further.

Figure no. 7. Gathering both types of data: transactional about sales and descriptive about products by using a single MS SQL Server query

The screenshot shows a Notepad window with the following SQL query:

```

SELECT [SalesOrderNumber], [EnglishProductCategoryName],
[EnglishProductSubcategoryName], [EnglishProductName], [UnitPrice]
FROM [dbo].[FactInternetSales], [dbo].[DimProductCategory], [dbo].
[DimProductSubcategory], [dbo].[DimProduct]
WHERE [dbo].[FactInternetSales].[ProductKey]=[dbo].[DimProduct].[ProductKey] AND
[dbo].[DimProduct].[ProductSubcategoryKey]=[dbo].[DimProductSubcategory].
[ProductSubcategoryKey] AND [dbo].[DimProductSubcategory].[ProductCategoryKey]=
[dbo].[DimProductCategory].[ProductCategoryKey]
    
```

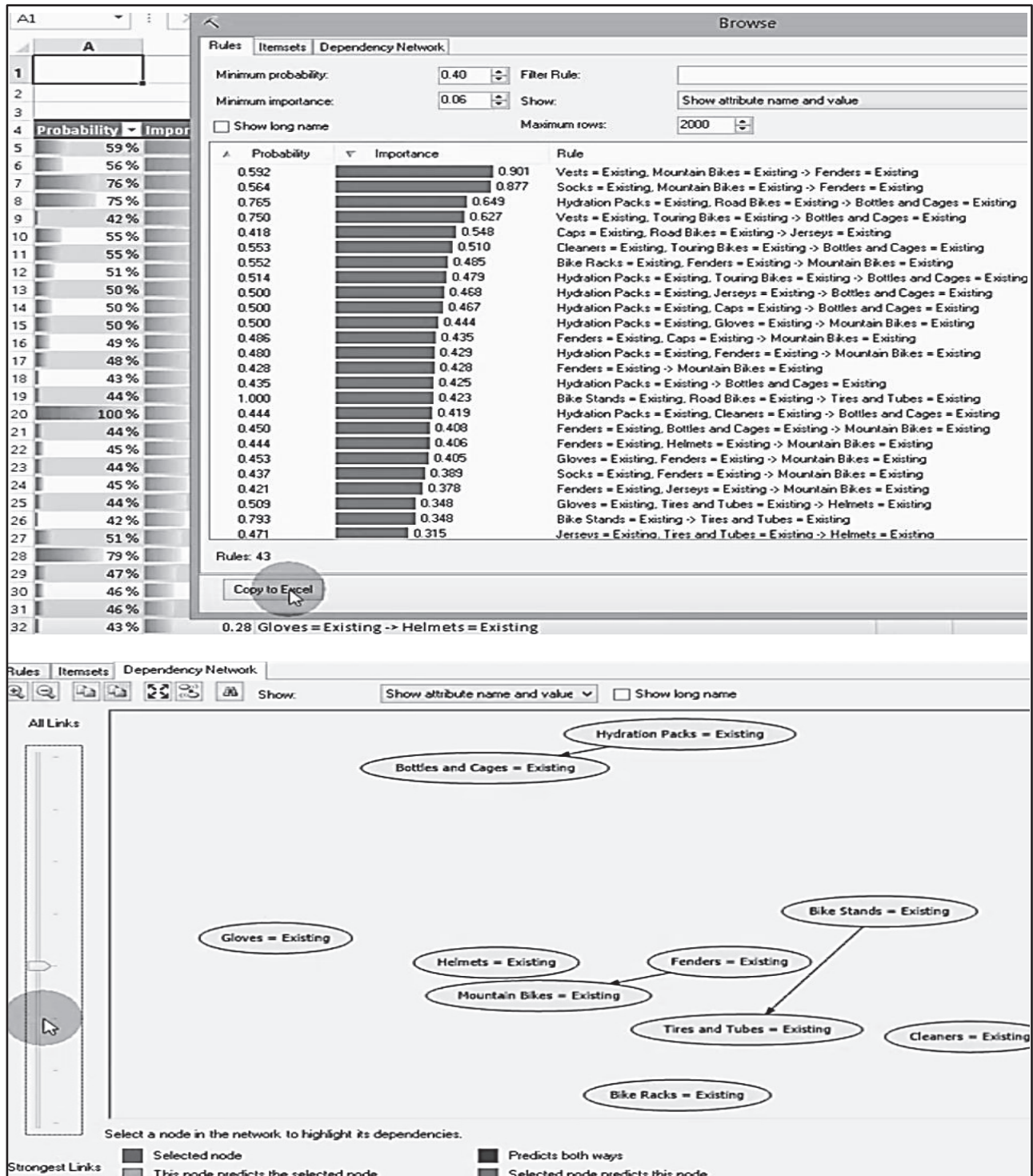
Below the Notepad window, the SQL Server Enterprise Manager interface shows the same query in the SQL Query window. The Results window displays the following data:

SalesOrderNumber	EnglishProductCategoryName	EnglishProductSubcategoryName	EnglishProductName	UnitPrice	
1	SO43697	Bikes	Road Bikes	Road-150 Red, 62	3578.27
2	SO43698	Bikes	Mountain Bikes	Mountain-100 Silver, 44	3399.99
3	SO43699	Bikes	Mountain Bikes	Mountain-100 Silver, 44	3399.99
4	SO43700	Bikes	Road Bikes	Road-650 Black, 62	699.0982
5	SO43701	Bikes	Mountain Bikes	Mountain-100 Silver, 44	3399.99
6	SO43702	Bikes	Road Bikes	Road-150 Red, 44	3578.27
7	SO43703	Bikes	Road Bikes	Road-150 Red, 62	3578.27
8	SO43704	Bikes	Mountain Bikes	Mountain-100 Black, 48	3374.99
9	SO43705	Bikes	Mountain Bikes	Mountain-100 Silver, 38	3399.99
10	SO43706	Bikes	Road Bikes	Road-150 Red, 48	3578.27
11	SO43707	Bikes	Road Bikes	Road-150 Red, 48	3578.27
12	SO43708	Bikes	Road Bikes	Road-650 Red, 52	699.0982
13	SO43709	Bikes	Road Bikes	Road-150 Red, 52	3578.27
14	SO43710	Bikes	Road Bikes	Road-150 Red, 56	3578.27
15	SO43711	Bikes	Road Bikes	Road-150 Red, 56	3578.27

The status bar at the bottom of the SQL Server window indicates: "Query executed successfully. MV-W81-32BITS (11.0 SP1) mv-w81-32bits\admin (56) AdventureWorksDW2012 00:00:00 60398 rows".

Source: The video tutorial created by the authors: y2u.be/2rW2wK77HD8

Figure no. 8. Results of applying the Microsoft Association Rules algorithms (the associate option of the add-in)



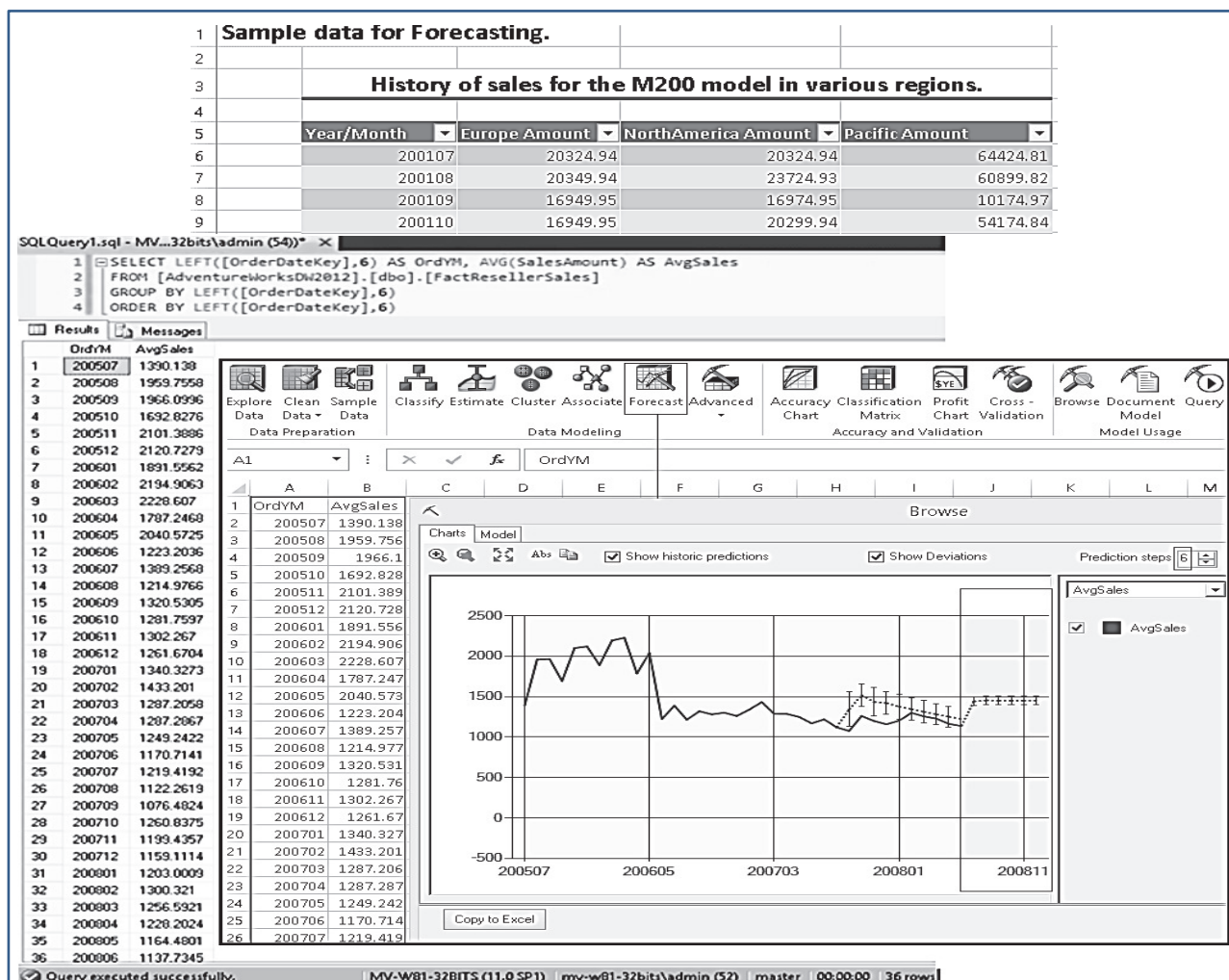
Source: The video tutorial created by the authors: y2u.be/3_8E01hnSD0

5. Forecasting starting from aggregated historical data

For more historical data than in the previous example (Figure no. 5) we have considered to create a special forecasting scenario closer to reality. We have started from scratch with a new example involving data on 36 months in four calendar years, this time by using a simple SQL query on a single table but with ORDER BY and

GROUP BY clauses for sorted results and aggregations meaning computing aggregated values as: sums, averages, total counts, counts for a specified condition and so on. In our case those were averages on every month of an year combined into a single numerical field derived / composed by passing from left to right in the specific order: years to months corresponding to larger to smaller units (Figure no. 9 - just like in Microsoft's data sample which is provided when installing the Data Mining add-in).

Figure no. 9. Historical data aggregation (time stamp style from the Microsoft's sample) using a SQL Server query (GROUP BY clause) followed by simply copying results to use them in forecasting (Excel's Data Mining add-in)

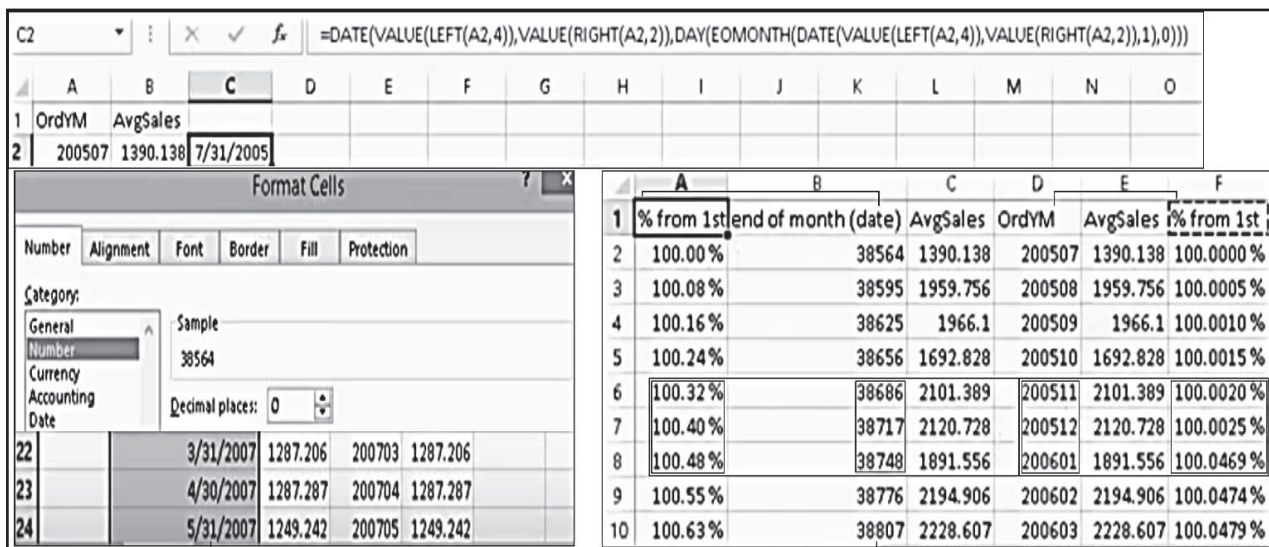


Source: The video tutorials created by the authors: y2u.be/RjTGWROD0TI and y2u.be/qHJ3Zm3JBT4

After the steps described above (Figure no. 9) and several other processing operations (Figure no. 10) we will get to a set of data suitable for forecasting implemented by using the Microsoft Time Series algorithms as a combination of ARIMA (Auto-Regressive

Integrated Moving Average - optimized for improving accuracy in long-term predictions) and ARTXP (Auto-Regression Trees with Cross-Prediction - optimized for predicting the next likely value in a time series - msdn.microsoft.com/.../bb677216.aspx) algorithms.

Figure no. 10. Deriving and explaining the correct time stamps as full dates internally stored (Excel) as numbers in right format data sources as support for undistorted forecasting



The screenshot shows an Excel spreadsheet with a formula bar containing the formula: `=DATE(VALUE(LEFT(A2,4)),VALUE(RIGHT(A2,2)),DAY(EOMONTH(DATE(VALUE(LEFT(A2,4)),VALUE(RIGHT(A2,2)),1),0)))`. Below the formula bar, a 'Format Cells' dialog box is open, showing the 'Number' category with 'Sample' value 38564 and 'Decimal places' set to 0. The spreadsheet data includes columns for '% from 1st', 'end of month (date)', 'AvgSales', 'OrdYM', and '% from 1st'.

	A	B	C	D	E	F
1	OrdYM	AvgSales				
2	200507	1390.138	7/31/2005			
22		3/31/2007	1287.206	200703	1287.206	
23		4/30/2007	1287.287	200704	1287.287	
24		5/31/2007	1249.242	200705	1249.242	

	A	B	C	D	E	F
1	% from 1st	end of month (date)	AvgSales	OrdYM	AvgSales	% from 1st
2	100.00%		38564	1390.138	200507	1390.138
3	100.08%		38595	1959.756	200508	1959.756
4	100.16%		38625	1966.1	200509	1966.1
5	100.24%		38656	1692.828	200510	1692.828
6	100.32%		38686	2101.389	200511	2101.389
7	100.40%		38717	2120.728	200512	2120.728
8	100.48%		38748	1891.556	200601	1891.556
9	100.55%		38776	2194.906	200602	2194.906
10	100.63%		38807	2228.607	200603	2228.607

Source: The video tutorial created by the authors: y2u.be/e0SkDwG9mNY

We have also thought at automatically deriving the correct (the MM/DD/YYYY format translated into an integer number - Figure no. 10) time stamp labels and we presented more details about their comparative behavior when getting trend line functions and forecasting results with this Excel Data Mining add-in (last three tutorials in the aforementioned playlist).

6. Support for querying persistent Data Mining models

First of all, persistent in this context refers to a model defined the way it will be deployed and stored on the server (SQL Server Analysis Services – a module other than the Database Engine) and available for querying (Figure no. 11).

The Data Mining add-in available in Excel offers many advantages over the direct use of SQL Server Analysis Service. Among others, one can mention here: speed of use of Excel's tabular environment and formula language, possibility of many exports / imports as / from spreadsheets starting from different database formats and to indirectly involve multiple source tables by using the Structured Query Language (SQL), the possibilities of exploiting the resulting structures and models directly (the "copy to Excel" option), by using queries (SQL DMX extension - Figures no. 12 and 13) or programmatically (Figure no. 12). Last two are conditioned by activating persistency when defining models (use temporary model option unchecked - Figure no. 13 vs. Figures no. 1 and 2).

Figure no. 11. Data Mining eXtensions (DMX) sales prediction query examples (SQL Server Analysis Services) based on a DM time series model from a wrong format data source (text time stamp: 200815 /15th month in 2008)

The figure displays two screenshots of the SQL Server Analysis Services (SSAS) interface. The top screenshot shows a DMX query executed successfully. The query is:

```

1 SELECT
2 PredictTimeSeries([forecast_AVG_sales_model].[AvgSales],3) AS PredAvgSales
3 FROM [forecast_AVG_sales_model]
    
```

The results table shows the following data:

\$TIME	AvgSales
200807	1443.24228516...
200808	1454.63464828...
200809	1450.83187874...

The bottom screenshot shows the same interface with the query modified to use a text time stamp '200815' in the PredictTimeSeries function. The results table now includes a prediction for 200815:

\$TIME	AvgSales
200811	1452.83271562
200812	1453.50726198
200813	1453.77358263
200814	1454.14759460
200815	1454.43486230

An Object Explorer window is also visible, showing the database structure for 'DMAddinsDB', including 'Mining Models' and the specific model 'forecast_AVG_sales_model'.

Source: The video tutorial created by the authors: <http://y2u.be/qHJ3Zm3JBT4>

Figure no. 12. Rough example of how to programmatically query a well-defined Data Mining model by using a DMX query in Visual Basic (VB).NET preceded by testing most of it on SQL Server Analysis Services

The screenshot displays a Visual Studio environment with a VB.NET project. The code in `Form1_Load` uses `Microsoft.AnalysisServices.AdomdClient` to connect to a Data Mining model and execute a DMX query. The query is: `SELECT FLATTENED PredictTimeSeries([forecast_model_correct_TS].[AvgSales],1,1) AS PredAvgSales FROM [forecast_model_correct_TS]`. The code iterates through the results and displays them in a `MessageBox`.

A file selection dialog is open, showing the path `Program Files\Microsoft.NET\ADOMD.NET\110`. A small dialog box shows the output: `7/31/2008: 1449.64`.

The SQL Server Enterprise Miner interface shows a mining model named `forecast_model_correct` with a time series chart for `AvgSales`. The chart shows historical data and a prediction line. A table at the bottom shows the results of the query:

PredAvgSales.\$TIME	PredAvgSales.AvgSales
7/31/2008 12:00:00 AM	1449.64321772663
8/31/2008 12:00:00 AM	1460.4264515793
9/30/2008 12:00:00 AM	1456.07251908031

Source: The authors' projection resulting after development attempts with VB and SQL Server

Figure no. 13. DMX prediction query example (houseowner) based on a persistent decision trees classification model (generic content view in background)

The screenshot displays the SQL Server Enterprise Miner interface. On the left, a tree view shows a decision tree structure with nodes for marital status and yearly income. The central pane shows the 'Node Details' for the 'houseowner' node, including its unique name, type, and rule. The right pane shows a DMX query: `SELECT [houseowner], PredictProbability([houseowner], 'Y') AS [HouseOwner = Yes], PredictProbability([houseowner], 'N') AS [HouseOwner = No] FROM [DT_m_HO] NATURAL PREDICTION JOIN (SELECT 'S' AS [marital_status], '$150K +' AS [yearly_income]) AS t`. The bottom pane shows the results of the query, with columns for 'houseowner', 'HouseOwner = Yes', and 'HouseOwner = No'.

MODEL_CATALOG	DMAddinsDB																								
MODEL_SCHEMA																									
MODEL_NAME	DT_m_HO																								
ATTRIBUTE_NAME	houseowner																								
NODE_NAME	0000000r0107																								
NODE_UNIQUE_NAME	0000000r0107																								
NODE_TYPE	4 (Distribution)																								
NODE_GUID																									
NODE_CAPTION	yearly_income = '\$150K +'																								
CHILDREN_CARDINALITY	0																								
PARENT_UNIQUE_NAME	0000000r01																								
NODE_DESCRIPTION	marital_status = 'S' and yearly_income = '\$150K +'																								
NODE_RULE	<compound-predicate op="and"> <predicate op="eq" value="S"> <simple-attribute name="marital_status" /> </predicate> <predicate op="eq" value="\$150K +"> <simple-attribute name="yearly_income" /> </predicate> </compound-predicate>																								
MARGINAL_RULE	<predicate op="eq" value="\$150K +"> <simple-attribute name="yearly_income" /> </predicate>																								
NODE_PROBABILITY	0.0094483812699736																								
MARGINAL_PROBABILITY	0.019036954087346																								
NODE_DISTRIBUTION	<table border="1"> <thead> <tr> <th>ATTRIBUTE_NAME</th> <th>ATTRIBUTE_VALUE</th> <th>SUPPORT</th> <th>PROBABILITY</th> <th>VARIANCE</th> <th>VALUETYPE</th> </tr> </thead> <tbody> <tr> <td>houseowner</td> <td>Missing</td> <td>0</td> <td>0</td> <td>0</td> <td>1 (Missing)</td> </tr> <tr> <td>houseowner</td> <td>Y</td> <td>55</td> <td>0.806155507559395</td> <td>0</td> <td>4 (Discrete)</td> </tr> <tr> <td>houseowner</td> <td>N</td> <td>13</td> <td>0.193844492440605</td> <td>0</td> <td>4 (Discrete)</td> </tr> </tbody> </table>	ATTRIBUTE_NAME	ATTRIBUTE_VALUE	SUPPORT	PROBABILITY	VARIANCE	VALUETYPE	houseowner	Missing	0	0	0	1 (Missing)	houseowner	Y	55	0.806155507559395	0	4 (Discrete)	houseowner	N	13	0.193844492440605	0	4 (Discrete)
ATTRIBUTE_NAME	ATTRIBUTE_VALUE	SUPPORT	PROBABILITY	VARIANCE	VALUETYPE																				
houseowner	Missing	0	0	0	1 (Missing)																				
houseowner	Y	55	0.806155507559395	0	4 (Discrete)																				
houseowner	N	13	0.193844492440605	0	4 (Discrete)																				

Source: The authors' projection resulting from development attempts with SQL Server

This last advantage reminds us that the programmatic generation (Airinei and Homocianu, 2009) of Excel dashboards and scorecards by using suggestive representations, warning indicators and dynamic formatting with support for BI has been simplified a lot since the 2007 version of the Microsoft Office suite. Combining that with the ability to programmatically determine behavioral patterns and generate predicted values starting from high performance and easy to use tools such as this Data Mining add-in available for Office

2010, 2013 and 2016 promises much in terms of productivity. All these advances were defined after many years of using dedicated and now well-known technologies (e.g. SQL Server tested by authors since early 2000).

When it comes to spreadsheet products (dssresources.com/.../sshistory.html) such as: VisiCalc, Lotus 1-2-3, Microsoft Excel, Microsoft Works Spreadsheet, Sun Open Office Spreadsheets, Polaris Office Sheet, and Google Sheets the average

experience of final users is up to decades. Furthermore the easiness of using these applications even just as interface instruments to connect to data from databases and data warehouses and display it was an objective reason to continue with testing the Data Mining component that led to making this article.

By using a way of reporting which identifies itself with a sequence of steps which borrow their names from those eighteen support tutorials and also some techniques previously defined namely: E2P4CAFR (Homocianu, 2015), ACCORD / CADRE (Homocianu and Airinei, September 2014) and S-DOT (Homocianu and Airinei, August 2014) one can reach in stages, but with a minimal number of steps to follow some representations that are dynamic, interactive, suggestive, based on causality and rooted in the current reality and in the history defined by data stored in the organization's data sources.

Conclusions

We can conclude that the possibilities of the Excel Data Mining add-in component are above the expectations of a business analyst, offering the advantage of integrating identified classification patterns, association rules and predictions with the support for connectivity to various data formats, data validations, advanced graphical representations, geographical referencing, automatic conditional formatting and key performance indicators (KPI), pivot and power pivot tables and charts, automatic solving of optimization problems (solver) and the DAX (Data Analysis eXpressions) language together with the

traditional formula language thereby increasing the chances of defining dashboards based on simulations, analyzes and Data Mining models truly useful for audit staff interested in performance monitoring.

We hope we have identified many real motivations to choose this Microsoft add-in for the Office suite as a near real time Data Mining tool, beyond many other recommendations available in the specialized literature and practice.

Beyond effective examples of working with well-known software applications available for a considerable range of users and providing advanced methods for analysis, query and representation of current and historical data particular to support tools for Data Mining and Business Intelligence, the paper also provides a brief theoretical description necessary in order to understand a rapid way of generating complex dynamic reports as dashboards based on analyzes and Data Mining models starting especially from sales and financial data.

The video tutorials developed by the authors, integrated in a playlist, and successively referenced in the paper prove the attempts to enrich the aforementioned way of reporting and to ensure the minimization of the number of steps required when trying to implement similar examples.

Overall, the article is trying and we hope that it succeeds to convey by clear examples some desirable traits as speed, simplicity, capacity of synthesis, transparency, flexibility and availability in reporting with support in Data Mining, as key elements of performance in preparing financial statements and supporting audit activities.

REFERENCES

1. Airinei, D. (2002), *Depozite de date*, Editura Polirom, Iaand.
2. Airinei, D. and Homocianu, D. (2009), The Geographical Dimension of DSS Applications, *Analele Științifice ale Universității „Alexandru Ioan Cuza” din Iaand*, Tome LVI, p. 637-642.
3. Chersan, I.C., Carp, M. and Mironiuc, M. (2013), Data mining – o provocare pentru auditorii financiari, *Audit Financiar*, vol. XI, no. 10, p.57-64.
4. Cleland, D.I. and King, W.R. (1975), Competitive Business Intelligence Systems, *Business Horizons Journal*, vol. 18, no. 6, pp. 19-28, DOI 10.1016/0007-6813(75)90036-1.
5. Fraser, L.E. (1998), Public Sector Audit - Business Integration and Causal Analysis, *Quality Audit Conference*, February 26-27, 1998, Louisville, KY, vol. 7.
6. Homocianu, D. (2015), Excel Power Pivot's Applications in Audit and Financial Reports, *Audit Financiar*, vol. XIII, no. 11, p.127-138.
7. Homocianu, D. and Airinei, D. (2014), Business Intelligence facilities with applications in audit and financial reporting, *Audit Financiar*, vol. XII, no. 9, p. 17-29.
8. Homocianu, D. and Airinei, D. (2014), Consolidating source data in audit reports, *Audit Financiar*, vol. XII, no. 8, p.10-19.

9. Inmon, W.H. and Linstedt, D. (2014), *Data architecture: A primer for the data scientist. Big Data, Data Warehouse and Data Vault*, Morgan Kaufmann, MA.
10. Rasmussen, N.H., Goldy, P.S., Solli, P.O. (2002), *Financial business intelligence – trends, technology, software selection and implementation*, John Wiley and Sons, Inc., New York, p. 98-99.
11. Sirikulvadhana, S. (2002), *Data mining as a financial auditing tool (thesis)*, Swedish School of Economics and Business Administration, pp.49-57, available online at: <https://pdfs.semanticscholar.org/2612/f764664796f911e9ff9a79b7bb9de84bf16c.pdf>, accessed on 15.03.2017.
12. Vintilescu Belciug, A., Crețu, D. and Gegea, C. (2010), Utilizarea tehnicilor de data mining ca metodă complementară în audit, *Audit Financiar*, vol. VIII, no. 7, p. 30-35.
13. Wang, J. and Yang, J.G.S. (2009), Data Mining Techniques for Auditing Attest Function and Fraud Detection, *Journal of Forensic & Investigative Accounting*, vol. 1, no. 1, pp. 1-24.
14. <http://bi-insider.com/business-intelligence/operational-bi-vs-strategic-bi/>
15. <http://dssresources.com/faq/index.php?action=artikel&id=174>
16. <http://dssresources.com/faq/index.php?action=artikel&id=199>
17. <http://dssresources.com/history/dsshhistory.html>
18. <http://dssresources.com/history/sshistory.html>
19. <http://inf.ucv.ro/documents/rstoean/5.%20Arbori%20de%20decizie.pdf>
20. http://profs.info.uaic.ro/~val/statistica/StatWork_10.pdf
21. <http://searchoracle.techtarget.com/tip/Optimizing-database-performance-part-2-Denormalization-and-clustering>
22. <http://searchsqlserver.techtarget.com/definition/data-mining>
23. <http://www.computerweekly.com/tip/Inmon-vs-Kimball-Which-approach-is-suitable-for-your-data-warehouse>
24. <http://www.techrepublic.com/article/10-tips-for-using-wildcard-characters-in-microsoft-access-criteria-expressions/>
25. <http://www.w3schools.in/dbms/database-normalization/>
26. <http://y2u.be/Xs2SWtBqdzl>
27. <https://deshpande.mit.edu/portfolio/project/hybrid-dbms-optimized-read-intensive-applications>
28. <https://developers.google.com/apps-script/guides/sheets>
29. <https://msdn.microsoft.com/en-us/library/bb677216.aspx>
30. <https://msdn.microsoft.com/en-us/library/dn282385.aspx>
31. <https://msdn.microsoft.com/en-us/library/ms174828.aspx>
32. <https://msdn.microsoft.com/en-us/library/office/ee814737.aspx>
33. <https://onlinecourses.science.psu.edu/stat504/node/149>
34. <https://sites.google.com/site/supp4excel2datamining2017af/d>