DATABASE SYSTEM AND SOFT COMPUTING

Abstract
The present article deals with use of advanced analytical functions resulting in information for support of decision-making. Utilisation of this information can bring about measurable economic effects to the company. There are various methods of soft computing, usable for this purpose, such as fuzzy logic, neural networks, evolutional algorithms, theory of chaos etc. Some of these methods are part of Business Intelligence (BI) over Microsoft SQL Server 2008 (SQL Server) platform. This article uses the Microsoft Neural Network algorithm, selected from the nine algorithms implemented in BI, for prediction of prices of used cars and verification of probability of the predicted prices.

1. Introduction

Soft computing pursuant to Zadeh may be defined as follows: "Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation". [9]

The main purpose is tolerance of imprecision and uncertainty which represent the basic attributes of the theories of soft computing and approximation to quality governance, robustness and low costs of solutions. Soft computing methods are run on high-performance computer technology today. The outputs then include high-standard stats further used for support of decision-making processes under unstable conditions. To achieve more stable results a combination of several different sof computing methods is often used.

In the present information society databases represent very strong tools containing not only the wealth of corporate information but the modern database engines even include means for finding out mutual correlations between the database data. Microsoft SQL Server 2008 thus allows companies to use within a single solution various analytical instruments, reports, multidimensional and predictive analyses. Data mining is currently one of the most rapidly developing parts of Business Intelligence allowing for use of artificial intelligence in the database segment.

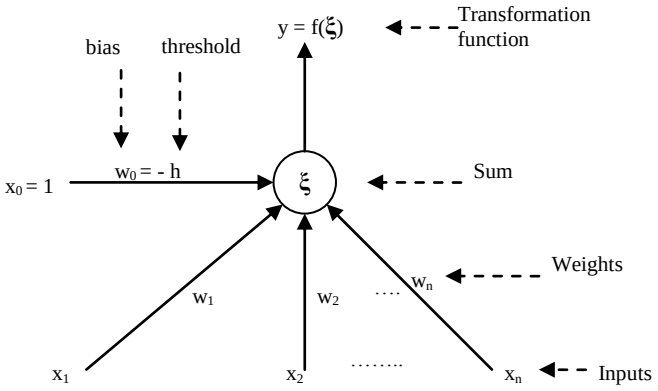
2. Theory of Neural Network

Neural networks represent a certain (albeit imperfect) model of human brain and its thinking. Neural networks are called "black boxes" for the inner structure of the system cannot be known in detail. The inner structure of the system modelled by the "black box" must only comply with a couple of prerequisites allowing to describe behaviour of the system of functions performing the input-output transformation. Neural networks are suitable in cases when a considerable role in the modelled process is performed by chance and where deterministic relations are so complex and interwoven that they cannot be separated, identified or analysed. [1],[2]

2.1 Single-Layer Neural Network

Artificial neuron is based on the principles of biological neuron. Input information is weighed (*weights*). The threshold value is subtracted (*threshold*) and by the activation function (*activation function*) the signal is transformed to the output signal forwarded to the following artificial neurons in the concealed or the output layer. Figure 1 shows the scheme of a mathematical neuron. You can see there that the individual inputs are weighed (***w)*** and transformed in the neuron with the help of the activation function.

Figure 1: Single-layer neural network



The method of the weight setting can be shown with the help of the simplest neural network (the so called "perceptron"). The input will be represented by *R* values p_1, p_2, p_3, …., p_R, multiplied by weight coefficients w_1,w_2, w_3, …., w_R. Further effect is represented by the threshold value *b*.

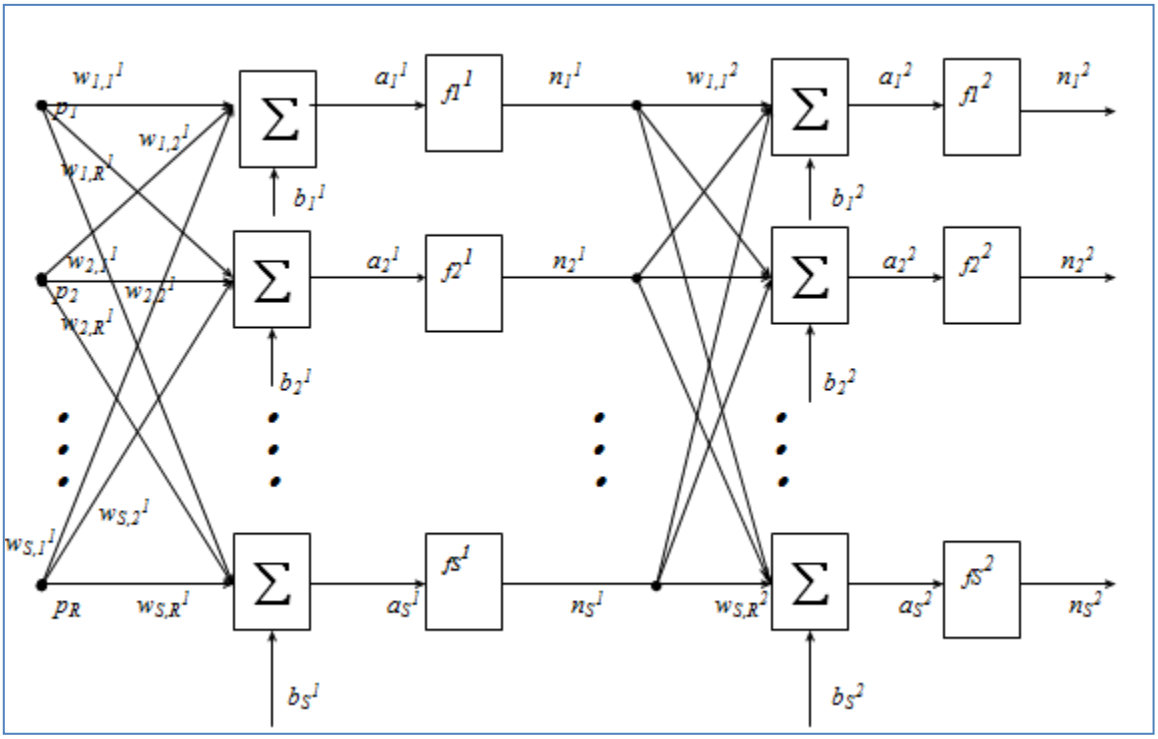
Therefore

$$a = w_1 p_1 + w_2 p_2 + w_3 p_3 + \ \ldots + w_R p_R \ + b = \sum_{i=1}^{R} w_i p_i + b \qquad [1],[2]$$

2.2 Multilayer Neural Network

The abovementioned single-layer network is not sufficient for more complex tasks. Complex tasks may only be resolved with the help of multilayer networks whose general structure in shown in Figure 2. (Note: The upper index of the variables ***w, a, n, b*** does not represent a power but the element belonging to a layer.) You can see the interconnection *R* x *S*, where *R* is the number of inputs and *S* the number of layers.
The same equation applies to this multilayer network as was used in the case of the perceptron, but in the matrix shape **n = f (w p + b)**. [1],[2]

Figure 2: Multilayer neural network



3. Case Study

The subject of the testing is use of the abovementioned method for prediction and check of reliability of the prediction. The case study checks reliability of the predicted data, in our case car prices, using the Microsoft Neural Network algorithm over Business Intelligence Microsoft SQL Server 2008. The case involves verification of the zero hypothesis that the price difference is statistically insignificant.

The prediction is based on the used car database represented by a table with input data such as car type, car equipment, engine type, colour type (metallic or not), engine output, production date, mileage and price. The car prices will be used as the values for the prediction testing.

Part of the input data is included in Table 1.

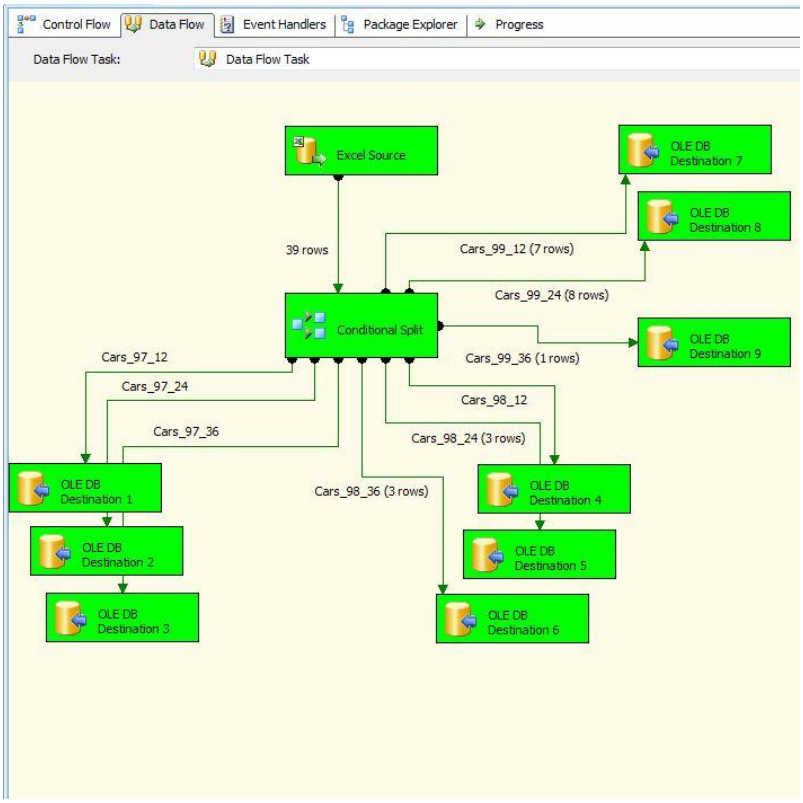
Table 1: Part of input data for prediction

ID	Type	Equipment	Engine	Colour	E power	Year	Mileage	Price
1	2	2	2	2	55	99	17725	287630
2	2	3	1	1	50	99	9289	287160
3	2	2	2	1	55	99	1879	296520
4	2	2	2	2	55	98	23989	255160
5	3	4	4	2	66	97	40230	309970
6	4	4	3	1	110	98	24968	427660
7	2	2	2	2	55	99	8425	272688

3.1. Input Data Preparation

Data entry for the purpose of analytical service is one of the major parts of the whole project. In our case the table in the Microsot Excel format may be imported directly to a database table. As the import is a singular action and periodic data entry is required in practice this study demonstrates use of the Integration Services of the SQL Server which can be used for the whole process automation.

Figure 3: Simple integration process for data entry in tables



The abovementioned Integration Process uses a source table in MS Excel format from which data are transferred to Conditional Split, where the input data are split to related groups classified by year of manufacture, engine type, engine output, equipment and mileage on the basis of the following three conditions:

Year == 97 && (Engine == 1 || Engine == 2) && (E_power == 50 || E_power == 55) && (Equipment == 2 || Equipment == 3) && (Kilometers >= 0 && Kilometers <= 9999)

Year == 97 && (Engine == 1 || Engine == 2) && (E_power == 50 || E_power == 55) && (Equipment == 2 || Equipment == 3) && (Kilometers >= 10000 && Kilometers <= 19999)

Year == 97 && (Engine == 1 || Engine == 2) && (E_power == 50 || E_power == 55) && (Equipment == 2 || Equipment == 3) && (Kilometers >= 20000 && Kilometers <= 35999)

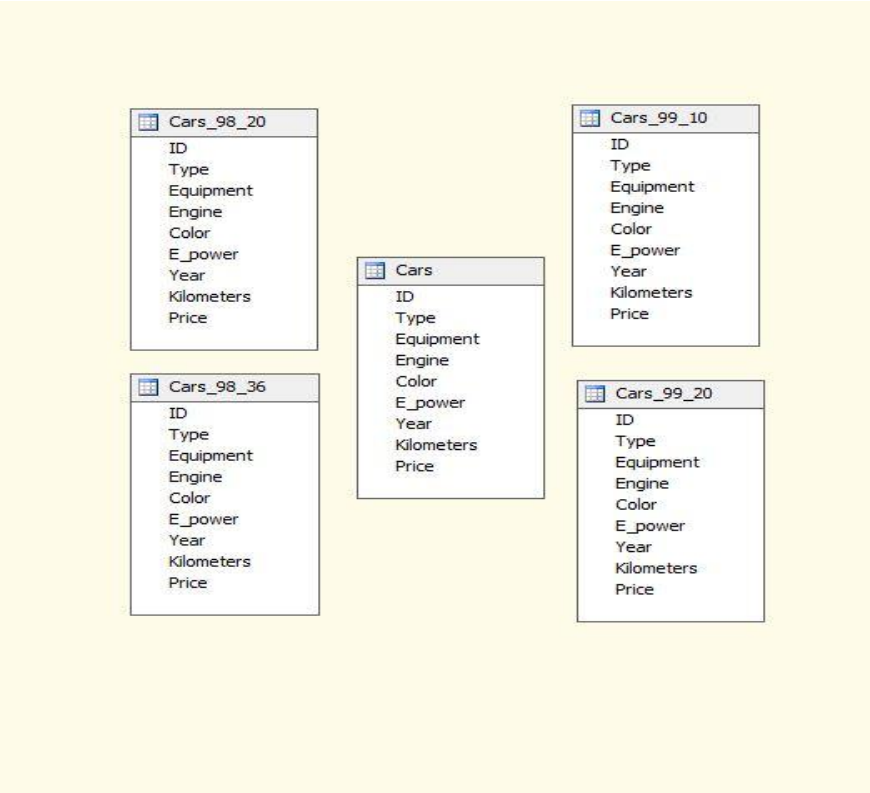
where || represents OR operator and && represents AND operator. The same conditions are then used for classification in the following years 98 and 99.

Thus classified data are then entered in separate tables on the SQL Server.

3.2. Analytical Project Preparation

The analytical project included five tables shown in Figure 4., where the Cars table was the testing (learning) table, where the algorithm learned to predict the car price on the basis of input data. The other tables were then designed for the actual car price predictions.

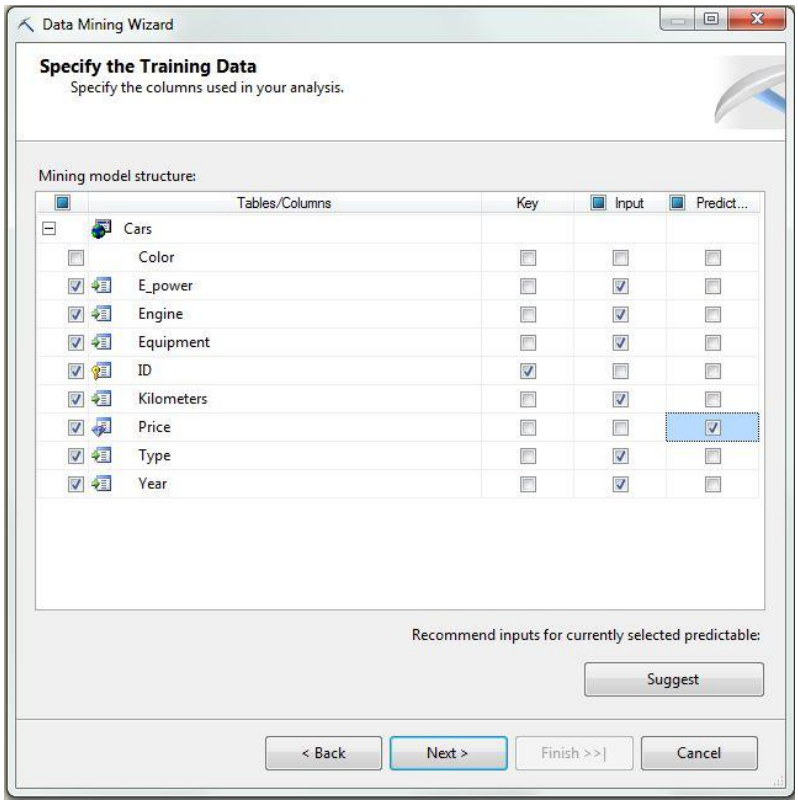
Figure 4: Testing table and prediction tables



3.3. Algorithm Selection and Data Specification

This step involves selection of the Microsoft Neuron Network algorithm used for teaching the model (testing table) including specification of the input data and the predicted quantity.

Figure 5: Specification of data for algorithm



The key column will be ID, and the predicted column will be Price. The other columns will include the algorithm input data except for the Colour column which would enter the algorithm with the lowest score. The input data scores are shown in Figure 6.

The data from the Cars table will subsequently be randomly divided to the trainign testing quantity. Regarding the size of the input file the training quantity was set to 30% without limit of the number of data for traning the model. The result of the algorithm is then shown in Figure 7. This tab allows for analysis of the results of the algorithm, including the individual links and relations.

Figure 6: Input data score

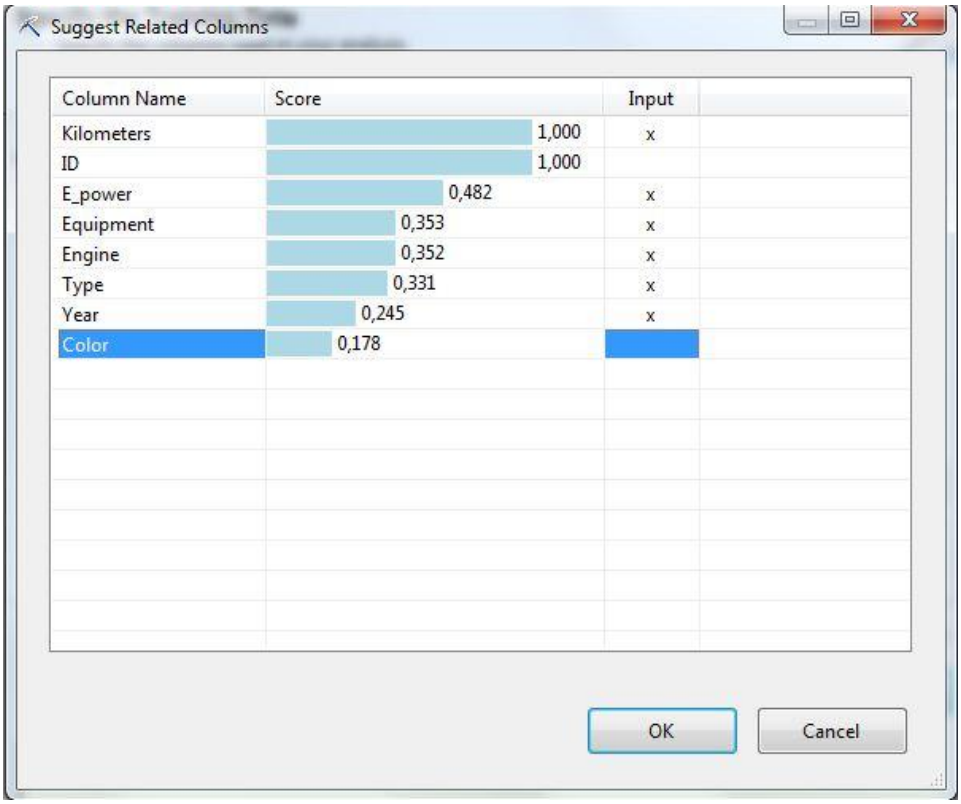
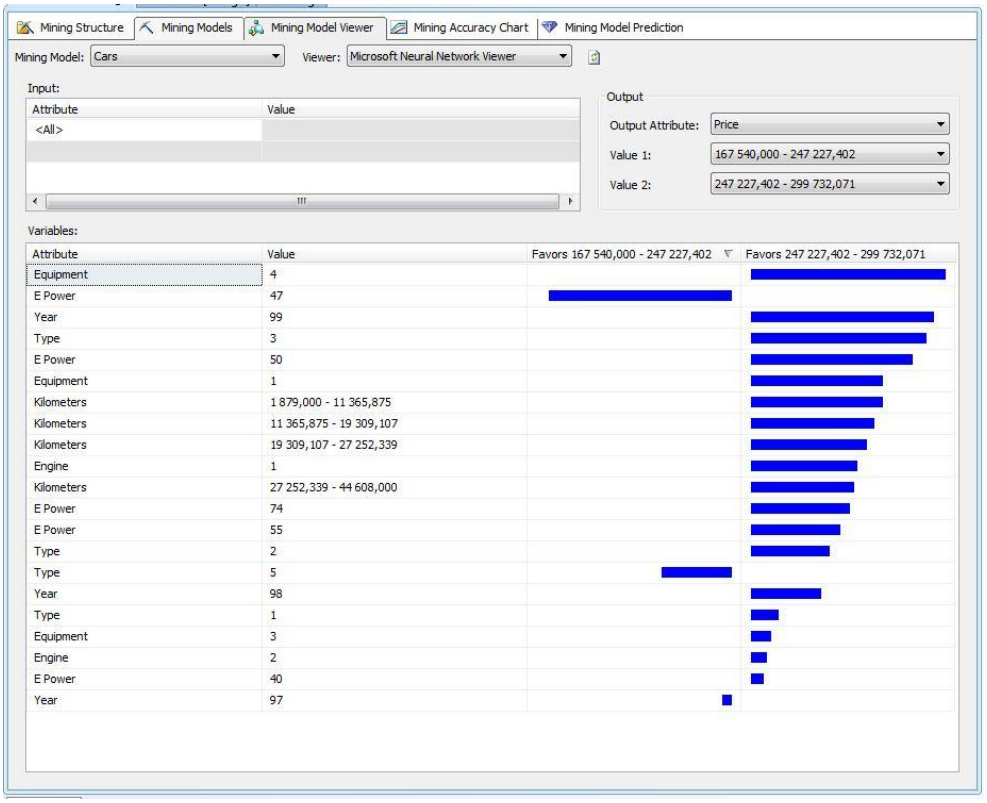


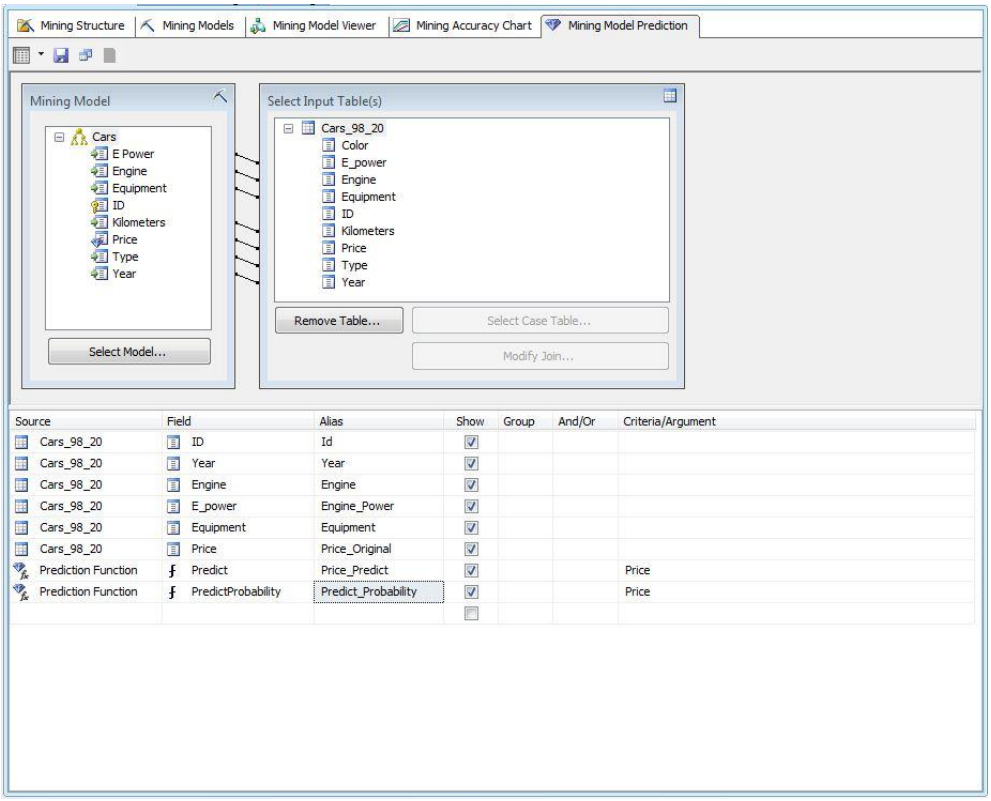
Figure 7: Result of Microsot Neural Network algorithm



3.4. Prediction

The prediction itself will be based on the tables created by the Integration Process. The tables will be used for car price predictions on the basis of the created model. Prediction setting is shown in Figure 8, where the Mining Model window includes coloumns of the model and the Select Input includes columns of the prediction table. On the basis of this coloumn the prediction inquiry mat be generated. The inquiry is shown in the bottom part of the screen.

Figure 8: Predict



The result of the predictive inquiry can then be entered as one of the options in the table on SQL Server. Table 2 shows the result of the prediction in the context of this case study. The table summarises all prediciton results for the tables of the Integration Process. This table shows the predicted price, in the Price Predict column, as well as the probability of the prediction in the Predict Probability column.

The Price Difference column includes the differnece between the original price (Price Original) and the predicted price (Price Predict).

Table 2: Prediction result

Id	Year	Engine	Engine Power	Equipment	Mileage	Price Predict	Predict Probability	Price Original	Price Difference
18	98	2	55	2	15729	252853	0,95	265970	**13117**
19	98	1	50	3	19289	243556	0,95	254870	**11314**
32	98	1	50	3	13212	243879	0,95	253740	**9861**
4	98	2	55	2	23989	252515	0,95	255160	**2645**
21	98	1	50	3	31453	244313	0,95	228290	**-16023**
33	98	2	55	2	29021	252315	0,95	252340	**25**
2	99	1	50	3	9289	277938	0,95	287160	**9222**
3	99	2	55	2	1879	279836	0,95	296520	**16684**
7	99	2	55	2	8425	278788	0,95	272688	**-6100**
11	99	1	50	3	7311	279024	0,95	276500	**-2524**
12	99	1	50	3	9068	278057	0,95	275920	**-2137**
13	99	2	55	2	7614	278919	0,95	286640	**7721**
14	99	2	55	2	9860	278552	0,95	284680	**6128**
1	99	2	55	2	17725	277106	0,95	287630	**10524**
10	99	2	55	2	12413	278114	0,95	267390	**-10724**
15	99	1	50	3	18437	273424	0,95	265140	**-8284**
16	99	1	50	3	11767	276625	0,95	271510	**-5115**
17	99	1	50	3	14035	275478	0,95	266830	**-8648**
22	99	2	55	2	12309	259764	0,95	261590	**1826**
36	99	2	55	2	18413	276964	0,95	283410	**6446**
37	99	1	50	3	11038	277005	0,95	272940	**-4065**

4. Hypothesis Test

Use of t-test is available for the two basic files (Price Original and Price Predict). As one of the basic prerequisites, independence of the values, is not met here (the original and the predicted price are always specified for the same car), we will view the found values as paired measurements. We will establish a new data file calculated as the difference between the original price and the predicted price (Price Difference). This new data file will be used for specification of the basic characteristics (mean = -1518.71, decisive tolerance = 8882.73).
Then we will introduce the zero hypothesis, in our case stating normal distribution of the difference between the original price and the predicted price. To confirm the hypothesis the t-test for paired measurements will be used. The only prerequisite of the t-test will be data normality (verified by Shapiro-Wilk Normality test).

Test result: p-value= 0.793, on the significance level of α=0.05
On the basis of the performed test the zero hypothesis is not rejected.

Then we will introduce the zero hypothesis and the alternative hypothesis for the price difference. The zero hypothesis in this case will mean statistically insignificant difference between the prices. In our case the hypothesis may also be formulated as the original price difference from the predicted price being statistically insignificant.

The alternative hypothesis will then be formulated as a statistically significant price difference. In our case the hypothesis may also be formulated as the original price being statistically significantly different from the predicted price.
Then we will calculate the test criterion t=-0.784. For the selected significance level α=0.05 the critical value will be specified. As we use the t-test, the critical value will be a quantile of the Student classification $t_{0,975}(20)$=2.086.
The results show that the test criterion absolute value is lower than the critical value. We will therefore accept the zero hypothesis. That means that there is a statistically insignificant difference between the prices (the original and the predicted price do not statistically significantly differ from each other).

5. Conclusion

This case study tried to show that the original and the predicted car prices were not statistically significantly different from each other, which was proved.

The present article tries to point out the option of use of soft computing in databases, in particular in Microsoft SQL Server 2008, which can thus provide to companies a unified solution for decision-making process support. Finally soft computing and Microsoft SQL Server 2008 may be said to represent a suitable alternative for managerial decision-making support.