Regression Analysis

SAP BW Release 3.5

Copyright

Contents

An Introduction to Regression Analysis

Regression Analysis is a statistical technique of estimating the conditional expected value of one variable *Y*, given the values of some other variable or variables *X.* The variable of interest, *Y*, is called the *dependent variable*. The other variables, X, are called the *independent variables*.

You can use regression analysis to make predictions about future customer behavior. You create a model in the data mining application to make predictions. After a model has been created based on historical data, it can then be applied to new data to make predictions. The prediction, that is, the output of the model is called a *Score*. You can create a single score for your customers by taking into account different dimensions.

There are two main types of regression methods that SAP provides:

- Regression Analysis
- Nonlinear Regression Analysis

Linear Regression: Linear regression is called "linear" because the relation of the dependent to the independent variables is a linear function of some parameters.

A linear regression line has an equation of the form ***Y = a + bX***, where ***X*** is the explanatory variable and ***Y*** is the dependent variable. The slope of the line is ***b***, and ***a*** is the intercept (the value of ***y*** when ***x*** = 0).

The *Linear Regression* in data mining enables you to automatically define a valuation function by generating a linear approximation of a numeric target figure. To do this, you need historic data in which the target value is already known.

Nonlinear Regression: Regression models which are not a linear function of the parameters are called *nonlinear regression* models. The *Nonlinear Regression* function in data mining enables you to automatically define a valuation function by generating a nonlinear approximation of a numeric target figure (using splines of order 1). This also requires you to have historic data in which the target figure is already known.

After you have determined the valuation function either by defining it directly or by training it on the basis of historic data, you can then apply it to other data sets for prediction purposes.

Use and Applications

Linear Regression

A beverage outlet wants to attract customers to its product range by introducing a product from a higher price category. For this purpose, the beverage outlet wants to estimate its revenue potential in the drinks market. Let us assume that the revenue from the sale of drinks has a linear dependency on income and household size. A *linear regression* is performed on data where the revenue is already known, that is, last year's sales revenue figures. You train this data to determine the influence that income, household size, and region have on the revenue from the sale of drinks. You can then apply this trained data to the new prospects in order to calculate the potential revenue from this market segment.

Nonlinear Regression

The beverage outlet also wants to investigate the relevance of the attribute "age" for its potential revenue in the drinks market. Revenue here is unlikely to have a linear dependency on age. You can analyze nonlinear dependencies by using *nonlinear regression*.

In another example, a newspaper publisher wants to identify customers with a high propensity to churn, that is, customers who have a high probability of canceling their newspaper subscriptions. The publisher's customer database contains details relating to age, income, household size, and academic qualifications, length of the subscription, and region, as well as a field for canceled subscriptions. If a customer canceled their subscription in the past quarter, this field contains the value 1, otherwise it contains the value 0.

The data is trained using the *Nonlinear Regression* method. The result of training should show the relationship between the different customer attributes and the canceled subscription field. The trained function generates a value for each customer in the customer database, and this value can be used to reflect that customer's propensity to churn.

Typical Input

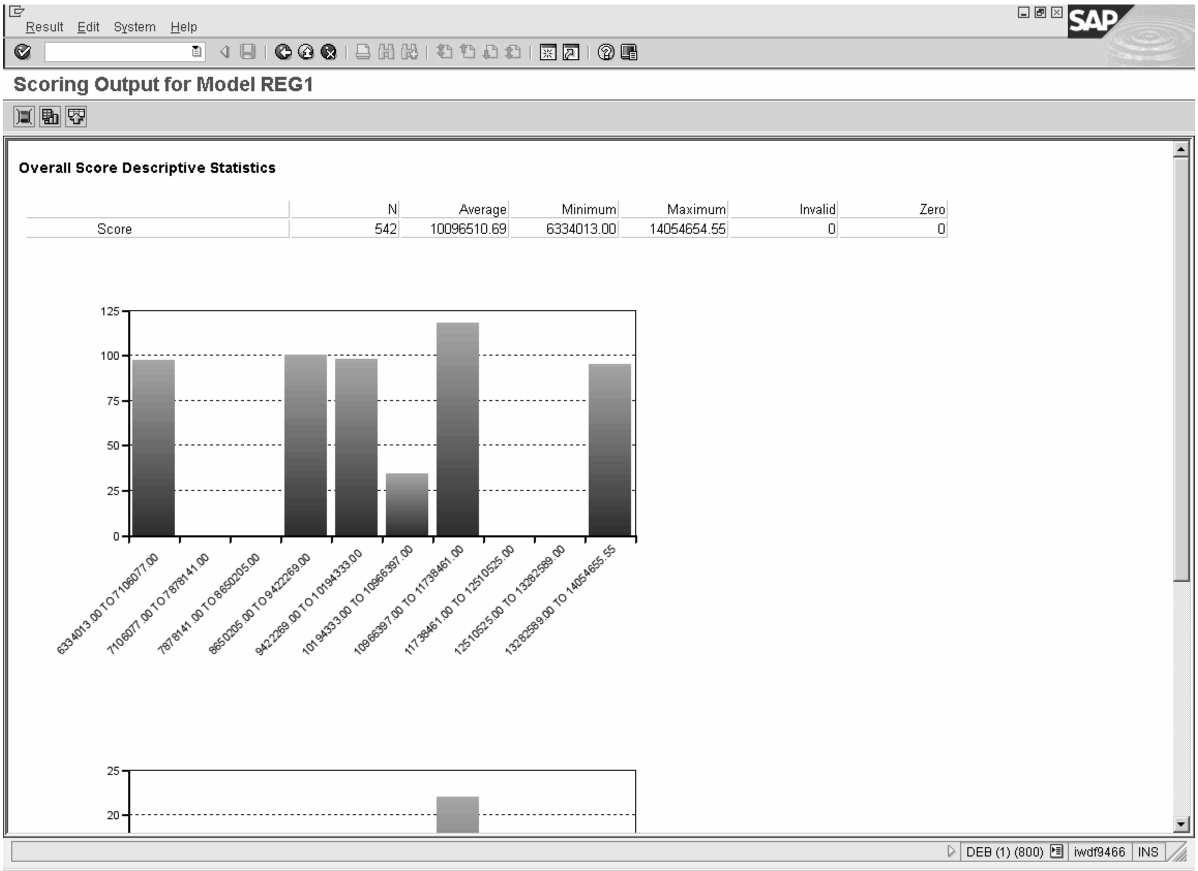
The following table contains data that could make up part of the typical input data for a regression function. The *business partner* is the table’s key, and *district* and *gender* are discrete fields. The remaining fields are continuous.

Business Partner	Revenue	CAM: District	Age	Family Size	Income	Gender
401015	2.825,00	SALT LAKE CITY	25	1	0.00 - 10000.00 USD	1
401016	707,00	DENVER	21	1	0.00 - 10000.00 USD	1
401017	3.078,00	SALT LAKE CITY	22	1	10000.01 - 25000.00 USD	1
401018	2.732,00	SALT LAKE CITY	19	1	10000.01 - 25000.00 USD	2
401019	2.664,00	SALT LAKE CITY	18	1	10000.01 - 25000.00 USD	1
401020	1.647,00	DENVER	25	1	10000.01 - 25000.00 USD	2
401021	2.718,00	DENVER	20	1	10000.01 - 25000.00 USD	1
401022	1.679,00	DENVER	24	1	10000.01 - 25000.00 USD	1
401023	3.034,00	SALT LAKE CITY	22	1	10000.01 - 25000.00 USD	1
401024	4.138,00	SAN DIEGO	31	3	25000.01 - 40000.00 USD	2
401025	3.745,00	SALT LAKE CITY	27	1	25000.01 - 40000.00 USD	1
401026	3.974,00	SAN DIEGO	36	2	25000.01 - 40000.00 USD	1

In the case of both linear and nonlinear regression, you could use this data to train a model with the prediction field *Revenue*, for example. You could then apply the trained model on data that does not contain the field *Revenue* to predict or estimate revenue.

Typical Output

The results can be displayed graphically in aggregated or in tabular form showing the individual records and their result values.



Scoring Output for Model REG1

Overall Score Descriptive Statistics

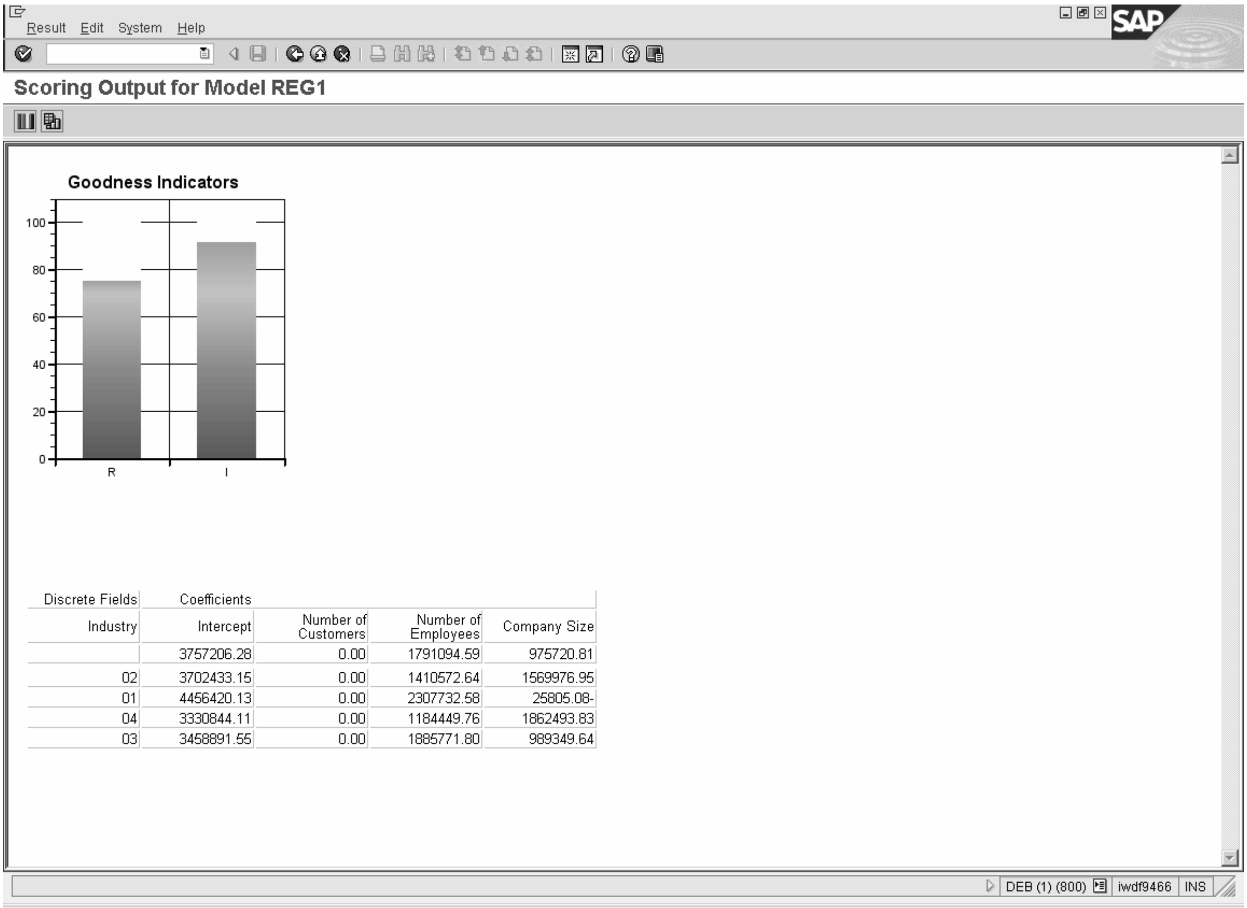
	N	Average	Minimum	Maximum	Invalid	Zero
Score	542	10096510.69	6334013.00	14054654.55	0	0

125
100
75
50
25
0
6334013.00 TO 7106077.00
7106077.00 TO 7878141.00
7878141.00 TO 8650205.00
8650205.00 TO 9422269.00
9422269.00 TO 10194333.00
10194333.00 TO 10966397.00
10966397.00 TO 11738461.00
11738461.00 TO 12510525.00
12510525.00 TO 13282589.00
13282589.00 TO 14054655.55
25
20

You can display the data from various sources for the selected model fields as aggregated values. These values include the number of records, minimum and maximum values, totals and the percentage distributions of the prediction variables.

In the above example, the overall score is displayed for the model with the prediction variable as *Revenue.*

The coefficients for the model are displayed. To display the Goodness Indicators for the regression analysis models, click on the icon.



Scoring Output for Model REG1

Discrete Fields	Coefficients			
Industry	Intercept	Number of Customers	Number of Employees	Company Size
	3757206.28	0.00	1791094.59	975720.81
02	3702433.15	0.00	1410572.64	1569976.95
01	4456420.13	0.00	2307732.58	25805.08-
04	3330844.11	0.00	1184449.76	1862493.83
03	3458891.55	0.00	1885771.80	989349.64

You use the goodness indicators to check how good the model is for prediction purposes. If the goodness indicators tend towards green, then the model is a good model for prediction of the given distribution of data. Hence, the calculated values of the coefficients, provide a better prediction if the values of R and I tend towards 100, that is, tend towards green.

The following formulae are used to calculate the values for R and I:

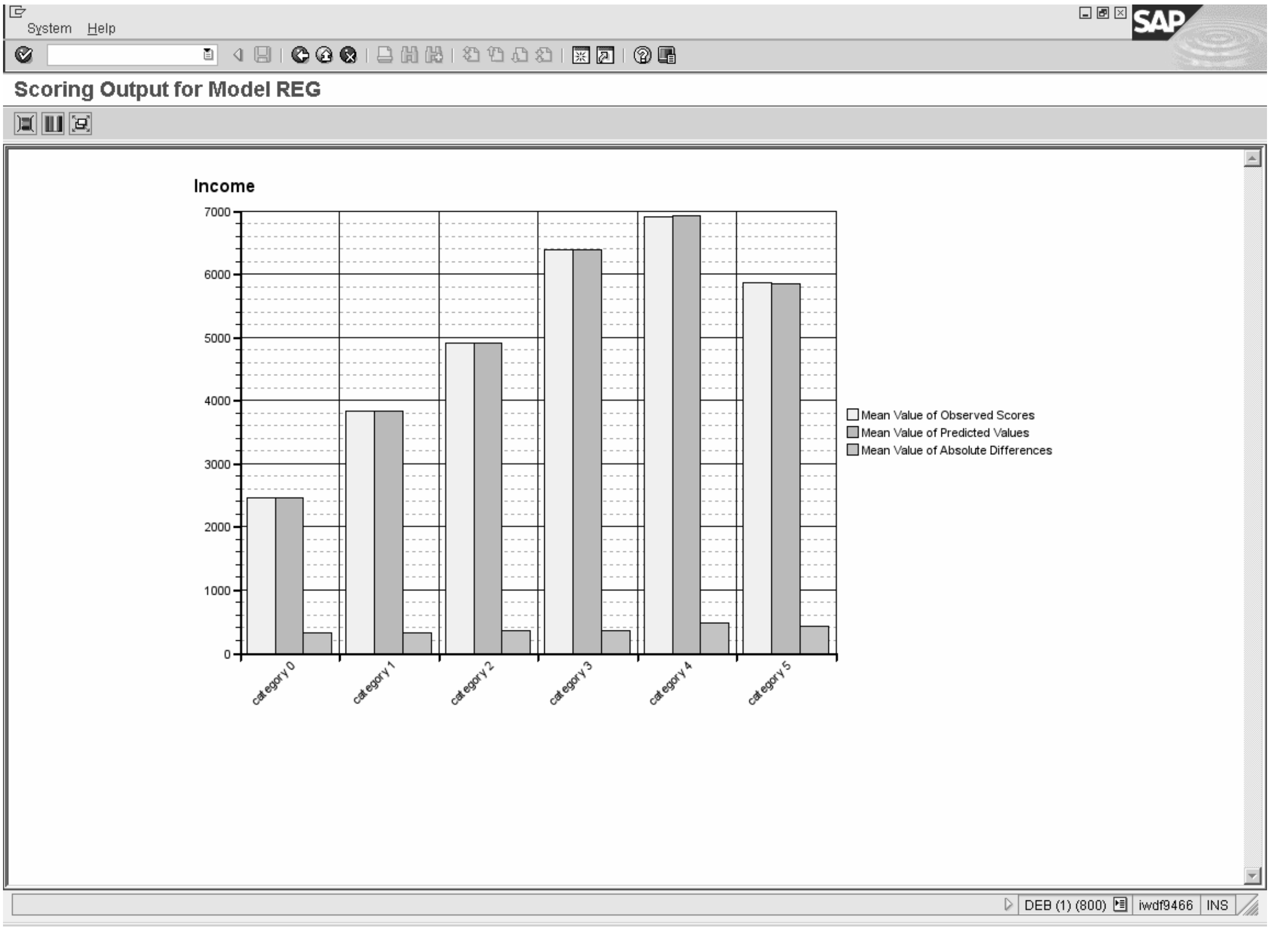
$$R := \sqrt{1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

$$I := 1 - \frac{\sum_{i=1}^{n} | y_i - \hat{y}_i |}{\sum_{i=1}^{n} | y_i | + | \hat{y}_i |}$$

where y_i are the observed values,
$\hat{y}_i$ are the predicted values and
$\bar{y}$ is the mean of the observed values.

Both indicators take the value 1, if the predicted values are equal to the observed ones. They tend against 0, when the differences become bigger.

In the case of linear or nonlinear regression, you can use the training data (here, *Income*) to display the mean value of the absolute differences of the observed and predicted values to assess the quality of the approximation.



Data Mining Functions in the Analysis Process Designer (APD)

The Analysis Process Designer (APD) is the application environment for the SAP data mining solution. From SAP BW Release 3.5, data mining functions are fully integrated into the APD. You can perform the following functions in the APD:

- Creating and changing data mining models
- Training data mining models with SAP BW data (data mining model as data target in the analysis process)
- Execution of data mining methods such as prediction with decision tree, with cluster model and integration of data mining models from third parties (data mining model as a transformation in the analysis process)
- Visualization of data mining models

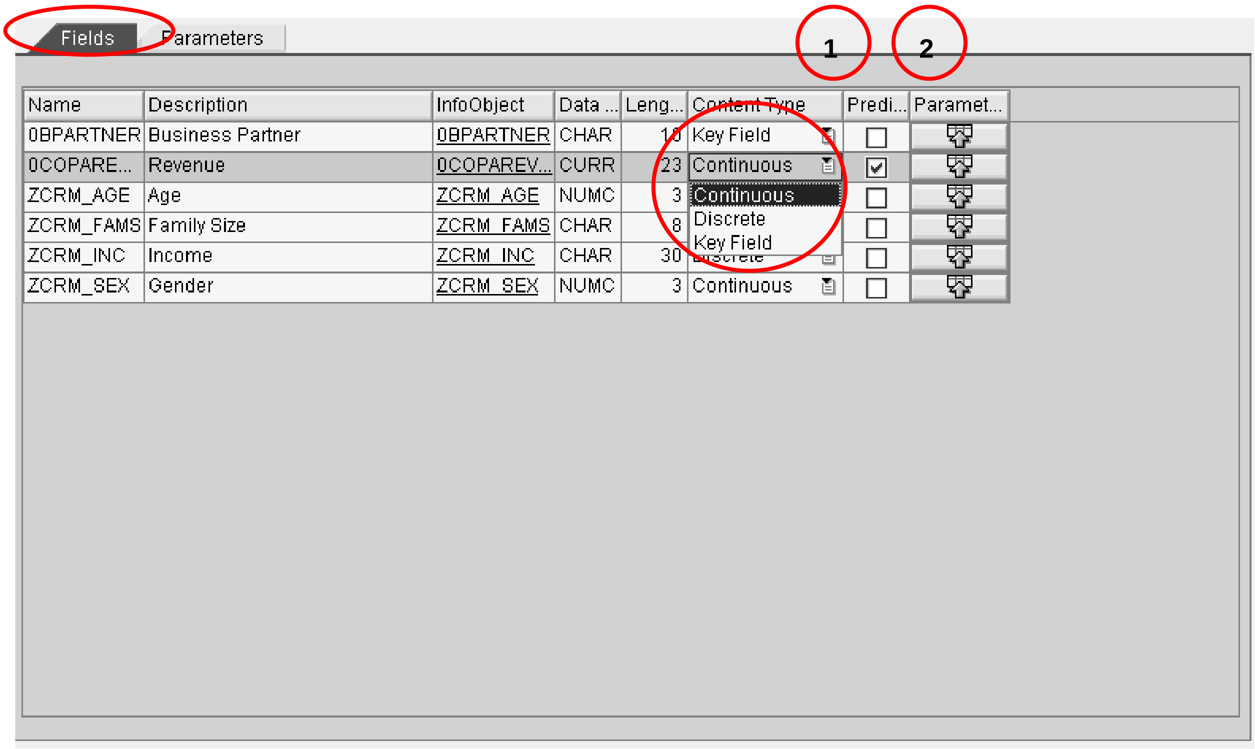
For more information, see SAP Library at help.sap.com under *SAP NetWeaver -> Release '04 -> Information Integration -> SAP Business Information Warehouse -> BI Platform -> Analysis Process Designer / Data Mining*

Settings for Regression Analysis

The input data for SAP’s Regression Analysis is divided into two parts:

- Model Fields
- Model Parameters

Model Fields for Linear and Nonlinear Regression Analysis



Name	Description	InfoObject	Data ...	Leng...	Content Type	Predi...	Paramet...
0BPARTNER	Business Partner	0BPARTNER	CHAR	10	Key Field	□	
0COPARE...	Revenue	0COPAREV...	CURR	23	Continuous	☑	
ZCRM_AGE	Age	ZCRM_AGE	NUMC	3		□	
ZCRM_FAMS	Family Size	ZCRM_FAMS	CHAR	8		□	
ZCRM_INC	Income	ZCRM_INC	CHAR	30	Discrete	□	
ZCRM_SEX	Gender	ZCRM_SEX	NUMC	3	Continuous	□	

Model fields are the attributes that define the object and the predictable field is the class label. In *Model Fields* screen, you can add the fields that are required for creating a regression analysis model. You must define the content type for each model field.

(1) Content Type

It defines the data in a model field. There are 4 content types for model fields used in decision tree classification.

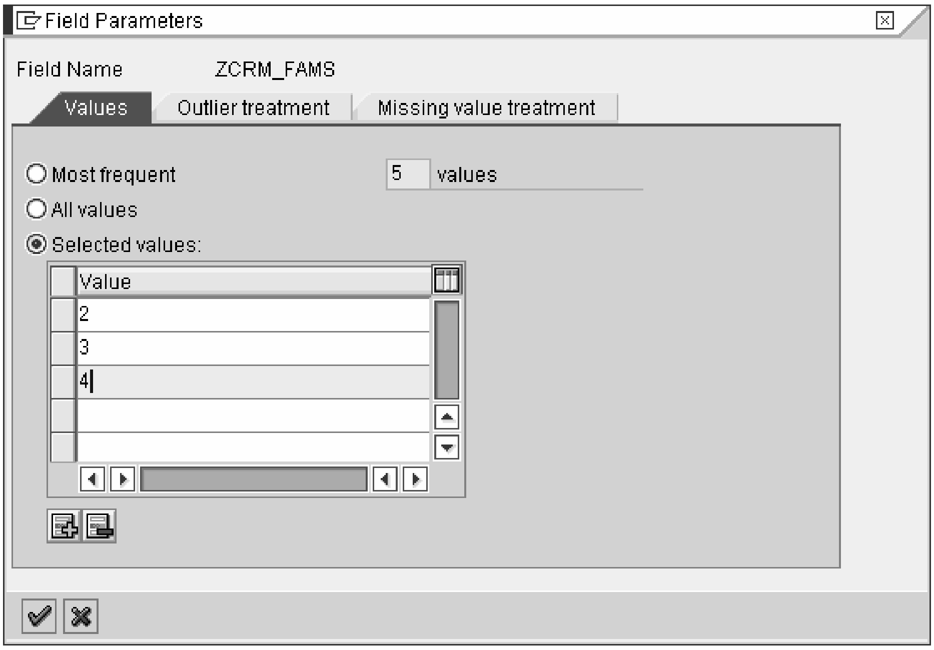
Key field: The key field acts as the record identifier. This field does not have any influence on the outcome.

Discrete: Also referred to as categorical, the data in the model field for this content type contains a finite set of values. For example, a model field ‘Gender’ has two values - Male and Female. Attributes like Color, Gender, and Status are examples of discrete attributes.

Continuous: Continuous data can have any value in an interval of real numbers. This impli es that the value does not have to be an integer. Attributes having infinite set of possible real values are called Continuous. Typically, they have a Minimum and Maximum value and attribute values could be anything within this interval. Attributes like Salary, Sales Revenue, Quantity sold etc are examples of Continuous attributes. You can discretize a Continuous attribute by defining fixed intervals. For example, if the salary ranges from $100 to $20000, then we can form intervals like $0 – 2000, $2000 – $4000, $4000 – $6000…. $18000 – $20000. An attribute value will fall into any one of these intervals.

(2) Field Parameters for Linear Regression

For **discrete** model fields, you can specify which values should be considered: only specific values, the most frequently occurring values, or all values:



Field Parameters

Field Name ZCRM_FAMS

Values | Outlier treatment | Missing value treatment

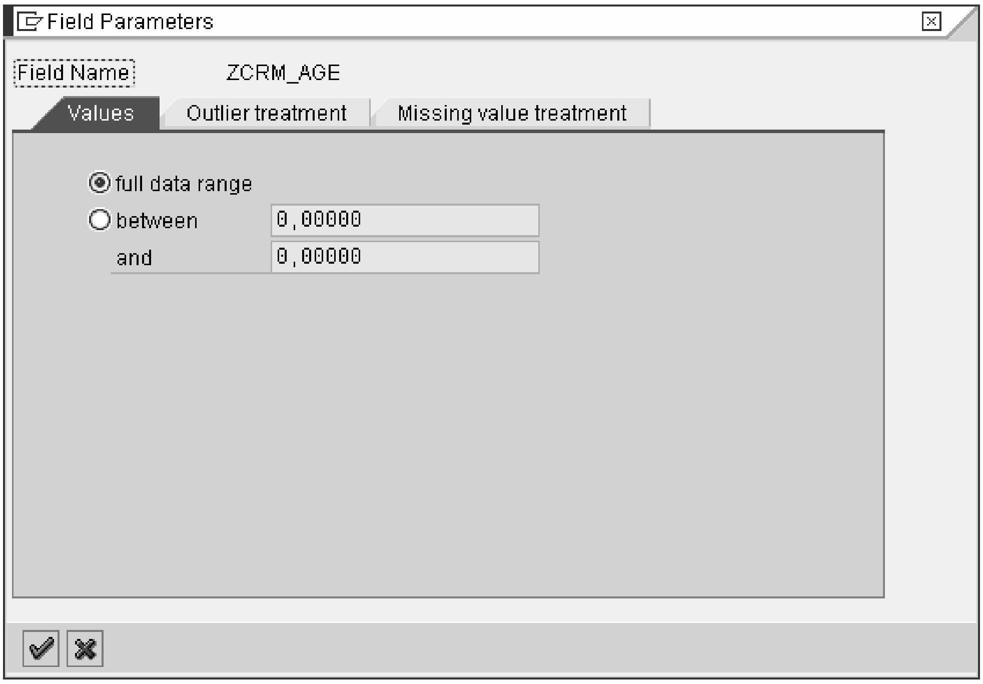
○ Most frequent 5 values

○ All values

● Selected values:

Value
2
3
4

For the **continuous** model fields, you can specify both borders of a value range explicitly, or you can choose the option *full data range* for the borders, that is, the lower interval and the upper interval to be specified automatically.



Field Parameters

Field Name ZCRM_AGE

Values | Outlier treatment | Missing value treatment

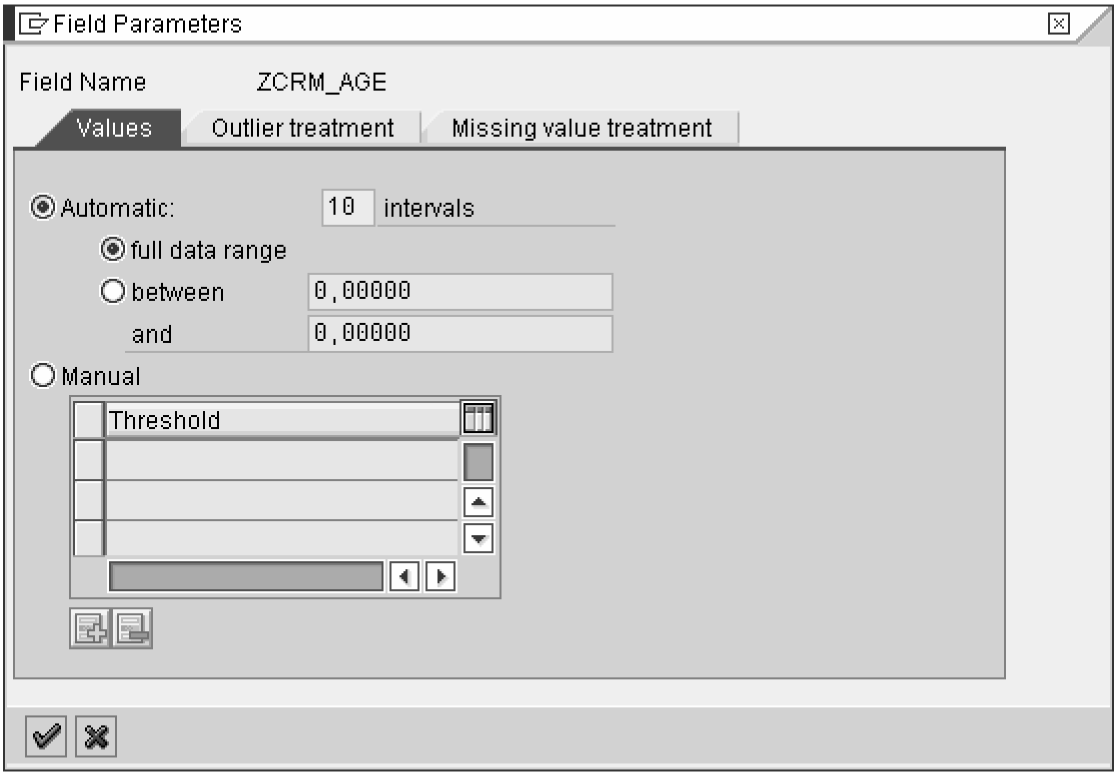
● full data range

○ between 0,00000

and 0,00000

(2) Field Parameters for Nonlinear Regression

The treatment of **discrete** field values is the same as that for Linear Regression. The treatment of **continuous** field values differs slightly though.



Field Parameters

Field Name ZCRM_AGE

● Automatic: 10 intervals

● full data range

○ between 0,00000

and 0,00000

○ Manual

You have to split the value ranges of **continuous** model fields into intervals. As in the case of linear regression, you can choose to have both outer borders of the interval determined automatically or you can specify them explicitly. Then you can specify the desired number of intervals of the same size. Alternatively, you can enter the interval borders explicitly.

Field Parameters for Outlier Treatment

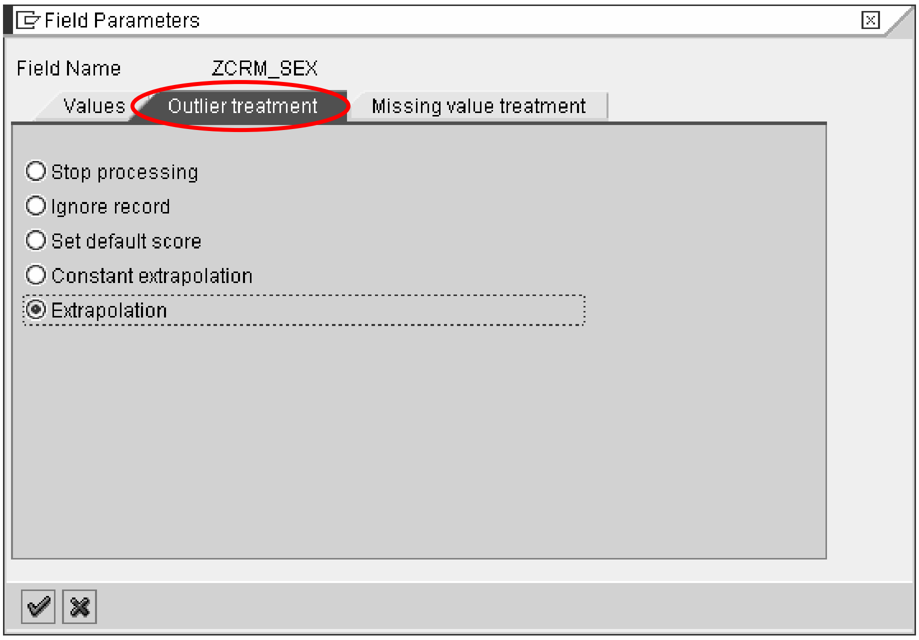
In both Linear and Nonlinear Regression methods, the parameters for the model fields offer control options for treating outliers.

Which values are treated as outliers?

For **discrete** model fields, outliers are values that do not belong to the values specified explicitly or to the most frequently occurring values.

For **continuous** model fields, outliers are values falling outside of the outer borders that are determined during the definition of the value ranges, either explicitly or automatically.

You can make an outlier treatment setting to decide whether processing is stopped, the record is ignored, or the default score is set when a record occurs containing an outlier.



Field Parameters

Field Name ZCRM_SEX

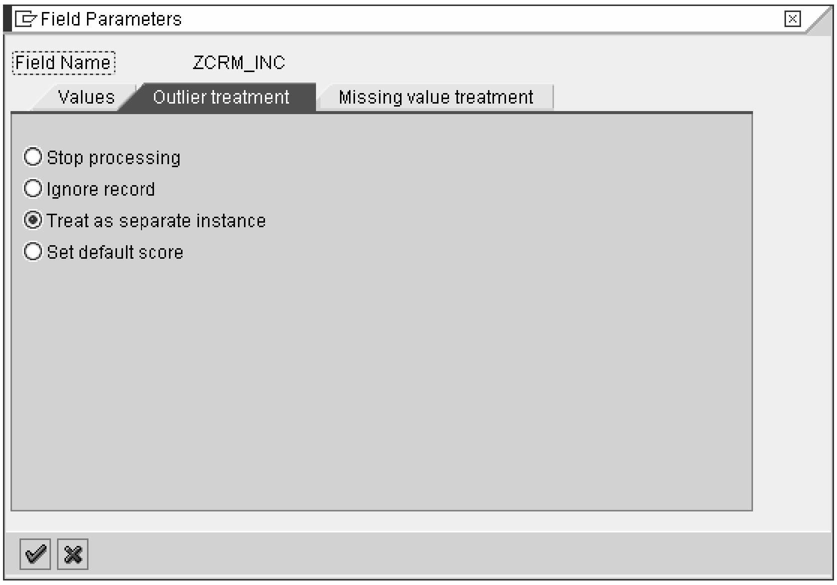
○ Stop processing
○ Ignore record
○ Set default score
○ Constant extrapolation
● Extrapolation

Continuous Model Fields

For continuous model fields, you can specify that an extrapolation is applied.

Extrapolation: Whenever a linear regression model is fit to a group of data, the range of the data should be carefully specified. Using the regression method, it is possible to make predictions for values outside this specified range of data. This process is known as ***extrapolation***. If you choose the option Extrapolation, the outliers, that is, the values lying outside the specified range of values will not be treated separately.

Constant Extrapolation: If you select this option, then the function is constantly extrapolated beyond the external borders.



Field Name ZCRM_INC

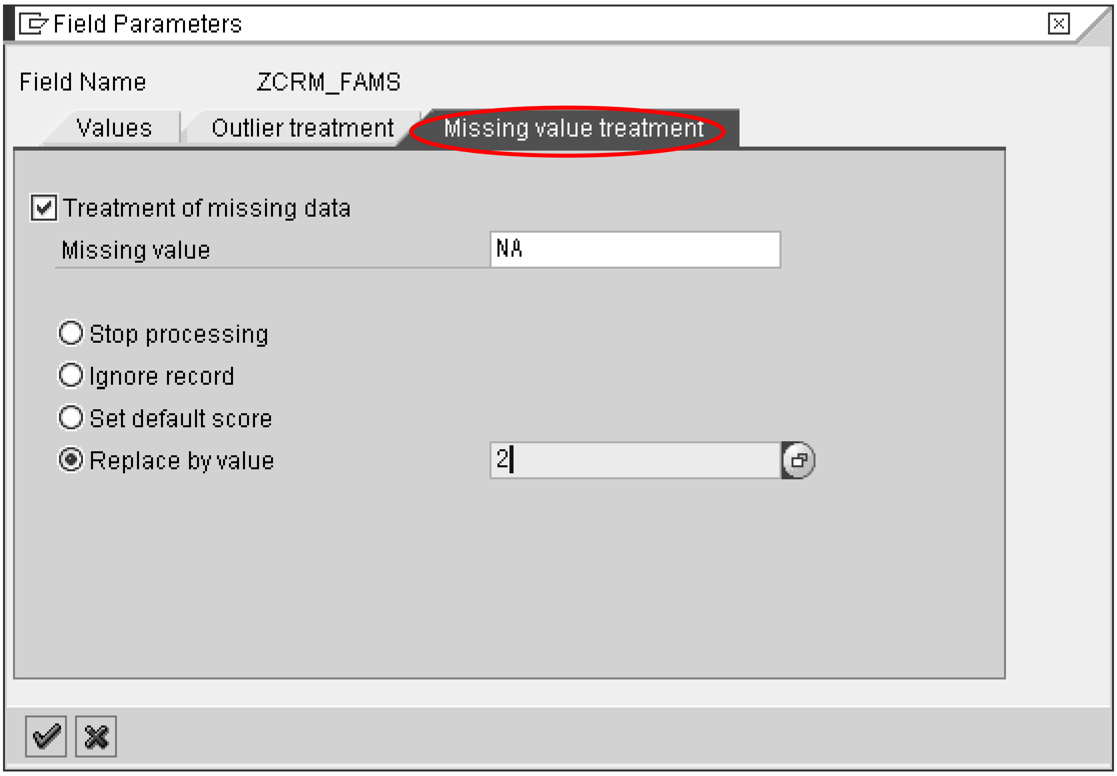
○ Stop processing
○ Ignore record
● Treat as separate instance
○ Set default score

Discrete Model Fields

Treat as separate instance: This option is valid only for discrete model fields. By setting this option, all outliers are treated as a single, common remainder.

Field Parameters for Treating Missing Values

In both Linear and Nonlinear Regression methods, the parameters for the model fields offer control options for handling missing values.



Field Name ZCRM_FAMS

☑ Treatment of missing data

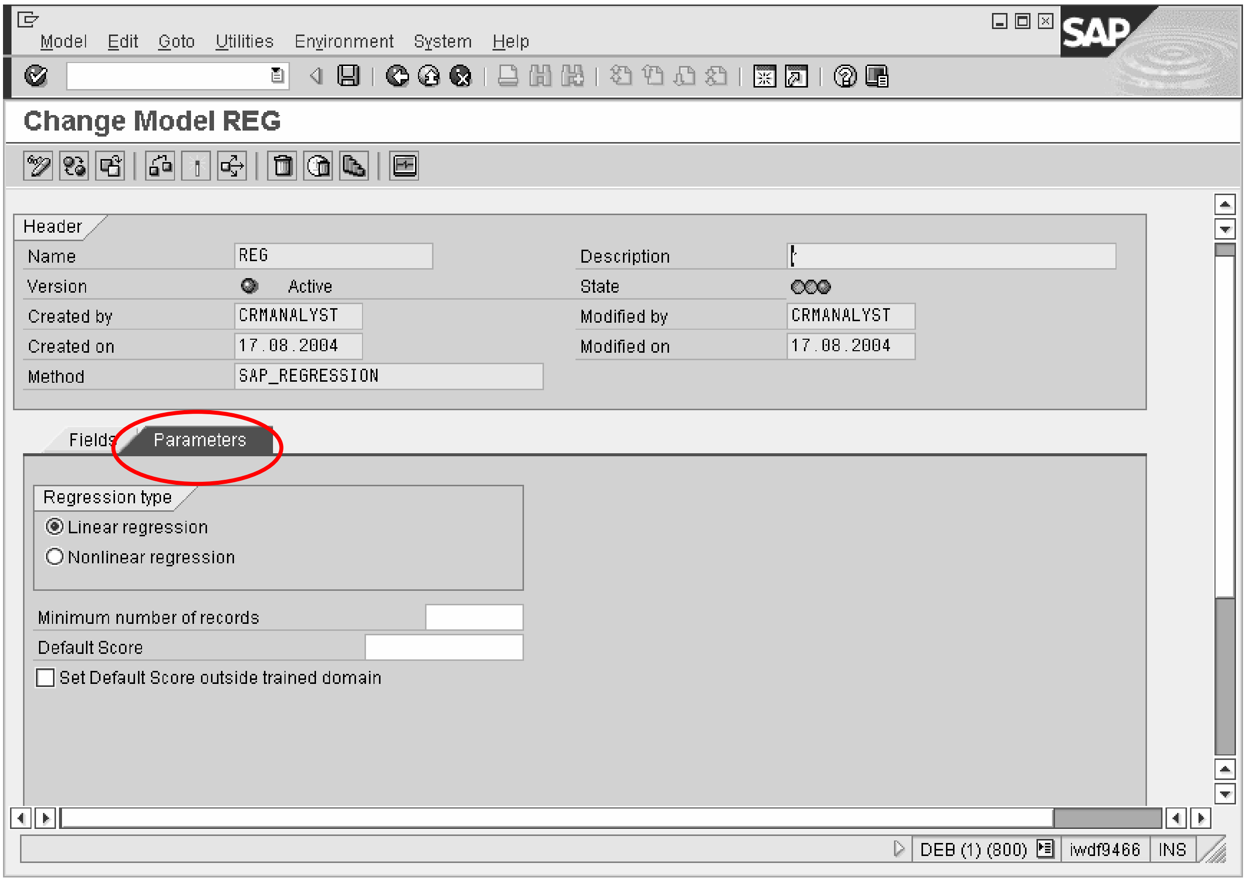
Missing value NA

○ Stop processing
○ Ignore record
○ Set default score
● Replace by value 2

For *treating missing values,* you first have to set the appropriate indicator and identify a *missing value*. If, for example, the size of family is denoted by a numeric value and NA has been used to denote a value that is unknown, you can enter NA as the Missing Value. You define a separate treatment for this value accordingly.

You can make a setting to decide whether processing is stopped, the record is ignored, or the default score is set when a value defined in this way occurs. Using the option *Replace by value*, you can substitute the missing value with another value.

Model Parameters for Regression Analysis



Change Model REG

Header

Name	REG	Description	
Version	Active	State	
Created by	CRMANALYST	Modified by	CRMANALYST
Created on	17.08.2004	Modified on	17.08.2004
Method	SAP_REGRESSION		

Regression type

● Linear regression

○ Nonlinear regression

Minimum number of records

Default Score

□ Set Default Score outside trained domain

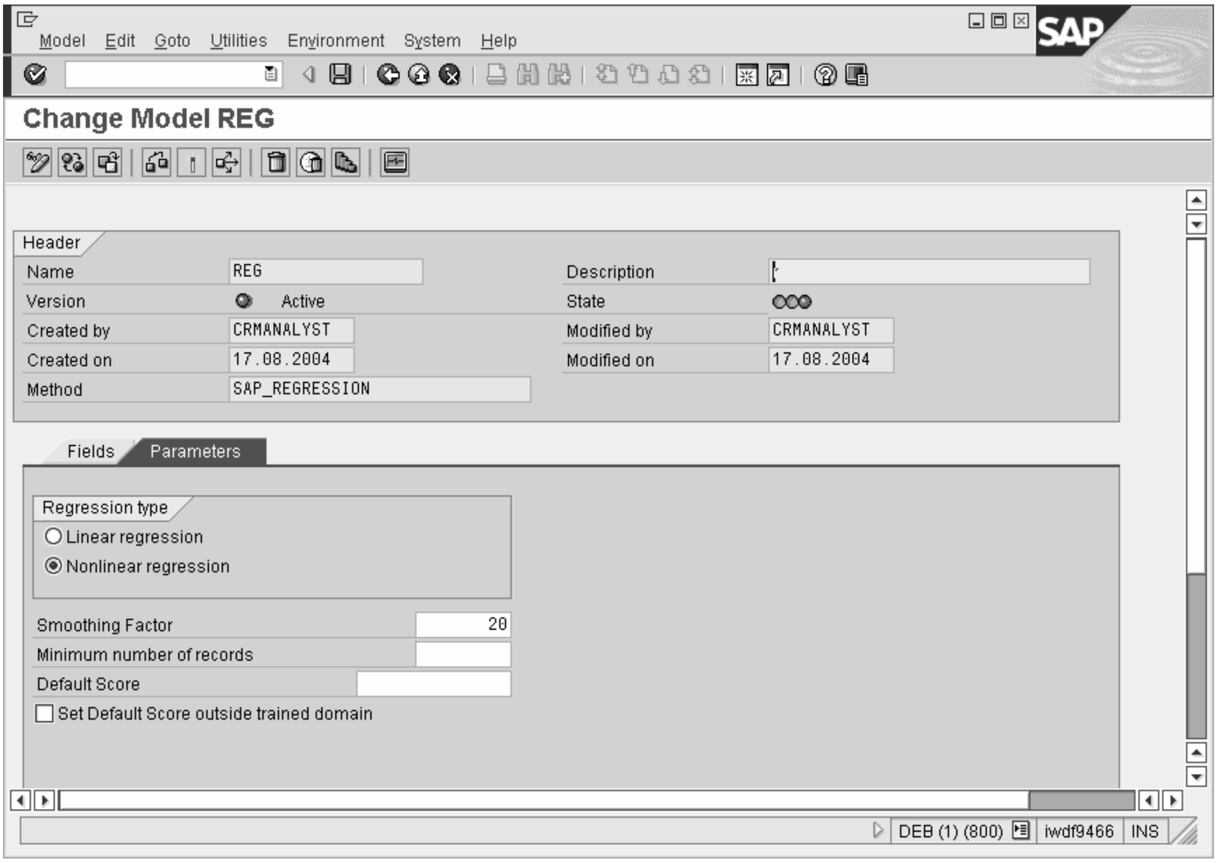
Regression Type: You use this parameter to specify which procedure should be applied by the regression algorithm, that is, linear regression or nonlinear regression.

Default Score: You use this parameter to specify a default output value for a regression function. If required, this value is always set whenever a record does not fulfill certain conditions (for example, it has missing data or outliers). The default value for this field is 0 (zero).

Minimal Number of Records: You can use this parameter to restrict the data area to be applied in training. This is only defined for combinations of discrete values for which the number of records in the training data set attains or exceeds the parameter.

For example, a model has the discrete fields *Occupational Group* and *Vehicle Category*. In the data set used, there are 83 records for the combination *Computer Programmer* and *Sports Car*. If the value entered for this field is 50, then the model is trained using this data area. If the value is set at 100, then this data area is not taken as the basis, and the domain for the trained regression model excludes any records with this combination.

Set default score outside trained domain: If you set this indicator, the system ignores any records falling outside of the domain for the trained function when performing a prediction with a model that has already been trained. (See also the parameter "Minimum number of records".) If the indicator is not set, then the default score is applied to these records. Only whole numbers are permitted.



Change Model REG

Header

Name	REG	Description	
Version	Active	State	
Created by	CRMANALYST	Modified by	CRMANALYST
Created on	17.08.2004	Modified on	17.08.2004
Method	SAP_REGRESSION		

Fields | Parameters

Regression type

○ Linear regression

● Nonlinear regression

Smoothing Factor	20
Minimum number of records	
Default Score	

□ Set Default Score outside trained domain

Smoothing Factor: If you choose nonlinear regression, you need to indicate the smoothing factor in addition to all the other parameters. You use this parameter to specify whether and to what extent the regression function should be smoothed by the algorithm. Smoothing factors in the region of 10 through 100, for example, prevent the function from over fitting the training data. Using very large smoothing factors ultimately produces linear regression.

Algorithms

Linear Regression

Training for linear regression produces a function with the following form:

$$f(x_1,...,x_n) = a_0(x_{p+1},...,x_n) + a_1(x_{p+1},...,x_n) \cdot x_1 + ... + a_p(x_{p+1},...,x_n) \cdot x_p$$

$x_1,...,x_p$ are variables for the continuous model fields and $x_{p+1},...,x_n$ are the variables for the discrete model fields. This means that, for each discrete value set $x_{p+1},...,x_n$, a linear function is produced in the continuous variables $x_1,...,x_p$.

Let us assume that the amount of training data consists of *k* records $(x_{i1},...,x_{in},y_i),\ i = 1,...,k$, where y_i is the observed value of the prediction variables. During training, the parameters $a_0(x_{p+1},...,x_n)$,..., $a_p(x_{p+1},...,x_n)$ are produced for a given value set $(x_{p+1},...,x_n)$ so that the following sum is minimal:

$$\sum_{\substack{i=1 \\ x_{ip+1}=x_{p+1},...,x_{in}=x_n}}^{k} \left(y_i - f(x_{i1},...,x_{in})\right)^2$$

If the conditioning of the system of equations defined by this condition is poor because there is a linear dependency between the data in the different model fields, then the system of equations is adjusted slightly.

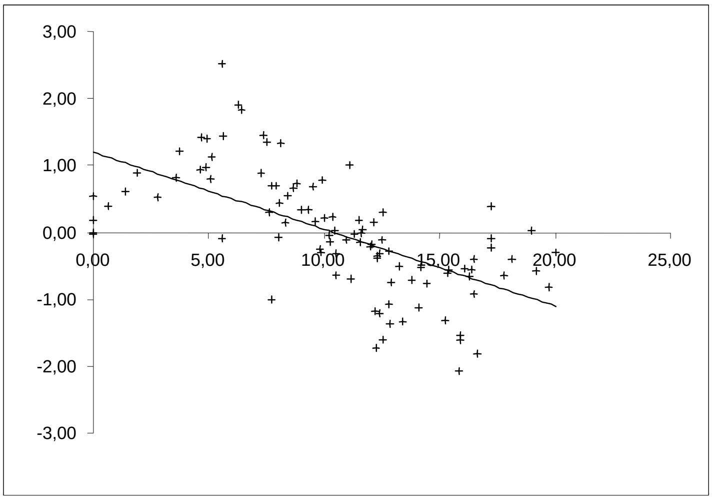
An interval is set for each continuous model field, either directly by the user or automatically by the rounded-off minimum and maximum values in the training data. Depending on the option selected by the user, the function can be continued as linear or continuous outside of the interval borders.

For example, you want to estimate the sales revenue to be made with prospects on the basis of their income. Working on the assumption that there is a linear dependency between the sales revenue and the income, you perform a linear regression on data in which the revenue is already known. After training the function on the basis of this data, you can then apply this function to customer data in which there is no sales revenue information but for which the potential sales revenue can be calculated on the basis of the customers' income.

The trained function has the following form:

Sales Revenue (Income) = a_0 + a_1 x Income

a_0 and a_1 are parameters that are determined automatically during training. When expressed in graphical form, the function produces a straight line that approximates the scatter plot representing the historic data.



If you would like to predict the sales revenue on the basis of both income and age, then the trained function takes the following form:

Sales Revenue (Income, Age) = a_0 + a_1 x Income + a_2 x Age.

The function creates a plane. If you include the discrete attribute *Region*, then a linear regression is performed automatically for **each** region. The parameters a_0, a_1, a_2 are then dependent on the region:

Sales Revenue (Income, Age, Region) = a_0(Region) + a_1(Region) x Income + a_2(Region) x Age.

Any other continuous or discrete attributes that you include in the function are handled in the same way.

Nonlinear Regression

Let $x_1,...,x_p$ be the variables for the continuous model fields and $x_{p+1},...,x_n$ be the variables for the discrete model fields. For each continuous model field, the value range is divided up into intervals – either automatically or directly by the user – and this division is determined by a sequence of interval borders. Let $t_{i1},...,t_{ij_i}$ be the interval borders of the ith model field. For $x_i \in [t_{i1}, t_{ij_i})$, the following is defined:

$$t_i^{(0)}(x_i) := \max\{t_{il} \mid t_{il} \leq x_i\}$$

$$t_i^{(1)}(x_i) := \min\{t_{il} \mid x_i < t_{il}\}$$

$$u_i^{(0)}(x_i) := \frac{t_i^{(1)}(x_i) - x_i}{t_i^{(1)}(x_i) - t_i^{(0)}(x_i)}$$

$$u_i^{(1)}(x_i) := \frac{x_i - t_i^{(0)}(x_i)}{t_i^{(1)}(x_i) - t_i^{(0)}(x_i)}$$

Then $x_i \in [t_i^{(0)}(x_i), t_i^{(1)}(x_i))$ and $u_i^{(0)}(x_i) + u_i^{(1)}(x_i) = 1$ apply.

Training for nonlinear regression produces a function with the following form:

$$f(x_1,...,x_n) \quad := \sum_{k_1=0}^{1} ... \sum_{k_p=0}^{1} a(t_1^{(k_1)}(x_1),...t_p^{(k_p)}(x_p), x_{p+1},...,x_n) \cdot \prod_{i=1}^{p} u_i^{(k_i)}(x_i)$$

This means that, for each discrete value set $x_{p+1},...,x_n$, a spline function of order 1 is defined in the continuous variables $x_1,...,x_p$. The following applies on the interval borders:

$$f(t_{1l_1},...,t_{pl_p}, x_{p+1},...,x_n) = a(t_{1l_1},...,t_{pl_p}, x_{p+1},...,x_n)$$

Furthermore:

$$\sum_{k_1=0}^{1} \cdots \sum_{k_p=0}^{1} \prod_{i=1}^{p} u_i^{(k_i)}(x_i) = 1$$

The function is therefore defined by its values at the points, and, between the points, the function is a weighted sum of the function values for the surrounding points.

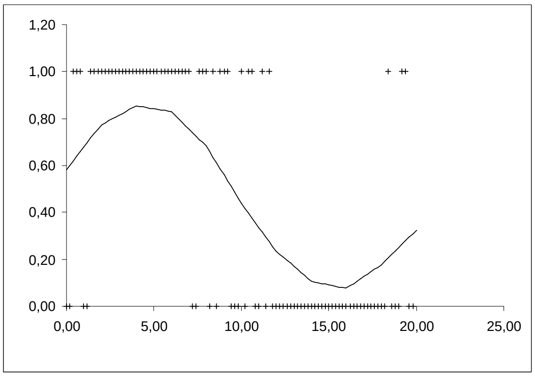
For example, the following function is obtained in the case of two continuous variables x_1 and x_2 and a discrete variable x_3 for values from x_1 in the third interval and from x_2 in the fifth interval:

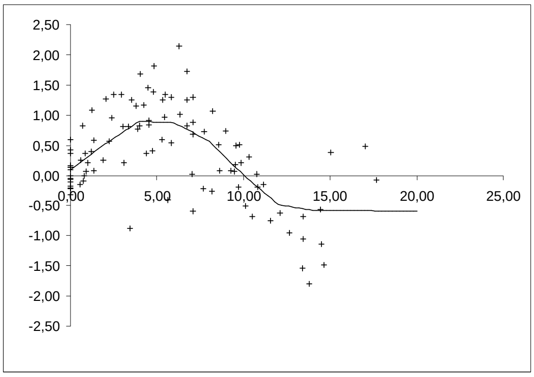
$$\begin{aligned} f(x_1, x_2, x_3) := a(t_{1,3}, t_{2,5}, x_3) \cdot \frac{t_{1,4} - x_1}{t_{1,4} - t_{1,3}} \cdot \frac{t_{2,6} - x_2}{t_{2,6} - t_{2,5}} \\ + a(t_{1,3}, t_{2,6}, x_3) \cdot \frac{t_{1,4} - x_1}{t_{1,4} - t_{1,3}} \cdot \frac{x_2 - t_{2,5}}{t_{2,6} - t_{2,5}} \\ + a(t_{1,4}, t_{2,5}, x_3) \cdot \frac{x_1 - t_{1,3}}{t_{1,4} - t_{1,3}} \cdot \frac{t_{2,6} - x_2}{t_{2,6} - t_{2,5}} \\ + a(t_{1,4}, t_{2,6}, x_3) \cdot \frac{x_1 - t_{1,3}}{t_{1,4} - t_{1,3}} \cdot \frac{x_2 - t_{2,5}}{t_{2,6} - t_{2,5}} \end{aligned}$$

The function value at a point is achieved by applying linear regression to the training data in an environment of the point. The smallest, multidimensional cuboid defined by the intervals that surrounds the point and contains a number of training records reaching or exceeding the smoothing factor set by the user is taken as the environment.

As in the previous example, you still want to estimate the sales revenue to be made with prospects on the basis of their income, but this time you assume that there is no linear dependency between sales revenue and income. You can then perform a nonlinear regression with splines of order 1. To train the nonlinear regression function, you need data in which the sales revenue is already known. After training the function on the basis of this data, you can then apply this f unction to customer data in which there is no sales revenue information but for which the potential sales revenue can be calculated on the basis of the customers' income.

When the function uses a dependency on just one key figure, it can be displayed graph ically as a polygon line that approximates the scattered dots representing the historic data.





When the function uses a dependency on several key figures, it acquires a more complex structure.

Recommendations for Modeling

Linear and Nonlinear Regression

- Start with small models containing few model fields. The use and applicability of a model does not increase with the size of the model.
- The prediction field must be continuous, and there must be at least one other continuous model field. With nonlinear regression, it can also be useful to use prediction fields with just two individual numeric values, such as 0 and 1.

  Only attributes with numeric content can be defined as continuous model fields. Generally, this is only useful if there is a continuous dependency between the attributes and the target figure. You should define the model field *Income* as continuous but a numeric region code as discrete. You could use the model field *Age* as continuous or discrete. In such cases, it is mostly recommended to opt for continuous.
- Try to use only a small number of discrete model fields, preferably just 0, 1, 2, or 3. The reason for this is that, for each combination of values for the discrete fields, a separate regression function is determined in the continuous fields, and these regression functions are independent of the regression functions of the other discrete value combinations. Having a great many combinations can lead to the entire function producing overdetermined results due to there not being enough data.

  Example: If the model has two discrete fields *Gender* and *Region* and these fields take the values *Male*, *Female* and *North*, *South*, *East*, *West* respectively, then there is a maximum of 8 combinations and consequently also a maximum of 8 independent regression functions. If the model has two discrete fields *Occupation* and *Make of Car* that each takes 10 to 20 possible values, then that already makes a possible total of 200 different combinations.
- For the same reason, generally only use discrete model fields with few values (≤10). If you use an attribute with many values, you have the option of only considering the most frequently occurring values individually and summarizing the rest of the values in the remainder category. However, this only makes sense if the distribution of values is concentrated primarily on few values. To assess this, you can use the distribution of values that you find in the settings for the model field parameters. If you have an attribute with a large number of relatively uniformly distributed values, then you should where necessary group the values during a pre-processing stage into a new attribute with just a few values or refrain from using this particular attribute.
- Use only those discrete model fields where you expect the different values to cause the prediction data to behave differently. Gender, for example, frequently lends itself well to this end. House number, on the other hand, would not be suitable.
- After training, check the results with the mean value of the absolute differences between the observed values and the predicted values of the training data. The smaller the mean value, the better the approximation. Compare the mean value with the mean value of the observed values. If, however, you use too many discrete model fields with too few combinations of values, then the model is overdetermined. In such instances, the model is inappropriate for the prediction, even if the approximation of the training data might be good.

Linear Regression

The following points are applicable only for Linear Regression:

- Linear regression is useful only if there is a linear dependency between the data of the prediction field and the data of the continuous model fields. View the quality of the approximation of the training data in the visualization of the results.
- The fewer the records in the training data set, the more straightforward the model should be. It would also be desirable for the number of the training records to be at least 10 to 100 times as big

as the number of combinations of values in the discrete fields. Similarly, a larger number of continuous model fields require a larger training data set.

Example: In the case of a model without discrete model fields, at least 100 training records would be desirable. In the case of two discrete model fields for which 10 and 20 values respectively are considered, at least 2 000 to 20 000 training records would be desirable, depending also on the number of continuous fields.

Nonlinear Regression

The following points are only applicable for Nonlinear Regression:

- Nonlinear regression is performed using multidimensional splines of order 1. This type of regression is only useful if:
  - There is a constant dependency between the data of the prediction field and that of the continuous model fields and
  - If the number of intervals is sufficiently large to track the fluctuations of the prediction data.

  You can view the quality of the approximation of the training data in the visualization of the results.
- With nonlinear regression, you should generally not use more than three continuous fields besides the prediction field. If the training data set remains the same, then, the more fields the model owns, the smaller the data density in the multidimensional dataspace becomes. Applying the algorithm leads to an increasing linearization of the model (requiring far greater calculation effort and memory than a linear model).
- Limit the complexity of the nonlinear model. The complexity increases with the number of model fields, the number of values considered for the discrete fields, and the number of intervals for the continuous fields. Increased complexity leads to a sharp increase in the calculation effort and memory consumption and could potentially have the undesired consequence of linearizing or over fitting the model.

  The product of the number of discrete values considered and the number of intervals ideally should not exceed 100 000.

  Example: In the case of a model with two discrete model fields each with 5 values to be considered and three continuous model fields each with ten intervals, the product is calculated thus: 5x5x10x10x10=25 000.
- The fewer records in the training data set, the more straightforward the model should be. It is desirable for the number of training records to be at least as great as the product referred to above.

  Example: For the above model, it is desirable to have 25 000 training records, but 100 000 would be better.
- For nonlinear regression, choose appropriate interval divisions for the continuous fields. Generally, five to ten intervals are appropriate. If the distribution of values is very inhomogeneous, then it can be useful to set the interval borders manually as opposed to having equidistant intervals. To assess this, you can use the distribution of values that you find in the settings for the model field parameters. To reduce the complexity of the model, you can also set just *one* interval for individual continuous model fields that you expect to have linear rather than nonlinear behavior and to make finer interval divisions for other fields that you suspect will have a nonlinear dependency.
- To avoid over fitting, use the parameter *Smoothing Factor* for the nonlinear regression. This parameter determines the minimum number of training records that are used for a local approximation. As the value of the parameter increases, the level of smoothing is increased. However, upward and downward peaks are then smoothed out to a greater extent. As a general rule, take values between 10 and 100 as the parameter. When a very large parameter is used, the function tends towards becoming linear.