1 Theme

Regression model deployment using Tanagra.

Model deployment is one of the main objectives of the data mining process. We want to apply a model learned on a training set on unseen cases i.e. any people coming from the population. In the classification framework, the aim is to assign to the instance its class value from their description (e.g. http://data-mining-tutorials.blogspot.com/2008/11/apply-classifier-on-new-dataset.html). In the clustering framework, we try to detect the group which is as similar as possible to the instance according their characteristics (e.g. http://data-mining-tutorials.blogspot.com/2008/12/k-means-classification-of-new-instance.html).

We are concerned about the regression framework here[1]. The aim is to predict the values of the dependent variable for unseen instances (or unlabeled instances) from the observed values on the independent variables. The process is rather basic if we handle a linear regression model. We apply the computed parameters on the unseen instances. But, it becomes difficult when we want to treat more complex models such as support vector regression with nonlinear kernels, or the models elaborated from a combination of techniques (e.g. regression from the factors of a principal component analysis). In this context, it is essential that the deployment process is directly ensured by the data mining tool.

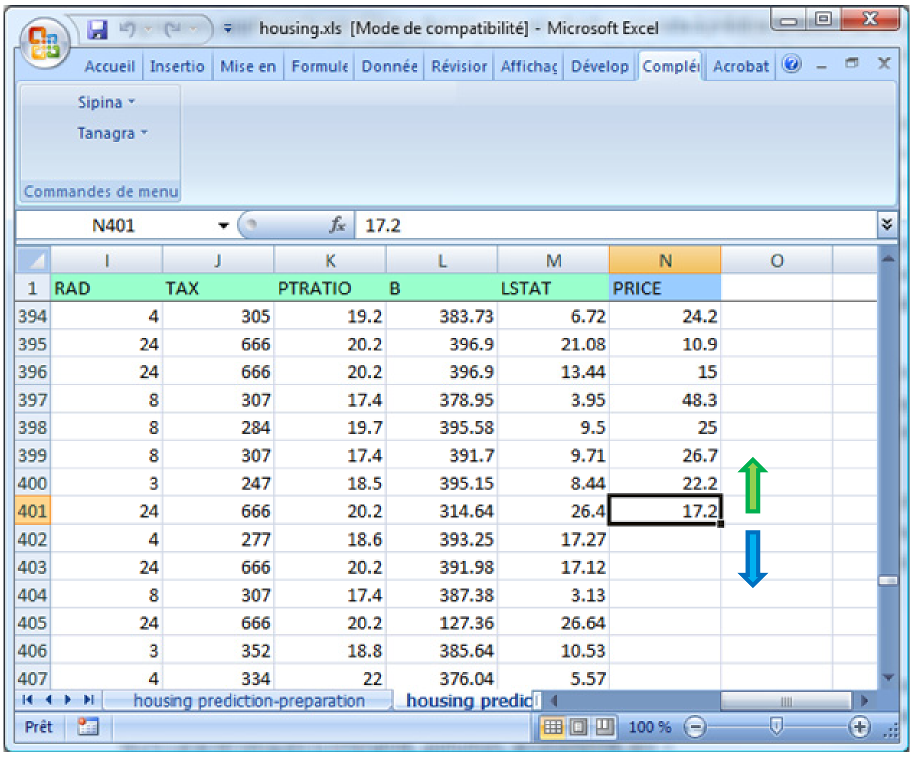
With Tanagra, it is possible to easily deploy the regression models, even when they are the result of a combination of technique. Simply, we must prepare the data file in a particular way. In this tutorial, we describe below how to organize the data file in order to deploy various models in an unified framework: a linear regression model, a PLS regression model, a support vector regression model with a RBF (radial basis function) kernel, a regression tree model , a regression model from the factors of a principal component analysis. Then, we export the results (the predicted values for the dependent variable) in a new data file. Last, we check if the predicted values are similar according to the various models.

2 Organizing the data file

We use the HOUSING dataset (http://archive.ics.uci.edu/ml/datasets/Housing). We want to predict the values of PRICE from the house characteristics (criminality, pollution, etc.). We have 400 instances (the training sample) for the construction of the models. Then, we want to apply them to 160 unseen instances i.e. the instances for which we have only the values of the independent variables.

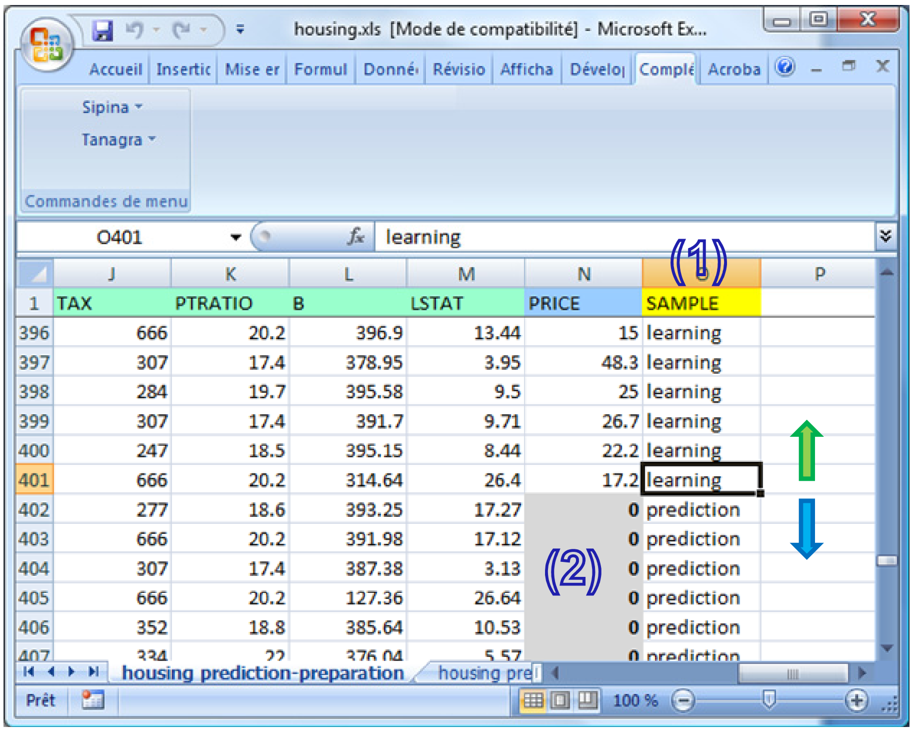
A natural way to organize the dataset is to put the training sample in the first rows of the data file. Then, we set the 106 unlabeled cases in the following rows. But, because Tanagra does not handle missing values, this data file will be truncated during the importation process. These unseen instances are not imported. Therefore, we cannot apply the models to these instances. And yet, this is our main goal.

[1] http://data-mining-tutorials.blogspot.com/search/label/Regression%20analysis


housing.xls [Mode de compatibilité] - Microsoft Excel
Sipina
Tanagra
Commandes de menu
N401
17.2
housing prediction-preparation
housing predic
Prêt
100 %

	RAD	TAX	PTRATIO	B	LSTAT	PRICE
394	4	305	19.2	383.73	6.72	24.2
395	24	666	20.2	396.9	21.08	10.9
396	24	666	20.2	396.9	13.44	15
397	8	307	17.4	378.95	3.95	48.3
398	8	284	19.7	395.58	9.5	25
399	8	307	17.4	391.7	9.71	26.7
400	3	247	18.5	395.15	8.44	22.2
401	24	666	20.2	314.64	26.4	17.2
402	4	277	18.6	393.25	17.27	
403	24	666	20.2	391.98	17.12	
404	8	307	17.4	387.38	3.13	
405	24	666	20.2	127.36	26.64	
406	3	352	18.8	385.64	10.53	
407	4	334	22	376.04	5.57	

So that Tanagra can handle the data correctly, we need to change the organization of the data file. First, we add the SAMPLE column. It specifies the status of observations: "learning" corresponds to the learning set; "prediction" corresponds to the unseen instants. Second, we set a default value for the dependent variable. The value 0 (zero) is entirely appropriate for that. The aim is to circumvent the missing value problem.


housing.xls [Mode de compatibilité] - Microsoft Ex...
Sipina
Tanagra
Commandes de menu
O401
learning
(1)
(2)
housing prediction-preparation
housing pre
Prêt
100 %

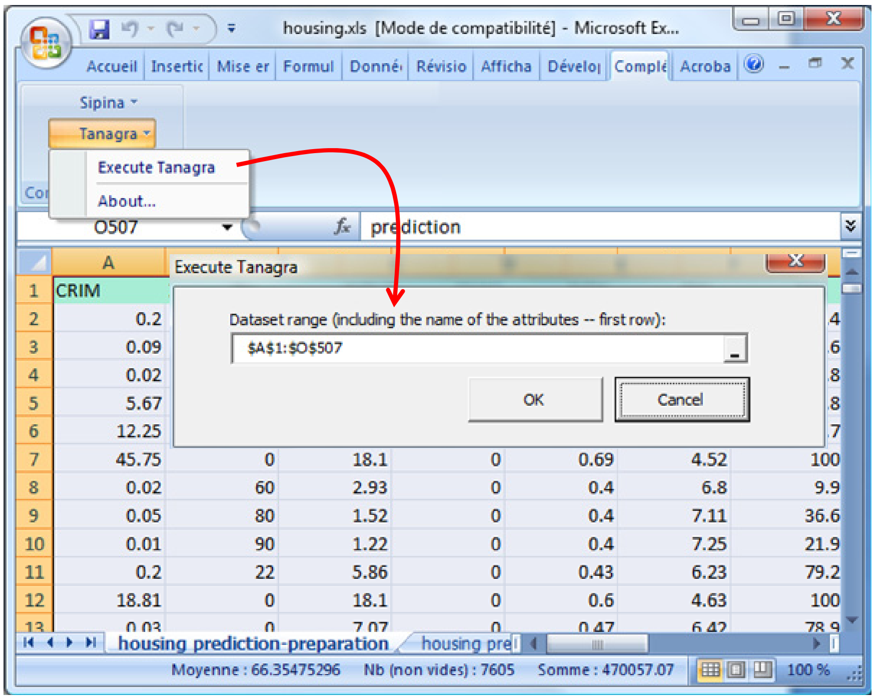
	TAX	PTRATIO	B	LSTAT	PRICE	SAMPLE
396	666	20.2	396.9	13.44	15	learning
397	307	17.4	378.95	3.95	48.3	learning
398	284	19.7	395.58	9.5	25	learning
399	307	17.4	391.7	9.71	26.7	learning
400	247	18.5	395.15	8.44	22.2	learning
401	666	20.2	314.64	26.4	17.2	learning
402	277	18.6	393.25	17.27	0	prediction
403	666	20.2	391.98	17.12	0	prediction
404	307	17.4	387.38	3.13	0	prediction
405	666	20.2	127.36	26.64	0	prediction
406	352	18.8	385.64	10.53	0	prediction
407	334	22	376.04	5.57	0	prediction

Thus, we can import the dataset into Tanagra.

3 Regression model deployment

3.1 Importing the data file

We load it into the Excel spreadsheet (we can also use OpenOffice or LibreOffice[2]). Then, using the Tanagra.xla add-in, we send the dataset to Tanagra (*see http://data-mining-tutorials.blogspot.com/2010/08/tanagra-add-in-for-office-2007-and.html for Excel 2007 and 2010; http://data-mining-tutorials.blogspot.com/2008/10/excel-file-handling-using-add-in.html for up to Excel 2003 version*).


housing.xls [Mode de compatibilité] - Microsoft Ex...
Accueil
Insertic
Mise er
Formul
Donné
Révisio
Afficha
Dévelo
Complé
Acroba
Sipina
Tanagra
Execute Tanagra
About...
O507
prediction
housing prediction-preparation
housing pre
Moyenne : 66.35475296 Nb (non vides) : 7605 Somme : 470057.07 100 %

Execute Tanagra

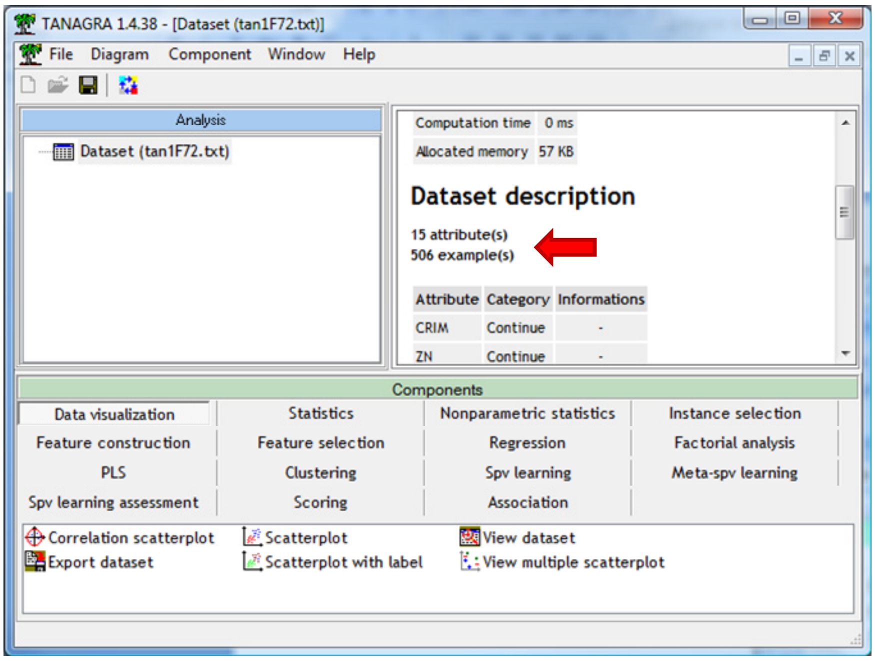
Dataset range (including the name of the attributes -- first row):

$A$1:$O$507

OK Cancel

	A						
1	CRIM						
2	0.2						4
3	0.09						6
4	0.02						8
5	5.67						8
6	12.25						7
7	45.75	0	18.1	0	0.69	4.52	100
8	0.02	60	2.93	0	0.4	6.8	9.9
9	0.05	80	1.52	0	0.4	7.11	36.6
10	0.01	90	1.22	0	0.4	7.25	21.9
11	0.2	22	5.86	0	0.43	6.23	79.2
12	18.81	0	18.1	0	0.6	4.63	100
13	0.03	0	7.07	0	0.47	6.42	78.9

All the instances (400 + 106 = 506) are now available into Tanagra.



Computation time	0 ms
Allocated memory	57 KB

Dataset description

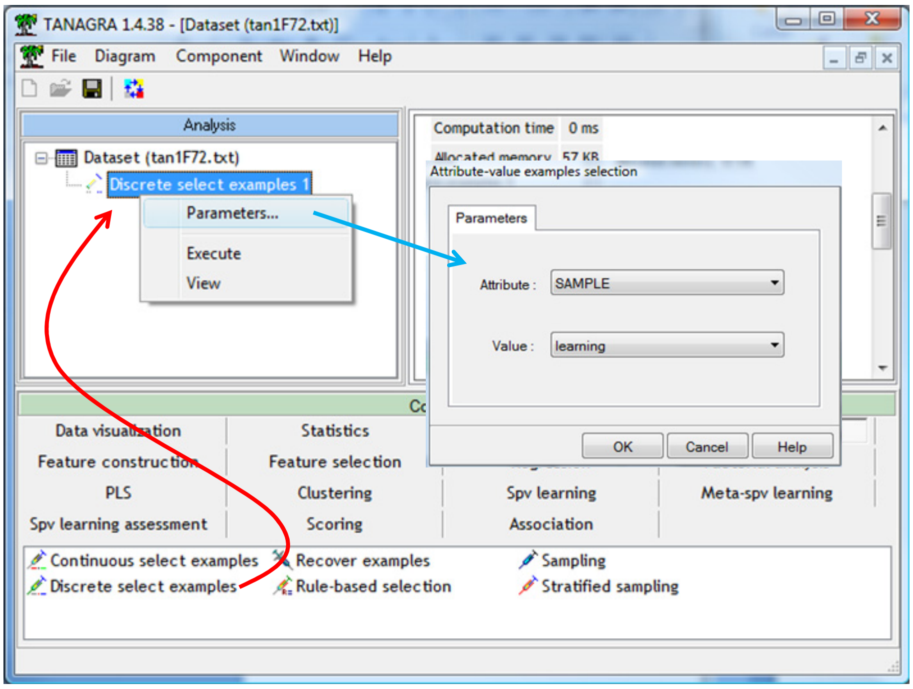
15 attribute(s)
506 example(s)

Attribute	Category	Informations
CRIM	Continue	-
ZN	Continue	-

2 http://data-mining-tutorials.blogspot.com/2011/07/tanagra-add-on-for-openoffice-calc-33.html

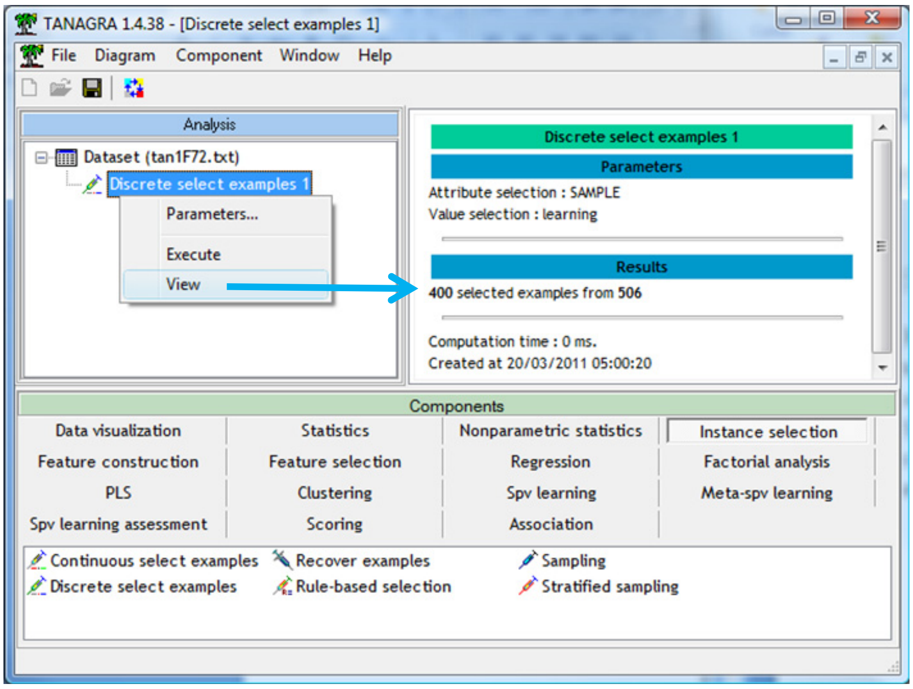
3.2 Partitioning the data file

We insert the DISCRETE SELECT EXAMPLES component (INSTANCE SELECTION tab) component into the diagram. It allows to specify the training sample i.e. the instances used for the construction of the regression model. The selection attribute is the SAMPLE column. LEARNING allows to specify the training sample.



Attribute : SAMPLE
Value : learning

Now, 400 instances are used for the learning phase.



Discrete select examples 1

Parameters

Attribute selection : SAMPLE
Value selection : learning

Results

400 selected examples from 506

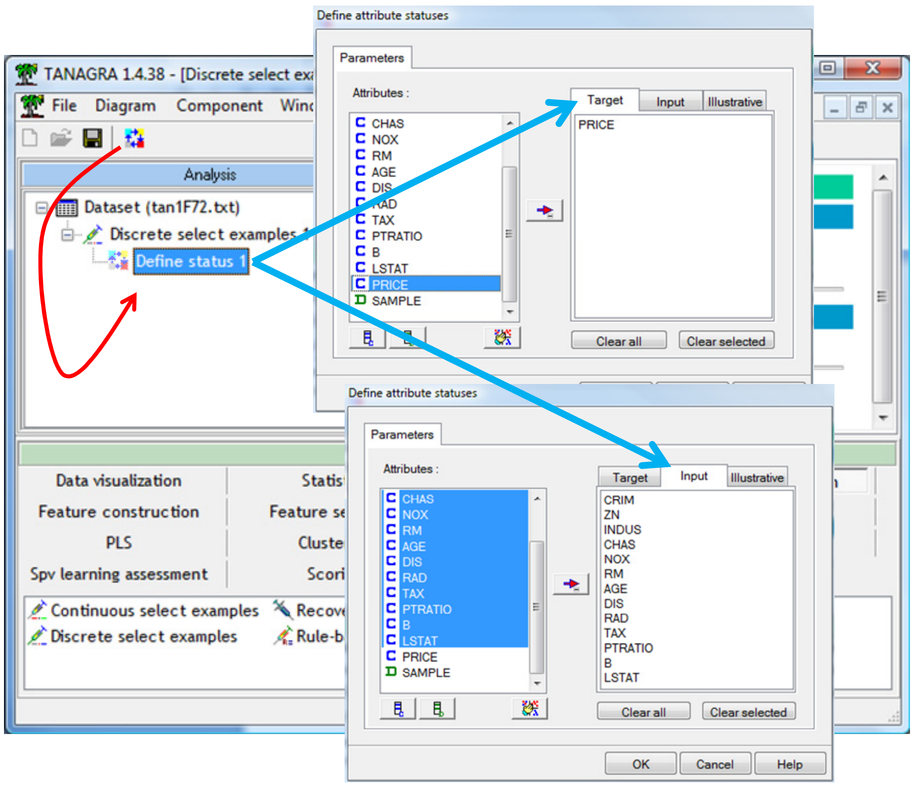
Computation time : 0 ms.
Created at 20/03/2011 05:00:20

The remaining rows are not used during learning. However, and this is the secret of the process, when the component applies the model on the data, it does on all the rows, including the unselected observations. **We take advantage to this property to get predictions on the unlabeled instances**.

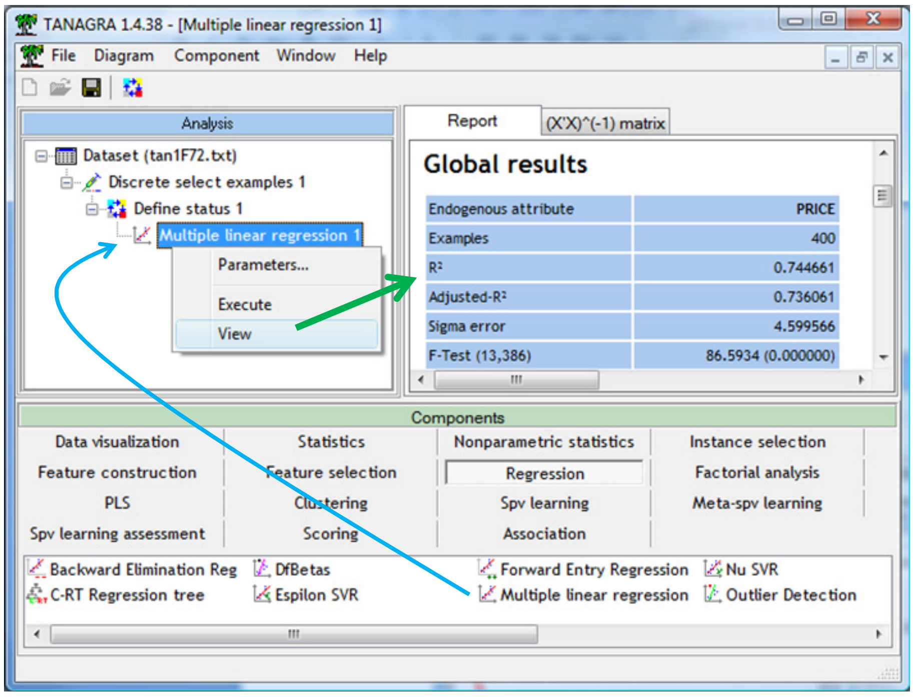
3.3 Construction of the models

3.3.1 Linear regression

We insert the DEFINE STATUS into the diagram in order to define the role of the variables into the learning process: PRICE is the TARGET attribute; the others are the INPUT ones. The SAMPLE column is no longer used at this step.



Then, we add the MULTIPLE LINEAR REGRESSION component (REGRESSION tab). We click on the VIEW menu to obtain the regression results.

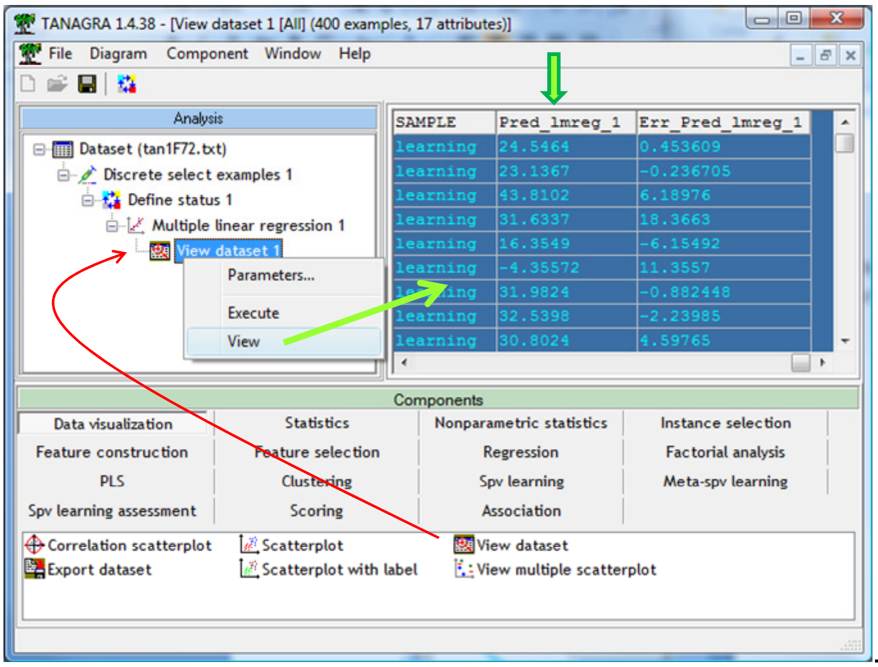


Global results

Endogenous attribute	PRICE
Examples	400
R²	0.744661
Adjusted-R²	0.736061
Sigma error	4.599566
F-Test (13,386)	86.5934 (0.000000)

The coefficient of determination is R2 = 0.744. Some independent variables are not significant at 5% significance level. We will not detail the analysis of the results in this tutorial. Our goal is to describe the deployment of learned models on unlabeled instances.

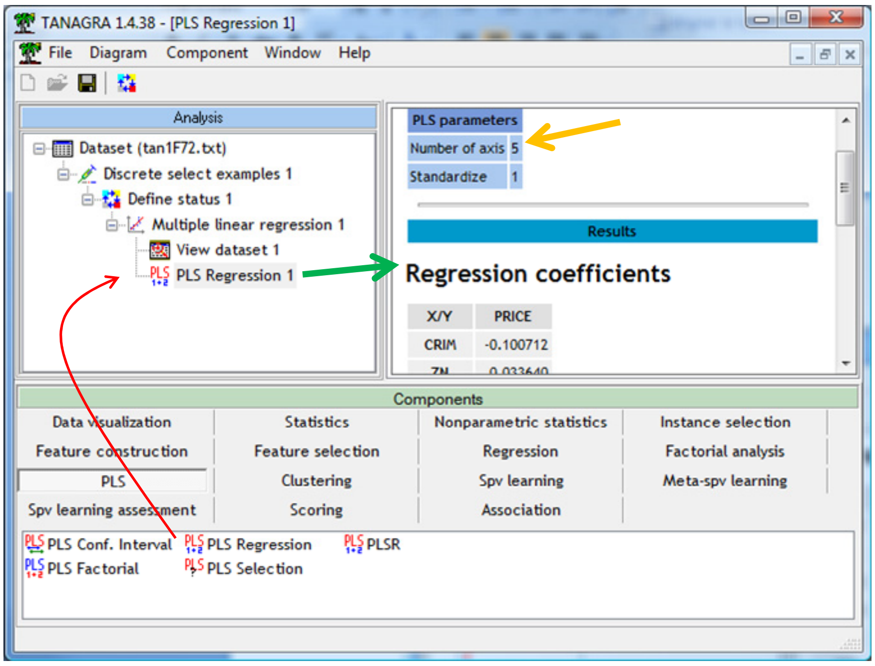
We can visualize the dataset using the VIEW DATASET (DATA VISUALIZATION tab) component. Two columns are added. PRED_LMREG_1 is the prediction of the model. Only the prediction for the learning set is displayed for the moment. But the same calculations are performed on the unselected instances. ERR_PRED_LMREG_1 corresponds to the residuals.



SAMPLE	Pred_lmreg_1	Err_Pred_lmreg_1
learning	24.5464	0.453609
learning	23.1367	-0.236705
learning	43.8102	6.18976
learning	31.6337	18.3663
learning	16.3549	-6.15492
learning	-4.35572	11.3557
learning	31.9824	-0.882448
learning	32.5398	-2.23985
learning	30.8024	4.59765

3.3.2 PLS (Partial Least Squares) Regression

We add the PLS REGRESSION component (PLS tab) behind the linear regression. We click on the VIEW menu to obtain the results according to the default settings.



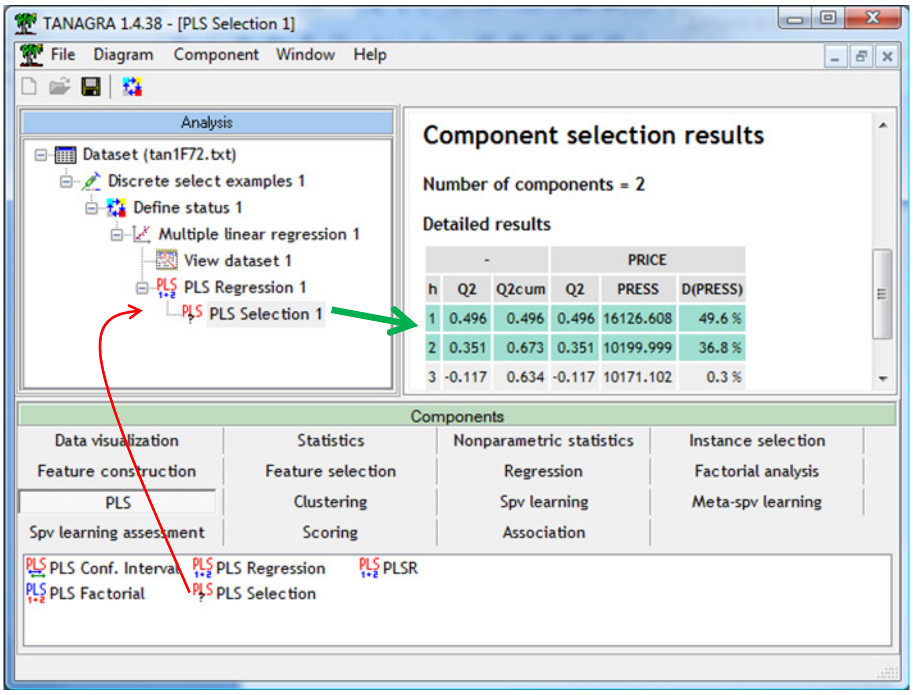
PLS parameters	
Number of axis	5
Standardize	1

Results

Regression coefficients

X/Y	PRICE
CRIM	-0.100712
ZN	0.033640

By default, 5 factors are computed. This choice is not necessarily appropriated. We can ask to Tanagra to detect the right number of factors using a cross-validation resampling approach. We use the PLS SELECTION component for that.



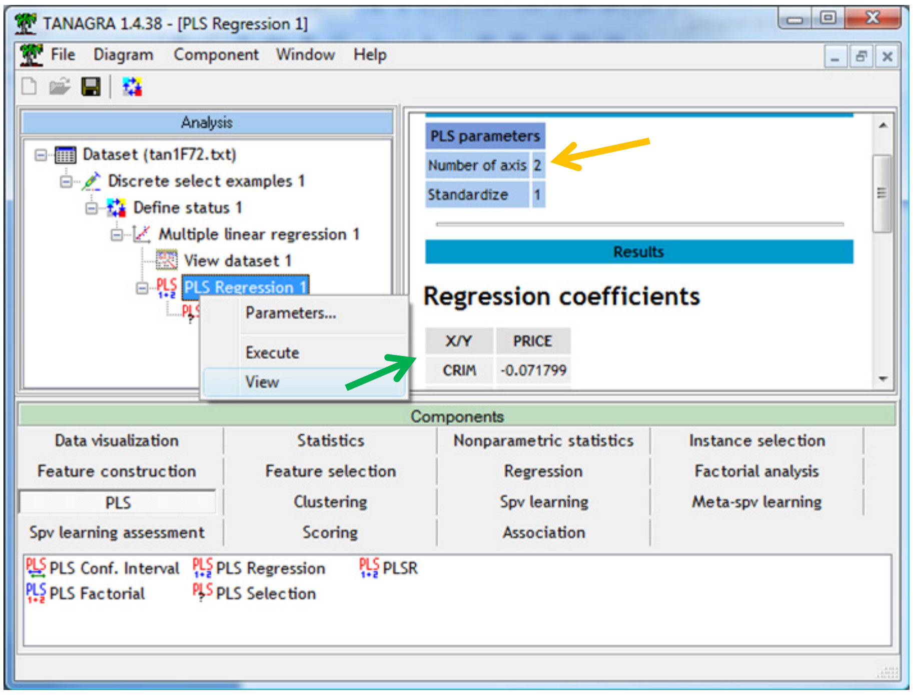
Component selection results

Number of components = 2

Detailed results

-			PRICE		
h	Q2	Q2cum	Q2	PRESS	D(PRESS)
1	0.496	0.496	0.496	16126.608	49.6 %
2	0.351	0.673	0.351	10199.999	36.8 %
3	-0.117	0.634	-0.117	10171.102	0.3 %

2 axes seem to be a good compromise. When we click on the VIEW menu of the PLS REGRESSION component, we obtain the definitive model coefficients.



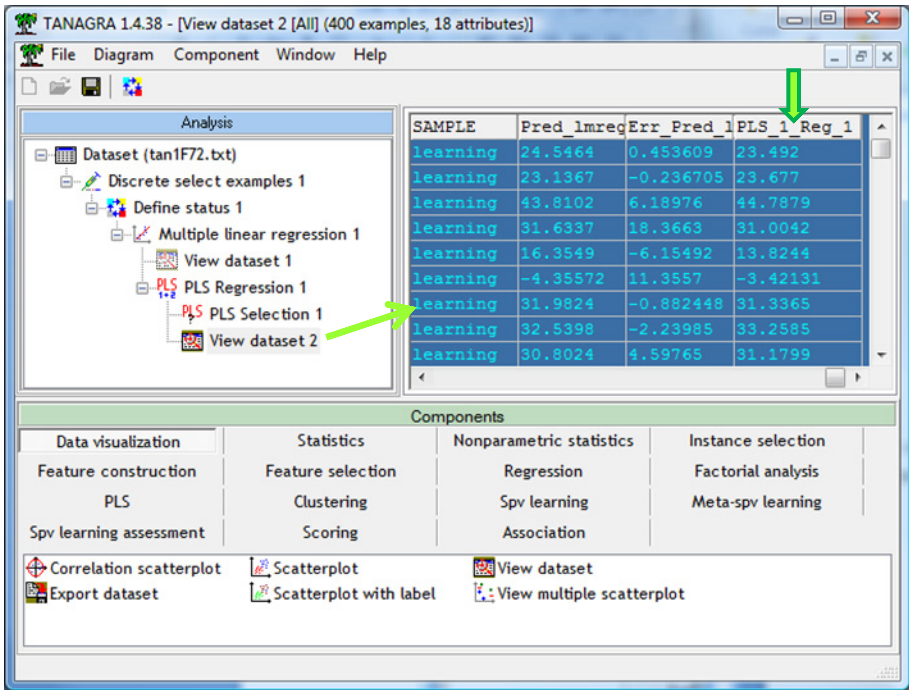
PLS parameters	
Number of axis	2
Standardize	1

Results

Regression coefficients

X/Y	PRICE
CRIM	-0.071799

Here also, when we visualize the dataset, we observe that the prediction column **PLS_1_REG_1** is added.



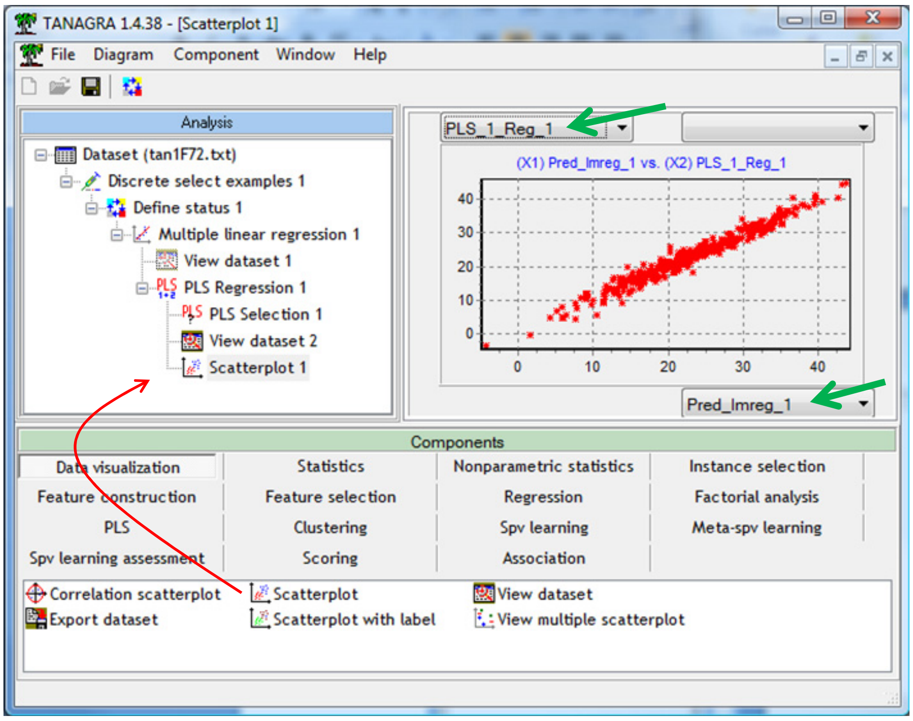
TANAGRA 1.4.38 - [View dataset 2 [All] (400 examples, 18 attributes)]

Analysis

- Dataset (tan1F72.txt)
  - Discrete select examples 1
    - Define status 1
      - Multiple linear regression 1
        - View dataset 1
        - PLS Regression 1
          - PLS Selection 1
          - View dataset 2

SAMPLE	Pred_lmreg	Err_Pred_1	PLS_1_Reg_1
learning	24.5464	0.453609	23.492
learning	23.1367	-0.236705	23.677
learning	43.8102	6.18976	44.7879
learning	31.6337	18.3663	31.0042
learning	16.3549	-6.15492	13.8244
learning	-4.35572	11.3557	-3.42131
learning	31.9824	-0.882448	31.3365
learning	32.5398	-2.23985	33.2585
learning	30.8024	4.59765	31.1799

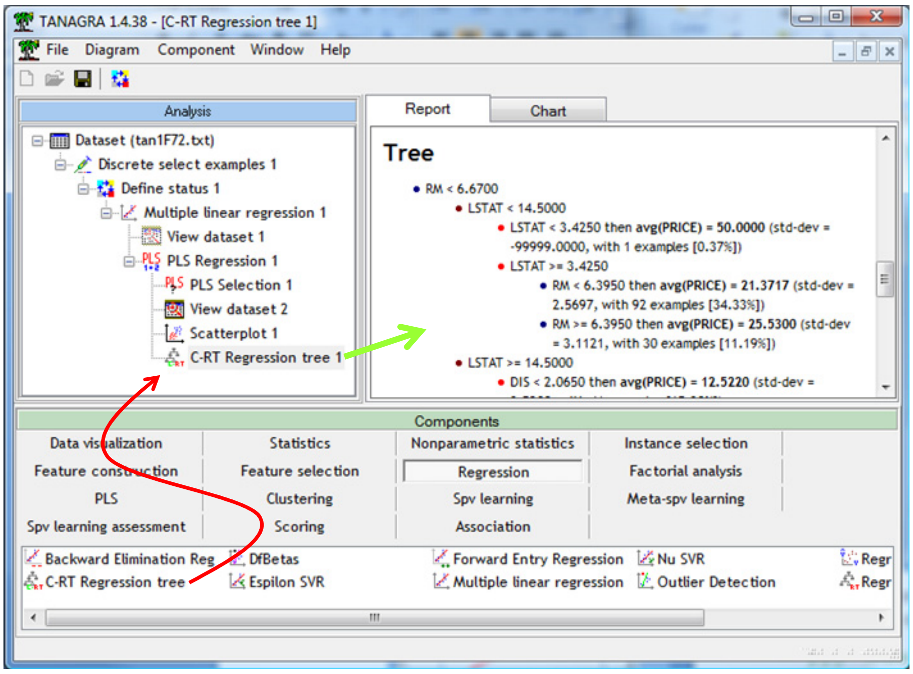
We can compare the predictions of the two models (Linear Regression and PLS Regression) on the learning set using the SCATTERPLOT (DATA VISUALIZATION tab) component.



We observe that they are consistent. This is not really surprising. These are both linear models. Their behavior is quite similar on the majority of problems. The PLS1 regression seems especially valuable when we have correlated independent variables (e.g. http://data-mining-tutorials.blogspot.com/2010/05/solutions-for-multicollinearity-in.html).

3.3.3 Regression tree

Now, we add the regression tree component (CART approach) into the diagram. We use the C-RT Regression Tree (REGRESSION tab).



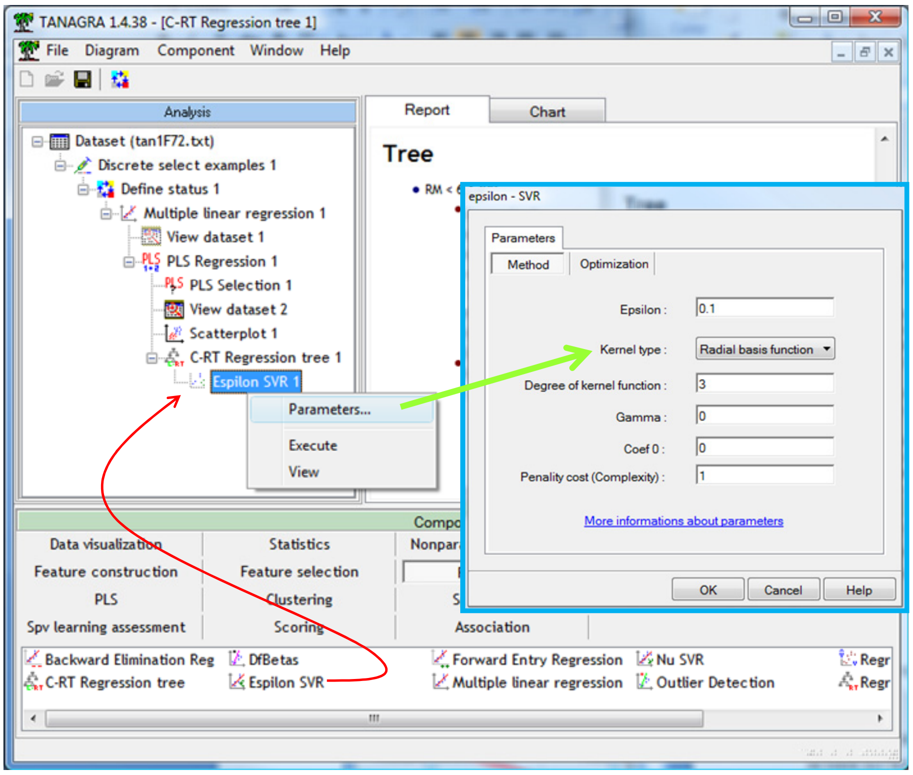
Tree

- RM < 6.6700
  - LSTAT < 14.5000
    - LSTAT < 3.4250 then avg(PRICE) = 50.0000 (std-dev = -99999.0000, with 1 examples [0.37%])
    - LSTAT >= 3.4250
      - RM < 6.3950 then avg(PRICE) = 21.3717 (std-dev = 2.5697, with 92 examples [34.33%])
      - RM >= 6.3950 then avg(PRICE) = 25.5300 (std-dev = 3.1121, with 30 examples [11.19%])
  - LSTAT >= 14.5000
    - DIS < 2.0650 then avg(PRICE) = 12.5220 (std-dev =

The tree consists of 12 leaves. Like the linear models, the deployment of a regression tree is very easy. We can transform the tree in a set of rules without loss of information.

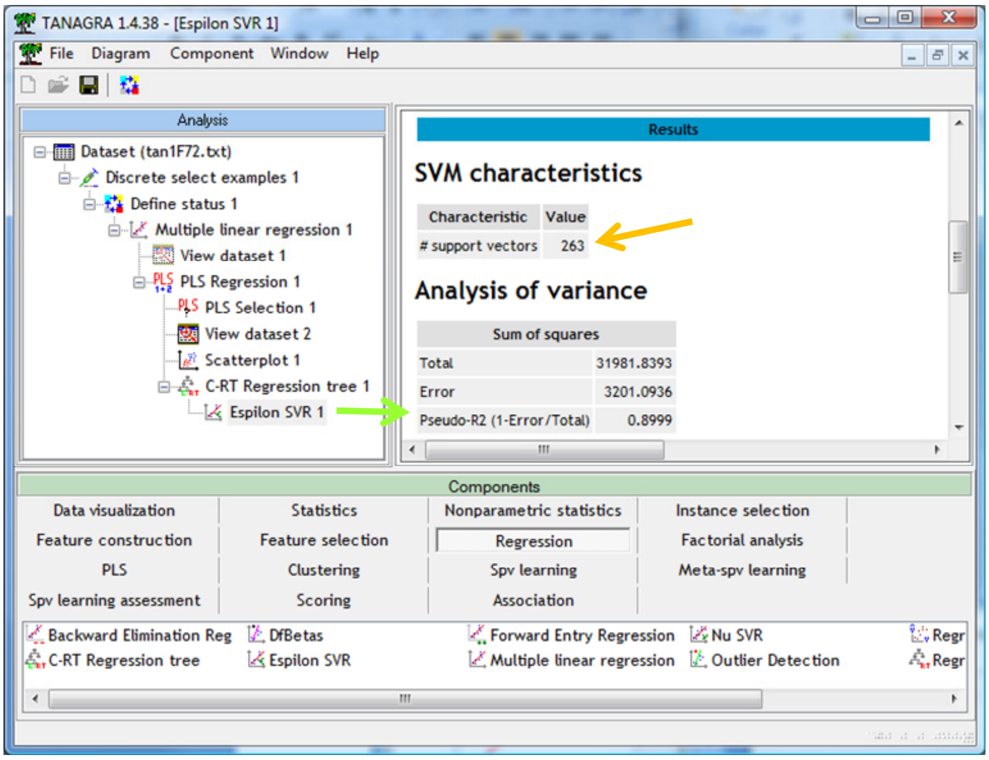
3.3.4 Support vector regression

We use the EPSILON SVR component (REGRESSION tab). We have described the behavior of this component elsewhere (this component is based on the LIBSVM library - http://data-mining-tutorials.blogspot.com/2009/04/support-vector-regression-svr.html). We set a RBF (Radial Basis Function) kernel.



Epsilon : 0.1
Kernel type : Radial basis function
Degree of kernel function : 3
Gamma : 0
Coef 0 : 0
Penality cost (Complexity) : 1

We click on the VIEW menu to obtain the results.



- Dataset (tan1F72.txt)
  - Discrete select examples 1
    - Define status 1
      - Multiple linear regression 1
        - View dataset 1
        - PLS Regression 1
          - PLS Selection 1
          - View dataset 2
          - Scatterplot 1
          - C-RT Regression tree 1
            - Espilon SVR 1

Results

SVM characteristics

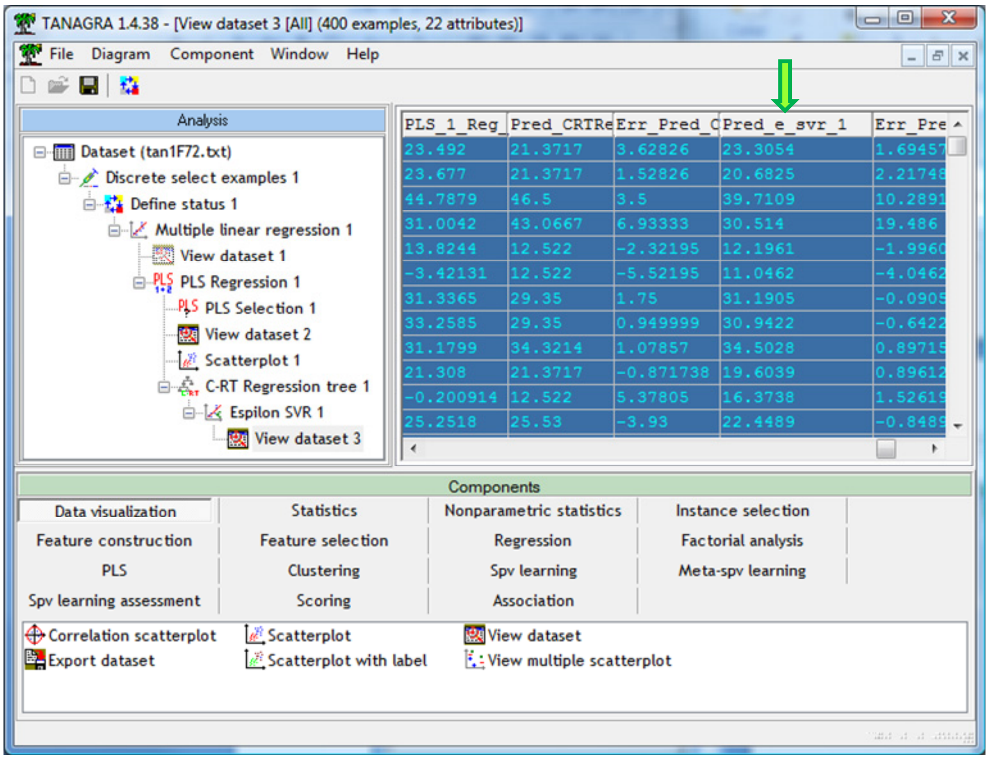
Characteristic	Value
# support vectors	263

Analysis of variance

Sum of squares	
Total	31981.8393
Error	3201.0936
Pseudo-R2 (1-Error/Total)	0.8999

Here, the difficulties really begin. Indeed, we do not have an explicit model. We need the support vectors (263 instances for our problem according the output of the tool) to make predictions on unlabeled instances. It is not really easy to make this outside the data mining tool used for the learning of the model. This is the reason for which Tanagra provides functionality for the prediction on unlabeled instances. Almost all the well known data mining tools perform this kind of operation.

With the VIEW DATASET component, we observe that EPSILON SVR adds also new columns for the predictions and the residuals.



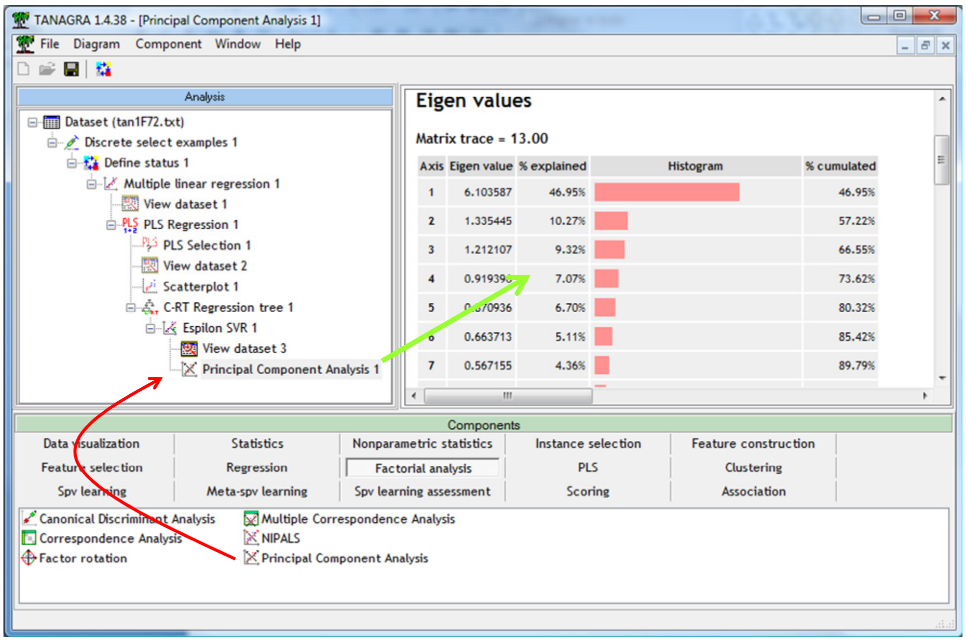
- Dataset (tan1F72.txt)
  - Discrete select examples 1
    - Define status 1
      - Multiple linear regression 1
        - View dataset 1
        - PLS Regression 1
          - PLS Selection 1
          - View dataset 2
          - Scatterplot 1
          - C-RT Regression tree 1
            - Espilon SVR 1
              - View dataset 3

PLS_1_Reg	Pred_CRTRe	Err_Pred_C	Pred_e_svr_1	Err_Pre
23.492	21.3717	3.62826	23.3054	1.69457
23.677	21.3717	1.52826	20.6825	2.21748
44.7879	46.5	3.5	39.7109	10.2891
31.0042	43.0667	6.93333	30.514	19.486
13.8244	12.522	-2.32195	12.1961	-1.9960
-3.42131	12.522	-5.52195	11.0462	-4.0462
31.3365	29.35	1.75	31.1905	-0.0905
33.2585	29.35	0.949999	30.9422	-0.6422
31.1799	34.3214	1.07857	34.5028	0.89715
21.308	21.3717	-0.871738	19.6039	0.89612
-0.200914	12.522	5.37805	16.3738	1.52619
25.2518	25.53	-3.93	22.4489	-0.8489

3.3.5 Regression on the factors of PCA

Until now, we use the learning algorithms individually. The deployment involves a single model. In this section, we want to implement a combination of methods. In a first step, we perform a principal component analysis from the independent variables. In a second step, we use the 5 first factors scores provided by the PCA as input variable in a linear regression approach. The dependent variable is still PRICE of course. For the deployment phase, we must sequentially use the two models. First, we compute the factor scores of unlabeled instances. Then, we obtain the prediction by applying the regression parameters of these factor scores.

We add the PRINCIPAL COMPONENT ANALYSIS component (FACTORIAL ANALYSIS tab) into the diagram. We obtain the following results.

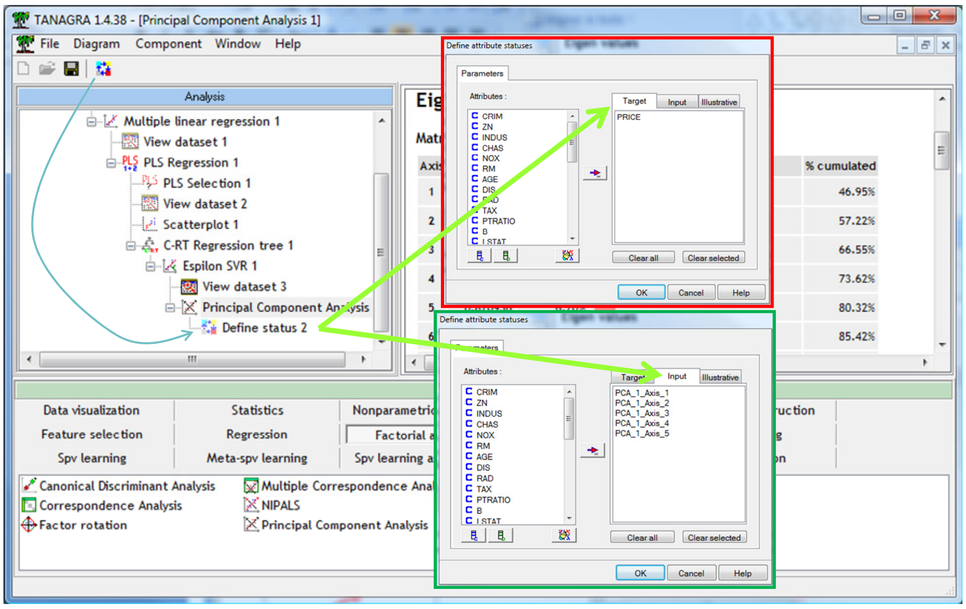


Eigen values

Matrix trace = 13.00

Axis	Eigen value	% explained	Histogram	% cumulated
1	6.103587	46.95%		46.95%
2	1.335445	10.27%		57.22%
3	1.212107	9.32%		66.55%
4	0.919390	7.07%		73.62%
5	0.870936	6.70%		80.32%
6	0.663713	5.11%		85.42%
7	0.567155	4.36%		89.79%

For the regression step, with the DEFINE STATUS component, we set the 5 first factors as INPUT variables, PRICE is the TARGET one.



Define attribute statuses

Attributes : CRIM, ZN, INDUS, CHAS, NOX, RM, AGE, DIS, RAD, TAX, PTRATIO, B, LSTAT

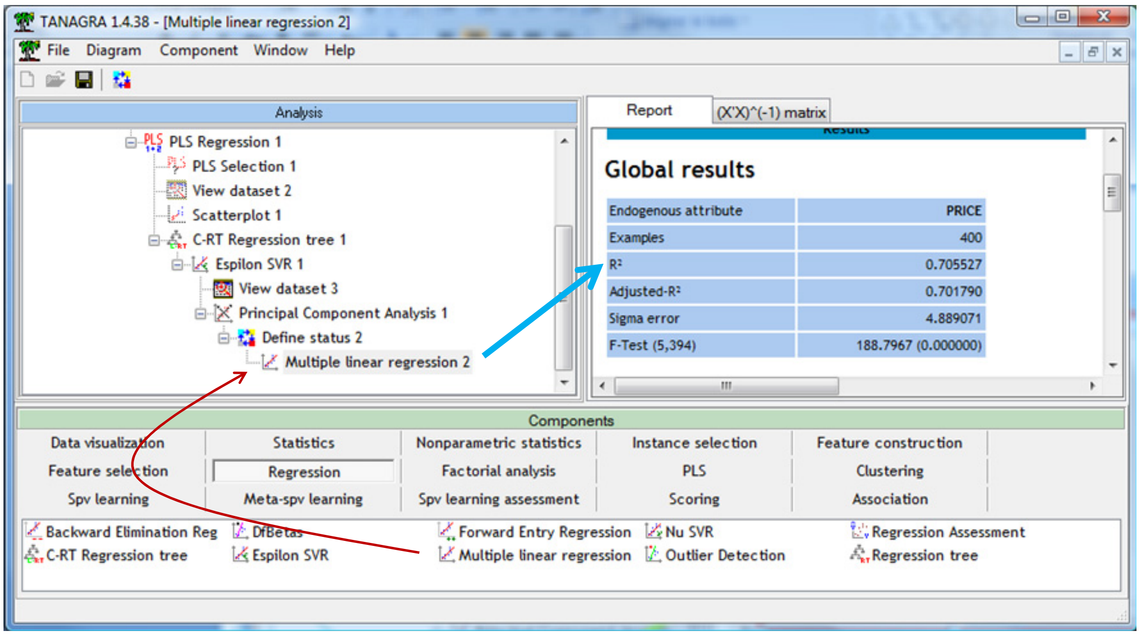
Target: PRICE

Define attribute statuses

Attributes : CRIM, ZN, INDUS, CHAS, NOX, RM, AGE, DIS, RAD, TAX, PTRATIO, B, LSTAT

Input: PCA_1_Axis_1, PCA_1_Axis_2, PCA_1_Axis_3, PCA_1_Axis_4, PCA_1_Axis_5

Then, we insert the linear regression component. We click on the VIEW menu.



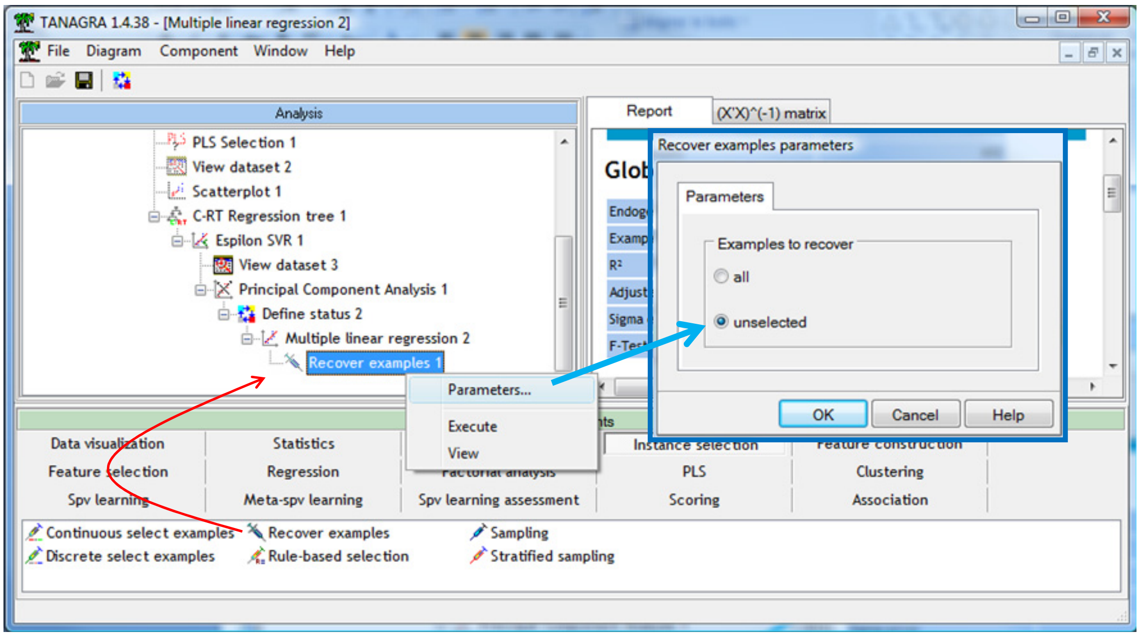
Global results

Endogenous attribute	PRICE
Examples	400
R²	0.705527
Adjusted-R²	0.701790
Sigma error	4.889071
F-Test (5,394)	188.7967 (0.000000)

3.4 Retrieving the predictions for the unlabeled instances

For each regression component, TANAGRA performs the predictions on the unselected instances i.e. on the unlabeled instances according our data file organization. To visualize them, we must reverse the instances selection.

We insert the RECOVER EXAMPLES (INSTANCES SELECTION tab) into the diagram. We set the following settings.

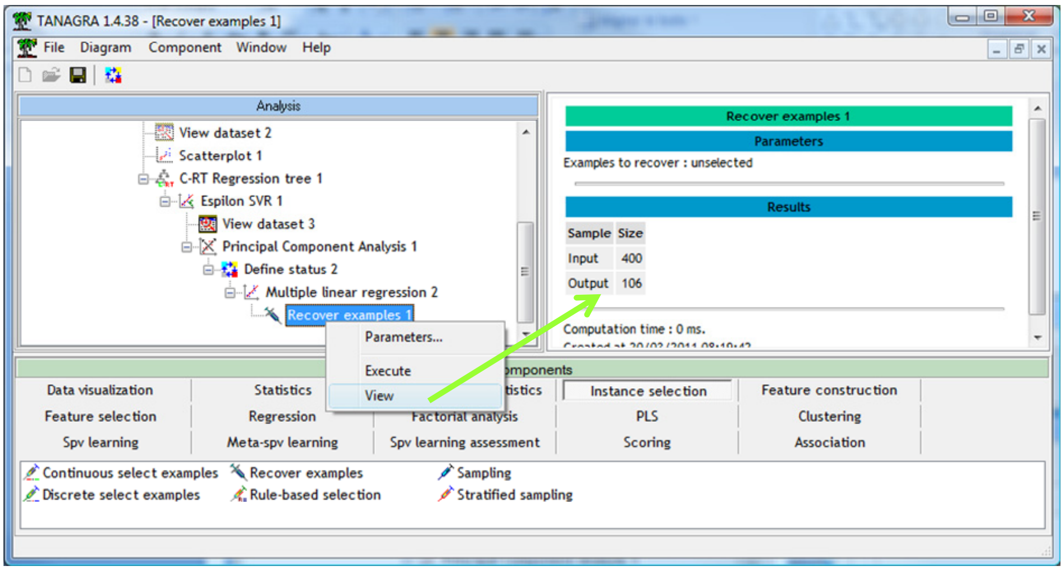


Recover examples parameters

Parameters

Examples to recover: all [✓ unselected]

We click on the VIEW menu, 106 instances are now selected i.e. the unlabeled instances.



Recover examples 1

Parameters

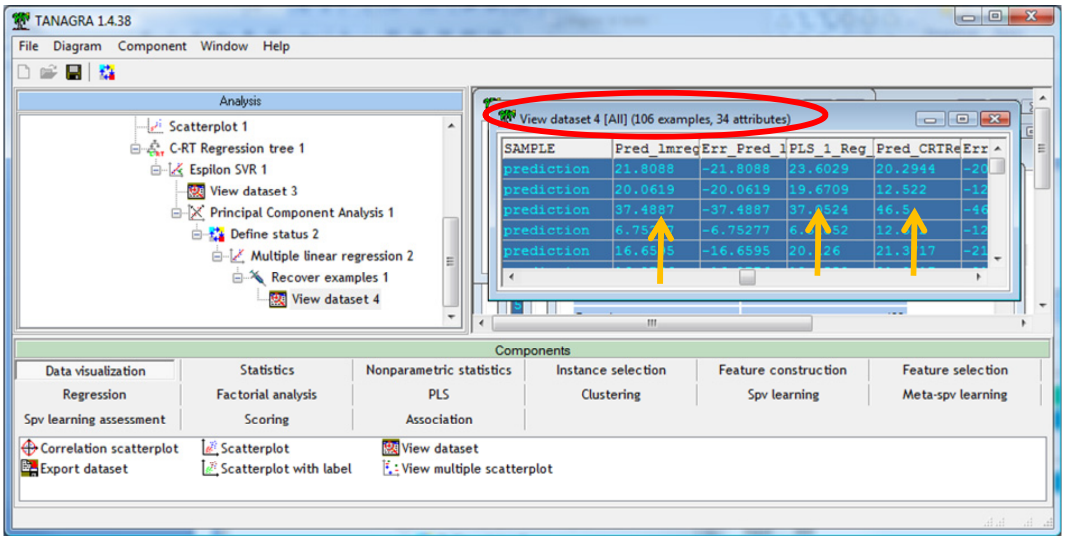
Examples to recover : unselected

Results

Sample	Size
Input	400
Output	106

Computation time : 0 ms.

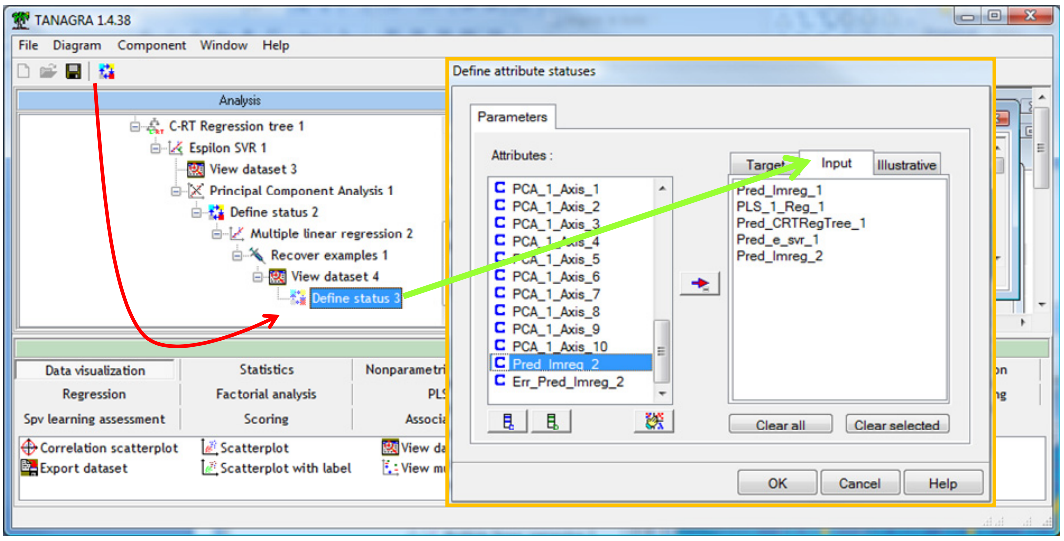
We can visualize them using the VIEW DATASET component. On the last columns, we observe the predictions provided by the various models.



View dataset 4 [All] (106 examples, 34 attributes)

SAMPLE	Pred_lmreg	Err_Pred_l	PLS_1_Reg_	Pred_CRTRe	Err
prediction	21.8088	-21.8088	23.6029	20.2944	-20
prediction	20.0619	-20.0619	19.6709	12.522	-12
prediction	37.4887	-37.4887	37.[illegible]524	46.5	-46
prediction	6.75[illegible]7	-6.75277	6.[illegible]52	12.[illegible]	-12
prediction	16.65[illegible]5	-16.6595	20.[illegible]26	21.3[illegible]17	-21

Last, we can export these values in an output file, in an Excel file format for instance. We add the DEFINE STATUS component into the diagram in order to set the predictions as INPUT.



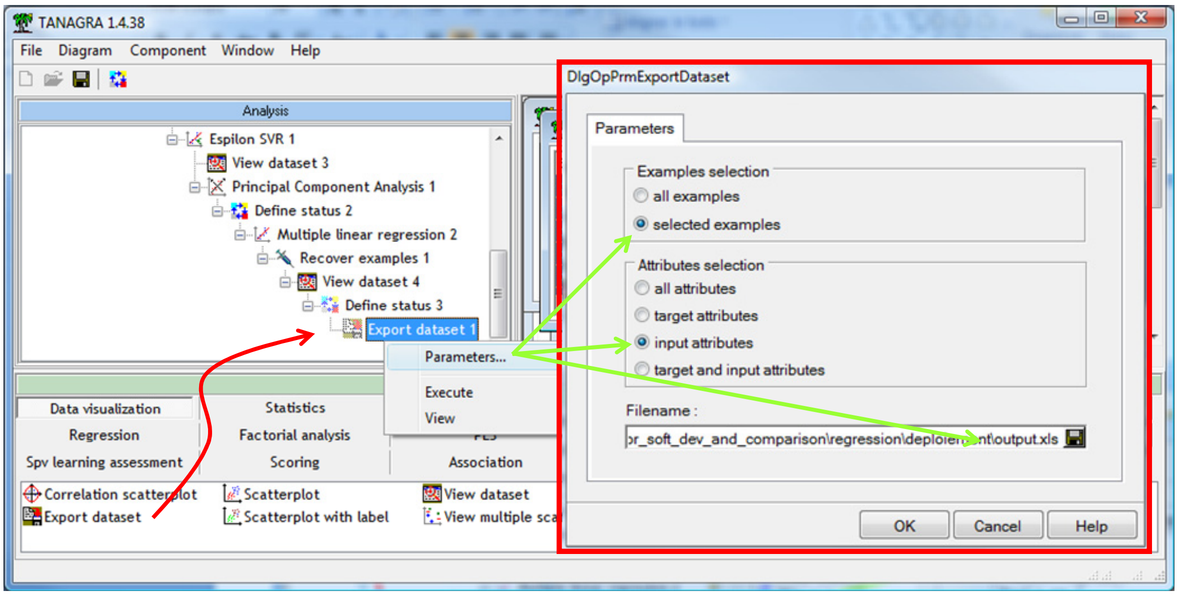
Attributes :

- PCA_1_Axis_1
- PCA_1_Axis_2
- PCA_1_Axis_3
- PCA_1_Axis_4
- PCA_1_Axis_5
- PCA_1_Axis_6
- PCA_1_Axis_7
- PCA_1_Axis_8
- PCA_1_Axis_9
- PCA_1_Axis_10
- Pred_lmreg_2
- Err_Pred_lmreg_2

Input:

- Pred_lmreg_1
- PLS_1_Reg_1
- Pred_CRTRegTree_1
- Pred_e_svr_1
- Pred_lmreg_2

We add the EXPORT DATASET. We export only the selected instances and the columns defined as INPUT above. The data file name is "**output.xls**".



DlgOpPrmExportDataset

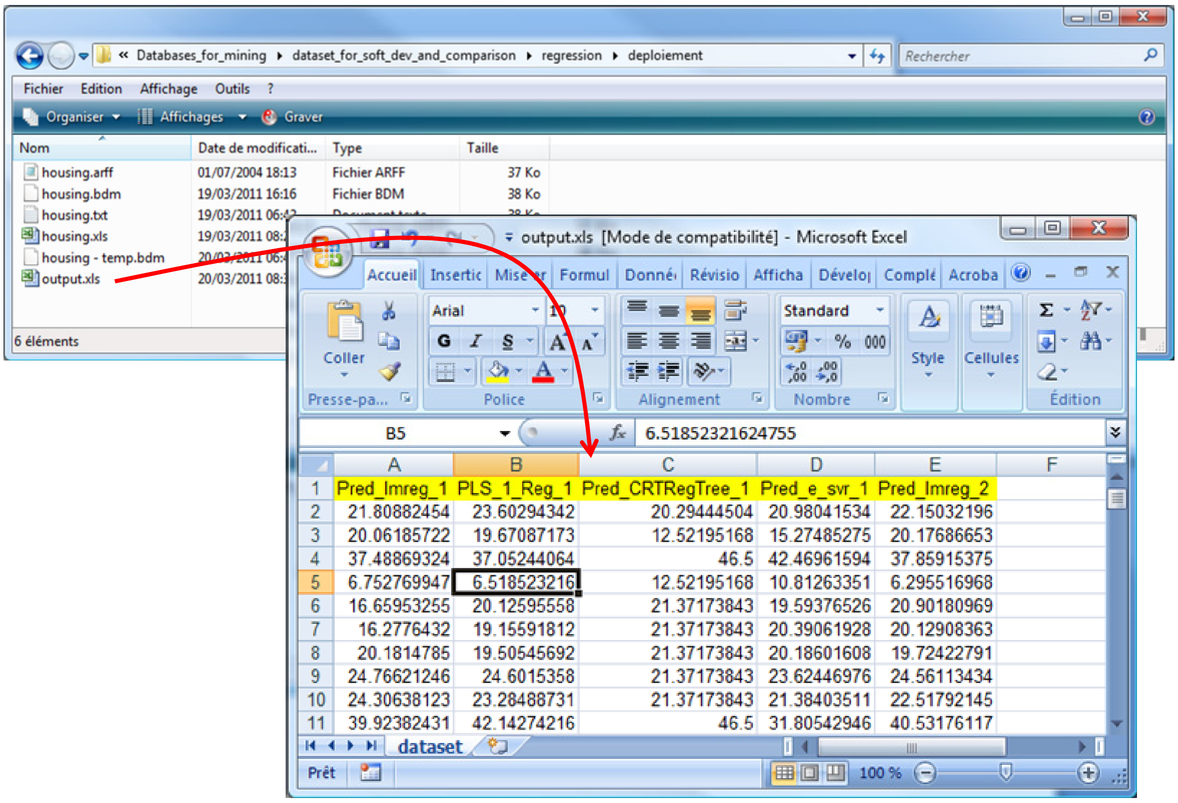
Parameters

Examples selection: all examples [✓ selected examples]

Attributes selection: all attributes / target attributes / [✓ input attributes] / target and input attributes

Filename : or_soft_dev_and_comparison\regression\deploiement\output.xls

We click on the VIEW menu. The data file is rightly generated. We can use a spreadsheet application (Excel or OpenOffice/LibreOffice) to visualize them.


« Databases_for_mining ▸ dataset_for_soft_dev_and_comparison ▸ regression ▸ deploiement
Rechercher
Fichier Edition Affichage Outils ?
Organiser Affichages Graver
6 éléments
output.xls [Mode de compatibilité] - Microsoft Excel
Accueil Insertic Mise er Formul Donné Révisio Afficha Dévelo Complé Acroba
Coller Presse-pa... Arial 10 Police Alignement Standard Nombre Style Cellules Édition
dataset
Prêt 100 %

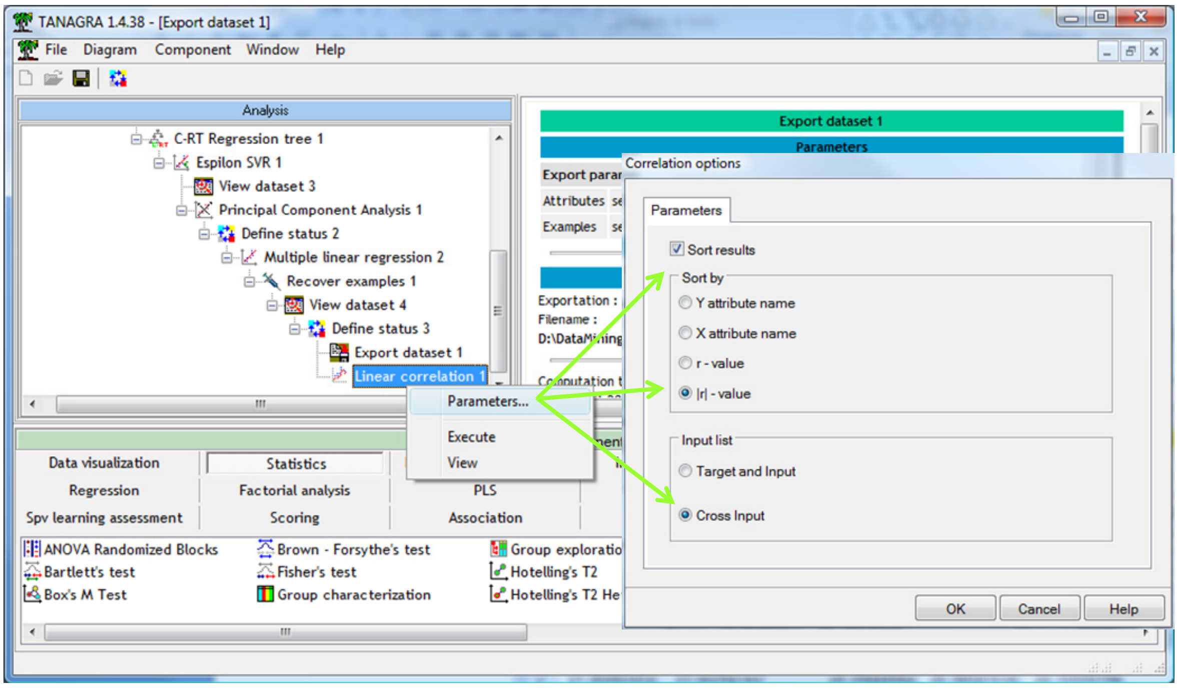
Nom	Date de modificati...	Type	Taille
housing.arff	01/07/2004 18:13	Fichier ARFF	37 Ko
housing.bdm	19/03/2011 16:16	Fichier BDM	38 Ko
housing.txt	19/03/2011 06:42	Document texte	38 Ko
housing.xls	19/03/2011 08:		
housing - temp.bdm	20/03/2011 06:		
output.xls	20/03/2011 08:		

B5 = 6.51852321624755

	A	B	C	D	E	F
1	Pred_lmreg_1	PLS_1_Reg_1	Pred_CRTRegTree_1	Pred_e_svr_1	Pred_lmreg_2	
2	21.80882454	23.60294342	20.29444504	20.98041534	22.15032196	
3	20.06185722	19.67087173	12.52195168	15.27485275	20.17686653	
4	37.48869324	37.05244064	46.5	42.46961594	37.85915375	
5	6.752769947	6.518523216	12.52195168	10.81263351	6.295516968	
6	16.65953255	20.12595558	21.37173843	19.59376526	20.90180969	
7	16.2776432	19.15591812	21.37173843	20.39061928	20.12908363	
8	20.1814785	19.50545692	21.37173843	20.18601608	19.72422791	
9	24.76621246	24.6015358	21.37173843	23.62446976	24.56113434	
10	24.30638123	23.28488731	21.37173843	21.38403511	22.51792145	
11	39.92382431	42.14274216	46.5	31.80542946	40.53176117	

3.5 Comparing the predictions

Last step of our study, we want to compare the predictions of the various models. The easiest way is to calculate the correlations between them. We insert the LINEAR CORRELATION component (STATISTICS tab) into the diagram. We set the following settings. We especially want to highlight the most highly correlated predictions.



Correlation options

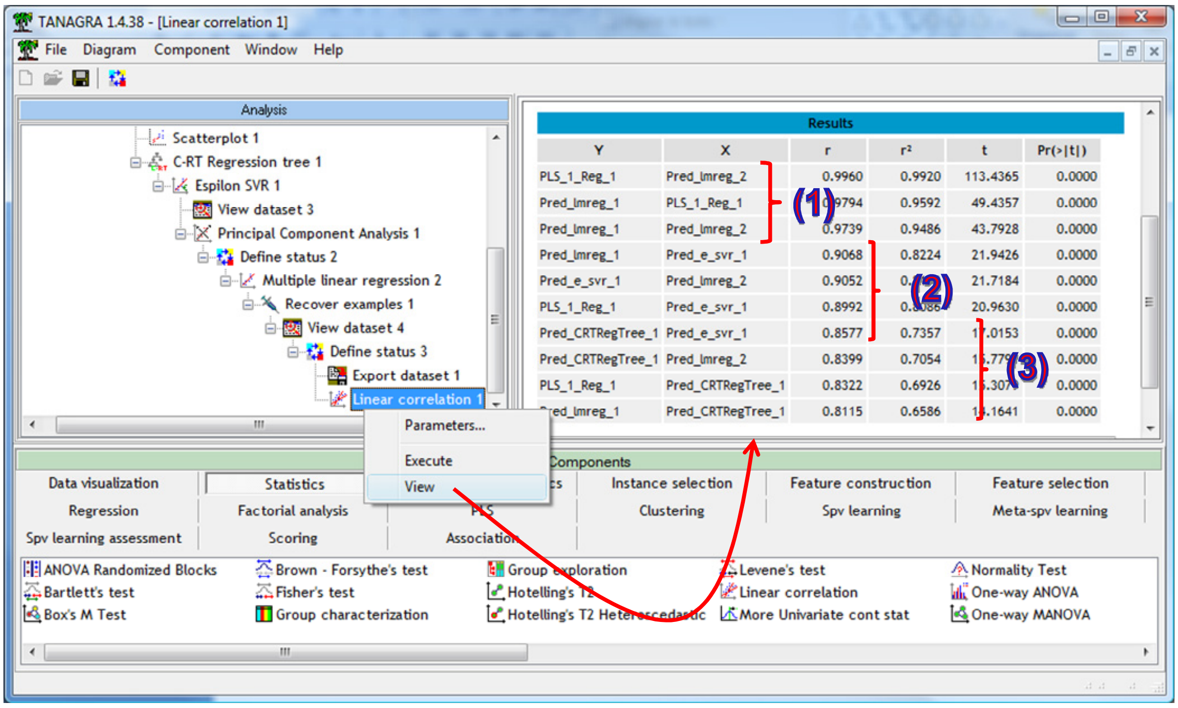
Parameters

[✓] Sort results

Sort by: ○ Y attribute name ○ X attribute name ○ r - value [✓ |r| - value]

Input list: ○ Target and Input [✓ Cross Input]

We observe that the predictions of linear models are very similar (1 - Linear regression, PLS Regression, Regression on PCA factors). The SVR with a RBF kernel, a nonlinear model, provides different predictions (2). And then, the tree which is also a nonlinear model but with different characteristics, it partitions the representation space in many regions where it makes a local prediction, it provides predictions which are not really correlated to the other ones (3).



Results

Y	X	r	r²	t	Pr(>|t|)
PLS_1_Reg_1	Pred_lmreg_2	0.9960	0.9920	113.4365	0.0000
Pred_lmreg_1	PLS_1_Reg_1	0.9794	0.9592	49.4357	0.0000
Pred_lmreg_1	Pred_lmreg_2	0.9739	0.9486	43.7928	0.0000
Pred_lmreg_1	Pred_e_svr_1	0.9068	0.8224	21.9426	0.0000
Pred_e_svr_1	Pred_lmreg_2	0.9052	[illegible]	21.7184	0.0000
PLS_1_Reg_1	Pred_e_svr_1	0.8992	0.8086	20.9630	0.0000
Pred_CRTRegTree_1	Pred_e_svr_1	0.8577	0.7357	17.0153	0.0000
Pred_CRTRegTree_1	Pred_lmreg_2	0.8399	0.7054	[illegible]	0.0000
PLS_1_Reg_1	Pred_CRTRegTree_1	0.8322	0.6926	[illegible]	0.0000
Pred_lmreg_1	Pred_CRTRegTree_1	0.8115	0.6586	14.1641	0.0000

4 Conclusion

Of course, other strategies exist for the regression model deployment. Especially into a professional context, when we wish to industrialize the model into a business decision-making process. A simple solution is to export the model in a standard format. The PMML format is a good example for that (http://www.dmg.org/v4-0-1/Regression.html for the linear regression; http://www.dmg.org/v4-0-1/TreeModel.html, for the trees; http://www.dmg.org/v4-0-1/SupportVectorMachine.html for the support vector regression, we note that the model description becomes more complex; etc.). Then we use specialized tools that can handle this kind of format for the deployment (Pentaho Data Integration for instance - http://www.pentaho.com/products/data_integration/).