Subject

Detecting potential customers is an essential task for data miners. TANAGRA now has new tools to perform this kind of task.

Dataset

We use the dataset of the CoIL Challenge 2000 (CoIL Challenge 2000 -- http://www.liacs.nl/~putten/library/cc2000/report2.html): targeting customers which will subscribe a particular insurance policy.

There were 2 datasets:

1. A learning set with 5822 examples. Target attribute is CLASS, there are 85 other descriptors, and 43 among them are socio-demographic attributes according of the zip code of the customer.
2. An unlabelled validation set of 4000 examples. We know that there are 238 positive examples in this dataset.

The challenge is to return to the organizers a file with 800 examples that contains the most positive customers.

In this tutorial, we joined together the whole dataset in one file (XLS file format); we added a descriptor (STATUS) that makes it possible to distinguish the training part of the evaluation part.

We, moreover, recovered the true labels of the individuals of the validation file, ***which was not possible during the competition***. In our case, that will enable us to simply accomplish all the evaluation process without having to handle several files.

Targeting customers with TANAGRA

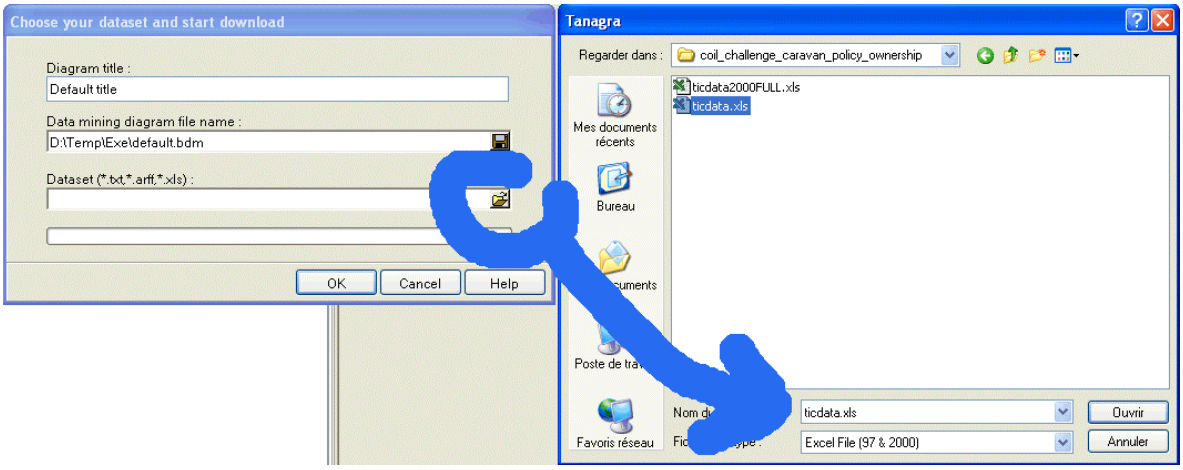
Prepare the dataset

The TICDATA.XLS contains 9822 examples: 5822 for the training set and 4000 for the validation set. The STATUS attribute enables to distinguish them. We can view the dataset in a spreadsheet.

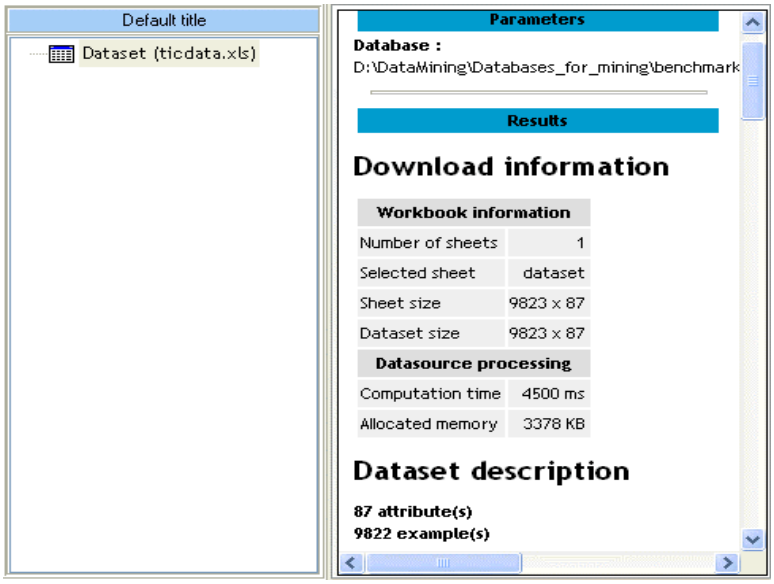
	A	CE	CF	CG	CH	CI	CJ
1	SD1	PO83	PO84	PO85	CLASS	STATUS	
5817	13	0	0	0	No	Learning	
5818	3	0	0	0	No	Learning	
5819	36	0	0	0	No	Learning	
5820	35	0	0	0	No	Learning	
5821	33	0	0	0	Yes	Learning	
5822	34	0	0	0	No	Learning	
5823	33	0	0	0	No	Learning	
5824	33	0	0	0	No	Test	
5825	6	0	0	0	Yes	Test	
5826	39	0	0	0	No	Test	
5827	9	0	0	0	No	Test	
5828	31	0	0	0	No	Test	

Download the dataset

Click on "FILE/NEW" and select the previous file.

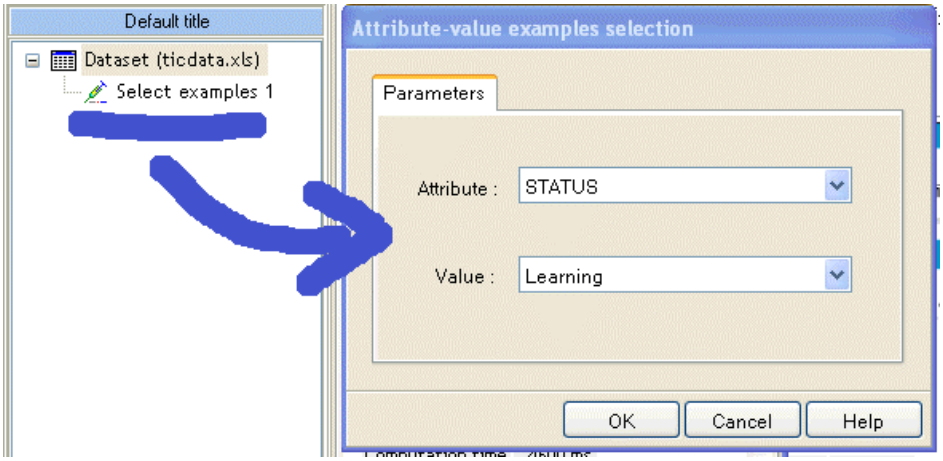


We must have 9822 examples and 87 attributes.



Select the training set

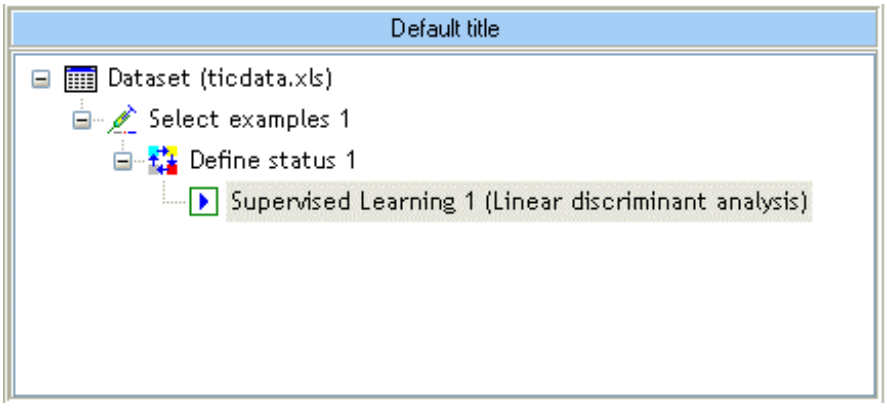
To select the training set, add the SELECT EXAMPLES component (INSTANCE SELECTION) and select the "Learning" value of the STATUS attribute.



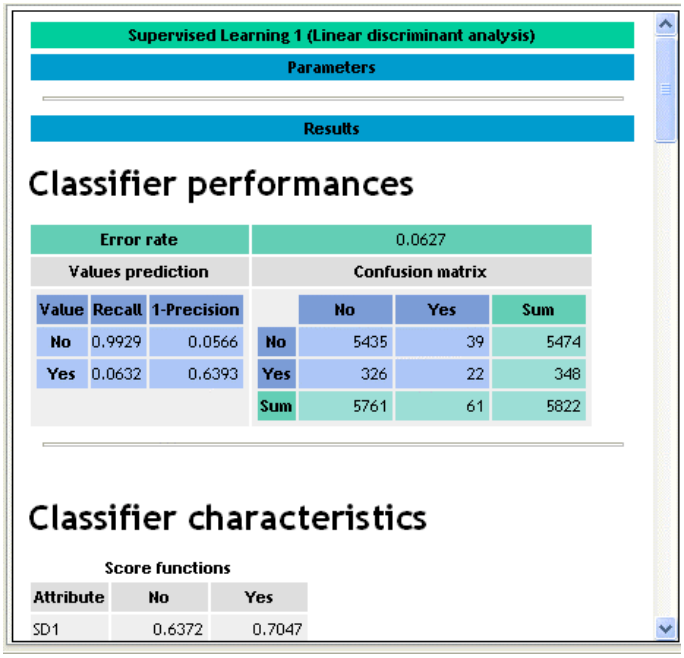
Linear Discriminant Analysis (LDA)

Set as INPUT all continuous attributes, and as TARGET the CLASS attribute. We do not use the STATUS attribute here.

Add the LDA component.



The error rate is rather disappointing (6.27%) if we compare it to the error rate of the default classifier (5.97% = 348 / 5822). This is not surprising because we have unbalanced dataset.


Supervised Learning 1 (Linear discriminant analysis)

Parameters

Results

Classifier performances

Error rate	0.0627
Values prediction	Confusion matrix

Value	Recall	1-Precision
No	0.9929	0.0566
Yes	0.0632	0.6393

	No	Yes	Sum
No	5435	39	5474
Yes	326	22	348
Sum	5761	61	5822

Classifier characteristics

Score functions

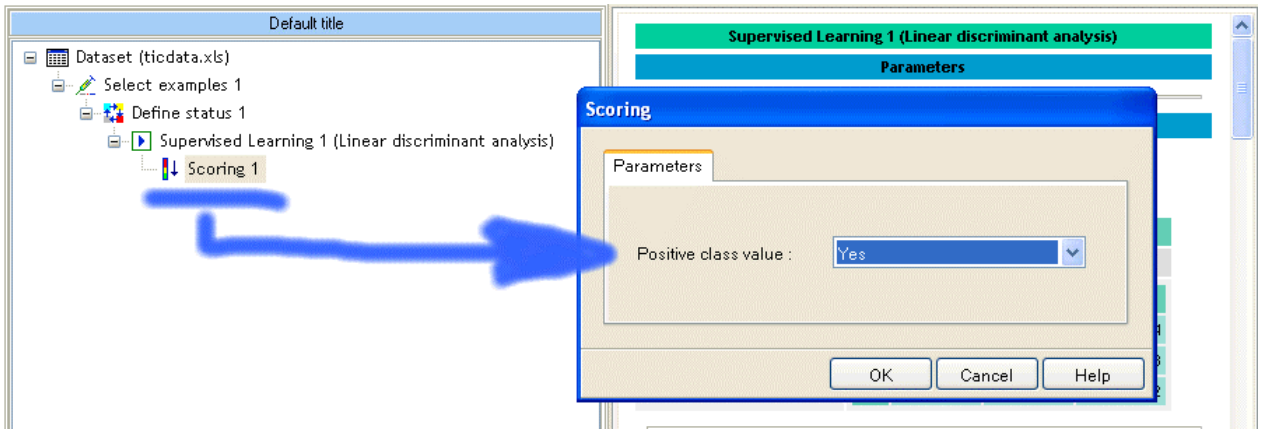
Attribute	No	Yes
SD1	0.6372	0.7047

In fact, the error rate is not the right indicator in this context. Our subject is not to globally classify the whole dataset but detect – with fixed cost, 800 examples -- the customers that subscribe the policy insurance.

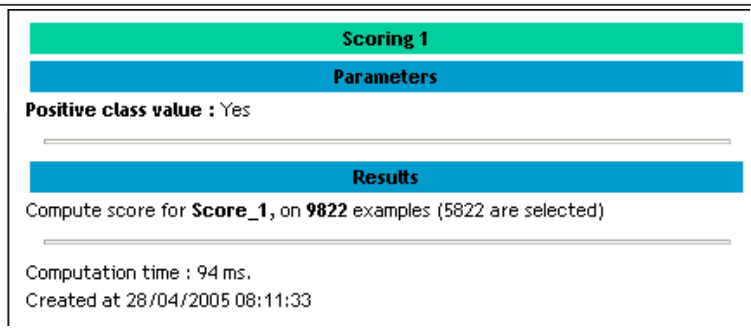
Set score to individuals

To compute the class membership probabilities for each example in the whole dataset (training and validation set), we add the SCORING component and set the "YES" value as the positive class value.

Let us note that some classifiers compute a ***score*** that is not a probability but represents the degree to which an instance is a member of the positive class value, it enables to sort the examples as well as a calibrated probability.



The score is computed on the whole dataset.

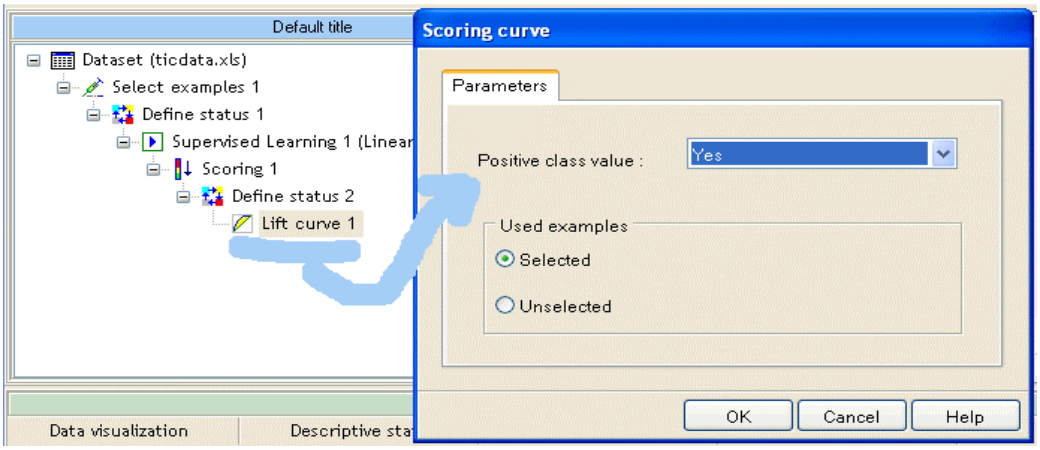


Build the LIFT CURVE

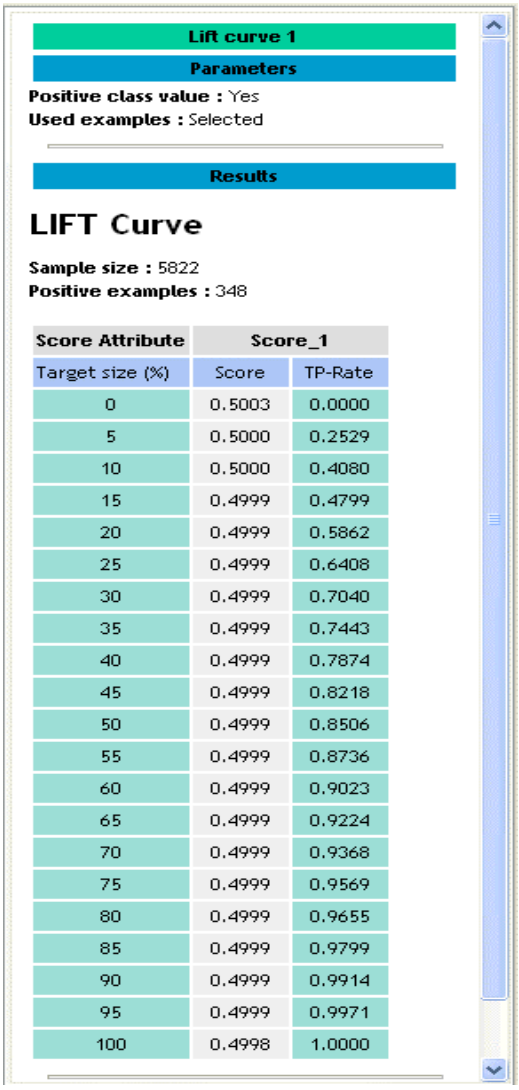
The lift curve shows the true positive rate for each targeting size.

Set as TARGET the class attribute, and set as INPUT the SCORE_1 attribute. We can select several attributes as INPUT, it is possible to compare classifiers or compare expert scoring with a classifier scoring.

Add the LIFT component, select the positive class value. Let us note that it is possible to perform a targeting process on a multiclass problem, we can select the positive class value when we want to compute the score and build the lift curve.



TANAGRA provides a table, which shows for each target size the true positive rate.



Lift curve 1

Parameters

Positive class value : Yes
Used examples : Selected

Results

LIFT Curve

Sample size : 5822
Positive examples : 348

Score Attribute	Score_1	
Target size (%)	Score	TP-Rate
0	0.5003	0.0000
5	0.5000	0.2529
10	0.5000	0.4080
15	0.4999	0.4799
20	0.4999	0.5862
25	0.4999	0.6408
30	0.4999	0.7040
35	0.4999	0.7443
40	0.4999	0.7874
45	0.4999	0.8218
50	0.4999	0.8506
55	0.4999	0.8736
60	0.4999	0.9023
65	0.4999	0.9224
70	0.4999	0.9368
75	0.4999	0.9569
80	0.4999	0.9655
85	0.4999	0.9799
90	0.4999	0.9914
95	0.4999	0.9971
100	0.4998	1.0000

We can read several results:

- The positive examples are the “YES” value of class attribute.
- We used the learning set.
- There are 5822 examples in this dataset and 348 positive examples.

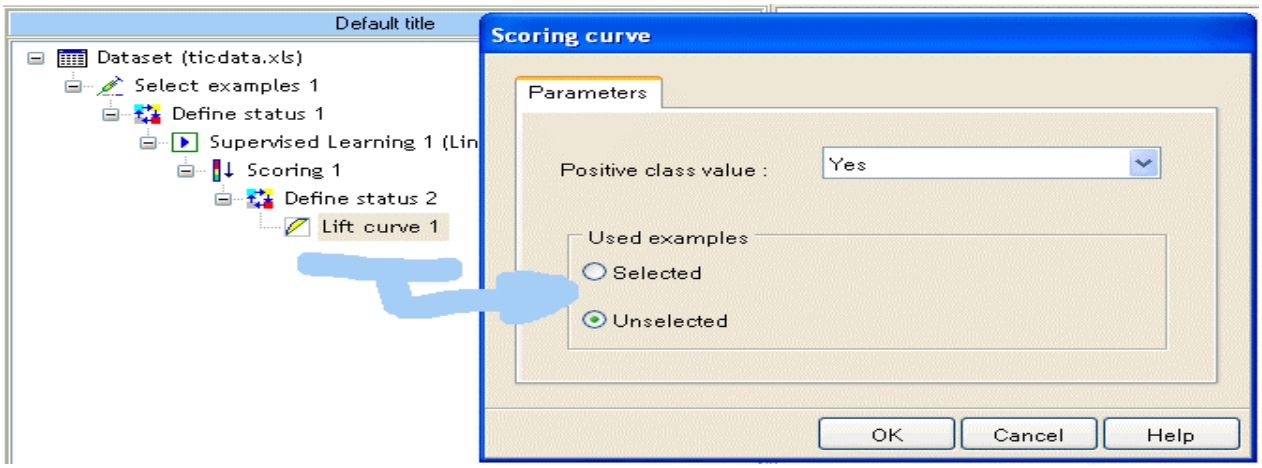
For a target size of 20% (20% of 5822 examples # 1164 examples), we can find 58.62% of positive examples, that is 58.62% of 348 positive examples # 204 positive examples.

Let us transpose this reasoning on the validation set. The target size is 800 examples (20% of 4000 = 800 examples), there are 238 positive examples in the validation set and we can find 58.62% of 238 # 139 positive examples.

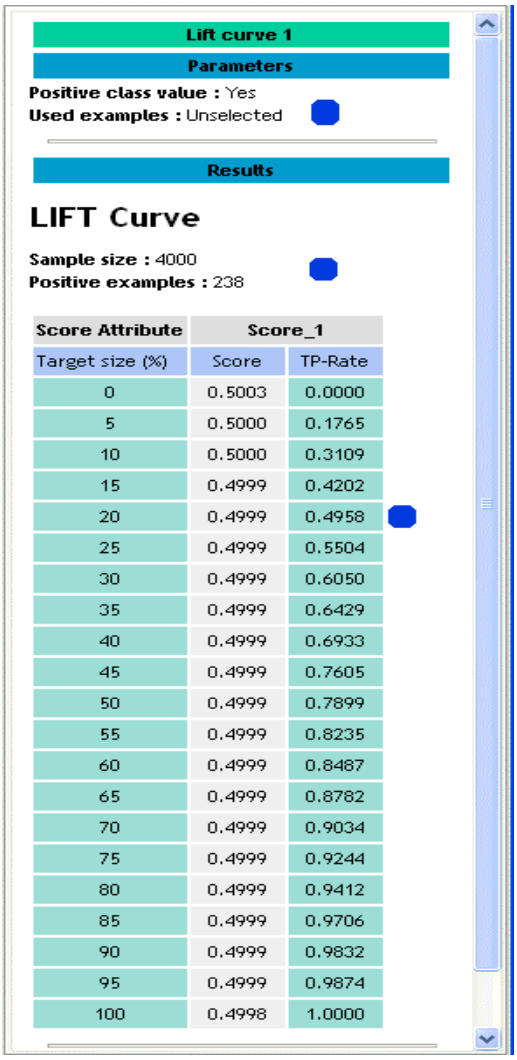
We know that it is a biased result because we use the same dataset in order to build and to evaluate the classifier. To obtain an honest estimate of the classifier performance, let us use the validation set.

Compute the lift curve on the validation set

Let us modify the LIFT component parameters for computing the curve on the validation set.



We obtain new results.



Lift curve 1

Parameters

Positive class value : Yes
Used examples : Unselected

Results

LIFT Curve

Sample size : 4000
Positive examples : 238

Score Attribute	Score_1	
Target size (%)	Score	TP-Rate
0	0.5003	0.0000
5	0.5000	0.1765
10	0.5000	0.3109
15	0.4999	0.4202
20	0.4999	0.4958
25	0.4999	0.5504
30	0.4999	0.6050
35	0.4999	0.6429
40	0.4999	0.6933
45	0.4999	0.7605
50	0.4999	0.7899
55	0.4999	0.8235
60	0.4999	0.8487
65	0.4999	0.8782
70	0.4999	0.9034
75	0.4999	0.9244
80	0.4999	0.9412
85	0.4999	0.9706
90	0.4999	0.9832
95	0.4999	0.9874
100	0.4998	1.0000

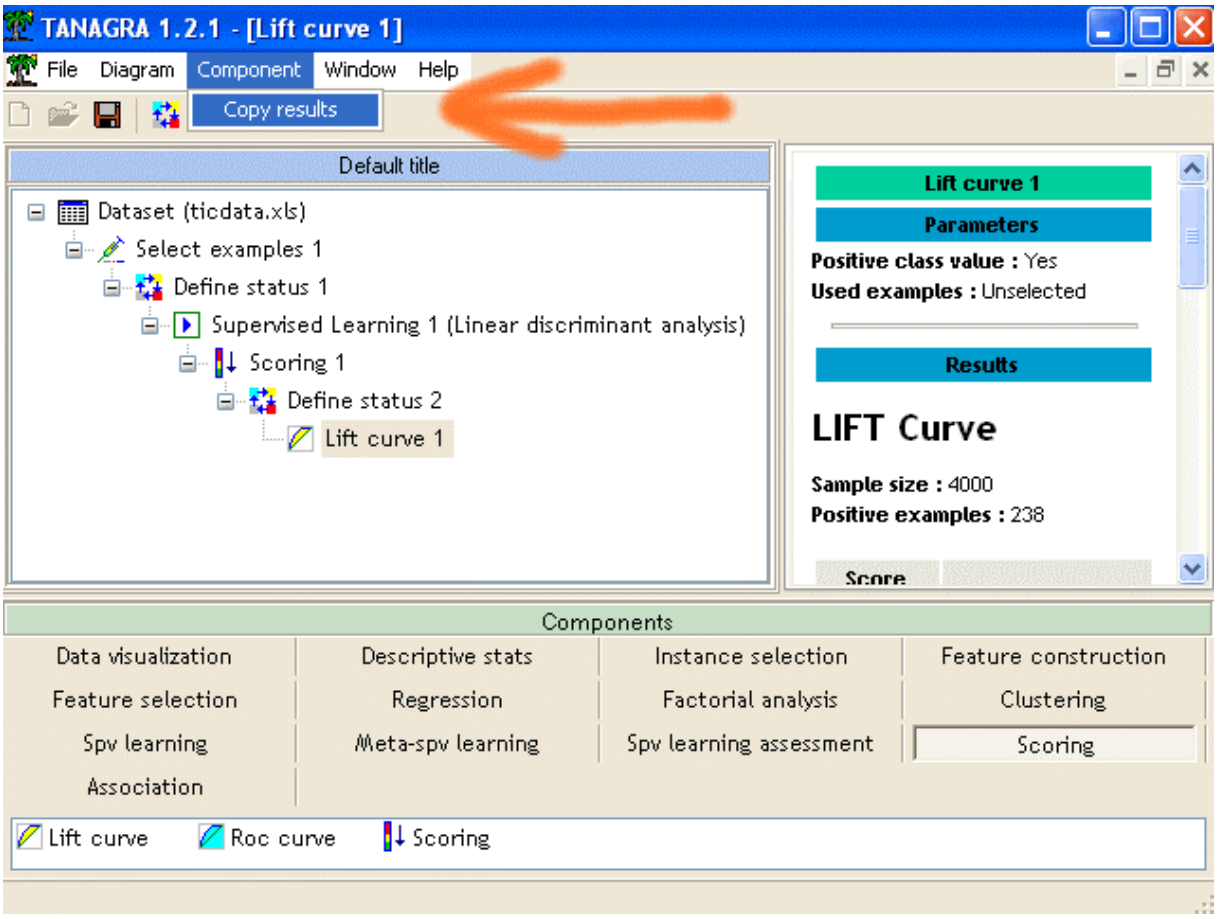
There are 4000 examples in the validation set and 238 positive examples. For a target size of 800 (20%), we find 49.58% of positive examples that is 49.58% of 238 # 118 customers.

In the proceedings of the conference, we see that linear models give the same performance. It seems that the best model is a naïve bayes classifier but the winner use some data pretreatments (feature selection and feature combination), the winner find 121 positive examples.

Lift curve in a spreadsheet

In order to build the graphical representation of the lift curve, we can copy the results in a spreadsheet.

Click on the "COMPONENT / COPY RESULTS" menu.



Building the figure is easy.

LIFT Curve

Sample size : 4000
Positive examples : 238

Score Attribute	Score_1	
Target size (%)	Score	TP-Rate
0	0.5003	0
5	0.5	0.1765
10	0.5	0.3109
15	0.4999	0.4202
20	0.4999	0.4958
25	0.4999	0.5504
30	0.4999	0.605
35	0.4999	0.6429
40	0.4999	0.6933
45	0.4999	0.7605
50	0.4999	0.7899
55	0.4999	0.8235
60	0.4999	0.8487
65	0.4999	0.8782
70	0.4999	0.9034
75	0.4999	0.9244
80	0.4999	0.9412
85	0.4999	0.9706
90	0.4999	0.9832
95	0.4999	0.9874
100	0.4998	1