R-Squared for CART regression trees

CART users often ask where they can find the value of the R-Squared for their regression trees. The answer is simple. In conventional statistics,

$$\text{R-Squared} = 1 - \text{SSE/SST}, \qquad (1)$$

where SSE is the sum of squared errors of the actual data around the model predictions, and SST, the total sum of squares, is the sum of squared deviations of the dependent variable around its mean. In traditional statistics R-Squared is always calculated using the training data (LEARN SET). CART users can read the R-Squared directly from the output:

$$\text{R-Squared} = 1 - \text{CART\_Relative\_Error} \qquad (2)$$

because

$$\text{CART\_Relative\_Error} = \text{SSE/SST}, \qquad (3)$$

where SSE is the sum of squared errors of the CART model and SST is the sum of squared errors of the dependent error around its mean in the root node. In other words, the relative error for the training data in CART is calculated exactly as 1 - R-Squared. In the CART regression tree below we display performance results for training data for the BOSTON.CSV data set. The relative error of 0.076 is literally equivalent to an R-Squared of 0.924 on the training data. *You can always find the training data performance in the classic output and it is this that you should report to readers wanting the conventional R-Squared.*

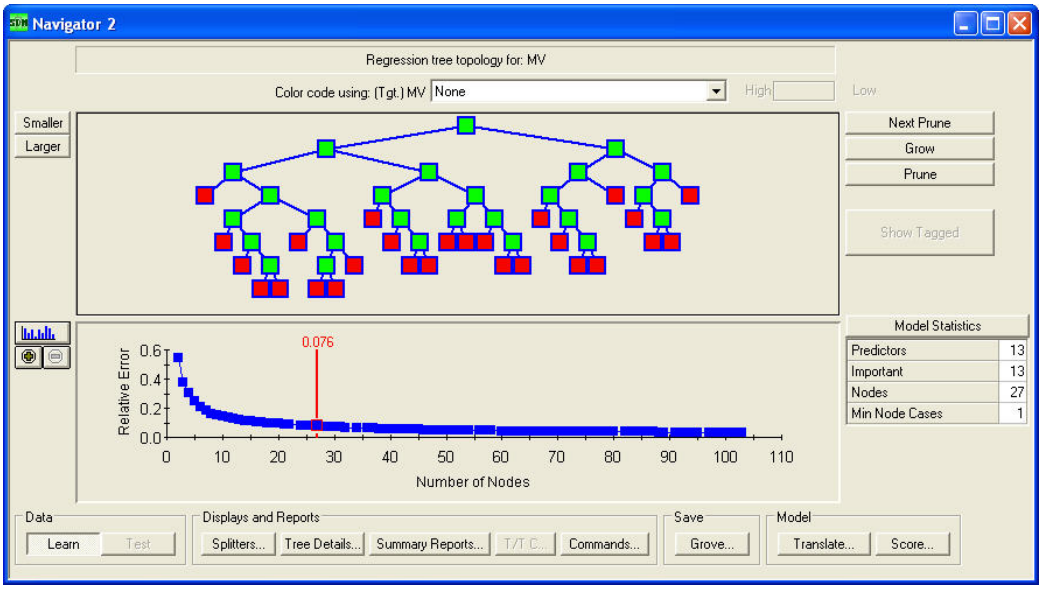


Figure 1. In the screen capture above we show a 27-terminal-node CART regression tree for the BOSTON.CSV data with performance measured on *training* data. The CART tree was run without a testing method. (We requested an "exploratory tree" on the TEST tab of the model setup dialog.) *The relative error reported for an exploratory tree is mathematically identical to the statistician's 1-R-Squared.*

It is important to understand that the R-Squared concept that statisticians (and editors of journals) have in mind is a measure of how well a model fits the data on which the model was built. The reason for any lingering confusion regarding R-Squared for CART trees is that in data mining we are often not particularly interested in such measures; thus, we have traditionally reported the training data performance only in the classic output. Why [are we not interested??]? It is well known that in any statistical model, including conventional regressions, R-Squared is an overly-optimistic prediction of model performance on previously unseen data. This is why experienced data miners insist on using independent test data to evaluate model performance, if possible. When a test sample is used, the relative error in the test sample measures how the model performs on data that were not used to construct the model. Cross validation is a special form of testing employed when there are insufficient data to put aside a true test sample; therefore, the relative error reported most prominently in the CART results pertains to the TEST data set or to the cross-validated results.

In the CART screen results we show the same tree reported above, but this time we display the performance results based on cross validation. Note that the relative error reported is 0.229, which we interpret as a cross-validated R-Squared of 0.771, much lower than the R-Squared of 0.924 reported for the training data. Which performance measure is correct? The training data R-Squared is a description of the goodness of fit obtained for the model on the training data and is the measure most similar to the R-Squared that conventional statisticians are accustomed to report. The test data R-Squared, on the other hand, is a prediction of how well the model is likely to perform on new data and thus is a more honest assessment of the predictive reliability of the model. Evaluating models on test data is mandatory in the field of data mining but has not yet caught on in conventional statistics.

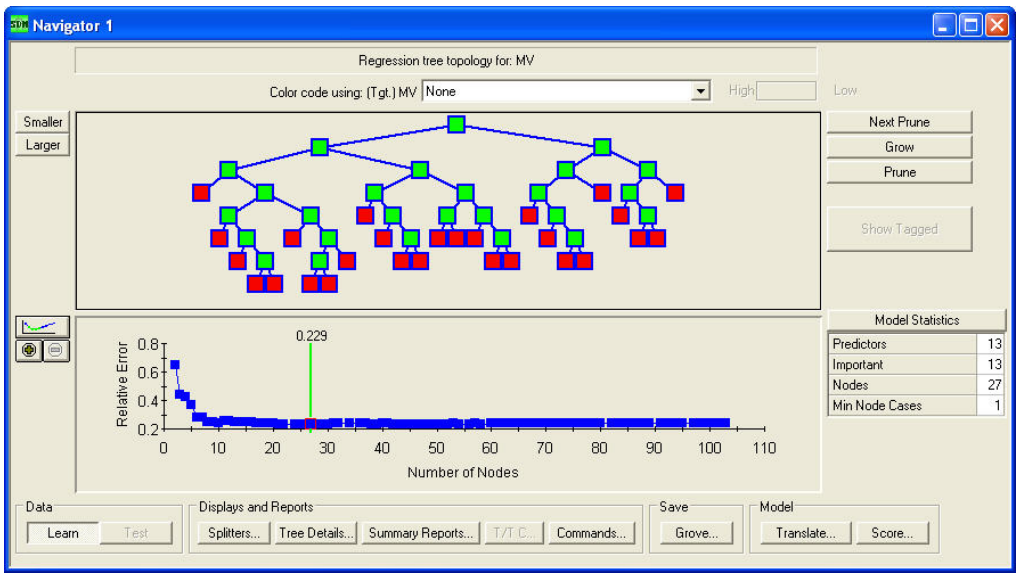


Figure 2. In the screen capture above we show a CART regression tree for the BOSTON.CSV data set included in the CART installation package. The Cross-Validated Relative Error for the optimal tree is .229, which is equivalent to a Cross-Validated R-Squared on test data of .771.

Because every CART regression tree reports a relative error for every tree in the tree sequence, you can read your R-Squared results not just for the tree CART has deemed optimal, but for any other tree you might prefer to use. Modelers often prefer to use trees that are smaller than the so-called "optimal" tree, especially when a much smaller tree can be obtained in exchange for a small increase in relative error (i.e., a very small decrease in R-squared). The CART navigator allows you to browse any of the trees in the CART tree sequence and also helps you make the best trade off between model simplicity (size of tree) and accuracy (R-Squared). In the CART navigator below we display the so-called “SE1” tree, which is the smallest tree within a 1 Standard Error band of the most accurate tree. This tree is much smaller, having only eight terminal nodes, but its accuracy is only slightly worse than the optimal 27-node tree.

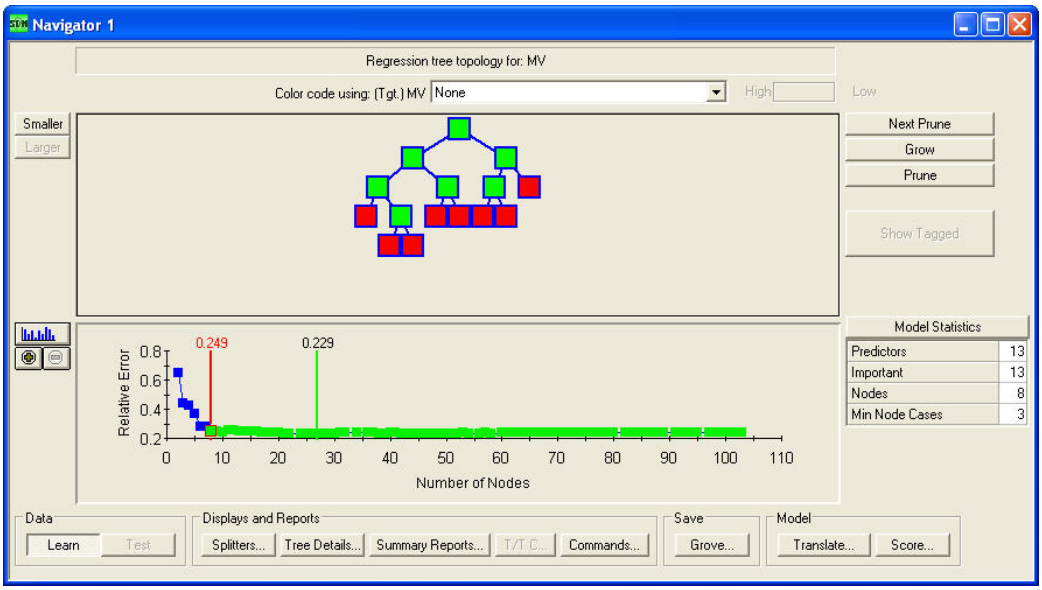


Figure 3. In the screen capture above we show the same CART tree displayed earlier but pruned back to just eight nodes (to the smallest tree within 1 SE of the best-performing tree). Observe that the cross-validated relative error is only slightly larger than the best-performing 27-node tree.

The classic CART output summarizes the performance of every tree available in the CART tree sequence, displaying both training and test data performance for every tree size. An extract from the tree sequence for the BOSTON data CART regression tree is shown below. (The table is taken from the CART Classic output window.)

As can be seen in the table below, the optimal tree, (the tree with the lowest *test* data relative error), has 27 terminal nodes. Notice that the much smaller eight-terminal node tree has only a slightly larger cross-validated relative error, .249 versus .229 (R-Squared of .751 versus .771).

If we naively used the training data (resubstitution) relative error to compare the two tree sizes, the eight-terminal model would appear to be much worse, .161 versus .076 (R-Squared of .839 versus .924). The test set measures have been shown to be far more reliable in many real world applications.

```
=============
Tree Sequence
=============

Dependent variable: MV
```

Terminal Tree	Nodes	Cross-Validated Relative Error	Resubstitution Relative Error	Complexity Parameter	Relative Complexity
1	103	0.241 +/- 0.047	0.034	0.000000	0.000
70**	27	0.229 +/- 0.045	0.076	89.376205	0.002
86	10	0.236 +/- 0.038	0.141	310.350128	0.007
87	9	0.252 +/- 0.040	0.148	317.400177	0.007
88	8	0.249 +/- 0.040	0.161	556.640015	0.013
89	7	0.279 +/- 0.042	0.185	1006.924927	0.024
90	6	0.282 +/- 0.042	0.212	1136.809204	0.027
91	5	0.367 +/- 0.049	0.245	1441.926392	0.034
92	4	0.421 +/- 0.048	0.304	2520.323730	0.059
93	3	0.438 +/- 0.047	0.376	3060.958496	0.072
94	2	0.644 +/- 0.051	0.547	7311.859863	0.171
95	1	0.982 +/- 0.017	1.000	.193396E+05	0.453

```
Initial mean = 22.533
Initial variance = 84.420
```

Figure 4. The table above, extracted from the CART classic output, reports both training (resubstitution) and cross-validated relative errors for trees pruned to different sizes (the tree sequence). The 27-node tree is marked with a double asterisk to indicate that it is the tree size with the lowest cross-validated relative error. Observe that training data relative error always declines with tree size, which is why we cannot use training data performance to select an optimal tree. Also note the performance measures reported for the eight-node tree, which we discuss further below.

Verifying the CART Regression

To demonstrate that you can replicate the CART predictions using a special form of regression we first display the CART predictions in each terminal node below:

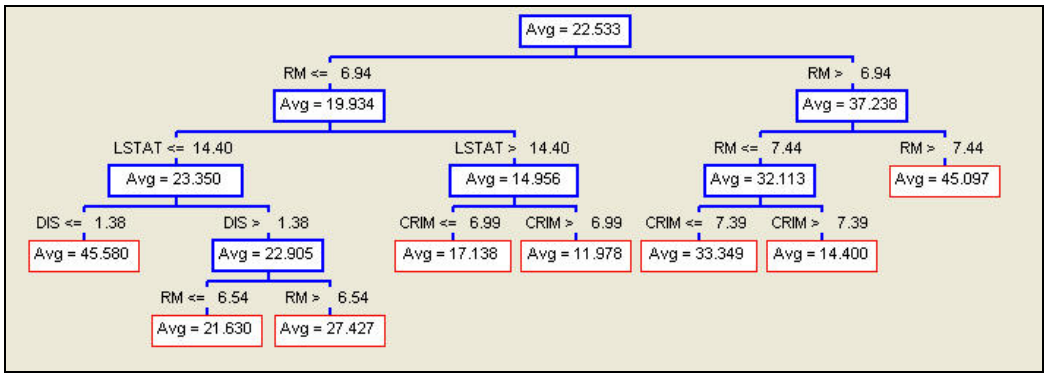
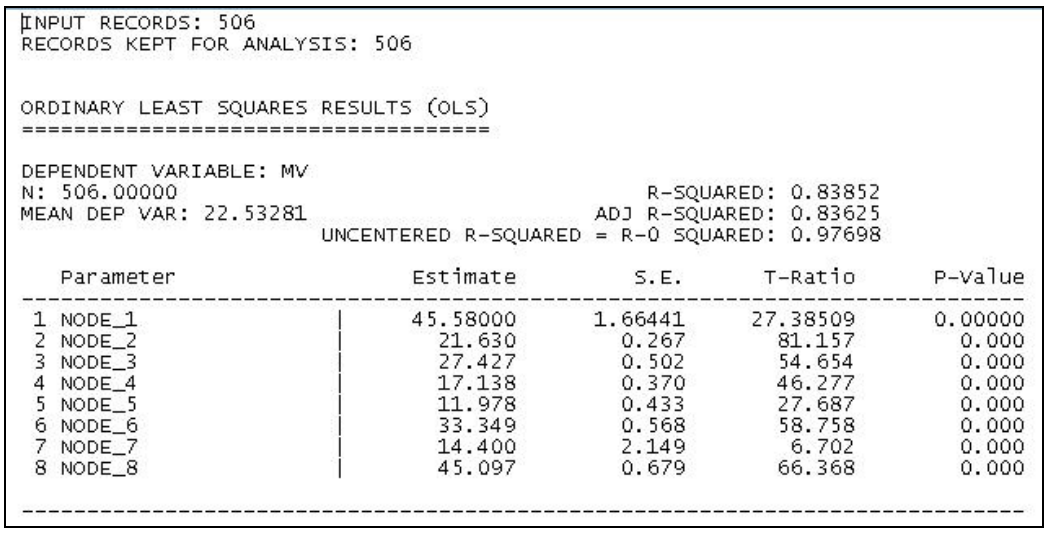


Figure 5. Tree Details for the eight-node tree are shown above. The average value of the dependent variable in each node is the CART prediction (the statistician's "yhat").

The CART regression yields results identical to running an ordinary least squares on a set of dummy variables for all the terminal nodes of the tree. If the data has no missing values, then the terminal node dummy variables are easily constructed from the rules defining terminal nodes. If there are missing values, then surrogate splitters become involved in determining how records are assigned to terminal nodes. Attempting to code this externally to CART by hand can be quite difficult. Your options instead include: (a) using the TRANSLATE facility in CART to generate computer code representing the primary and surrogate splitters of the tree, or (b) using the SCORE facility in CART to generate a new data set that will include the terminal node assignments in a variable called NODE. NODE contains an integer value running from 1 to K, where K is the number of terminal nodes in the tree you have selected for scoring. Running a regression using just this one (categorical) variable is sufficient to reproduce the predictions of CART on the training data and will yield the exact same R-Squared result as well. The results of just such a regression are shown below.



```
INPUT RECORDS: 506
RECORDS KEPT FOR ANALYSIS: 506


ORDINARY LEAST SQUARES RESULTS (OLS)
====================================

DEPENDENT VARIABLE: MV
N: 506.00000                                  R-SQUARED: 0.83852
MEAN DEP VAR: 22.53281                    ADJ R-SQUARED: 0.83625
                      UNCENTERED R-SQUARED = R-0 SQUARED: 0.97698

   Parameter                  Estimate         S.E.      T-Ratio      P-Value
-------------------------------------------------------------------------------
 1 NODE_1                |     45.58000      1.66441     27.38509      0.00000
 2 NODE_2                |       21.630        0.267       81.157        0.000
 3 NODE_3                |       27.427        0.502       54.654        0.000
 4 NODE_4                |       17.138        0.370       46.277        0.000
 5 NODE_5                |       11.978        0.433       27.687        0.000
 6 NODE_6                |       33.349        0.568       58.758        0.000
 7 NODE_7                |       14.400        2.149        6.702        0.000
 8 NODE_8                |       45.097        0.679       66.368        0.000

-------------------------------------------------------------------------------
```

Figure 6. Results of running a standard linear regression of the dependent variable MV on dummy variables for the terminal nodes of the CART tree. Above we have elected to use the eight-node tree. Observe that the reported R-Squared is exactly equal to the 1-Resubstitution Relative Error for the eight-node tree on the training data (resubstitution error).

As can be seen in the figure above, the regression recapitulates the CART model. For example, the regression coefficients are the same as the terminal node values shown in the figure for the Tree Details of the eight-node tree. If you run such a regression yourself to verify results from a CART tree you will want to run it with a constant to obtain the correct R-Squared. In our example above we report this R-Squared but show the coefficients from a regression without a constant in order to make the results easier to read. When run without a constant most statistical packages will report only the "uncentered R-Squared" and this is not the relevant concept for us. (To guarantee that the regression output will contain the correct R-Squared, run the regression on the CART terminal node dummies *with* a constant. We manipulated the results because we wanted to show the regression coefficients in a form that was most easily read.)

The core formulas governing CART regression trees appear in the original CART monograph Breiman, Friedman, Olshen and Stone (BFOS), *Classification and Regression Trees*, section 8.3.1, pages 223-225, where the authors explain that the measure of error for any CART tree is the sum of squared errors, summing across all observations and all terminal nodes. In the root node, before any splits are made (which is equivalent to the null model), error is measured as the sum of squared deviations of the dependent variable from its sample mean. Of course, dividing this sum of squares by N-1, where N is the sample size, is just the variance of the dependent variable. Splits are chosen to maximize the reduction in this sum of squared deviations across the terminal nodes. In other words, splits are chosen to maximize the resulting training data R-Squared.

Following BFOS, let $d(x_n)$ denote the CART tree prediction for observation "*n*," where x_n is the vector of independent variables for observation "n." The CART prediction is simply the mean

value of the dependent variable for the terminal node in which the given observation falls. The resubstitution estimate (i.e., the training data estimate) of the error rate for a least squares regression tree is defined by BFOS as

$$R(d) = \sum(y_n - d(x_n))^2 /N, \qquad (4)$$

where N is the sample size of the training data and the sum is over the training data. In traditional statistics texts you will often find R-Squared defined as

$$\text{R-Squared} = 1 - \text{SSE/SST}, \qquad (5)$$

where SST is equivalent to the sum of squared deviations of the dependent variable about the mean in the root node, and SSE is the sum of squared deviations across all terminal nodes. In CART, we report instead the SSE/SST (Relative Error) for all trees in the tree sequence, and report both training and test data results (when available).

Cart users sometimes look at one minus the test sample (or the cross-validated) relative error as an “honest” R-Squared. This has some merit, but it is important to be aware that such a fair measure of R-Squared is rarely computed in ordinary least squares regression and would, on average, give a lower value than any regularly reported type of R-Squared. If you are reporting the R-Squared for a CART model in order to compare its performance with a conventional regression, keep in mind that the conventional regression R-Squared will be based on the training data (unless you take special steps to measure performance on test data). To offer a fair comparison, you will want to report the CART training data R-Squared as well.