Euredit binary tree function: tree_score

1 Purpose

tree_score scores a decision tree computed by **anova_tree**, **gini_tree** or **waid_tree**.

2 Specification

```
#include <euredit_sys.h>


void tree_score (long rec1, long nvar, long nrec, long dblk, double data[],
                 void (*dfun) (long , long , double [] , int *), int root,
                 long opt_rand, long opt_score, long seed[], double res[],
                 double acc[], int *info)
```

3 Parameters

rec1

Input: the first data point in the data block.

Constraint: **rec1** ≥ 0.

nvar

Input: the number of variables in the data.

Constraint: **nvar** ≥ 1.

nrec

Input: the number of consecutive records, beginning at **rec1**, used in the calculations.

Constraints: **nrec** > 1 and (**rec1** + **nrec**) $\leq$ **dblk**.

dblk

Input: the number of records in the data block.

Constraint: **dblk** > 1.

data[dblk∗nvar]

Input: the element **data**$[i * \mathbf{nvar} + j]$ contains the (clean) data value for the jth variable of the ith data point, for $j = 0, 1, \ldots,$ **nvar** $- 1$; for $i = 0, 1, \ldots,$ **dblk** $- 1$.

Constraint: if **dfun** is a valid pointer, the value of **data** must be NULL.

dfun

Input: the function **dfun**, supplied by the user, must return the **dblk** data records starting at record **rec1**.

Constraint: if the value of **dfun** is NULL, **data** must be a valid pointer.
The specification of **dfun** is:

```
void dfun (long rec1, long dblk, double x[], int *error)
```

rec1

Input: the first record to be returned.

dblk

Input: the number of records to be returned.

x

Output: **x**$[i * \mathbf{nvar} + j]$ is the value for the jth variable, for $j = 0, 1, \ldots,$ **nvar** $- 1$; for $i =$ **rec1**, **rec1**$+1, \ldots,$ **rec1** + **nrec** $- 1$.

error

Output: if the value pointed to by **error** on return is greater than 100, the function will terminate immediately and **info** will point to this value.

opt_rand

Input: if the value of **opt_rand** is set equal to 1, a random number will be used to resolve dichotomies in the binary tree.

opt_score

Input: the value of **opt_score** determines the method used to score the binary decision tree. For a **gini_tree**, **opt_score** $= 1$ scores data with the modal class and **opt_score** $= 3$ uses random imputation from data at a leaf node. For an **anova_tree** or a **waid_tree**, **opt_score** $= 1$ scores data with mean values, **opt_score** $= 2$ scores data with the modal class and **opt_score** $= 3$ uses random imputation from data at a leaf node.

Constraint: **opt_score** $\in \{1, 3\}$ for a Gini decision tree.

Constraint: **opt_score** $\in \{1, 2, 3\}$ for an ANOVA or WAID decision tree.

seed[5]

Input: the value of the random seed used to resolve dichotomies. the elements of **seed** should be set by using the routine **set_rand_seed**. If the value of **opt_rand** is not equal to 1, **seed** is not referenced.

Constraint: if **opt_rand** $= 1$, **seed** must be a valid pointer.

res[nrec]

Output: **res**$[i]$ contains the decision tree value for the (**rec1** $+ i$)th data point, for $i = 0, 1, \ldots,$ **nrec** $- 1$.

acc[nrec]

Output: **acc**$[i]$ contains a measure of the accuracy of the prediction. For ANOVA trees, **acc**$[i]$ contains the variance about the mean value at the leaf node giving the ith prediction, for $i = 0, 1, \ldots,$ **nrec** $- 1$. For Gini trees, **acc**$[i]$ contains the probability of class membership at the leaf node giving the ith prediction, for $i = 0, 1, \ldots,$ **nrec** $- 1$.

root

Output: **root** points to an integer cast of the root of the decision tree.

info

Output: the value pointed to by **info** gives information on the success of the function call:

0: the function successfully completed its task.
-32: a path down the decision tree could not be found for a data point.
i; $i = 1, 2, 3, 4, 5, 6, 9, 10, 11, 12$: the specification of the ith formal parameter was incorrect.
99: the function failed to allocate enough memory.
100: an internal error occurred during the execution of the function.
> 100: an error occurred in a function specified by the user.
