Regression trees in the context of imputation of item non-response: an experimental application on business data

Enrico Grande (*), Orietta Luzi (*)

(*) Istat, Servizio Metodologie, Tecnologie e Software per la Produzione Statistica

Sommario

Le tecniche di partizione ricorsiva note come alberi di regressione sono comunemente utilizzate per predire valori mancanti di variabili numeriche continue dovuti a fenomeni di mancata risposta. I valori mancanti di un certo item statistico sono predetti sulla base di un insieme di covariate considerate esplicative del fenomeno oggetto di ricostruzione (o *target*). Diversi modelli per la costruzione di alberi di regressione possono essere utilizzati in combinazione con diverse tecniche di imputazione al fine di ottimizzare l'accuratezza dei risultati finali. Nel lavoro viene presentata un'applicazione sperimentale in cui diversi modelli di alberi di regressione sono utilizzati in combinazione con tradizionali tecniche di imputazione per la ricostruzione di valori mancanti simulati su dati di impresa.

Abstract

Regression trees are tree-structured methods generally used to predict missing values for continuous variables affected by non response. For a given statistical item (*target variable*), missing values are predicted on the basis of a set of explanatory variables (*covariates*). Different regression tree models can be used in combination with different imputation techniques, in order to optimise the accuracy of final results. In the paper an experimental application in which different regression tree algorithms are combined to traditional imputation methods to predict artificial missing values in a business survey is illustrated.

1. Introduction[1]

In Official Statistics non-responses represent the failure to obtain some of the asked items of information for some individual sample members. Generally, a distinction is made among *unit* (*total*) *non-response* and *item non-response*. A unit non-response corresponds to the absence of any information for a given sampling observation. A item non-response occurs when a given unit does not provide information for a subset of the questionnaire items.

Different procedures can be used to compensate for non-response (Kalton et al., 1986; Grande et al., 2003). Total non-responses are generally compensated by using *re-weighting* procedures, while item non-response is generally dealt with by *imputation*, which covers many techniques aiming at predicting suitable values for the missing items.

In the paper the quality of some imputation methods, supported by the use of regression trees, is evaluated through an experimental application on business data. Regression Trees are tree-structured methods used to predict the unobserved values of a continuous variable (*target variable*) by using appropriate explanatory variables (*covariates*). These methods, introduced by Breiman et al. (1984), perform a recursive partition of the measurement space in order to create subgroups of the target variable values characterised by increasing internal homogeneity.

In the application presented in the paper, different tree-models are built on experimental data. To this aim, the algorithms available in the SPSS software Clementine 6.5 have been used. The resulting data partitions are used in the imputation process: two imputation techniques have been used in each partition to compensate for non-response: the mean within cell and the random donor within cell techniques. A comparative evaluation of the statistical effects of imputation for each regression tree model has been performed.

In section 2 general issues on imputation and its link with regression trees are discussed. Section 3 contains the description of methodological aspects relating to regression trees. In section 4 the regression trees algorithms available in the software

[1] Il lavoro è stato svolto nell'ambito del programma di stage 2002 ISTAT/Agenzia Lazio Lavoro (Area Metodologica) presso la struttura MPS/B dell'ISTAT. A Enrico Grande vanno attribuite le attività di ricerca sulle metodologie illustrate, la loro sperimentazione e la valutazione comparativa dei risultati, nonché la stesura del lavoro. A Orietta Luzi vanno attribuiti l'impostazione generale e il supporto metodologico nelle attività di ricerca, sperimentazione e valutazione.

Clementine 6.5 are described. Finally, section 5 contains the description of the experimental application and results.

2. **Imputation and Regression trees**

In imputation, missing information is "predicted" by exploiting the available information coming from observed data. In other words, missing values are expressed as function of one or more observed items (covariates) used as explanatory variables for the investigated phenomenon. Almost all imputation techniques can be considered as special cases of the following general regression model:

$$y_{mi} = \beta_{r0} + \sum_j \beta_{rj} z_{mij} + e_{mi}$$

where y_{mi} represents the predicted value for variable Y missing in unit *i*, z_{mij} is the jth covariate value in unit *i*, β_{r0} and β_{rj} are the regression coefficients of Y on Z computed on respondents, e_{mi} is a residual random term. If we assume $e_{mi} = 0$ for any *i*, the imputation technique is said to be *deterministic,* otherwise it is defined as *stochastic.* The choice among different imputation techniques depends on the statistical objectives of the survey. Some applications can be found in Cirianni et al., 2001, Laaksonen, 2000, Di Zio et al., 2003.

In Official Statistics, the use of imputation for dealing with item non-responses is justified in literature on the basis of the following considerations (Kalton et al., 1982): imputation aims at reducing the effects on final estimates (in terms of both bias and precision) due to the presence of missing information (roughly, the effects due to the differences among the observed and non-observed data); imputation is carried out mainly for obtaining complete and consistent data sets that can be handled by traditional statistical data analysis tools; finally, imputation avoids inconsistent analyses that could arise if incomplete data are used. On the other hand, imputation can only reduce non-response bias with respect to known information: if this information is not properly modelled or if it is not used at all in the imputation model, imputation may distort the variables distributions and/or the data relations. Furthermore, once data have been completed by imputation, there is also the risk that analysts may treat the completed data set as if all the data were actual responses,

thereby overstating the precision of the survey estimates. A critical problem when dealing with imputation is the measurement of its effects on data relations (Kalton et al., 1986): for example if the study of data relations among a variable Y to be imputed and other survey variables is one important survey objective, these variables are to be used in the imputation model adopted for Y. For justifications and a discussion about statistical effects of imputation see Kalton et al. (1982).

Imputation techniques require that missing data are *missing at random* (MAR) (Rubin, 1986). Roughly, a mechanism originating missing data on a variable Y is said to be MAR when the probability that the Y is missing does not depend on the value of the variable itself. One way to approach this hypothesis, consists in performing imputation inside the so called *imputation cells* (or *classes*), resulting by appropriate *stratification* of units with respect to known covariates $X_1, X_2 ... X_n$ for the variable Y subject to imputation. Using imputation cells also produce a reduction of the biasing effects due to imputation (Haziza, 2002).

One way of obtaining imputation cells consists in the application of regression trees. This methodology is widely discussed in literature as a powerful tool for predicting the unobserved values of a continuous *target* variable by using appropriate sets of explanatory variables. In fact, the leaves of trees obtained through the recursive partition of the measurement space and containing subgroups of the target variable values internally homogeneous, correspond to imputation cells.

3. **Regression Trees**

The use of tree-structured methods, like **regression trees,** in the imputation process can be really seen as an application of a regression model in which some explanatory variables are used as covariates to predict the target variable values on the basis of some decisional rules.

In a regression problem an observation is represented by the couple (**x** , *y*) where *y* is a real value associated to the *response variable* Y, and **x** is a set of observed values (*covariates*) in the measurement space χ. The main purpose in a regression analysis consists in building a model "capturing" the relationships between the dependent variable and the covariates, thus allowing to predict the dependent variable values in the most accurate way.

Such a model is represented by a function $d(\mathbf{x})$ which takes real values in χ, called *predictor*. A regression tree is a *tree-structured predictor*, i.e. a set of (prediction) rules that partitions a data set into mutually exhaustive and non-overlapping subsets (*nodes*). These rules are defined using the values of a pre-defined set of categorical explanatory variables, and the model is built by successively splitting the data sets into subsets that are increasingly more homogeneous with respect to the continuous response variable of interest (*target variable*). The splitting continues until either some stopping rules are met, or the generated subsets are as much homogeneous as possible. The final subsets obtained from this process are called *tree terminal nodes* (or *leaves*). The typical structure of regression trees is shown in figure 1.

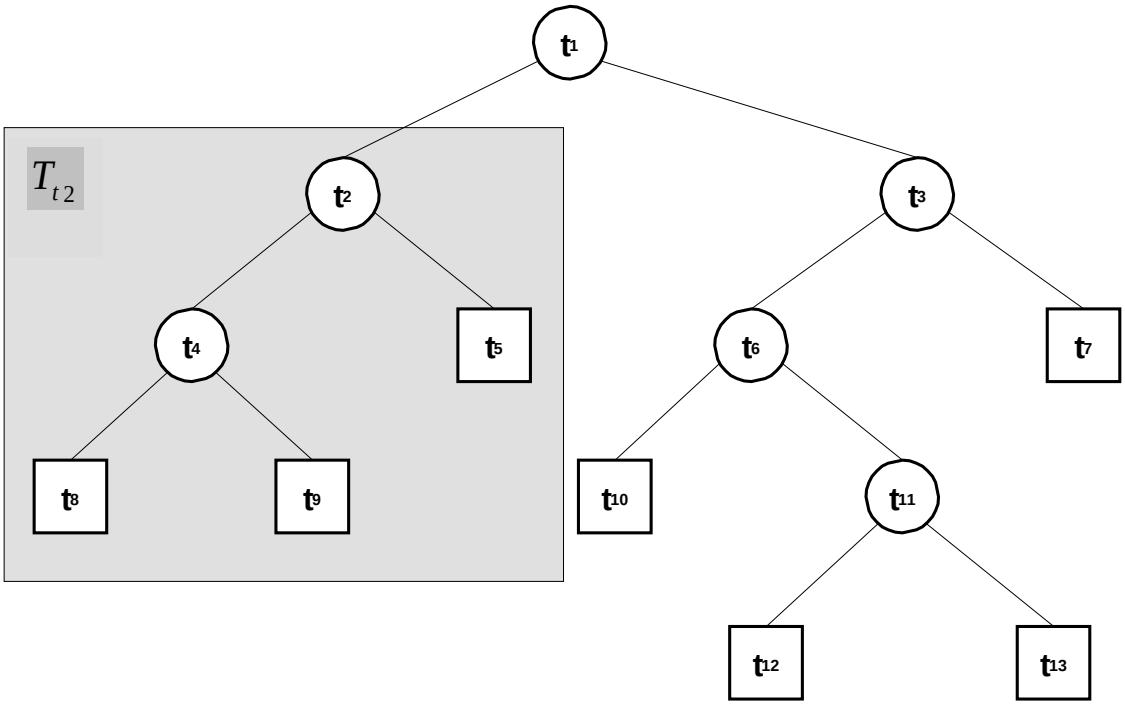
Using the terminology introduced by Breiman et al. (1984) we can classify the tree nodes into *non-terminal nodes* (round-shaped in figure 1), i.e. nodes are successively split, and *terminal nodes* (square-shaped in figure 1) no further sub-separable. Out of non-terminal nodes, t_1 represents the *root* (or *initial*) *node*. In general, a node splitting into two new nodes is called *parent node,* while the descending (or generated) nodes are called *child nodes*. Each terminal node of the tree can be viewed as an *imputation cell* for the response variable Y. A *branch* T_t of a regression tree *T* consists of a single node t and all its descending nodes. In Figure 1 the branch T_{t_2} includes all the nodes within the shadowed area.

3.1 Accuracy and Validation

Once a regression tree has been generated, it is important to evaluate its *accuracy*, i.e. the power of the tree to correctly predict missing values. For this purpose we can estimate the *error* of the tree, i.e. a measure of its inaccuracy. As error estimate of a generic predictor *d(x)* the mean squared error $R(d) = E(Y - d(x))^2$. The simplest (and most used) way to obtain the error estimate is to calculate it on the same data used for the tree building process (*re-substitution estimate*), but this approach often leads to an underestimate of the real error. An alternative good method consists in subdividing data into two distinct subsets: a *training* set (*learning sample*) which is used only to build the model, and a *test* set (*test sample*) which is used only to evaluate the goodness of the model. In practice, once the model has been generated (on training data), it is applied to test data in order to estimate the corresponding error. This validation process is suitable mostly for medium or

large dimension data sets. When the data set is small, better results can be obtained by using a *cross-validation* method (for more details see Breiman et al., 1984).

Figure 1 – *Structure of a binary tree*



3.2. Split mechanism and stopping rules

As already mentioned, in regression trees the tree-growing process is based on a *split mechanism* that recursively splits the sample into two subsets characterised by increasing internal homogeneity with respect to the target variable values. In other words, for each node the tree-growing algorithm chooses the partition of data generating nodes "purer" than their parents. In general, this process stops when the lowest impurity level is reached (the limit case occurs when a generated node is "pure") or when some *stopping rules* are met. These rules are generally settled by the analyst and relate to the maximum tree depth (i.e. the level reached by successive splitting starting from the root node), the minimum number of units in (parent and child) nodes, or the threshold for the minimum change in impurity provided by new splits.

In the regression tree-growing algorithm, the impurity of a node is measured by the *Least-Squared Deviation* (LSD) *R(t)*, which is simply the *within* variance for the node *t*. It can be expressed as follows:

$$R(t) = \frac{1}{N(t)} \sum_{i \in t} (y_i - \bar{y}(t))^2 \qquad [1]$$

where *N(t)* is the number of sample units in the node *t*, y_i is the value of the response variable for the i-th unit and $\bar{y}(t)$ is the mean (and the predicted value) of the response variable in the node t.

The *LSD criterion function* for split *s* at the node *t* is defined as follows:

$$\Phi(s,t) = R(t) - p_L R(t_L) - p_R R(t_R) \qquad [2]$$

where t_L and t_R are the left and right nodes generated by the split *s*, respectively, while p_L and p_R are the portions of units assigned to the left and right child node. The split *s* maximizing the value of $\Phi(s,t)$ is chosen. This value relates to the "improvement" in the tree, since it expresses the impurity reduction that can be obtained by generating the two child nodes. In other words the split providing the highest improvement in terms of tree homogeneity is chosen.

3.3. The Pruning Process

When the tree-growing process terminates due to some stopping rules, it often happens that the dimension of the generated tree is not appropriate (e.g. in terms of data homogeneity in final nodes). One critical problem relates to cases in which a very large tree is generated having a number of superfluous terminal nodes. In this case the generated tree is closely depending on the structure of training data and cannot be applied successfully to an external set of data. For this reason, sometimes a *pruning* process is used in order to obtain an "optimal" dimension tree. In regression trees this process (described in detail in Breiman et al., 1984) is called *Error-Complexity Pruning*: once the tree has been generated, the *Error-Complexity Pruning* process consists in removing the "weakest" (i.e. superfluous) splits of the tree. Briefly, this method is based on a measure of both *error* and *complexity* (the number of nodes) for the generated binary tree, defined as follows:

$$R_\alpha(T) = R(T) + \alpha \left| \tilde{T} \right| \quad , \qquad \alpha \geq 0 \qquad [3]$$

where $\left|\tilde{T}\right|$ is the number of terminal nodes for the tree *T*, *R(T)* (= $\sum_{t\in\tilde{T}} R(t)$) is the error associated to the tree *T*, and α represents the additional cost, in terms of complexity, for each terminal node. The tree-growing process is carried out by successive splits in order to minimize *R(T)* without taking into account the increasing complexity of the tree. For this reason the initial step of the pruning algorithm consists in growing the tree with the maximum size ($T_{\max}$), in order to capture all potentially important splits. Starting from $T_{\max}$, a hierarchical sequence of $T_{\max}$ sub-trees $(T_1 > T_2 > ... > T_{root})$, and a corresponding increasing sequence of α values $(0 = \alpha_1 < \alpha_2 < ...)$, where $T_k = T(\alpha_k)$, is then generated, cutting out at each step the nodes whose additional cost (in terms of complexity) does not contribute significantly to the reduction of the tree error-complexity. The optimally sized tree, resulting from this process, is the smallest sub-tree of $T_{\max}$ minimizing $R_\alpha(T)$ (for further details on the pruning algorithm see Breiman et al., 1984).

4. **The application**

The application aims at evaluating the performance of imputation strategies in which regression trees and different imputation algorithms are combined together for predicting item non responses. The application has been developed as follows.

For our evaluation purposes, the simulation approach has been adopted: given an initial set of complete data (*true* or *original data*) for a pre-defined target variable, the variable values have been contaminated by artificially generating missing values among them. In this way, the quality of imputations (in terms of capability of correctly predicting the true data) can be measured by simply comparing original and imputed data. Only the MCAR mechanism (Missing Completely At Random, see Rubin, 1987) has been simulated. Two different percentages (5% and 10%) of missing values have been generated in order to evaluate the effect of the missing response amount on the quality of imputations.

For the target variable a pre-defined set of covariates have been identified. As further analysis case, different percentages of missing values have been simultaneously simulated on the target variable (10%) and on one of its covariates (5%). In this way the impact of missing values for covariates on the imputation process has been evaluated too.

When all variables to be used for the tree-growing process have been identified, different data partitions have been generated through the Clementine 6.5 regression trees algorithms. For each algorithm/partition, two different imputation techniques have been used: the *mean within cell* and the *random donor within cell* techniques.

In this general setting, the expression *imputation strategy* indicates a combination of each applied regression tree algorithm and each imputation technique. For each percentage of missing values, the evaluation of results produced by each imputation strategy has been performed through simple indicators (Chambers, 2000) measuring the quality of the performed imputations in terms of preservation of marginal and joint distributions.

4.1 Regression Trees in Clementine 6.5

Clementine (version 6.5) is an SPSS generalised software for data mining (SPSS, 2001c). It offers many modelling techniques, such as prediction, classification, segmentation and association detection algorithms, integrated by a visual programming interface. Clementine is designed to operate on computer systems running Windows 95, Windows 98, Windows 2000 and Windows NT 4.0 with service pack 3 or higher. The data sources can be free or fixed-field ASCII data, ODBC data, SPSS or SAS imported data.

Clementine allows using many parametric and non-parametric data mining and data analysis models, including Classification and Regression Trees (C&RT in the following).

The C&RT algorithms implemented in Clementine allow generating regression trees through two different procedures: a standard pre-defined procedure (called *Simple option*), and a customized one, in which the so-called *Expert options* can be used to modify some parameters of the standard pre-defined algorithm.

Using the *Simple option* the tree-growing algorithm runs using some default settings: the *minimum number of units* in a parent node (2% of total cases) and in a child node (1%), or the *minimum change in impurity* produced by each new split (0,0001). Using the Simple option only the *maximum tree depth* can be changed.

When using the Expert options, alternative settings for the different steps of the algorithm can be specified. Each option corresponds to a different approach in the tree building up process. Relating to regression trees two options are available:

- *Stopping and Pruning*. With this option, the tree is generated and then pruned in order to find the "optimally sized" sub-tree by using the *pruning algorithm* described in

paragraph 3.3. The *minimum number of units* (both percentage and absolute values) in a parent or in a child node can be specified.

- *Impurity and Surrogates.* This option allows to specify a different threshold (with respect to the default one) for the *minimum change in impurity* for each generic split. The maximum number of *surrogates*, i.e. variables automatically used by the algorithm when one or more records have missing values in the split variables, can be also specified.

A *Weighting option* (not used in this application), allowing the control of records *weighting settings*, is available in the software.

Once the tree-based model has been generated on training data, it can be applied to test data including the missing values for the output variable. For regression trees, in each terminal node the software automatically imputes each missing value by using the response variable *mean*.

4.2 Data description

Data used in the application correspond to a subset of anonymous observations surveyed in the U.K Annual Business Inquiry (ABI)[2]. The survey collects information on variables like *turnover*, *employees* and *employment costs*, *taxes* and *purchases*. The data set consists of 6,099 units, each corresponding to either a *long* (1,481 units) or a *short* (4,618 units) form, the latter asking for a restricted amount of information. In this application only *short forms* data have been used.

After carefully examining data, we have chosen (as also suggested by ONS) to perform two separate analyses on two sub-groups of firms identified by using the *registered turnover* (TURNREG), corresponding to the business turnover resulting from administrative registers. The original data set has been thus divided into the subsets *Small* firms (TURNREG < 1 million £), consisting of 3,731 cases, and *Large* firms (TURNREG $\geq$ 1 million £), containing 887 cases. This data grouping allows to take into account the differences existing between large and small businesses which have reasonably to be considered as distinct subgroups of analysis.

[2] Data have been provided by the British Agency for Official Statistics (ONS) in the context of the EUREDIT project (*www.cs.york.ac.uk/euredit/*) supported by the European Union in the Fifth IST Program for Research and Development.

4.3 Model settings

A regression trees-based model requires a continuous output variable (i.e. the variable to be imputed) and a set of categorical covariates. In this case the output variable is the TURNOVER, while two complete (i.e. without missing values) categorical variables have been used as model covariates: the class of economic activity for each business (CLASSACT), and number of employees for each inquired business (CLASSEMP), whose categories are classes of employees. The covariates used in this application allow to take into account both high (CLASSEMP) and low (CLASSACT) correlations with the output variable TURNOVER.

Concerning the variable CLASSACT, it has 7 categories, corresponding to the main branches of economic activity, while the variable CLASSEMP has 5 categories in the *Large* case, and 3 categories in the *Small* dataset, according to the different internal composition of the two data sets (table 1).

Table. 1 – *Categorization of the variable EMPLOY into the variable CLASSEMP*

Small (TURNREG < 1 million £*)*		*Large (TURNREG* >= 1 million £*)*	
N° of employees	**CLASSEMP**	**N° of employees**	**CLASSEMP**
< 10	1	< 10	1
10 <= … < 20	2	10 <= … < 20	2
>= 20	3	20 <= … < 50	3
		50 <= … < 100	4
		>= 100	5

The low number of categories in both the two considered covariates should prevent the creation of imputation cells containing very few cases, which would lead to trees that don't apply successfully to an external data set (i.e. a data set which differs from the *training* set).

4.4 Missing values simulation procedure

Once variables to be used in the analysis have been selected, a pre-defined amount of missing values has been artificially generated on TURNOVER in order to simulate a real situation in which missing data randomly contaminate the target variable. The simulation procedure has been carried out using the generalised software E.S.S.E. (Della Rocca et al., 2000), that allows to generate errors and item non-responses on a set of complete data. This approach allows the evaluation of the performance of imputation methods through the

comparison of the original complete data and the corresponding imputed ones. In E.S.S.E., the simulation can be carried out according to different error models (MAR, MCAR). In this application the following simulations have been performed:

1. 10% of MCAR values generated on the output variable TURNOVER;
2. 5% of MCAR values generated on the output variable TURNOVER;
3. 10% of MCAR values generated on the output variable TURNOVER and 5% of MCAR values generated on the covariate CLASSEMP[3].

The contamination of an input variable aims at testing the performance of tree-growing processes in more complex and realistic situations, characterised by incomplete training data sets. The presence of covariates affected by missing data is automatically dealt with in Clementine 6.5 by using the so-called *surrogates* (see par. 4.1).

After each simulation procedure the resulting perturbed data have been split into *Small* and *Large* data sets (see par. 4.2) before data modelling and imputation.

4.5 Tree-growing algorithms.

For both *Small* and *Large* subgroups, complete cases represent the training data on which the tree has to be estimated, while cases with missing values for the output variable TURNOVER represent test data on which the tree-model has to be applied. Only in case of missing values generated also on the covariate CLASSEMP, the training data set is an incomplete data set which needs to be treated as a complete one by using surrogates (see par. 4.1, Clementine 6.5 options).

Once training and test data sets have been defined, the most suitable Clementine's C&RT algorithm to be used on data has to be chosen. This is a very important phase of the application, involving many theoretical considerations about the tree-growing algorithms implemented in the software.

In our experimental application, the *Simple* and the *Stopping and pruning* algorithms with different parameters settings (including a "pure" *pruning* model), have been applied to data. The basic idea underlying the choice of such tree-based models is to carry out the analysis by firstly applying a "basis" algorithm (using the *Simple* option, see section 4.1) and then more specific algorithms (using the *Expert* options) in which the parameters

[3] Missing values has been generated on the continuous variable "Number of Employees", subsequently transformed in the categorical variable CLASSEMP.

settings of the basic model have been modified in the appropriate way. Out of the available *Expert* options, in our application the *Stopping and Pruning* model came out the only one allowing the use of a tree-growing algorithm considerably different from that obtained by the *Simple* option. In particular, the *Stopping and Pruning* option was the only allowing the application of an effective pruning process. On the contrary, the *Impurity and Surrogates* model (allowing to set a different threshold for the minimum decrease in impurity and a different number of variables to be used as surrogates) has been rejected because it was not able to produce trees different from those generated by the *Simple* model.

The tree-growing algorithms used in this application and their main characteristics are summarised in table 2. As it can be seen in the table, three different algorithms have been applied using the *Stopping and Pruning* option. Among them, two algorithms (*Stopping & Pruning 1 and Stopping & Pruning2)* represent variants of the *Simple* model obtained, respectively, by increasing and reducing the absolute minimum number of cases required in parent and in child nodes. It is important to underline that when setting a different minimum number of cases in nodes, both the corresponding default proportions in the *Simple* model and the size of the data sets (*Small* and *Large*) have to be taken into account. The third algorithm (*Pure Pruning*) aims at creating an optimally sized tree using a pruning procedure, which represents a tree-growing strategy substantially different from that followed by the other algorithms.

Table 2 - Clementine's 6.5 tree-growing algorithms used in the application

Using the Simple option:

Model name	Max tree depth	Min change in impurity	Min number of cases in parent nodes (percent.)	Min number of cases in child nodes (percent.)	Prune tree option
SIMPLE	10	0.0001	2%	1%	NO

Using the Stopping and Pruning option:

Model name	Max tree depth	Min change in impurity	Min number of cases in parent nodes (abs. values)	Min number of cases in child nodes (abs. values)	Prune tree option
STOPPING & PRUNING 1	10	0.0001	Small: 100 - Large: 100	Small: 50 - Large: 50	NO
STOPPING & PRUNING 2	10	0.0001	Small: 30 - Large: 8	Small: 15 - Large: 4	NO
PURE PRUNING	10	0.0001	Small: 30 - Large: 8	Small: 15 - Large: 4	YES

4.6 Generated models

Once a given tree model has been generated, it is possible to visualize its structure as in the example shown in figure 2. The figure illustrates the successive splits originated by a regression tree algorithm in Clementine (the covariate used in the split and the values defining the split rule are indicated) and some information about nodes like the number of cases in each node (in italics), the mean value of the target variable in each node (*Ave*), its change after the split (*Effect*), and the proportion of cases for which the splitting rule is true (*confidence*). Moreover, Clementine also provides for each terminal node of the tree the mean values to be used to impute the target variable in that node (marked values).

In general we observed some differences between the generated models reflecting the different algorithms parameters settings. In fact, except for the *Small* subset affected by 10% of missing values on the target variable and by 5% on the CLASSEMP covariate, the procedure *Stopping & Pruning 1* led to a contraction of the tree structure, especially in the *Large* subgroup, while when using the procedure *Stopping & Pruning* 2, in general more complex trees have been obtained.

The *Pure Pruning* procedure has been used in order to estimate the effects of a pruning process on the models creation. Such procedure operates by carrying out the pruning process on the tree generated using the *Stopping & Pruning 2* algorithm. The main result of the latter algorithm was that in all analysed cases, nearly all branches of the tree were cut out. In other words, almost all of the generated splits were considered superfluous. In the more extreme cases, only the first binary split (always using the CLASSEMP variable) remained in the final tree, and sometimes also this split was removed, leading to the conclusion that any kind of stratification of the target variable with respect to the analysed covariates did not produce subsets of values more homogenous than the initial one. A similar result can be due to the particular structure of data, which originated some problems also during the models setting. This can be seen for example for the data set *Small*, where a little group of observations (115) turned out to be the most homogenous (so no further subdivisible) with respect to the variable CLASSEMP (see figure2) just after the first split carried out by Clementine, making the tree to grow up only in one direction. For this reason in models built by using the *Stopping and Pruning* option, very low thresholds for the minimum number of cases in *parent* or *child* nodes have been defined, obviously

leading to a deeper growth of the tree, in part due to the superfluous splits then removed by the pruning process.

Once models have been estimated on training data, they have been applied to test data, thus providing imputation cells for all the missing values of the target variable.

Figure 2 – *An example of regression tree structure as shown in Clementine's browser window.*

classemp [2.000000 1.000000] [Ave: 252.639, Effect: -47.856] (3171)
  classemp [1.000000] [Ave: 204.682, Effect: -47.957] (2793)
    classact [6.000000] [Ave: 444.5, Effect: +239.818] (104, 1.0) -> 444.5
    classact [7.000000 5.000000 4.000000 3.000000 2.000000 1.000000] [Ave: 195.407, Effect: -9.275] (2689)
      classact [4.000000 2.000000] [Ave: 207.486, Effect: +12.079] (1726)
        classact [2.000000] [Ave: 205.366, Effect: -2.121] (1603, 1.0) -> 205.366
        classact [4.000000] [Ave: 235.122, Effect: +27.636] (123, 1.0) -> 235.122
      classact [7.000000 5.000000 3.000000 1.000000] [Ave: 173.758, Effect: -21.649] (963, 1.0) -> 173.758
  classemp [2.000000] [Ave: 606.987, Effect: +354.348] (378)
    classact [6.000000 4.000000] [Ave: 782.912, Effect: +175.925] (34, 1.0) -> 782.912
    classact [7.000000 5.000000 3.000000 2.000000 1.000000] [Ave: 589.599, Effect: -17.388] (344)
      classact [3.000000] [Ave: 499.629, Effect: -89.97] (70, 1.0) -> 499.629
      classact [7.000000 5.000000 2.000000 1.000000] [Ave: 612.584, Effect: +22.985] (274)
        classact [5.000000 1.000000] [Ave: 591.021, Effect: -21.563] (47, 1.0) -> 591.021
        classact [7.000000 2.000000] [Ave: 617.048, Effect: +4.465] (227, 1.0) -> 617.048
classemp [3.000000] [Ave: 1620.061, Effect: +1319.566] (115, 1.0) -> 1620.061

4.7 Imputation techniques

In each terminal node of each regression tree model obtained as shown in previous section, two different imputation techniques have been applied: the *mean within cell* without residual term (*mwc* in the following) and the *random donor within cell* (*rdwc* in the following).

Let Y be the target variable observed on a sample S of units s_i (i=1,...,n). In the *mwc* technique, all the y_i corresponding to missing values of Y in cell c are replaced by using the Y observed mean in c, say $mean^c(Y_{obs})$. In the *rdwc* technique, each missing y_i is replaced by a value y_j ($j \neq i$) randomly selected among the Y_{obs} in cell c. Chen et al. (2000) describe the adva5ntages relating to the use of donor methods, and provide evidence of some theoretical results on their validity.

In Clementine the *mwc* technique is automatically performed by simply applying the generated models to test data. For each algorithm, the results of the imputation process can be analysed in a table showing the new distribution of the target variable.

The *rdwc* method has been applied using an ad hoc SAS procedure.

4.8 Evaluation measures

The comparative evaluation of the applied imputation strategies has been performed by comparing the original (true) data and the final data produced by each imputation strategy.
The evaluation has been lead by using some quality indexes available in the software IDEA (Index for Data Editing Assessment) (Della Rocca et al., 2003; Luzi et al., 2001). This tool allows the evaluation of the effects of editing and imputation procedures on a set of statistical survey through a number of indicators measuring the impact of the data treatment at two levels: *micro* (i.e. preservation of single values for each variable) and *macro* (i.e. preservation of marginal distributions and relations between variables).
Out of the indicators available in the software, in this experiment the following measures have been considered for the evaluation purpose:

- univariate characteristics (*quartiles, standard deviation, maximum and minimum values, median, mode*) of the target variable marginal distribution before and after imputations;
- the Kolmogorov-Smirnov index (*KS*), measuring the "distance" between the original and the final marginal distributions of the target variable;
- the Pearson's correlation matrix between the target variable and other survey variables, before and after imputations.

4.9 Results and considerations

In this section the results of the experimental application are illustrated and some general considerations are drawn. The evaluation of the imputation procedure obviously relates to results obtained for the target variable TURNOVER.

4.9.1 Evaluation of the main effects on the marginal distribution of the target variable

For each type of missing data simulation carried out in the application, tables 3, 4 and 5 illustrate the values of the performance indicators measuring the effects on the TURNOVER distribution of the *mwc* and *rdwc* imputation techniques combined with different regression-trees algorithms. In the first two columns of each table, the distribution characteristics (*quartiles, standard deviation, maximum and minimum values, median, mode*) of original data are reported separately for *Small* and *Large* businesses. In the following columns the same indicators after the imputation process plus the KS index are shown.

4.9.2 The Kolmogorov-Smirnov index

An overall evaluation of the ability of each imputation strategy in restoring the original marginal distribution of the target variable can be done by considering the *KS* value.

According to the *KS* index, it seems that for the three percentages of simulated missing data, all the imputation strategies give good results in terms of preservation of the marginal distribution (the index values don't exceed 0,085). In particular, as expected, in case of 5% of missing values, results are better than in the other two missing data levels (due to the intuitive fact that a lower amount of missing values reduces the risk of introducing distortions on the overall data distributions through imputation). Also the different composition of *Small* and *Large* businesses data sets seems to influence the imputation results: with *rdwc* imputation the highest values of *KS* are obtained for the *Large* data set while with *mwc* the highest values of *KS* are those corresponding to the *Small* data set.

In general, the performance of strategies involving the *rdwc* technique is slightly better than those using the *mwc* one, mostly because of the deterministic nature of the latter imputation method.

In particular we can notice that in case of 10% of missing values on the only variable TURNOVER (table 3) the better results with *rdwc* have been obtained with the *Simple* model (*KS* equal to 0,004 for *Small* businesses and 0,008 for *Large* ones). With 5% of missing values (table 4) it seems to be preferable to apply the *rdwc* method in combination with the tree generated by a pruning process (*Pure Pruning* model). Considering the case TURNOVER 10% of missing values and CLASSEMP 5% of missing values (table 5), the better model supporting *rdwc* is the *Stopping & Pruning 1* (*KS* equal to 0,005 for *Small* and 0,010 for *Large*), i.e. the model leading to trees wider than those created by the basic model *Simple*.

4.9.3 Preservation of the distribution mean

Concerning the preservation of the mean, in case of TURNOVER with 10% of perturbed values (table 3), good results have been obtained with both the applied imputation techniques. Indeed, for the *Small* data set, the *rdwc* imputation seems to provide estimates slightly better than those turned out from the *mwc* procedures. On the contrary, for *Large* businesses the *mwc* method often leads to estimates substantially better than those

produced by the *rdwc* method, in particular using a tree smaller than the one generated with the *Simple* model (*Stopping & Pruning 1*) or a pruned tree (*Pure Pruning*).

The analysis carried out on data affected by 5% of missing values on TURNOVER (table 4) shows a general improvement in the sub-group means under both imputation methods. It must be underlined that in some cases *rdwc* results are slightly better for both *Small* and *Large* businesses.

When both TURNOVER and the CLASSEMP covariate are perturbed (table 5), we observe a general better performance of *rdwc*, particularly in the *Small* data set and under models generated with the *Simple* and *Stopping & Pruning 1* algorithms. It has to be noted that in this case, both imputation techniques produce an evident positive bias on means in both *Large* and *Small* sub-samples, while in case of 10% missing values on only the TURNOVER item, the resulting bias was every time negative.

4.9.4 Effects on variability

Another important element to be taken into account when evaluating imputation is represented by the analysis of the effects it produces on the variability of the target variable distribution. Here the adopted measure for variability is the Standard Deviation (*STD*).

The only evident effect on the distribution is represented by the consistent reduction of data variability produced by all strategies on the *Small* data set when the percentage of missing values is 10%. In this case, the most reasonable explanation relates to the distortion introduced on the target distribution during the error-simulation phase: as we can see in table 3, some critical distribution values, like the *maximum* value (56,000), have been turned into missing values. In this case, neither the *rdwc* nor the *mwc* imputation methods were obviously able to restore the original data variability.

Out of this particular case, we observe a general good performance of both the applied imputation techniques in terms of preservation of the distribution shape.

In case of 10% missing values on only the TURNOVER, in *Large* businesses all imputation strategies produce a slight negative bias in the *STD.* Note that in this sub-set of units, the distortion is lower in the *mwc* than in the *rdwc* method, except for the pure pruning scheme, probably because of the higher data variability characterising the terminal nodes in the *Pure Pruning* scheme.

Of course, better results are obtained when the percentage of missing values is low, (table 4): in this case the better result corresponds to tree-models built up by using pure pruning algorithms (*Pure Pruning*), while both the imputation techniques produce positive bias in *Large* businesses when the *Stopping & Pruning 2* algorithms is adopted.

On the contrary, when also the covariate CLASSEMP is perturbed (table 5) we notice good results for the *Small* data set with all used algorithms, while in *Large* businesses the *rdwc* algorithm produces a remarkable increase of the *STD* with respect to the *mwc* technique under all the tree algorithms. This fact could be explained taking into account the stochastic nature of the method combined with the higher variability of large businesses and the biasing effects on predicting the target variable when surrogates are used in the model.

In general, the *rdwc* method performs better than the *mwc* in terms of preservation of median and quartiles, mainly due to its stochastic nature.

Out of the application where the maximum was removed (10% of missing values on TURNOVER), imputation strategies preserve variability and shape better in the *Small* case than in the *Large* one under all the imputation strategies. We can explain this fact taking into account the high variability of data in the *Large* sub-set, in which variations are more difficult to control.

4.9.5 Effects on relations between variables

Another key element to be considered when evaluating imputation procedures is represented by the effects they produce on relations between the output variable and other target survey variables. To this aim, for each missing data level, the changes occurred in correlations between variables (measured by the *Pearson correlation index*) have been evaluated. Variables *Turnover*, *Total purchase* (PURTOT), *Total taxes* (TAXTOT) and *Registered turnover* (TURNREG) have been considered. Tables 6, 7 and 8 show these correlations before (first two columns) and after (remaining columns) imputation.

Because of the fact that these variables were no used neither in the regression tree model, nor in the imputation models, we expect biasing effects on their relationships with the output variable subject to imputation. On the other hand, in the regression tree model we were forced to use variables free of missing values and errors, in order avoid that the Clementine algorithm selected “surrogates”, in case of covariates affected by missing

values, in a not controlled way. What we want to measure is the possible levels of biases on data relationships that we risk to have in this type of applications.

Results obtained in case of 10% of missing values on variable TURNOVER (table 6) give rise to many discussions. The most peculiar case relates to the *Small* data set. While we observe a low reduction of the correlation level between *Turnover* and *Total purchase* (with the *mwc* surprisingly performing better than the *rdwc* technique) and between a light increase of the correlation level (originally very low in the *Small* case) between *Turnover* and *Registered turnover*, considerable distortions on relationships between *Turnover* and *Total taxes* have been introduced by all imputation strategies. In fact the correlation index decreases from 0,7383 (in the original data set) to values that do not exceed 0,0579 (in the correct data set). Taking into account that the same result characterises all combinations tree algorithm/imputation method, and that such situation represents an "isolated" case in the analysis, we can conclude that the distortion has to be largely considered as a consequence of the removal of *maximum* (just seen in the previous paragraphs) during the missing simulation procedure. Out of this particular case, relationships between variables involved in the analysis are slightly biased by all the applied procedures both for *Small* and *Large* businesses.

In case of 10% of missing values on the only variable TURNOVER (table 6), the lowest bias corresponds to variable *Total taxes* in Large businesses. Note that the correlation between the output variable and the *Registered turnover*, originally higher in the *Large* sub-set, is slightly positively biased in the *Small* stratum while it results slightly decreased in *Large* businesses.

A general improvement of results when the percentage of the missing values on TURNOVER diminished from 10% to 5% (table 7) was obviously expected. On the other hand, low bias in relations can be observed also after perturbing cases of the covariate (table 8) used in the tree algorithm. In general, we observe that in both these two last applications the most relevant effect produced by imputations consisted in a slight reduction of the correlation between variables, more evident when the covariate is affected by missing values and a pruning procedure is used to generate the tree (*Pure Pruning* model). As previously underlined, this is mainly due to the fact that the tree generated by a pruning process is in general more generalised, i.e. it applies more successfully to data sets different from those used for the "training" of the model, but at the same time it supplies a

very small number of imputation cells leading to poorer results in terms of preservation of marginal distribution and relationships.

Table 3 – Preservation of distribution and aggregates, Kolmogorov-Smirnov Index
Variable: **TURNOVER**
Percentage of simulated missing values **10%**

	Original Data		Simple				Stopping & Pruning 1			
			Mean Within Cell		Random Donor		Mean Within Cell		Random Donor	
	Small	Large	Small	Large	Small	Large	Small	Large	Small	Large
Mean	313,158	5.250,788	299,065	5.277,073	299,116	5.151,601	298,957	5.240,313	299,379	5.122,241
STD	1.299,484	8.391,659	915,115	8.156,523	930,733	8.131,587	916,046	8.134,787	931,001	8.116,460
Max	56.000,000	113.405,000	50.355,000	113.405,000	50.355,000	113.405,000	50.355,000	113.405,000	50.355,000	113.405,000
Q3	370,000	5.835,000	356,000	5.913,000	369,000	5.848,000	356,000	5.835,000	370,000	5.913,000
Median	168,000	2.638,000	184,000	2.589,000	170,000	2.638,000	184,000	2.649,000	170,000	2.588,000
Q1	87,000	1.560,000	94,000	1.566,000	87,000	1.535,000	94,000	1.566,000	87,000	1.561,000
Min	1,000	0,000	1,000	0,000	1,000	0,000	1,000	0,000	1,000	0,000
Mode	100,000	1.200,000	205,000	2.541,000	200,000	1.000,000	205,000	2.451,000	200,000	1.500,000
KS	-	-	**0.061**	**0.024**	**0.004**	**0.008**	**0.061**	**0.034**	**0.004**	**0.009**

Table 3

	Original Data		Stopping & Pruning 2				Stopping & Pruning 3			
			Mean Within Cell		Random Donor		Mean Within Cell		Random Donor	
	Small	Large	Small	Large	Small	Large	Small	Large	Small	Large
Mean	313,158	5.250,788	299,456	5.223,177	299,571	5.128,652	300,436	5.259,192	301,795	5.178,346
STD	1.299,484	8.391,659	916,946	8.157,323	930,611	8.106,073	911,472	8.121,587	933,620	8.145,127
Max	56.000,000	113.405,000	50.355,000	113.405,000	50.355,000	113.405,000	50.355,000	113.405,000	50.355,000	113.405,000
Q3	370,000	5.835,000	356,000	5.913,000	371,000	5.835,000	330,000	5.835,000	375,000	5.922,000
Median	168,000	2.638,000	184,000	2.589,000	170,000	2.638,000	203,000	2.918,000	169,000	2.673,000
Q1	87,000	1.560,000	94,000	1.566,000	86,000	1.535,000	94,000	1.642,000	86,000	1.542,000
Min	1,000	0,000	1,000	0,000	1,000	0,000	1,000	0,000	1,000	0,000
Mode	100,000	1.200,000	205,000	2.541,000	120,000	1.500,000	300,000	2.918,000	80,000	1.300,000
KS	-	-	**0.060**	**0.024**	**0.005**	**0.009**	**0.085**	**0.064**	**0.007**	**0.010**

Table 4 – Preservation of distribution and aggregates, Kolmogorov-Smirnov Index
Variable: **TURNOVER**
Percentage of simulated missing values **5%**

	Original Data		Simple				Stopping & Pruning 1			
			Mean Within Cell		Random Donor		Mean Within Cell		Random Donor	
	Small	Large	Small	Large	Small	Large	Small	Large	Small	Large
Mean	313,158	5.250,788	315,088	5.275,024	312,046	5.242,158	315,080	5.275,405	312,020	5.243,976
STD	1.299,484	8.391,659	1.299,306	8.379,891	1.299,339	8.379,266	1.299,294	8.379,706	1.299,202	8.377,023
Max	56.000,000	113.405,000	56.000,000	113.405,000	56.000,000	113.405,000	56.000,000	113.405,000	56.000,000	113.405,000
Q3	370,000	5.835,000	367,000	5.750,000	370,000	5.750,000	367,000	5.750,000	370,000	5.827,000
Median	168,000	2.638,000	180,000	2.649,000	167,000	2.632,000	180,000	2.651,000	167,000	2.638,000
Q1	87,000	1.560,000	90,000	1.561,000	86,000	1.560,000	90,000	1.561,000	86,000	1.561,000
Min	1,000	0,000	1,000	0,000	1,000	0,000	1,000	0,000	1,000	0,000
Mode	100,000	1.200,000	100,000	1.454,000	100,000	1.000,000	204,000	2.431,000	100,000	1.200,000
KS	-	-	**0.027**	**0.014**	**0.005**	**0.005**	**0.027**	**0.011**	**0.004**	**0.005**

Table 4

	Original Data		Stopping & Pruning 2				Stopping & Pruning 3			
			Mean Within Cell		Random Donor		Mean Within Cell		Random Donor	
	Small	Large	Small	Large	Small	Large	Small	Large	Small	Large
Mean	313,158	5.250,788	315,332	5.286,534	312,191	5.262,092	314,949	5.280,682	311,996	5.236,271
STD	1.299,484	8.391,659	1.299,570	8.413,229	1.299,424	8.520,012	1.297,633	8.373,426	1.299,275	8.375,911
Max	56.000,000	113.405,000	56.000,000	113.405,000	56.000,000	113.405,000	56.000,000	113.405,000	56.000,000	113.405,000
Q3	370,000	5.835,000	367,000	5.750,000	369,000	5.740,000	350,000	5.750,000	369,000	5.750,000
Median	168,000	2.638,000	180,000	2.624,000	167,000	2.624,000	184,000	2.783,000	168,000	2.632,000
Q1	87,000	1.560,000	90,000	1.561,000	86,000	1.550,000	90,000	1.600,000	87,000	1.563,000
Min	1,000	0,000	1,000	0,000	1,000	0,000	1,000	0,000	1,000	0,000
Mode	100,000	1.200,000	204,000	1.454,000	100,000	1.000,000	315,000	2.987,000	100,000	1.300,000
KS	-	-	**0.028**	**0.014**	**0.005**	**0.006**	**0.039**	**0.027**	**0.003**	**0.005**

Table 5 - Preservation of distribution and aggregates, Kolmogorov-Smirnov

Variable: **TURNOVER**

Percentage of simulated missing values: TURNOVER**10%**, CLASSEMP**5%**

	Original Data		Simple				Stopping & Pruning 1			
			Mean Within Cell		Random Donor		Mean Within Cell		Random Donor	
	Small	Large	Small	Large	Small	Large	Small	Large	Small	Large
Mean	313,158	5.250,788	319,181	5.408,554	313,382	5.350,803	319,181	5.410,717	313,382	5.419,693
STD	1.299,484	8.391,659	1.291,346	8.377,239	1.292,534	8.520,492	1.291,346	8.347,023	1.292,534	8.793,853
Max	56.000,000	113.405,000	56.000,000	113.405,000	56.000,000	113.405,000	56.000,000	113.405,000	56.000,000	113.405,000
Q3	370,000	5.835,000	361,000	5.947,000	367,000	5.913,000	361,000	5.835,000	367,000	5.948,000
Median	168,000	2.638,000	191,000	2.847,000	168,000	2.652,000	191,000	2.847,000	168,000	2.706,000
Q1	87,000	1.560,000	97,000	1.564,000	87,000	1.560,000	97,000	1.631,000	87,000	1.560,000
Min	1,000	0,000	2,000	200,000	2,000	200,000	2,000	200,000	2,000	200,000
Mode	100,000	1.200,000	209,000	3.543,000	150,000	1.200,000	209,000	3.543,000	150,000	1.000,000
KS	-	-	**0.063**	**0.041**	**0.005**	**0.011**	**0.063**	**0.041**	**0.005**	**0.010**

Table 5

	Original Data		Stopping & Pruning 2				Stopping & Pruning 3			
			Mean Within Cell		Random Donor		Mean Within Cell		Random Donor	
	Small	Large	Small	Large	Small	Large	Small	Large	Small	Large
Mean	313,158	5.250,788	318,860	5.414,926	312,010	5.337,310	319,849	5.410,268	315,190	5.326,779
STD	1.299,484	8.391,659	1.291,236	8.376,211	1.292,198	8.524,563	1.287,056	8.335,942	1.291,949	8.603,165
Max	56.000,000	113.405,000	56.000,000	113.405,000	56.000,000	113.405,000	56.000,000	113.405,000	56.000,000	113.405,000
Q3	370,000	5.835,000	361,000	5.947,000	367,000	5.913,000	332,000	5.835,000	376,000	5.913,000
Median	168,000	2.638,000	191,000	2.808,000	167,000	2.632,000	204,000	3.127,000	173,000	2.632,000
Q1	87,000	1.560,000	97,000	1.573,000	86,000	1.542,000	97,000	1.639,000	87,000	1.526,000
Min	1,000	0,000	2,000	200,000	2,000	200,000	2,000	200,000	2,000	200,000
Mode	100,000	1.200,000	209,000	3.897,000	80,000	1.000,000	320,000	3.393,000	120,000	1.500,000
KS	-	-	**0.062**	**0.035**	**0.008**	**0.009**	**0.085**	**0.070**	**0.009**	**0.014**

Table 6 – Preservation of relationships between variables (Pearson correlation index)
Variable: **TURNOVER**
Percentage of simulated missing values: 10%

	Dati originali		Simple				Stopping & Pruning 1			
			Mean		Donor		Mean		Donor	
	Small	*Large*	*Small*	*Large*	*Small*	*Large*	*Small*	*Large*	*Small*	*Large*
PURTOT	0,9760	0,9783	0,8097	0,9547	0,7943	0,9437	0,8098	0,9539	0,7941	0,9447
TAXTOT	0,7383	0,6212	0,0569	0,6223	0,0506	0,6159	0,0568	0,6228	0,0505	0,6119
TURNREG	0,1621	0,9660	0,2372	0,9460	0,2360	0,9377	0,2370	0,9461	0,2365	0,9387

Table 6 (*continues*)

	Dati originali		Stopping & Pruning 2				Stopping & Pruning 3			
			Mean		Donor		Mean		Donor	
	Small	*Large*	*Small*	*Large*	*Small*	*Large*	*Small*	*Large*	*Small*	*Large*
PURTOT	0,9760	0,9783	0,8103	0,9548	0,8097	0,9480	0,8085	0,9516	0,7884	0,9396
TAXTOT	0,7383	0,6212	0,0574	0,6224	0,0579	0,6189	0,0551	0,6205	0,0508	0,6071
TURNREG	0,1621	0,9660	0,2372	0,9462	0,2267	0,9420	0,2200	0,9439	0,2151	0,9341

Table 7 – Preservation of relationships between variables (Pearson correlation index)
Variable: **TURNOVER**
Percentage of simulated missing values: 5%

	Dati originali		Simple				Stopping & Pruning 1			
			Mean		Donor		Mean		Donor	
	Small	*Large*	*Small*	*Large*	*Small*	*Large*	*Small*	*Large*	*Small*	*Large*
PURTOT	0,9760	0,9783	0,9750	0,9784	0,9742	0,9768	0,9750	0,9783	0,9744	0,9771
TAXTOT	0,7383	0,6212	0,7384	0,6208	0,7382	0,6212	0,7383	0,6207	0,7381	0,6216
TURNREG	0,1621	0,9660	0,1568	0,9658	0,1542	0,9643	0,1567	0,9657	0,1545	0,9648

Table 7 (*continues*)

	Dati originali		Stopping & Pruning 2				Stopping & Pruning 3			
			Mean		Donor		Mean		Donor	
	Small	*Large*	*Small*	*Large*	*Small*	*Large*	*Small*	*Large*	*Small*	*Large*
PURTOT	0,9760	0,9783	0,9749	0,9773	0,9743	0,9689	0,9747	0,9782	0,9734	0,9772
TAXTOT	0,7383	0,6212	0,7382	0,6182	0,7382	0,6084	0,7386	0,6210	0,7377	0,6217
TURNREG	0,1621	0,9660	0,1572	0,9644	0,1544	0,9556	0,1505	0,9655	0,1495	0,9645

Table 8 – Preservation of relationships between variables (Pearson correlation index)
Variable: **TURNOVER**
Percentage of simulated missing values: TURNOVER 10%, CLASSEMP 5%.

	Dati originali		Simple				Stopping & Pruning 1			
			Mean		Donor		Mean		Donor	
	Small	*Large*	*Small*	*Large*	*Small*	*Large*	*Small*	*Large*	*Small*	*Large*
PURTOT	0,9760	0,9783	0,9658	0,9601	0,9644	0,9466	0,9658	0,9647	0,9644	0,9189
TAXTOT	0,7383	0,6212	0,7420	0,6156	0,7406	0,6071	0,7420	0,6175	0,7406	0,5881
TURNREG	0,1621	0,9660	0,1517	0,9488	0,1517	0,9402	0,1517	0,9531	0,1517	0,9158

Table 8 (*continues*)

	Dati originali		Stopping & Pruning 2				Stopping & Pruning 3			
			Mean		Donor		Mean		Donor	
	Small	*Large*	*Small*	*Large*	*Small*	*Large*	*Small*	*Large*	*Small*	*Large*
PURTOT	0,9760	0,9783	0,9651	0,9599	0,9630	0,9469	0,9647	0,9646	0,9608	0,9414
TAXTOT	0,7383	0,6212	0,7420	0,6154	0,7408	0,6072	0,7426	0,6171	0,7402	0,6037
TURNREG	0,1621	0,9660	0,1512	0,9485	0,1475	0,9404	0,1399	0,9525	0,1407	0,9367

5. **Concluding remarks**

In this paper the performance of some imputation methods, supported by the use of regression trees, has been evaluated through an experimental application on experimental business data. In the application, starting from a complete set of business survey data, different amounts of MCAR item non-responses have been artificially generated on a pre-defined target variable, in order to simulate a raw data set contaminated by missing values: this allows to evaluate the quality of imputations by comparing the original "true" data with the corresponding predicted ones. On the artificial raw data set, different tree-models have been applied by using the algorithms available in the SPSS tool for data mining Clementine 6.5. Among each terminal node of each regression tree, the mean within cell (*mwc*) and random donor within cell (*rdwc*) imputation techniques have been used to compensate for missing data in nodes.

The analysis of results suggests that for the different missing data scenarios all imputation strategies give good results in terms of preservation of the marginal distribution of the target variable.

As expected, by diminishing the percentage of missing values on the target variable, results are remarkably improved because of the reduced risk of introducing distortions on data distributions through imputation. Furthermore, using covariates contaminated by missing values does produce higher biasing effects on results than those obtained in the other scenarios, particularly in terms of distribution mean and variability.

As expected, strategies involving the *rdwc* technique perform slightly better than those using the *mwc* one in terms of preservation of distribution characteristics (like quartiles, maximum and minimum values), mostly because of its stochastic nature. On the other hand, this random nature often implies higher biasing effects on the target variable standard deviation. In terms of preservation of means, the two approaches show low biasing effects in all the simulation scenarios and under all the regression tree models. The analysis concerning the preservation of data relationships between the target variable and some other survey main variables (*Total purchase, Total taxes, Registered turnover*), show that correlation levels are not highly distorted in all the experimented imputation strategies.

Looking at the regression trees algorithms generated in the application, we found that using the *Simple* option it's really easy to create quite *robust* trees. On the other hand, with the

pure *pruning* option more *generalised* trees can be created: many of the splits in the tree growing process were considered "weak splits", i.e. not significantly improving the tree homogeneity, so they were cut out in the pruning process. In our application this fact resulted in a poorer imputation quality than in the *simple* algorithm because the pruned trees often consisted of either a unique node, or too few terminal nodes to be effectively used as imputations cells.

Furthermore, the choice of carrying out two different analyses for *Small* and *Large* businesses has put in evidence the effects produced by the different structure of training data on the tree-growing process. In particular, when data are characterised by a low internal homogeneity (as in the *Large* businesses case) it is more difficult to build any type of tree.

In general, results confirm that regression trees algorithms represent a valid methodological support in the imputation process: the modelling effort seems to be balanced by satisfactory results in terms of univariate and bivariate statistics with all the applied imputation techniques. Further studies are needed particularly in three directions: 1) evaluating the performance of imputation strategies in terms of biasing effects on estimates precision, by measuring the non sampling variance component due to missing data and imputation; 2) assessing the robustness and the performance of these models when more covariates containing missing data are used; 3) applying other imputation techniques (such as regression or nearest-neighbour imputation).

References

AUTIMP (2001). Sito internet del ProgettoAutImp : *http://www.cbs.nl/en/services/autimp/autimp.htm.*

BLOCH, D. A and SEGAL, M.R. (1989). Empirical Comparison of Approaches to Forming Strata using Classification Trees to adjust for Covariates, *Journal of American Statistical Association, Applications & Case Studies,* Vol. 84, n. 408, pp. 897-905.

BREIMAN, L., FRIEDMAN, J.H., OLSHEN, R.A. and STONE C.J. (1984). *Classification and Regression Trees,* Belmont, CA: Wadsworth International.

CHOUP, A. (1991). Optimal partitioning for Classification and Regression Trees, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 13 (4), pp. 340-354.

CHOUP, A., LOOKBAUGH, T. and GRAY, R. M. (1989). Optimal Pruning with Applications to Tree-Structured Source Coding and Modeling, *IEEE Transactions on Information Theory*, 35 (2), pp. 299-315.

CHAMBERS, R.L., MESA , D.M. and TSAI, P. (2000). Using Tree-Based Models for Missing Data Imputation: an evaluation using UK Census Data. Di prossima pubblicazione su *Research Papers of Statistics Netherlands*. Attualmente disponibile sul sito del Progetto AutImp.

CHEN J., SHAO J. (2000). Nearest Neighbor Imputation for Survey Data. *Journal of Official Statistics*, No 16, pp. 113-131

CIAMPI, A. (1991). Generalized Regression Trees, *Computational Statistics & Data Analysis*, n. 12, pp. 57-78.

CIRIANNI A., DI ZIO M., LUZI O., PALMIERI A., SEEBER A.C. (2001). Comparing the effect of different adjustment methods for units with large amounts of item non-response: a case study, *Proceedings of the International Conference on Quality in Official Statistics* (First version), Stockholm, Sweden, May 14-15.

DELLA ROCCA G., DI ZIO M., MANZARI A., LUZI O. (2000). "E.S.S.E. Editing System Standard Evaluation", *Proceedings of the SEUGI 18*, Dublin, June 20-23.

DELLA ROCCA G., LUZI O., SCAVALLI M., SIGNORE M., SIMEONI G. (2003). Documenting and monitoring Editing and Imputation Processes in the Italian Institute of Statistics, Relazione invitata alla *UN/ECE Work Session on Statistical Data Editing*, Madrid, 20-22 ottobre 2003.

DI ZIO M., GUARNERA U., LUZI O., SEEBER A.C. (2003). Model-based and non parametric imputation of item non response in business data: the EUREDIT Project experience, *Atti del Convegno Intermedio della Società Italiana di Statistica* su *Analisi Statistica Multivariata per le Scienze Economico-Sociali le Scienze Naturali e la Tecnologia*, Napoli, 9-11 Giugno.

GELFAND, S.B., RAVISHANKAR, C.S. and DELP, E.J. (1991). An iterative Growing and Pruning Algorithm for Classification Tree Design, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13 (2), pp. 163-174.

GUEGUEN, A., NAKACHE J.P. (1988). Méthode de discrimination basée sur la construction d'un arbre de decision binaire, *Revue de Statistique Appliquèe*, XXXVI (1), pp. 19-38.

GRANDE E., LUZI O. (2003). Metodologie per l'imputazione delle mancate risposte parziali: analisi critica e soluzioni disponibili in ISTAT, *Contributi ISTAT*, N.6/2003.

HAZIZA, D. (2002). Imputation Classes, *The Imputation Bulletin*, Statistics Canada, Vol.2, n. 1.

HYUNJOONG, K. and LOH, W.Y. (2001). Classification Trees with Unbiased Multiway Splits, *Journal of American Statistical Association, Theory and Methods*, Vol. 96, n. 454, pp. 589-604.

KALTON, G. (1986). Handling wave non response in Panel Surveys. *Journal of Official Statistics*, Vol. 2, n.3, pp. 303-314.

KALTON, G. and KASPRZYK, D. (1982). Imputing for Missing Survey Responses. *Proceedings of the Section on Survey Research Methods, American Statistical Association,* pp. 22-31.

KALTON, G. and KASPRZYK, D. (1986). The Treatment of Missing Survey Data. *Survey Methodology,* Vol. 12, n. 1, pp. 1-16.

LAAKSONEN, S. (2000). How to find the best imputation tecnique? Test with various methods, (manca riferimento) pp. 1-14.

LITTLE, R. J. A. (1984). Survey nonresponse adjustments. *Proceedings of the Section on Survey Research Methods, American Statistical Association,* pp. 1-10.

LITTLE, R. J. A. (1986a). Survey nonresponse adjustments for estimates of means. *International Statistical Review,* 54, pp. 139-157.

LITTLE, R. J. A. (1988). Missing-Data Adjustments in Large Surveys. *Journal of Business & Economic Statistics, American Statistical Association,* July, Vol. 6, n. 3.

LITTLE, R. J. A. and RUBIN, D. B. (1987). *Statistical analysis with missing data.* New York: John Wiley & Sons.

LOH, W. Y. and VANICHSETAKUL, N. (1988) Tree-Structured Classification Via Generalized Discriminant Analysis, *Journal of American Statistical Association,* Vol. 83, n. 405, pp. 715-725.

LOWTHIAN, P. (2001) Using WAID to discover implicit edit rules. *Working Paper 4.1*, Euredit. Disponibile sul sito del Progetto AutImp.

LUZI O., SCAVALLI E. (2001). La valutazione della qualità dei metodi per il controllo e l'imputazione dei dati nel progetto europeo EUREDIT, *Atti del Convegno Intermedio della Società Italiana di Statistica* su *Processi e Metodi di Valutazione - Comunicazioni delle Sessioni Spontanee,* Roma, 4-6 giugno.

MICHIE, D., SPIEGELHALTER, D.J. and TAYLOR, C.C. (1994). *Machine learning, neural and statistical classification*, Ellis Horwood Series in Artificial Intelligence.

MINGERS, J. (1989) An Empirical Comparison of Selection Measures for Decision-Tree Induction, *Machine Learning*, 3, pp. 319-342.

PALLARA, A. (1992). Binary decision trees approach to classification: a review of CART and other methods with some applications to real data. *Statistica Applicata*, Vol. 4, n. 3, pp. 255-285.

PIELA, P. and LAAKSONEN, S. (2001). Automatic Interaction Detection for Imputation-Tests with the WAID software package, *Statistic Finland*. Disponibile sul sito del Progetto AutImp.

QUINLAN, J. R. (1986). Induction of Decision Trees, *Machine Learning,* 1, pp. 81-106.

SAFAVIAN, S. R. and LANDGREBE, D. (1991). A Survey of Decision Tree Classifier Methodology, *IEEE Transactions on System Man and Cybernetics,* 21 (3), pp. 660-674.

SCHULTE NORDHOLT, E. (1998). Imputation: Methods, Simulation, Experiments and Practical Examples. *International Statistical Review,* Vol. 66, n. 2, pp. 159-180.

SEGAL, M.R. (1988). Regression Trees for Censored Data, *Biometrics*, 44, pp. 35-47.

SEGAL, M.R. (1992). Tree-Structured methods for longitudinal data, *Journal of American Statistical Association, Theory and Methods,* Vol. 87, n. 418, pp. 407-418.

SPSS (2001a). *Answer Tree 3.0, User's Guide,* SPSS Inc..

SPSS (2001b). *The SPSS C&RT Component,* White paper-technical report, SPSS Inc..

SPSS (2001c). *Clementine 6.5, User's Guide,* SPSS Inc..