Journal of Computational and Applied Mathematics 427 (2023) 115130

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A proposal of a mixed diagnostic system based on decision trees and probabilistic experts rules

Gabriel Aguilera-Venegas [a], Eugenio Roanes-Lozano [b], Gemma Rojo-Martínez [c], José Luis Galán-García [a,*]

[a] Depto. de Matemática Aplicada, Universidad de Málaga, Spain
[b] Instituto de Matemática Interdisciplinar (IMI) & Depto. de Didáctica de las Ciencias Experimentales, Sociales y Matemáticas, Universidad Complutense de Madrid, Spain
[c] UGC Endocrinología y Nutrición. Hospital Regional Universitario de Málaga. CIBERDEM. IBIMA-Plataforma BIONAND, Málaga, Spain

ARTICLE INFO

Article history:
Received 25 October 2022
Received in revised form 6 February 2023

Keywords:
Type 2 diabetes mellitus (T2DM)
Decision tree
Probabilistic rule
Rule-based expert system (RBES)

ABSTRACT

Decision trees and rule-based expert systems (RBES) are standard diagnostic tools. We propose a mixed technique that starts with a probabilistic decision tree where information is obtained from a real world data base. The decision tree is automatically translated into a set of probabilistic rules. Meanwhile a panel of experts proposes their own set of probabilistic rules, according with their experience on the subject. Both sets of rules are combined, generating a mixed RBES with probabilistic rules. The expected probabilities of the rules translating the knowledge in the decision tree are discretized by considering a mapping from intervals of expected probabilities into a set of five values. This way, knowledge coming from real data is completed with the experience of the panel of experts in order to provide a more accurate prediction of suffering from type 2 diabetes mellitus (T2DM) before seven and a half years in the future.

The proposed technique is illustrated with a real case using a diabetes diagnosis probabilistic decision tree built using 1350 out of 1800 real cases and the rules provided by a panel of experts in diabetes. The final result takes into account both the probabilities of the rules and the number of times that each possible consequent is reached, giving a probabilistic result among seven possibilities.

For modeling the decision tree, 75% of the individuals in the database (randomly selected) have been used and the rest (25%) have been used to test the results. The results of the Mixed RBES have been compared with the results of the Tree RBES (the RBES built using only the rules from the decision tree) and the results of the Experts' RBES (the RBES built using only the rules from the panel of experts). The accuracy of the predictions of the Mixed RBES is much better.

1. Introduction

The International Diabetes Federation (IDF) states in the 10th edition of the IDF Diabetes Atlas [1] that "537 million people worldwide suffer from diabetes. This number is projected to reach 643 million by 2030, and 783 million by 2045". This "edition confirms that diabetes is one of the fastest growing global health emergencies of the 21st century".

* Corresponding author.
E-mail addresses: gaguilera@uma.es (G. Aguilera-Venegas), eroanes@ucm.es (E. Roanes-Lozano), gemma.rojo.m@gmail.com (G. Rojo-Martínez), jlgalan@uma.es (J.L. Galán-García).

https://doi.org/10.1016/j.cam.2023.115130

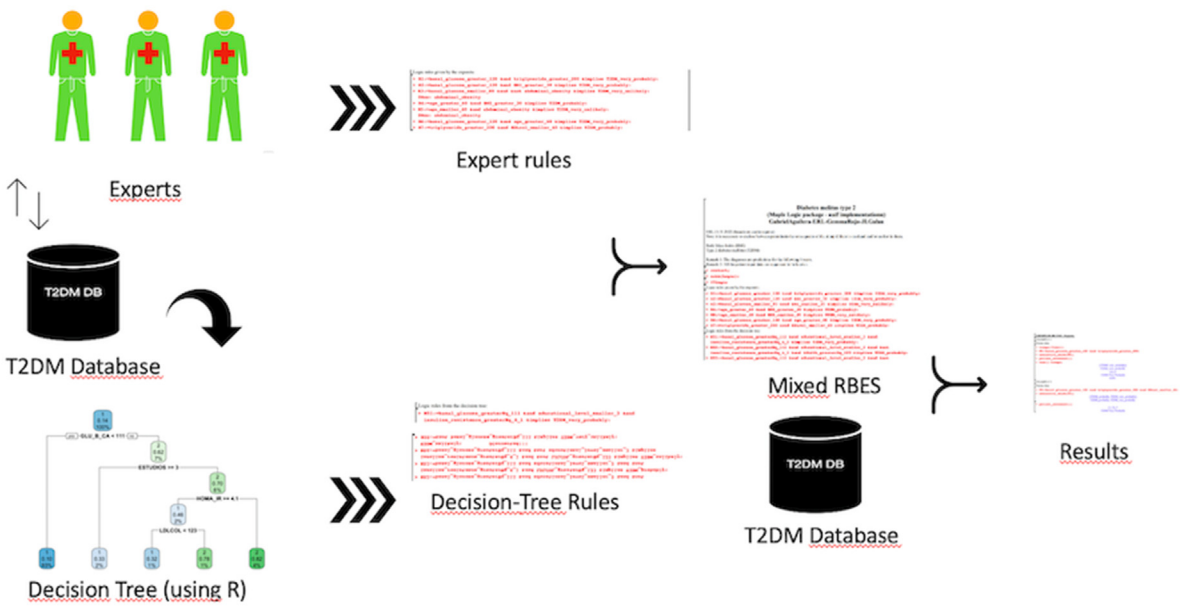

Fig. 1. Scheme of the work.

In [2] we can find: "Type 2 diabetes mellitus is one of the most serious health problems of our time. Data from numerous studies agree on the continuous growth of its incidence and prevalence rates throughout the world, with unacceptable human, social, and economic costs. Diabetes has become one of the main causes of cardiovascular disease, blindness, non-traumatic amputations of the lower limbs, kidney failure, and death throughout the world. In addition, its association with the presence of cancer has recently been demonstrated".

In light of this data, the importance of prevention from suffering type 2 diabetes mellitus (T2DM) can hardly be overrated.

The goal of this paper is the prediction of suffering from T2DM less than seven and a half years in the future, taking as data several (social and medical) variables from each individual. It uses a mixed RBES which probabilistic rules are obtained, both from a Decision Tree derived from the T2DM Database and a set of probabilistic rules developed by a panel of experts. The results of the RBES take into account both, the probabilities of the rules and the number of times each consequent of the rules is reached. The final result is the probability of suffering T2DM seven and a half years in the future. The probabilities of the rules are discretized in a set of 5 values and the probability of the result is discretized in a set of 7 values.

For managing the probability of the rules, several probabilistic logics can be applied, for example the probabilistic logic described in [3]. This is a rewriting logic and, in this particular case, directly applying one rule in this logic, since the antecedent of the rules are stated as true (probability 1), the probability of the rule is transferred to the consequent.

A scheme of the work is shown in Fig. 1.

1.1. T2DM database

The Di@bet.es Study was the first national study in Spain to examine the prevalence of diabetes and impaired glucose regulation [4]. Later on, the incidence of type 2 diabetes mellitus in a nation-wide population based cohort from Spain was determined in di@bet.es study [2]. More than five thousands adults (older than 18 years) were randomly selected all over Spain. Of these, 1800 completed the follow-up study with at least one fasting blood glucose data. This study was reviewed and approved by the Ethics and Clinical Investigation Committee of Málaga and written informed consent was obtained from all subjects.

The variables considered in the T2DM Database are:

- Age (years)
- Serum high sensitivity C-reactive protein (mg/dL).
- Serum uric acid (mg/dL).
- Serum HDL-Cholesterol (mg/dL).

Name of the procedure: ***Decision_tree***

Input

- Needed packages: rpart and rpart.plot
- Database
- Percentage of training (and test)
- Seed for the pseudo-random number generation

Algorithm

1. Install needed packages (rpart and rpart.plot)
2. Set the seed for the pseudo-random sample generation
3. Read the database in a dataframe
4. Generate as many pseudo-random samples of a Continuous Uniform [0,1] distribution as individuals in the dataframe
5. Assign a pseudo-random number to each individual in the dataframe
6. Assign individuals which random number is less than 0.75 to the Training Set and the rest to the Test Set.
7. Generate the model of the decision tree
8. Print the decision tree

Output (detail)

- The model of the decision tree generated

Fig. 2. Pseudocode of procedure *Decision_tree*.

- LDLCOL Serum LDL-Cholesterol (mg/dL).
- Serum creatinine (mg/dL).
- Serum triglycerides (mg/dL).
- Insulin resistance: Insulin resistance index (obtained by the homeostatic model method)
- Basal glucose: Serum glucose (mg/dL).
- Sex (male or female).
- Sampling spanish region
- Educational level
- Smoking
- Familiar history of diabetes
- High blood pressure presence
- Obesity presence
- Abdominal obesity presence
- T2DM (DM2_s): Type 2 diabetes after 7.5 years of follow up.

1.2. The Decision Tree

The Decision Tree technique has been previously applied to diabetes diagnosis, for example in [5]. General Artificial Intelligence techniques have been applied to the same goal, for example in [6].

In this new article the decision tree has been obtained from the database using *R* language [7]. The *R* packages `rpart` and `rpart.plot` [8] have been used too.

The set of individuals in the database has been randomly split into two subsets: the training subset and the test subset (containing approximately the 75% and the 25% of the individuals respectively)

The *R* pseudocode and flowchart describing how the decision tree is obtained from the training subset can be found in Figs. 2 and 3 (the complete *R* code can be found in Appendix A):

The decision tree obtained is shown in Fig. 4 and will be explained in Section 3.

1.3. RBES generalities

The underlying logic chosen is classic Boolean logic. As said above, two sets of rules are considered:

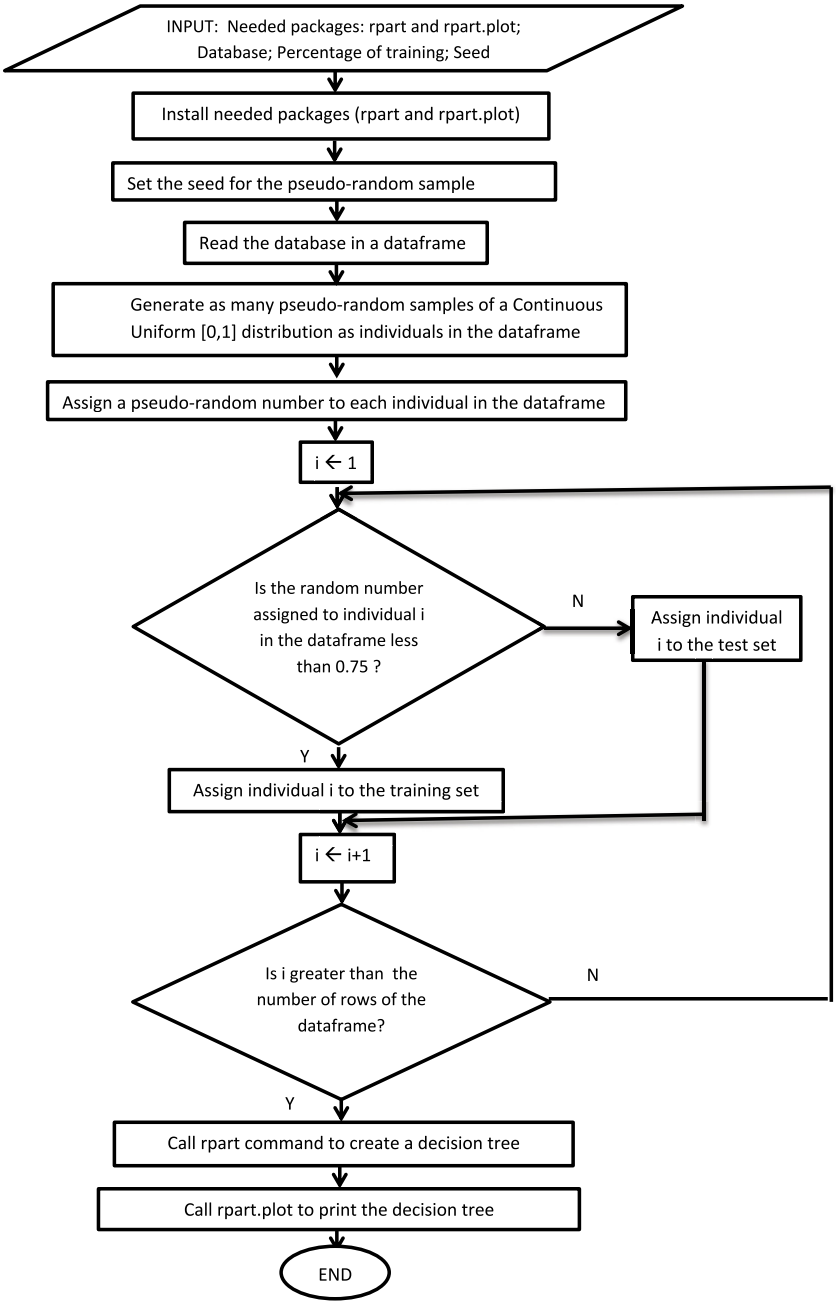

Fig. 3. Flowchart of *Decision_tree*.

- one is obtained from the knowledge of the experts in the panel (itself derived from their experience, practice guidelines, scientific articles, …)
- the other one synthesizes the information in the decision tree.

2. Probabilistic rules obtained from the knowledge of the experts in the panel (in logic notation)

The rules obtained from the knowledge of the experts in the panel were translated from natural language into Boolean logic rules. They are the following:

$R1: basal\_glucose\_greater\_120 \wedge triglycerids\_greater\_200 \rightarrow T2DM\_very\_probably$
$R2: basal\_glucose\_greater\_120 \wedge BMI\_greater\_30 \rightarrow T2DM\_very\_probably$
$R3: basal\_glucose\_smaller\_85 \wedge \neg abdominal\_obesity \rightarrow T2DM\_very\_unlikely$

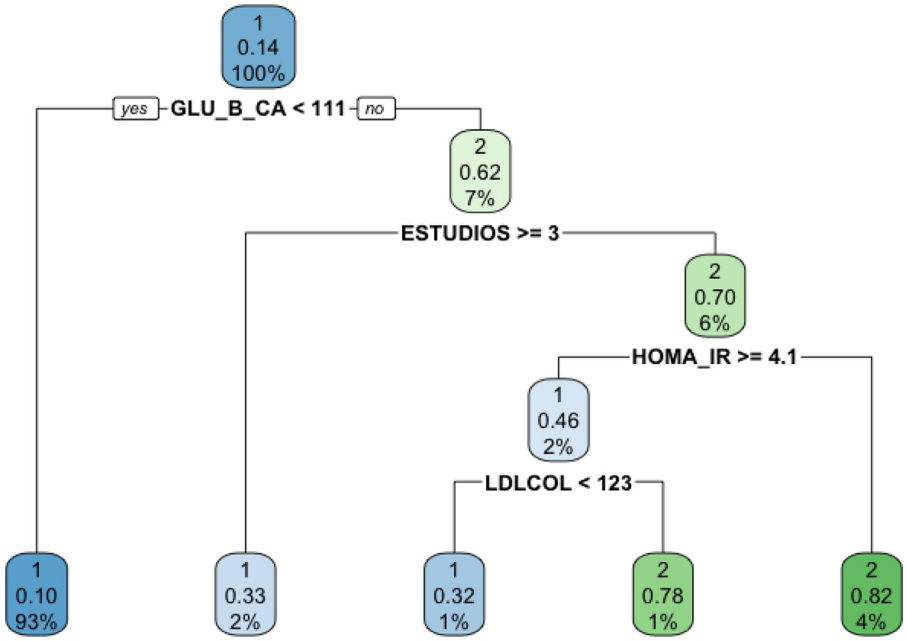

Fig. 4. Decision Tree obtained from the T2DM Database.

$R4$: $age\_greater\_60 \wedge BMI\_greater\_30 \rightarrow T2DM\_probably$
$R5$: $age\_smaller\_40 \wedge \neg abdominal\_obesity \rightarrow T2DM\_very\_unlikely$
$R6$: $basal\_glucose\_greater\_120 \wedge age\_greater\_60 \rightarrow T2DM\_very\_probably$
$R7$: $triglycerids\_greater\_200 \wedge HDLcol\_smaller\_40 \rightarrow T2DM\_probably$

where $\wedge$, $\vee$, $\neg$ and $\rightarrow$ represent "and", "or", "not" and "implies" respectively, as usual in Boolean logic.

We have kept the names used for the variables in the implementation, as they are clearly understandable. For instance:

$basal\_glucose\_greaterEq\_111$

means "the basal glucose of this patient is $\geq$ 111" because this way it is easier to relate the rule in logic notation with the code.

According to the MD's knowledge, these diagnoses are the predictions of suffering from T2DM seven and a half years in the future.

2.1. Integrity constraints added by the designers of the RBES (in logic notation)

In order to check if contradictory data exist, the following integrity constraints have been added by the designers of the RBES:

$R201$: $(basal\_glucose\_greaterEq\_111 \vee basal\_glucose\_greaterEq\_120) \wedge$
$basal\_glucose\_smaller\_85 \rightarrow contradictory\_input\_data$
$R202$: $age\_smaller\_40 \wedge age\_greater\_60 \rightarrow contradictory\_input\_data$

Observe that these are purely logical contradictions.

3. Probabilistic rules from the decision tree obtained with *R* (in logic notation)

The decision tree of Fig. 4 is obtained using the corresponding package of *R* software environment and the real data of 1350 patients $= \frac{3}{4} \times 1800$ patients (the "training subset").

The following rules summarize the information contained in the decision tree:

$R51$: $basal\_glucose\_greaterEq\_111 \wedge educational\_level\_smaller\_3 \wedge$
$insuline\_resistence\_greaterEq\_4\_1 \rightarrow T2DM\_very\_probably$
$R52$: $basal\_glucose\_greaterEq\_111 \wedge educational\_level\_smaller\_3 \wedge$
$\neg insuline\_resistence\_greaterEq\_4\_1 \wedge LDLCOL\_greaterEq\_123 \rightarrow$
$T2DM\_probably$

Name of the procedure: `exhaustive_check`

Input

- Logic rules given by the experts.
- Logic rules derived from the decision tree.
- Integrity constraints (IC).
- Patients data.

Algorithm (no concatenation of rules is applied!)

1. If contradictory input data exist (from the patients data and the IC) then return a warning with information of the IC fired (and stop).
2. Construct the list *diagnoses_list* obtained by forward firing the rules, given a set of data.
3. Let *diagnoses_set* be the set of diagnoses (repetitions are eliminated).
4. If contradictory diagnoses can be found in *diagnoses_set* then include a warning in this set.
5. If *diagnoses_set* is the empty set, include in it *not_classified*.
6. Print *diagnoses_list*.
7. Return the elements of *diagnoses_set* set.

Output (detail)

- It prints a list of diagnoses (possibly with repetitions): *T2DM_very_probably*, *T2DM_probably*, *T2DM_unlikely*, *T2DM_very_unlikely*.
- It returns a finite sequence (without repetitions) of the diagnoses: *T2DM_very_probably*, *T2DM_probably*, *T2DM_unlikely*, *T2DM_very_unlikely*.

Fig. 5. Pseudocode of procedure *exhaustive_check*.

$$
\begin{aligned}
R53:\ & basal\_glucose\_greaterEq\_111 \wedge educational\_level\_smaller\_3 \wedge \\
& \neg insuline\_resistence\_greaterEq\_4\_1 \wedge \neg LDLCOL\_greaterEq\_123 \rightarrow \\
& T2DM\_unlikely \\
R54:\ & basal\_glucose\_greaterEq\_111 \wedge \neg educational\_level\_smaller\_3 \rightarrow T2DM\_unlikely \\
R55:\ & \neg basal\_glucose\_greaterEq\_111 \rightarrow T2DM\_very\_unlikely :
\end{aligned}
$$

4. RBES design

In a usual medical rule-based expert system the goal is to check whether a diagnosis is reached by firing forward a set of (interconnected) rules or not. Usually the rules can concatenate with other several times.

However, in this case we do not want to know whether a diagnosis is reached or not, we would like to know how many times the different diagnoses are reached by forward firing a set of independent rules.

The rule-base expert system is therefore somehow trivial from the logical point of view. Nevertheless, it is very important to have a computerized approach to it, as it is going to be applied to the data of hundreds of patients.

We have considered two procedures, that we have denoted `exhaustive_check` and `patient_assessment`. The pseudocodes and flowcharts of these procedures can be found in Figs. 5–8.

A tiresome consequence of the restriction for the concatenation of rules in data introduction is the following: if a datum implies another datum, we cannot add a rule in order to only introduce the more strict datum. For instance if *basal glucose* $\geq$ 120 holds and there are two variables: *basal glucose* $\geq$ 111 and *basal glucose* $\geq$ 120, both *basal glucose* $\geq$ 120 and *basal glucose* $\geq$ 111 have to be stated as true.

All the patient input data are supposed to be known (logical variables or their negations), otherwise it could happened that no diagnosis was reached.

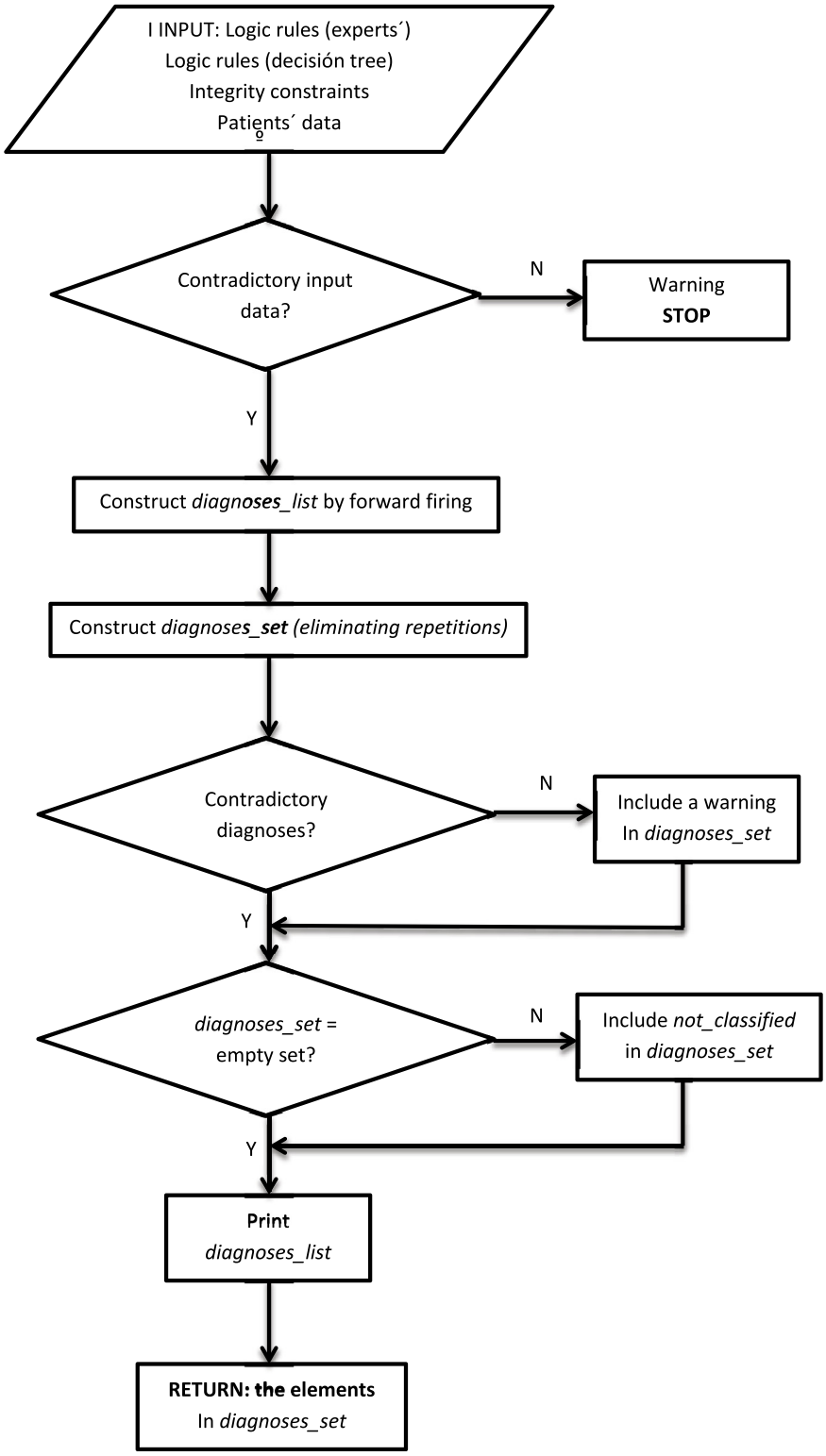

Fig. 6. Flowchart of procedure *exhaustive_check*.

Name of the procedure: `patient_assesment` (always to be executed after `exhaustive_check` procedure).

Input

- *diagnoses_list*.

Algorithm (no concatenation of rules is applied!)

1. Assign values in $\{-2,-1,0,,1,2\}$ to the elements of *diagnoses_list*, from $-2 \leftarrow T2DM\_very\_unlikely$ to $2 \leftarrow T2DM\_very\_probably$).
2. Sum the values of the list of Step 1 (repetitions included).
3. Print the list of the Step 1 and the value obtained in the Step 2.
4. According to the sum of Step 2 (≥ 4, ≥ 2, ≥ 1, > -1 , > -2, > -4, else), print one of 7 possible assessments, from *T2DM Extremely Probably* to *T2DM Extremely Unlikely*.

Output (detail)

- It prints a list of numerical values and its sum, corresponding and summarizing, respectively, the diagnoses found.
- It prints the final diagnosis, that is, one of: *T2DM Extremely Probably*, *T2DM Very Probably*, *T2DM Probably*, *TD2M with Low Probability*, *T2DM Unlikely*, *T2DM Very Unlikely*, *T2DM Extremely Unlikely*.

Fig. 7. Pseudocode of procedure *patient_assessment*.

5. Running the *maple* implementation of the RBES: some examples

The implementation of this RBES uses *Maple*'s "Logic Package" and is similar to the one used in the equine cardio diagnosis RBES [9], although the inconsistency checking has been developed. The complete code can be found in Appendix B.

It is convenient to reset the *Maple* session. Afterwards, *Maple*'s logic package, the code, allocated in file `MixedRBES.mpl`, and the `rules` and `integrity constraints`, allocated in file `MixedRBES_Rules.mpl`, have to be loaded.

```
restart;
with(Logic):
read(‘C:/.../MixedRBES.mpl‘):
%read(‘C:/.../MixedRBES_Rules.mpl‘):
```

The next step is to load the rules and integrity constraints of the RBES and to collect them in two sets (`ERS` and `IC`, respectively). The corresponding *Maple* code is located in the file `Rules_MixedRBES.mpl`.

```
read(‘C:/.../Rules_MixedRBES.mpl‘):
```

Then the system is ready.

The *Maple* code in file `Rules_MixedRBES.mpl` can be found in Appendix C.

Example 1. Let us begin with an example where contradictory data are included:

$$age\_greater\_60 \wedge age\_smaller\_40 \wedge basal\_glucose\_greater\_120$$

Let us check it with *Maple*:

```
PT:=age_greater_60 &and age_smaller_40 &and basal_glucose_greater_120:
exhaustive_check(PT);
```

and the output is.

$$(age\_smaller\_40 \wedge age\_greater\_60) \Rightarrow contradictory\_input\_data$$

(clarifying where the problem is).

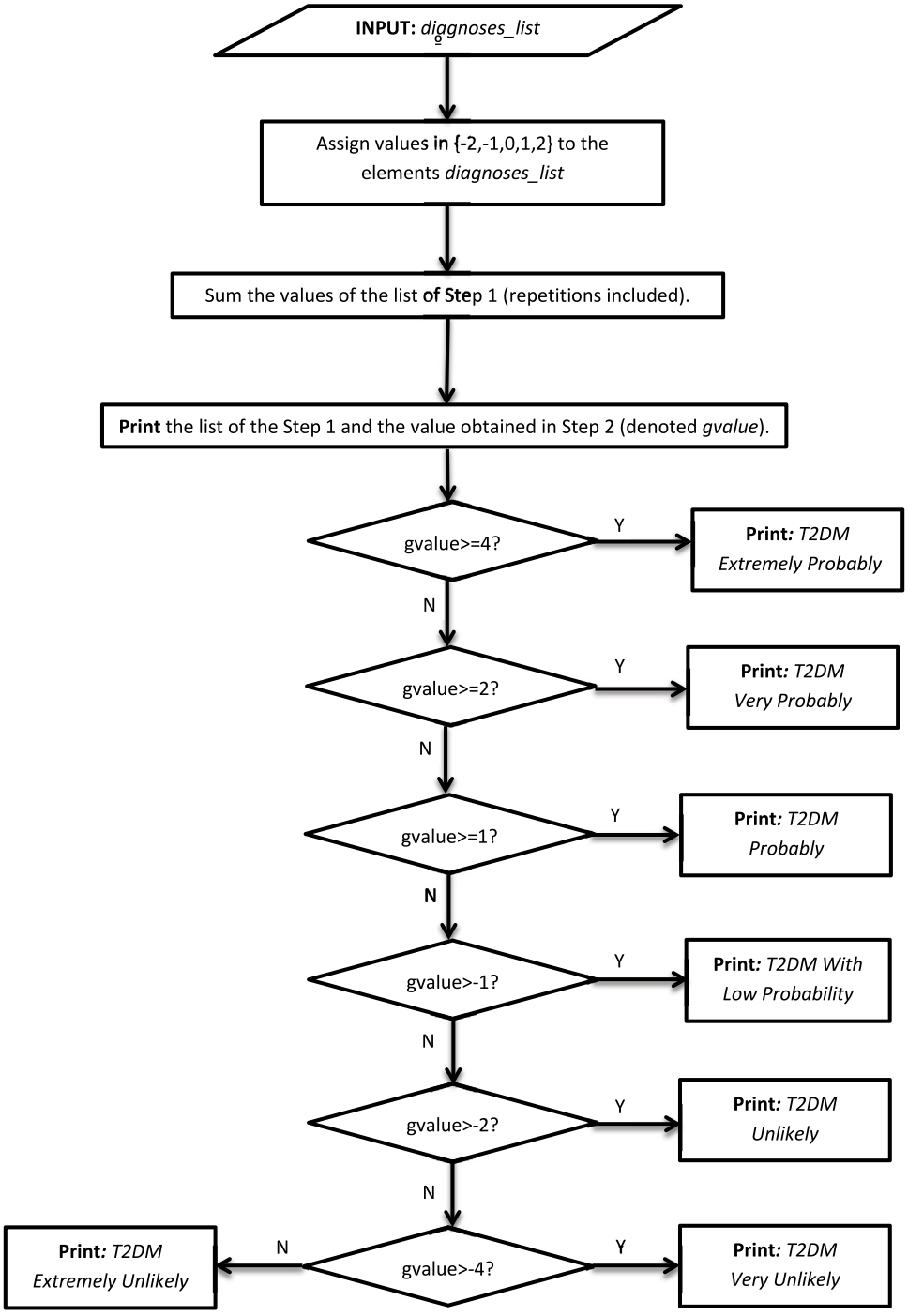

Fig. 8. Flowchart of procedure *exhaustive_check*.

Example 2. Let us continue with a very simple example. The data of a patient are:

*basal_glucose_greater*_120 $\wedge$ *triglycerids_greater*_200 .

Then rule $R1$ is fired, but no more rules can be fired, so only one diagnosis is obtained (*T2DM_Very_Probably*). Let us check it with *Maple*:

```
PT:=basal_glucose_greater_120 &and triglycerids_greater_200:
```

```
  exhaustive_check(PT);
             [T2DM_very_probably]
              T2DM_very_probably
```

That is, *T2DM_very_probably* is initially obtained by `exhaustive_check(PT)` procedure.
Now we can run `patient_assessment()` procedure:

```
patient_assessment();
                  [2], 2
               T2DM Very Probably
```

That is, `patient_assessment()` procedure returns as final diagnosis *T2DM Very Probably*, that corresponds to the numerical value 2.

The computation time is less than two hundredths of a second on a standard 8 GB RAM computer running *Maple 2022*.

Example 3. Let us continue with an extreme example. The data of a patient are:

*basal_glucose_greater*_120 $\wedge$ *triglycerids_greater*_200 $\wedge$

*age_greater*_60 $\wedge$ $\neg$*abdominal_obesity* $\wedge$ *BMI_greater*_30

Let us check it with *Maple*:

```
PT:=basal_glucose_greater_120 &and triglycerids_greater_200 &and
  age_greater_60 &and &not abdominal_obesity and BMI_greater_30:
exhaustive_check(PT);
         [T2DM_probably, T2DM_very_probably,
        T2DM_very_probably,T2DM_very_probably]
         T2DM_probably, T2DM_very_probably
```

That is, *T2DM_probably* is obtained once and *T2DM_very_probably* is obtained 3 times by `exhaustive_check(PT)` procedure.

Now we can run `patient_assessment()` procedure:

```
patient_assessment();
                [1, 2, 2, 2], 7
              T2DM Extremely Probably
```

That is, `patient_assessment()` procedure returns $[1, 2, 2, 2], 7$ (the 7 comes from $1 + 2 + 2 + 2 = 7$) and *T2DM Extremely Probably* as final diagnosis. The computation time is less than two hundredths of a second on a standard 8 GB RAM computer running *Maple 2022*.

Example 4. As final example we will show how to apply the RBES to a massive set of patients data (450), stored in file `AllPatients_new.txt`. First of all, the content of the file has to be loaded.

```
read(‘C:/.../AllPatients_new.txt‘):
```

The following lines of code run the two procedures for all the patients and precede each output with the number of the patient:

```
for i to 450 do
print(‘----------‘,’PT’,i,‘----------‘);
exhaustive_check(PT||i);
patient_assessment();
print(‘--------------------‘);
end do;
```

The output begins as follows:

```
          ----------, PT, 1, ----------
            [T2DM_very_unlikely]
            T2DM_very_unlikely
               [-2], -2
            T2DM Very Unlikely
             --------------------
```

Table 1
Probabilistic classification for each RBES.

	Extremely unlikely			Very unlikely			Low probability			Medium probability			Probably			Very probably			Extremely probably		
	T	+	%	T	+	%	T	+	%	T	+	%	T	+	%	T	+	%	T	+	%
Ref. Prob.			1			10			20			50			80			90			99
Mixed RBES	135	2	1.48	226	16	7.08	55	18	32.73	8	5	62.5	8	8	100	14	8	57.14	4	4	100
Tree RBES	0	0	–	402	27	6.71	30	22	73.33	0	0	–	6	4	66.67	12	8	66.67	0	0	–
Experts' RBES	29	0	0	106	2	1.89	4	0	0	0	0	–	50	17	34	11	8	72.73	2	2	100

```
          ----------, PT, 2, ----------
       [T2DM_very_unlikely, T2DM_very_unlikely]
            T2DM_very_unlikely
              [-2, -2], -4
           T2DM Extremely Unlikely
             --------------------
          ----------, PT, 3, ----------
[T2DM_very_unlikely, T2DM_very_unlikely, T2DM_very_unlikely]
            T2DM_very_unlikely
              [-2, -2, -2], -6
           T2DM Extremely Unlikely
             --------------------
                  ...
                  ...
```

The total computation time is smaller than 18 seconds on a standard 8 GB RAM computer running *Maple 2022*. As there are 450 patients, the average computation time per patient is approximately 4 hundredths of a second. Timings are even better with the mute version mentioned above, as no time is wasted presenting the output on the screen.

6. Results

In order to compare the Mixed RBES with

- the RBES obtained using only the rules from the decision tree ("Tree RBES"), and
- the RBES obtained using only the rules from the panel of experts ("Experts' RBES"),

these two later versions of the RBES have been developed and implemented in *Maple*.

The three resulting RBES have been run with the data of the individuals in the "test subset", consisting of one fourth of the individuals in the database (those who are not in the training subset), that is, $\frac{1}{4} \times 1800$ individuals $= 450$ individuals.

A summary of the results obtained for the three RBES applied to the testing individuals are shown in Table 1. For each of the seven intervals of probability considered, 3 columns are shown:

1. The first column (labeled with a "T") includes the number of individuals of the test subset that each RBES has classified in this column.
2. The second column (labeled with a "+") includes how many of the individuals predicted to suffer T2DM with that interval of probability by each RBES have really been diagnosed with T2DM after 7.5 years of follow up.
3. The third column (labeled with a "%") is the percentage of individuals in this column and in this row that have really been diagnosed with T2DM. This percentage is only shown when it is possible to calculate it (it is not possible when the denominator is 0), in other case, a "-" is shown.

The calculation of the Mean Squared Error (MSE) for each of the three RBES is obtained using the following steps:

1. For each of the seven probability intervals, the number of individuals classified within this interval are divided by the total number of individual and this proportion is the observed proportion.
2. For each of the seven probability intervals a theoretical probability is assigned. The theoretical probabilities are: 0.01,0.1,0.2,0.5,0.8,0.9,0.99 and are associated to the intervals: Extremely Unlikely, Very Unlikely, Low Probability, Medium Probability, Very Probably, Extremely Probably.
3. For each of the seven probability intervals the difference between the theoretical probability and the observed probability is squared and the squared root of the pondered sum of all of them are divided by 7 (the number of intervals).

Table 2
Errors for each RBES.

	Not classified error	MSE
Mixed RBES	0 (0%)	0.388%
Tree RBES	0 (0%)	0.693%
Experts' RBES	$\frac{450-248}{450}$ (44.89%)	0.762%

The formula used is:

$$MSE = \frac{\sqrt{\sum_{i=1}^{7} n_i * (P_o[i] - P_t[i])^2}}{N}$$

where:

- n_i is the number of observed individuals in interval i
- $P_o[i]$ is the observed probability of interval i.
- $P_t[i]$ is the theoretical probability of interval i.
- N is the total number of individuals.

In Table 2 the errors corresponding to *Not_classified* patients and Mean Squared Error (MSE) for each RBES are shown:

Please, note that the Experts' RBES is not an exhaustive system of rules and therefore some individuals have been not classified (55.11%). In this group of not diagnosed individuals, 12.9% will suffer from T2DM. However, both the Tree RBES and the Mixed RBES are exhaustive systems of rules and therefore all the individuals are classified in one of the probabilistic groups.

We can observe that the MSE is similar for the Tree RBES and the Experts' RBES (0.00693 and 0.00762), however for the Mixed RBES is almost a half (0.388).

Our idea of combining the experts' rules with the rules obtained from a decision tree not only has the advantage of being exhaustive but has fewer errors than the two RBES obtained without combining the two sets of rules.

7. Explainable artificial intelligence

One frequent criticism to AI is the lack of justification for the decisions that are made or the recommendations given. This is for instance the case when a neural network is trained with thousands of examples.

Our mixed proposal obtains the rules from two sources (a decision tree and the knowledge of experts in the field). We have chosen not to organize the RBES using rules that can be concatenated but using standalone rules instead. In this way, the RBES is much simpler, and how many rules that conclude each of the partial diagnoses considered have been triggered can be counted, thus justifying the final diagnosis (see, for instance, Example 3).

In case there is an input data contradiction, the system details the IC violated (see Example 1).

8. Conclusions

A decision tree model has been obtained using *R* language and `rpart` and `rpart.plot` packages with a set of 1350 individuals (75% of the individuals in the T2DM database). The leaves of the tree have been translated to probabilistic rules, and a RBES has been obtained using the probabilistic rules from the decision tree (Tree RBES).

Independently, a set of probabilistic rules has been developed by a panel of experts, and a RBES has been obtained using the probabilistic rules of the experts (Experts' RBES).

A mixed RBES has been obtained combining the rules from the decision tree and from the panel of experts (Mixed RBES).

The three RBES have been compared using the test subset (consisting of 25% of the individuals in the T2DM database, those who were not used for obtaining the decision tree).

The results of the Mixed RBES have been better than the results of the other RBES in two ways:

- No *Not_classified* results have been obtained (it is not the case of the Experts' RBES).
- The Mean Squared Error (MSE) of the classification of the probability of suffering T2DM in less than seven and a half years in the future for the Mixed RBES is lower than the MSE for the Tree RBES and for the Experts' RBES (almost one half).

The obtained results confirm our two previous ideas:

- It is easy for the experts to give probabilistic rules without giving an exact probability but giving natural language intervals and not requiring a comprehensive set of rules, since the rules obtained from the decision tree assure their completeness.

- Combining the rules from the experts and the rules from the decision tree gives better results than using only the rules from the experts or only the rules from the decision tree.

A future extension of this work could be to explain more in detail why the partial diagnoses are obtained, specifying the rules fired and their origin (from the decision tree or from the experts). This could help in improving the rules based on the knowledge of the experts in case a strange partial diagnosis (coming from the experts' knowledge) was found.

Data availability

Data will be made available on request.

Acknowledgments

This work was partially supported by the research projects PGC2018-096509-B-I00 and PID2021-122905NB-C21 (Government of Spain).

This work was partially supported by the Ministerio de Sanidad, Servicios Sociales e Igualdad-ISCIII, Instituto de Salud Carlos III (PI20/01322), European Regional Development Fund (ERDF) "A way to build Europe".

Funding for open access charge: Universidad de Málaga / CBUA.

We are very grateful to the reviewers for their comments and suggestions, that have helped to improve the paper.

Appendix A. *R* code

The *R* pseudocode for obtaining the decision tree from the training subset is:

```
# Installing needed packages
install.packages("rpart","rpart.plot")
library(rpart)
library(rpart.plot)
# Reading the database
diabetes=read.csv("simple.csv", header=TRUE, sep=";")
# Randomly splitting the database in two subsets:
# Training (approximately 75 set.seed(19354) # Random number
partition=runif(nrow(diabetes))
Training=diabetes[partition<0.75,]
Test=diabetes[partition>=0.75,]
#Using rpart command to obtain a model of the decision tree
tree=rpart(DM2_s~. , data=Training, method="class")
rpart.plot(tree)
```

Appendix B. *Maple* code

The code of the exhaustive search procedure is:

```
exhaustive_check:=proc(patient_data)
 local vars_conseq,vars_conseq_IC,i,j,k,res;
 global diagnoses_set,diagnoses_list;
  diagnoses_set:={};
  diagnoses_list:=[];
  vars_conseq:=map(vars_con,ERS);
  vars_conseq_IC:=map(vars_con,IC);
                  #Check if contradictory input data exist
  for i in IC do
    res := Implies( (i &and patient_data) , op(vars_conseq_IC) );
    if res=true then RETURN(i) end if;
   end do;
                  #List of multiple diagnoses (independent rules)
  for j in vars_conseq do
    for k in ERS do
      if Implies(k &and patient_data,j)
        then diagnoses_list:=[op(diagnoses_list),j]; end if;
     end do;
   end do;
```

```
                #This line substitutes "Look for diagnoses"
 diagnoses_set:={op(diagnoses_list)}:
                #Check if contradictory diagnoses arise
 if (member(T2DM_very_probably,diagnoses_set)
     or member(T2DM_probably,diagnoses_set))
   and (member(T2DM_very_unlikely,diagnoses_set)
     or member(T2DM_unlikely,diagnoses_set))
   then diagnoses_set:=diagnoses_set union {conflicting_diagnoses} end if;
                #Check if empty diagnoses_set
 if diagnoses_set={} then diagnoses_set:={Not_classified} end if;
 print(diagnoses_list);
 op(diagnoses_set);
end proc:
```

And the code of the procedure patient global assessment, that returns a numerical valuation of the knowledge extraction (7 possibilities): T2DM Extremely Probably, T2DM Very Probably, T2DM with Low Probability, T2DM Unlikely, T2DM Very Unlikely, T2DM Extremely Unlikely; is:

```
patient_assessment:=proc()
 global diagnoses_list,values,gvalue;
 values:=diagnoses_list;
 values:=subs(T2DM_very_probably=2, values);
 values:=subs(T2DM_probably=1,    values);
 values:=subs(T2DM_unlikely=-1,   values);
 values:=subs(T2DM_very_unlikely=-2,values);
 if values=[] then RETURN(values) end if;
 gvalue:=add(values);
 print(values,gvalue);
 if gvalue>=4 then print(‘T2DM Extremely Probably‘)
   elif gvalue>=2 then print(‘T2DM Very Probably‘)
      elif gvalue>=1 then print(‘T2DM Probably‘)
         elif gvalue>-1 then print(‘T2DM with Low Probability‘)
            elif gvalue>-2 then print(‘T2DM Unlikely‘)
               elif gvalue>-4 then print(‘T2DM Very Unlikely‘)
                  else print(‘T2DM Extremely Unlikely‘)
  end if;
end proc:
```

Let us mention that there is an alternative version for procedure exhaustive_check, denoted `exhaustive_check_Mute` that just returns the numerical output between -3 and 3 instead. It is not included for the sake of brevity.

The following auxiliary functions extract the first and the second elements of a list:

```
first:=R->op(1,R):
second:=R->op(2,R):
```

and the following ones obtain the propositional variables in the antecedents and the consequents of the rules:

```
vars_ant:=R->op(indets(Export(first(R),form=MOD2))):
vars_con:=R->op(indets(Export(second(R),form=MOD2))):
```

No more *Maple* source code is needed.

Appendix C. Translation into *Maple* code of the logic rules in Sections 2 and 3

The *Maple* code corresponding to the logic rules in Sections 2 and 3 and defining the sets `ERS` and `IC` can be found afterwards (the code is stored in file `Rules_MixedRBES.mpl`, as said in Section 5):

```
R1:=basal_glucose_greater_120 &and triglycerids_greater_200 &implies
  T2DM_very_probably:
R2:=basal_glucose_greater_120 &and BMI_greater_30 &implies
  T2DM_very_probably:
R3:=basal_glucose_smaller_85 &and &not abdominal_obesity &implies
  T2DM_very_unlikely:
```

```
R4:=age_greater_60 &and BMI_greater_30 &implies T2DM_probably:
R5:=age_smaller_40 &and &not abdominal_obesity &implies
  T2DM_very_unlikely:
R6:=basal_glucose_greater_120 &and age_greater_60 &implies
  T2DM_very_probably:
R7:=triglycerids_greater_200 &and HDLcol_smaller_40 &implies
  T2DM_probably:

R51:=basal_glucose_greaterEq_111 &and educational_level_smaller_3 &and
   insuline_resistence_greaterEq_4_1 &implies T2DM_very_probably:
R52:=basal_glucose_greaterEq_111 &and educational_level_smaller_3 &and
   &not insuline_resistence_greaterEq_4_1 &and LDLCOL_greaterEq_123
   &implies T2DM_probably:
R53:=basal_glucose_greaterEq_111 &and educational_level_smaller_3 &and
   &not insuline_resistence_greaterEq_4_1 &and
   &not LDLCOL_greaterEq_123 &implies T2DM_unlikely:
R54:=basal_glucose_greaterEq_111 &and &not educational_level_smaller_3
   &implies T2DM_unlikely:
R55:=&not basal_glucose_greaterEq_111 &implies T2DM_very_unlikely:

R201:=(basal_glucose_greaterEq_111 &or basal_glucose_greaterEq_120)
    &and basal_glucose_smaller_85 &implies contradictory_input_data:
R202:=age_smaller_40 &and age_greater_60 &implies
    contradictory_input_data:

ERS:={R1,R2,R3,R4,R5,R6,R7,R51,R52,R53,R54,R55}:
IC:={R201,R202}:
```

References

[1] International Diabetes Federation, IDF Diabetes Atlas, 10th ed., Brussels, Belgium, 2021, Available online: https://www.diabetesatlas.org. (Accessed 24 October 2022).

[2] G. Rojo-Martínez, et al., Incidence of diabetes mellitus in Spain as results of the nation-wide cohort di@bet.es study, Sci. Rep. 10 (1) (2020) 2765, http://dx.doi.org/10.1038/s41598-020-59643-7.

[3] I. Fortes, M.A. Galán, G. Aguilera, A. Burrieza, J. Morones, S. Sánchez, A logic with imprecise probabilities and an application to automated reasoning using rewriting techniques, Fuzzy Sets and Systems 218 (2013) 53–72, http://dx.doi.org/10.1016/j.fss.2012.08.004.

[4] F. Soriguer, et al., Prevalence of diabetes mellitus and impaired glucose regulation in Spain: the Di@bet.es study, Diabetologia 55 (2012) 88–93, http://dx.doi.org/10.1007/s00125-011-2336-9.

[5] A.A. Al Jarullah, Decision tree discovery for the diagnosis of type II diabetes, in: 2011 International Conference on Innovations in Information Technology, 2011, pp. 303–307, http://dx.doi.org/10.1109/INNOVATIONS.2011.5893838.

[6] M.M. Rashid, M.R. Askari, C. Chen, Y. Liang, K. Shu, A. Cinar, Artificial intelligence algorithms for treatment of diabetes, Algorithms 15 (9) (2022) 299, http://dx.doi.org/10.3390/a15090299.

[7] R.B. Dessau, C.B. Pipper, "R"–project for statistical computing, in: Ugeskrift for Laeger, Vol. 170, No. 5, 2008, pp. 328–330, PMID: 18252159.

[8] rpart: Recursive partitioning and regression trees, 2022, Available online: https://cran.r-project.org/web/packages/rpart/index.html. (Accessed 24 October 2022).

[9] M. Villalba-Orero, E. Roanes-Lozano, A prototype of a decision support system for equine cardiovascular diseases diagnosis and management, Mathematics 9 (20) (2021) 2580, http://dx.doi.org/10.3390/math9202580.