

METHODOLOGY ARTICLE

Open Access



Multiobjective grammar-based genetic programming applied to the study of asthma and allergy epidemiology

Rafael V. Veiga^{1,2*} , Helio J. C. Barbosa^{2,3}, Heder S. Bernardino², João M. Freitas², Caroline A. Feitosa⁴, Sheila M. A. Matos⁴, Neuza M. Alcântara-Neves⁵ and Maurício L. Barreto^{1,4}

Abstract

Background: Asthma and allergies prevalence increased in recent decades, being a serious global health problem. They are complex diseases with strong contextual influence, so that the use of advanced machine learning tools such as genetic programming could be important for the understanding the causal mechanisms explaining those conditions. Here, we applied a multiobjective grammar-based genetic programming (MGGP) to a dataset composed by 1047 subjects. The dataset contains information on the environmental, psychosocial, socioeconomics, nutritional and infectious factors collected from participating children. The objective of this work is to generate models that explain the occurrence of asthma, and two markers of allergy: presence of IgE antibody against common allergens, and skin prick test positivity for common allergens (SPT).

Results: The average of the accuracies of the models for asthma higher in MGGP than C4.5. IgE were higher in MGGP than in both, logistic regression and C4.5. MGGP had levels of accuracy similar to RF, but unlike RF, MGGP was able to generate models that were easy to interpret.

Conclusions: MGGP has shown that infections, psychosocial, nutritional, hygiene, and socioeconomic factors may be related in such an intricate way, that could be hardly detected using traditional regression based epidemiological techniques. The algorithm MGGP was implemented in c++ and is available on repository: <http://bitbucket.org/ciml-ufjf/ciml-lib>.

Keywords: Genetic programming, Asthma, Allergy, Classifier, Multiobjective

Background

One of the major aims of epidemiology is to identify risk and protective factors associated with the occurrence of specific diseases in humans. However, the study of these relationships in complex diseases, such as asthma and allergies, has proven to be difficult due to the large number of factors found to be related with these disorders [1]. Epidemiological studies very often use statistical tools like multivariate logistic regression and correlation analysis to model the relationships between risk factors and dichotomous outcomes [2]. While generally very

powerful, these approaches assume that the predictive variables are independent and that the data can be modeled using linear combinations of these variables [3, 4]. As a consequence, when the relationships between these variables are nonlinear or they are interdependent (or conditionally dependent), the performance of the statistical approaches decreases. As many biological systems are fundamentally nonlinear and their parameters are conditionally dependent [4], the use of other approaches must be considered. In those situations, machine learning techniques has emerged as an useful alternative.

Machine learning is a branch of artificial intelligence that employs a variety of statistical, probabilistic and optimization techniques that allow computers to “learn” from past examples and to detect hard-to-discern patterns from

*Correspondence: rafaelvalenteveiga@gmail.com

¹Center of Data and Knowledge Integration for Health (CIDACS), Instituto Gonçalo Muniz, Fundação Oswaldo Cruz, Salvador, Brazil

²Universidade Federal de Juiz de Fora, Juiz de Fora, Minas Gerais, Brazil

Full list of author information is available at the end of the article



large, noisy or complex systems. Machine learning has become a popular tool for medical researchers interested in predictive models, as well as, in the identification and exploration of patterns from biological phenomena. For instance, applications of machine learning techniques to different epidemiological problems can be found in the literature [4–7]. Many of these methods are efficient in modeling complex relationships between the independent variables. Unfortunately, these techniques often generate models that are difficult to interpret. Thus, the models generated by some machine learning approaches can be useless in clarifying the complex epidemiological relationships. Genetic programming [8] techniques in general, and the grammar-based [9] ones in particular, are exceptions as they are capable to generate interpretable models.

The theory of natural selection of Charles Darwin and Alfred Wallace has influenced much of human knowledge. The great ability of natural selection to generate biological complexity, efficiency of biological organisms, and adequate processes provided inspiration for the development of machine learning techniques such as the Genetic Algorithm. The Genetic Algorithm solves problems based on the process of natural selection. Genetic Programming is a particular type of Genetic Algorithm that can be used to generate computational artifacts (such as computer programs, mathematical models, logical models) that help explain observed data.

Grammar-based Genetic Programming is a specific type of genetic programming which uses a formal grammar that contains the rules and syntax used to generate appropriate solutions by the algorithm [9]. The use of a formal grammar to generate epidemiological models has two major advantages: (i) it enables the algorithm to generate more interpretable models by the use of a language closer to the human language, like the use of conditional relationships (if, else), logic (and, or) and comparatives (greater, lesser, equal); and (ii) it enables the researcher to establish his/her own rules for forming models and to introduce their knowledge in order to generate more appropriate models. Some studies have shown that grammatical genetic programming can be applied to several problems obtaining good results [10–12].

Multiobjective optimization problems (MOOP) are ubiquitous in real-world decision making. It is generally the case that a decision maker must simultaneously account for multiple criteria, with each criterion contributing to different objective to be optimized. Solving an MOOP involves obtaining a set of solutions that provide optimal tradeoff among all the relevant objectives constrain a Pareto-optimal solutions. In other words, a solution is considered optimal in the multiobjective sense if an attempted improvement in any one of its objectives is necessarily accompanied by the deterioration of at least one other objective [13].

The choice of an epidemiological model can be considered an MOOP because we can consider two criteria for selecting the best model: (i) the choice of more accurate models; (ii) the choice of models with reduced complexity (and thus more parsimonious). The Nondominated Sorting Genetic Algorithm II (NSGA) [14] applied in this study is a type of MOOP that uses the concept of dominance and the distances between the solutions for establishing the set of best solutions.

Asthma is among the most common chronic diseases worldwide, causing high levels of morbidity [15]. It is a heterogeneous condition with different phenotypes. It has been causally associated with diverse environmental factors as well as genetic backgrounds [16]. The prevalence of allergy and asthma has increased in affluent countries over recent decades, and has increased also in cities of non-affluent countries such as in Latin America [16, 17]. Such temporal trends occurring over a relatively short period of time are unlikely to be explained by changes in genetic susceptibility and are most likely explained by changes in environmental exposures such as those associated with the adoption of a modern or “westernized” lifestyle [18]. There are many studies showing different factors related to these disorders such as environmental factors [19, 20], socioeconomic [20, 21], infections [22–25], nutritional [26, 27], psychosocial [28, 29] and genetic [30, 31]. Recently, evidence has emerged to suggest that asthma causation may involve interactions between different exposures [30, 32]. Thus asthma and allergies can benefit from the use of techniques able to identify complex relationships.

This study evaluates the use of Multiobjective Grammatical Genetic Programming (MGGP) to find relationships between environmental, socioeconomic, infections, psychosocial and nutritional factors that may be related to the occurrence of asthma and allergies. To our knowledge, this methodology has not been applied before to solve an epidemiological condrum. We believe this approach could be of great use also in many other epidemiological problems lacking advanced tools for analyzing large and complex causal relationships.

Methods

This section describes the study population, how asthma states were defined, allergy markers and genetic programming technique. This section also describes the techniques and methodologies applied in obtaining the data used to search for relationships between various exposures with the occurrence of asthma and allergies. The expositions studied in this work cover anthropometric, psychosocial, diet, environmental and infections aspects.

Study population and data collection

The study was a post hoc analysis of data collected during a survey of 1445 children aged 4–11 years and living in 24 poor neighborhoods in the city of Salvador, Northeast Brazil, performed in 2005 as part of a cohort study to investigate risk factors for asthma and allergy, and is described in detail elsewhere [33]. The neighborhoods and the children were selected as part of a previous study designed to measure the impact of sanitation on diarrhea [34]. Data on asthma symptoms were collected using a Portuguese-adapted ISAAC Phase II questionnaire, also a psychosocial and nutritional questionnaire was applied. The following measurements were performed for each child: anthropometric measurements, SPT testing and serum IgE for four aeroallergens, circulating IgG against six different pathogens, stool examination for detection of intestinal helminthic infections. The presence of mold on household walls was determined by direct inspection.

Anthropometric measurements

The children were weighed on portable electronic scales (Filizola®, model E-150/3P, with a capacity of 150 kg and accuracy of 100 g) and height was measured using stadiometers (Leicester Height Measure). Each measurement was done twice by different examiners and variations of 100 g for weight and 0.1 cm for height were accepted with the mean of duplicate observations used for calculation of body mass index (BMI) ($\text{weight}[\text{kg}]/\text{height}[\text{m}]^2$). Z scores for BMI by age and gender were calculated against WHO 2006 reference values. Children with z-scores greater than 1.0 were considered to be overweight or obese [35, 36]. Previous studies carried out in Salvador indicated that overweight or obese may have important role in development of asthma and allergy [37].

Psychological disorder in the mother

The SRQ-20 questionnaire was used to assess minor psychiatric disorders in the mother. This instrument was developed by the World Health Organization [38] and validated in Brazil by Mari and Williams [39]. It is composed of 20 questions with dichotomous (yes/no) answers referring to the presence or absence of symptoms of depression, anxiety and somatic disorders in the previous month. A cut-off point for the definition of suspected cases of minor psychiatric disorders was established as 8 or more positive answers, a condition that, although not characterizing a psychiatric diagnosis, indicates significant psychic suffering. This cut-off point was defined in accordance with studies previously carried out in Brazil [39]. Also, previous studies carried out in Salvador reported an important association between minor psychiatric disorders in the mother and asthma symptoms in the child [40, 41].

Dietary patterns

Information about the dietary patterns were obtained based on questionnaire of food frequency, validated by [42]. This questionnaire consists of 98 foods, related to food consumption in the last 12 months. A principal components analysis was used to obtain four food patterns that were represented by the numbers 1–4, and their value was discretized by their tertiles to create 4 levels. Details analysis to define the dietary pattern can be found in the work of [43]. The pattern 1 was characterized by the predominance of fruits, vegetables, legumes, cereals and fish. The pattern 2 was characterized by the predominance of milk and dairy products, ketchup / mayonnaise / mustard and chicken. The pattern 3 was characterized by the predominance of fried foods, sweets, snacks, coolant / artificial juice. The pattern 4 was characterized by the predominance of sausages, eggs and red meat.

Allergen SPTs

SPTs were performed by two trained technicians using a standardized protocol and extracts of *D. pteronyssinus*, *B. tropicalis*, *B. germanica*, *P. americana*, dog and cat epithelia, and a fungal allergen mix (ALK-Abelló, São Paulo, Brazil). Extracts, saline and histamine controls were pricked onto the forearm skin using a disposable lancet (ALK-lancet; ALK-Abelló, São Paulo, Brazil). Reactions were read after 15 minutes and a reaction was considered positive if the mean diameter of the wheal was 3 mm or larger than the saline control wheal. Frequencies of positive skin test reactions to dog and cat epithelia and a fungal allergen mix were low (< 4%) and were excluded from further analysis.

Detection of intestinal helminth ova in fecal samples

Two fecal samples were collected two days apart and analyzed using the Hoffman sedimentation method and the Kato-Katz thick-smear technique [44] for the presence of helminth parasites (*Trichuris trichiura*, *Ascaris lumbricoides*, hookworms and *Schistosoma mansoni*). Hookworms and *S. mansoni* infections were rare (< 1%) and were not considered further in this analysis.

Serum immunoassay for IgG to bacteria, protozoa, and viruses

Serum IgG antibodies to *Helicobacter pylori*, *Toxoplasma gondii*, herpes simplex virus (HSV), herpes zoster virus (HZV), Epstein-Barr virus (EBV) were measured using commercial ELISA kits (Diamedix, Miami, Florida, USA; Adaltis, Toronto, Canada). For the hepatitis A virus (HAV), kits from ADALTIS were used (Toronto, Canada). The assays were performed following the manufacturer's instructions.

Detection of allergen specific IgE by Immunocap

IgEs reacting with *Dermatophagoides pteronyssinus*, *Blomia tropicalis*, *Blatella germanica* and *Periplaneta americana* were measured in sera, using the Immunocap System (Pharmacia AB, Uppsala, Sweden), according to the manufacturer's instructions. Sera containing 0.70kUIgE/L or more were considered positive.

Genetic Programming (GP)

Genetic programming (GP) is a special type of genetic algorithm which creates computational artifacts (for instance, computer programs written in a given language) to perform a given task. Although GP as it is known nowadays starts with Cramer [45], it was the work by Koza [8] that defined and popularized the method which was subsequently known as “standard GP”. In GP, the candidate solutions are referred to as “programs”, a high-level structure able to represent a large class of computational artifacts, such as a standard computer program, a numerical function or a classifier in symbolic form.

A population of candidate solutions is improved in GP following the same steps of a genetic algorithm. The steps of these techniques are presented in Algorithm 1 where “createInitialPopulation” creates the initial population, “evaluatePopulation” finds out how well the candidate solutions perform, “selectFittest” selects the best solutions with respect to their fitness, “crossover” combines each pair of parents generating new candidate solutions which are then mutated in “mutation”, and “replace” generates a new population by combining candidate solutions from the current(parent) and the offspring populations [46].

Algorithm 1: Algorithm of a GP [46].

```

population ← createInitialPopulation();
while a stop condition is not met do
    evaluatePopulation(population);
    selected ← selectFittest(population);
    offspring ← crossover(selected);
    mutation(offspring);
    replace(offspring, population);
end

```

Typically, GP differs from standard genetic algorithm (i) in its representation of the candidate solutions, whereas GAs are intended to find an array of characters or numbers for representing the solution of a given problem, the goal of a GP process is to produce a “program” (or, as in our case, an expression) solving the optimization problem at hand, and (ii) in the definition of the move operators: crossover and mutation. The different possible ways of representing a candidate solution (such as trees, graphs, etc.) can be used to classify the GP variants.

Genetic programming has been applied to find solutions from a wide variety of fields. Producing patents and about 76 results that equals or surpasses the solutions

found by human experts in their fields of research [47]. Among the fields of research we can highlight development of electric and quantum circuits [48], development of communication antennas [49], finite algebra [50], image recognition [51], symbolic regression [52] and reverse engineering [53].

In the study of biological systems the GP has been little applied, however recently several works have applied GP in the study of gene expression [54, 55], modeling of algal growth [56], prediction of cancer [57, 58], prediction of medical diagnosis [59], in the identification and classification of different types of scoliosis [60] and one area that GP has attracted interest is genome-wide association studies [61].

Grammar guided GP

Grammar guided GP [62], or grammar-based GP, uses grammars as a way to constrain the representation of the candidate solutions. Grammars can be used to create structures which belong to a specific language, and a formal grammar G can be defined as [63]

$$G = \{N, \Sigma, R, S\}, \quad (1)$$

where N is a finite set of nonterminals (each nonterminal is formally delimited by $<$ and $>$), Σ is a finite set of terminals or token symbols which are items that can appear in the language (such as constants, variables, and functions), $S \in N$ is the start symbol, and R is a finite set of rules (or productions) which are as

$$(\Sigma \cup N)^* N (\Sigma \cup N)^* ::= (\Sigma \cup N)^*, \quad (2)$$

where $*$ is the Kleene star operator¹, $\cup$ denotes set union.

Typically, Grammar guided GP techniques use context-free grammars, a type of grammar in which the left-hand side of each production rule consists of a single nonterminal symbol, that is,

$$N ::= (\Sigma \cup N)^*. \quad (3)$$

The candidate programs in Grammar guided GP are represented by derivation trees, in which the internal nodes are the nonterminals of the grammar and the leaf nodes are symbols which appear in the language (terminals). An example of a derivation is available in the Fig. 1.

Grammar guided GP uses a grammar to guide the allowed representation of the candidate programs. The use of grammar delimits the creation of the initial population as well as the application of the variational operators as mutation and recombination. For both mutation and recombination, it is only permissible to exchange a non-terminal N for another of the same type, thus maintaining the consistency of the models. The recombination operator is shown in Fig. 2. It is randomly selected a non-terminal that exists in both parents and occurs

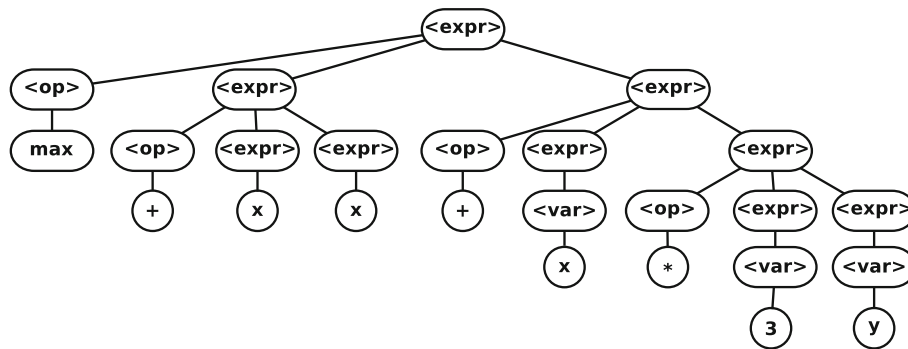


Fig. 1 Example of a derivation tree [46]

the exchange of subtrees between parents. The mutation operator is shown in Fig. 3, a randomly selected subtree is replaced by another randomly created with the same non-terminal as root.

Multiobjective Grammatical Genetic Programming (MGGP)
An optimization problem seeks to find a solution that maximizes or minimizes an objective. However, many problems require finding the best solutions according to multiple objectives, thus being a multiobjective optimization problems (MOOP). The search for the relationships between factors associated with complex diseases such as asthma can be studied as a MOOP, where it is aim to maximize the accuracy and minimize the complexity of the relations simultaneously. This multiobjective approach aims to find the models that best explain this pathology

being as simple as possible and therefore more parsimonious. The Grammar guided GP usually is applied to a mono-objective problem. To create the capability to solve MOOP, instead of using the obtained value of the objective function as criterion for selecting the best solutions in mono-objective problem. The MOOP algorithm NSGA [14] is based on dominance idea. Where one solution dominates the other if this solution is better in relation to all objectives, otherwise the solution is non-dominated. The NSGA uses two criteria for selecting the best solutions based on the objective functions:

- The dominance rank. All solutions which there is no other solution that is better than it for all objective functions simultaneously is call a nondominated solution. The rank 1 is formed by all nondominated

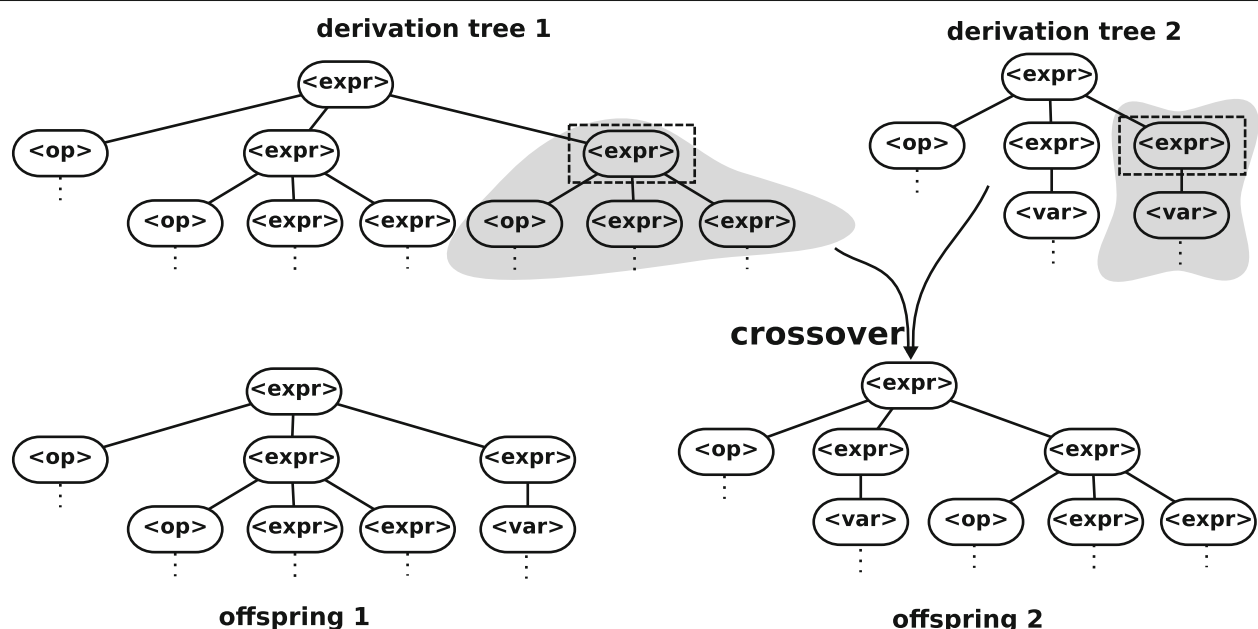
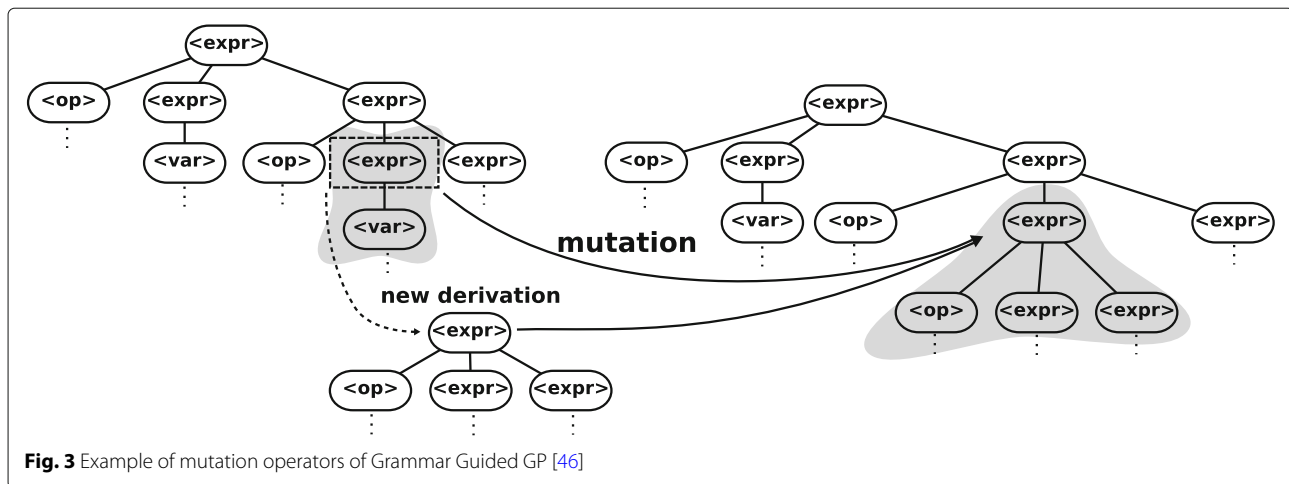


Fig. 2 Example of crossover operators of Grammar Guided GP [46]

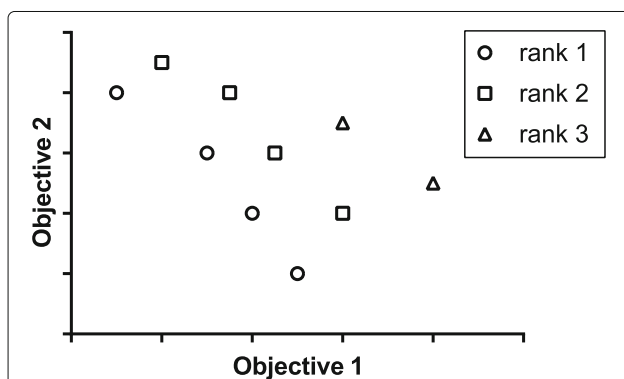


solutions, rank 2 is formed for all solutions that are dominated by only rank1, and so on. This idea is illustrated in Fig. 4.

- The crowding distance computation requires sorting the population according to each objective function value. Thereafter, for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned as an infinite crowding distance value. All other intermediate solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions. This calculation is continued with other objective functions. The overall crowding distance value is calculated as the sum of individual distance values corresponding to each objective. Each objective function is normalized before calculating the crowding distance.

Computational experiments

This section describes the details regarding the the computational experiments methodology. Firstly, it explains



how the variables were chosen for modeling and then described as the preparation of the data with the formations of the study groups.

Model construction

Models were created for the explanation of 3 outcomes: Asthma, SPT and IgE against allergens. The exposure variables chosen were those that potentially represent the aspects that may be related directly or indirectly with asthma and allergy. The exposure variables and their frequency can be seen in Table 1 and were: gender, age, parental asthma, number of siblings, body mass index (BMI), HSV, HZV, EBV, HAV, *T. gondii*, *H. pylori*, *A. lumbricoides*, *T. trichiura* infections, four dietary patterns, daily calories, gross national income (GNI), mother psychological disorder, daycare ever, smokers at home, sewage disposal system, linen bed exchange, cat at home, dog at home, the presence of mold or moisture, piped water system, fly at home, paving of the street.

Data preparation

Most of the children who had missing data were due to refusal to withdraw blood samples, consequently they has missing for all serological data, or failing to provide all stool samples which made them missing all parasitological variables. This made it difficult to apply a methodology for imputation missing data. The use of individuals with missing data in the analyzes would cause different models to present different number of instances, which would compromise their adequate evaluation. We prefer to exclude all children who had missing data for any of the variables studied were excluded from the study then from the original 1445 children, 1047 has complete data. For realization of computational experiments the population was divided into groups. We randomly selected 10% of individuals (instances) to form the test group. This draw was made keeping the frequency of the outcome in the

Table 1 Variables used to build Models

Variables	Type	Freq %
<i>N</i> = 1046		
Target variables		
IgE (positives)	Boolean	38.6%
SPT(positives)	Boolean	30.3%
Asthma (positives)	Boolean	22.9%
Input variables		
Gender (males)	Boolean	52.7%
Age	Categorical	
4 and 5		35.9%
6 and 7		35.1%
8 to 11		29.0%
Parental asthma (presence)	Boolean	12.6%
HSV (positives)	Boolean	54.9%
HZV (positives)	Boolean	45.8%
EBV (positives)	Boolean	88.4%
HAV (positives)	Boolean	16.7%
<i>T. gondii</i> (positives)	Boolean	18.4%
<i>H. pylori</i> (positives)	Boolean	27.6%
<i>A. lumbricoides</i> (positives)	Boolean	16.2%
<i>T. trichiura</i> (positives)	Boolean	11.2%
Sibling number	Categorical	
none		18.9%
1		35.2%
2		24.0%
3 or more		21.9%
Daycare ever (yes)	Boolean	15.4%
Smoke at home (presence)	Boolean	27.1%
Sewage disposal system (presence)	Boolean	83.5%
Change bed linen ≥ 1 per week	Boolean	45.0%
Cat at home (presence)	Boolean	17.6%
Dog at home (presence)	Boolean	39.8%
Mold/moisture at home (presence)	Boolean	68.6%
Piped water system (presence)	Boolean	91.9%
Paving of the street (absence)	Boolean	35.1%
Fly at home (presence)	Boolean	51.5%
Mother Psychological disorder (suspect)	Boolean	37.2%
Dietary patterns 1 to 4	Categorical	Split by tertiles
Daily calories (<i>Kcal mean(sd)</i>)	Numerical	2210(929)
BMI	Categorical	
Overweight / Obesity		12.2%
Eutrophic		75.1%
Slimness		12.7%
GNI	Categorical	Split by tertiles

group equal to the frequency of the same in the original population. Of the remaining 90%, it was performed 6 times cross-validation where the population was randomly divided into 5 parts, maintaining the proportions in relation to the outcome equivalent to that of the original population. The first part of the population is defined as validation group and its respective training group consists of the other 4 parts. The process continue for for each part been validation group and the other ones been their respective training group. for each cross-validation was produced 5 validation group and their respective training group. At the end of the run 6 times the cross-validation was obtained 30 training groups with 751 subjects, their 30 validation groups with 190 subjects and one group with 105 subjects, respectively. All groups have relative frequencies similar to the original population. The same groups were used in all analyzes.

The study population showed more negative individuals for asthma and allergies than positive individuals (unbalanced database), and we therefore applied random over-sampling [64] in each training and validation group in order to prevent the negative group for asthma and allergies from having a greater influence on the accuracy than the positive group. Random over-sampling technique was not applied to test group.

MGGP

The MGGP was executed in 30 independent times for each training group. The MGGP was applied according to the standard algorithm for GP shown in the Algorithm 1. An initial population of 500 candidate solutions was randomly generated. The population was evaluated using NSGA based on two objective functions, (i) minimizing the classification error of the model in the training group (ii) minimizing the complexity of the model, given by the number of terminals in the tree representation of the candidate solution. The selection of parental solutions was carried out using tournament: two solutions were randomly selected and the best one of them was chosen to be a parent solution. Then the combination of two parental solutions generate two offspring solutions which suffer crossing and mutation. This process was repeated until 500 offspring solutions were generated and evaluated. The 500 best solutions between parent and offspring solutions were selected to form the next generation. The MGGP executed a total of 20,000 generations to obtain the final population. From the population of solutions at the end of the 20,000 generations, the solutions chosen as best were those that were non-dominated using the error in the validation group instead of the error in the training group to avoid problems with overfitting.

The context-free grammar built for this work contains comparison operators ($>$, $<$, $>=$, $<=$, $==$ and $!=$), logic

operators (*and*, *or*, *xor*) and the ternary operator *if – then – else*. To be easier to handle the solution computationally, it was used postfix notation. The grammar was composed by the following rules (R):

```

< expr >::= < expr1 >< expr2 >< bool > if-else < binaryClass >
< expr1 >::= < expr2 >< expr2 >< bool > if-else < binaryClass >
< expr2 >::= < binaryClass >< binaryClass >< bool > if-else |
    < binaryClass >
< bool >::=
    < varBin > < binaryClass > < compbin > |
    < varCat > < CatClass > < compcat > |
    < varNum > < numValue > < compCont > |
    < bool > < bool > < log >
< compcat >::=
    < | <= | == | >= | > | !=
< compbin >::=
    == | !=
< compCont >::=
    < | <= | > | >=
    < log >::=
    and | or | xor
< binaryClass >::=
    0 | 1

```

where the symbol “|” was used to delimit multiple derivation possibilities, “< varBin >” were the set of binary variables, “< varCat >” were the set of categorical variables, “< CatClass >” were the possible values of each of categorical variables, “< varNum >” were the set of numerical variables, and “< numValue >” were the possible values of each of numerical variables. As we can observe, the rules presented for this work limit the number of nested *if – then – else* operations to 3 levels. This is to prevent the application of a new operation on a very small and non-representative group in terms of number of individuals.

Multivariate logistic regression (RL)

The RL models were generated for each of the training groups and then these generated models were evaluated on their respective validation and testing groups. For choice of exposure variables, the gender and age variables were considered a priori variables and always in the RL models. The choice of other variables was by stepwise bidirectional selection [65] keeping the significant variables ($p - value < 0.05$) in the model. As we want to compare a regression with classification models, the RL has been converted into a classification model by applying a step function on the predicted value, meaning that if the value predicted by the model is greater than 0.5 then the predicted value is 1, otherwise it is 0. These analyses were performed in Weka V3.6².

Classification algorithm C4.5

Models using the classification algorithm C4.5 [66], were also generated for each of the training groups and then were evaluated on their respective validation and testing groups. To avoid overfitting, the parameter of minimum number of instances per leaf was set to maximize the mean accuracy of the models for all executions in the validation groups. These analyses were performed in Weka. J48 is the Java implementation of C4.5 in Weka tool.

Classification algorithm random forest (RF)

The RF [67] algorithm was applied in the 30 training groups. The parameter maximum size of the trees chosen was 3, because this presents the smallest errors in the validation group after the models be generated in the training groups. These analyses were performed in Weka.

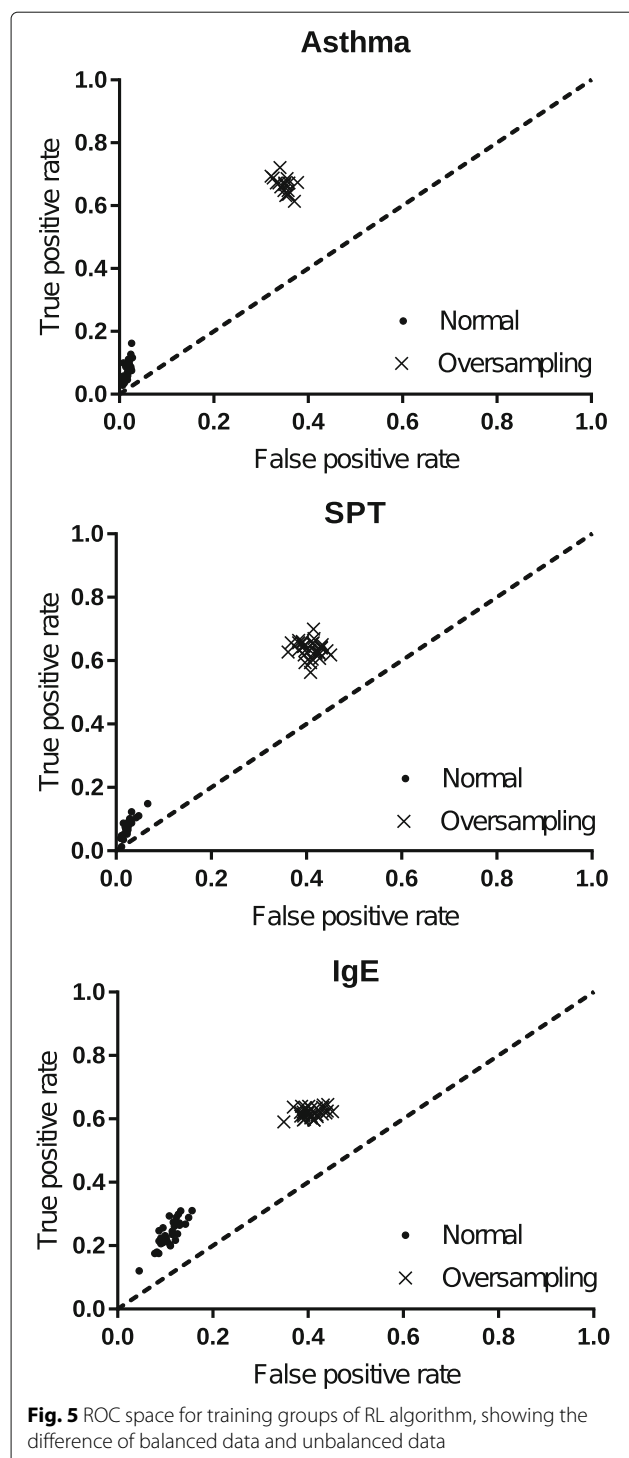
Results

The variables used in this study and the variable frequencies are shown in Table 1. This population had high prevalence of asthma (22.9%), SPT (30.3%) and IgE (38.6%) positivity. Such high prevalence has as consequence, the number of positive cases approaching the number of negatives cases, so that an unbalanced problem was not expected. However, as shown in Fig. 5, the data balancing had a profound effect on improving the ability to predict positive cases for these conditions, thus balancing type 1 and type 2 errors. Other studies also showed the importance of data balancing in classification algorithms applied to epidemiological problems [7, 68].

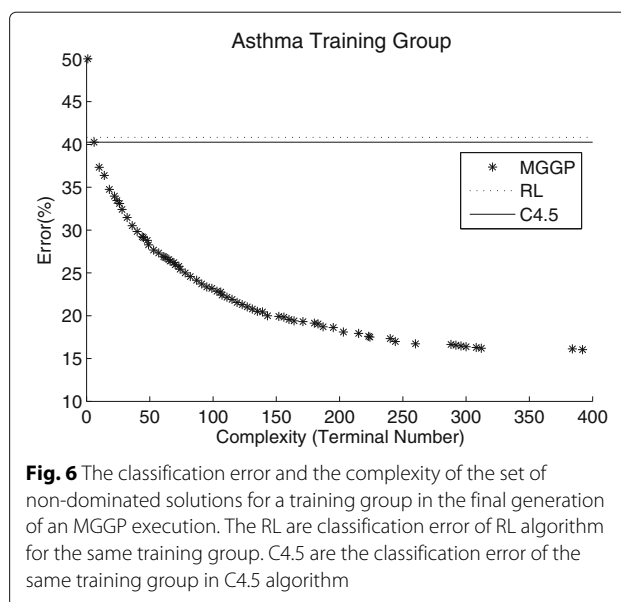
All executions of MGGP showed a good range of trade-off between complexity and error. An execution example is displayed in Fig. 6. This shows that the MGGP was able to find a diverse set of optimal solutions, each with different tradeoff between complexity and accuracy. It is evident that for the set of non-dominated solutions be large, it is not possible to generate low complexity solutions with low misclassification, because that would make this solution dominate the other solutions and reduce the size of the non-dominated set. The list with the best models found by MGGP can be downloaded in Additional file 1.

Solutions with low complexity are too simple to explain asthma and allergy and consequently have low accuracy. With increasing complexity the misclassification number tends to drop, however very complex models tend to get very specific to the studied sample and lose the ability to explain other databases. To avoid losing such ability, at the end of execution non-dominated models with respect to the validation group are selected. Despite the best model be the one with the smallest error in the validation group, the solutions with less complexity should not be discarded, as they have the potential to highlight relationships relevant to the understanding of the problem.

The set of solutions obtained by MGGP are non-dominated solutions with respect to the validation group obtained at the last generation. To evaluate these solutions the accuracy in the test group was adopted. Table 2 shows the performance of the different techniques evaluated in the test group. The test group is a single group for every 30 runs of the algorithm. Despite the fact that the test group is small (10% presents data), it is composed of data unseen in any execution. So it used to test the general performance of a given solution in different executions.



Although most of the best solutions obtained by MGGP showed complexity lower than 50 terminals, a few complex solutions with good accuracy and generalization were found. Each MGGP run took an average of 28.1h on an intel i7 7500 2.7GHz computer with 8GB DDR4 ram. The current version does not have parallelism capability and we expect to have great performance impact when



parallelization is implemented in a future release. The average accuracy comparison among RL, C4.5, RF and MGGP with different complexity ranges is shown in Fig. 7. With respect to asthma, RF, C4.5 and MGGP solutions showed no significant differences in average accuracy. However MGGP with complexity between 10 and 13 had significantly greater average accuracy than RL p -value = 0.003 test T student. With respect to SPT, the RL, RF and MGGP showed no significant differences in accuracy, while C4.5 shows low performance. In IgE outcome, the MGGP with complexities between 10-13 and 14-25 showed higher average accuracy than RL (p -value < 0.001 and 0.002 respectively test T student) and C4.5.

Discussion

Most epidemiological studies use techniques that capture only linear relationships between predictor variables, as for example RL. MGGP for asthma and IgE finds solutions with accuracies better than RL, indicating that there are important complex relationships that RL solutions cannot capture. Although C4.5 and RF was able to find complex solutions, this algorithm showed a lower average accuracy than MGGP.

The RF presented accuracy equivalent to MGGP, but the objective is not to predict asthma and allergy, as this would not be expected based only on studied factors. Because these are complex pathologies with multiples still unknown risk factors. The objective of this work is to find relationships between the studied factors that could potentially be related to asthma and allergies. RF is not useful for that objective because has little capacity to clarify these relations. Another application of computational intelligence in the study of asthma in children, found 62%

Table 2 Accuracy obtained in the test groups for different techniques, where RL is logistic regression, RF is random forest and * indicates that all executions converged to the same model

Asthma					
	Mean	Median	sd	Min	Max
RL	56.67	56.19	4.82	45.71	63.81
C4.5	61.97	62.86	3.21	55.24	66.67
RF	61.81	62.38	3.78	53.33	68.57
MGGP	61.15	62.36	5.53	50.64	71.87
SPT					
RL	54.19	55.24	3.60	44.76	60.00
C4.5	50.38	50.48	0.72	46.67	51.43
RF	57.87	57.62	2.91	52.38	65.71
MGGP	56.69	57.74	4.18	49.02	66.46
IgE					
RL	55.43	55.24	2.94	50.48	63.81
C4.5 *	53.33	53.33	0	53.33	53.33
RF	58.39	58.49	2.92	52.83	64.15
MGGP	58.39	58.57	3.05	48.39	63.26

accuracy using environmental and genetic information [6]. The fact that MGGP achieves equivalent values for asthma using only environmental data, and better results than RL and C4.5, shows the potential of MGGP to discovery relations when applied to complex epidemiological studies.

Examples of relations obtained by MGGP are shown in Table 3. For asthma we note that an important feature that appears in many relationships is the low age. Asthma is a heterogeneous condition with different phenotypes and clinical expressions. A common phenotype of asthma is the transient wheezing phenotype that is not commonly associated with a family history of asthma or with atopy. For this phenotype, the symptoms tend to regress at age 3-5 years old [69], and the high prevalence of this phenotype may explain this relation with low age. Some less complex relationships commonly found were: (i) low age or dog at home are related to asthma, indicating that dog at home is also related with increased asthma, (ii) cat at home or low age increasing asthma, indicating that cat at home is also related with increased asthma (iii) suspected mother psychological disorder also show increased chance to be asthmatic. Some relationships found that affect the chance of being positive for SPT were the presence of infections *T. trichiura* and HSV, with hygiene marker as frequent linen exchange bed and sewage disposal. Other important relation found with SPT, was the high consumption of foods rich in frying (pattern 3) and predominance of sausages, eggs and red meat (pattern 4). This results indicating that those infections, environment, and feeding behavior may influence SPT positivity.

The models generated by MGGP to explain IgE showed that male gender was related to having IgE. The absence of infections such as *T. gondii* and *T. trichiura* as well as sewage disposal are associated with increased IgE. A model that may provide some information for understanding IgE was the one given by: “if(((Nutritional Factor1 = 1) or (Gender = 1)) and ((sewage disposal = 1) xor (Tgondi = 0))) { 1 } else { 0 }”. This model indicates that when a person has moderate levels of consumption of fish, fruit, cereals, or is male, and also shows the absence of *T. gondii* infections or exclusive presence of sewage system, the chance of having IgE increases. Many biological phenomena do not have a linear behavior. Immune cells like lymphocytes, when stimulated have their response increased. However, excess stimulation leads to anergy or apoptosis of these cells, thus reducing the response. This kind of behavior is hardly detected properly using RL. In case of male gender or moderate values of feed pattern 1, it is possible to see in this model and others that both the presence of sewage and the absence of *T. gondii* infection increase the chance of being positive for IgE. This model indicates that excess risk factors may lead to a reduction in the chances of being IgE positive. This type of IgE behavior is reinforced by the frequent occurrence of the “xor” logical operator in more complex models.

We also performed MGGP runs for each outcome on all 1047 individuals without separating by groups. Even knowing that we could not avoid problems of overfitting, we want to observe models that take into account the maximum number of people possible. For asthma we

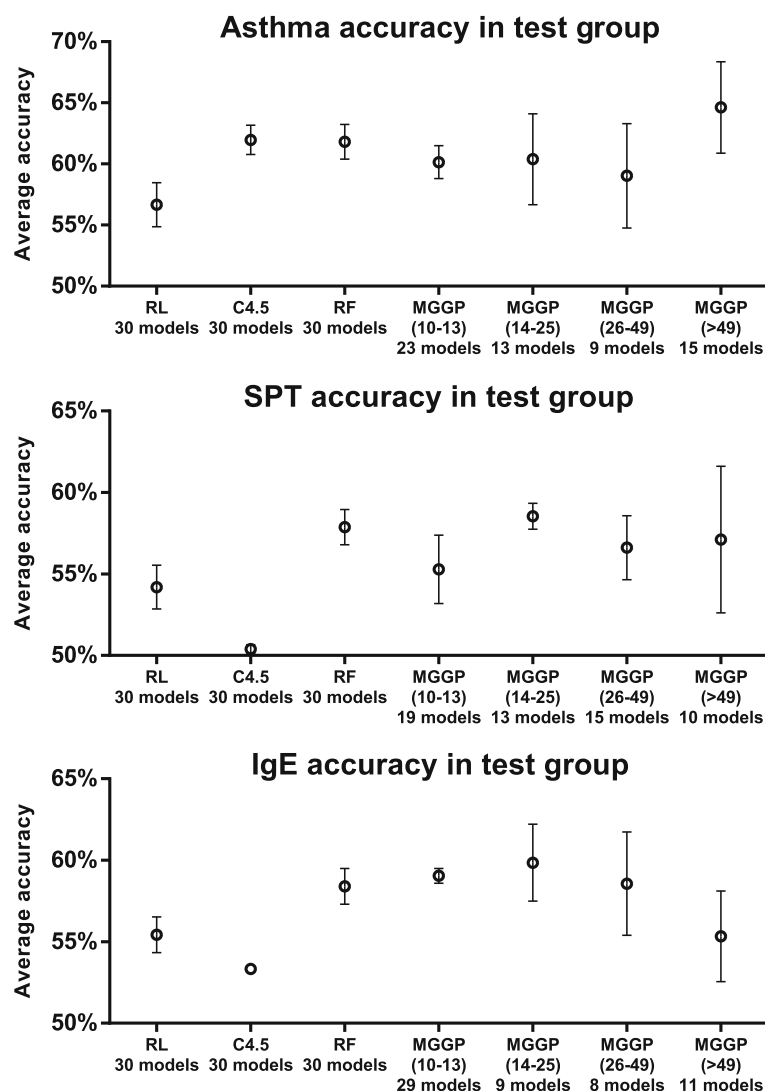


Fig. 7 Average accuracy and their 95% confidence interval for solutions of asthma, SPT and IgE in the test group obtained by algorithms RL, C4.5, RF and different ranges of complexity for solutions obtained by MGGP

found if((age_i = 0) or (gat == 1)) 1 else 0, suggesting that low age is important in asthma as discussed earlier. The presence of a cat in the house and its association with asthma has presented contradictory results in literature. Some studies find a positive association with asthma [70, 71]. Others found a negative association [72]. One of the reasons for such disagreements between the works is that the presence of a cat may enhance asthma symptoms, so it is common for parents with asthmatic children to avoid cats, which could cause a negative association in most studies. The list of the best models generated by MGGP in all individuals is shown in the material supplements.

Conclusion

The use of MGGP can be a good alternative to the understanding of epidemiological problems mainly in the study

of complex diseases. Among the qualities presented by this technique, we can highlight:

- MGGP works with classification models and non-linear regression.
- MGGP can generate models with a wide variety of operations such as conditionals (if, else), comparisons ($\geq$, $\leq$, $=$, $\neq$), arithmetic ($+$, $\div$) and specific operations customized to the application domain.
- MGGP makes it possible to define rules to deal with variables of different types such as continuous, discrete, categorical, among others. It is also possible to define how and what operations are possible between the different types of variables.
- MGGP employs rules that restrict the construction of the models, allowing the researcher to add knowledge

Table 3 Examples of relations found by MGGP for IgE, SPT and asthma. The first column is odds ratio of all database without test group, second column is odds ratio in the test group, and the third column is accuracy of relation express. Where “{1}” means positive for outcome, “{0}” negative for outcome

All database without test group		Test group	Accuracy (%)	Complexity	Important relations
OR(CI.95%)	OR(CI.95%)				
Asthma					
2.42(1.96; 2.99)	1.28(0.69; 2.41)		53.1	10	if((Dog at Home != 0) or (Age = 0)){ 1 }else{ 0 }
2.48(2.01; 3.06)	3.80(2.01; 7.55)		66.0	10	if((Cat at Home = 1) or (Age = 0)){ 1 }else{ 0 }
2.64(2.11; 3.31)	3.25(1.64; 6.42)		63.0	10	if((Mother Psychological disorder = 1) or (Age = 0)){ 1 }else{ 0 }
2.33(1.89; 2.88)	2.36(1.25; 4.43)		60.5	10	if((Dog at Home != 1) and (Mother Psychological disorder = 0)){ 0 }else{ 1 }
3.25(2.62; 4.03)	3.23(1.69; 6.14)		64.2	14	if((Age = 0) or ((Cat at Home = 1) and (Nutritional Factor3 <= 0))) { 1 }else{ 0 }
3.26(2.63; 4.07)	2.92(1.53; 5.56)		63.0	14	if(((Age > 0) or (Nutritional Factor2 = 2)) and (Cat at Home != 1)){ 0 }else{ 1 }
3.59(2.87; 4.49)	2.56(1.34; 4.88)		61.1	18	if(((Cat at Home = 1) and (Nutritional Factor3 <= 0)) or (Nutritional Factor2 < 2) and (Age = 0)){ 1 }else{ 0 }
3.86(3.10; 4.80)	2.50(1.32; 4.73)		61.1	22	if((Age != 0) or ((Mother Psychological disorder = 0) and (Dog at Home != 1)) and (HZV = 1)) and (Cat at Home != 1)){ 0 }else{ 1 }
3.91(3.12; 4.93)	1.73(0.93; 3.23)		56.8	22	if((HSV != 1) and ((Linen Bed Exchange != 0) and (Age = 1))) xor ((Dog at Home != 0) or (Age <= 0)){ 1 }else{ 0 }
4.45(3.57; 5.56)	2.78(1.46; 5.28)		62.3	31	if(Cat at Home = 1){ 1 }else{ if(((Alumbricoides != 0) or (Nutritional Factor2 < 2)) and ((Dog at Home != 0) or ((HZV != 1) or (Mother Psychological disorder = 1)) and (Age < 1)))){ 1 }else{ 0 } }
18.01(13.85; 23.60)	3.77(1.91; 7.46)		64.8	231	too large to show
SPT					
2.03(1.62; 2.55)	2.44(1.22; 4.86)		60.3	10	if((Linen Bed Exchange != 0) and (Trichiura != 1)){ 1 }else{ 0 }
2.01(1.61; 2.51)	1.58(0.81; 3.06)		55.5	10	if((Nutritional Factor4 > 0) and (Linen Bed Exchange != 0)){ 1 }else{ 0 }
1.93(1.55; 2.41)	2.92(1.46; 5.86)		62.3	10	if((Linen Bed Exchange = 0) or (BMI != 0)){ 0 }else{ 1 }
2.11(1.67; 2.66)	1.52(0.77; 2.99)		54.8	10	if((HSV = 0) and (Linen Bed Exchange != 0)){ 1 }else{ 0 }
2.46(1.97; 3.08)	2.06(1.06; 3.98)		58.9	14	if(((HSV = 0) or (Nutritional Factor3 >= 1)) and (Linen Bed Exchange != 0)){ 1 }else{ 0 }
2.68(2.15; 3.35)	2.45(1.26; 4.77)		60.9	18	if((HSV = 0) or (daycare = 1)) and ((Nutritional Factor4 != 1) or (Linen Bed Exchange != 0)){ 1 }else{ 0 }
2.45(1.96; 3.07)	2.23(1.14; 4.38)		60.0	20	if((HSV != 0) and (Nutritional Factor3 = 0)){ if(Nutritional Factor1 != 1) { 0 }else{ 1 } }else{ if(Linen Bed Exchange != 0){ 1 }else{ 0 } }
3.57(2.85; 4.49)	2.31(1.19; 4.48)		60.3	39	if(Nutritional Factor4 < 1) xor ((Nutritional Factor2 >= 1) or (num siblings = 1) and (Fly at Home = 0))) and (((Mother Psychological disorder != 1) or (num siblings >= 1)) xor (Linen Bed Exchange = 0))) { if((HSV != 1) or (Tgondi != 1)) { 1 }else{ 0 } }else{ 0 }
6.73(5.30; 8.59)	2.35(1.14; 4.85)		58.9	124	too large to show

Table 3 Examples of relations found by MGGP for IgE, SPT and asthma. The first column is odds ratio of all database without test group, second column is odds ratio in the test group, and the third column is accuracy of relation express. Where “1” means positive for outcome, “0” negative for outcome (*Continued*)

Asthma	All database without test group OR(CI.95%)	Test group OR(CI.95%)		Accuracy (%)	Complexity	Important relations
IgE	1.76(1.39; 2.22)	2.28(1.12; 4.63)	60.0	10	if((Gender != 1) or (Cat at Home != 0)){ 0 }else{ 1 }	
	2.00(1.58; 2.53)	1.87(0.93; 3.75)	57.7	10	if((Gender != 0) and (sewage disposal != 1)){ 1 }else{ 0 }	
	1.63(1.28; 2.08)	1.81(0.88; 3.72)	56.9	10	if((Nutritional Factor1 != 1) and (Gender != 0)){ 1 }else{ 0 }	
	2.17(1.71; 2.75)	2.37(1.11; 5.06)	59.2	14	if(((Tgondi = 0) and (Gender != 0)) and (sewage disposal != 1)){ 1 }else{ 0 }	
	2.02(1.60; 2.57)	1.98(0.99; 3.98)	58.5	14	if(((Gender != 1) and (Cat at Home = 0)) or (sewage disposal = 1)){ 0 }else{ 1 }	
	2.39(1.88; 3.04)	1.75(0.87; 3.52)	56.9	18	if(((Nutritional Factor1 = 1) or (Gender = 1)) and ((sewage disposal = 1) xor (Tgondi = 0))){ 1 }else{ 0 }	
	2.46(1.94; 3.12)	2.13(1.05; 4.31)	59.2	22	if(((Tgondi = 1) or (sewage disposal = 1)) or ((Gender != 1) and ((Nutritional Factor2 <= 0) xor (sewage disposal = 0))){ 0 }else{ 1 }	
	3.64(2.84; 4.69)	2.92(1.43; 5.96)	63.1	46	if((((Tgondi != 1) or ((Nutritional Factor4 = 1) xor (paving of the street != 1))) xor (sewage disposal != 1)) or ((Gender != 1) and (((Age <= 1) and (Nutritional Factor2 < 1)) xor (sewage disposal = 0)) xor (((Nutritional Factor1 <= 0) xor (Trichiura = 1)) and (num siblings > 2))))) { 0 }else{ 1 }	
	3.89(3.05; 4.98)	2.14(1.06; 4.35)	59.2	58	if((Tgondi = 0) xor ((Nutritional Factor4 <= 0) and (Age < 2))) { if(((Gender = 1) xor (Trichiura = 1)) or ((sewage disposal != 0) xor (Nutritional Factor2 <= 0))){ (if((((BMI = 1) xor (HAV = 0)) or (Nutritional Factor1 >= 1)) xor (sewage disposal != 1)){ 0 }else{ 1 }} }else { 0 }} }else{ (if(Gender != 0){ (if(HZV = 0){ 1 }else{ 0 }) }else{ 0 }) }	

A variable followed by “0” means negative for this variable, and “1” is positive

through the insertion of known relations and the removal of relations that do not make sense.

- MGGP allows the researcher to define groups according to some criteria (such as economic, environmental and nutritional). This type of constraint allows for the definition of different operations upon members of different groups.
- In a single MGGP run, solutions with different levels of complexity can be generated, which improve the understanding of intricate relationships among variables in epidemiological studies.

The use of MGGP performed well compared to RL and C4.5. The application of MGGP in a study focused on asthma and allergies has shown that infections, psychosocial, nutritional, hygiene, and socioeconomic factors may be related in intricate ways with these outcomes. For instance, MGGP showed that the presence of concurrent risk factors for IgE may lead to a reduction in the chances of being IgE positive. This kind of finding could be hardly detected properly using traditional regression based epidemiological techniques.

Endnotes

¹ Given a set of symbols V , then V^* is the set of all strings over symbols in V or the empty string.

² <http://www.cs.waikato.ac.nz/ml/weka/index.html>

Additional file

Additional file 1: List of the best models found by MGGP. List of the best models found by MGGP for Asthma, SPT and IgE. (TXT 112 kb)

Abbreviations

BMI: body mass index; EBV: Epstein-Barr virus; GNI: gross national income; HSV: Herpes simplex virus; HZV: Herpes zoster virus; HAV: Hepatitis A virus; IgE: Immunoglobulin E antibody; IgG: Immunoglobulin G antibody; MGGP: Multiobjective grammar-based genetic programming

Acknowledgements

The authors thank Nicolas Carels for providing the cluster used for computational experiments.

Funding

The data collection has supported by Wellcome Trust (grant 072405/Z/03/Z). The Wellcome Trust was not involved in the design of the study, analysis, and interpretation of data, and in writing the manuscript. The following agencies from the Brazilian government gave support to the researchers: CAPES, CNPq (grant 310778/2013-1), FAPEMIG (grant APQ-03414-15). RVV acknowledges a PhD scholarship from CAPES and is currently recipient of a post-doctorate grant from CNPq (438732/2016-2).

Availability of data and materials

The algorithm MGGP was implemented in c++ and is available on repository: <http://bitbucket.org/ciml-uff/ciml-lib>.

Authors' contributions

RVV, HJCB and HSB contributed to the study design. RVV, HJCB, HSB, CAF, SMAM, NMAM and MLB contributed to the analytical preparation and the writing of the manuscript. JMF, HSB and RVV create MGGP program. All authors read and approved this version of the manuscript.

Ethics approval and consent to participate

Ethical approval was obtained from the Brazilian National Ethical Committee. Written, informed consent detailing all procedures to be carried out on the children was signed by the child's parent or legal guardian.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹Center of Data and Knowledge Integration for Health (CIDACS), Instituto Gonçalo Muniz, Fundação Oswaldo Cruz, Salvador, Brazil. ²Universidade Federal de Juiz de Fora, Juiz de Fora, Minas Gerais, Brazil. ³Laboratório Nacional de Computação Científica, Petrópolis, Rio de Janeiro, Brazil. ⁴Instituto de Saúde Coletiva, Universidade Federal da Bahia, Salvador, Bahia, Brazil. ⁵Instituto de Ciências da Saúde, Universidade Federal da Bahia, Salvador, Bahia, Brazil.

Received: 16 January 2018 Accepted: 4 June 2018

Published online: 26 June 2018

References

- Ilmarinen P, Tuomisto LE, Kankaanranta H. Phenotypes, risk factors, and mechanisms of adult-onset asthma. *Mediat Inflamm*. 2015;2015:1–19.
- Barros AJ, Hirakata VN. Alternatives for logistic regression in cross-sectional studies: an empirical comparison of models that directly estimate the prevalence ratio. *BMC Med Res Methodol*. 2003;3(1):1.
- Domínguez-Almendros S, Benítez-Parejo N, Gonzalez-Ramirez AR. Logistic regression models. *Allergol Immunopathol*. 2011;39(5):295–305.
- Cruz JA, Wishart DS. Applications of Machine Learning in Cancer Prediction and Prognosis. *Cancer Informat*. 2006;2:59–77.
- Mansiaux Y, Carrat F. Detection of independent associations in a large epidemiologic dataset: a comparison of random forests, boosted regression trees, conventional and penalized logistic regression for identifying independent factors associated with H1N1pdm influenza infectio. *BMC Med Res Methodol*. 2014;14(1):99.
- Bornelöv S, Sääf A, Melén E, Bergström A, Torabi Moghadam B, Pulkkinen V, Acevedo N, Orsmark Pietras C, Ege M, Braun-Fahrlander C, Riedler J, Doekes G, Kabesch M, van Hage M, Kere J, Scheynius A, Söderhäll C, Pershagen G, Komorowski J. Rule-based models of the interplay between genetic and environmental factors in childhood allergy. *PLoS ONE*. 2013;8(11):80080.
- Ramezankhani A, Pournik O, Shahabi J, Azizi F, Hadaegh F, Khalili D. The Impact of Oversampling with SMOTE on the Performance of 3 Classifiers in Prediction of Type 2 Diabetes. *Med Dec Making Int J Soc Med Decis Making*. 2014;12(24):1–8.
- Koza JR. Genetic Programming: on the Programming of Computers by Means of Natural Selection, vol. 1. Cambridge: MIT press; 1992.
- Whigham PA, et al. Grammatically-based genetic programming. In: *Proceedings of the Workshop on Genetic Programming: from Theory to Real-world Applications*; 1995. p. 33–41.
- Sosa-Ascencio A, Ochoa G, Terashima-Marin H, Conant-Pablos SE. Grammar-based generation of variable-selection heuristics for constraint satisfaction problems. *Genet Program Evolvable Mach*. 2016;17(2):119–44.
- El Den MB, Poli R. Grammar-based genetic programming for timetabling. In: *Evolutionary Computation, 2009. CEC'09. IEEE Congress On. IEEE*; 2009. p. 2532–9. <https://doi.org/10.1109/CEC.2009.4983259>.
- Bader-El-Den M, Poli R, Fatima S. Evolving timetabling heuristics using a grammar-based genetic programming hyper-heuristic framework. *Memet Comput*. 2009;1(3):205.
- Gupta A, Ong Y-S, Feng L, Tan KC. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Trans Cybern*. 2016;47(7):1652–65.
- Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput*. 2002;6(2):182–97.
- National Heart, Lung, and Blood Institute. Global Initiative for Asthma (GINA). Global Strategy for Asthma Management and Prevention. 2015. Revised 2015. <http://www.ginasthma.org>.
- Marks GB. Environmental factors and gene-environment interactions in the aetiology of asthma. *Clin Exp Pharmacol Physiol*. 2006;33(3):285–9.

17. Asher MI. Recent perspectives on global epidemiology of asthma in childhood. *Allergol Immunopathol*. 2010;38(2):83–7.
18. Cooper PJ, Rodrigues LC, Cruz aa, Barreto ML. Asthma in Latin America: a public health challenge and research opportunity. *Allergy*. 2009;64(1):5–17.
19. Idris IB, Ghazi HF. Environmental Air Pollutants as Risk Factors for Asthma Among Children Seen in Pediatric Clinics in UKMMC, Kuala Lumpur. *Ann Glob Health*. 2016;82(1):202–8.
20. Galobardes B, Granell R, Sterne J, Hughes R, Mejia-Lancheros C, Smith GD, Henderson J. Childhood Wheezing, Asthma, Allergy, Atopy, and Lung Function: Different Socioeconomic Patterns for Different Phenotypes. *Am J Epidemiol*. 2015;182(9):763–74.
21. Chen JH. Asthma and child behavioral skills: Does family socioeconomic status matter? *Soc Sci Med*. 2014;115:38–48.
22. Mendonça LR, Veiga RV, Dattoli VCC, Figueiredo CA, Fiaccone R, Santos J, Cruz AA, Rodrigues LC, Cooper PJ, Pontes-de-Carvalho LC, Barreto ML, Alcantara-Neves NM. Toxocara seropositivity, atopy and wheezing in children living in poor neighbourhoods in urban Latin American. *PLoS Negl Trop Dis*. 2012;6(11):1886.
23. Alcantara-Neves NM, Veiga RV, Dattoli VCC, Fiaccone RL, Esquivel R, Cruz AA, Cooper PJ, Rodrigues LC, Barreto ML. The effect of single and multiple infections on atopy and wheezing in children. *J Allergy Clin Immunol*. 2012;129(2):359–67. doi:10.1016/j.jaci.2011.11.111.
24. Veiga RV, Cunha SS, Dattoli VCC, Cruz AC, Cooper PJ, Rodrigues LC, Barreto ML, Alcantara-Neves NM. Chronic virus infections suppress atopy but not asthma in a set of children from a large Latin American city: a cross-section study. *BMC Pulm Med*. 2011;11(1):24.
25. Daley D. The evolution of the hygiene hypothesis: the role of early-life exposures to viruses and microbes and their relationship to asthma and allergic diseases. *Curr Opin Allergy Clin Immunol*. 2014;14(5):390–6.
26. de Cássia Ribeiro Silva R, Assis AMO, Cruz AA, Fiaccone RL, Dinnocenzo S, Barreto ML, da Silva LA, Rodrigues LC, Alcantara-Neves NM. Dietary Patterns and Wheezing in the Midst of Nutritional Transition: A Study in Brazil. *Pediatric allergy, immunology, and pulmonology*. 2013;26(1):18–24.
27. Matos SMA, Assis AMO, Rodrigues LC. Padrão alimentar, asma e sibilos atópicos e não atópicos em crianças e adolescentes: estudo SCAALA, Salvador, Bahia, Brasil. *Cad Saúde Pública*. 2014;30(9):1849–60.
28. Feitosa CA, Santos DN, Barreto do Carmo MB, Santos LM, Teles CAS, Rodrigues LC, Barreto ML. Behavior problems and prevalence of asthma symptoms among Brazilian children. *J Psychosom Res*. 2011;71(3):160–5.
29. McQuaid EL, Kopel SJ, Nassau JH. Behavioral Adjustment in Children with Asthma: A Meta-Analysis. *Dev Behav Pediatr*. 2001;22(6):430–9.
30. Campbell DE, Boyle RJ, Thornton CA, Prescott SL. Mechanisms of allergic disease - environmental and genetic determinants for the development of allergy. *Clin Exp Allergy*. 2015;45(5):844–58.
31. Costa GNO, Dudbridge F, Fiaccone RL, da Silva TM, Conceição JS, Strina A, Figueiredo CA, Magalhães WCS, Rodrigues MR, Gouveia MH. A genome-wide association study of asthma symptoms in Latin American children. *BMC Genet*. 2015;16(1):1–11.
32. Rabinovitch N, Silveira L, Gelfand EW, Strand M. The response of children with asthma to ambient particulate is modified by tobacco smoke exposure. *Am J Respir Crit Care Med*. 2011;184(12):1350–7.
33. Barreto ML, Cunha SS, Alcântara-Neves N, Carvalho LP, Cruz AA, Stein RT, Genser B, Cooper PJ, Rodrigues LC. Risk factors and immunological pathways for asthma and other allergic diseases in children: background and methodology of a longitudinal study in a large urban center in Northeastern Brazil (Salvador-SCAALA study). *BMC Pulm Med*. 2006;6:15.
34. Barreto ML, Genser B, Strina A, Teixeira MG, Assis AMO, Rego RF, Teles CA, Prado MS, Matos SMA, Santos DN, dos Santos LA, Cairncross S. Effect of city-wide sanitation programme on reduction in rate of childhood diarrhoea in northeast Brazil: assessment by two cohort studies. *Lancet*. 2007;370(9599):1622–8.
35. Onis M. Who child growth standards based on length/height, weight and age. *Acta Paediatr*. 2006;95(S450):76–85.
36. Onis Md, Onyango AW, Borghi E, Siyam A, Nishida C, Siekmann J. Development of a who growth reference for school-aged children and adolescents. *Bull World Health Organ*. 2007;85(9):660–7.
37. Matos SM, Jesus SR, Saldiva SR, Prado MS, D'Innocenzo S, Assis AM, Rodrigues LC, Alcantara-Neves NM, Cruz AA, de Magalhães Simões S, et al. Overweight, asthma symptoms, atopy and pulmonary function in children of 4–12 years of age: findings from the scaala cohort in Salvador, Bahia, Brazil. *Public Health Nutr*. 2011;14(7):1270–8.
38. World Health Organization WHO. A user's guide to the Self Reporting Questionnaire (SRQ). Geneva World Health Organ. 1994;1–84.
39. Mari JJ, Williams P. A validity study of a psychiatric screening questionnaire (SRQ-20) in primary care in the city of Sao Paulo. *Br J Psychiatr*. 1986;148(1):23–6.
40. Barreto do Carmo MB, Neves Santos D, Alves Ferreira Amorim LD, Fiaccone RL, Souza da Cunha S, Cunha Rodrigues L, Barreto ML. Minor psychiatric disorders in mothers and asthma in children. *Soc Psychiatry Psychiatr Epidemiol*. 2009;44(5):416–20.
41. Marques dos Santos L, Neves dos Santos D, Rodrigues L, Barreto M. Maternal mental health and social support: effect on childhood atopic and non-atopic asthma symptoms. *J Epidemiol Community Health*. 2012;66(11):1011–6.
42. Matos SMA, Prado MS, Santos CaST, D'Innocenzo S, Assis AMO, Dourado LS, Oliveira NS, Rodrigues LC, Barreto ML. Validation of a food frequency questionnaire for children and adolescents aged 4 to 11 years living in Salvador, Bahia. *Nutrición Hosp*. 2012;27(4):1114–9.
43. D'Innocenzo S, Marchioni DML, Prado MS, Matos SMA, Pereira SRS, Barros AP, Sampaio LR, Assis AMO, Rodrigues LC, Barreto ML. Condições socioeconômicas e padrões alimentares de crianças de 4 a 11 anos: estudo SCAALA - Salvador/ Bahia. *Revista Brasileira de Saúde Materno Infantil*. 2011;11(1):41–9.
44. Hoffman W, Pons J, Janer J. The sedimentation-concentration method in schistosomiasis mansonii. *Puerto Rico J Publ Hlth*. 1934;9:281–98.
45. Cramer NL. A representation for the adaptive generation of simple sequential programs. In: *Proceedings of the First International Conference on Genetic Algorithms*. Pittsburgh: Carnegie-Mellon University; 1985. p. 183–7.
46. Barbosa HJC, Bernardino HS. Genetic Programming in Civil, Structural and Environmental Engineering. *Computational Technology Reviews*. vol. 4. Civil-Comp; 2011. p. 115–45. <https://doi.org/10.4203/ctr.4.5>; ISSN 2044-8430.
47. Koza JR. Human-competitive results produced by genetic programming. *Genet Program Evolvable Mach*. 2010;11(3–4):251–84.
48. Toulouse M. Automatic quantum computer programming: A genetic programming approach. *Genet Program Evolvable Mach*. 2006;7(1):125–6.
49. Lohn JD, Hornby GS, Linden DS. An evolved antenna for deployment on NASA's space technology 5 mission. In: *Genetic Programming Theory and Practice II*. Springer; 2005. p. 301–15. https://doi.org/10.1007/0-387-23254-0_18.
50. Spector L, Clark DM, Lindsay I, Barr B, Klein J. Genetic programming for finite algebras. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. ACM; 2008. p. 1291–8. <https://doi.org/10.1145/1389095.1389343>; ISBN: 978-1-60558-130-9.
51. Torres RdS, Falcão AX, Gonçalves MA, Papa JP, Zhang B, Fan W, Fox EA. A genetic programming framework for content-based image retrieval. *Pattern Recog*. 2009;42(2):283–92.
52. Uy NQ, Hoai NX, O'Neill M, McKay RI, Galván-López E. Semantically-based crossover in genetic programming: application to real-valued symbolic regression. *Genet Program Evolvable Mach*. 2011;12(2):91–119.
53. Bongard J, Lipson H. Automated reverse engineering of nonlinear dynamical systems. *Proc Natl Acad Sci*. 2007;104(24):9943–8.
54. Yu J, Yu J, Almal AA, Dhanasekaran SM, Ghosh D, Worzel WP, Chinnaiyan AM. Feature selection and molecular classification of cancer using genetic programming. *Neoplasia*. 2007;9(4):292–3.
55. Hong J-H, Cho S-B. The classification of cancer based on dna microarray data that uses diverse ensemble genetic programming. *Artif Intell Med*. 2006;36(1):43–58.
56. Muttill N, Chau K-W. Neural network and genetic programming for modelling coastal algal blooms. *Int J Environ Pollut*. 2006;28(3–4):223–38.
57. Guo H, Nandi AK. Breast cancer diagnosis using genetic programming generated feature. *Pattern Recognit*. 2006;39(5):980–7.
58. Paul TK, Iba H. Prediction of cancer class with majority voting genetic programming classifier using gene expression data. In: *IEEE/ACM Trans Comput Biol Bioinform (TCBB)*; 2009. p. 353–67.
59. Tan KC, Yu Q, Heng C, Lee TH. Evolutionary computing for knowledge discovery in medical diagnosis. *Artif Intell Med*. 2003;27(2):129–54.
60. Wong ML, Lam W, Leung KS, Ngan PS, Cheng JC. Discovering knowledge from medical databases using evolutionary algorithms. *IEEE Eng Med Biol Mag*. 2000;19(4):45–55.

61. Sze-To H-Y, Lee KY, Tso KY, Wong MH, Lee KH, Tang NL, Leung KS. GP-Pi: Using Genetic Programming with Penalization and Initialization on Genome-Wide Association Study. In: International Conference on Artificial Intelligence and Soft Computing. Berlin: Springer; 2013. p. 330–41.
62. Whigham PA. Proc. of the Workshop on Genetic Programming: From Theory to Real-World Applications. In: Rosca JP, editor. Tahoe City, California; 1995. p. 33–41.
63. Chomsky N. Syntactic Structures. Boston: Mouton de Gruyter; 2002.
64. Chawla NV. Data mining for imbalanced datasets: An overview. In: Data mining and knowledge discovery handbook. Boston: Springer; 2009. p. 875–86.
65. Billings S, Voon W. A prediction-error and stepwise-regression estimation algorithm for non-linear systems. *Int J Control*. 1986;44(3):803–22.
66. Quinlan R. C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann Publishers; 1993.
67. Breiman L. Random forests. *Mach Learn*. 2001;45(1):5–32.
68. Lee PH. Resampling methods improve the predictive power of modeling in class-imbalanced datasets. *Int J Environ Res Public Health*. 2014;11(9): 9776–89.
69. Stein RT, Martinez FD. Asthma phenotypes in childhood: Lessons from an epidemiological approach. *Paediatr Respir Rev*. 2004;5(2):155–61.
70. Heissenhuber A, Heinrich J, Fahlbusch B, Borte M, Wichmann H-E, Bolte G. Health impacts of second-hand exposure to cat allergen fel d 1 in infants. *Allergy*. 2003;58(2):154–7.
71. Polk S, Sunyer J, Muñoz-Ortiz L, Barnes M, Torrent M, Figueroa C, Harris J, Vall O, Antó JM, Cullinan P. A prospective study of fel d1 and der p1 exposure in infancy and childhood wheezing. *Am J Respir Crit Care Med*. 2004;170(3):273–8.
72. Celedón JC, Litonjua AA, Ryan L, Platts-Mills T, Weiss ST, Gold DR. Exposure to cat allergen, maternal history of asthma, and wheezing in first 5 years of life. *Lancet*. 2002;360(9335):781–2.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions



Asthma

```

(if((age <= 0) or (cat == 1)){ 1 }else{ 0 })
(if(((age == 0) or (dog == 1)) or (cat == 1)){ 1 }else{ 0 })
(if((((Nutritional Factor1 != 2) or (Mother Psychological disorder == 1)) and
(age <= 0)) or (cat != 0)){ 1 }else{ 0 })
(if((((VZV != 1) or (Mother Psychological disorder == 1)) and (age <= 0)) or
(dog == 1)) or (cat != 0)){ 1 }else{ 0 })
(if((((VZV != 1) or (Mother Psychological disorder == 1)) and (age == 0)) or
(((cat == 1) and (Nutritional Factor3 <= 0)) or (dog != 0))){ 1 }else{ 0 })
(if(((VZV == 0) or (Mother Psychological disorder == 1)) and (age == 0)) or
(((cat != 0) and (Nutritional Factor3 <= 0)) or ((dog != 0) and (Mother
Psychological disorder != 1))){ 1 }else{ 0 })
(if(dog != 0){ 1 }else{ (if(((VZV == 0) and (Nutritional Factor1 != 2)) or
(Mother Psychological disorder != 0)) and (age == 0)){ 1 }else{ (if((cat != 0)
and (Nutritional Factor3 == 0)){ 1 }else{ 0 }) } } })
(if((((Tgondi == 1) or (BMI == 1)) or ((sewage disposal == 1) xor (Nutritional
Factor2 == 0))) and (age == 0)) or (((cat == 1) and (Nutritional Factor3 == 0))
or (dog == 1)){ 1 }else{ 0 })
(if((((Tgondi == 1) or (BMI >= 1)) or ((sewage disposal == 1) xor (Nutritional
Factor2 <= 0))) and (age < 1)) or (((cat != 0) and (Nutritional Factor3 <= 0))
or ((dog != 0) and (Nutritional Factor1 != 2))){ 1 }else{ 0 })
(if((((Tgondi == 1) or (BMI != 0)) or ((sewage disposal == 1) xor (Nutritional
Factor2 == 0))) and (age < 1)) or (((cat == 1) and (Nutritional Factor3 == 0))
or ((dog == 1) and ((Nutritional Factor1 == 1) xor (GNI == 1))){ 1 }else{ 0 })
(if((((Tgondi == 1) or (BMI > 0)) or ((sewage disposal == 1) xor (Nutritional
Factor2 == 0))) and (age < 1)) or (((cat == 1) and (Nutritional Factor3 == 0))
or ((dog == 1) and ((Nutritional Factor1 <= 1) xor ((HAV != 1) and (GNI >=
1))))){ 1 }else{ 0 })
(if((((A_lumbricoides == 1) xor (BMI >= 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 < 1))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (age == 0)) or (((cat == 1) and (Nutritional Factor3 <= 0))
or ((dog == 1) and ((Nutritional Factor1 == 1) xor (GNI == 1))){ 1 }else{ 0 })
(if((((A_lumbricoides != 0) xor (BMI == 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 <= 0))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (age <= 0)) or (((cat == 1) and (Nutritional Factor3 == 0))
or ((dog == 1) and ((Nutritional Factor1 <= 1) xor ((HAV != 1) and (GNI ==
1))))){ 1 }else{ 0 })
(if((((A_lumbricoides == 1) xor (BMI >= 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 <= 0))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (age <= 0)) or (((cat == 1) and (sewage disposal != 1))
and (Nutritional Factor3 == 0)) or ((dog == 1) and ((Nutritional Factor1 <= 1)
xor ((HAV == 0) and (GNI == 1))){ 1 }else{ 0 })
(if((((A_lumbricoides != 1) xor (BMI == 0)) or ((sewage disposal == 1) xor
(Nutritional Factor2 == 0))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (age <= 0)) or (((cat == 1) and (Nutritional Factor3 == 0))
xor (((dog == 1) and ((VZV == 0) or (age < 2))) and ((Nutritional Factor1 <= 1)
xor ((HAV == 0) and (GNI == 1))){ 1 }else{ 0 })
(if((((A_lumbricoides != 0) xor (BMI >= 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 < 1))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (age <= 0)) or (((cat == 1) xor ((A_lumbricoides == 1)
and (Change bed linen == 1))) and (Tgondi == 0)) and (Nutritional Factor3 == 0))
or ((dog == 1) and ((Nutritional Factor1 <= 1) xor ((HAV != 1) and (GNI >=
1))))){ 1 }else{ 0 })
(if((((A_lumbricoides != 0) xor (BMI >= 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 < 1))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (age <= 0)) or (((cat == 1) xor ((A_lumbricoides == 1)
and (Change bed linen == 1))) and (Tgondi == 0)) and (Nutritional Factor3 == 0))
or ((dog == 1) and ((Nutritional Factor1 <= 1) xor ((HAV != 1) and (GNI >=
1))))){ 1 }else{ 0 })
(if((((A_lumbricoides != 1) xor (BMI < 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 < 1))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (age <= 0)) or (((cat == 1) xor ((A_lumbricoides == 1)
and (Change bed linen == 1))) and ((VZV != 1) or (age < 2))) and (Nutritional
Factor3 == 0)) or ((dog == 1) and ((Nutritional Factor1 <= 1) xor ((HAV != 1)
and (GNI == 1))){ 1 }else{ 0 })
(if((((A_lumbricoides != 1) xor (BMI != 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 <= 0))) and (((Daycare != 0) or (Mother Psychological

```

```

disorder == 1)) and (age == 0))) or ((((((cat == 1) and (Tgondi == 0)) xor
((A_lumbricoides == 1) and (Mold/moisture == 1))) and ((cat != 0) or
(Nutritional Factor1 < 2))) and (Nutritional Factor3 == 0)) or ((dog == 1) and
((Nutritional Factor1 <= 1) xor ((HAV != 1) and (GNI == 1))))))){ 1 }else{ 0 }}
(if((((A_lumbricoides == 1) xor (BMI == 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 == 0))) and (((Daycare != 0) or (Mother Psychological
disorder != 0)) and (age <= 0))) or (((((((cat == 1) and (Tgondi == 0)) xor
((A_lumbricoides == 1) and (Nutritional Factor1 < 2))) and ((A_lumbricoides0 ==
1) xor (age < 2))) and (Nutritional Factor3 <= 0)) and (Mold/moisture == 1)) xor
((dog == 1) and ((Nutritional Factor1 < 2) xor ((HAV == 0) and (GNI >= 1))))))){
1 }else{ 0 }}
(if((((A_lumbricoides == 1) xor (BMI == 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 == 0))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (((Nutritional Factor4 >= 1) or (HSV == 1)) and (age <=
0)))) or (((((((cat == 1) and (Tgondi == 0)) xor ((A_lumbricoides == 1) and
(Mold/moisture == 1))) and ((cat == 1) or (Nutritional Factor1 < 2))) and
(Nutritional Factor3 <= 0)) or ((dog != 0) and ((Nutritional Factor1 <= 1) xor
((HAV != 1) and (GNI >= 1))))))){ 1 }else{ 0 }}
(if((((A_lumbricoides == 1) xor (BMI > 0)) or ((sewage disposal == 1) xor
Nutritional Factor2 == 0))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (((Nutritional Factor4 >= 1) or (HSV == 1)) and (age ==
0)))) or (((((((cat == 1) and (Tgondi == 0)) xor ((A_lumbricoides != 0) and
(Mold/moisture == 1))) and (((cat == 1) and (VZV == 0)) or (Nutritional Factor1
!= 2))) and (Nutritional Factor3 == 0)) or ((dog == 1) and ((Nutritional Factor1
< 2) xor ((HAV == 0) and (GNI == 1))))))){ 1 }else{ 0 }}
(if((((cat != 0) and (sewage disposal != 1)) xor ((A_lumbricoides != 0) and
(Change bed linen == 1))) and (((VZV == 0) or (dog == 1) or (age < 2))) and
(Nutritional Factor3 <= 0)) or (((dog == 1) and ((VZV != 1) or (age < 2))) and
((Nutritional Factor1 < 2) xor ((HAV != 1) and (GNI == 1))))){ 1 }else{
(if((((A_lumbricoides != 0) xor (BMI == 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 <= 0))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (age <= 0))) { 1 }else{ 0 }} )}
(if((((cat == 1) and (Tgondi == 0)) xor ((A_lumbricoides != 0) and
(Nutritional Factor1 < 2))) and ((A_lumbricoides0 == 1) xor (age != 2))) and
(Nutritional Factor3 == 0)) and (Mold/moisture == 1)) xor ((dog == 1) and
((Nutritional Factor1 < 2) xor ((HAV == 0) and (GNI >= 1))))){ 1 }else{
(if((((A_lumbricoides == 1) xor (BMI == 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 <= 0))) and (((Nutritional Factor4 == 1) or (HSV == 1)) and
(age < 1))) { (if((Daycare != 0) or (Mother Psychological disorder == 1)) { 1
}else{ 0 }} }else{ 0 }} )}
(if((((cat == 1) and (Tgondi == 0)) xor ((A_lumbricoides == 1) and
((Nutritional Factor1 <= 1) xor (age <= 0)))) and ((sewage disposal != 0) xor
(age < 2))) and (Nutritional Factor3 == 0)) and (Mold/moisture == 1)) xor ((dog
== 1) and ((Nutritional Factor1 != 2) xor ((HAV != 1) and (GNI >= 1))))){ 1
}else{ (if((((A_lumbricoides == 1) xor (BMI != 0)) or ((sewage disposal == 1)
xor (Nutritional Factor2 <= 0))) and (((Nutritional Factor4 == 1) or (HSV == 1))
and (age == 0))) { (if((Daycare != 0) or (Mother Psychological disorder == 1)) { 1
}else{ 0 }} }else{ 0 }} )}
(if((((cat == 1) and (Tgondi == 0)) xor ((A_lumbricoides == 1) and
((Nutritional Factor1 <= 1) xor (age == 0)))) and ((A_lumbricoides0 != 0) xor
(age < 2))) and ((Daily calories <= 2434,17) xor (Nutritional Factor3 >= 1)))
and (Mold/moisture == 1)) xor ((dog == 1) and ((Nutritional Factor1 <= 1) xor
((HAV != 1) and (GNI == 1))))){ 1 }else{ (if((((A_lumbricoides == 1) xor (BMI >=
1)) or ((sewage disposal == 1) xor (Nutritional Factor2 < 1))) and
(((Nutritional Factor4 >= 1) or (HSV == 1)) and (age <= 0))) { (if((Daycare != 0)
or (Mother Psychological disorder == 1)) { 1 }else{ 0 }} }else{ 0 }} )}
(if((((A_lumbricoides != 0) or (BMI == 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 == 0))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (((Nutritional Factor4 == 1) or (HSV == 1)) and (age <=
0)))) or (((((((cat == 1) and (Tgondi == 0)) xor ((A_lumbricoides == 1) and
((Sibling number >= 1) xor (Mold/moisture != 1))) and ((VZV == 0) xor
(A_lumbricoides == 1)) or (Nutritional Factor1 <= 1))) and (Nutritional Factor3
< 1)) or ((dog == 1) and ((Nutritional Factor4 != 1) or (BMI == 0))) and
((Nutritional Factor1 <= 1) xor ((HAV != 1) and (GNI == 1))))))){ 1 }else{ 0 }}
(if((((A_lumbricoides == 1) or (BMI == 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 == 0))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (((Nutritional Factor4 >= 1) or (HSV == 1)) and (age ==
0)))) or (((((((cat != 0) and (Tgondi == 0)) xor ((A_lumbricoides == 1) and

```

```

((Sibling number >= 1) xor (Mold/moisture != 1))) and (((VZV != 1) xor
(A_lumbricoides == 1)) or (Nutritional Factor1 != 2))) and (Nutritional Factor3
== 0)) or ((EBV == 1) and (((dog == 1) and ((Nutritional Factor4 != 1) or (BMI
== 0))) and ((Nutritional Factor1 != 2) xor ((HAV != 1) and (GNI == 1))))))){ 1
}else{ 0 }}
(if((((A_lumbricoides == 1) xor (BMI == 1)) or ((sewage disposal == 1) xor
(Nutritional Factor2 <= 0))) and (((Daycare != 0) or (Mother Psychological
disorder == 1)) and (((Nutritional Factor4 == 1) or (HSV != 0)) and (age < 1))))
or ((((((cat != 0) and (Tgondi == 0)) xor ((A_lumbricoides == 1) and
(((Nutritional Factor2 == 2) or (Sibling number != 0)) xor (Mold/moisture !=
1)))) and (((VZV == 0) xor (A_lumbricoides == 1)) or (Nutritional Factor1 !=
2))) and (Nutritional Factor3 <= 0)) or ((EBV == 1) and (((dog != 0) and
((Nutritional Factor4 < 1) or (BMI == 0))) and ((Nutritional Factor1 <= 1) xor
((HAV == 0) and (GNI == 1))))))){ 1}else{ 0 }}
(if(dog != 0){ (if(Nutritional Factor2 != 2){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 != 0) and (A_lumbricoides != 0))) and (sewage disposal
== 0)){ 1}else{ 0 }} )}else{ (if((sewage disposal != 0) xor (Sibling number >=
3)){ 1}else{ 0 }} )} }else{ (if(((Tgondi == 1) or ((VZV == 0) or (Mother
Psychological disorder == 1))) and (age <= 0)){ (if(((water system == 0) xor
(sewage disposal == 1)) xor (BMI >= 1)) or (((Nutritional Factor1 == 0) xor
(Nutritional Factor4 != 1)) or (Nutritional Factor2 == 0))){ 1}else{ 0 }}
}else{ (if((Nutritional Factor1 <= 1) and (((cat == 1) or ((A_lumbricoides ==
1) and (Mold/moisture == 1))) and ((sewage disposal != 0) xor (age < 2))) and
(Nutritional Factor3 == 0))){ 1}else{ 0 }} )} }
(if(dog == 1){ (if(Nutritional Factor2 <= 1){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides == 1))) and (sewage disposal
== 0)){ 1}else{ 0 }} )}else{ (if((sewage disposal != 0) xor (Sibling number >=
3)){ 1}else{ 0 }} )} }else{ (if(((VZV != 1) or (Mother Psychological disorder
!= 0)) and (age == 0)){ (if(((water system == 0) xor ((sewage disposal == 1) or
((Mold/moisture != 0) and (EBV == 0)))) xor (BMI >= 1)) or (((Nutritional
Factor1 == 0) xor (Nutritional Factor4 <= 0)) or (Nutritional Factor2 <= 0))){ 1
}else{ 0 }} }else{ (if((Nutritional Factor1 <= 1) and (((cat == 1) or
((A_lumbricoides == 1) and (Mold/moisture == 1))) and ((sewage disposal != 0)
xor (age < 2))) and (Nutritional Factor3 <= 0))){ 1}else{ 0 }} )} }
(if(dog != 0){ (if(Nutritional Factor2 < 2){ (if(((Nutritional Factor1 <= 1) xor
((Nutritional Factor1 != 0) and (A_lumbricoides == 1))) and (sewage disposal ==
0)){ 1}else{ 0 }} )}else{ (if((sewage disposal != 0) xor (Sibling number == 3)){
1}else{ 0 }} )} }else{ (if(((Tgondi == 1) or ((VZV == 0) or (Mother
Psychological disorder == 1))) and (age == 0)){ (if(((water system == 0) xor
((sewage disposal == 1) or ((Mold/moisture != 0) and (EBV == 0)))) xor (BMI !=
0)) or (((Nutritional Factor1 == 0) xor (Nutritional Factor4 == 0)) or
(Nutritional Factor2 <= 0))){ 1}else{ 0 }} }else{ (if((Nutritional Factor1 <=
1) and (((cat == 1) or ((A_lumbricoides == 1) and (Mold/moisture == 1))) and
((sewage disposal != 0) xor (age < 2))) and (Nutritional Factor3 <= 0))){ 1
}else{ 0 }} )} }
(if(dog != 0){ (if(Nutritional Factor2 != 2){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 != 0) and (A_lumbricoides == 1))) and (sewage disposal
== 0)){ 1}else{ 0 }} )}else{ (if((sewage disposal != 0) xor (Sibling number ==
3)){ 1}else{ 0 }} )} }else{ (if(((Tgondi == 1) or ((VZV == 0) and (water
system != 0)) or (Mother Psychological disorder == 1))) and (age == 0)){
(if(((water system == 0) or ((sewage disposal == 1) or ((Mold/moisture != 0)
and (EBV == 0)))) xor (BMI > 0)) or (((Nutritional Factor1 <= 0) xor
(Nutritional Factor4 < 1)) or (Nutritional Factor2 == 0))){ 1}else{ 0 }} }else{
(if((Nutritional Factor1 <= 1) and (((cat == 1) or ((A_lumbricoides == 1) and
(Mold/moisture == 1))) and ((sewage disposal != 0) xor (age < 2))) and
(Nutritional Factor3 <= 0))){ 1}else{ 0 }} )} }
(if(dog != 0){ (if(Nutritional Factor2 < 2){ (if(((Nutritional Factor1 <= 1) xor
((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and (sewage disposal ==
0)){ 1}else{ 0 }} )}else{ (if((sewage disposal != 0) xor (Sibling number == 3)){
1}else{ 0 }} )} }else{ (if(((Tgondi == 1) or ((VZV == 0) and (water system !=
0)) or (Mother Psychological disorder == 1))) and (age <= 0)){ (if(((water
system == 0) xor ((sewage disposal == 1) or ((Mold/moisture != 0) and (EBV ==
0)))) xor (BMI >= 1)) or (((Nutritional Factor1 <= 0) xor (Nutritional Factor4 <
1)) or (Nutritional Factor2 <= 0))){ 1}else{ 0 }} }else{ (if((Nutritional
Factor1 != 2) and (((cat == 1) and (Tgondi == 0)) xor ((A_lumbricoides == 1)
and (Mold/moisture == 1))) and ((sewage disposal != 0) xor (age != 2))) and
(Nutritional Factor3 == 0))){ 1}else{ 0 }} )} }
(if(dog != 0){ (if(Nutritional Factor2 != 2){ (if(((Nutritional Factor1 <= 1)

```

```

xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and (sewage disposal
== 0)){ 1 }else{ 0 }} }else{ (if((sewage disposal != 0) xor (Sibling number >=
3)){ 1 }else{ 0 }} )} }else{ (if(((Tgondi != 0) or ((VZV == 0) or (Mother
Psychological disorder == 1))) and (age == 0)){ (if(((water system == 0) or
((sewage disposal == 1) or ((Mold/moisture != 0) and (EBV == 0)))) xor (BMI >=
1)) or (((Nutritional Factor1 == 0) xor (Nutritional Factor4 <= 0)) or
(Nutritional Factor2 == 0))) { 1 }else{ 0 }} }else{ (if((Sibling number != 0) and
((((cat == 1) and (Tgondi == 0)) or ((A_lumbricoides == 1) and (Mold/moisture
== 1)) and (Nutritional Factor1 != 2))) and (((VZV == 0) or (Nutritional Factor4
== 0)) or (age != 2))) and (Nutritional Factor3 == 0))) { 1 }else{ 0 }} )} )}
(if(dog != 0){ (if(Nutritional Factor2 < 2){ (if(((Nutritional Factor1 <= 1) xor
((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and (sewage disposal ==
0)){ 1 }else{ 0 }} }else{ (if((sewage disposal != 0) xor (Sibling number == 3)){
1 }else{ 0 }} )} }else{ (if(((Tgondi == 1) or ((VZV == 0) or (Mother
Psychological disorder == 1))) and (age == 0)){ (if((((Gender == 1) and (water
system != 1)) or ((sewage disposal == 1) or ((Mold/moisture != 0) and (EBV ==
0)))) xor (BMI == 1)) or (((Nutritional Factor1 <= 0) xor (Nutritional Factor4
== 0)) or (Nutritional Factor2 == 0))) { 1 }else{ 0 }} }else{ (if((Sibling number
!= 0) and (((cat == 1) and (Tgondi == 0)) or ((A_lumbricoides == 1) and
(Mold/moisture == 1)) and (Nutritional Factor1 != 2))) and (((VZV == 0) or
(Nutritional Factor4 == 0)) or (age < 2))) and (Nutritional Factor3 <= 0))) { 1
}else{ 0 }} )} )}
(if(dog != 0){ (if(Nutritional Factor2 != 2){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and (age <= 2)) and
((Sibling number == 2) or (sewage disposal == 0))) { 1 }else{ 0 }} }else{
(if((sewage disposal != 0) xor (Sibling number == 3)){ 1 }else{ 0 }} )} }else{
(if(((Tgondi == 1) or ((VZV == 0) or (Mother Psychological disorder == 1))) and
(age <= 0)){ (if(((water system == 0) xor ((sewage disposal == 1) or
((Mold/moisture != 0) and (EBV == 0)))) xor (BMI >= 1)) or (((Nutritional
Factor1 == 0) xor (Nutritional Factor4 <= 0)) or (Nutritional Factor2 == 0))) { 1
}else{ 0 }} }else{ (if((Sibling number != 0) and (((cat == 1) and (Tgondi ==
0)) or ((A_lumbricoides == 1) and (Mold/moisture == 1)) and (Nutritional
Factor1 != 2))) and (((VZV == 0) or (Mother Psychological disorder == 1)) or
(age != 2))) and (Nutritional Factor3 == 0))) { 1 }else{ 0 }} )} )}
(if(dog == 1){ (if(Nutritional Factor2 <= 1){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 != 0) and (A_lumbricoides == 1))) and (sewage disposal
== 0)){ 1 }else{ 0 }} )} }else{ (if((sewage disposal != 0) xor (Sibling number ==
3)){ 1 }else{ 0 }} )} }else{ (if(((Tgondi == 1) or ((VZV == 0) and (water
system != 0)) or (Mother Psychological disorder == 1))) and (age == 0)){
(if(((water system == 0) xor ((sewage disposal == 1) or ((Mold/moisture == 1)
and (EBV == 0)))) xor (BMI == 1)) or (((Nutritional Factor1 <= 0) xor
(Nutritional Factor4 == 0)) or (Nutritional Factor2 <= 0))) { 1 }else{ 0 }}
}else{ (if((Sibling number > 0) and (((cat == 1) or ((A_lumbricoides == 1) and
(Nutritional Factor1 <= 1))) and (((VZV == 0) or (Nutritional Factor4 != 1)) or
(age < 2))) and (Nutritional Factor3 != 1)) and (((VZV == 1) xor (cat == 1))
xor (age <= 0)) or (Change bed linen == 1))) { 1 }else{ 0 }} )} )}
(if(dog == 1){ (if(Nutritional Factor2 <= 1){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and (sewage disposal
== 0)){ 1 }else{ 0 }} )} }else{ (if((sewage disposal != 0) xor (Sibling number ==
3)){ 1 }else{ 0 }} )} }else{ (if(((Tgondi == 1) or ((VZV == 0) and (water
system != 0)) or (Mother Psychological disorder == 1))) and (age == 0)){
(if(((water system == 0) or ((sewage disposal == 1) or ((Mold/moisture == 1)
and (EBV != 1)))) xor (BMI == 1)) or (((Nutritional Factor1 <= 0) xor
(Nutritional Factor4 <= 0)) or (Nutritional Factor2 == 0))) { 1 }else{ 0 }}
}else{ (if((Sibling number > 0) and (((cat == 1) or ((A_lumbricoides == 1) and
((Nutritional Factor1 <= 1) xor (age <= 0)))) and (((VZV == 0) or (Mother
Psychological disorder == 1)) or (age != 2))) and (Nutritional Factor3 <= 0))
and (((VZV == 1) xor (cat == 1)) or (age <= 0)) or (Change bed linen == 1))) {
1 }else{ 0 }} )} )}
(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and (sewage disposal
== 0)){ 1 }else{ 0 }} )} }else{ (if((sewage disposal != 0) xor (Sibling number ==
3)){ 1 }else{ 0 }} )} }else{ (if(((VZV == 0) or (Mother Psychological disorder
== 1)) and (age <= 0)){ (if(((water system == 0) or ((sewage disposal == 1) or
((Mold/moisture != 0) and (EBV == 0)))) xor (BMI == 1)) or (((Nutritional
Factor1 <= 0) xor (Nutritional Factor4 <= 0)) or (Nutritional Factor2 <= 0))) { 1
}else{ 0 }} }else{ (if((Sibling number > 0) and (((cat != 0) or ((Nutritional
Factor1 < 2) and ((Hpilory != 1) xor (Nutritional Factor2 == 0)))) and (((VZV

```

```

== 0) and (dog != 1)) or (age < 2)) or (Mother Psychological disorder == 1)) or
(BMI >= 1))) and (Nutritional Factor3 < 1)) and (((VZV == 1) xor (cat == 1))
xor (age <= 0)) or ((Daycare != 0) and (Change bed linen == 1))))){ 1 }else{ 0
}} } }
(if(dog != 0){ (if(Nutritional Factor2 < 2){ (if(((Nutritional Factor1 <= 1) xor
((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and (sewage disposal ==
0)){ 1 }else{ 0 }} }else{ (if((sewage disposal != 0) xor (Sibling number == 3)){
1 }else{ 0 }} }} }else{ (if(((Tgondi == 1) or ((VZV == 0) or (Mother
Psychological disorder == 1))) and (age <= 0)){ (if(((water system == 0) or
((sewage disposal == 1) or ((Mold/moisture != 0) and (EBV == 0)))) xor (BMI ==
1)) or (((Nutritional Factor1 <= 0) xor (Nutritional Factor4 != 1)) or
(Nutritional Factor2 <= 0))) { 1 }else{ 0 }} }else{ (if((Sibling number > 0) and
((((cat == 1) or ((Nutritional Factor1 <= 1) and ((Hpilory != 1) xor
(Nutritional Factor2 <= 0)))) and (((VZV != 1) or (Mother Psychological
disorder == 1)) or (age < 2)) or (Mother Psychological disorder == 1)) or (BMI
== 1))) and (Nutritional Factor3 < 1)) and (((VZV == 1) xor (cat == 1)) xor
(age <= 0)) or ((Daycare != 0) and (Change bed linen == 1))))){ 1 }else{ 0 }} }
}
(if(dog == 1){ (if(Nutritional Factor2 <= 1){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 != 0) and (A_lumbricoides == 1))) and (sewage disposal
== 0)){ 1 }else{ 0 }} }else{ (if((sewage disposal != 0) xor (Sibling number ==
3)){ 1 }else{ 0 }} }} }else{ (if(((Tgondi == 1) or ((VZV == 0) and (water
system != 0)) or (Mother Psychological disorder == 1))) and (age == 0)){
(if(((water system == 0) xor ((sewage disposal == 1) or ((Mold/moisture == 1)
and (EBV == 0)))) xor (BMI == 1)) or ((Nutritional Factor1 == 0) xor
(Nutritional Factor4 == 0)) or (Nutritional Factor2 <= 0))) { 1 }else{ 0 }}
}else{ (if((Sibling number != 0) and (((cat == 1) or ((Nutritional Factor1 <=
1) and ((Hpilory != 1) xor (Nutritional Factor2 <= 0)))) and (((VZV != 1) or
(Mother Psychological disorder == 1)) or (age < 2)) or (Nutritional Factor3 >=
2)) or (BMI == 1))) and (Nutritional Factor3 != 1)) and (((VZV == 1) xor (cat
== 1)) xor (age <= 0)) or ((Daycare != 0) and (Change bed linen == 1))))){ 1
}else{ 0 }} } }
(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and (BMI < 2)) and
((Sibling number == 2) or (sewage disposal == 0))) { 1 }else{ 0 }} }else{
(if((sewage disposal != 0) xor (Sibling number == 3)){ 1 }else{ 0 }} }} }else{
(if(((Tgondi != 0) or ((VZV == 0) or (Mother Psychological disorder == 1))) and
(age <= 0)){ (if((((Gender == 1) and (water system != 1)) or ((sewage disposal
== 1) or ((Mold/moisture == 1) and (EBV == 0)))) xor (BMI == 1)) or
(((Nutritional Factor1 <= 0) xor (Nutritional Factor4 <= 0)) or (Nutritional
Factor2 <= 0))) { 1 }else{ 0 }} }else{ (if((Sibling number != 0) and (((cat ==
1) or ((Nutritional Factor1 <= 1) and ((Hpilory != 1) xor (Nutritional Factor2
<= 0)))) and (((VZV == 0) or (age < 2)) or (Mother Psychological disorder ==
1)) or (BMI == 1))) and (Nutritional Factor3 < 1)) and (((VZV == 1) xor (cat ==
1)) xor (age <= 0)) or ((Daycare != 0) and (Change bed linen == 1))))){ 1 }else{
0 }} } }
(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and (BMI < 2)) and
((Sibling number == 2) or (sewage disposal == 0))) { 1 }else{ 0 }} }else{
(if((sewage disposal != 0) xor (Sibling number == 3)){ 1 }else{ 0 }} }} }else{
(if(((Tgondi != 0) or ((VZV == 0) or (Mother Psychological disorder == 1))) and
(age <= 0)){ (if((((Gender == 1) and (water system != 1)) or ((sewage disposal
== 1) or ((Mold/moisture == 1) and (EBV == 0)))) xor (BMI == 1)) or
(((Nutritional Factor1 <= 0) xor (Nutritional Factor4 <= 0)) or (Nutritional
Factor2 <= 0))) { 1 }else{ 0 }} }else{ (if((Sibling number != 0) and (((cat ==
1) or ((Nutritional Factor1 <= 1) and ((Hpilory != 1) xor (Nutritional Factor2
<= 0)))) and (((VZV == 0) or (age < 2)) or (Mother Psychological disorder ==
1)) or (BMI == 1))) and (Nutritional Factor3 < 1)) and (((VZV == 1) xor (cat ==
1)) xor (age <= 0)) or ((Daycare != 0) and (Change bed linen == 1))))){ 1 }else{
0 }} } }
(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and ((age == 1) or
((GNI <= 0) and (Daily calories <= 2434,17)) xor (Sibling number > 1))) and
((Sibling number == 2) or (sewage disposal == 0))) { 1 }else{ 0 }} }else{
(if((sewage disposal != 0) xor (Sibling number == 3)){ 1 }else{ 0 }} }} }else{
(if(((Tgondi != 0) or ((VZV == 0) or (Mother Psychological disorder == 1))) and
(age == 0)){ (if(((water system == 0) or ((sewage disposal == 1) or
((Mold/moisture == 1) and (EBV == 0)))) xor (BMI == 1)) or ((Nutritional

```

```

Factor1 <= 0) xor (Nutritional Factor4 <= 0)) or (Nutritional Factor2 <= 0))) { 1
}else{ 0 }} }else{ (if((Sibling number > 0) and (((cat == 1) or ((Nutritional
Factor1 <= 1) and ((Hpilory != 1) xor (Nutritional Factor2 <= 0)))) and (((VZV
== 0) or (Nutritional Factor3 == 1)) or (age < 2)) or (Mother Psychological
disorder == 1)) or (BMI == 1))) and (Nutritional Factor3 != 1)) and (((VZV ==
1) xor (cat != 0)) xor (age == 0)) or ((Daycare != 0) and (Change bed linen ==
1)))) { 1 }else{ 0 }} } }
(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if((((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and ((age == 1) or
(((GNI == 0) and (sewage disposal == 0)) xor (Sibling number > 1)))) and
((Sibling number == 2) or (sewage disposal == 0))) { 1 }else{ 0 }} }else{
(if((sewage disposal != 0) xor (Sibling number >= 3)) { 1 }else{ 0 }} } } }else{
(if(((Tgondi != 0) or ((VZV == 0) or (Mother Psychological disorder == 1))) and
(age <= 0)) { (if((((Gender == 1) and (water system != 1)) or ((sewage disposal
== 1) or ((Mold/moisture == 1) and (EBV == 0)))) xor (BMI == 1)) or
(((Nutritional Factor1 <= 0) xor (Nutritional Factor4 <= 0)) or (Nutritional
Factor2 <= 0))) { 1 }else{ 0 }} }else{ (if((Sibling number != 0) and (((cat ==
1) or ((Nutritional Factor1 <= 1) and ((Hpilory != 1) xor (Nutritional Factor2
<= 0)))) and (((VZV != 1) or (Mother Psychological disorder == 1)) or (age <
2)) or (Mother Psychological disorder == 1)) or (BMI == 1)) and (Nutritional
Factor3 != 1)) and (((VZV == 1) xor (cat == 1)) xor (age == 0)) or ((Daycare !=
0) and (Change bed linen == 1)))) { 1 }else{ 0 }} } }
(if(dog == 1){ (if(Nutritional Factor2 < 2){ (if(((Nutritional Factor3 < 2) and
((age == 1) or ((GNI == 0) and (sewage disposal == 0)) xor (Sibling number >
1)))) and ((age <= 0) or ((cat == 1) and (Nutritional Factor3 == 0)) or
((Nutritional Factor1 < 2) xor ((HAV != 1) and (GNI >= 1)))) { 1 }else{ 0 }}
}else{ (if((sewage disposal != 0) xor (Sibling number == 3)) { 1 }else{ 0 }} }
}else{ (if(((Tgondi != 0) or ((VZV == 0) or (Mother Psychological disorder ==
1))) and (age == 0)) { (if((((Gender == 1) and (water system != 1)) xor ((sewage
disposal == 1) or ((Mold/moisture == 1) and (EBV == 0)))) xor (BMI == 1)) or
(((Nutritional Factor1 <= 0) xor (Nutritional Factor4 == 0)) or (Nutritional
Factor2 <= 0))) { 1 }else{ 0 }} }else{ (if((Sibling number != 0) and (((cat ==
1) or ((Nutritional Factor1 <= 1) and ((Hpilory != 1) xor (Nutritional Factor2
== 0)))) and (((VZV != 1) or (Daily calories >= 2434,17)) or (age != 2)) or
(Mother Psychological disorder == 1)) or (BMI == 1)) and (Nutritional Factor3
!= 1)) and (((VZV == 1) xor (cat == 1)) xor (age == 0)) or ((Daycare != 0) and
(Change bed linen == 1)))) { 1 }else{ 0 }} } }
(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if((((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and ((age == 1) or
(((GNI == 0) and (sewage disposal == 0)) xor (Sibling number > 1)))) and ((age
<= 0) or ((cat == 1) or ((Nutritional Factor1 < 2) xor ((HAV != 1) and (GNI >=
1)))) { 1 }else{ 0 }} }else{ (if((sewage disposal != 0) xor (Sibling number ==
3)) { 1 }else{ 0 }} } } }else{ (if(((Tgondi != 0) or ((VZV == 0) or (Mother
Psychological disorder == 1))) and (age <= 0)) { (if((((Gender == 1) and (water
system != 1)) xor ((sewage disposal == 1) or ((Mold/moisture == 1) and (EBV ==
0)))) xor (BMI == 1)) or (((Nutritional Factor1 <= 0) xor (Nutritional Factor4
<= 0)) or (Nutritional Factor2 == 0))) { 1 }else{ 0 }} }else{ (if((Sibling number
!= 0) and (((cat == 1) or ((Nutritional Factor1 <= 1) and ((Hpilory != 1) xor
(Nutritional Factor2 <= 0)))) and (((VZV == 0) or (Sibling number == 0)) or
(age < 2)) or (Mother Psychological disorder == 1)) or (BMI == 1)) and
(Nutritional Factor3 != 1)) and (((VZV == 1) xor (cat == 1)) xor (age <= 0)) or
((Daycare != 0) and (Change bed linen == 1)))) { 1 }else{ 0 }} } }
(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if((((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 != 0) and (A_lumbricoides != 0))) and ((age == 1) or
(((GNI == 0) xor (sewage disposal != 0)) xor (Sibling number > 1)))) and
(((Sibling number <= 2) and (age == 0)) or ((cat == 1) or ((Nutritional Factor1
<= 1) xor ((HAV != 1) and (GNI == 1)))) { 1 }else{ 0 }} }else{ (if((sewage
disposal != 0) xor (Sibling number == 3)) { 1 }else{ 0 }} } } }else{ (if(((Tgondi
!= 0) or ((VZV == 0) or (Mother Psychological disorder == 1))) and (age <= 0)) {
(if((((Gender == 1) and (water system != 1)) xor ((sewage disposal == 1) or
((Mold/moisture == 1) and (EBV == 0)))) xor (BMI == 1)) or (((Nutritional
Factor1 <= 0) xor (Nutritional Factor4 == 0)) or (Nutritional Factor2 <= 0))) { 1
}else{ 0 }} }else{ (if((Sibling number != 0) and (((cat == 1) or ((Nutritional
Factor1 <= 1) and ((Hpilory != 1) xor (Nutritional Factor2 <= 0)))) and (((VZV
== 0) or (Nutritional Factor3 > 2)) or (age < 2)) or (Mother Psychological
disorder == 1)) or (BMI >= 1)) and (Nutritional Factor3 != 1)) and (((VZV ==
1) xor (cat == 1)) xor (age == 0)) or ((Daycare != 0) and (Change bed linen ==
1)))) { 1 }else{ 0 }} } }

```

```

(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if(((Daily calories < 2434,17)
and ((age == 1) or (((GNI == 0) and (sewage disposal == 0)) xor (Sibling number
> 1)))) and (((age <= 0) xor (Daily calories >= 1692,98)) and (Nutritional
Factor4 != 0)) or (((cat == 1) and (Nutritional Factor2 != 2)) or ((Nutritional
Factor1 < 2) xor ((HAV != 1) and (GNI >= 1))))))){ 1 }else{ 0 }} }else{
(if((sewage disposal != 0) xor (Sibling number == 3)){ 1 }else{ 0 }} ) }else{
(if(((Tgondi != 0) or ((VZV == 0) or (Mother Psychological disorder == 1))) and
(age == 0)){ (if((((Gender == 1) and (water system != 1)) or ((sewage disposal
== 1) or ((Mold/moisture == 1) and (EBV == 0)))) xor (BMI == 1)) or
(((Nutritional Factor1 <= 0) xor (Nutritional Factor4 <= 0)) or (Nutritional
Factor2 == 0))) { 1 }else{ 0 }} }else{ (if((Sibling number != 0) and (((cat ==
1) or ((Nutritional Factor1 < 2) and ((Hpilory != 1) xor (Nutritional Factor2 <=
0)))) and (((VZV == 0) xor (BMI > 0)) or (cat == 1)) or ((HSV == 0) and (GNI
>= 1)) or (BMI == 1)) and (Nutritional Factor3 < 1)) and (((VZV == 1) xor
(cat == 1)) xor (age <= 0)) or ((Daycare != 0) and (Change bed linen == 1))))){
1 }else{ 0 }} ) } }
(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and ((age == 1) or
(((GNI == 0) and (sewage disposal == 0)) xor (Sibling number > 1)))) and
(((Nutritional Factor1 != 0) and ((Mother Psychological disorder != 1) and (age
< 1))) or (((cat == 1) and (Nutritional Factor3 <= 0)) or ((Nutritional Factor1
< 2) xor ((HAV != 1) and (GNI >= 1))))))){ 1 }else{ 0 }} }else{ (if((sewage
disposal != 0) xor (Sibling number == 3)){ 1 }else{ 0 }} ) }else{ (if(((Tgondi
!= 0) or ((VZV == 0) or (Mother Psychological disorder == 1))) and (age <= 0)){
(if((((Gender == 1) and (water system != 1)) or ((sewage disposal == 1) or
((Mold/moisture == 1) and (EBV == 0)))) xor (BMI == 1)) or (((Nutritional
Factor1 <= 0) xor (Nutritional Factor4 <= 0)) or (Nutritional Factor2 == 0))) { 1
}else{ 0 }} }else{ (if((Sibling number != 0) and (((cat == 1) or ((Nutritional
Factor1 <= 1) and ((Hpilory != 1) xor (Nutritional Factor2 <= 0)))) and (((VZV
== 0) or (Daily calories >= 2434,17)) or (age < 2)) or (Mother Psychological
disorder == 1)) or (BMI == 1))) and (Nutritional Factor3 < 1)) and (((VZV == 1)
xor (cat == 1)) xor (age <= 0)) or ((Daycare != 0) and (Change bed linen ==
1))))){ 1 }else{ 0 }} ) } }
(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and ((age == 1) or
(((GNI == 0) and (sewage disposal == 0)) xor (Sibling number > 1)))) and
(((Nutritional Factor1 != 0) and ((Mother Psychological disorder != 1) and (age
< 1))) or (((cat == 1) and (Nutritional Factor3 <= 0)) or ((Nutritional Factor1
< 2) xor ((HAV != 1) and (GNI >= 1))))))){ 1 }else{ 0 }} }else{ (if((sewage
disposal != 0) xor (Sibling number == 3)){ 1 }else{ 0 }} ) }else{ (if(((Tgondi
!= 0) or ((VZV == 0) or (Mother Psychological disorder == 1))) and (age <= 0)){
(if((((Gender == 1) and (water system != 1)) or ((sewage disposal == 1) or
((Mold/moisture == 1) and (EBV == 0)))) xor (BMI == 1)) or (((Nutritional
Factor1 <= 0) xor (Nutritional Factor4 <= 0)) or (Nutritional Factor2 == 0))) { 1
}else{ 0 }} }else{ (if((Sibling number != 0) and (((cat == 1) or ((Nutritional
Factor1 <= 1) and ((Hpilory != 1) xor (Nutritional Factor2 <= 0)))) and (((VZV
== 0) or (Daily calories >= 2434,17)) or (age < 2)) or (Mother Psychological
disorder == 1)) or (BMI == 1))) and (Nutritional Factor3 < 1)) and (((VZV == 1)
xor (cat == 1)) xor (age <= 0)) or ((Daycare != 0) and (Change bed linen ==
1))))){ 1 }else{ 0 }} ) } }
(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and ((age == 1) or
(((GNI == 0) and (sewage disposal == 0)) xor (Sibling number > 1)))) and (((age
<= 0) xor (Daily calories >= 1692,98)) and (Nutritional Factor4 != 0)) or (((cat
== 1) and (Nutritional Factor2 != 2)) or ((Nutritional Factor1 < 2) xor ((HAV !=
1) and (GNI >= 1))))))){ 1 }else{ 0 }} }else{ (if((sewage disposal != 0) xor
(Sibling number == 3)){ 1 }else{ 0 }} ) }else{ (if(((Tgondi != 0) or ((VZV ==
0) or (Mother Psychological disorder == 1))) and (age == 0)){ (if((((Gender ==
1) and (water system != 1)) or ((sewage disposal == 1) or ((Mold/moisture == 1)
and (EBV == 0)))) xor (BMI == 1)) or (((Nutritional Factor1 <= 0) xor
(Nutritional Factor4 <= 0)) or (Nutritional Factor2 == 0))) { 1 }else{ 0 }}
}else{ (if((Sibling number != 0) and (((cat == 1) or ((Nutritional Factor1 <
2) and ((Hpilory != 1) xor (Nutritional Factor2 <= 0)))) and (((VZV == 0) xor
(BMI > 0)) or (cat == 1)) or ((HSV == 0) and (GNI >= 1)) or (BMI == 1))) and
(Nutritional Factor3 < 1)) and (((VZV == 1) xor (cat == 1)) xor (age <= 0)) or
((Daycare != 0) and (Change bed linen == 1))))){ 1 }else{ 0 }} ) } }
(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if(((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and ((age == 1) or

```

```

((((GNI == 0) and (sewage disposal == 0)) xor (Sibling number > 1)))) and (((age
<= 0) xor (Daily calories >= 1692,98)) and ((Mother Psychological disorder != 1)
and (dog == 1))) or (((cat == 1) and (Nutritional Factor3 == 0)) or
((Nutritional Factor1 < 2) xor ((HAV != 1) and (GNI >= 1))))){ 1 }else{ 0 }
}else{ (if((sewage disposal != 0) xor (Sibling number == 3)){ 1 }else{ 0 } )}
}else{ (if(((Tgondi != 0) or ((VZV == 0) or (Mother Psychological disorder ==
1))) and (age == 0)){ (if((((water system == 0) xor ((sewage disposal == 1) or
(Mold/moisture == 1) and (EBV == 0)))) xor (BMI == 1)) or (((Nutritional
Factor1 <= 0) xor (Nutritional Factor4 == 0)) or (Nutritional Factor2 == 0))) { 1
}else{ 0 } }else{ (if((Sibling number != 0) and (((cat == 1) or ((Nutritional
Factor1 < 2) and ((Hpilory != 1) xor (Nutritional Factor2 <= 0)))) and (((VZV
== 0) xor (Daily calories > 1692,98)) or ((Nutritional Factor2 == 2) and
((((cat == 1) or (A_lumbricoides == 1) and (Nutritional Factor1 <= 1))) and
(dog == 1) or (age != 2))) and (Daily calories <= 2434,17)) and (Daily calories
< 2434,17)))) or ((HSV == 0) and (GNI >= 1))) or (BMI == 1))) and (Nutritional
Factor3 < 1)) and (((VZV == 0) xor (cat == 1)) xor (age != 0)) or ((Daycare !=
0) and (Change bed linen == 1))))){ 1 }else{ 0 } } }
(if(dog == 1){ (if(Nutritional Factor2 != 2){ (if((((Nutritional Factor1 <= 1)
xor ((Nutritional Factor1 > 0) and (A_lumbricoides != 0))) and ((age == 1) or
(((GNI == 0) and (sewage disposal == 0)) xor (Sibling number > 1)))) and (((age
<= 0) xor (Daily calories >= 1692,98)) and ((Mother Psychological disorder != 1)
and (dog == 1))) or (((cat == 1) and (Nutritional Factor3 == 0)) or
((Nutritional Factor1 < 2) xor ((HAV != 1) and (GNI >= 1))))){ 1 }else{ 0 }
}else{ (if((sewage disposal != 0) xor (Sibling number == 3)){ 1 }else{ 0 } )}
}else{ (if(((Tgondi != 0) or ((VZV == 0) or (Mother Psychological disorder ==
1))) and (age == 0)){ (if((((Gender == 1) and (water system != 1)) or ((sewage
disposal == 1) or ((Mold/moisture == 1) and (EBV == 0)))) xor (BMI == 1)) or
(((Nutritional Factor1 <= 0) xor (Nutritional Factor4 == 0)) or (Nutritional
Factor2 == 0))) { 1 }else{ 0 } }else{ (if((Sibling number != 0) and (((cat ==
1) or ((Nutritional Factor1 < 2) and ((Hpilory != 1) xor (Nutritional Factor2 <=
0)))) and (((VZV == 0) xor (Daily calories > 1692,98)) or ((Nutritional
Factor2 == 2) and (((cat == 1) or (A_lumbricoides == 1) and (Nutritional
Factor1 <= 1))) and ((dog == 1) or (age != 2))) and (Daily calories <= 2434,17))
and (Daily calories < 2434,17)))) or ((HSV == 0) and (GNI >= 1))) or (BMI ==
1))) and (Nutritional Factor3 < 1)) and (((VZV == 0) xor (cat == 1)) xor (age
!= 0)) or ((Daycare != 0) and (Change bed linen == 1))))){ 1 }else{ 0 } } }

```

SPT

```

(if((Change bed linen != 0) and ((Nutritional Factor3 >= 1) or ((Sibling number
== 2) xor ((age >= 2) or (HSV == 0))))){ 1 }else{ 0 }
(if((Change bed linen != 0) and ((Nutritional Factor3 > 0) or ((Sibling number
== 2) xor (((Fly at home == 1) and (Daycare != 0)) or (HSV == 0))))){ 1 }else{ 0
})
(if((((Change bed linen == 0) and (Nutritional Factor1 != 0)) xor ((Nutritional
Factor2 >= 1) or (Gender != 0))) and ((Sibling number == 2) xor ((age > 1) or
(HSV == 0))))){ 1 }else{ 0 }
(if((((Change bed linen == 0) and ((Nutritional Factor1 > 0) or (Gender == 1)))
xor ((Nutritional Factor2 != 0) or (Gender != 0))) and (((age < 2) and (HSV !=
0)) xor (Sibling number != 2)))){ 1 }else{ 0 }
(if((Change bed linen == 0) xor ((Nutritional Factor2 >= 1) or (Gender != 0)))
{
  (if(((HSV != 0) and (age != 2)) and ((age > 0) or (Fly at home == 0))) xor
  (Sibling number == 2)){ 0 }else{ 1 } }
  }else{ 0 }
  (if((Change bed linen == 0) xor ((Nutritional Factor2 >= 1) or (Gender == 1)))
  and ((Sibling number == 2) xor (((HAV == 0) xor ((age != 2) and ((age >= 1) or
  (Fly at home == 0)))) or (HSV == 0))))){ 1 }else{ 0 }
  (if((((Change bed linen == 0) and (Nutritional Factor1 > 0)) xor ((Nutritional
  Factor2 != 0) or (Gender == 1))) and ((Sibling number == 2) xor (((HAV == 0) xor
  ((Daycare == 0) and (age != 1)) xor (age != 2))) or (HSV != 1))))){ 1 }else{ 0
  })
  (if((((Change bed linen == 0) and ((Nutritional Factor1 != 0) or (Gender != 0)))
  xor ((Nutritional Factor2 != 0) or (Gender != 0))) and ((Sibling number == 2)
  xor (((HAV == 0) xor ((age != 2) and ((age == 1) or (Fly at home != 1)))) or
  (HSV == 0))))){ 1 }else{ 0 }
  (if((((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or ((Hpilory == 1)
  or (Gender != 0)))) xor ((Nutritional Factor2 > 0) or (Gender != 0))) and
  ((Sibling number == 2) xor (((HAV == 0) xor ((age != 2) and ((age == 1) or (Fly
  at home == 0)))) or (HSV == 0))))){ 1 }else{ 0 }

```

```

(if((((Change bed linen != 0) xor ((Nutritional Factor1 > 0) or ((sewage
disposal != 1) xor (Gender == 0)))) xor ((Nutritional Factor2 != 0) or (Gender
!= 0))) and ((Sibling number == 2) xor (((HAV != 0) xor (((Nutritional Factor1
!= 2) and ((pavimenta != 1) and (Daycare == 0))) xor (age == 2))) or (HSV ==
0)))){ 1 }else{ 0 })
(if((((Change bed linen != 0) xor ((Nutritional Factor1 > 0) or ((sewage
disposal != 1) xor (Gender != 1)))) xor ((Nutritional Factor2 != 0) or (Gender
!= 0))) and ((Sibling number == 2) xor (((HAV != 0) xor (((sewage disposal !=
0) or (Nutritional Factor1 != 2)) and ((pavimenta == 0) and (Daycare == 0))) xor
(age >= 2))) or (HSV == 0)))){ 1 }else{ 0 })
(if((((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or ((sewage
disposal == 1) xor ((cat != 1) xor (Gender != 1)))) xor ((Nutritional Factor2 >
0) or (Gender != 0))) and ((Sibling number == 2) xor (((HAV == 1) xor (((Gender
!= 0) or (Nutritional Factor1 != 2)) and ((pavimenta == 0) and (Daycare == 0)))
xor (age >= 2))) or (HSV == 0)))){ 1 }else{ 0 })
(if((((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or ((sewage
disposal == 1) xor ((cat != 1) xor (Gender != 1)))) xor ((Nutritional Factor2 >
0) or (Gender != 0))) and ((Sibling number == 2) xor (((HAV == 1) xor
((((Nutritional Factor3 >= 1) and (age < 2)) or (Nutritional Factor1 != 2)) and
((pavimenta == 0) and (Daycare == 0))) xor (age == 2))) or (HSV == 0)))){ 1
}else{ 0 })
(if((((Change bed linen != 0) xor ((Nutritional Factor1 > 0) or ((sewage
disposal != 1) xor (Gender == 0)))) xor ((Nutritional Factor2 > 0) or (Gender !=
0))) and ((Sibling number == 2) xor (((HAV != 1) xor ((age != 2) and ((age ==
1) or (Fly at home == 0)))) or ((HAV == 0) xor ((Nutritional Factor1 < 2) and
((pavimenta != 1) and (Daycare == 0))) xor (age != 2))) or (HSV == 0)))){ 1
}else{ 0 })
(if((((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or ((sewage
disposal != 1) xor (Gender != 1)))) xor ((Nutritional Factor2 != 0) or (Gender
!= 0))) and ((Sibling number == 2) xor (((HAV == 0) xor ((HAV != 1) and ((age
== 1) or (Fly at home != 1))) or ((HAV != 0) xor (((Nutritional Factor3 == 1)
or (Nutritional Factor1 <= 1)) and ((pavimenta == 0) and (Daycare == 0))) xor
(age == 2))) or (HSV == 0)))){ 1 }else{ 0 })
(if((((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or ((sewage
disposal != 1) xor (Gender != 1)))) xor ((Nutritional Factor2 != 0) or (Gender
!= 0))) and ((Sibling number == 2) xor (((HAV == 0) xor ((HAV != 1) and ((age
== 1) or (Fly at home != 1))) or ((HAV != 0) xor (((Nutritional Factor3 == 1)
or (Nutritional Factor1 <= 1)) and ((pavimenta == 0) and (Daycare == 0))) xor
(age == 2))) or (HSV == 0)))){ 1 }else{ 0 })
(if((((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or ((Gender != 0)
xor ((Nutritional Factor2 < 1) xor (Sibling number > 2)))) xor ((Nutritional
Factor2 != 0) or (Gender != 0))) and ((Sibling number == 2) xor (((HAV == 0) xor
((age != 2) and ((age >= 1) or (Fly at home != 1))) or (HSV == 0)))){ 1 }else{
(if((Sibling number < 2) and (HSV != 0)){ (if((Nutritional Factor2 <= 1) and
((pavimenta == 0) and (Daycare == 0))) xor (Nutritional Factor3 <= 1)){ 0 }else{
1 }} }else{ 0 }) })
(if((((Change bed linen != 0) xor ((Nutritional Factor1 > 0) or (((cat != 1) xor
(Gender != 1) or ((Nutritional Factor2 < 1) xor (Sibling number > 2)))) xor
(Nutritional Factor2 > 0) or (Gender != 0))) and ((Sibling number == 2) xor
(((HAV == 0) xor ((age != 2) and ((age == 1) or (Fly at home != 1))) or (HSV ==
0)))){ 1 }else{ (if((Sibling number < 2) and (HSV != 0)){ (if((Nutritional
Factor2 <= 1) and ((pavimenta == 0) and (Daycare == 0))) xor (Sibling number !=
2)){ 0 }else{ 1 }} }else{ 0 }) })
(if((((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or ((Gender != 0)
or (Nutritional Factor3 > 0)))) xor ((Nutritional Factor2 > 0) or (Gender !=
0))) and ((Sibling number == 2) xor (((HAV == 0) xor ((age < 2) and ((age >= 1)
or (Fly at home != 1))) or (HSV == 0)))){ 1 }else{ (if((Sibling number <= 2)
and (HSV != 0)){ (if((Nutritional Factor2 <= 1) and ((pavimenta == 0) and
(Daycare == 0))) xor ((Daily calories <= 2434,17) xor ((Mold/moisture != 0) and
((Change bed linen == 0) and (age > 1)))){ 0 }else{ 1 }} }else{ 0 }) })
(if((((Change bed linen != 0) xor ((Sibling number == 2) or ((Nutritional
Factor1 != 0) or ((Nutritional Factor3 == 0) xor (Gender != 1)))) xor
((Nutritional Factor2 != 0) or (Gender != 0))) and ((Sibling number == 2) xor
(((HAV == 0) xor ((age <= 1) and ((age == 1) or (Fly at home != 1))) or (HSV ==
0)))){ 1 }else{ (if((Sibling number < 3) and (HSV != 0)){ (if((Nutritional
Factor2 <= 1) and ((pavimenta == 0) and (Daycare == 0))) xor ((fuma != 0) xor
((Mold/moisture != 0) and ((Change bed linen == 0) and (age > 1)))){ 0 }else{ 1
}} }else{ 0 }) })

```

```

(if((((Change bed linen != 0) xor ((Mold/moisture != 1) or ((Nutritional Factor1
!= 0) or ((Nutritional Factor3 <= 0) xor (Gender != 1)))))) xor ((Nutritional
Factor2 > 0) or (Gender != 0))) and ((Sibling number == 2) xor (((HAV == 0) xor
((age != 2) and ((age >= 1) or (Fly at home == 0)))) or (HSV == 0)))){ 1 }else{
(if((Sibling number <= 2) and (HSV != 0)){ (if(((Nutritional Factor2 <= 1) and
((pavimenta == 0) and ((age <= 2) and (Daycare == 0)))) xor ((fuma != 0) xor
((Mold/moisture != 0) and ((Change bed linen == 0) and (age > 1)))))) { 0 }else{ 1
} }else{ (if(Daily calories < 1692,98){ 0 }else{ 0 } } } )
(if((((Change bed linen != 0) and ((age > 1) or ((GNI != 0) or ((Sibling number
>= 2) or (HSV == 0)))))) and ((age != 1) or ((age < 0) or (((Sibling number < 2)
or (Fly at home == 0)) and ((Gender == 1) or (pavimenta == 1)))))) and (((dog ==
1) xor (Mold/moisture == 1)) or ((Nutritional Factor4 <= 0) xor (Nutritional
Factor2 >= 1)))){ (if((Nutritional Factor3 == 0) and ((HSV != 0) and (Fly at
home != 1)) xor (Sibling number == 2)) { 0 }else{ 1 } }else{ (if(((Sibling
number <= 2) xor ((Nutritional Factor1 > 1) and (BMI < 1))) and (HSV == 1)){
(if((Daycare == 0) xor (age <= 1)) { 0 }else{ 1 } } }else{ 0 } } }
(if((((Change bed linen != 0) and ((age > 1) or ((GNI > 0) or ((Sibling number
>= 2) or (HSV == 0)))))) and (((age != 1) or (Gender == 1) or (pavimenta != 0)))
and (((dog == 1) xor (Mold/moisture == 1)) or ((Nutritional Factor4 == 0) xor
(Nutritional Factor2 >= 1)))){ (if(Nutritional Factor3 == 1){ 1 }else{
(if((((A_lumbricoides == 0) xor ((HAV != 0) and (Nutritional Factor1 < 2))) and
(Fly at home == 0)) and (HSV != 0)) xor (Sibling number == 2)) { 0 }else{ 1 } } }
}else{ (if(((Sibling number <= 2) xor ((Nutritional Factor1 >= 2) and (BMI <
1))) and (HSV == 1)){ (if((Daycare == 0) xor (age <= 1)) { 0 }else{ 1 } } }else{ 0
} } }
(if((((Change bed linen != 0) xor (((age == 1) or (Fly at home != 1)) and (HSV
!= 0)) or ((Nutritional Factor1 != 0) or ((Nutritional Factor3 == 0) xor (Gender
!= 1)))))) xor ((Nutritional Factor2 != 0) or (Gender != 0))) and ((Sibling
number == 2) xor (((HAV == 0) xor ((age != 2) and ((age >= 1) or (Fly at home ==
0)))) or (HSV == 0)))){ 1 }else{ (if((Sibling number <= 2) and (HSV != 0)){
(if(((Nutritional Factor2 <= 1) and ((pavimenta == 0) and ((age <= 2) and
(Daycare == 0)))) xor ((fuma != 0) xor ((Mold/moisture != 0) and ((Change bed
linen == 0) and (age > 1)))))) { 0 }else{ 1 } } }else{ (if(Daily calories <
2434,17){ 0 }else{ 0 } } } )
(if((((Change bed linen != 0) xor ((Sibling number == 2) or ((Nutritional
Factor1 > 0) or ((Nutritional Factor3 == 0) xor (Gender != 1)))))) xor
((Nutritional Factor2 != 0) or (Gender != 0))) and ((Sibling number == 2) xor
(((HAV == 0) xor ((age < 2) and ((age == 1) or (Fly at home != 1))) or (HSV ==
0)))){ (if(((age != 1) or (Gender == 1) or (pavimenta != 0)) { 1 }else{
(if(Nutritional Factor2 != 0){ 0 }else{ 1 } } } }else{ (if((Sibling number <= 2)
and (HSV != 0)){ (if(((Nutritional Factor2 <= 1) and ((pavimenta == 0) and
(Daycare == 0))) xor ((fuma != 0) xor ((Mold/moisture != 0) and ((Change bed
linen == 0) and (age > 1)))))) { 0 }else{ 1 } } } }
(if(((Change bed linen != 0) and ((age != 1) or (Gender == 1) or (pavimenta ==
1))) and (((Nutritional Factor3 != 0) and (Gender == 0) or (age >= 2)) or
((GNI >= 1) or ((Sibling number == 2) xor (HSV == 0)))))) { (if((Sibling number >=
2) xor (Fly at home == 0)) { (if(((cat != 1) xor (Nutritional Factor3 > 0)) and
((Sibling number <= 2) and (HSV != 0))) or (((Nutritional Factor2 > 0) and (age
!= 2)) and (BMI > 0)) xor (Sibling number == 2)) { 0 }else{ 1 } } }else{
(if(Nutritional Factor2 >= 1){ 1 }else{ 0 } } } }else{ (if((Sibling number <= 2)
and (HSV == 1)){ (if(((Nutritional Factor2 <= 1) and ((pavimenta == 0) and
(Daycare == 0))) xor ((Nutritional Factor3 <= 1) xor ((Mold/moisture != 0) and
((Nutritional Factor1 < 2) and (age > 1)))))) { 0 }else{ 1 } } }else{ 0 } } }
(if(((Change bed linen != 0) and ((age != 1) or (Gender == 1) or (pavimenta ==
1))) and (((Nutritional Factor3 > 0) and (Gender == 0) or (age > 1)) or ((GNI
== 1) or ((Sibling number == 2) xor (HSV == 0)))))) { (if((Sibling number <= 1)
xor (Fly at home != 0)) { (if((((HSV != 0) and ((dog == 1) xor (Nutritional
Factor2 != 0))) or (sewage disposal == 1)) and (Nutritional Factor3 != 1)) or
(((Nutritional Factor2 > 0) and (age < 2)) and (BMI > 0)) xor (Sibling number
== 2)) { 0 }else{ 1 } } }else{ (if(Nutritional Factor2 == 0){ 0 }else{ 1 } } }
}else{ (if((Sibling number < 3) and (HSV != 0)) { (if((Nutritional Factor2 <= 1)
and ((pavimenta == 0) and (Daycare == 0))) xor ((Daily calories <= 1692,98) xor
((Mold/moisture != 0) and ((Nutritional Factor1 < 2) and (age > 1)))))) { 0 }else{
1 } } }else{ 0 } } } )
(if(((Change bed linen == 0) xor ((age != 0) or ((GNI == 1) or ((Sibling number
!= 2) xor ((Nutritional Factor3 != 1) and (HSV == 1)))))) and ((water system !=
0) and (age != 1)) or ((HAV != 0) xor (((Sibling number == 2) xor (HSV == 0))
and (((cat != 1) xor (Gender != 1)) or (pavimenta != 0)))))) { (if((Sibling

```

```

number <= 1) xor (Fly at home != 0)){ (if(((Fly at home != 1) and ((HSV == 1)
and (Nutritional Factor3 == 0))) or (((Nutritional Factor2 > 0) and (age <= 1))
and (BMI != 0)) xor (Sibling number == 2))) { 0 }else{ 1 }} }else{
(if(Nutritional Factor2 == 0){ 0 }else{ 1 }} ) }else{ (if((Sibling number < 3)
and (HSV == 1)){ (if(((Nutritional Factor2 <= 1) and ((pavimenta == 0) and
(Daycare == 0))) xor ((age >= 0) xor ((Mold/moisture != 0) and ((Nutritional
Factor1 < 2) and (age > 1))))){ 0 }else{ 1 }} }else{ 0 }} )
(if(((Change bed linen == 0) xor ((age >= 2) or ((GNI == 1) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 > 0) or (Gender
!= 0))) xor ((Sibling number <= 2) or (A_lumbricoides == 1))) and (HSV ==
1)))))) and (((water system != 0) and (age != 1)) or ((HAV != 0) xor (((Sibling
number == 2) xor (HSV == 0)) and (((cat != 1) xor (Gender != 1)) or (pavimenta
!= 0))))))){ (if((Sibling number <= 1) xor (Fly at home != 0)){ (if((Daily
calories >= 2434,17) or (((Nutritional Factor2 != 0) and (age <= 1)) and (BMI >
0)) xor (Sibling number == 2))) { 0 }else{ 1 }} }else{ (if(Nutritional Factor2 <=
0){ 0 }else{ 1 }} ) }else{ (if((Sibling number != 3) and (HSV == 1)){
(if(((Nutritional Factor2 <= 1) and ((pavimenta == 0) and (Daycare == 0))) xor
((age >= 0) xor ((Mold/moisture != 0) and (age == 2)))) { 0 }else{ 1 }} }else{ 0
}} )
(if(((Change bed linen == 0) xor ((age >= 2) or ((GNI == 1) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 > 0) or (Gender
!= 0))) xor ((Sibling number <= 2) or (A_lumbricoides == 1))) and (HSV ==
1)))))) and (((water system != 0) and (age != 1)) or ((HAV != 0) xor (((Sibling
number == 2) xor (HSV == 0)) and (((cat == 1) xor (Gender != 0)) or (pavimenta
== 1))))))){ (if((Sibling number <= 1) xor (Fly at home != 0)){ (if((BMI >= 2) or
(((Nutritional Factor2 != 0) and (age <= 1)) and (BMI > 0)) xor (Sibling number
== 2))) { 0 }else{ 1 }} }else{ (if(Nutritional Factor2 == 0){ 0 }else{ 1 }} )
}else{ (if((Sibling number < 3) and (HSV != 0)){ (if(((Nutritional Factor2 <= 1)
and ((pavimenta == 0) and (Daycare == 0))) xor ((Daily calories <= 1692,98) xor
((Mold/moisture != 0) and ((Nutritional Factor1 < 2) and (age > 1))))){ 0 }else{
1 }} }else{ 0 }} )
(if(((Change bed linen == 0) xor ((age == 2) or ((GNI == 1) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or (Gender
!= 0))) xor (Sibling number <= 2)) and (HSV == 1)))))) and (((water system !=
0) and (age != 1)) and ((HSV == 1) or ((GNI == 1) xor ((Sibling number >= 2) or
(parents_asthma == 0)))))) or ((HAV != 0) xor ((Sibling number == 2) xor (HSV ==
0)) and (((cat != 1) and (Gender == 1)) or (pavimenta == 1))))){ (if((Sibling
number >= 2) xor (Fly at home == 0)){ (if(((Nutritional Factor2 > 0) and (BMI !=
0)) xor (Sibling number == 2)) { 0 }else{ 1 }} }else{ (if(Nutritional Factor2 ==
0){ 0 }else{ 1 }} ) }else{ (if((Sibling number != 3) and (HSV == 1)){
(if(((Nutritional Factor2 <= 1) and ((pavimenta == 0) and (Daycare == 0))) xor
((Daily calories <= 2434,17) xor ((Mold/moisture != 0) and ((Nutritional Factor1
!= 2) and (age > 1)))) { 0 }else{ 1 }} }else{ 0 }} )
(if(((Change bed linen == 0) xor ((age >= 2) or ((GNI == 1) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 > 0) or (Gender
!= 0))) xor ((Sibling number <= 2) or (A_lumbricoides == 1))) and (HSV ==
1)))))) and (((water system != 0) and (age != 1)) or ((HAV != 0) xor (((Sibling
number == 2) xor (HSV == 0)) and (((cat == 1) xor (Gender != 0)) or (pavimenta
== 1))))))){ (if((Sibling number <= 1) xor (Fly at home != 0)){ (if(((Fly at home
!= 1) and ((HSV == 1) and (Nutritional Factor3 == 0))) or (((Nutritional
Factor2 != 0) and (age <= 1)) and (BMI > 0)) xor (Sibling number == 2))) { 0
}else{ 1 }} }else{ (if(Nutritional Factor2 <= 0){ 0 }else{ 1 }} ) }else{
(if((Sibling number < 3) and (HSV != 0)){ (if(((Nutritional Factor2 <= 1) and
((pavimenta == 0) and (Daycare == 0))) xor ((GNI != 2) xor ((Mold/moisture != 0)
and ((Nutritional Factor1 < 2) and (age > 1)))) { 0 }else{ 1 }} }else{ 0 }} )
(if(((Change bed linen == 0) xor ((age == 2) or ((GNI == 1) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or (Gender
!= 0))) xor ((Sibling number <= 2) or (Gender == 1))) and (HSV == 1)))))) and
(((water system != 0) and (age != 1)) and (Nutritional Factor3 <= 2)) or ((HAV
!= 0) xor ((Sibling number == 2) xor (HSV == 0)) and (((cat != 1) and (Gender
== 1)) or (pavimenta == 1))))))){ (if((Sibling number <= 1) xor (Fly at home ==
1)){ (if(((Sibling number < 3) and ((HSV == 1) and (Nutritional Factor3 == 0)))
or (((Nutritional Factor2 > 0) and (age != 2)) and (BMI != 0)) xor (Sibling
number == 2))) { 0 }else{ 1 }} }else{ (if(Nutritional Factor2 == 0){ 0 }else{ 1
}} ) }else{ (if((Sibling number < 3) and (HSV != 0)){ (if(((Nutritional Factor2
<= 1) and ((pavimenta == 0) and (Daycare == 0))) xor ((Daily calories <=
1692,98) xor ((Mold/moisture != 0) and ((Nutritional Factor1 < 2) and (age >
1)))) { 0 }else{ 1 }} }else{ 0 }} )

```

```

(if(((Change bed linen == 0) xor ((age >= 2) or ((GNI > 0) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or (Gender
!= 0))) xor ((Sibling number != 3) or (Fly at home == 0))) and (HSV == 1))))))
and (((water system != 0) and (age != 1)) and ((HSV == 1) or (Nutritional
Factor4 > 0))) or ((HAV != 0) or ((Sibling number == 2) xor (HSV == 0)) and
((Gender != 0) or (pavimenta == 1)))))) { (if((Sibling number >= 2) xor (Fly at
home == 0)) { (if(((cat != 1) xor (Nutritional Factor3 > 0)) and ((Sibling
number <= 2) and (HSV != 0))) or (((HAV != 1) and (age < 2)) and (BMI != 0))
xor (Sibling number == 2))) { 0 } else { 1 } } } else { (if(Nutritional Factor2 >= 1) {
1 } else { 0 } } } } else { (if((Sibling number < 3) and (HSV == 1)) {
(if(((Nutritional Factor2 <= 1) and ((pavimenta == 0) and (Daycare == 0))) xor
((Nutritional Factor4 <= 1) xor ((Mold/moisture != 0) and ((Nutritional Factor1
!= 2) and (age > 1)))))) { 0 } else { 1 } } } else { 0 } } }
(if(((Change bed linen == 0) xor ((age == 2) or ((GNI == 1) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or (Gender
!= 0))) xor (Sibling number <= 2)) and (HSV == 1)))))) and (((water system !=
0) and (age != 1)) and ((HSV == 1) or ((GNI == 1) xor ((Sibling number >= 2) xor
(parents_asthma == 0)))))) or ((HAV != 0) xor ((Sibling number == 2) xor (HSV ==
0)) and (((cat != 1) and (Gender == 1)) or (pavimenta == 1)))))) { (if((Sibling
number <= 1) xor (Fly at home == 1)) { (if(((Sibling number < 3) and ((HSV == 1)
and (Nutritional Factor3 == 0))) or (((Nutritional Factor2 > 0) and (age != 2))
and (BMI != 0)) xor (Sibling number == 2))) { 0 } else { 1 } } } else {
(if(Nutritional Factor2 == 0) { 0 } else { 1 } } } } else { (if((Sibling number != 3)
and (HSV == 1)) { (if(((Nutritional Factor2 <= 1) and ((pavimenta == 0) and
(Daycare == 0))) xor ((age >= 0) xor ((Mold/moisture != 0) and ((Nutritional
Factor1 < 2) and (age > 1)))))) { 0 } else { 1 } } } else { 0 } } }
(if(((Change bed linen == 0) xor ((age == 2) or ((GNI == 1) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or (Gender
!= 0))) xor ((Sibling number <= 2) or (A_lumbricoides == 1))) and (HSV ==
1)))))) and (((water system != 0) and (age != 1)) and ((HSV == 1) or ((GNI ==
1) xor ((Sibling number >= 2) or (parents_asthma == 0)))))) or ((HAV != 0) xor
(((Sibling number == 2) xor (HSV == 0)) and (((cat != 1) and (Gender == 1)) or
(pavimenta == 1)))))) { (if((Sibling number <= 1) xor (Fly at home == 1)) {
(if(((Sibling number < 2) and ((HSV == 1) and (VZV == 1))) or (((Nutritional
Factor2 > 0) and (age < 2)) and (BMI != 0)) xor (Sibling number == 2))) { 0
} else { 1 } } } else { (if(Nutritional Factor2 <= 0) { 0 } else { 1 } } } } else {
(if((Sibling number != 3) and (HSV == 1)) { (if(((Nutritional Factor2 <= 1) and
(pavimenta == 0) and (Daycare == 0))) xor ((HSV == 1) xor ((Mold/moisture != 0)
and ((Nutritional Factor1 != 2) and (age > 1)))))) { 0 } else { 1 } } } else { 0 } } }
(if(((Change bed linen == 0) xor ((age == 2) or ((GNI == 1) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or (Gender
!= 0))) xor (Sibling number <= 2)) and (HSV == 1)))))) and (((water system !=
0) and (age != 1)) and ((HSV == 1) or ((GNI == 1) xor ((Sibling number >= 2) or
(parents_asthma == 0)))))) or ((HAV != 0) xor ((Sibling number == 2) xor (HSV ==
0)) and (((cat != 1) and (Gender == 1)) or (pavimenta == 1)))))) { (if((Sibling
number >= 2) xor (Fly at home == 0)) { (if((((EBV != 0) xor ((HAV == 0) xor
((age != 2) or (Mold/moisture == 1)))) xor (Nutritional Factor3 == 1)) and
((Sibling number <= 2) and (HSV == 1)) or ((Nutritional Factor2 > 0) and (BMI
!= 0)) xor (Sibling number == 2))) { 0 } else { 1 } } } else { (if(Nutritional Factor2
== 0) { 0 } else { 1 } } } } else { (if((Sibling number != 3) and (HSV == 1)) {
(if(((Nutritional Factor2 <= 1) and ((pavimenta == 0) and (Daycare == 0))) xor
((age >= 0) xor ((Mold/moisture != 0) and ((Nutritional Factor1 < 2) and (age >
1)))))) { 0 } else { 1 } } } else { 0 } } }
(if(((Change bed linen == 0) xor ((age == 2) or ((GNI == 1) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or (Gender
!= 0))) xor ((Sibling number <= 2) or (dog == 1))) and (HSV == 1)))))) and
(((water system != 0) and (age != 1)) and ((HSV == 1) or ((GNI == 1) xor
((Sibling number >= 2) or (parents_asthma == 0)))))) or ((HAV != 0) xor
(((Sibling number == 2) xor (HSV == 0)) and (((cat != 1) and (Gender == 1)) or
(pavimenta == 1)))))) { (if((Sibling number <= 1) xor (Fly at home == 1)) {
(if((((HSV == 1) and ((HAV != 0) or ((Sibling number >= 3) xor (EBV != 0))) or
(age < 0)) and (Nutritional Factor3 != 1)) or (((Nutritional Factor2 >= 1) and
(age <= 1)) and (BMI > 0)) xor (Sibling number == 2))) { 0 } else { 1 } } } else {
(if(Nutritional Factor2 == 0) { 0 } else { 1 } } } } else { (if((Sibling number < 3)
and (HSV == 1)) { (if(((Nutritional Factor2 <= 1) and ((pavimenta == 0) and
(Daycare == 0))) xor ((Nutritional Factor4 <= 1) xor ((Mold/moisture != 0) and
((Nutritional Factor1 != 2) and (age > 1)))))) { 0 } else { 1 } } } else { 0 } } }
(if(((Change bed linen == 0) xor ((age == 2) or ((GNI == 1) or ((Sibling number

```

```

!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 > 0) or (Gender
!= 0))) xor ((Sibling number < 3) or (A_lumbricoides == 1))) and (HSV == 1))))
and (((water system != 0) and (age != 1)) and ((HSV == 1) or ((GNI == 1) xor
((Sibling number >= 2) or (parents_asthma == 0)))) or ((HAV != 0) xor
(((Sibling number == 2) xor (HSV == 0)) and (((cat != 1) and (Gender == 1)) or
(pavimenta == 1)))))) { (if((Sibling number >= 2) xor (Fly at home == 0)) {
(if((((EBV != 0) and ((HAV != 0) xor ((age != 2) or (Mold/moisture == 1)))) xor
Nutritional Factor3 >= 1)) and ((Sibling number <= 2) and (HSV == 1))) or
((((Nutritional Factor2 != 0) and (age <= 1)) and (BMI == 1)) xor (Sibling
number == 2))) { 0 } else { 1 } } } else { (if(Nutritional Factor2 == 0) { 0 } else { 1
} } } } else { (if((Sibling number != 3) and (HSV == 1)) { (if(((Nutritional
Factor2 <= 1) and ((pavimenta == 0) and (Daycare == 0))) xor ((age >= 0) xor
((Mold/moisture != 0) and ((Nutritional Factor1 < 2) and (age > 1)))))) { 0 } else {
1 } } } else { 0 } } }
(if(((Change bed linen == 0) xor ((age == 2) or ((GNI == 1) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or (Gender
!= 0))) xor ((Sibling number <= 2) or (A_lumbricoides == 1))) and (HSV ==
1)))))) and (((water system == 1) and (age != 1)) and ((HSV == 1) or ((GNI ==
1) xor ((Sibling number >= 2) or (parents_asthma == 0)))) or ((HAV != 0) xor
(((Sibling number == 2) xor (HSV == 0)) and (((cat != 1) and (Gender == 1)) or
(pavimenta == 1)))))) { (if((Sibling number > 1) xor (Fly at home == 0)) {
(if((((EBV != 0) and ((HAV != 0) xor ((age != 2) or (Mold/moisture == 1)))) xor
Nutritional Factor3 >= 1)) and ((Sibling number <= 2) and (HSV == 1))) or
(((Nutritional Factor2 > 0) and (BMI > 0)) xor (Sibling number == 2))) { 0 } else {
1 } } } else { (if(Nutritional Factor2 == 0) { 0 } else { 1 } } } } else { (if((Sibling
number != 3) and (HSV == 1)) { (if((((GNI >= 1) or ((sewage disposal == 0) xor
(Nutritional Factor3 < 0))) and ((Nutritional Factor2 <= 1) and ((pavimenta ==
0) and (Daycare == 0)))) xor ((GNI <= 2) xor ((Mold/moisture != 0) and
((Nutritional Factor1 != 2) and (age > 1)))))) { 0 } else { 1 } } } else { 0 } } }
(if(((Change bed linen == 0) xor ((age == 2) or ((GNI == 1) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or (Gender
!= 0))) xor ((Sibling number <= 2) or (A_lumbricoides == 1))) and (HSV ==
1)))))) and (((water system != 0) and (age != 1)) and ((HSV == 1) or ((GNI ==
1) xor ((Sibling number >= 2) or (parents_asthma == 0)))) or ((HAV != 0) xor
(((Sibling number == 2) xor (HSV == 0)) and (((cat != 1) and (Gender == 1)) or
(pavimenta == 1)))))) { (if((Sibling number >= 2) xor (Fly at home == 0)) {
(if((((EBV != 0) and ((HAV != 0) xor ((age != 2) or (Mold/moisture == 1)))) xor
Nutritional Factor3 == 1)) and ((Sibling number <= 2) and (HSV == 1))) or
(((Nutritional Factor2 != 0) and ((age <= 1) xor (Sibling number <= 1)) and (HSV
== 1))) or (((Nutritional Factor2 > 0) and (BMI != 0)) xor (Sibling number ==
2))) { 0 } else { 1 } } } else { (if(Nutritional Factor2 <= 0) { 0 } else { 1 } } }
} else { (if((Sibling number != 3) and (HSV == 1)) { (if((((GNI >= 1) or ((sewage
disposal == 0) xor (Nutritional Factor3 < 0))) and ((Nutritional Factor2 <= 1)
and ((pavimenta == 0) and (Daycare == 0)))) xor ((GNI <= 1) xor ((Mold/moisture
!= 0) and ((Nutritional Factor1 != 2) and (age > 1)))))) { 0 } else { 1 } } } else { 0
} } }
(if(((Change bed linen == 0) xor ((age >= 2) or ((GNI >= 1) or ((Sibling number
!= 2) xor (((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or (Gender
!= 0))) xor ((Sibling number <= 2) or (A_lumbricoides == 1))) and (HSV ==
1)))))) and (((water system != 0) and (age != 1)) and ((HSV == 1) or ((GNI ==
1) xor ((Sibling number >= 2) or (parents_asthma == 0)))) or ((HAV != 0) xor
(((Sibling number == 2) xor (HSV == 0)) and (((cat != 1) xor (Gender != 1)) or
(pavimenta == 1)))))) { (if((Sibling number >= 2) xor (Fly at home == 0)) {
(if((((EBV != 1) xor ((HAV != 0) xor ((age != 2) or (Mold/moisture == 1)))) xor
Nutritional Factor3 == 1)) and ((Sibling number <= 2) and (HSV == 1))) or
(((Nutritional Factor2 != 0) and ((age <= 1) xor (Nutritional Factor1 < 0)))
and (BMI != 0)) xor (Sibling number == 2))) { 0 } else { 1 } } } else {
(if(Nutritional Factor2 <= 0) { 0 } else { 1 } } } } else { (if((Sibling number < 3)
and (HSV == 1)) { (if((((GNI == 1) or ((sewage disposal == 0) or (Change bed
linen != 0))) and ((Nutritional Factor2 <= 1) and ((pavimenta == 0) and (Daycare
== 0)))) xor ((GNI <= 2) xor ((Mold/moisture != 0) and ((Nutritional Factor1 !=
2) and (age > 1)))))) { 0 } else { 1 } } } else { 0 } } }
(if(((Change bed linen == 0) xor (((Nutritional Factor3 != 0) and (Sibling
number >= 2)) or (age == 2)) or ((GNI >= 1) or ((Sibling number != 2) xor
(((Change bed linen != 0) xor ((Nutritional Factor1 != 0) or (Gender != 0)))
xor ((Sibling number <= 2) or (A_lumbricoides == 1))) and (HSV != 0)))))) and
(((water system != 0) and (age != 1)) and ((HSV == 1) or ((GNI >= 1) xor
((Sibling number >= 2) or (parents_asthma == 0)))) or ((HAV != 0) xor
(((Sibling number != 2) xor (HSV != 0)) and (((cat != 1) xor (Gender != 1)) or

```

```
(pavimenta == 1))))))){ (if((Sibling number >= 2) xor (Fly at home == 0)){
(if((((EBV != 0) and ((HAV != 0) xor ((age != 2) or (Mold/moisture == 1)))) xor
(Nutritional Factor3 == 1)) and ((Sibling number <= 2) and (HSV == 1))) or
((((Nutritional Factor2 != 0) and ((age <= 2) xor ((age > 0) and ((EBV == 1) and
(age >= 2)))))) and (BMI != 0)) xor (Sibling number == 2))) { 0 }else{ 1 } } }else{
(if(Nutritional Factor2 != 0){ 1 }else{ 0 } } } }else{ (if((Sibling number != 3)
and (HSV == 1)){ (if((((GNI == 1) or ((sewage disposal == 0) xor (Nutritional
Factor3 < 0))) and ((Nutritional Factor2 <= 1) and ((pavimenta == 0) and
(Daycare == 0)))) xor ((GNI <= 1) xor ((Mold/moisture != 0) and ((Nutritional
Factor1 != 2) and (age > 1)))))) { 0 }else{ 1 } } }else{ 0 } } }
```

IgE

```
(if((Gender == 1) and (sewage disposal != 1)){ 1 }else{ 0 } }
(if(((age >= 1) xor (Tgondi != 0)) and (Gender == 1)){ 1 }else{ 0 } }
(if(((age < 1) and (Hpilory != 1)) xor (sewage disposal != 1)) and (Gender ==
1)){ 1 }else{ 0 } }
(if(((age <= 0) and (Hpilory != 1)) xor (sewage disposal != 1)) and ((HAV == 1)
or (Gender == 1)){ 1 }else{ 0 } }
(if(((age <= 0) and ((Hpilory != 1) and (dog == 0))) xor (sewage disposal !=
1)) and ((HAV != 0) or (Gender == 1)){ 1 }else{ 0 } }
(if((HAV == 0) and (((pavimenta == 1) or (Mother Psychological disorder == 0))
and (Gender != 0))){ 1 }else{ (if(Tgondi != 0){ 0 }else{ (if(HAV == 1){ 1 }else{
0 } } ) } }
(if(((pavimenta != 0) or (Mother Psychological disorder != 1)) and (Gender !=
0)){ (if(sewage disposal != 1){ 1 }else{ 0 } } }else{ (if(Tgondi != 0){ 0 }else{
(if(HAV != 0){ 1 }else{ 0 } ) } ) }
(if(((age == 0) and (Hpilory != 1)) xor (sewage disposal != 1)) and ((HAV == 1)
or (((pavimenta != 0) or (Mother Psychological disorder == 0)) and (Gender !=
0)))){ 1 }else{ 0 } }
(if(((age == 0) and ((Hpilory != 1) and (dog != 1))) xor (sewage disposal !=
1)) and ((HAV != 0) or (((pavimenta == 1) or (Mother Psychological disorder ==
0)) and (Gender == 1)))){ 1 }else{ 0 } }
(if((HAV != 0) or (((pavimenta == 1) or (Mother Psychological disorder != 1))
and (Gender != 0))){ (if((sewage disposal != 1) and ((HAV != 1) or ((Tgondi ==
0) and (Nutritional Factor1 != 0)))){ 1 }else{ 0 } } }else{ 0 } }
(if(((pavimenta == 1) or (Mother Psychological disorder != 1)) and (Gender ==
1)){ (if(((age == 0) and (Hpilory != 1)) xor (sewage disposal != 1)){ 1 }else{ 0
} } }else{ (if(Tgondi != 0){ 0 }else{ (if(HAV != 0){ 1 }else{ 0 } ) } ) }
(if(((pavimenta == 1) or (Mother Psychological disorder != 1)) and (Gender ==
1)){ (if(((age == 0) and (Hpilory != 1)) xor (sewage disposal != 1)){ 1 }else{ 0
} } }else{ (if(Tgondi != 0){ 0 }else{ (if(HAV != 0){ 1 }else{ 0 } ) } ) }
(if(((pavimenta != 0) or (Mother Psychological disorder == 0)) and (Gender !=
0)){ (if((age < 1) xor (sewage disposal != 1)){ 1 }else{ (if(Hpilory != 0){ 1
}else{ 0 } ) } } }else{ (if(Tgondi != 0){ 0 }else{ (if(HAV != 0){ 1 }else{ 0 } )
} ) }
(if(((pavimenta == 1) or (Mother Psychological disorder != 1)) and (Gender ==
1)){ (if(((age < 1) and ((Hpilory != 1) and (Change bed linen != 0))) xor
(sewage disposal != 1)){ 1 }else{ 0 } } }else{ (if(Tgondi != 0){ 0 }else{ (if(HAV
!= 0){ 1 }else{ 0 } ) } ) }
(if((HAV != 0) or (((pavimenta == 1) or (Mother Psychological disorder == 0))
and (Gender == 1))){ (if(((age <= 0) and (Hpilory != 1)) xor (sewage disposal
!= 1)) and ((HAV != 1) or ((Tgondi == 0) and (Nutritional Factor1 >= 1)))){ 1
}else{ 0 } } }else{ 0 } }
(if(((pavimenta != 0) or (Mother Psychological disorder != 1)) and (Gender ==
1)){ (if(((age <= 0) and ((dog != 1) or (HAV != 0))) and (Hpilory != 1)) xor
(sewage disposal != 1)){ 1 }else{ 0 } } }else{ (if(Tgondi != 0){ 0 }else{ (if(HAV
!= 0){ 1 }else{ 0 } ) } ) }
(if((HAV != 0) or (((pavimenta == 1) or (Mother Psychological disorder == 0))
and (Gender != 0))){ (if(((age == 0) and ((Hpilory != 1) and (dog != 1))) xor
(sewage disposal != 1)) and ((HAV != 1) or ((Tgondi == 0) and (Nutritional
Factor1 > 0)))){ 1 }else{ 0 } } }else{ 0 } }
(if((HAV != 0) or (((pavimenta == 1) or (Mother Psychological disorder == 0))
and (Gender != 0))){ (if(((age < 1) and ((Hpilory != 1) and (dog != 1)) and
(cat != 1))) xor (sewage disposal != 1)) and ((HAV != 1) or ((Tgondi == 0) and
(Nutritional Factor1 != 0)))){ 1 }else{ 0 } } }
(if((HAV != 0) or (((pavimenta == 1) or (Mother Psychological disorder == 0))
and (Gender != 0))){ (if(((age < 1) and ((Hpilory != 1) and (dog != 1)) and
```

```

((Nutritional Factor4 >= 1) or (GNI == 1))) xor (sewage disposal != 1) and
((HAV != 1) or ((Tgondi == 0) and (Nutritional Factor1 != 0))) { 1 } else { 0 } }
} else { 0 } }
(if(((pavimenta == 1) or (Mother Psychological disorder != 1)) and (((EBV == 0)
or ((Nutritional Factor2 <= 1) or ((Daycare != 0) xor (Gender == 1)))) xor
(Gender == 0))) { (if(((age == 0) and ((Hpilory != 1) and (dog == 0))) xor
(sewage disposal != 1)) { 1 } else { 0 } } } else { (if(Tgondi != 0) { 0 } else { (if(HAV
!= 0) { 1 } else { 0 } } ) } } }
(if(((pavimenta != 0) or (Mother Psychological disorder == 0)) and (((Sibling
number >= 3) or ((EBV != 1) or ((Nutritional Factor2 != 2) or ((Daycare != 0)
xor (Gender == 1)))) xor (Gender != 1))) { (if(((age == 0) and ((Hpilory != 1)
and (dog != 1))) xor (sewage disposal != 1)) { 1 } else { 0 } } } else { (if(Tgondi !=
0) { 0 } else { (if(HAV != 0) { 1 } else { 0 } } ) } } }
(if(((pavimenta == 1) or (Mother Psychological disorder == 0)) and (((Sibling
number == 3) or ((EBV != 1) or ((Nutritional Factor2 != 2) or ((Daycare != 0)
xor (Gender != 0)))) xor (Gender == 0))) { (if(((age < 1) and ((dog != 1) or
(HAV == 1))) and (Hpilory != 1)) xor (sewage disposal != 1)) { 1 } else { 0 } }
} else { (if(Tgondi == 1) { 0 } else { (if(HAV != 0) { 1 } else { 0 } ) } } } }
(if(((pavimenta == 1) or (Mother Psychological disorder == 0)) and (((Sibling
number > 2) or ((EBV != 1) or ((Nutritional Factor2 < 2) or ((Daycare != 0) xor
(Gender == 1)))) xor (Gender != 1))) { (if(((age < 1) and ((Hpilory != 1) and
(dog != 1))) xor (sewage disposal != 1)) and ((HAV != 1) or (Nutritional Factor1
!= 0))) { 1 } else { 0 } } } else { (if(Tgondi != 0) { 0 } else { (if(HAV == 1) { 1 } else {
0 } ) } } } }
(if(((pavimenta != 0) or (Mother Psychological disorder == 0)) and (((Sibling
number > 2) or ((EBV != 1) or ((Nutritional Factor2 != 2) or ((Daycare != 0) xor
(Gender == 1)))) xor (Gender != 1))) { (if(((age <= 0) and ((Hpilory != 1) and
(dog != 1))) xor (sewage disposal != 1)) and ((HAV != 1) or ((Tgondi == 0) and
(Nutritional Factor1 != 0))) { 1 } else { 0 } } } else { (if(Tgondi != 0) { 0 } else {
(if(HAV != 0) { 1 } else { 0 } ) } } } }
(if(((pavimenta != 0) or (Mother Psychological disorder == 0)) and (((Sibling
number >= 3) or ((EBV != 1) or ((Nutritional Factor2 != 2) or ((Daycare != 0)
xor (Gender == 1)))) xor (Gender != 1))) { (if(((age < 1) and ((Hpilory != 1)
and (dog != 1)) and (cat != 1))) xor (sewage disposal != 1)) and ((HAV != 1) or
((Tgondi == 0) and (Nutritional Factor1 != 0))) { 1 } else { 0 } } } else {
(if(Tgondi != 0) { 0 } else { (if(HAV != 0) { 1 } else { 0 } ) } } } }
(if(((pavimenta != 0) or (Mother Psychological disorder == 0)) and (((Sibling
number >= 3) or ((EBV != 1) or ((Nutritional Factor2 < 2) or ((Daycare != 0) xor
((Gender != 0) and ((A_lumbricoides == 0) or ((Nutritional Factor4 < 1) or
(Mold/moisture != 1)))))) xor (Gender != 1))) { (if(((age == 0) and ((Hpilory
!= 1) and (dog != 1))) xor (sewage disposal != 1)) and ((HAV != 1) or ((Tgondi
== 0) and (Nutritional Factor1 > 0))) { 1 } else { 0 } } } else { (if(Tgondi != 0) { 0
} else { (if(HAV != 0) { 1 } else { 0 } ) } } } }
(if(((Tgondi == 1) xor ((Nutritional Factor3 > 0) and (HSV != 1))) xor (Gender
!= 0)) { (if(((sewage disposal == 1) xor ((HAV == 0) or (Nutritional Factor1 >
0))) and (((Mother Psychological disorder == 0) or (HAV != 0)) or (Nutritional
Factor2 <= 0)) and (Gender == 1))) { (if((age <= 0) and ((Hpilory != 1) and (dog
== 0))) { 0 } else { 1 } } } else { 0 } } } else { (if((age >= 1) xor (Nutritional
Factor1 < 2)) { (if(Fly at home != 1) { 0 } else { 1 } } } else { (if(((Sibling number
> 0) and ((Sibling number == 1) xor (Hpilory != 0))) and (GNI > 0)) { 1 } else { 0
} ) } } } }
(if(((Tgondi == 1) xor ((Nutritional Factor3 != 0) and (HSV == 0))) xor (Gender
!= 0)) { (if(((sewage disposal == 1) xor ((HAV == 0) or (Nutritional Factor1 >
0))) and (((Mother Psychological disorder != 1) or (HAV != 0)) or (Nutritional
Factor2 == 0)) and (Gender != 0))) { (if((age <= 0) and ((Hpilory != 1) and
(parents_asthma != 1))) and (Sibling number != 3)) { 0 } else { 1 } } } else { 0 } }
} else { (if((age >= 1) xor (Nutritional Factor1 < 2)) { (if(Fly at home != 1) { 0
} else { 1 } } } else { (if(((Sibling number != 0) and ((Sibling number == 1) xor
(Hpilory != 0))) and (GNI == 1)) { 1 } else { 0 } ) } } } }
(if(((Tgondi != 1) xor ((Nutritional Factor3 != 0) and (HSV == 0))) xor (Gender
!= 1)) { (if((sewage disposal != 1) and (((Mother Psychological disorder == 0)
or (water system != 1)) or (Nutritional Factor2 == 0)) and (Gender == 1))) {
(if((age < 1) and (Hpilory != 1)) and (dog != 1)) { 0 } else { 1 } } } else { 0 } }
} else { (if((age >= 1) xor (Nutritional Factor1 != 2)) { (if(Fly at home != 1) { 0
} else { 1 } } } else { (if(((sewage disposal != 1) and (((Nutritional Factor1 != 0)
and ((Tgondi != 0) or (Nutritional Factor2 != 1))) and (Sibling number < 3)) or
(HAV != 0))) and ((Fly at home != 1) or (GNI == 1))) { 1 } else { 0 } ) } } } }
(if(((Tgondi != 1) xor ((Nutritional Factor3 > 0) and (HSV == 0))) xor (Gender

```

```

!= 1)) { (if((sewage disposal != 1) and (((Mother Psychological disorder == 0)
or (water system != 1)) or (Nutritional Factor2 == 0)) and (Gender == 1))) {
  (if(((age == 0) and ((Hpilory != 1) and (parents_asthma != 1))) and (Sibling
number <= 2)) { 0 } else { 1 }) } else { 0 }) } else { (if((age >= 1) xor (Nutritional
Factor1 != 2)) { (if(Fly at home != 1) { 0 } else { 1 }) } else { (if(((sewage
disposal != 1) and (((Nutritional Factor1 != 0) and ((Tgondi == 1) or
(Nutritional Factor2 != 1))) and (Sibling number < 3)) or (HAV != 0))) and ((Fly
at home != 1) or (GNI == 1))) { 1 } else { 0 }) } })
  (if(((Tgondi != 1) xor ((Nutritional Factor3 > 0) and (HSV != 1))) xor (Gender
!= 1)) { (if(((sewage disposal == 1) xor ((HAV == 0) or (Nutritional Factor1 !=
0))) and (((Mother Psychological disorder == 0) or (HAV != 0)) or (Nutritional
Factor2 < 1)) and (Gender != 0))) { (if((age == 0) and ((Hpilory != 1) and (dog
!= 1))) { 0 } else { 1 }) } else { 0 }) } else { (if((age >= 1) xor (Nutritional
Factor1 != 2)) { (if(Fly at home != 1) { 0 } else { 1 }) } else { (if(((sewage
disposal != 1) and (((Nutritional Factor1 >= 1) and ((Gender != 0) or
(Nutritional Factor2 != 1))) and (Sibling number < 3)) or (HAV != 0))) and ((Fly
at home != 1) or (GNI >= 1))) { 1 } else { 0 }) } })
  (if(((Tgondi != 1) xor ((Nutritional Factor3 > 0) and (HSV != 1))) xor (Gender
!= 1)) { (if(((sewage disposal == 1) xor ((HAV == 0) or (Nutritional Factor1 >
0))) and (((Mother Psychological disorder != 1) or (HAV != 0)) or (Nutritional
Factor2 <= 0)) and (Gender == 1))) { (if((age <= 0) and ((Hpilory != 1) and
(parents_asthma != 1))) and (Sibling number < 3)) { 0 } else { 1 }) } else { 0 })
  } else { (if((age >= 1) xor (Nutritional Factor1 < 2)) { (if(Fly at home != 1) { 0
} else { 1 }) } else { (if(((sewage disposal != 1) and (((Nutritional Factor1 >= 1)
and ((Gender != 0) or (Nutritional Factor2 != 1))) and (Sibling number != 3)) or
(HAV != 0))) and ((Fly at home != 1) or (GNI == 1))) { 1 } else { 0 }) } })
  (if(((Tgondi != 1) xor ((Nutritional Factor3 > 0) and (HSV != 1))) xor (Gender
== 0)) { (if(((sewage disposal == 1) xor ((HAV == 0) or (Nutritional Factor1 !=
0))) and (((Mother Psychological disorder == 0) or (HAV != 0)) or (Nutritional
Factor2 == 0)) and (Gender == 1))) { (if((age == 0) and ((Hpilory != 1) and (dog
!= 1))) { 0 } else { 1 }) } else { 0 }) } else { (if((age >= 1) xor (Nutritional
Factor1 < 2)) { (if(Fly at home != 1) { 0 } else { 1 }) } else { (if((((Nutritional
Factor2 <= 1) and (Hpilory == 1)) or (sewage disposal != 1)) and (((Nutritional
Factor1 >= 1) and ((Gender == 1) or (Nutritional Factor2 != 1))) and (Sibling
number < 3)) or (HAV != 0))) and ((Fly at home != 1) or (GNI == 1))) { 1 } else { 0
} }) } })
  (if(((Tgondi == 1) xor ((Nutritional Factor3 > 0) and (HSV == 0))) xor (Gender
!= 0)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV == 1)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 == 0)) and
((HAV != 1) or (Nutritional Factor1 > 0)))) and (((HAV == 1) xor ((pavimenta
!= 0) xor (EBV != 0))) or (Mother Psychological disorder == 0)) or ((Tgondi ==
1) xor (VZV == 0))) xor (Gender != 1))) { (if((age <= 0) and ((Hpilory != 1) and
(dog != 1))) { 0 } else { 1 }) } else { 0 }) } else { (if((age >= 1) xor (Nutritional
Factor1 != 2)) { (if(Fly at home != 1) { 0 } else { 1 }) } else { (if(((Sibling number
!= 0) and ((Sibling number > 1) xor (Hpilory == 0))) and (GNI == 1)) { 1 } else { 0
} }) } })
  (if(((Tgondi == 1) xor ((Nutritional Factor3 != 0) and (HSV == 0))) xor (Gender
!= 0)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV == 1)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 == 0)) and
((HAV != 1) or (Nutritional Factor1 > 0)))) and (((HAV == 1) xor ((pavimenta
!= 0) xor (EBV == 0))) or (Mother Psychological disorder == 0)) or ((Tgondi !=
0) xor (VZV == 0))) xor (Gender != 1))) { (if((age == 0) and ((Hpilory != 1)
and (dog != 1))) and (Sibling number < 3)) { 0 } else { 1 }) } else { 0 }) } else {
  (if((age >= 1) xor (Nutritional Factor1 < 2)) { (if(Fly at home != 1) { 0 } else { 1
} }) } else { (if(((Sibling number > 0) and ((Sibling number == 1) xor (Hpilory !=
0))) and (GNI == 1)) { 1 } else { 0 }) } }) }
  (if(((Tgondi != 1) xor ((Nutritional Factor3 >= 1) and (HSV == 0))) xor (Gender
== 0)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV != 0)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 <= 0)) and

```

```

((HAV != 1) or (Nutritional Factor1 != 0))) and (((Mother Psychological
disorder != 1) or (HAV == 1)) or ((pavimenta != 0) or (VZV == 0)) xor (Gender
== 0))) { (if((age == 0) and ((Hpilory != 1) and (dog != 1))) { 0 } else { 1 })
} else { 0 } } } else { (if((age >= 1) xor (Nutritional Factor1 != 2)) { (if(Fly at
home != 1) { 0 } else { 1 }) } else { (if(((sewage disposal != 1) and (((Nutritional
Factor1 >= 1) and ((Gender != 0) or (Nutritional Factor2 != 1))) and (Sibling
number <= 2)) or (HAV != 0))) and ((Fly at home != 1) or (GNI >= 1))) { 1 } else {
0 } } } } }
(if(((Tgondi == 1) xor ((Nutritional Factor3 != 0) and (HSV != 1))) xor (Gender
== 1)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV == 1)
or (Mother Psychological disorder == 0)) or (Nutritional Factor2 <= 0)) and
((HAV != 1) or (Nutritional Factor1 != 0)))) and (((HAV == 1) xor (pavimenta
!= 0)) or (Mother Psychological disorder == 0)) or ((Tgondi != 1) xor (VZV ==
1))) xor (Gender != 1))) { (if((age < 1) and (Hpilory != 1) and (dog != 1)) { 0
} else { 1 }) } else { 0 } } } else { (if((age >= 1) xor (Nutritional Factor1 != 2)) {
(if(Fly at home != 1) { 0 } else { 1 }) } else { (if(((sewage disposal != 1) and
(((Nutritional Factor1 >= 1) and ((Gender == 1) or (Nutritional Factor2 != 1)))
and (Sibling number <= 2)) or (HAV != 0))) and ((Fly at home != 1) or (GNI >=
1))) { 1 } else { 0 } } } } }
(if(((Tgondi != 1) xor ((Nutritional Factor3 != 0) and (HSV == 0))) xor (Gender
== 0)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV == 1)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 <= 0)) and
((HAV != 1) or (Nutritional Factor1 > 0)))) and (((HAV == 1) xor (pavimenta !=
0)) or (Mother Psychological disorder == 0)) or ((Tgondi != 0) xor (VZV == 0)))
xor (Gender != 1))) { (if((age < 1) and (Hpilory != 1) and (dog != 1)) { 0
} else { 1 }) } else { 0 } } } else { (if((age >= 1) xor (Nutritional Factor1 < 2)) {
(if(Fly at home != 1) { 0 } else { 1 }) } else { (if((((Nutritional Factor2 <= 1)
and (Hpilory == 1)) or (sewage disposal != 1)) and (((Nutritional Factor1 >= 1)
and ((Gender == 1) or (Nutritional Factor2 != 1))) and (Sibling number != 3)) or
(HAV != 0))) and ((Fly at home != 1) or (GNI >= 1))) { 1 } else { 0 } } } } }
(if(((Tgondi != 1) xor ((Nutritional Factor3 > 0) and (HSV == 0))) xor (Gender
== 0)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV == 1)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 == 0)) and
((HAV != 1) or (Nutritional Factor1 != 0)))) and (((HAV == 1) xor (pavimenta
!= 0) or (EBV == 0))) or (Mother Psychological disorder == 0)) or ((Tgondi != 0)
xor (VZV == 0))) xor (Gender != 1))) { (if((age == 0) and ((Hpilory != 1) and
(dog != 1))) { 0 } else { 1 }) } else { 0 } } } else { (if((age >= 1) xor (Nutritional
Factor2 <= 1) and (Hpilory == 1)) or (sewage disposal != 1)) and (((Nutritional
Factor1 >= 1) and ((Gender != 0) or (Nutritional Factor2 != 1))) and (Sibling
number != 3)) or (HAV != 0))) and ((Fly at home != 1) or (GNI == 1))) { 1 } else {
0 } } } } }
(if(((Tgondi != 1) xor ((Nutritional Factor3 > 0) and (HSV == 0))) xor (Gender
== 0)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV == 1)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 == 0)) and
((HAV != 1) or (Nutritional Factor1 > 0)))) and (((HAV == 1) xor (pavimenta
== 1) xor (EBV == 0))) or (Mother Psychological disorder == 0)) or ((Tgondi !=
0) xor (VZV == 0))) xor (Gender != 1))) { (if((age <= 0) and ((Hpilory != 1)
and (dog != 1))) and (Sibling number <= 2)) { 0 } else { 1 }) } else { 0 } } } else {
(if((age >= 1) xor (Nutritional Factor1 < 2)) { (if(Fly at home != 1) { 0 } else { 1
}) } else { (if((((Nutritional Factor2 <= 1) and (Hpilory == 1)) or (sewage
disposal != 1)) and (((Nutritional Factor1 >= 1) and ((Gender != 0) or
(Nutritional Factor2 != 1))) and (Sibling number < 3)) or (HAV != 0))) and ((Fly
at home != 1) or (GNI == 1))) { 1 } else { 0 } } } } }
(if(((Tgondi != 1) xor ((Nutritional Factor3 != 0) and (HSV == 0))) xor (Gender
== 0)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV != 0)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 == 0)) and

```

```

((HAV != 1) or (Nutritional Factor1 > 0))) and (((((HAV == 1) xor ((pavimenta
!= 0) xor (EBV == 0))) or (Mother Psychological disorder == 0) or ((Tgondi !=
0) xor (VZV == 0))) xor (Gender != 1)))){ (if((age < 1) and ((Hpilory != 1) and
(dog != 1))) { 0 } else { 1 } } } else { 0 } } } else { (if((age >= 1) xor (Nutritional
Factor1 != 2)) { (if(Fly at home != 1) { 0 } else { 1 } } } else { (if((((Nutritional
Factor2 <= 1) and (Hpilory == 1)) or (sewage disposal != 1)) and
((((((parents_asthma == 0) and (Hpilory == 0)) and (Gender == 0)) or
(Nutritional Factor1 > 0)) and ((Gender != 0) or (Nutritional Factor2 != 1)))
and (Sibling number != 3)) or (HAV != 0))) and ((Fly at home != 1) or (GNI ==
1))) { 1 } else { 0 } } } }
(if(((Tgondi != 1) xor ((Nutritional Factor3 != 0) and (HSV == 0))) xor (Gender
== 0)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV != 0)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 == 0)) and
((HAV != 1) or (Nutritional Factor1 > 0)))) and (((((HAV == 1) xor ((pavimenta
!= 0) xor (EBV == 0))) or (Mother Psychological disorder == 0) or ((Tgondi !=
0) xor (VZV == 0))) xor (Gender != 1)))) { (if((age < 1) and ((Hpilory != 1) and
(dog != 1))) { 0 } else { 1 } } } else { 0 } } } else { (if((age >= 1) xor (Nutritional
Factor1 != 2)) { (if(Fly at home != 1) { 0 } else { 1 } } } else { (if((((Nutritional
Factor2 <= 1) and (Hpilory == 1)) or (sewage disposal != 1)) and
((((((parents_asthma == 0) and (Hpilory == 0)) and (Gender == 0)) or
(Nutritional Factor1 > 0)) and ((Gender != 0) or (Nutritional Factor2 != 1)))
and (Sibling number != 3)) or (HAV != 0))) and ((Fly at home != 1) or (GNI ==
1))) { 1 } else { 0 } } } }
(if(((Tgondi != 1) xor ((Nutritional Factor3 != 0) and (HSV == 0))) xor (Gender
== 0)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV != 0)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 == 0)) and
((HAV != 1) or (Nutritional Factor1 != 0)))) and (((((HAV == 1) xor ((pavimenta
!= 0) or (EBV == 0))) or (Mother Psychological disorder == 0) or ((Tgondi != 0)
xor (VZV == 0))) xor (Gender != 1)))) { (if((age <= 0) and ((Hpilory != 1) and
(dog != 1))) and (Sibling number <= 2)) { 0 } else { 1 } } } else { 0 } } } else {
(if((age >= 1) xor (Nutritional Factor1 < 2)) { (if(Fly at home != 1) { 0 } else { 1
} } } else { (if((((Nutritional Factor2 <= 1) and (Hpilory == 1)) or (sewage
disposal != 1)) and (((((((parents_asthma == 0) and (A_lumbricoides0 == 0)) and
(Tgondi == 0)) or (Nutritional Factor1 != 0)) and ((Gender != 0) or (Nutritional
Factor2 != 1))) and (Sibling number < 3)) or (HAV != 0))) and ((Fly at home !=
1) or (GNI >= 1))) { 1 } else { 0 } } } }
(if(((Tgondi != 1) xor ((Nutritional Factor3 > 0) and (HSV == 0))) xor (Gender
== 0)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV != 0)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 <= 0)) and
((HAV != 1) or (Nutritional Factor1 > 0)))) and (((((HAV == 1) xor ((pavimenta
!= 0) xor (EBV == 0))) or (Mother Psychological disorder == 0) or ((Tgondi ==
0) xor (VZV == 1))) xor (Gender != 1)))) { (if((age == 0) and ((Hpilory != 1)
and (dog != 1))) and (Sibling number != 3)) { 0 } else { 1 } } } else {
(if(Nutritional Factor3 > 0) { 0 } else { 0 } } } } else { (if((age >= 1) xor
(Nutritional Factor1 < 2)) { (if(Fly at home != 1) { 0 } else { 1 } } } else {
(if((((Nutritional Factor2 <= 1) and (Hpilory == 1)) or (sewage disposal != 1))
and (((((((parents_asthma == 0) and (Tgondi == 0)) and (Hpilory == 0)) or
(Nutritional Factor1 > 0)) and ((Gender != 0) or (Nutritional Factor2 != 1)))
and (Sibling number < 3)) or (HAV != 0))) and ((Fly at home != 1) or (GNI >=
1))) { 1 } else { 0 } } } }
(if(((Tgondi == 0) xor ((Nutritional Factor3 >= 1) and (HSV == 0))) xor (Gender
!= 1)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV != 0)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 == 0)) and
((HAV != 1) or (Nutritional Factor1 != 0)))) and (((((Mother Psychological
disorder != 1) or (HAV == 1)) or ((pavimenta != 0) or (VZV == 0))) xor (Gender
== 0)))) { (if((age <= 0) and ((Hpilory != 1) and (dog != 1))) { 0 } else { 1 } }
} else { 0 } } } else { (if((age > 0) xor (Nutritional Factor1 < 2)) { (if(Fly at home
!= 1) { 0 } else { 1 } } } else { (if((((Nutritional Factor2 <= 1) or (sewage disposal
!= 1)) and (((Sibling number != 3) or ((Nutritional Factor1 == 0) xor (age ==
1))) and (((((((parents_asthma == 0) and (Nutritional Factor2 <= 1)) and (Mother
Psychological disorder != 1)) xor (Nutritional Factor1 == 0)) xor (Sibling
number >= 1)) and ((age == 2) or ((Nutritional Factor1 == 1) and ((dog != 0)
xor (Mother Psychological disorder == 0))) xor (GNI >= 1)))) or (HAV == 1)))
and ((HAV != 1) or (GNI >= 1))) { 1 } else { 0 } } } }
(if(((Tgondi == 1) xor ((Nutritional Factor3 != 0) and (HSV != 1))) xor (Gender
== 1)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV == 1)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 <= 0)) and
((HAV != 1) or (Nutritional Factor1 > 0)))) and (((((HAV == 1) xor (pavimenta !=

```

```

0)) or (Mother Psychological disorder == 0)) or ((Tgondi != 0) xor (VZV == 0)))
xor (Gender != 1))) { (if(((age < 1) and (Hpilory != 1)) and (dog != 1)) { 0
}else{ 1 }} }else{ 0 }} }else{ (if((age > 0) xor (Nutritional Factor1 < 2)) {
(if(Fly at home != 1) { 0 }else{ 1 }} }else{ (if(((Nutritional Factor2 <= 0) or
(Sibling number != 0)) and (((Sibling number < 3) or ((Nutritional Factor1 <= 0)
or (age == 1))) and (((((((parents_asthma == 0) and (Nutritional Factor2 <= 1))
and (Mother Psychological disorder != 1)) xor (Nutritional Factor1 == 0)) xor
(Sibling number >= 1)) and ((age == 2) or (((Nutritional Factor1 == 1) and ((dog
!= 0) or (Mother Psychological disorder == 0))) xor (GNI == 1)))) or (HAV ==
1)))) and ((HAV != 1) or (GNI >= 1))) { 1 }else{ 0 }} } }
(if(((Tgondi == 1) xor ((Nutritional Factor3 != 0) and (HSV != 1))) xor (Gender
== 1)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV == 1)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 <= 0)) and
((HAV != 1) or (Nutritional Factor1 > 0)))) and (((HAV == 1) xor (pavimenta !=
0)) or (Mother Psychological disorder == 0)) or ((Tgondi != 0) xor (VZV == 0)))
xor (Gender != 1))) { (if(((age < 1) and (Hpilory != 1)) and (dog != 1)) { 0
}else{ 1 }} }else{ 0 }} }else{ (if((age > 0) xor (Nutritional Factor1 < 2)) {
(if(Fly at home != 1) { 0 }else{ 1 }} }else{ (if(((Nutritional Factor2 <= 0) or
(Sibling number != 0)) and (((Sibling number < 3) or ((Nutritional Factor1 <= 0)
or (age == 1))) and (((((((parents_asthma == 0) and (Nutritional Factor2 <= 1))
and (Mother Psychological disorder != 1)) xor (Nutritional Factor1 == 0)) xor
(Sibling number >= 1)) and ((age == 2) or (((Nutritional Factor1 == 1) and ((dog
!= 0) or (Mother Psychological disorder == 0))) xor (GNI == 1)))) or (HAV ==
1)))) and ((HAV != 1) or (GNI >= 1))) { 1 }else{ 0 }} } }
(if(((Tgondi == 0) xor ((Nutritional Factor3 > 0) and ((Sibling number > 0) and
(HSV == 0)))) xor (Gender == 0)) { (if((sewage disposal != 1) and (((Sibling
number == 2) or (((HAV == 1) or (Mother Psychological disorder != 1)) or
(Nutritional Factor2 <= 0)) and ((HAV != 1) or (Nutritional Factor1 != 0)))) and
((((HAV == 1) xor ((pavimenta != 0) or (EBV == 0))) or (Mother Psychological
disorder == 0)) or ((Tgondi != 0) xor (VZV == 0))) xor (Gender != 1))) {
(if((age == 0) and ((Hpilory != 1) and (dog != 1))) { 0 }else{ 1 }} }else{ 0 }}
}else{ (if((age != 0) xor (Nutritional Factor1 != 2)) { (if(Fly at home != 1) { 0
}else{ 1 }} }else{ (if(((Nutritional Factor2 <= 1) or (sewage disposal != 1))
and (((Sibling number != 3) or ((Nutritional Factor1 <= 0) or (age == 1))) and
((((((((parents_asthma == 0) and (Nutritional Factor2 <= 1)) and (Mother
Psychological disorder != 1)) xor (Nutritional Factor1 <= 0)) xor (Sibling
number >= 1)) and ((age >= 2) or ((Nutritional Factor1 == 1) and ((dog != 0)
xor (Mother Psychological disorder == 0))) xor (GNI >= 1)))) or (HAV == 1))))
and ((HAV != 1) or (GNI == 1))) { 1 }else{ 0 }} } }
(if(((Tgondi == 0) xor ((Nutritional Factor3 != 0) and (HSV == 0))) xor (Gender
== 0)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV == 1)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 == 0)) and
((HAV != 1) or (Nutritional Factor1 > 0)))) and (((HAV == 1) xor ((pavimenta
!= 0) or (EBV == 0))) or (Mother Psychological disorder == 0)) or ((Tgondi !=
0) xor (VZV == 0))) xor (Gender != 1))) { (if((age <= 0) and ((Hpilory != 1)
and (dog != 1))) and (Sibling number < 3)) { 0 }else{ 1 }} }else{ 0 }} }else{
(if((age > 0) xor (Nutritional Factor1 != 2)) { (if(Fly at home != 1) { 0 }else{ 1
}} }else{ (if(((Nutritional Factor2 <= 1) or (sewage disposal != 1)) and
(((Sibling number != 3) or ((Nutritional Factor1 != 0) xor (age != 1))) and
((((((((parents_asthma == 0) and (Tgondi != 1)) and (Daycare != 0)) xor
(Nutritional Factor1 != 0)) and ((HSV == 1) or (((Nutritional Factor3 <= 1) and
(Gender != 0)) or (Nutritional Factor2 > 1)))) and ((age == 2) or (((Nutritional
Factor1 == 1) and ((dog != 0) or (Mother Psychological disorder == 0))) xor (GNI
>= 1)))) or (HAV != 0)))) and ((HAV != 1) or (GNI >= 1))) { 1 }else{ 0 }} } }
(if(((Tgondi == 0) xor ((Nutritional Factor3 != 0) and (HSV == 0))) xor (Gender
== 0)) { (if((sewage disposal != 1) and (((Sibling number == 2) or (((HAV == 1)
or (Mother Psychological disorder != 1)) or (Nutritional Factor2 == 0)) and
((HAV != 1) or (Nutritional Factor1 > 0)))) and (((HAV == 1) xor ((pavimenta
!= 0) or (EBV == 0))) or (Mother Psychological disorder == 0)) or ((Tgondi != 0)
xor (VZV == 0))) xor (Gender != 1))) { (if((age <= 0) and ((Hpilory != 1) and
(dog != 1))) and (Sibling number != 3)) { 0 }else{ 1 }} }else{ 0 }} }else{
(if((age > 0) xor (Nutritional Factor1 != 2)) { (if(Fly at home != 1) { 0 }else{ 1
}} }else{ (if(((Nutritional Factor2 <= 1) or (sewage disposal != 1)) and
(((Sibling number != 3) or ((Nutritional Factor1 != 0) xor (age != 1))) and
((((((((parents_asthma == 0) and (Tgondi == 0)) and (A_lumbricoides0 != 1)) and
(Daycare != 0)) xor (Nutritional Factor1 != 0)) and ((HSV == 1) or
((Nutritional Factor3 <= 1) and (Daycare != 0)) or (Nutritional Factor2 > 1))))
and ((age == 2) or ((Nutritional Factor1 == 1) and ((dog == 1) or (Mother

```

```

Psychological disorder == 0))) xor (GNI >= 1))) or (HAV != 0))) and ((HAV !=
1) or (GNI >= 1))) { 1 } else { 0 } } } }
(if((Nutritional Factor4 == 0) or (((Tgondi != 1) or ((HSV != 1) and
(A_lumbricoides != 1))) xor (Gender == 0))) { (if((BMI <= 1) and (((HAV == 0) or
(Nutritional Factor1 != 0)) and (((((Tgondi != 1) and (((((pavimenta == 1) xor
(GNI == 0)) xor (Nutritional Factor4 < 1)) or (age >= 2)) and (Mother
Psychological disorder != 1))) or (HAV != 0)) or ((BMI == 1) xor (VZV == 0)))
and ((sewage disposal != 1) and ((Gender != 0) and ((HAV != 1) or ((Nutritional
Factor4 == 1) or (GNI == 1)))))) { (if((age <= 0) and ((Hpilory != 1) and
(((pavimenta != 0) or (Nutritional Factor2 != 1)) and (dog != 1))) and (Gender
!= 0)) { 0 } else { 1 } } } else { 0 } } } else { (if((age > 0) xor (Nutritional Factor1
!= 2)) { (if(Fly at home != 1) { 0 } else { 1 } } } else { (if(((Nutritional Factor2
!= 2) or (sewage disposal != 1)) and (((Sibling number < 3) or ((BMI < 2) xor
(age != 1))) and ((((((parents_asthma == 0) and (Tgondi == 0)) and (pavimenta
!= 0)) xor (Nutritional Factor1 != 0)) and ((HSV == 1) or ((Nutritional Factor3
> 0) or (Nutritional Factor2 >= 2)))) and ((age == 2) or ((Nutritional Factor1
== 1) and ((dog != 0) or (Mother Psychological disorder == 0))) xor (GNI ==
1)))) or (HAV != 0))) and ((HAV != 1) or (GNI >= 1))) { 1 } else { 0 } } } }
(if((Nutritional Factor4 == 0) or ((Tgondi != 1) or ((HSV != 1) and
(A_lumbricoides != 1))) xor (Gender == 0))) { (if((BMI != 2) and (((Mother
Psychological disorder == 0) xor (Nutritional Factor2 == 0)) or ((Nutritional
Factor2 != 1) and ((GNI == 0) or (Nutritional Factor1 != 0)))) and (((((Tgondi
!= 1) and (((((pavimenta != 0) xor (GNI < 1)) xor (Nutritional Factor4 < 1)) or
(age >= 2)) and (Mother Psychological disorder != 1))) or (HAV != 0)) or ((BMI
== 1) xor (VZV == 0))) and ((sewage disposal != 1) and ((Gender != 0) and ((HAV
!= 1) or ((Nutritional Factor4 == 1) or (GNI == 1)))))) { (if((age <= 0) and
((Hpilory != 1) and (((pavimenta != 0) or (Nutritional Factor2 != 1)) and (dog
!= 1))) and (Mother Psychological disorder == 0)) { 0 } else { 1 } } } else { 0 }
} else { (if((age > 0) xor (Nutritional Factor1 < 2)) { (if(Fly at home != 1) { 0
} else { 1 } } } else { (if(((Nutritional Factor2 != 2) or (sewage disposal != 1))
and (((Sibling number < 2) or (Sibling number <= 2)) and ((((((parents_asthma
== 0) and (Tgondi == 0)) and (pavimenta != 0)) xor (Nutritional Factor1 > 0))
and ((HSV == 1) or ((Nutritional Factor3 != 0) or (Nutritional Factor2 > 1))))
and ((age >= 2) or ((Nutritional Factor1 == 1) and ((dog != 0) xor (Mother
Psychological disorder == 0))) xor (GNI == 1)))) or (HAV != 0))) and ((HAV !=
1) or (GNI == 1))) { 1 } else { 0 } } } }
(if((Nutritional Factor4 == 0) or ((Tgondi != 1) or ((HSV != 1) and
(A_lumbricoides != 1))) xor (Gender == 0))) { (if((BMI <= 1) and (((Mother
Psychological disorder == 0) xor (Nutritional Factor2 == 0)) or ((Nutritional
Factor2 != 1) and ((GNI == 0) or (Nutritional Factor1 != 0)))) and (((((Tgondi
!= 1) and (((((pavimenta != 0) xor (GNI < 1)) xor (Nutritional Factor4 < 1)) or
(age == 2)) and (Mother Psychological disorder != 1))) or (HAV != 0)) or ((BMI
== 1) xor (VZV == 0))) and ((sewage disposal != 1) and ((Gender != 0) and ((HAV
!= 1) or ((Nutritional Factor4 == 1) or (GNI >= 1)))))) { (if((age == 0) and
((Hpilory != 1) and (((pavimenta != 0) or (Nutritional Factor2 != 1)) and (dog
!= 1))) and (Mother Psychological disorder == 0)) { 0 } else { 1 } } } else { 0 }
} else { (if((age > 0) xor (Nutritional Factor1 < 2)) { (if(Fly at home != 1) { 0
} else { 1 } } } else { (if(((Nutritional Factor2 < 2) or (sewage disposal != 1))
and (((Sibling number < 3) or ((BMI != 2) xor (age != 1))) and
(((((((parents_asthma == 0) and (Tgondi == 0)) and (pavimenta != 0)) xor
(Nutritional Factor1 > 0)) and ((HSV == 1) or ((Nutritional Factor3 != 0) or
(Nutritional Factor2 > 1)))) and ((age == 2) or ((Nutritional Factor1 == 1) and
((dog != 0) xor (Mother Psychological disorder == 0))) xor (GNI == 1)))) or (HAV
!= 0))) and ((HAV != 1) or (GNI == 1))) { 1 } else { 0 } } } }
(if((Nutritional Factor4 == 0) or ((Tgondi != 1) or ((HSV != 1) and
(A_lumbricoides != 1))) xor (Gender == 0))) { (if((BMI <= 1) and (((Mother
Psychological disorder == 0) xor (Nutritional Factor2 == 0)) or (((sewage
disposal != 0) or (water system != 1)) or (Nutritional Factor2 <= 0)) and ((GNI
== 0) or (Nutritional Factor1 != 0)))) and (((((Tgondi != 1) and (((((pavimenta
!= 0) xor (GNI < 1)) xor (Nutritional Factor4 < 1)) or (age >= 2)) and (Mother
Psychological disorder != 1))) or (HAV != 0)) or ((BMI == 1) xor (VZV == 0)))
and ((sewage disposal != 1) and ((Gender != 0) and ((HAV != 1) or (Nutritional
Factor1 >= 1)))))) { (if((age <= 0) and ((Hpilory != 1) and (((pavimenta != 0)
or (Nutritional Factor2 != 1)) and (dog != 1))) and (Sibling number <= 3)) { 0
} else { 1 } } } else { 0 } } } else { (if((age != 0) xor (Nutritional Factor1 != 2)) {
(if(Fly at home != 1) { 0 } else { 1 } } } else { (if(((Nutritional Factor2 < 2) or
(sewage disposal != 1)) and (((Sibling number < 3) or ((BMI <= 2) xor (age !=
1))) and ((((((parents_asthma == 0) and (Tgondi == 0)) and (pavimenta != 0))

```

```

xor (Nutritional Factor1 > 0)) and ((HSV == 1) or ((Nutritional Factor3 > 0) or
(Nutritional Factor2 > 1)))) and ((age >= 2) or (((Nutritional Factor1 == 1) and
((dog != 0) xor (Mother Psychological disorder == 0))) xor (GNI == 1)))) or (HAV
!= 0)))) and ((HAV != 1) or (GNI >= 1))) { 1 } else { 0 } } } }
(if((Nutritional Factor4 == 0) or (((Tgondi != 1) or ((HSV != 1) and
(A_lumbricoides != 1))) xor (Gender == 0)))) { (if((fuma != 1) and (((Mother
Psychological disorder == 0) xor (Nutritional Factor2 == 0)) or (((sewage
disposal != 0) or (water system != 1)) or (Nutritional Factor2 == 0)) and ((GNI
== 0) or (Nutritional Factor1 != 0)))) and (((((Tgondi != 1) and (((pavimenta
!= 0) xor (GNI < 1)) xor (Nutritional Factor4 < 1)) or (age >= 2)) and (Mother
Psychological disorder != 1))) or (HAV != 0)) or ((BMI == 1) xor (VZV == 0)))
and ((sewage disposal != 1) and ((Gender != 0) and ((HAV != 1) or ((Nutritional
Factor4 == 1) or (GNI == 1)))))) { (if((age == 0) and ((Hpilory != 1) and
(((pavimenta != 0) or (Nutritional Factor2 != 1)) and (dog != 1))) and (Sibling
number <= 3)) { 0 } else { 1 } } } else { 0 } } } else { (if((age > 0) xor (Nutritional
Factor1 < 2)) { (if(Fly at home != 1) { 0 } else { 1 } } } else { (if(((Nutritional
Factor2 < 2) or (sewage disposal != 1)) and (((Sibling number < 3) or
((Nutritional Factor4 != 0) xor (age != 1))) and ((((((parents_asthma == 0) and
(Tgondi == 0)) and (pavimenta != 0)) xor (Nutritional Factor1 != 0)) and ((HSV
== 1) or ((Nutritional Factor3 > 0) or (Nutritional Factor2 > 1)))) and ((age >=
2) or ((Nutritional Factor1 == 1) and ((dog != 0) xor (Mother Psychological
disorder == 0))) xor (GNI == 1)))) or (HAV != 0)))) and ((HAV != 1) or (GNI ==
1))) { 1 } else { 0 } } } } }
(if((Nutritional Factor4 == 0) or (((Tgondi != 1) or ((HSV != 1) and
(A_lumbricoides != 1))) xor (Gender == 0)))) { (if((Nutritional Factor3 <= 2) and
(((Mother Psychological disorder != 1) xor (Nutritional Factor2 <= 0)) or
(((sewage disposal != 0) or (water system != 1)) or (Nutritional Factor2 <= 0))
and ((GNI == 0) or (Nutritional Factor1 != 0)))) and (((((Daily calories <
1692,98) and ((Tgondi != 1) and (((pavimenta != 0) xor ((GNI == 0) or (Sibling
number >= 3))) xor (Nutritional Factor4 < 1)) or (age == 2)) and ((Mother
Psychological disorder == 0) or (pavimenta != 0)))) or (HAV != 0)) or ((BMI ==
1) xor (VZV != 1))) and ((sewage disposal != 1) and ((Gender != 0) and ((HAV !=
1) or ((Nutritional Factor4 == 1) or (GNI >= 1)))))) { (if((age <= 0) and
((Hpilory != 1) and (((pavimenta != 0) or (Nutritional Factor2 != 1)) and (dog
!= 1))) and (Sibling number <= 3)) { 0 } else { 1 } } } else { 0 } } } else { (if((age >
0) xor (Nutritional Factor1 != 2)) { (if(Fly at home != 1) { 0 } else { 1 } } } else {
(if(((Mother Psychological disorder == 0) or (sewage disposal != 1)) and
(((Sibling number != 3) or ((Nutritional Factor4 >= 0) xor (age != 1))) and
(((((((parents_asthma == 0) and (Tgondi == 0)) and (pavimenta != 0)) xor
(Nutritional Factor1 > 0)) and ((HSV == 1) or ((Nutritional Factor3 != 0) xor
(Nutritional Factor2 > 1)))) and ((age == 2) or ((Nutritional Factor1 == 1) and
((dog != 0) or (Mother Psychological disorder == 0))) xor (GNI == 1)))) or (HAV
!= 0)))) and ((HAV != 1) or (GNI >= 1))) { 1 } else { 0 } } } } }
(if((Nutritional Factor4 == 0) or (((Tgondi != 1) or ((Change bed linen != 0)
and ((sewage disposal != 1) and (HSV != 1))) and (A_lumbricoides != 1))) xor
(Gender == 0))) { (if((Daily calories < 1692,98) and (((Mother Psychological
disorder != 1) xor (Nutritional Factor2 <= 0)) or (((sewage disposal != 0) or
(water system != 1)) or (Mother Psychological disorder != 1)) and ((GNI == 0) or
(Nutritional Factor1 > 0)))) and (((((Nutritional Factor1 <= 2) and ((Tgondi !=
1) and (((pavimenta != 0) xor ((GNI == 0) or (Sibling number == 3))) xor
(Nutritional Factor4 < 1)) or (age == 2)) and ((Mother Psychological disorder !=
1) or (pavimenta == 1)))) or (HAV != 0)) or ((BMI == 1) xor (VZV != 1))) and
((sewage disposal != 1) and ((Gender != 0) and ((HAV != 1) or (Nutritional
Factor1 != 0)))))) { (if((age == 0) and ((Hpilory != 1) and ((pavimenta != 0)
or (Nutritional Factor2 != 1)) and (dog != 1))) and (Gender == 1)) { 0 } else { 1
} } } else { 0 } } } else { (if((age >= 1) xor (Nutritional Factor1 != 2)) { (if(Fly at
home != 1) { 0 } else { 1 } } } else { (if(((Nutritional Factor2 <= 1) or (sewage
disposal != 1)) and (((Sibling number < 3) or ((VZV == 1) xor (age != 1))) and
(((((((parents_asthma == 0) and (Gender == 0)) and (pavimenta != 0)) xor
(Nutritional Factor1 > 0)) and ((HSV == 1) or (Nutritional Factor3 != 0) or
(Nutritional Factor2 > 1)))) and ((age >= 2) or ((Nutritional Factor1 == 1) and
((dog != 0) or (Mother Psychological disorder == 0))) xor (GNI == 1)))) or (HAV
!= 0)))) and ((HAV != 1) or (GNI == 1))) { 1 } else { 0 } } } } }
(if((Nutritional Factor4 == 0) or (((Tgondi != 1) or ((Change bed linen != 0)
and ((sewage disposal != 1) and (HSV != 1))) and (A_lumbricoides != 1))) xor
(Gender == 0))) { (if((Daily calories < 1692,98) and (((Mother Psychological
disorder != 1) xor (Nutritional Factor2 <= 0)) or (((sewage disposal != 0) or
(water system != 1)) or (Mother Psychological disorder != 1)) and ((GNI == 0) or

```

```

(Nutritional Factor1 > 0)) and (((((Nutritional Factor1 <= 2) and ((Tgondi !=
1) and (((((pavimenta != 0) xor ((GNI == 0) or (Sibling number == 3))) xor
(Nutritional Factor4 < 1)) or (age == 2)) and ((Mother Psychological disorder !=
1) or (pavimenta == 1)))))) or (HAV != 0)) or ((BMI == 1) xor (VZV != 1))) and
((sewage disposal != 1) and ((Gender != 0) and ((HAV != 1) or ((Nutritional
Factor4 >= 1) or (GNI == 1))))))){ (if(((age == 0) and ((Hpilory != 1) and
(((pavimenta != 0) or (Nutritional Factor2 != 1)) and (dog != 1)))) and (Gender
== 1)){ 0 }else{ 1 } } }else{ 0 } } }else{ (if((age >= 1) xor (Nutritional Factor1
!= 2)){ (if(Fly at home != 1){ 0 }else{ 1 } } }else{ (if(((Nutritional Factor2
<= 1) or (sewage disposal != 1)) and (((Sibling number < 3) or ((VZV == 1) xor
(age != 1))) and ((((((parents_asthma == 0) and (Gender == 0)) and (pavimenta
!= 0)) xor (Nutritional Factor1 > 0)) and ((HSV == 1) or ((Nutritional Factor3
!= 0) or (Nutritional Factor2 > 1)))) and ((age >= 2) or ((Nutritional Factor1
== 1) and ((dog != 0) or (Mother Psychological disorder == 0))) xor (GNI ==
1)))) or (HAV != 0))) and ((HAV != 1) or (GNI == 1))){ 1 }else{ 0 } } } }
(if((Nutritional Factor4 <= 0) or ((Tgondi != 1) or (((Change bed linen != 0)
and ((sewage disposal != 1) and (HSV != 1))) and (A_lumbricoides != 1))) xor
(Gender == 0))){ (if((sewage disposal != 1) and (((Mother Psychological
disorder != 1) xor (Nutritional Factor2 <= 0)) or (((sewage disposal != 0) or
(water system != 1)) or (Mother Psychological disorder != 1)) and ((GNI == 0) or
(Nutritional Factor1 > 0)))) and (((((Nutritional Factor1 <= 2) and ((Tgondi !=
1) and (((((pavimenta != 0) xor ((GNI <= 0) or (Sibling number >= 3))) xor
(Nutritional Factor4 < 1)) or ((age >= 2) xor ((Sibling number >= 3) and
(pavimenta != 0)))) and ((Mother Psychological disorder != 1) or (pavimenta ==
1)))) or (HAV != 0)) or ((BMI == 1) xor (VZV != 1))) and ((sewage disposal !=
1) and (Gender != 0) and ((HAV != 1) or (Nutritional Factor4 == 1) or (GNI ==
1))))))){ (if(((age == 0) and ((Hpilory != 1) and ((pavimenta != 0) or
(Nutritional Factor2 != 1)) and (dog != 1)))) and (VZV == 0)){ 0 }else{ 1 } }
}else{ 0 } } }else{ (if((age > 0) xor (Nutritional Factor1 != 2)){ (if(Fly at
home != 1){ 0 }else{ 1 } } }else{ (if(((Nutritional Factor2 <= 1) or (sewage
disposal != 1)) and (((Sibling number != 3) or ((VZV == 1) xor (age != 1))) and
(((((((parents_asthma == 0) and (Gender == 0)) and (pavimenta != 0)) xor
(Nutritional Factor1 > 0)) and ((HSV == 1) or ((Nutritional Factor3 > 0) xor
(Nutritional Factor2 > 1)))) and ((age >= 2) or ((Nutritional Factor1 == 1) and
((dog != 0) or (Mother Psychological disorder == 0))) xor (GNI == 1)))) or (HAV
!= 0))) and ((HAV != 1) or (GNI == 1))){ 1 }else{ 0 } } } }
(if((Nutritional Factor4 == 0) or ((Tgondi != 1) or ((HSV != 1) and
(A_lumbricoides != 1))) xor (Gender == 0))){ (if((((age == 0) and (Change bed
linen == 1)) xor (Gender == 1)) or (Nutritional Factor1 > 1)) and (((Mother
Psychological disorder != 1) xor (Nutritional Factor2 == 0)) or (((sewage
disposal != 0) xor (water system != 1)) or (Mother Psychological disorder != 1))
and ((GNI <= 0) or (Nutritional Factor1 > 0))) and (((Daily calories <=
1692,98) and ((Tgondi != 1) and (((pavimenta != 0) xor ((GNI == 0) or (Sibling
number >= 3))) xor (Nutritional Factor4 < 1)) or ((age == 2) xor ((Sibling
number >= 3) and (pavimenta != 0)))) and ((Mother Psychological disorder != 1)
or (pavimenta == 1)))) or (HAV != 0)) or ((BMI == 1) xor (VZV != 1))) and
((sewage disposal != 1) and ((Gender != 0) and ((HAV != 1) or ((Nutritional
Factor4 >= 1) or (GNI == 1))))))){ (if((age == 0) and ((Hpilory != 1) and
((pavimenta != 0) or (Nutritional Factor2 != 1)) and (dog != 1))) and (VZV ==
0)){ 0 }else{ 1 } } }else{ 0 } } }else{ (if((age > 0) xor (Nutritional Factor1 <
2)){ (if(Fly at home != 1){ 0 }else{ 1 } } }else{ (if(((Nutritional Factor2 !=
2) or (sewage disposal != 1)) and (((Sibling number < 3) or ((BMI != 2) xor (age
!= 1))) and ((((((parents_asthma == 0) and (Tgondi == 0)) and (pavimenta != 0))
xor (Nutritional Factor1 != 0)) and ((HSV == 1) or ((Nutritional Factor3 > 0) or
(Nutritional Factor2 > 1)))) and ((age == 2) or ((Nutritional Factor1 == 1) and
((dog != 0) or (Mother Psychological disorder == 0))) xor (GNI == 1)))) or (HAV
!= 0))) and ((HAV != 1) or (GNI >= 1))){ 1 }else{ 0 } } } }
(if((Nutritional Factor4 <= 0) or ((Tgondi != 1) or (((Change bed linen != 0)
and ((sewage disposal != 1) and (HSV != 1))) and (A_lumbricoides != 1))) xor
(Gender == 0))){ (if((((age == 0) and (Change bed linen == 1)) xor (sewage
disposal != 1)) or (Nutritional Factor1 > 1)) and (((Mother Psychological
disorder != 1) xor (Nutritional Factor2 <= 0)) or (((sewage disposal != 0) or
(water system != 1)) or (Mother Psychological disorder != 1)) and ((GNI == 0) or
(Nutritional Factor1 > 0)))) and (((((Nutritional Factor1 <= 2) and ((Tgondi !=
1) and (((((pavimenta != 0) xor ((GNI <= 0) or (Sibling number >= 3))) xor
(Nutritional Factor4 < 1)) or ((age == 2) xor ((Sibling number >= 3) and
(pavimenta != 0)))) and ((Mother Psychological disorder != 1) or (pavimenta ==
1)))) or (HAV != 0)) or ((BMI == 1) xor (VZV != 1))) and ((sewage disposal !=

```

```

1) and ((Gender != 0) and ((HAV != 1) or ((Nutritional Factor4 >= 1) or (GNI == 1))))))){ (if(((age == 0) and ((Hpilory != 1) and (((pavimenta != 0) or (Nutritional Factor2 != 1)) and (dog != 1)))) and (VZV == 0)){ 0 }else{ 1 }} }else{ 0 }} }else{ (if((age != 0) xor (Nutritional Factor1 != 2)){ (if(Fly at home != 1){ 0 }else{ 1 }} }else{ (if((((Nutritional Factor2 <= 1) or (HAV != 1)) and (((Sibling number < 3) or ((VZV == 1) xor (age != 1))) and (((((((parents_asthma == 0) and (Gender == 0)) and (pavimenta != 0)) xor (Nutritional Factor1 != 0)) and ((HSV == 1) or ((Nutritional Factor3 != 0) xor (Nutritional Factor2 > 1)))))) and ((age >= 2) or (((Nutritional Factor1 == 1) and ((dog != 0) or (Mother Psychological disorder == 0))) xor (GNI == 1)))) or (HAV != 0)))) and ((HAV != 1) or (GNI == 1))) { 1 }else{ 0 }} } } }
(if((Nutritional Factor4 == 0) or (((Tgondi != 1) or (((Change bed linen != 0) and ((sewage disposal != 1) and (HSV != 1))) and (A_lumbricoides != 1))) xor (Gender == 0))) { (if((((age == 0) and ((Change bed linen == 1) xor (((HAV == 1) xor ((age == 0) or (EBV == 0))) or (Mother Psychological disorder == 0)) or ((Tgondi != 0) xor (VZV == 0)))) or (Nutritional Factor1 > 1)) and (((Mother Psychological disorder != 1) xor (Nutritional Factor2 <= 0)) or (((sewage disposal != 0) xor (water system != 1)) or (Nutritional Factor2 < 2)) and ((GNI == 0) or (Nutritional Factor1 != 0)))) and (((fuma != 1) and ((Tgondi != 1) and (((pavimenta != 0) xor ((GNI <= 0) or (Nutritional Factor2 == 0))) xor (Nutritional Factor4 < 1)) or ((age >= 2) xor ((Sibling number == 3) and (pavimenta != 0)))) and (GNI <= 1))) or (Nutritional Factor3 >= 2)) or ((BMI == 1) xor (VZV != 1))) and ((sewage disposal != 1) and ((Gender != 0) and ((HAV != 1) or ((Nutritional Factor4 >= 1) or (GNI == 1))))))){ (if(((age == 0) and ((Hpilory != 1) and (((pavimenta != 0) or (Nutritional Factor2 != 1)) and (dog != 1)))) and (Sibling number <= 2)){ 0 }else{ 1 }} }else{ (if(Fly at home != 1){ 0 }else{ 0 }} } } }else{ (if((age >= 1) xor (Nutritional Factor1 != 2)){ (if(Fly at home != 1){ 0 }else{ 1 }} }else{ (if((((Nutritional Factor2 <= 1) or (sewage disposal != 1)) and (((Sibling number < 3) or ((HSV == 1) xor (age != 1))) and (((((((parents_asthma == 0) and (Gender == 0)) and (pavimenta != 0)) xor (Nutritional Factor1 > 0)) and ((HSV == 1) or ((Nutritional Factor3 > 0) or (Nutritional Factor2 > 1)))))) and ((age >= 2) or (((Nutritional Factor1 == 1) and ((dog != 0) xor (Mother Psychological disorder == 0))) xor (GNI == 1)))) or (HAV != 0)))) and ((HAV != 1) or (GNI == 1))) { 1 }else{ 0 }} } } }
(if((Nutritional Factor4 == 0) or (((Tgondi != 1) or (((HAV != 1) and ((sewage disposal != 1) and (HSV != 1))) and (A_lumbricoides != 1))) xor (Gender == 0))) { (if((((age == 0) and (Change bed linen == 1)) xor (((HAV == 1) xor ((pavimenta != 0) or (EBV == 0))) or (Mother Psychological disorder == 0)) or ((Tgondi != 0) xor (VZV == 0)))) or (Nutritional Factor1 > 1)) and (((Mother Psychological disorder != 1) xor (Nutritional Factor2 <= 0)) or (((sewage disposal != 0) or (water system != 1)) or (Nutritional Factor2 < 2)) and ((GNI == 0) or (Nutritional Factor1 != 0)))) and (((Daily calories < 1692,98) and ((Tgondi != 1) and (((pavimenta != 0) xor ((GNI <= 0) or (Sibling number > 2))) xor (Nutritional Factor4 != 1)) or ((age >= 2) xor ((Sibling number >= 3) and (pavimenta != 0)))) and (((Fly at home != 1) or (Mother Psychological disorder == 0)) or (A_lumbricoides == 1)))) or (HAV != 0)) or ((BMI >= 1) xor (VZV != 1))) and ((sewage disposal != 1) and ((Gender != 0) and ((HAV != 1) or ((Nutritional Factor4 >= 1) or (GNI == 1))))))){ (if(((age == 0) and ((Hpilory != 1) and (((pavimenta != 0) or (Nutritional Factor2 != 1)) and (dog != 1))) and (Sibling number <= 2)){ 0 }else{ 1 }} }else{ 0 }} }else{ (if((age > 0) xor (Nutritional Factor1 != 2)){ (if(Fly at home != 1){ 0 }else{ 1 }} }else{ (if((((Nutritional Factor2 <= 1) or (sewage disposal != 1)) and (((Sibling number < 3) or ((Daily calories <= 2434,17) xor (age != 1))) and (((((((parents_asthma == 0) and (Tgondi == 0)) and (pavimenta != 0)) xor (Nutritional Factor1 > 0)) and ((HSV == 1) or ((Nutritional Factor3 != 0) or (Nutritional Factor2 > 1)))))) and ((age == 2) or ((Nutritional Factor1 == 1) and ((dog != 0) or (Mother Psychological disorder == 0))) xor (GNI >= 1)))) or (HAV != 0)))) and ((HAV != 1) or (GNI >= 1))) { 1 }else{ 0 }} } } }

```