# Chapter 7:

# Feature selection in AP

**CHAPTER 7 Contents:**

[7.1. Feature Selection Problem 169](#_Toc125225728)

[7.2. Feature Categorization 170](#_Toc125225729)

[7.3. General Procedure of Feature Selection 170](#_Toc125225730)

[7.4. Feature Selection Methods 172](#_Toc125225731)

[7.5. Metaheuristic Algorithms 173](#_Toc125225732)

[7.6. An Introduction to Genetic Algorithm 176](#_Toc125225733)

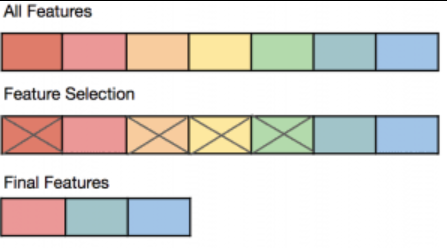
[7.7. Basics of Genetic Algorithm 177](#_Toc125225734)

[7.8. The Initial Design of Genetic Algorithm 178](#_Toc125225735)

[7.9. Feature Selection Using Genetic Algorithm 180](#_Toc125225736)

[7.10. Genetic Algorithm's Application in Psychology 181](#_Toc125225737)

[7.11. Genetic Algorithm’s Application in Neural Network Sciences 182](#_Toc125225738)



Abstract: In machine learning there are often a large range of possible features to use for classification into groups. This chapter concentrates on methods of feature selection to narrow down characteristics of interest to create more parsimonious and cost effective models. Aspects of feature selection such as choice of method (wrapper, embedded and filter), evaluation functions used to identify an optimal subset of features and validation of model fit are described. Worked examples using a random forest algorithm in R for classification is presented which introduces diagnostics to show how the most important classification features are selected. Feature selection is then considered for a specific set of models which use algorithms that treat the data as if it is genetic information based upon pairs of chromosomes. These models incorporate concepts in genetic models such as parents, children, reproduction and mutation. An example of the use of this genetic approach to feature selection in machine learning is illustrated in R using two 10 item subscales from a questionnaire measuring sexual pain.

Keywords: Feature selection; Group classification; Random forest algorithm; Metaheuristic algorithms; Genetic algorithms; Questionnaire data; Neural networks

## Feature Selection Problem

In most real-world problems, a large fraction of the sample features is not useful and have a negative impact on model performance. Therefore, in machine learning, it is important to identify good features that affect model accuracy. Feature selection methods aim to find the smallest possible subset of features that is necessary and sufficient for determining the target. In fact, the distribution of the selected features should be close to the distribution of the main class (Dash, Liu,1997).

Since this is an optimization problem, choosing the right feature selection method is in itself a refinement.

**Advantages of Feature Selection:**

Feature selection has several advantages, such as:

* Better classifiers: Since most redundant features generate noise and affect model accuracy, the main advantage of feature selection is improved accuracy.
* Better knowledge discovery: Feature selection also provides insights into the data. Separating important and influential features from unusable ones can be very informative.
* Reducing the cost of data collection: where data collection is costly, such as in medical applications, identifying a minimal set of features for classification saves money.
* Computational costs: Identifying appropriate subsets of features simplifies the model and reduces computational and implementation costs.
* Dimensionality: Existing theories suggest that a higher number of features exponentially increases the volume of data required to build a classifier (Cunningham, 2021).

## Feature Categorization

In general, features are divided into three subgroups:

1. Related Features: These features directly affect the output and target and cannot be replaced by other features.
2. Unrelated Features: These features are unrelated to the output and create random values in the sample.
3. Feature Redundancy: In this case, one feature replaces another. Therefore, these features do not provide any additional information.

According to these definitions, feature selection should identify features with high correlation, while redundant features should be removed as much as possible (Dash, Liu,1997).

## General Procedure of Feature Selection

Typically, there are four stages and functions in the process of feature selection, which are shown in relation to each other in the following diagram.

1. Generation procedure
2. Evaluation function
3. Stopping criterion
4. Validation procedure (Tang, et al, 2014)

Generation

Evaluation

Validation

Stopping criterion

Subset

Goodness of the subset

Yes

No

Original feature set

1. Stages of Feature Selection with Validation

**Generation Procedure**

The generator function is actually a search function that generates various subsets and is checked by the evaluation function. In this case, it can start from the initial featureless state, with a set of features, or with a random subset.

**Evaluation Function**

After generation, the subset of features is evaluated using different methods. In this phase, the goodness of a set is examined and the new subset is compared with the old one. If it is better, it replaces the older subset. In this case, finding a suitable subset of features depends directly on the evaluation function used, because if the evaluation function does not provide a suitable subset, the subset of features will never find an optimal value.

**Stopping Criterion**

The criterion needed to end and stop the algorithm can be one of the following:

* 1. A certain number of features
  2. A certain number of iterations
  3. When adding or removing a feature does not result in a better subset
  4. When the optimal subset is reached based on the evaluation function.

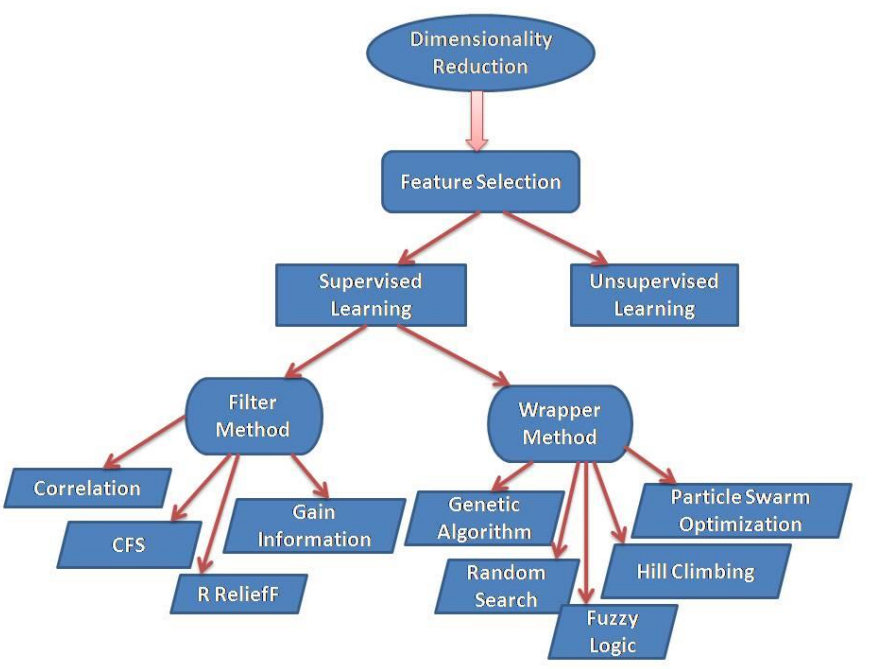
**Validation Function**

Although this function is not part of feature selection, it checks the validity of the subsets. The results of the data can be evaluated based on prior knowledge (Ansari, N. 2021)

## Feature Selection Methods

In many studies, feature selection methods are divided into three categories (Ansari, N. 2021):

1. Wrapper Method: This method is considered as a prediction method, and its performance is crucial for subset selection. At each stage, a subset of features is selected, based on which the performance of the machine learning algorithm is evaluated and the performance result is used for feature selection.
2. Embedded Method: These methods simultaneously use machine learning algorithms with model fitting for feature selection.
3. Filter Method: These methods measure the importance of features without machine learning algorithms. In this method, features related to the input are selected to obtain the output. The filter method typically employs the ranking method. Different features are ranked in the order of their acceptability. Due to its simplicity, the ranking strategy can be applied to any application. Before the features enter the classification phase, they are ranked, i.e., filtered. Incompatible elements are filtered into compatible elements. Each ranked element must have a unique feature to identify that class. This method is faster than packing methods, but may give poor results if the data is not properly correlated. These methods are also used when the spatial dimensions of the feature are large and the computational cost of the wrapper method is high, so it is more economical to use the filter method.



1. Classification of feature selection methods from (Ansari, N. 2021)

Genetic algorithm, a wrapper method, will be discussed in detail below.

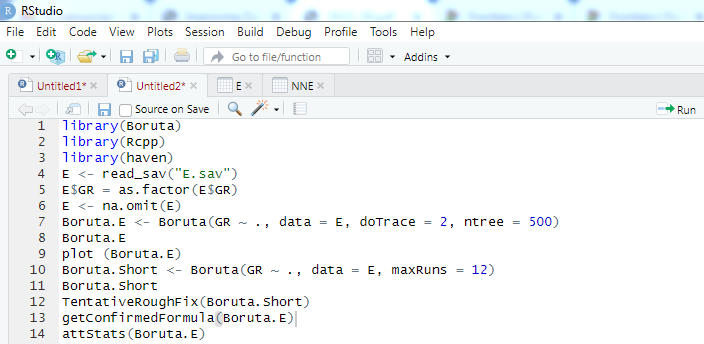
This section will first introduce the basics of the genetic algorithm and its feature selection stages.

### Practical example using R

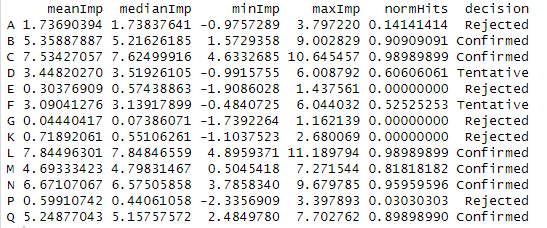
An artificial psychologist tries to implement a classification model, but he realizes that the number of predictor variables may decrease the accuracy of the model. It means some variables may be noisy variables, therefore, he decides to select the Feature Selection feature to determine a subset of the most important variables. In this model, he decides to first determine the most important features in predicting group membership with the Feature Selection algorithm. He wants to determine which of the 13 variables A,B,C,D,E,F,G,K,L,M,N,P and Q, are the most important variables. He uses the Boruta method. This is a wrapper method based on a Random forest algorithm. The random forest itself is based on the decision tree algorithm. A decision tree is a sequence of steps that are performed in the training phase. Random forest is an ensemble of decision tree algorithms. The random forest algorithm trains hundreds of decision trees, each of which only has access to a random set of columns in the data (variables).

Boruta is a technique that takes this randomization much further. Based on this method, it captures all the features in the data that are related to the target variable. ( Kursa, & Rudnicki, ,2010) .

Listing 1 R codes for feature selection implement

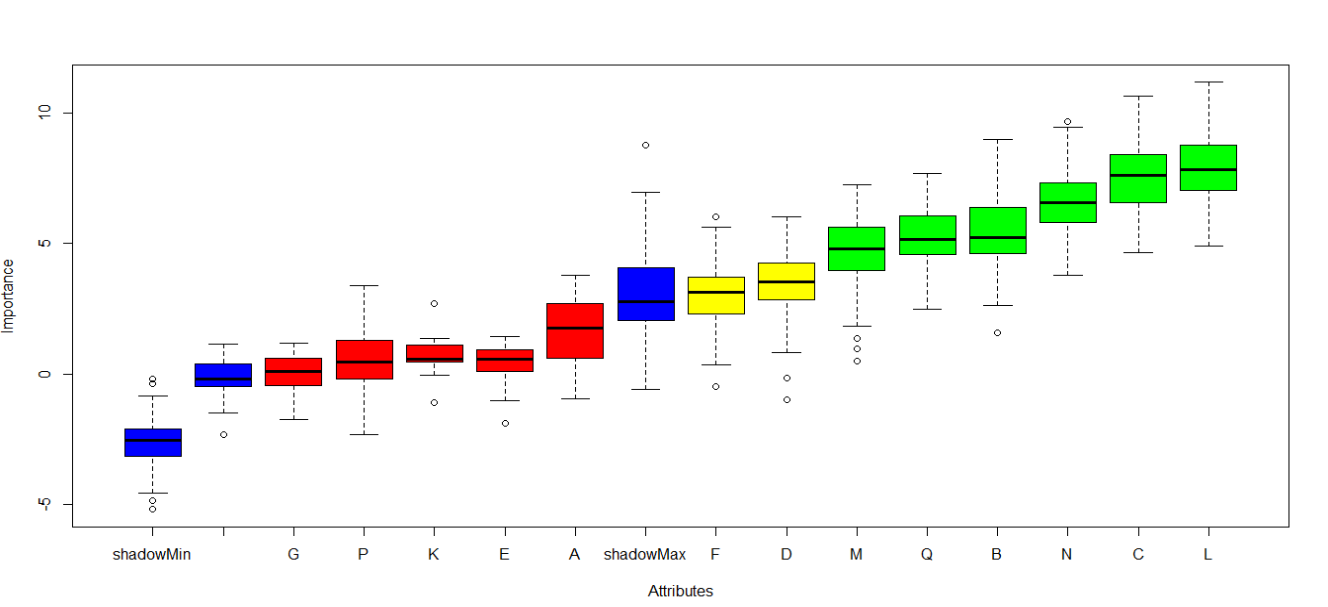


Listing 1 shows the R codes for feature selection implemented using the Boruta algorithm. The results are shown in Figure 7.4. This plot shows that the variables M, Q, B, N, C, and L are the most important variables in determining group membership, respectively.



1. R output of feature selection using Boruta algorithm

Figure 7.3 shows the R output, where meanIMP shows the average importance, which is the highest average importance of each predictor to show how it is related to group classification. The last column is about making a decision about which variable remains in the model or is removed. This column has three options: confirmed, rejected, and tentative, which is the last option that includes the variables that need further investigation.

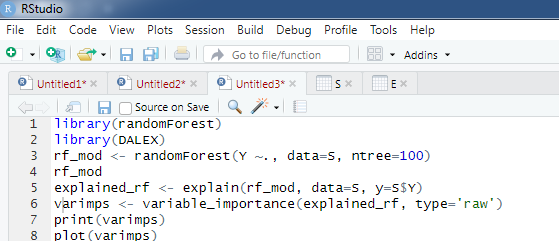


1. R output of feature selection using Boruta algorithm

**Example 2**

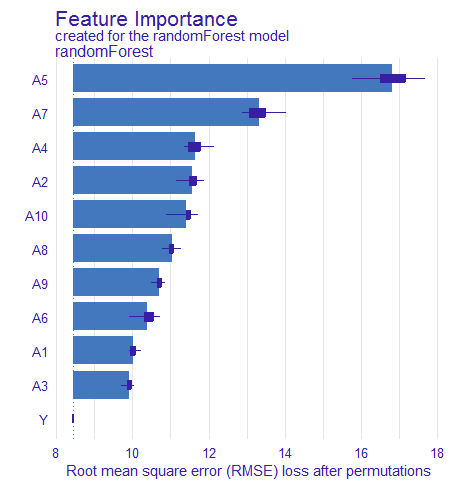
An artificial psychologist wants to predict the post-corona depression rate (y) based on variables A1 to A10. He performs feature selection based on the random forest algorithm using the DALEX Package in R and prepares the necessary codes (Listing 2). This is a powerful package. By using the function variable-dropout (), which determines the importance of a variable based on a dropout loss, how much loss is imposed on the model by removing a variable.

Listing 2 R codes for feature selection implement using Random Forest



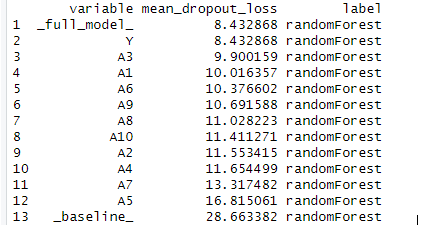
As the importance plot and the output of the software in Figure 7.5 and Figure 7.6 show, variables A5, A7, A4, A2 and A10 are the 5 most important variables of Feature Selection

Figure ----- shows the specifications of the model that the mean of residuals of the model is equal to 0.0384, which indicates the average difference between the actual and predicted y scores.

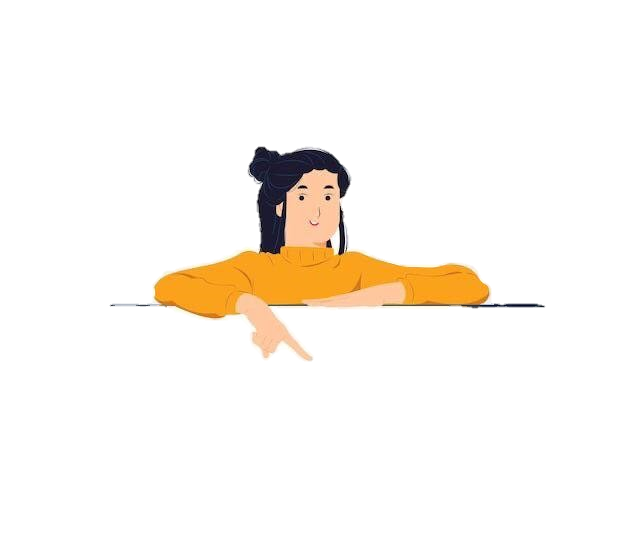
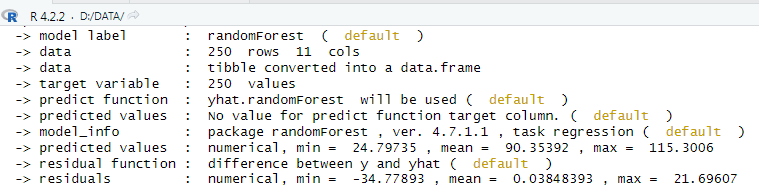


1. Importance plot

Figure 7.6. It shows that the mean loss is maximized if variable A5 is dropped from the model. This dropout loss is the highest value among the 10 predictor variables and is equal to 16.81..



1. Mean dropout –loss of each variable



1. Model summary

As Figure 7.8 indicates, the mean prediction error using this random forest model is 0.038.

## Metaheuristic Algorithms

In general, optimization algorithms are divided into two categories: exact algorithms and approximate algorithms. Exact algorithms provide the optimal solution accurately, but have exponentially longer resolution times. In approximate algorithms, on the other hand, the solution is close to the optimum and takes less time for difficult problems.

Therefore, exact algorithms do not seem to be suitable for many problems. There are three types of approximate algorithms, namely heuristic algorithms, meta-heuristic algorithms, and hyper-heuristic algorithms. In this book, we will discuss the genetic algorithm, which is a meta-heuristic algorithm (Fraser, A. S. 1957).

In meta-heuristic algorithms, the behavior is random, i.e., the search for the solution and the optimal value starts at a random point (i.e., from absolute ignorance to knowledge). The conceptually simple meta-heuristic algorithms can be implemented easily and flexibly. In other words, the algorithms can be easily modified depending on the problem. The main feature of meta-heuristic algorithms is their ability to prevent early convergence of algorithms (Golberg, D. E. 1989).

These algorithms have been successfully applied to various engineering and scientific problems, e.g., in electrical engineering (to find the optimal solution for power generation), industry (planning, transportation, vehicle routing, location determination), civil engineering (for bridge and building design), communications (radar and network design), data mining (classification, prediction, clustering, system modeling), and psychology and cognitive science (to reduce feature dimensionality in questionnaires and for features obtained from brain imaging)( Engelbrecht, A. P. 2007).

Meta-heuristic algorithms are divided into the following two main categories:

**Single Solution-Based Metaheuristic Algorithms:** In these techniques, the problem typically starts with a solution that is updated in iterations. The problem is that one falls into the trap of local optimization and does not fully explore the search space.

**Population (Multiple) Solution-Based Metaheuristic Algorithms**: These algorithms generate a population of solutions and start the optimization process. The population is updated with the number of generations or iterations. These algorithms do not suffer from the problem of single solution based algorithms, i.e., they do not fall into the local optima because multiple solutions help each other and there are many searches in the search space. Moreover, population-based algorithms are used for solving most real-world problems.

Due to the aforementioned properties, much attention is paid to meta-heuristic algorithms. These algorithms are classified into four categories based on their behavior, namely, evolution-based, swarm intelligence-based, physics-based, and human-related algorithms (Agrawal, et al, 2021).

Metaheuristic Algorithms

Evolution based Algorithms

Swarm intelligence based Algorithms

Physics based Algorithms

Human behavior related Algorithms

1. Four categories of Metaheuristic algorithms

**Evolution-Based Algorithms** are inspired by natural evolution and start with a population of random solutions. They combine the best solutions to create new individuals through mutation, combination, and selection of the best population. The genetic algorithm (GA) based on Darwin's evolution is the most popular in this category. This category also includes other algorithms such as evolution strategy, genetic programming, tabu search, and differential evolution (Agrawal, et al, 2021).

**Swarm Intelligence-Based Algorithms** are inspired by the social behavior of insects, animals, fish, or birds, such as the group behavior of birds and how they fly around the search space and find their best location. Other swarm intelligence algorithms include ant colony optimization, the honey bee swarm optimization algorithm, monkey optimization, and more (Agrawal, et al, 2021).

**Physics-Based Algorithms:** These algorithms are inspired by the laws of physics in the world and include simulated annealing (Goldberg, et al, 1987), harmony search (Higashi, Iba, 2003), and so on.

**Human Behavior-Based Algorithms**: These algorithms are purely inspired by human behavior. The approaches humans take to various activities affect their performance. Popular algorithms include the teaching learning-based optimization algorithm (TLBO), the League Championship algorithm, and more (Agrawal, et al, 2021).

As mentioned earlier, meta-heuristic algorithms mimic the principles of natural evolution to search for the optimal solution. The genetic algorithm is a meta-heuristic algorithm rooted in Darwin's theory of evolution. In this theory, a better generation typically emerges from the combination of chromosomes. In the meantime, possible mutations can improve or worsen the generation. The genetic algorithm is explained in detail below.

## An Introduction to the Genetic Algorithm

Many human inventions are inspired by nature, as its evolution over a long period of time means that it always provides the best and most optimal solutions and processes. Given the ever-changing environment around us, the ability to change and adapt to conditions is necessary for any living thing to survive. Phenomena such as natural selection, reproduction, mutation, and symbiosis are involved in this adaptation and evolution. Living organisms also rely on chemical elements to store information about each organism. DNA contains all the necessary information for the reproduction of a living organism and is the known regulator of genetic information. In fact, genetics is the alphabet of nature. Hereditary factors and individual characteristics exist along the chromosome. The human body usually has four bases for making chromosomes from DNA, namely bases A[[1]](#footnote-1) , C[[2]](#footnote-2) , G[[3]](#footnote-3) and T[[4]](#footnote-4) . DNA also has four combinations in DNA, namely TA, AT, GC and CG, which can produce different phenotypes. Consequently, there is a one-to-one correspondence between each living organism and its gene sequence, which is called coding.

The genetic algorithm is a family of "computing models" that are conceptually based on evolution. This field in artificial intelligence is inspired by the evolution of natural living organisms. The genetic algorithm is based on the principle of survival of the fittest and reproduction of the superior. It is often considered as a "function optimizer", that is, it is used to optimize the objective functions in problems. In fact, however, it has other applications.



**Crossover Child Mutation Child**

1. Illustration of crossover child and mutation child

## Basics of the Genetic Algorithm

**Definition of Gene:** In the genetic algorithm, a gene is a parameter of the problem and the genotype is considered as a bit string with a fixed length. The length, which must be specified before optimization, shows the dimensions of the problem and can sometimes be considered as a variable or vector (Wang, L., & Jiang, T. 1994).

**Definition of Chromosome:** A set contains all genes that have been valued and represent a possible solution to the problem. Answers, suitable or unsuitable, are actually a string or sequence of bits that act as natural genes in chromosomes. The coding in bits can be binary or n. In nature, a chromosome is a long and complex string called "deoxyribonucleic acid" or DNA.

**Genetic Population:** A genetic population is a set of chromosomes. Instead of focusing on one point or one chromosome, the genetic algorithm operates on a population of chromosomes that has more desired features than those of the previous stage. The population size indicates the number of chromosomes within a generation. If the population size is too small, the algorithm performs poorly; if it is too large, it slows down. The population size is usually between 20 and 30 chromosomes and can be as large as 50-100 depending on the problem. Note that increasing the population size beyond a certain limit does not help solve the problem faster (Wang, L., & Jiang, T. 1994).

**Fitness Function:** the genetic algorithm has a stage to evaluate the obtained answers and determine their value. Namely, the suitability of the answer is checked with the fitness function, with more suitable answers having a higher fitness value. Suitable solutions with a higher chance of survival have higher fitness values. They have a greater probability of producing children and more sequences. Therefore, optimal chromosomes have a greater chance of joining with other chromosomes.

**Evolutionary Operators**

In genetics, evolutionary operators include reproduction, mutation, and selection.

**Reproduction:** This is the reproduction of genetic information between chromosomes, which means sharing information between chromosomes. In this operator, children inherit the characteristics of their parents, which are then reproduced to create a better generation. Reproduction usually occurs between a pair, but can also occur in several parents. In this case, the traits of more than two parents are passed to the child. This operator improves fitness and its implementation is related to chromosome coding.

**Mutation**: The mutation operator generates and introduces new genetic material and contributes to the diversity of the population, which can be achieved by adding a random value. In nature, mutation does not usually produce good results, but it is necessary for continuous evolution, and the results may eventually include a favorable outcome.

**Selection**: Selection is the stage of choosing the parents and ideal promotion of the good gene. In selecting the best individual, all individuals are included in the population based on merit, and individuals with greater fitness are marked and selected for the next generation.

## The Initial Design of the Genetic Algorithm

Now that we have explained the basics, it is time to implement the genetic algorithm. First, the variables should be introduced according to the problem, i.e., the encoded chromosomes. Then, a fitness function for the chromosomes is considered based on the objective function. This algorithm first generates a population of chromosomes (the initial population is randomly selected). Then, the fitness of each chromosome is checked and selected according to the following stages.

Stage One: An appropriate number of chromosome pairs are selected for the next stages based on their fitness.

Stage Two: Parents are selected and reproduced to produce a population of children. In this stage, the reproduction operator is applied. The chromosomes change through the reproduction process. The parent chromosomes are randomly exchanged through the crossover process. Therefore, the children have some of the characteristics of their parents.

Stage Three: The members of the population are selected to apply the mutation and create the mutation population. As mentioned earlier, there is also the rare process of mutation that changes the characteristics of living organisms. For example, there could be an error in the chromosome copying process, mitosis. Often mutations destroy living organisms, but in the long run they create new and better species. Mutation often creates traits that are certainly not possible in reproduction. In other words, this is the way of entering new information.

Stage Four: The original population is merged with the population of children and mutants to create a new original population. Now the fitness of the new children is calculated.

In stage five, if the stopping conditions are not met, the process is repeated from stage two and the new population enters the next stages as the initial population. In this case, the generated data structure (chromosomes) is evaluated. The chromosomes that can better represent the optimal solution or goal of the problem have a greater chance of reproduction than others, i.e. they are given more opportunities.

**Stopping Conditions**

1. A threshold or optimal value is considered for the cost function.
2. There is also a criterion for the number of iterations or one relative to time.

The stall iteration is stopped if it exceeds a limit. In this case, when changes are less than a specified level and the result is not better, the process stops and starts from another point (Sahdra, et al, 2016).

Initialization

Individuals test for fitness

Individuals selection for

Reproduction

Reproduction

Final generation

Termination

Yes

No

1. Diagram of Genetic Algorithm Implementation

## Feature Selection Using the Genetic Algorithm

The genetic algorithm is implemented to select optimal features. This is a robust machine learning method that limits the number of features without significant loss of information.

Conventional scale abbreviation methods must manually check multiple criteria to select elements. However, the genetic algorithm is a fully automatic and complex optimization tool. Fortunately, its software implementation is relatively simple (e.g., the Precis tool in Python and the GAabbreviate tool in R) ( Sahdra, et al, 2016).

## The Genetic Algorithm's Application in Artificial Psychology

Self-report criteria are a common form of assessment in psychology. Although not entirely accurate and appropriate because of substantial bias due to social desirability, arbitrary criteria, and interference with these reports, these reports are very common, simple, and inexpensive to use. At the same time, many questionnaires are long and time-consuming, which leads to people not completing the questionnaires accurately and faithfully. In contrast, with short questionnaires, people are more likely to fill them out honestly. In one meta-analysis, the same short and long questionnaires were randomly distributed to people, and researchers found that participants who received a shorter questionnaire had a greater return rate (Rolstad, Adler, & Rydén, 2011). Another study found an inverse correlation between questionnaire length and response rate (Fan & Yan, 2010).

In addition, shorter questionnaires can help develop science through description, prediction, and causal inference. In fact, lower numbers of variables can be useful in predicting theories and examining cause and effect. Thus, although short questionnaires are more interesting, in many cases they do not meet psychometric standards. In fact, traditional methods of shortening questionnaires require careful consideration of competing factors. This approach requires finding items with high autocorrelation, low cross-loading, low correlated uniqueness, low chance of missingness, high face validity of construct coverage, and high internal consistency of the resulting scale.

Reducing the 15-item questionnaire to six items results in over 2,500 possible permutations that must be considered. Because of this complexity, researchers must rely on unwritten heuristics that often result in poor performance.

One efficient method is to use machine learning to find items that explain the greatest diversity. The genetic algorithm has recently been used to shorten questionnaires in several domains, including personality (Yarkoni, 2010), values (Sandy et al., 2014), psychopathy (Eisenbarth, Lilienfeld, & Yarkoni, 2015), experiential avoidance (Sahdra et al., 2016), and body image- acceptance (Basarkod et al., 2018).

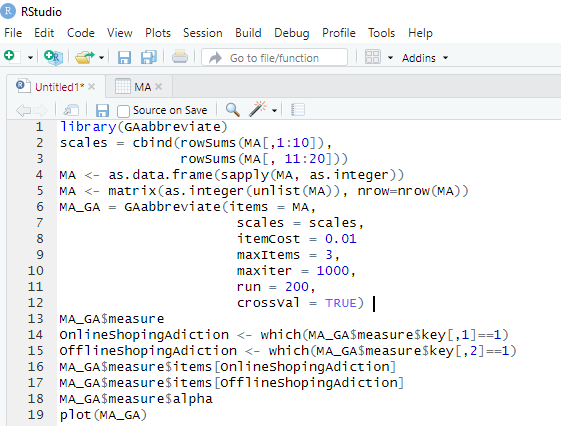
In the algorithm, the items are like genes and a set of items that make up a scale are like chromosomes. As explained above, the algorithm first generates a sample of chromosomes that forms a random set of subscales from the whole. Then the chromosomes compete for fitness. The fitness scale in shortening a questionnaire includes chromosomes that explain more variability in the full questionnaire. Then, the selection operator is used to remove the chromosomes with the lowest fitness from the gene pool and reproduce those with the highest fitness (usually the top 5%) to create a new set of chromosomes (Yarkoni, 2010).

As mentioned in the stages of the genetic algorithm, chromosomes also mutate and genes crossover. Mutation replaces random elements in the chromosome. In reproduction, the genes of one fit item are exchanged with the genes of another. After mutation and reproduction, the new set of chromosomes is generated in terms of fitness and examined according to the fitness function. This process is repeated until the algorithm finds a stable solution. Items shortened by the genetic algorithm often yield scores that are as valid as those derived from traditional psychometric approaches (Sandy et al., 2014; Noetel,et al,2019)

### Practical example using R

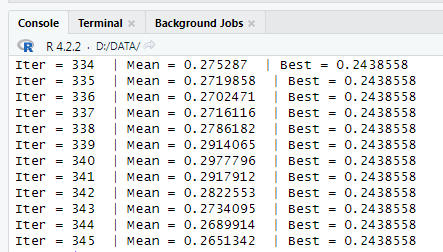
An artificial psychologist tries to select the best items based on the genetic algorithm from a psychological scale for measuring sexual pain that has 20 items with two 10-item subscales. A classical psychologist with psychometric techniques can select the best items based on corrected item-total correlations, factor loadings, the correlation of each item with a criterion or using the discriminant coefficient in Item Response Theory (IRT). Although this works, the artificial psychologist seeks to select the important items with the genetic algorithm in such a way as to keep the cost function to a minimum. The cost function in this case was proposed by Yarkoni in 2010 and explained in the previous sections of this chapter. The genetic algorithm iteratively tries to select and mate strong individuals (solutions) who are more likely to survive.

Listing 3 R codes to implement the genetic algorithm



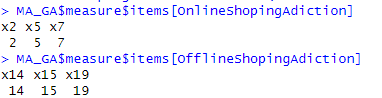
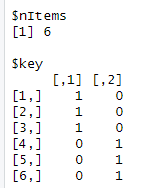
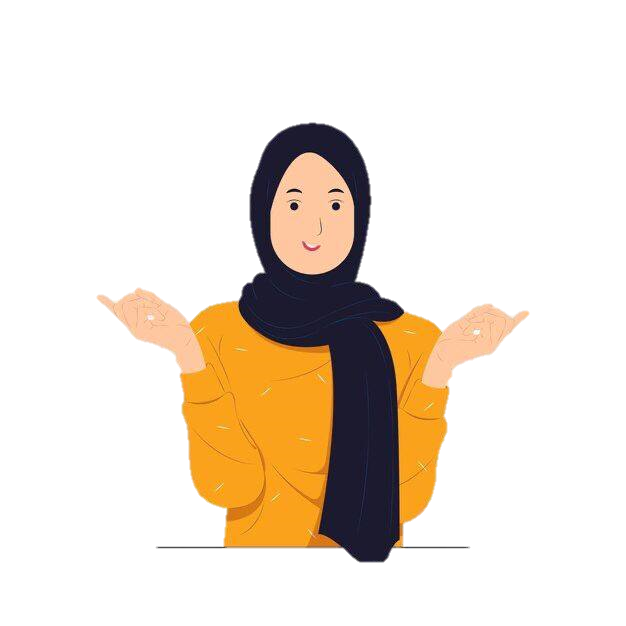
Listing 3 shows the R codes to run the genetic algorithm. Item cost indicates the cost of each item, which is equal to 0.05 by default. Max Items is the maximum number of items that we want to remain in each subscale or factor in the end. In this example, the artificial psychologist considers it to be 3 items, maxIter indicates the maximum number of iterations, which in this case is set to 1000 items and if this function is not able to find a solution, its number can be increased. Cross-Validation is also done for unbiasedness.. If the GA abbreviate function finds the optimal solution before the maximum number of iterations, it will stop automaticallyotherwise, the iteration continues until it reaches an optimal solution. Figure 7.12 shows which items belong to which dimension of the factor or scale. It is marked with $key in the output.

As the figure 7.12 shows, this algorithm has reached the optimal solution in the iteration equal to .24, and the GA abbreviate function terminates..



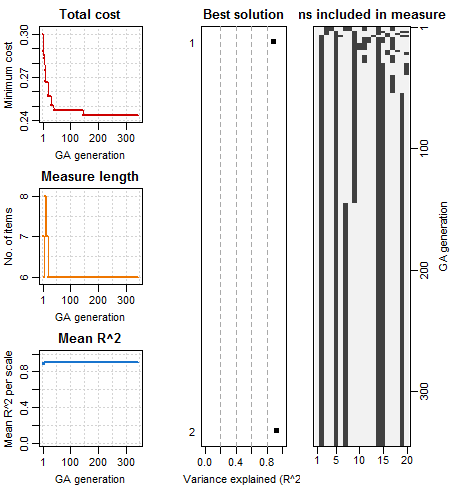
1. Output of Genetic Algorithm Implementation

The output of this method shows that items number 2, 5, 7, 14, 15, and 19 are selected (3 items for each subscale) (Figure 7.13).



1. Output of Genetic Algorithm Implementation

Figure 7.14 shows a visual summary of the search process using the Plot function. Diagnostic plots, which are the three graphs on the left, show how Total Cost and the length of the scale and the average value of R² change during the search process. The middle graph of this figure shows the percentage of explained variance, that is, R² for the best solution, which in this example is excellent and close to 0.9 for each subscale.



1. Plot of search process

And finally, the Plot on the right side of the figure shows which items are selected during the Search process. Items 2, 5, 7, 14, 15, and 19 are the items that were finally selected. You can easily see them in this plot Kursa, M. B., & Rudnicki, W. R. 2010).

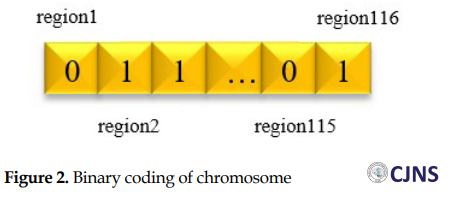
## The Genetic Algorithm’s Application in Neural Network Sciences

In a new approach, there are studies for identifying biomarkers from the functional matrix formation strategy (see Chapter 5). Functional matrices are considered a good method for diagnosing many mental disorders, such as schizophrenia, autism, and MCI. After obtaining the functional matrix, it is necessary to use feature selection techniques, one of which is the genetic algorithm discussed here. In fact, rs-fmri data is used and a functional matrix is provided based on time series to detect abnormal patterns even from a younger age, which can be very beneficial for early diagnosis and treatment.

One of the main challenges in fMRI analysis is the high dimensionality of data. Although data from the functional connectivity matrix (FCM) provide comprehensive information about connections between different regions, the high dimensionality of feature space poses several challenges in classification and analysis. Feature selection is hence considered a critical issue in biomarker detection. Feature selection aims to reduce dimensionality, accelerate operation, increase the accuracy of classification algorithms, and better understand the results.

Mapping the brain's functional connections is used for determining the function and correlation of time series between different brain regions. Brain FCM is determined based on correlation coefficients between regions. Here, for instance, the brain is divided into 116 regions. As a meta-heuristic optimization algorithm, the genetic algorithm considers a set of feature subsets in each iteration and can find an optimal or near-optimal feature subset.

As mentioned, the reduction of data dimensionality using genetic algorithm includes chromosome coding, selection, reproduction, and mutation. Each chromosome represents an array of zeros and ones, i.e. the total number of features (length 1x116). The values 0 and 1 respectively denote the absence and presence of the respective region in the FCM calculation (Sadeghian, et al, 2021).



1. Binary coding of chromosome (Sadeghian, et al, 2021).

A set of the initial population is created and the population’s fitness is evaluated based on the fitness function. For example, here, the KNN classifier's accuracy based on the selected regions can be considered a fitness function. Reproduction is applied to a pair of parents. After producing a member in a new population, some genes mutate randomly, which allows for searching the entire feature space and escaping the local optima. Finally, the genetic algorithm stops after a specific number of iterations, and the solution with the highest classification accuracy determines the final feature subset. This method has been implemented for distinguishing autism patients from healthy individuals, removing 49 regions to improve the genetic algorithm's diagnostic accuracy by 9%.

The GAabbreviate uses the GA package (Scrucca, 2013) to efficiently implement Yarkoni’s (2010) scale abbreviation cost function:

**(7.1)**

where I represent a user-specified fixed item cost, k represents the number of items retained by the GA (in any given iteration), s is the number of subscales in the measure, wi are the weights (by default wi = 1 for any i) associated with each subscale (if there are any subsets to be retained), and is the amount of variance in the ith subscale that can be explained by a linear combination of individual item scores. Decreasing or increasing the value of yields longer or shorter measures respectively. When the cost of each individual item retained in each generation outweighs the cost of a loss in explained variance, the GA yields a relatively brief measure. When the cost of each individual iterm is low, the GA yields a relatively longer measure maximizing explained variance (Yarkoni, 2010).

1. Adenine [↑](#footnote-ref-1)
2. Cytosine [↑](#footnote-ref-2)
3. Guanine [↑](#footnote-ref-3)
4. Thymine [↑](#footnote-ref-4)