

FUZZY LOGIC LEARNING METHODS IN STUDENT EDUCATION

Peter Grabusts

Rezekne Academy of Technologies, Latvia

***Abstract.** There is a rapidly growing interest in Artificial Intelligence applications in various modern areas. Students are very interested in modern data mining methods such as artificial neural networks, fuzzy logic and clustering. Teaching experience in study work shows that students perceive graphical information better than analytical relationships during learning process. Many training courses operate with models that were previously only available in mathematics disciplines. The solution would be to use the Matlab package to implement different models in Artificial Intelligence areas. Often, an analytical solution or simulation model is much simpler than a visual Matlab model, but it provides an insight into the usefulness of using such models for prospective training purposes. In previous articles, the author has provided examples of how Matlab's capabilities can be used in economic studies, artificial neural networks, and clustering. Fuzzy logic methods are often undeservedly forgotten, although the implementation of their algorithms is relatively simple and can be implemented even for students. In the research part of the study the modelling capabilities in data mining studies are demonstrated with fuzzy logic algorithms and real examples.*

***Keywords:** data analysis, fuzzy logic, Matlab, modelling, teaching.*

Introduction

The purpose of studying various simulation models is to provide students with theoretical knowledge and practical skills in applying simulation methods to research in solving specific problems of modelling real applications and programming - cluster analysis, neural networks or fuzzy learning methods (Grabusts, 2019). During the study of such methods, students get acquainted with the means of simulation of the processes of functioning of systems, master the methods of simulation, typical stages of modelling the processes that make up the chain: building a conceptual model and its formalization - model algorithmization and its computer implementation - simulation experiment and interpretation of simulation results; master practical skills in the implementation of modelling algorithms to study the characteristics and work of complex systems (Kay, 1984; Karel & Tomas, 2015; Karris, 2006; Smith, 2013; Xue & Chen, 2013).

As a result, students acquire knowledge of:

- the basics of the theory and practice of simulation;
- main classes of domain system models, technology for their modelling.

They should be able to:

- use the method of simulation for research, design of various systems;
- develop schemes of modelling algorithms for processes and systems;
- implement models using the application package of modelling software Matlab Fuzzy Logic.

Simulation is the most powerful and versatile method for studying and evaluating the effectiveness of systems which behavior depends on the influence of random factors.

Implementing such features in programming languages is a very difficult matter. Currently, there are quite a few software products that allow you to simulate processes. You can set aside a software product that allows you to solve these problems very effectively - the MATLAB package, which contains the visual modelling tool Fuzzy Logic Toolbox. This is a tool that allows you to quickly simulate the system and obtain indicators of the expected effect and compare them with the energy required to achieve them (Esfandiari, 2013; Kiusalaas, 2016).

The aim of the article is to show Matlab suitability for the purpose of visualizing simulation models of Fuzzy Logic Toolbox. To reach the aim, the following research tasks have been set: identification of Matlab possibilities for fuzzy logic realization; demonstrate visualization models on the basis of examples. Common research methods are used in this research: descriptive research method, statistical method, mathematical modelling, fuzzy logic algorithms.

Fuzzy Logic concept

Classical logic only admits conclusions that are either true or false. The mathematical theory of fuzzy sets and fuzzy logic are generalizations of classical set theory and classical formal logic (Dubois & Prade, 1980). These concepts were first proposed by the American scientist Lotfi Zadeh in 1965. The main reason for the emergence of a new theory was the presence of fuzzy and approximate reasoning when a person described processes, systems, objects (Zadeh, 1965; Zadeh, 1989).

Membership function is a characteristic of a fuzzy set. Let $MFC(x)$ denote the degree of belonging to the fuzzy set C , which is a generalization of the concept of the characteristic function of an ordinary set. Then a fuzzy set C is a set of ordered pairs of the form $C = \{MFC(x) / x\}, MFC(x) [0,1]$. The value $MFC(x) = 0$ means the absence of membership in the set, 1 - full membership (Mamdani, 1977).

We formalize the inaccurate definition of “hot coffee”. The use of fuzzy logic is mainly applied when some elements are not precisely defined by a classification; e.g. are 35^o considered as hot for a cup of coffee? It might be too extreme to say that 35^o is not hot for a cup of coffee. However, it might more precise to say that 35^o are considered 40% hot for a cup of coffee.

The temperature scale in degrees Celsius will act as X (the domain of reasoning). Obviously, it will vary from 0 to 100 degrees. A fuzzy set for the term “hot coffee” may look like this:

$$C = \{0/0^0; 0/10^0; 0/20^0; 0,15/30^0; 0,30/40^0; 0,60/50^0; 0,80/60^0; 0,90/70^0; 1/80^0; 1/90^0; 1/100^0\}.$$

So, coffee with a temperature of 60^o belongs to the set “hot” with a degree of membership 0.80. For one person, coffee at 60^o may turn out to be hot; for another, it may not be too hot. This is where the fuzziness of the task of the corresponding set indicates itself.

For fuzzy sets, as for ordinary ones, the basic logical operations are defined. The most basic ones needed for calculations are intersection and union. To describe fuzzy sets, the concepts of fuzzy and linguistic variables are introduced.

A fuzzy variable is described by the set (N, X, A) , where N is the name of the variable, X is the universal set (the domain of reasoning), A is the fuzzy set on X .

Possibilities of the software package Matlab Fuzzy Logic Toolbox

Of particular interest for modelling is the Fuzzy Logic Toolbox, designed specifically for modelling dynamic systems. It has a library of standard graphic blocks with built-in mathematical functions. Sometimes it is called a visual modelling tool (Fuzzy Logic, 2019).

Although Matlab is intended primarily for solving engineering, scientific and technical problems, the possibilities of its application are practically unlimited. The input parameters are set in an interactive mode by graphically assembling the circuit diagram of the elementary blocks, resulting in a model of the system under study. The blocks included in the model are interconnected, both by information and by management. The type of connection depends on the type of block and the work logic of the model.

The Fuzzy Logic Toolbox is the MATLAB extension package that contains tools for designing fuzzy logic systems.

The package allows to create expert systems based on fuzzy logic, conduct clustering with fuzzy algorithms, as well as design fuzzy neural networks. The package includes a graphical interface for interactive step-by-step design of fuzzy

systems, command line functions for developing programs, as well as special blocks for building fuzzy logic systems in Simulink (Karel & Tomas, 2015).

All functions of the package are written in the open language MATLAB, which allows to control the execution of algorithms, change the initial code, and also create its own functions and procedures.

Fuzzy Logic Toolbox features:

- graphical interface for interactive step-by-step design of fuzzy systems;
- functions for creating expert systems based on fuzzy logic;
- standard types of expert systems of fuzzy logic (Mamdani, Sugeno);
- functions for neuroadaptive and fuzzy clustering with learning;
- inclusion of fuzzy systems in the Simulink model.

When modelling, the principle of visual programming is implemented, according to which the user on the screen from the library of standard blocks creates a model of a device, process or system and performs calculations. At the same time, unlike classical modelling methods, the user does not need to thoroughly study the programming language and numerical methods of mathematics, but rather general knowledge is required when working on a computer, and, of course, knowledge of the subject area in which he works.

Having created the model in this way, running it, you can observe the simulation results. When modelling, the user can choose a solution method, as well as a method for changing the model time (with a fixed or variable step). During the simulation, it is possible to monitor the processes occurring in the system. Simulation results can be presented in the form of graphs or tables.

Research part

Fuzzy Logic Toolbox provides MATLAB functions for analysing, designing, and simulating systems based on fuzzy logic. Functions are provided for many common methods, including fuzzy clustering and adaptive neuro-fuzzy learning. That lets you model complex system behaviours using simple logic rules, and then implement these rules in a fuzzy inference system. You can use it as a stand-alone fuzzy inference engine (Fuzzy logic, 2019).

A series of studies were carried on to demonstrate the suitability of the Fuzzy Logic Toolbox for visualization of simulation models of various engineering disciplines. It should be noted that often the analytical solution is much simpler than the visual model, but in perspective it gives an idea of the usefulness of using such models.

The Tipping Problem: this example creates a fuzzy system using on a two-input, one-output tipping problem based on tipping practices. Given a number between 0 and 10 that represents the quality of service at a restaurant (where 10 is

excellent), and another number between 0 and 10 that represents the quality of the food at that restaurant. For this article, the example is adapted from (Fuzzy Example, 2019).

The starting point is to write down the three rules of tipping:

- If the service is bad or the food is bad, then tip is small.
- If the service is good, then tip is average.
- If the service is excellent or the food is good, then tip is big.

Assume that a small tip is 5%, an average tip is 10%, and a big tip is 20%.

In the Matlab Command Window run *fuzzy*. The Fuzzy Logic Designer opens and displays a diagram of the fuzzy inference system with the names of each input variable on the left and those of each output variable on the right, as shown in Fig. 1.

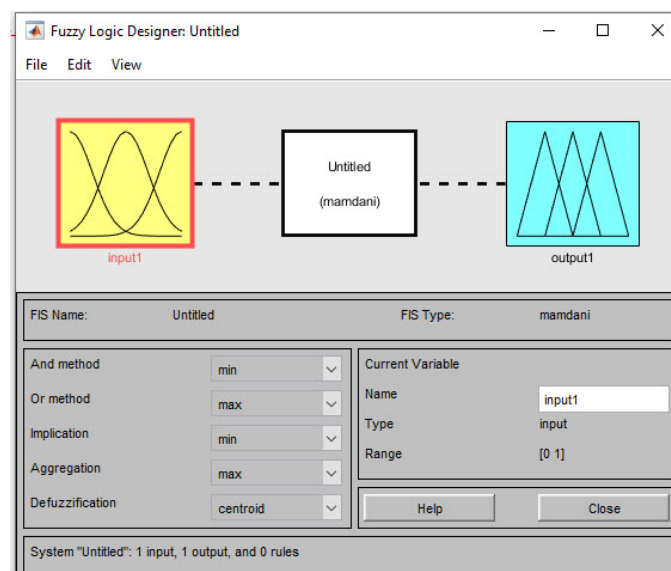


Figure 1 Fuzzy logic designer window

In this example, we construct a two-input, one output system. The two inputs are “service” and “food”. The one output is “tip”.

The First step is defining the variables

Click on *input1* and change the name to “service” then add a new variable “food” to *Edit / Add Variable / Input*. Change *Otput1* to “tip” (see Fig.2).

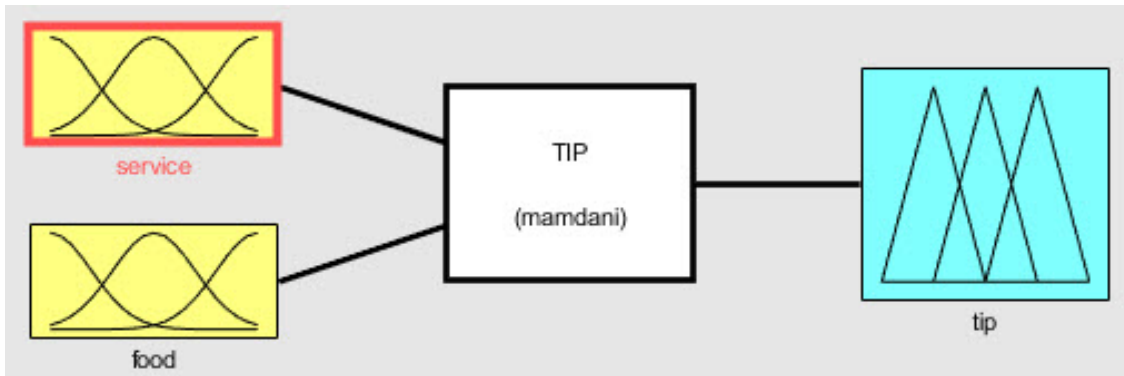


Figure 2 Fuzzy logic model variables

Step Two - Membership Functions (Edit / Membership Functions)

Next, define the membership functions associated with each of the variables. To do this, open the Membership Function Editor. The Membership Function Editor is the tool that lets display and edit all of the membership functions associated with all of the input and output variables for the entire fuzzy inference system.

For the “service” enter the following data (see Fig. 3):

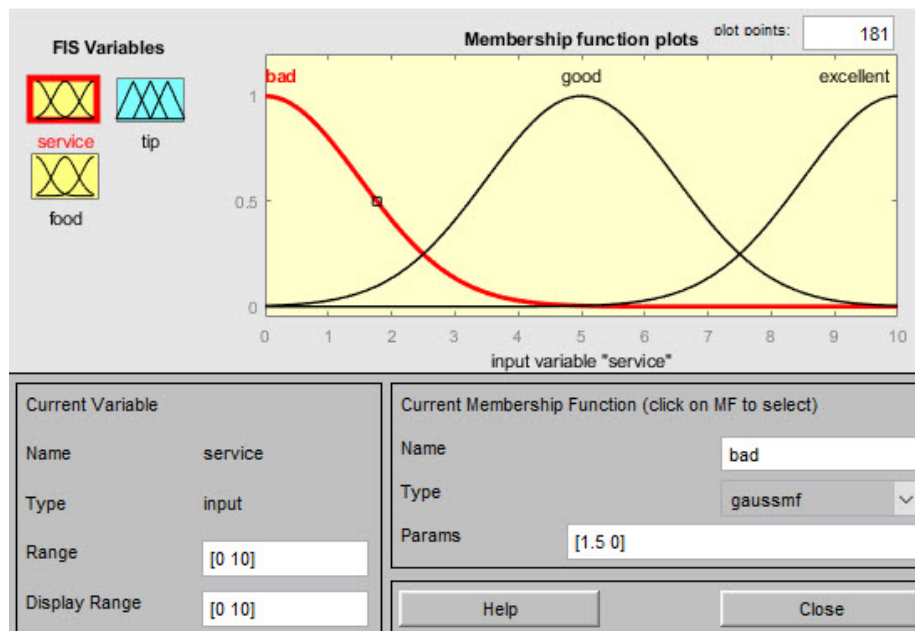


Figure 3 Membership function editor for variable “service”

For “food”, enter the following data (see Fig. 4):

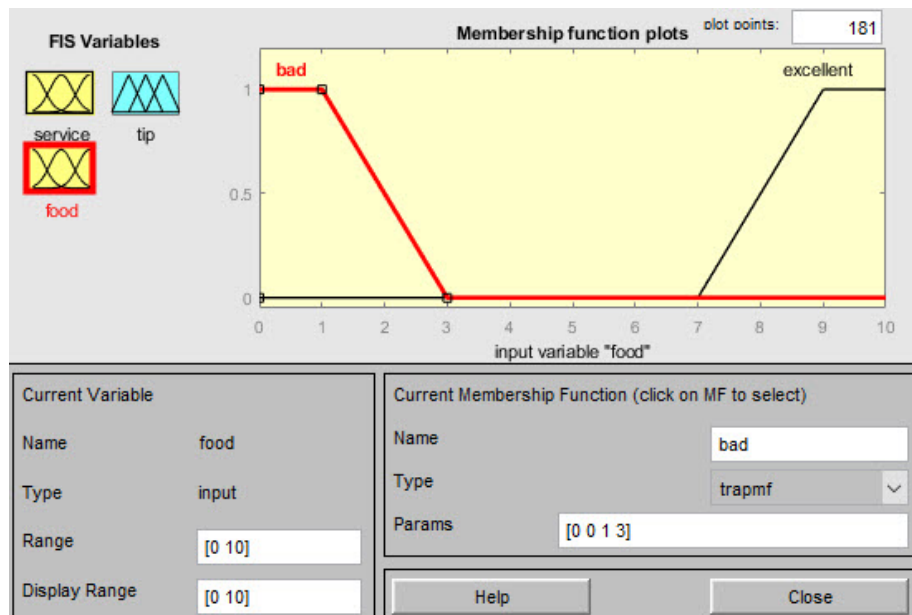


Figure 4 Membership function editor for variable “food”

For “tip” enter following data: for “small” [0 5 10], for “average” [5 10 15], for “big” [15 20 25] (see Fig. 5):

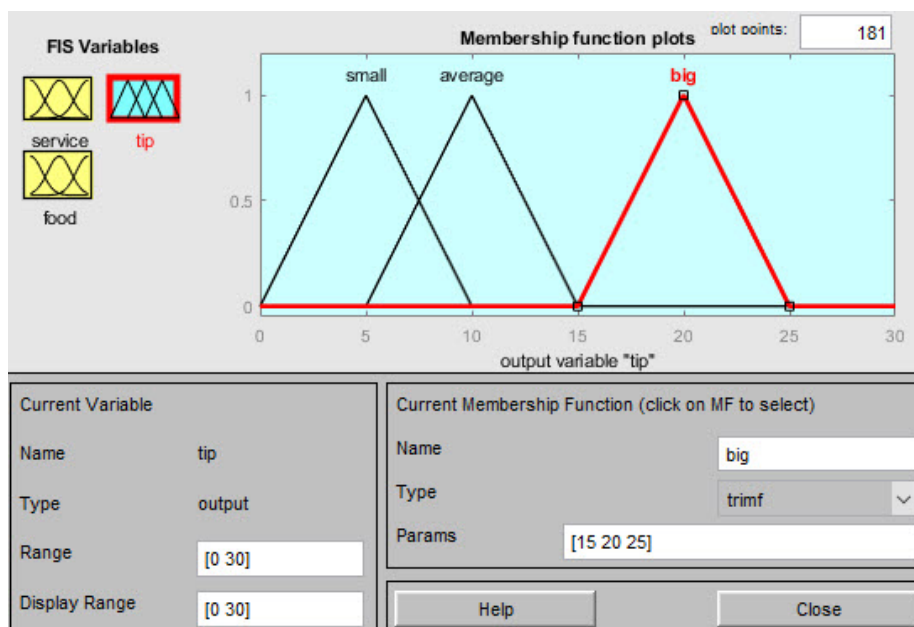


Figure 5 Membership function editor for variable “tip”

Step Three - Rule Editor (Edit/ Rules/ Add Rule)

The result is shown in Fig. 6.

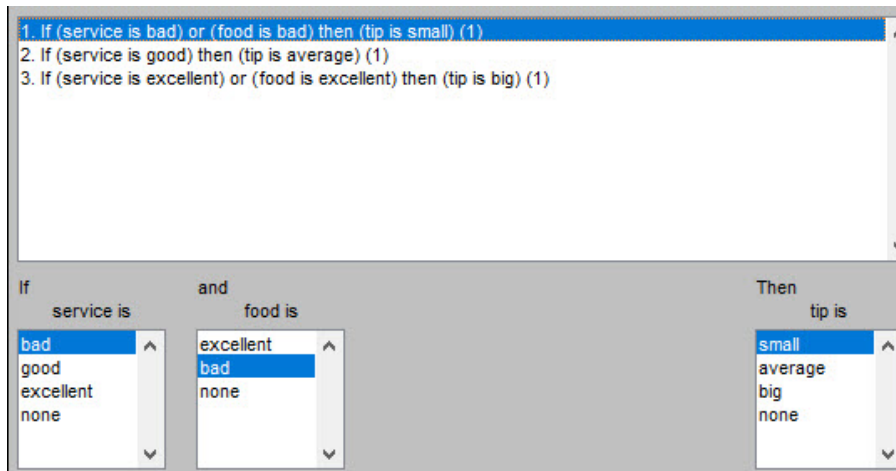


Figure 6 Rule editor window

At this point, the fuzzy inference system has been completely defined, in that the variables, membership functions, and the rules, necessary to calculate tips, are in place.

Step Four – View Rules (see Fig. 7)

The Rule Viewer displays a roadmap of the whole fuzzy inference process. The Rule Viewer allows you to interpret the entire fuzzy inference process at once. The Rule Viewer also shows how the shape of certain membership functions influences the overall result.

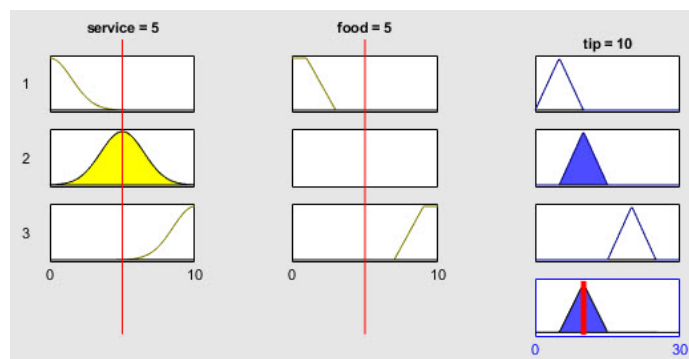


Figure 7 Rule viewer window

Step Five - Checking the acquired Rules.

By moving the vertical red line for “service” and “food” with the mouse, you can verify the accuracy of the rules.

- a) If the “service” is bad or the “food” is bad, then “tip” is small – result “tip” is 5.03 (see Fig. 8).

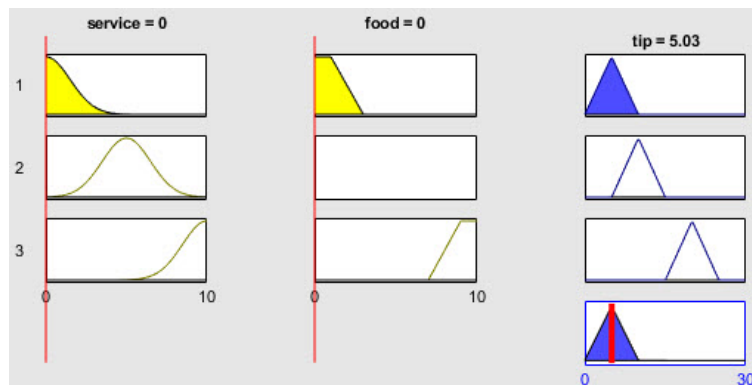


Figure 8 Result for first rule

- b) If the “service” is good, then “tip” is average – result “tip” is 12.5 (see Fig. 9).

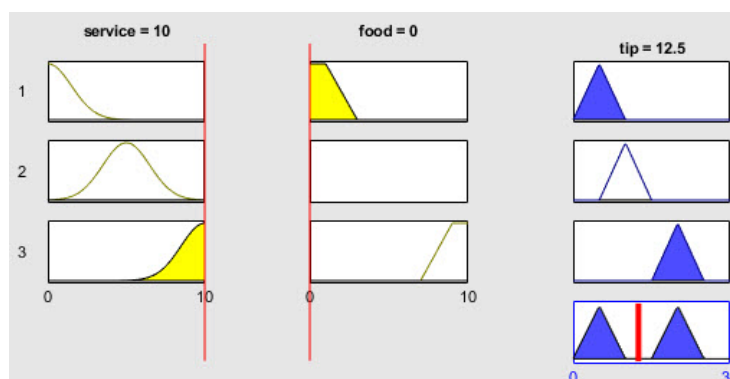


Figure 9 Result for second rule

- c) If the service is “excellent” or the “food” is good, then “tip” is big – result is 19.9 (see Fig. 10).

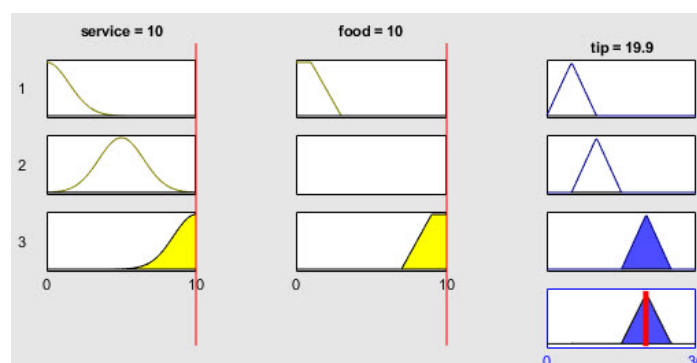


Figure 10 Result for third rule

The graphical depiction of “tip” versus “food” quality is shown in Fig. 11.

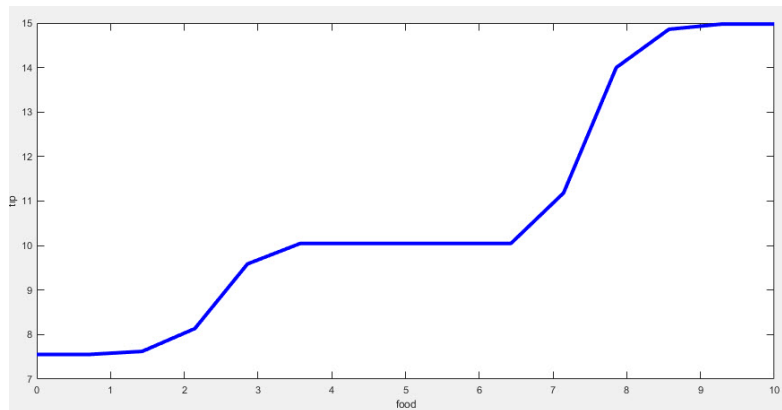


Figure 11 “Tip” dependence of “food” quality

Other variants can be tested. In this system “service” is valued higher than “food” quality. For example, for “service = 10, food = 3” the system recommends “tip” = 20, but for “service” = 3, “food” = 10 the system recommends “tip” = 15. It means that the fuzzy logic system is working correctly.

Conclusions

This work substantiates the usefulness of introducing simulation models in the initial learning process, when simulation models can also be introduced for the acquisition of analytical relationships for modelling purposes. This allows not only to understand the possibilities of using formulas, but also to visualize different relationships graphically.

The following research was carried out in this work: in the field of data analysis, Matlab's Fuzzy Logic package is explored with the aim of finding out how it solves practical problems. These types of packages make it easier for the user to take working decisions, increasing the efficiency and quality of taken decisions.

It has been concluded that fuzzy logic class tasks can be successfully applied in the early stages of data analysis. Such systems can show good results in information selection and analysis.

Thus, it can be concluded that Matlab Fuzzy Logic modelling tool is a very suitable tool not only for calculations in engineering, but it can also serve as a visualization tool for simulation models in various artificial intelligence applications.

References

- Dubois, D., & Prade, H. (1980). *Fuzzy sets and systems: theory and applications*. Academic Press, New York.
- Esfandiari, R.S. (2013). *Numerical methods for engineers and scientists using MATLAB*. Chapman & Hall/CRC.
- Grabusts, P. (2019). The possibilities of clustering learning methods in student education. *Proceedings of the International scientific Conference „Society. Integration. Education”*, Rezekne, May, 24-25, Vol. 5., 344-354. DOI: <http://dx.doi.org/10.17770/sie2019vol5.3723>
- Fuzzy Example. (2019). Retrieved from <https://es.mathworks.com/help/fuzzy/building-systems-with-fuzzy-logic-toolbox-software.html#brzqs45>
- Fuzzy Logic Toolbox. (2019). Retrieved from <https://www.mathworks.com/products/fuzzy-logic.html>
- Kay, C. (1984). *Mathematics for computer programmers*. New Jersey: Prentice Hall.
- Karel, P., & Tomas, Z. (2015). Multimedia teaching aid for students of basics of control theory in Matlab and Simulink. *Procedia Engineering*, 100, 150–158. <https://doi.org/10.1016/j.proeng.2015.01.353>
- Karris, S.T. (2006). *Introduction to Simulink ® with engineering applications*. Orchard Publications.
- Kiusalaas, J. (2016). *Numerical methods in engineering with MATLAB*. 3e. Cambridge University Press.
- Mamdani, E.H. (1977). Applications of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers*, 26(12), 1182-1191. DOI: <https://doi.org/10.1109/TC.1977.1674779>
- Smith, D. (2013). *Engineering computation with MATLAB*. 3e. Pearson Education Inc.
- Xue, D., & Chen, Y. (2013). *System simulation techniques with MATLAB and Simulink*. John Wiley & Sons, Inc.
- Zadeh, L.A. (1965). Fuzzy sets. *Informations and Control*, 8, 338-353. DOI: [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
- Zadeh, L.A. (1989). Knowledge representation in fuzzy logic. *IEEE Transactions on Knowledge and Data Engineering*, 1, 89-100. DOI: <https://doi.org/10.1109/69.43406>