Fuzzy Logic: Machine Learning Assignment 4

Alan Smith and Manas Gaur
Wright State University

November 29, 2017

1 Introduction

There have been various domains where binary encoding have really solved various complex problems. For instance Logic gates, transistor, binary classification problem etc. The commonality between these application areas is the very fact that problem can be represented in yes/no or 0/1 form. But, these encoding fails to provide a representation of natural language problem. For instance, Chris is taller than Steve but relatively shorter than Simon. In this example "taller", "relatively shorter" does not convey any useful meaning until they are augmented with a real or integral number. For example, Chris is 0.5 taller than Steve but 0.4 shorter than Simon. These non-zero, non-one values are termed as Fuzzy values. They are useful in scenarios where the output is not always zeros and ones. The theory of fuzzy was coined by Prof. Lotfi Zadeh and it served as a model of uncertainty in human interaction. In this assignment, we will be using flu dataset to show the expressiveness of fuzzy model. In the flu dataset, there are variables/features which are represented by Likert Scale and ternary values. We will try to model these Likert scale based features to ternary value based feature. The assignment is structured as follows: First, we will define the notion of membership function and provide an example of the same from our assignment. Secondly, we will exemplify the concept of Fuzzy rules and de-fuzzification. Thirdly, we will provide details of our implementation environment. Fourthly we will show the results that were obtained when fuzzy modeling and execution was done over the Flu dataset. Finally, we will conclude with a conclusion.

2 Features for Fuzzy Model

In this section, we detail about the features that are used in the fuzzy model. Flu dataset has 18 features of which we have selected 8 features. Selection of these features is governed by their datatype. The inputs to these features are Likert scale values in the range of [1,5] or in the range of [1,9], where 1 is poor and 5 or 9 is excellent. Our output feature *Sick* is a ternary feature having input belonging to the set {0,1,2}.

	Vacc.	**HWQ**	**HWF**	**SD**	**NTF**	**RE**	**PD**	**Sick**
mean	3.4	4.0	3.2	2.9	2.5	4.2	3.4	1.0
std	1.4	1.0	1.1	1.2	1.3	1.2	1.2	0.7
min	1.0	1.0	1.0	1.0	1.0	1.0	1.0	-1.5
25%	2.0	3.0	3.0	2.0	2.0	3.0	3.0	1.0
50%	3.0	3.0	3.0	3.0	2.0	5.0	3.0	1.0
75%	5.0	4.0	4.0	3.0	3.0	5.0	4.0	1.0
max	9.0	5.0	5.0	5.0	9.0	9.0	9.0	2.0

Table 1: Basic Statistics of the WPBC Flu dataset. Features:{ Vacc:Vaccine, HWQ:HndWshQual, HWF:HndWshFreq, SD:SocialDist, NTF:NotTchFace, RE:RespEtiq, PD:PrsnlDist, }

Proposed Input Features: Vaccine, HandWshQual, HandWshFreq, NotTchFace, RespEtiq, SocialDist, PrsnlDist. Output Features: Sick For the experiments carried out in this assignment, we have taken *Vaccine* and *Hand Wash Quality (HndWshQual)* as input variables and *Sick* as output variable. Since, the frequency of occurrence of each value (1,2,3,4,5, 1,2,3,4,5,6,7,8,9, 0,1,2) in the input dataset affect the fuzzy inference system, we provide a frequency table 2 showing the number of time each discrete integer has occurred. This shows that the fuzzy inference system has never seen a score of 6/7/8 while being modeled over the flu dataset. These input will test the fuzzy inference system. The output generated from the fuzzy inference sysem for sparse integral values will most likely to possess high errors. Hence there is a need to try experiments with changing the defuzzification methods, Min-Max composition based rule generation (switch between AND and OR) and selection of appropriate membership function.

Likert/Integer Score	Vaccine	HndWshQual	Sick
0			83
1	37	7	213
2	89	42	78
3	83	78	
4	95	147	
5	108	136	
6	0		
7	0		
8	0		
9	2		

Table 2: Frequency table Vaccine, HndWshQual and Sick features values. This table is created so as to give a rough idea, how fuzzy inference system will work. This means, some values occur ones or twice, hence won't affect fuzzy inferecing.

3 Membership Functions

Membership functions can be seen analogous to activation function in a neural network architecture. Given a set of the inputs to membership function, it provides values between 0 and 1. The input set for the membership function is called Universe of Discourse. There is a fairly straightforward transition between set theory and fuzzy membership. This can be formulated as follows:

$$A = \{x | x \leq 7\} \tag{1}$$

$$A_f = \{x, \mu_x | x \epsilon Universe of Discourse\} \tag{2}$$

where A is the normal set and A_f is a fuzzy set.
Most common example to elucidate the concept of the membership function is the height of the people. Let us assume that in the universe of discourse, we have heights in the range

from 1 to 10 feet. The word *tall* can be defined by a stating a range in the Universe of Discourse. This crisp logic is prevalent in set theory but does not mimic real sense. For example, we cannot say one guy a taller than other when the difference between their height in minuscule. To sum it all, fuzzy sets using membership function describe vague concepts (e.g. taller, shorter, hotter, faster etc.). Moreover, membership function can express partial membership. For example, Tomorrow is a sort of holiday. The membership value obtained from membership function states the degree to which the input belongs to a fuzzy set. In the fuzzy model, one needs to define two membership functions; Input features membership function and output feature membership function.
Note: Membership functions for the input and output can be same or different. It is concretely defined by the nature of values of the features.
We state three membership functions that we have used in our work for modeling the input features and target features.

3.1 Triangular Membership Function

It is a membership function that is defined by a lower limit x, an upper limit y and a value y, where y ϵ (x,z).

$$trimf(I; [x, y, z]) = max(0, min(\frac{I - x}{y - x}, \frac{z - I}{z - y})) \tag{3}$$

In this assignment, we are using sklearn and python based fuzzy logic library called skfuzzy. The triangular membership function is defined in this library as follows :

$$input\_variable = numpy.arange(0, 11, 1) \tag{4}$$

$$output\_variable = fuzz.trimf(input\_variable, [0, 0, 5, 5]) \tag{5}$$

Using the programmatic statement in 4 and 5, we create a triangular membership function which takes the user defined input and a predefined range (values of x,y,z) and generate membership function values in the range (0,1). Using the statements 4 and 5, the fuzzy values defined in the *output_variable* is

array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 0.8, 0.6, 0.4, 0.2, 0.])

It's corresponding membership function plot is showing in figure 1.

3.2 Trapezoidal Membership Function

It is a membership function that is defined by a lower limit w, an upper limit z, a lower support limit x and an upper support limit y, where $w < x < y < z$. There are two special cases of the trapezoidal function :

1. L-Function : A trapezoidal membership function is termed as L-function when $y = z = +\infty$.

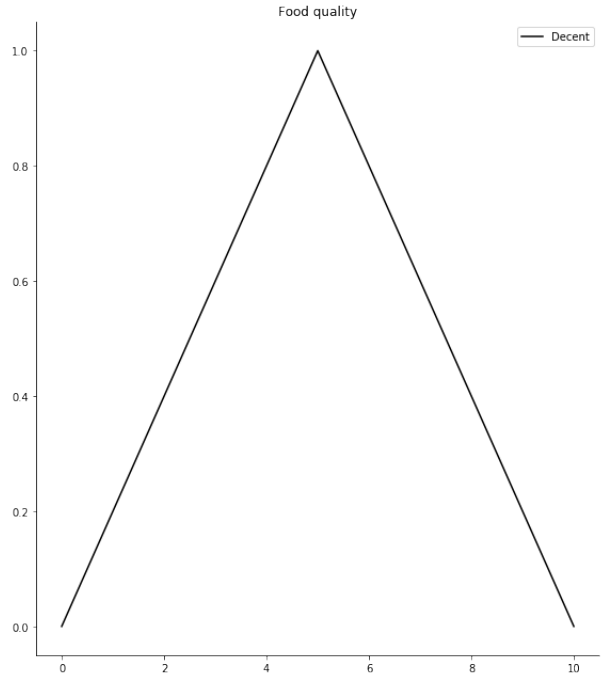


Figure 1: Example plot of triangular membership function.

2. R-Function : A trapezoidal membership function is termed as R-function when $w = x = -\infty$.

$$output\_variable = fuzz.trapmf(input\_variable, [0, 2, 8, 10]) \tag{6}$$

Using the programmatic statement in 4 and 6, we create a trapezoidal membership function which takes the user defined input and a predefined range (values of x,y,z) and generate membership function values in the range (0,1). Using the statements 4 and 6, the fuzzy values defined in the *output_variable* is

array([0. , 0.5, 1. , 1. , 1. , 1. , 1. , 1. , 1. , 0.5, 0.])

It's corresponding membership function plot is showing in figure 2.

$$trapmf(I; [w, x, y, z] = max(0, min(1, \frac{I - w}{x - w}, \frac{z - I}{z - y})) \tag{7}$$

3.3 Gaussian Membership Function

It is a symmetric membership function that depends on two parameters; σ and c. These two parameters define the shape of the Gaussian membership function. σ is the called the standard deviation of the curve and the smaller its value the narrower is the curve (always

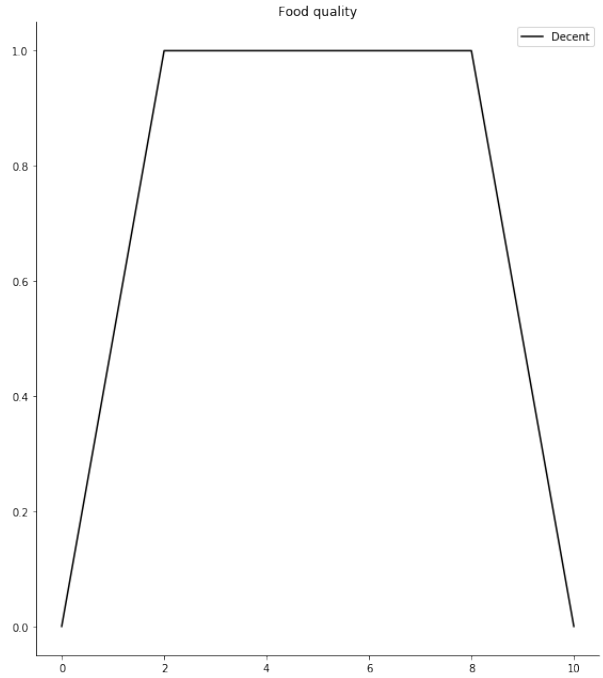


Figure 2: Example plot of trapezoidal membership function.

$\sigma > 0$). Another parameter c is the *central value* of the curve. Central Value is defined as the point of infection of the curve mapped onto x-axis.

$$gaussmf(I; \sigma, c) = e^{\frac{-(I-c)^2}{2\sigma^2}} \tag{8}$$

$$output\_variable = fuzz.gaussmf(input\_variable, numpy.mean(input\_variable), numpy.std(input\_variable)) \tag{9}$$

Using the programmatic statement in 4 and 9, we create a gaussian membership function which takes the user defined input and a predefined range (values of x,y,z) and generate membership function values in the range (0,1). Using the statements 4 and 9, the fuzzy values defined in the *output_variable* is

array([0.2865048 , 0.44932896, 0.63762815, 0.81873075, 0.95122942, 1. , 0.95122942,
0.81873075, 0.63762815, 0.44932896, 0.2865048])

It's corresponding membership function plot is showing in figure 3.

4 Fuzzy Rules

In this section, we state and define the fuzzy rules that have been created for this assignment. There are various ways of generating the fuzzy rules, we have used Max-Min composition function. The intuition behind using Max-Min composition is two folds :

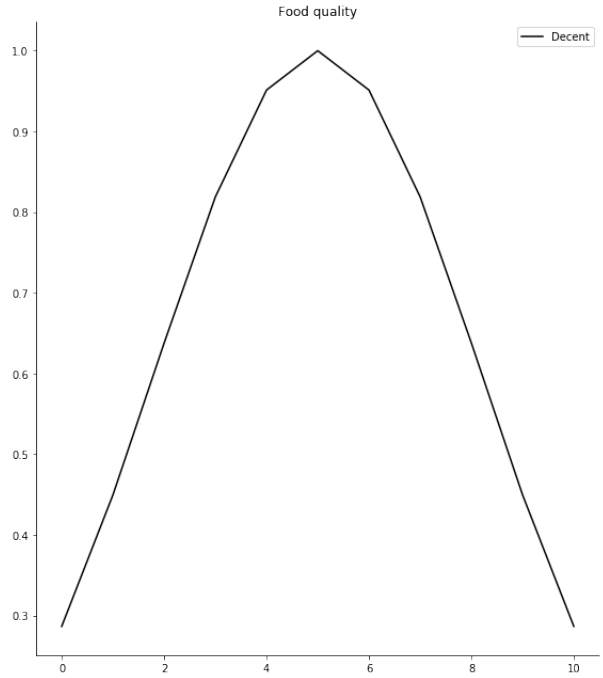


Figure 3: Example plot of Gaussian membership function.

1. Weights of the features according to their importance is unknown. Hence we use their membership function values as input to the Max function.

2. Features values generated from surveys using Likert scale barely obey some curvature or linear function. **Hence, Min function preserves the re-presentable value**.

As stated, we are using the python based sk-fuzzy library for fuzzy logic modeling, we define the methodology of creating the fuzzy rules as follows : **Note** : *vacc* represent vaccine, *hwq* represent hand wash quality, *sick* represent target variable.

1. vacc = fuzzy.interp_membership(vacc, mf(vacc), inputA): this is the definition of interpolated membership function. It takes the feature values as first input, membership function values as second input and *inputA* as a third input value (not an array) that does not belong to the feature values in first input.

2. hwq = fuzzy.interp_membership(hwq, mf(hwq), inputB): this definition is created for another feature. Since this is a 2 variable fuzzy logic model showing the fuzzy relationship between Hand Wash Quality and Vaccine on Sickness.

3. rule = numpy.fmax(vacc, hwq): This rule finds the maximum value out of vaccine and hand wash quality fuzzy values.

4. sick_activn = numpy.fmin(rule, sick_val): The maximum value identified in the previous step is compared with actual sickness value (represented as sick_val) and find the minimum. This statement completes the fuzzy rule creation step a pair of inputs: inputA and inputB. Similarly, fuzzy rules can be created for multiple input pairs.

5 Implementation Details

There are two prominent implementations of Fuzzy logic: MATLAB and Sci-kit Fuzzy. We have used Sci-kit fuzzy library of Python for our implementation. Our intuition of using this library is to learn and understand, how membership functions are defined, how these membership functions are formulated into rules and finally, how de-fuzzification is performed using centroid function. Apart from using the sci-kit fuzzy library, we used Matplotlib for plotting the membership functions. Also, we tested our rules MATLAB environment using MATLAB built-in Fuzzy Logic Toolbox.

6 Results and Analysis

In this section, we will exhibit the results of our experiments that have been carried out using two environments viz. MATLAB Fuzzy Logic Toolbox and Python Scikit learn Fuzzy Library. Fuzzy logic toolbox allows the user to use their GUI for creating the membership functions, fuzzy rules and perform defuzzification for evaluating the fuzzy inference system. Scikit-Fuzzy library allows you to create pythonic programs that using library-builtin membership functions, fuzzy rules using numpy Min-Max composition and either use built-in or user-defined centroid functions for defuzzification. We henceforth, partition this section into two folds; Sk-Fuzzy and MATLAB Fuzzy.

6.1 MATLAB Fuzzy

The Fuzzy logic toolbox created in MATLAB has been used by a majority of researchers working in the domain of computational intelligence. The three major components of this toolbox are; Membership function editor, Rules editor and Rules Visualizer. Membership function editor is a GUI interface where the user states the type of membership function, define the range of the inputs and outputs (as shown in 4). Access to the membership editor and other GUI components is granted from Fuzzy Logic Designer (as shown in 5).

After stating the membership functions for the input and output features, we define the rules of these features using Rules editor (as shown in 6).

After stating the rules, we visualize the rules and evaluate the performance of the Fuzzy Inference System (FIS) using the rules visualizer (as shown in 8). In addition to this, one can view the influence (trend) of the input variable over the output variable. From figure 7, we observe that feature values of *Vaccine* can create vibration in the range (0, 0.999) for the output variable whereas *HndWshQual* can create vibration in the range (0, 2). This

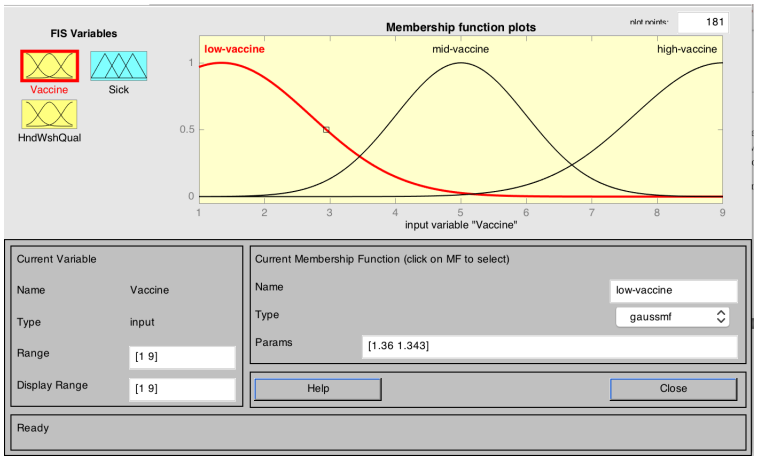


Figure 4: Membership function editor GUI in MATLAB fuzzy logic toolbox

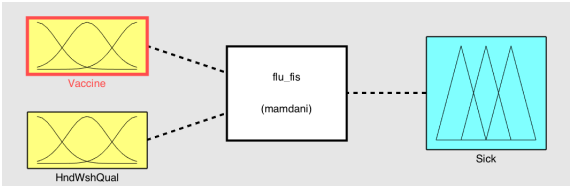
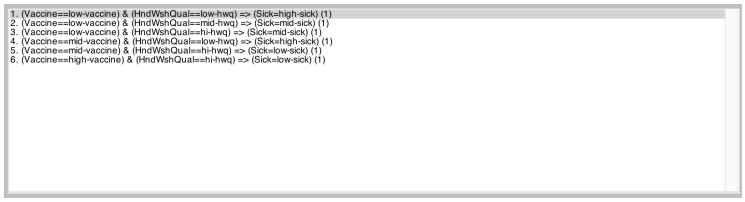
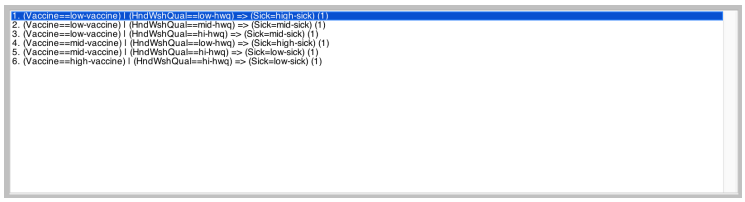


Figure 5: Fuzzy logic Designer GUI showing the Mamdani fuzzy model in MATLAB

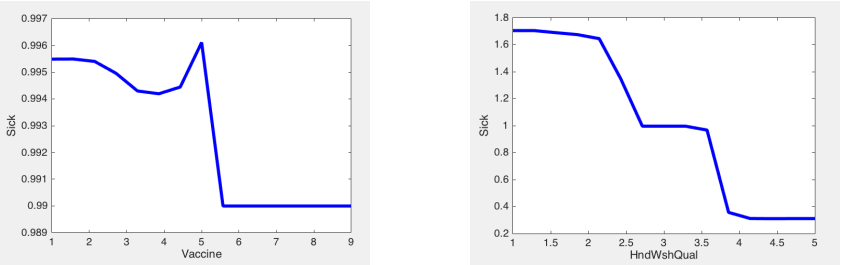


(a) Conjunctive rules (AND)



(b) Disjunctive Rules (OR)

Figure 6: MATLAB Fuzzy Logic GUI for rules: Rule Editor.



(a) A (b) B

Figure 7: (A). The trend in the Sickness with changing Human behavior towards Vaccine Intake. The decline in non-monotonous and it does not really affect the Sickness value. (B) The trend in the Sickness with changing Human behavior toward hygiene by adherence to quality standards for Hand Washing. The curve is monotonous and its decline largely affects the Sickness value.

creates an assumption that Hand Wash Quality (HndWshQual) is more influential input variable over Vaccine for the classification of Sickness (Sick).

Using the Fuzzy Logic toolbox we have carried out series of experiment and evaluated the performance using differential error.

6.1.1 Changing the membership functions of dataset features

In this section, we assess the impact on FIS when we change the membership function of input and output features.

Triangular Input Features and Output Feature In this experiment, we created the FIS making triangular membership function for the input (*Vaccine, Hand Wash Quality (HWQ)*) and output features (*Sick*). As shown in table 3, Triangular membership function shown an error of 0.420 which is better than trapezoidal (shown later) and poor than Gaussian (shown later).

Trapezoidal Input Feature and Triangular Output Feature In this experiment, we changed the membership function for Vaccine and Hand Wash Quality (input feature) and kept intact the membership function of Sick (output feature). The reason for changing the membership function for input and output features is to examine which function best mimic the characteristic properties dataset features. We observed that there is an increase in the differential error when both input features were modeled using the trapezoidal membership function (tables 3 and 4). This is due to lack of tuning of parameters (this membership function has more parameters than triangular and Gaussian) and other aspects such as defuzzification method, AND-OR rules. In order to affirm our claim, we subsequently carry out further experiments keeping Trapezoidal input and Triangular output membership functions.

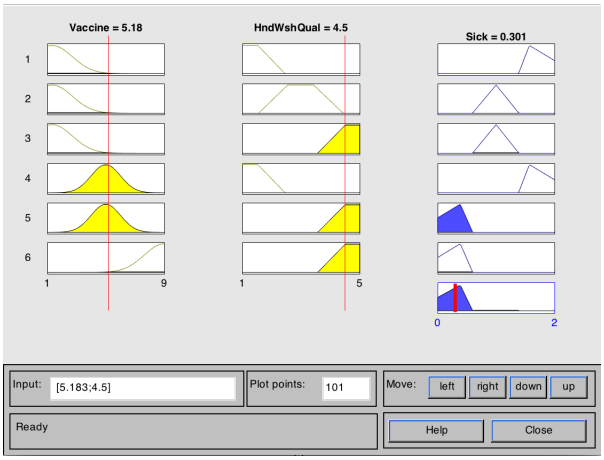


Figure 8: Rules Visualizer and FIS evaluation in the Toolbox

Vaccine	HWQ	Sick	Actual-Value	Error
3	4	0.828	1	0.172
4	4	0.502	1	0.498
5	5	0.275	1	0.725
9	4	0.271	0	0.271
			Avg. Error	0.420
	Test Data			
2.8	4.6	0.919		
6.4	4.5	0.274		

Table 3: Triangular Membership Function for input and output features

Vaccine	HWQ	Sick	Actual-Value	Error
3	4	0.995	1	0.005
4	4	0.272	1	0.728
5	5	0.275	1	0.725
9	4	0.271	0	0.271
			Avg. Error	0.432
	Test Data			
2.8	4.6	0.996		
6.4	4.5	0.277		

Table 4: Trapezoidal Membership Function for input feature and Triangular Membership Function for output feature

Gaussian, Trapezoidal, and Triangular Membership Function In this experiment, we used all the three membership function to assess the performance of FIS. We made

Vaccine of Gaussian type, *HWQ* of Trapezoidal type and *Sick* of Triangular type. By doing so, we observed a decrease of 3% in the differential error when compared with the results shown in table 4. The results of this experiment have been stated in table 5.

Vaccine	HWQ	Sick	Actual-Value	Error
3	4	0.837	1	0.163
4	4	0.489	1	0.511
5	5	0.311	1	0.689
9	4	0.271	0	0.271
			Avg. Error	0.410
	Test Data			
2.8	4.6	0.896		
6.4	4.5	0.278		

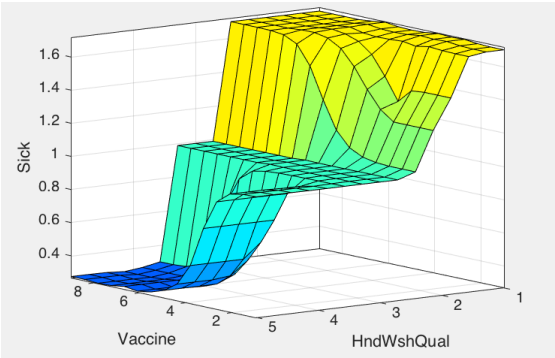
Table 5: Trapezoidal Membership Function for HWQ, Gaussian Membership Function for Vaccine and Triangular Membership Function for output feature

6.1.2 Changing AND to OR in the rules connection

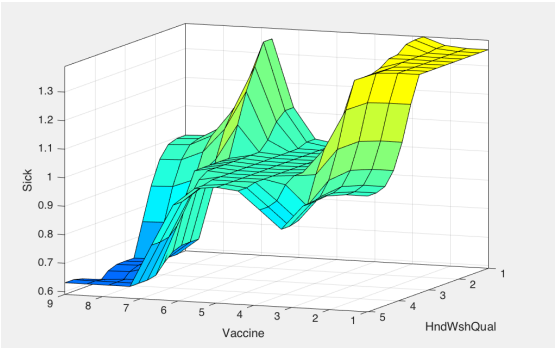
In this experiment, we changed the connection between two input feature from *AND* to *OR*. In other words, we moved from conjunctive to disjunctive rules when *Vaccine and HWQ* are modeled using Trapezoidal Membership Function. As a result of this, there is a change in the surface plot (3-D plot) of the rules created (as shown in 9). The transformation from AND connectivity to OR connectivity has significantly improved the results by decreasing the error by 6.5% in comparison to table 4.

Vaccine	HWQ	Sick	Actual-Value	Error
3	4	1.01	1	0.01
4	4	1.00	1	0.00
5	5	0.996	1	0.004
9	4	0.571	0	0.571
			Avg. Error	0.150
	Test Data			
2.8	4.6	0.939		
6.4	4.5	0.873		

Table 6: Assessing improvement in Trapezoidal Membership function when compared with table 4 after changing the rules connection from AND to OR.



(a) A



(b) B

Figure 9: Surface Plot of Rules created using Fuzzy logic Toolbox in MATLAB. (A). Surface plot with conjunctive rules. (B). Surface plot with disjunctive rules.
Note: Input features are trapezoidal and output feature is triangular

6.1.3 Changing Defuzzification Method

We further enhance the performance of FIS for trapezoidal function by changing the defuzzification function from *centroid* to *bisector*. Defuzzification is a procedure intrinsic to a fuzzy logic model which convert the values from fuzzy logic to crisp logic (0/1). There are different types of defuzzification methods in Matlab such as Centroid, Bisector, Medium of Maximum (MOM), Small of Maximum (SOM), etc. which can be used depending on the problem that is modeled using the toolbox. In this experiment, when we changed the defuzzification method from centroid to bisector, we observe a decrease of 2% in-comparison to table 6 and 72% in comparison to table 4.

Vaccine	HWQ	Sick	Actual-Value	Error
3	4	1.02	1	0.02
4	4	1.00	1	0.00
5	5	1.00	1	0.00
9	4	0.44	0	0.44
			Avg. Error	0.115∼0.12
	Test Data			
2.8	4.6	0.94		
6.4	4.5	0.90		

Table 7: Assessing improvement in Trapezoidal Membership function when compared with table 4after changing the defuzzification function from Centroid to Bisector

Hence, it is experimentally proved that parameter tuning, AND-OR conversion, and changing defuzzification method improves the performance of the trapezoidal (will work for any other membership function) FIS system.

6.2 Scikit-Fuzzy

In this section, we exhibit the results derived using the pythonic library: Scikit-Fuzzy. The experiments carried out using this library are defined under three categories: Triangular membership function, Trapezoidal membership function, and Gaussian Membership function. We detail the analysis figuratively.

	Vaccine	HWQ	Sick(X-axis)	Actual-Value	Error
Trimf	3	4	0.24	1	0.76
Trapmf	3	4	0.72	1	0.28
Gaussmf (with split)	3	4	0.12	1	0.88
Gaussmf (without split)	3	4	0.17	1	0.83

Table 8: Error recorded on a single data tuple input to the FIS created using Sk-fuzzy

6.2.1 Triangular Membership Function

In this experiment, similar to FIS in MATLAB, we generate fuzzy values of the input features (Vaccine and Hand Wash Quality) using triangular membership function (see figure 10) defined in section 3.1. The colored zig-zag in the figure is because of the input *Vaccine* feature values from dataset. The same reason is attributed for *Sick* and *HndwshQual*. We first generate a fuzzy space based on the domain and range of the input feature values. Then map the input feature values onto the fuzzy space using the fuzzy rules. Finally, we defuzzify the outcome (transform to crisp logic) for real-world representation. From table 8, we observe that there is a high error rate in the FIS when we model the dataset features using Triangular membership function. But, triangular membership function performed better than Gaussian membership function.

6.2.2 Trapezoidal Membership Function

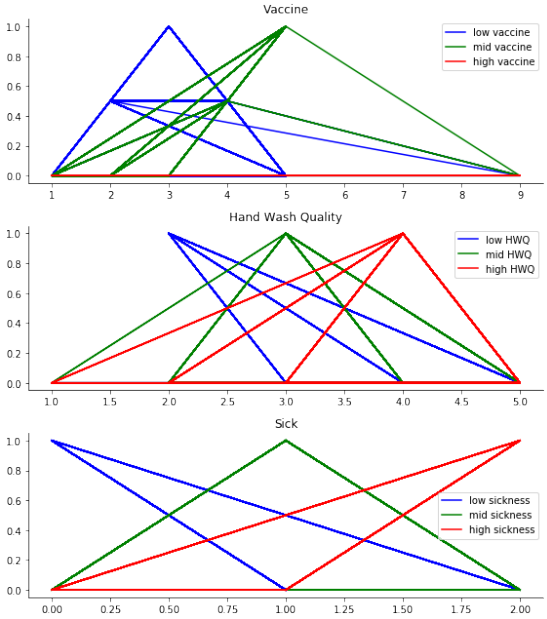
When we modeled the data features using the Trapezoidal membership function (see figure 11), we achieved a minimal error of 0.28 in comparison with Triangular and Gaussian functions (shown in table 8). This completely aligns with the result shown by FIS created in MATLAB. The reason for Trapezoidal function outperforming Triangular and Gaussian is that trapezoidal function is shift-invariant, scale-invariant and union-invariant [1].

6.2.3 Gaussian Membership Function

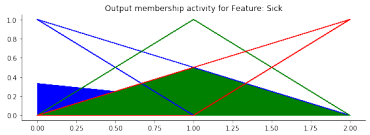
In this section, we describe the results obtained when the Gaussian function transforms the data features into Fuzzy space. Since, input to the Gaussian function is the mean and standard deviation of the data feature values (there is no upper or lower limit, like in triangular or trapezoidal), we perform two types of test:

Gaussian Membership function over split feature values : First, we split the data feature values into three categories; Low ("lo"), Medium ("md") and High ("hi"). Then we generate fuzzy membership function values for each categories. This can be seen in figure 12, with three bell shaped curves.

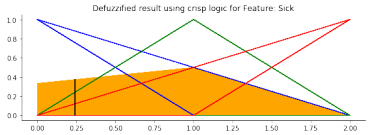
Gaussian Membership function over non-split feature values : we do not split the data feature values into these three categories. One can identify the difference between the figures 12 and 13 as there no three curve being formed in the input features and there is a zig-zag. This zig-zag is because Vaccine and HndWshQual has different domain and range. Moreover, the we send as input the complete set of feature values for both the in-variables. As a result of this experiment, we found that there was no major difference in the result (see table 8) but there was a gap between the membership function values (split feature values: 0.94, non-split feature values: 0.44). This result can visualized in part C of both the figures 12 and 13. In both the scenarios, the output feature (which is triangular membership function) is always splitted into low, medium and high categories.



(a) A

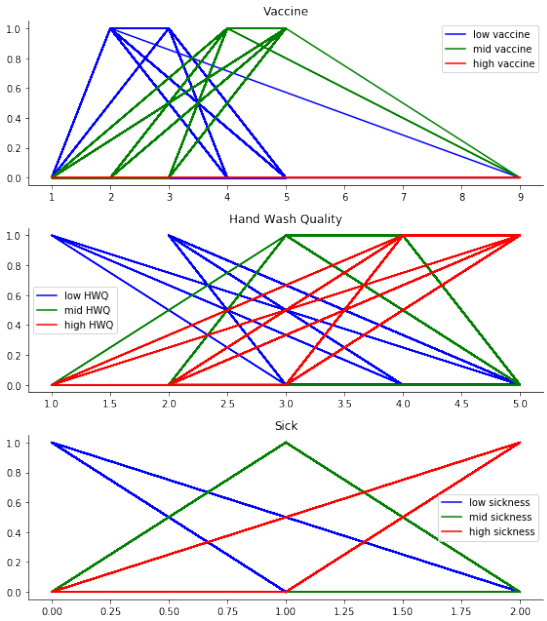


(b) B

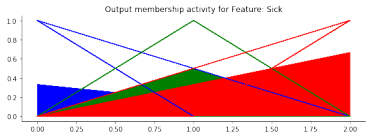


(c) C

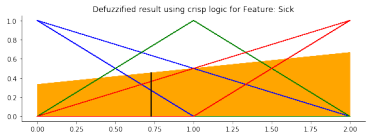
Figure 10: (A). Mapping of input and output data feature values onto Fuzzy space using Triangular Membership Function. (B) Output generated as a result of Fuzzy rules. The shaded region is the desired region (intersection of input and output features), similar to FIS in MATLAB figure 8 (C). Outcome after defuzzification. The black line is the result generated after giving following input to the FIS : Vaccine = 3 and Hand Wash Quality = 4



(a) A

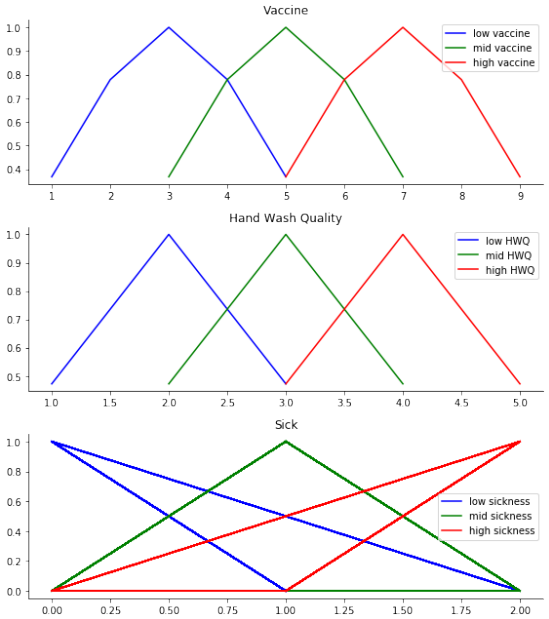


(b) B

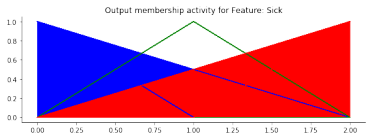


(c) C

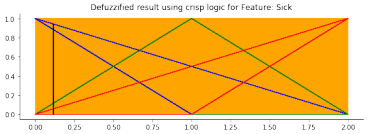
Figure 11: (A). Mapping of input and output data feature values onto Fuzzy space using Trapezoidal Membership Function. (B) Output generated as a result of Fuzzy rules. The shaded region is the desired region (intersection of input and output features), similar to FIS in MATLAB figure 8 (C). Outcome after defuzzification. The black line is the result.



(a) A

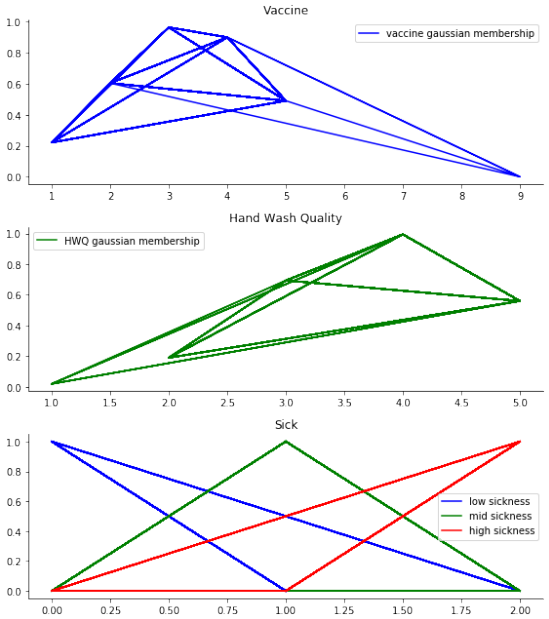


(b) B

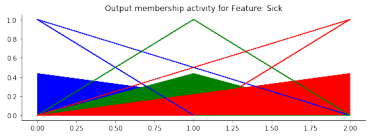


(c) C

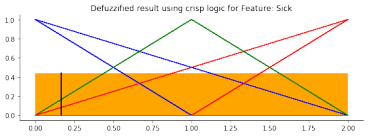
Figure 12: (A). Mapping of input and output data feature values onto Fuzzy space using Gaussian Membership Function with splitting the dataset features into three categories: Low, Medium and High. (B) Output generated as a result of Fuzzy rules. The shaded region is the desired region (intersection of input and output features), similar to FIS in MATLAB figure 8 (C). Outcome after defuzzification. The black line is the result.



(a) A



(b) B



(c) C

Figure 13: (A). Mapping of input and output data feature values onto Fuzzy space using Gaussian Membership Function without splitting. (B) Output generated as a result of Fuzzy rules. The shaded region is the desired region (intersection of input and output features), similar to FIS in MATLAB figure 8 (C). Outcome after defuzzification. The black line is the result.

7 Reference

1. Barua, Aditi, Lalitha Snigdha Mudunuri, and Olga Kosheleva. "Why trapezoidal and triangular membership functions work so well: Towards a theoretical explanation." (2013)