Project 2:
Fuzzy Logic

Andy Varner
12153597
Fall 2022

Introduction / Project Description

The purpose of this project is to gain a meaningful introductory understanding of the topic of Fuzzy Logic. As a beginner to Machine Learning / Artificial Intelligence, I would like to develop an appreciation for how fuzzy inference systems (FIS) work, what is unique and advantageous about them, and what drawbacks are associated with them. I will be successful at this project if I can demonstrate a command of the theoretical foundations of the field, the ability to implement and tune a Mamdani fuzzy system for a classification task, and the ability to validate its performance. I would like to be able to answer the question, "Is this a good problem for a fuzzy system? Why or why not?" I will further challenge myself and solidify my understanding of fuzzy logic by performing logical operations using Zadeh's compositional rule of inference.

I begin by outlining the theoretical foundations of fuzzy systems in the background section. I describe how to establish linguistic rules to govern a fuzzy set and how to fuzzify input variables. Next, I discuss the logical inference process, followed by defuzzification. Once the basic system is established, I move to implementation. First, I describe the Mamdani FIS Implementation for the classification of the Iris dataset, and then I describe Zadeh's FIS Implementation for performing logical operations. The experiments and results section contains details about the Iris dataset and how I establish rules to classify it using a Mamdani system. After evaluating its performance, I describe my implementation and evaluation of Zadeh's system for the application of logical operations. I end this report with conclusions and a discussion of future work.

Background: Fuzzy Systems

For our discussion of fuzzy systems, we are interested in data, information, and uncertainty. We desire computational systems that optimize as they reason, so that we do not implement something that must later be undone. This is David Marr's "principle of a least commitment" [1]. Achieving this is difficult if the system makes early decisions based on crisp sets which have hard boundaries. Fuzzy logic offers a theoretical framework for softening the decision-making process to make room for things like uncertainty, vagueness, or incompleteness of information.

Because fuzzy logic is an extension of Boolean logic, we offer an overview of Boolean logic here. Boolean logic is a two-valued logical system that assigns total membership to a class or not such that class membership is crisp. We can consider belonging to have a designation of 1 and exclusion to have a designation of 0. While technological successes have been achieved with a Boolean logic foundation, we would like a framework for processing incomplete, imprecise, or vague information. Take, for example, the classification problem of Figure 1. With Boolean logic, only absolute belonging to and absolute exclusion from a class are possible, so Andy either belongs to the class "Smart" or not. In this system, it is impossible to be "kind of smart." Alternatively, Fuzzy logic allows for a *degree* of membership to a class. This extends Boolean logic and its accompanying mathematics to handle partial truth and the vagueness of natural language.

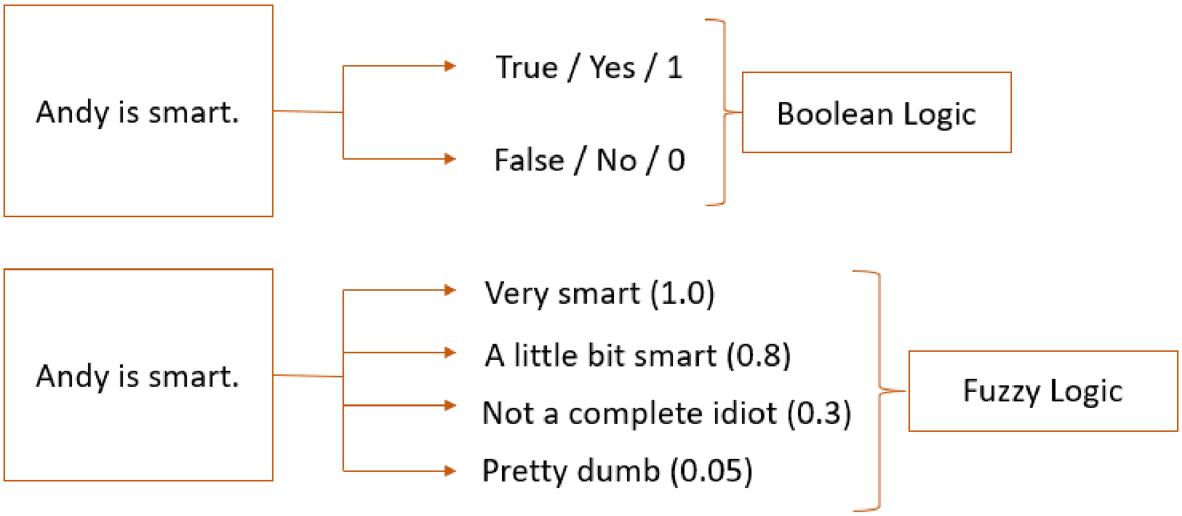

Figure 1. Boolean logic vs. Fuzzy logic

Developed from the work of Aristotle, Boolean logic allows us to solve a problem by mapping the problem domain to two-valued variables. Plato gave us the foundation of what we now call fuzzy logic, but it was not formalized until the 1900s with the work of Lucasiewicz. Lucasiewicz extended two-valued, true/false logic to three-valued logic, which assigns a value between true and false. Eventually he extended this to four- and five-valued logic, enabling the fuzzy logic characterization of Figure 1. In 1965, Lofti Zadeh further extended and formalized multi-valued logic systems with the mathematics of fuzzy set theory [2]. Fuzzy set theory gives us infinite-valued logic, greatly extending the number of problems that can be solved in control, information systems, decision-making, and a variety of other machine learning tasks.

A Fuzzy Logic System

We have established the purpose of the fuzzy logic system: to soften crisp sets into fuzzy sets so that their membership to a particular class may be evaluated considering uncertainty. In the following sections, we outline the general method that allows us to achieve this, guided by the diagram in Figure 2.

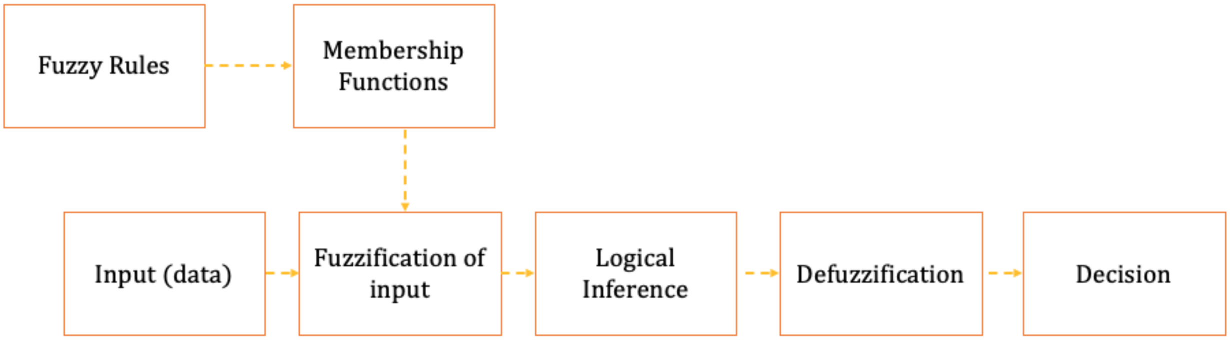

Figure 2. Fuzzy inference system overview

I. Establish linguistic (fuzzy) rules

We anchor our description of the fuzzy logic system to the well-known tipping problem outlined in [3]. The linguistic rules are as follows:

- Rule 1: If the service is poor or the food is bad, then tip is cheap.
- Rule 2: If the service is good, then tip is average.
- Rule 3: If the service is excellent or food is delicious, then tip is generous.

Words like "poor," "good", and "excellent," as well as "cheap," "average," and "generous," are subjective terms which have uncertainty and bias built into them. In a fuzzy logic system, these words are called *linguistic values*. When combined with *linguistic variables*, they form propositions and implications, as shown in Figure 3. Antecedents, which become propositions, and implications make up our fuzzy rules. The linguistic variable/value pairs, "service is poor" or "food is bad" constitute the proposition (A') of the rule. This is a disjunctive rule, because it uses "or." It has the implication, "tip is cheap," where the linguistic value, "cheap" is the consequence. We determine membership to a class via membership functions, which are discussed in the next section, fuzzification of input variables.

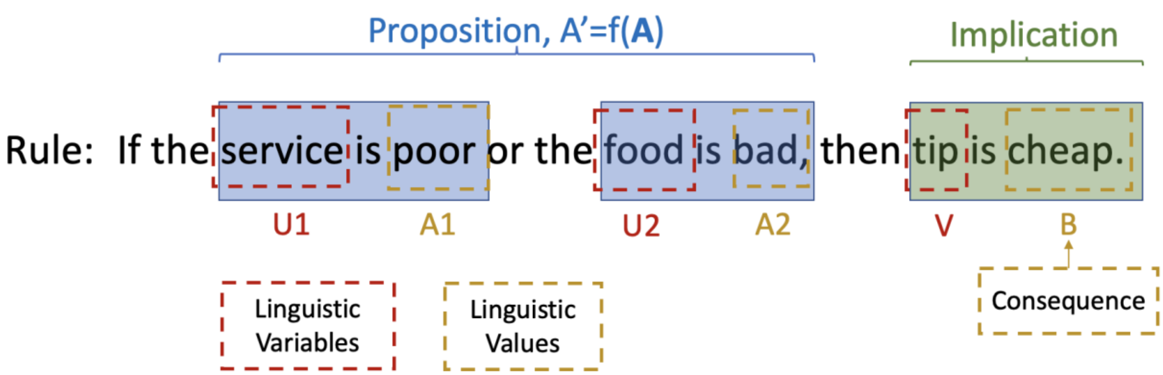

Figure 3. Fuzzy Rule breakdown

II. Fuzzification of input variables

Fuzzification is the process of transforming a crisp set into a fuzzy set. The fuzzy logic system takes in a set of inputs, which could be sensor readings, simulation data, expert knowledge, etc., and performs fuzzification on the inputs. For our tipping problem, the inputs are ratings on two domains: (1) Service and (2) Quality of Food. Let the data points on the graphs in Figure 4 represent crisp data points, or ratings, assigned by an "expert" (in this case, a restaurant patron) to be used as inputs to the system. These inputs are evaluated according to our rules via membership functions.

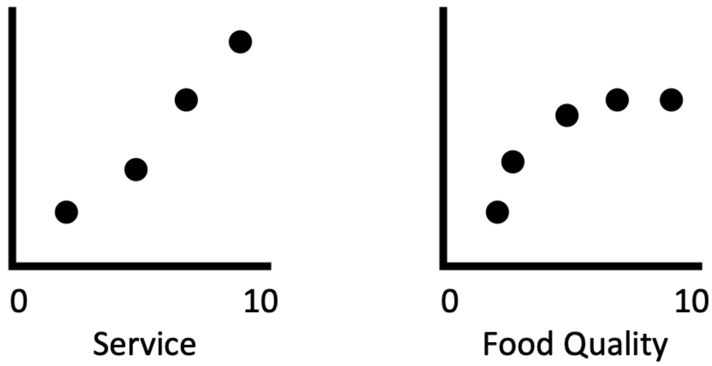

Figure 4. Crisp input data for fuzzy logic system on two domains

Membership functions are created from the fuzzy rules to determine the degree to which a given input belongs to a particular fuzzy set. For an observation, *X*, where,

$$X = \{x | x \in X\} | X \in \mathbf{R}$$

we can develop a membership function, A, to fuzzify $X \mid A(X) \rightarrow [0, 1]$. This membership function has the form,

$$A(X) = \{\mu(x) | x \in X\} | 0 \geq \mu(x) \leq 1$$

An example of a membership function is shown in Figure 5. On the domain *X* in the somewhat arbitrary range [0, 10], the subset $A \subset X$ is defined in the range [10, 50]. Thus, the membership function for set *A* for Figure 5(a) is as follows:

$$\mu_A(x) = 1 \quad for\ x \in [2,5]$$
$$\mu_A(x) = 0 \quad else$$

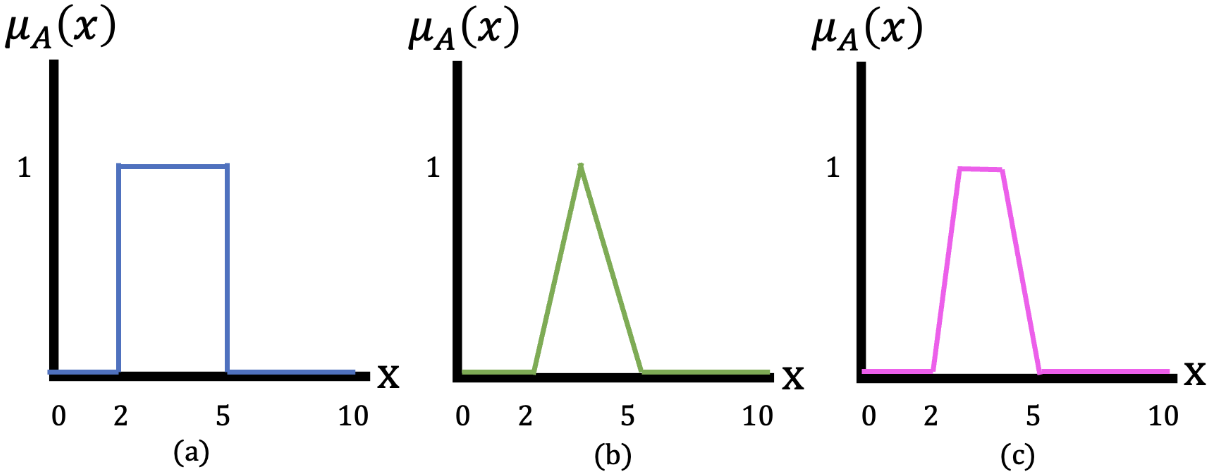

Figure 5. Membership function examples: (a) Step, (b) Triangular, (c) Trapezoidal

The shape of the membership function depends on the needs of the problem and is determined by the developer. A triangular, trapezoidal, or gaussian function may be better for characterizing the fuzzy set, depending on the application. The trapezoidal and gaussian function are more inclusive membership functions. They assign a membership of '1' to a greater number of observations than a triangular membership function, which is constrains complete membership to a limited observation.

Once a dataset has been fuzzified, we know the degree to which each proposition is satisfied for each rule. The fuzzy sets and rules we have established are used to make predictions. We discuss logical inference in the next section.

III. Logical Inference

If a proposition contains more than one antecedent, a fuzzy operator is used to obtain a single truth value from more than one membership function. Fuzzy operators commonly used for this task are Intersection (AND), Union (OR), and Complement (NOT). Intersection, $A_1 \wedge A_2$, is calculated as a T-norm, using minimum or product operators. Union, $A_1 \vee A_2$, is calculated as a T-conorm, using maximum or sum. Complement is calculated as $A^c = 1 - A$. Once these operators are applied, we have our fuzzy propositions defined.

The next task is Fuzzy Implication, in which we map propositions to their implications. There are multiple operators to choose from for defining fuzzy relations. Some commonly used fuzzy operators Lukasiewicz, Correlation Min, and Correlation Product, are defined in Table 1 for the fuzzy relation, *R*, for the proposition *A'* and implication *B*, where

$$R:\ A' \rightarrow B$$

Operator	Formula	Relation	Implication
Lucasiewicz	$R(\mu_A, \mu_B) = min(1,1 - \mu_{A'} + \mu_B)$	✓	✗
Correlation min	$R(\mu_A, \mu_B) = min(\mu_{A'} + \mu_B)$	✓	✓
Correlation product	$R(\mu_A, \mu_B) = \mu_{A'} * \mu_B$	✓	✓
Composition	$B' = A' \circ R = sup\ min\{A', R\},$ $= \vee \left(\wedge (A', R)\right)$	✗	✓

Table 1. Fuzzy relation and implication operators for FLS (Keller [1])

The result of inferencing is the mapping of fuzzified inputs to the rule base, producing a fuzzy output for each rule.

IV. Defuzzification

Once the fuzzy outputs have been determined, we convert those outputs to a scalar value; that is to say, we *defuzzify* the outputs of the fuzzy rules, reducing the number of outputs to a decision. Defuzzification methods include bisector of area, center of area, and fuzzy mean, among others. The implementation of this project uses centroid defuzzification, a method that returns the center of area under the aggregated fuzzy set, shown in Figure 6. The center of mass is calculated as follows:

$$y_c = \frac{\Sigma_{y \in Y} y * B'(y)}{\Sigma_{y \in Y} B'(y)}$$

With defuzzification complete, we have converted a fuzzy set to a crisp output, and a decision, i.e. a prediction or classification, can be made.

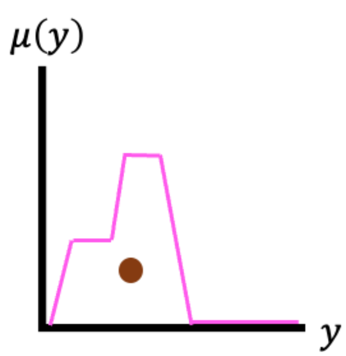

Figure 6. Visualization of center of area under an aggregated fuzzy set

Implementation

This section contains two parts: we develop a Mamdani FIS, and we develop Zadeh's discrete FIS for fuzzy-valued inputs.

The Mamdani system is implemented with support for any number of rules, multiple conjunctive antecedents, and both gaussian and trapezoidal membership functions. The system is single consequent, meaning our rules have single "then" outputs. The rules have different consequent domains and support both "max" and "sum" aggregation operators. For defuzzification, we use the centroid operator. After we validate the Mamdani fuzzy inference system, we use it to classify the Iris dataset.

Zadeh's FIS supports any number of rules, modes ponens, and all the implication methods outlined in Table 1. Like the Mamdani system, the Zadeh system supports up to three conjunctive antecedents, single consequent rules as well as rules with different consequent domains, and can handle both max and sum aggregation operators.

Mamdani FIS Implementation

The Mamdani fuzzy inference system uses correlation min (Table 1) as its fuzzy relation operator. The flowchart in Figure 7 describes the implementation of this system. The fuzzy rules are the handcrafted, explainable part of the fuzzy inference system, and are designed according to the features of the specific application, which in our case is the classification of the Iris dataset. We go into more detail on how this is achieved when we discuss the dataset in the Experiments and Results section. Membership functions are created based on the fuzzy rules to determine the degree to which an observation belongs to the fuzzy set defined in the rule. Membership functions are developed for each antecedent, and another set of membership functions is developed for consequences, that is, to determine the degree of belonging to a class. Once this is complete, we perform logical inference. We calculate propositions by applying the fuzzy operator to get an aggregation of rules. Then the Mamdani system uses correlation min to calculate the implications. The final part of this step is aggregation of all the outputs using correlation max. This project requires the centroid method for defuzzification, and finally, a decision is made based on the outcome of defuzzification.

The Experiments and Results section of this report goes into detail about how the Mamdani fuzzy inference system is used to classify the Iris dataset. We also validate the system based on known classes of the Iris dataset.

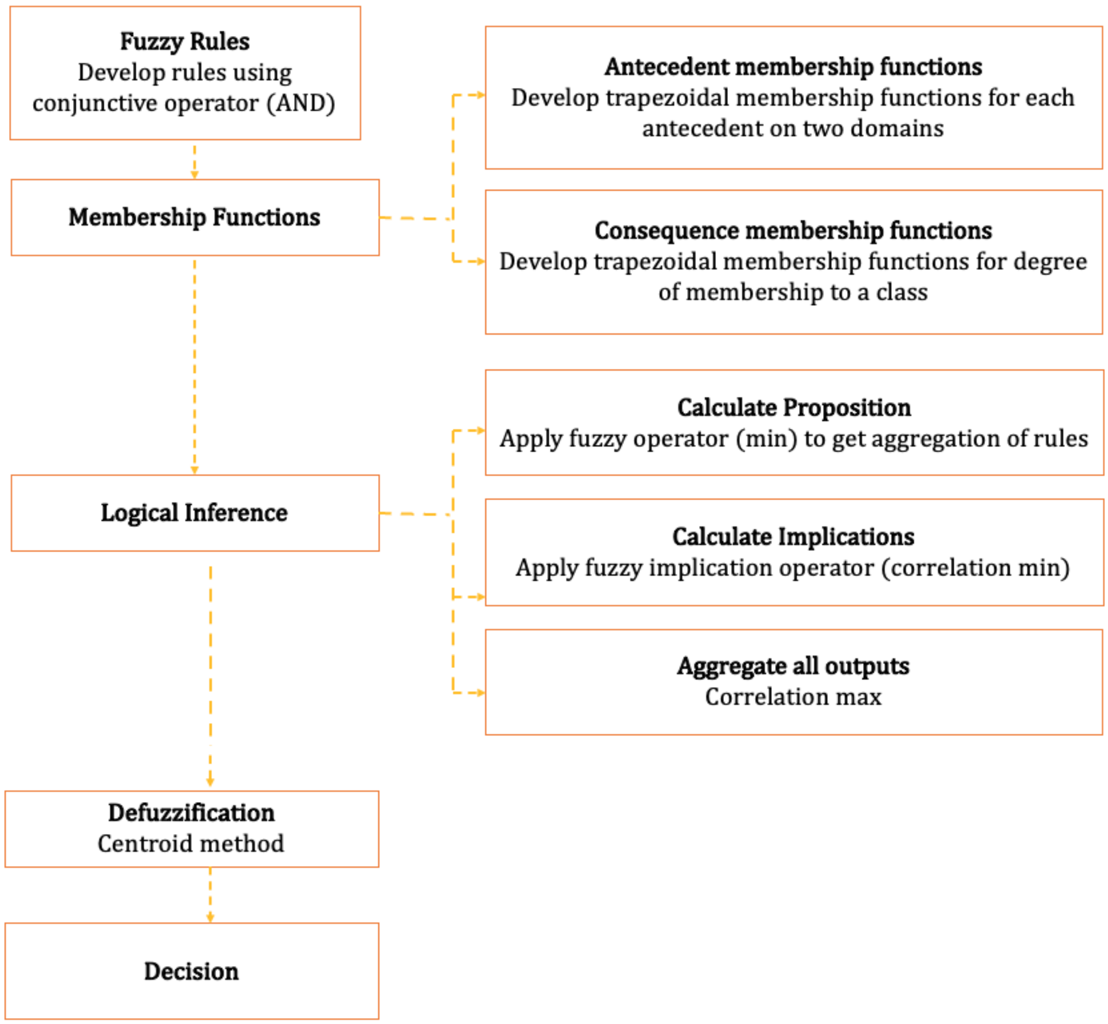

Figure 7. Mamdani Fuzzy Inference System

Zadeh's FIS Implementation

Zadeh's Fuzzy Inference system uses the compositional rule of inference:

Rule:	If U is A then V is B
Fact:	U is A'
Conclusion:	V is B'

The conclusion of this rule is the composition operation in Table 1. The *sup* is the supremum of the set, such that the least element that is greater than or equal to each element in the set. The *min* can be calculated using Lukasiewiz, correlation min, or correlation product. This project examines the implementation of Lukasiewiz and correlation min. For finite domains, the compositional rule of inference looks like matrix multiplication where min replaces multiplication and sup replaces summation [1]. The implementation of Zadeh's system requires the use of the composition operator. We will show how Zadeh's system facilitates logic using the Lukasiewicz operator, and then we will see how it behaves using correlation min, following the flowchart of Figure 8.

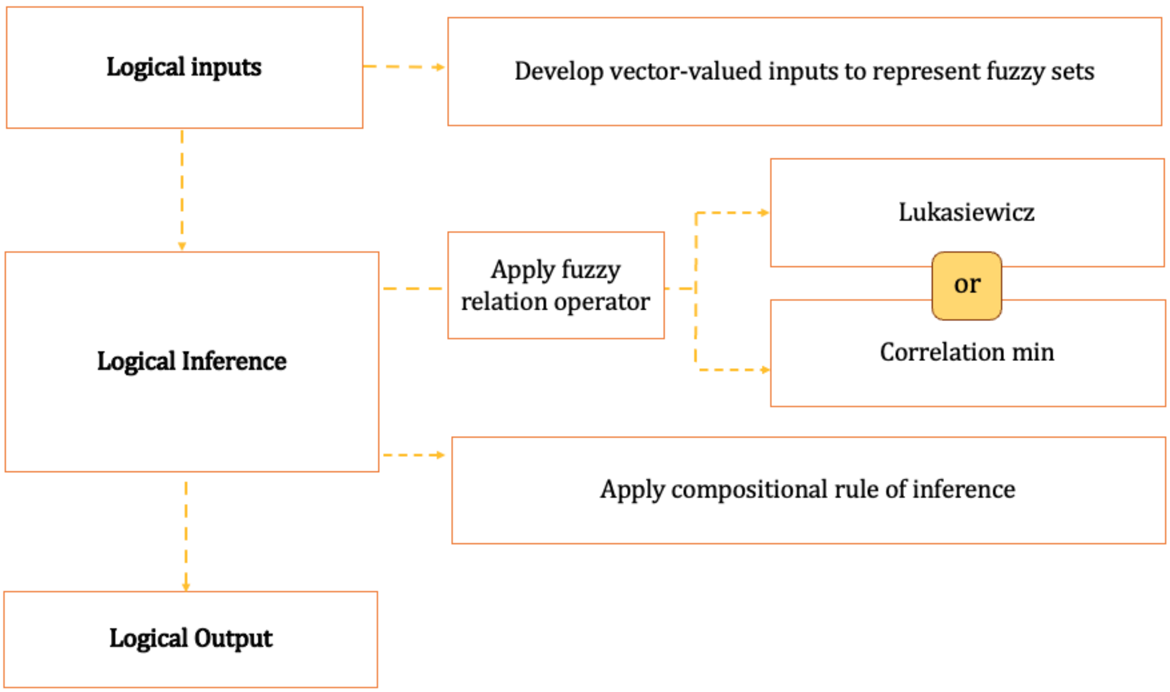

Figure 8. Flowchart for implementation of Zadeh's FLS

Experiments and Results

The experiments conducted for this report are designed to classify the Iris dataset [4]. We provided the skeletons of our inference systems in the Implementation section above, but we have not yet captured an important fact about fuzzy inference systems. Since fuzzy inference systems are handcrafted, explainable systems, their usefulness relies heavily on the efficacy of the rules which govern them. For this task, it is worthwhile to spend some time investigating the iris dataset to try to learn the features around which its classes are organized. This will help us to design rules that will capture the dataset and help us effectively build a fuzzy set for classification.

Rules for the Iris dataset

The Iris dataset consists of 150 instances of Iris, which are categorized into three classes, or species, with 50 instances of each in the dataset. The classes are Setosa, Versicolor, and Virginica, which we will refer to as Class 0, Class 1, and Class 2, respectively. These classes are organized by four features: sepal length, sepal width, petal length and petal width. Because we have four features, we have a four-dimensional problem, which is difficult to visualize. We can begin to visualize the dataset by making numerous two-dimensional plots which may give us a starting point for how best to separate the classes. Each of the plots in Figure 9 shows us a different view of how the classes are organized in two-dimensional space and provides a foundation on which we can begin to build some fuzzy rules.

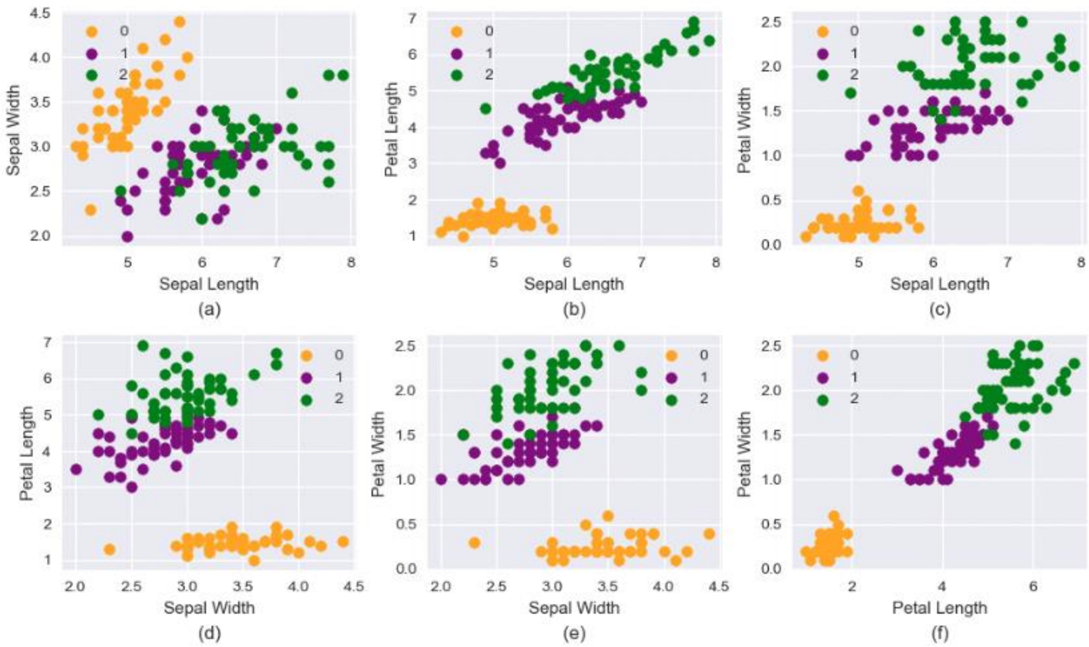

Figure 9. Two dimensional plots of the Iris dataset according to the four features we use for classification

It is clear from the plots in Figure 9 that Class 0 is completely separable and trivial to classify. From Figure 9(f), we have the tightest cluster of Class 0, which allows us to build Rule 1: If petal length is less than 2, the instance belongs to Class 0.

Classes 1 and 2 are a little bit trickier. The graph of Figure 9(a) is particularly unhelpful, with classes 1 and 2 sitting right on top of each other. The remaining graphs show *some* separability, but nothing unambiguous arises. We can plot petal width and petal length against the other two features in three-dimensional space, as in Figure 10, to visualize the classes from another perspective. From this view, we still see some class overlap in both (a) and (b), but we can still try to differentiate classes 1 and 2 from this information.

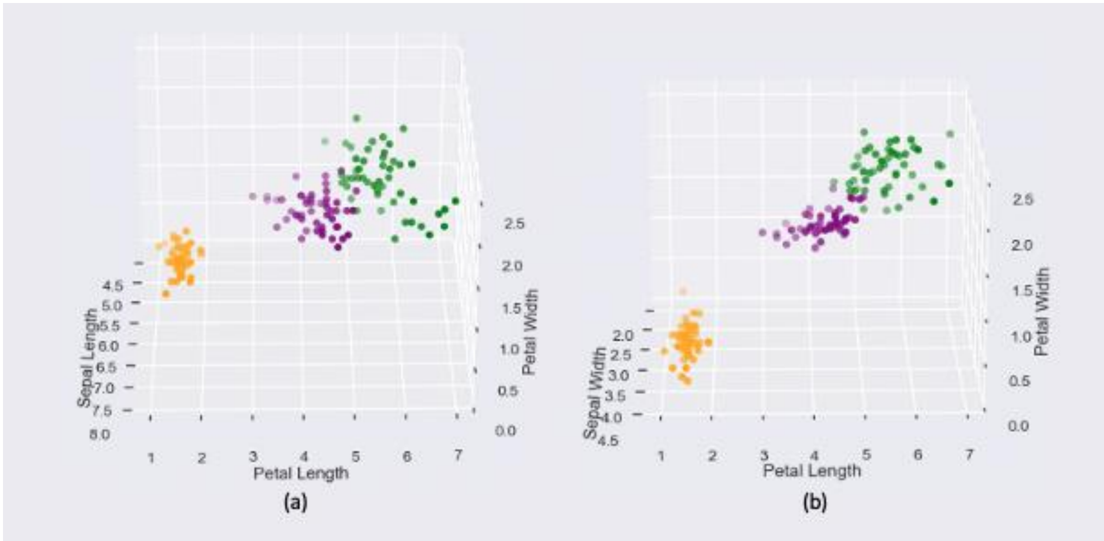

Figure 10. Three dimensional plots visualizing petal width and petal length against (a) sepal length and (b) sepal width

From this, we develop the next two rules: If petal length is medium and petal width is medium, then the instance belongs to Class 1. If petal length is large and petal width is large, then the instance belongs to Class 2. The set of fuzzy rules for classifying Iris in this project are summarized in Table 2.

Table 2. Fuzzy rules for categorizing Iris

1	If petal length is short, then class 0
2	If petal length is medium and petal width is medium, then Class 1
3	If petal length is large and petal width is large, then Class 2

Mamdani FLS for classification of Iris dataset

In this subsection, we proceed with developing rules for classification of Iris and describe how the implementation flowchart of Figure 7 plays out for the Mamdani system for classification.

Membership functions

As discussed in the Implementation section, we require two sets of membership functions. One to define our antecedents, and another to define our consequences, or implications. To develop our antecedent membership functions, we refer to Figure 3, which defines how we turn a fuzzy rule set into a set of propositions and implications. Let’s examine the logic of Figure 3 with our rule for Class 2 in mind. The diagram of Figure 11 shows how we obtain our proposition and implication for Rule 3 of Table 2.

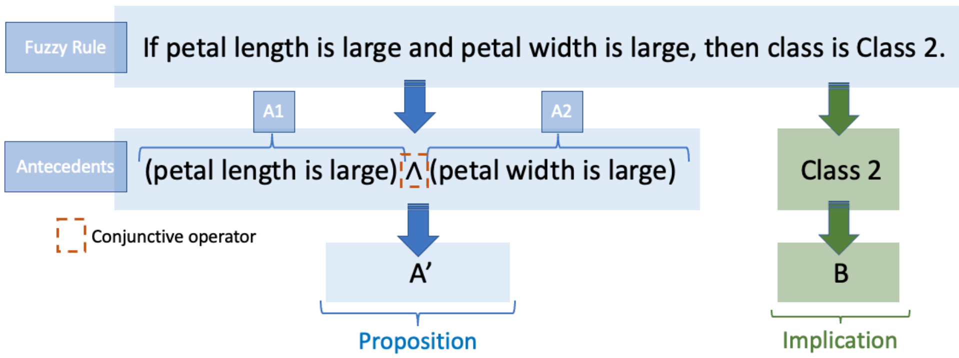

Figure 11. Flowchart for identifying the antecedents in Rule 3

Repeating the process outlined in Figure 11 for Rules 1 and 2, we find our rule set contains five total antecedents for which we will create five membership functions. This is a classification problem, so we do not want to use a step function or a triangular function, because these functions give us little room for degrees of membership. Instead, a gaussian or a trapezoidal membership function would be more appropriate. We choose trapezoidal functions for this experiment. The plotted antecedent membership functions are displayed in Figure 12. The plots follow the color-coding scheme for class membership already established in this paper (class 0 – orange, class 1 – purple, class 2 – green).

Our rules require us to build membership functions for more than one domain. Note that each membership function is either on the Petal Length or Petal Width domain. Our rules do not consider the Sepal Length or Sepal Width domains, but we have the potential to build antecedent membership functions for up to four domains for the Iris dataset.

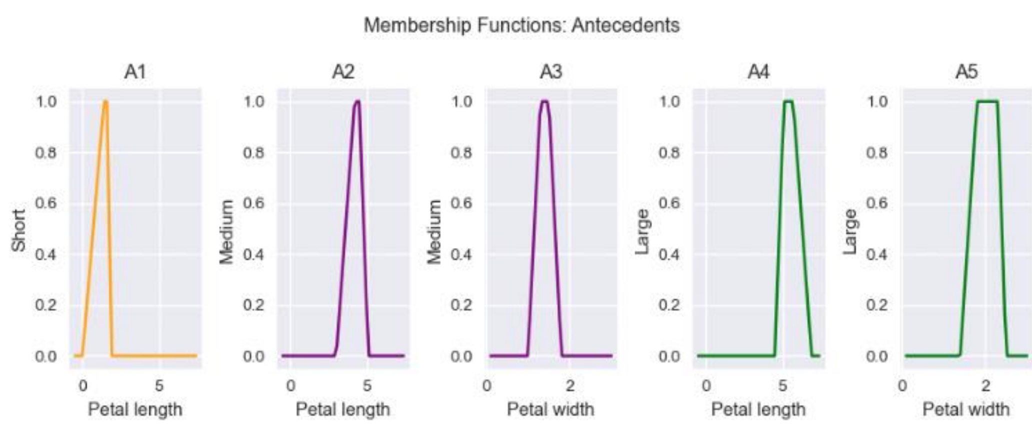

Figure 12. Antecedent membership functions for five antecedents pertinent to the fuzzy classification problem

Next, we develop membership functions for the consequences. The consequence, or implication, of Figure 11 is membership in class 2. The possible consequences for this classification problem are Class 0, Class 1, and Class 2. The consequence membership functions define degree of membership or confidence of belonging to a particular class. These membership functions all belong to the same domain, which is Class.

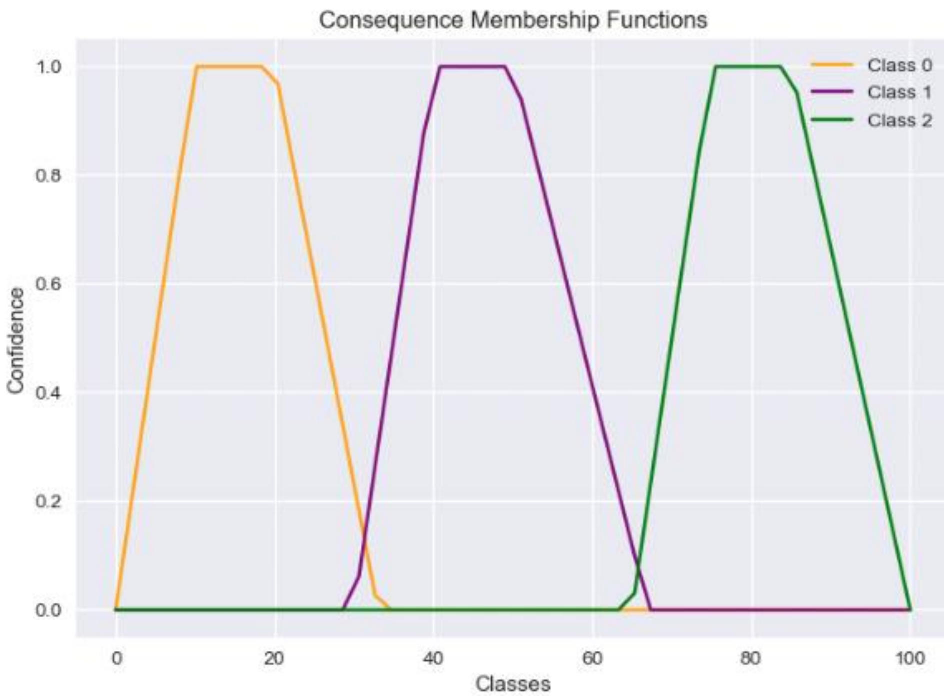

Figure 13. Consequence / Implication membership functions

We test our system by giving it inputs (observations) from the data set and letting the fuzzy system calculate the input's degree of membership to each class and making a decision based on the flowchart of Figure 7. Let's take index 80 as an individual test case. We know from the dataset that this observation has a truth label of 1, so it belongs to Class 1. First the Mamdani system assigns belonging to each of the antecedent membership functions, shown in Figure 14. This was an "easy" case, because this particular observation has zero membership to all the antecedents associated with Classes 0 and 2.

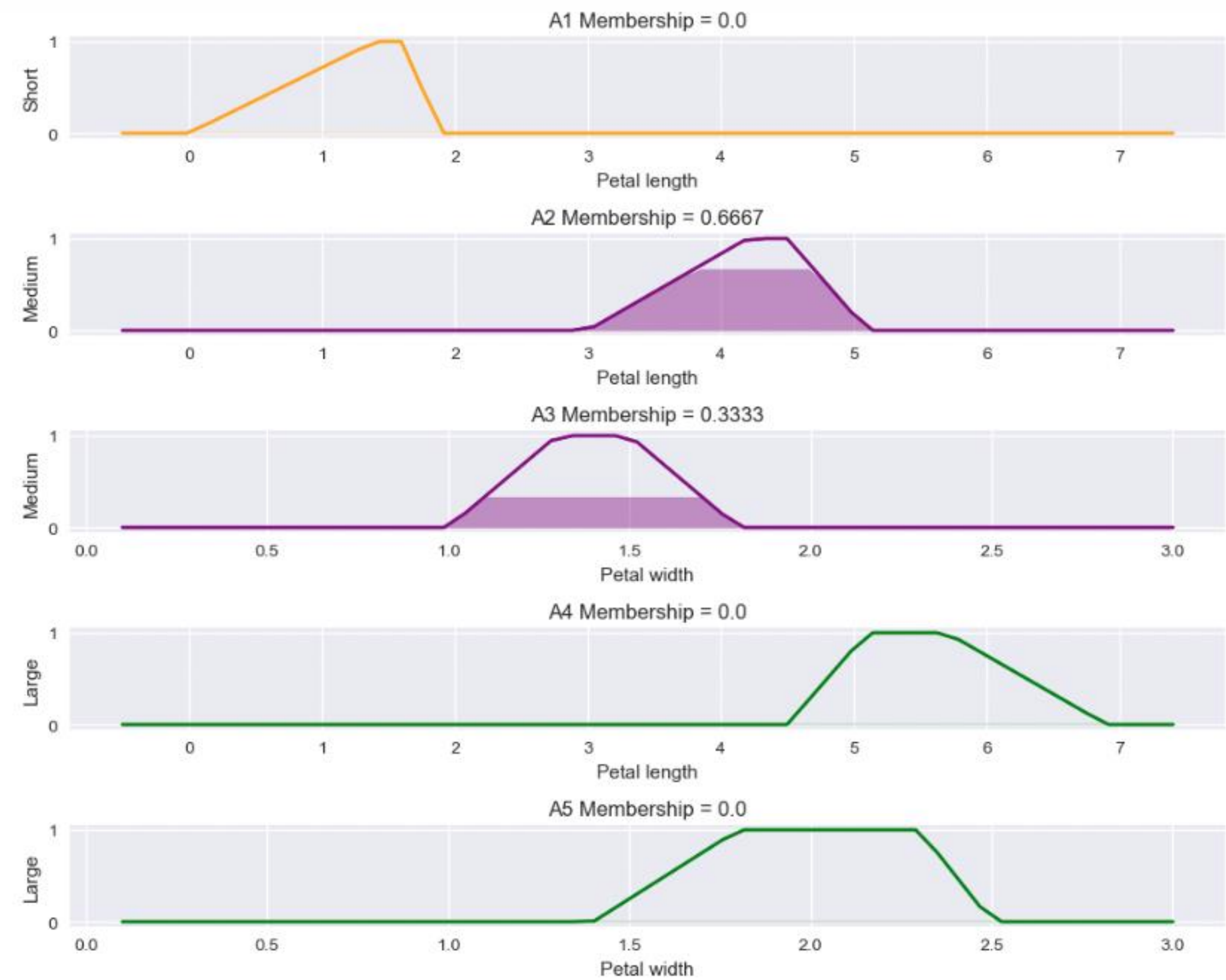

Figure 14. Antecedent membership designations for Index 80 (which we know belongs to class 1)

Testing the Mamdani system

The Mamdani system is uses correlation min to assign class belonging. A2 was assigned 0.6667, and A3 was assigned 0.3333, so correlation min chooses 0.3333, and Figure 15 shows 0.3333 confidence for C1 membership and 0 for C0 and C1.

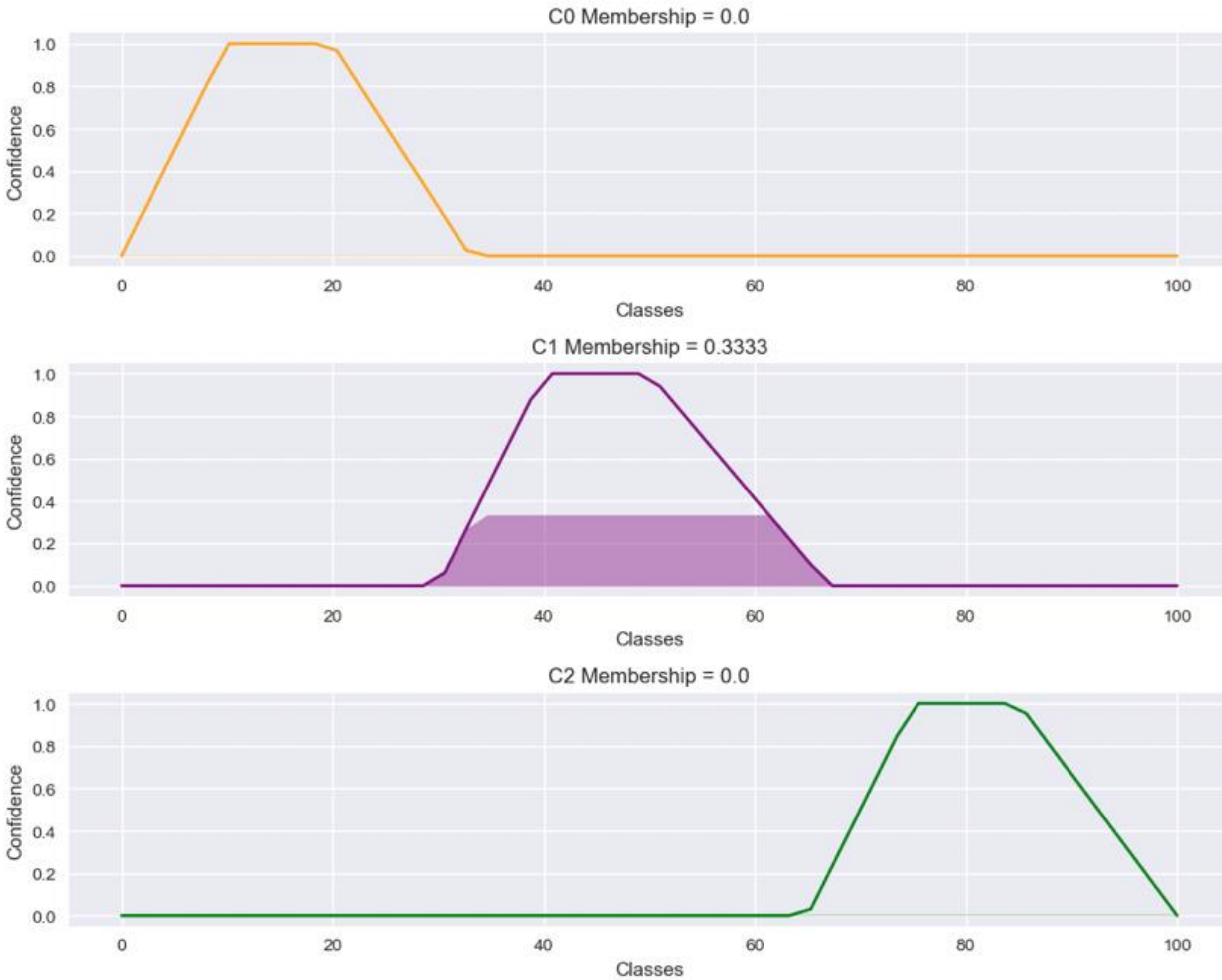

Figure 15. Class membership using correlation min for Index 80

Our logical inference system uses the max operator for aggregation. Since class 0 and Class 2 have 0 degree membership, 0.3333 is the max, and Index 80 is assigned to Class 1. A visual representation of the decision made after centroid defuzzification is shown in Figure 16.

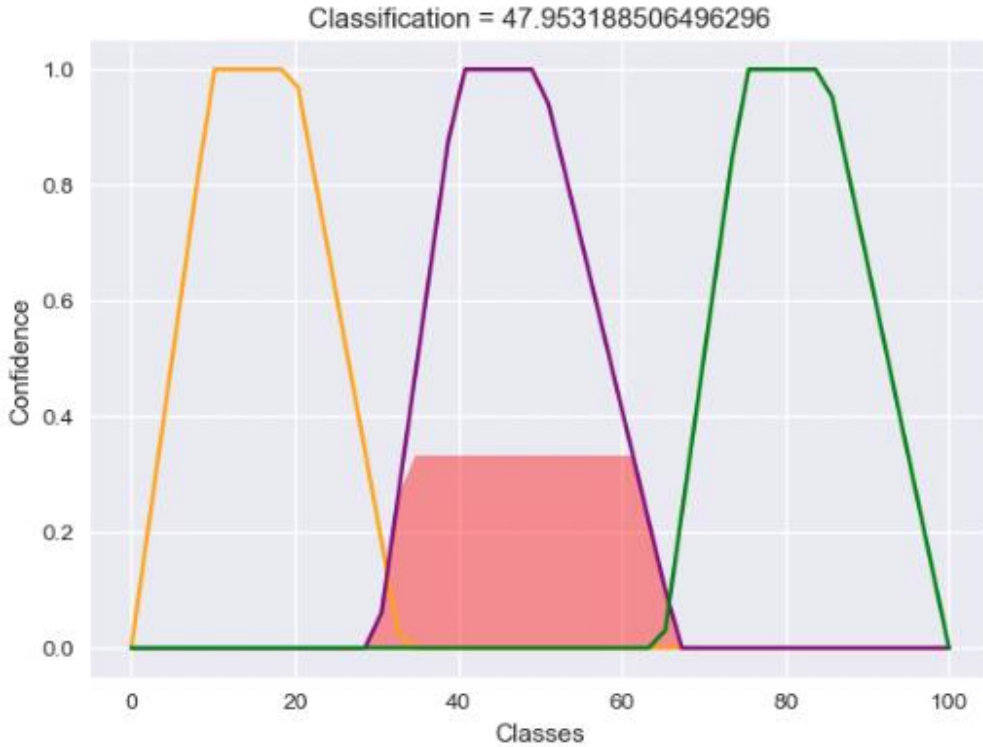

Figure 16. Mamdani decision for Index 80: Class 1 (correct! ☺)

We ran every observation of the Iris dataset through the Mamdani system with the rules we handcrafted, and the system returned the correct classification 97.3% of the time. The system fails to correctly classify indices 70, 77, 83, and 106.

Mamdani FIS Failure

Let's see what one of these failures looks like. Index 70 of the dataset belongs to Class 1. When we give this input to the Mamdani system, it calculates the degree of membership of the input to each antecedent, shown in Figure 17. The degree of membership is shown visually as well as analytically. We can see this observation has 0.5 membership to A2, which is associated with its correct class, but it has very low membership to A3. Since Mamdani uses correlation min, we lose that high A2 membership, and end up with 0.0394 for Class 1, shown in Figure 18. Both A4 and A5 have high membership for this observation, so even with correlation min, the class membership to Class 2 is calculated to be higher than class membership to Class 1. As a result, the system wrongly classifies this observation, shown in Figure 19.

This outlier shows us how dramatic correlation min can be when class membership is ultra-fuzzy.

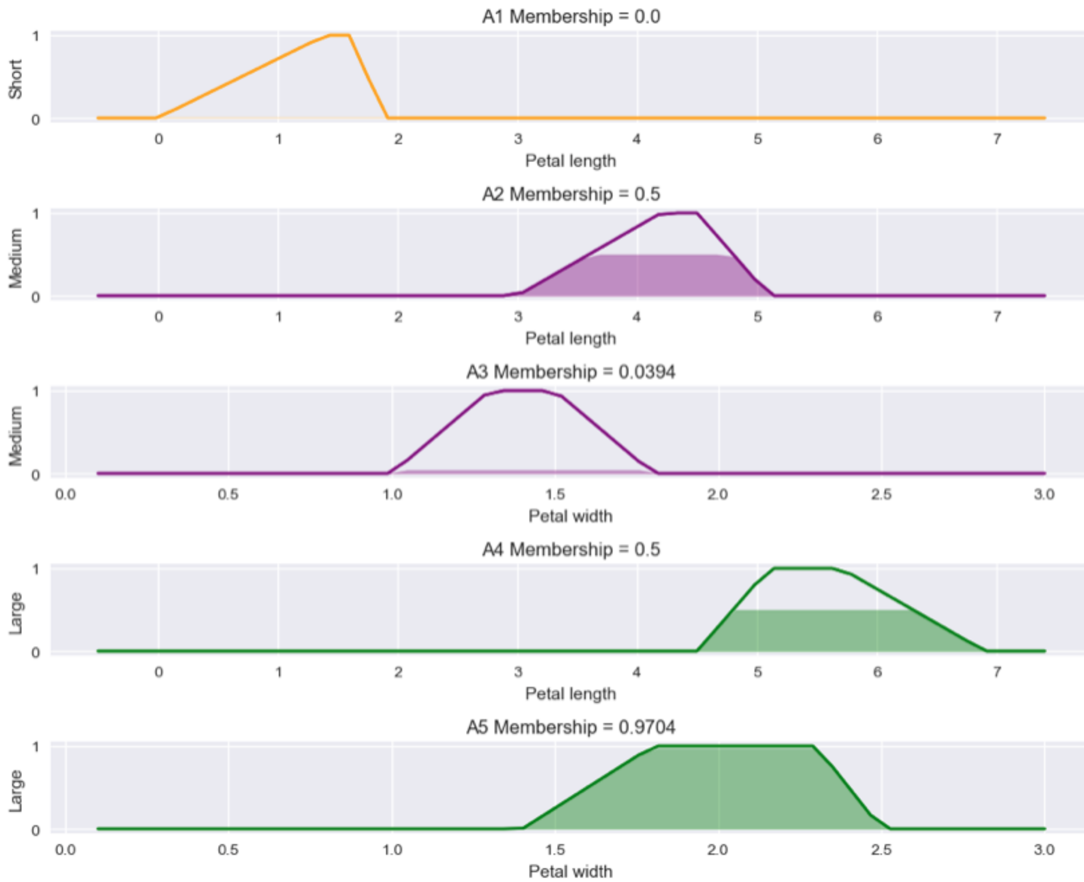

Figure 17. A single input passed through the antecedent membership functions

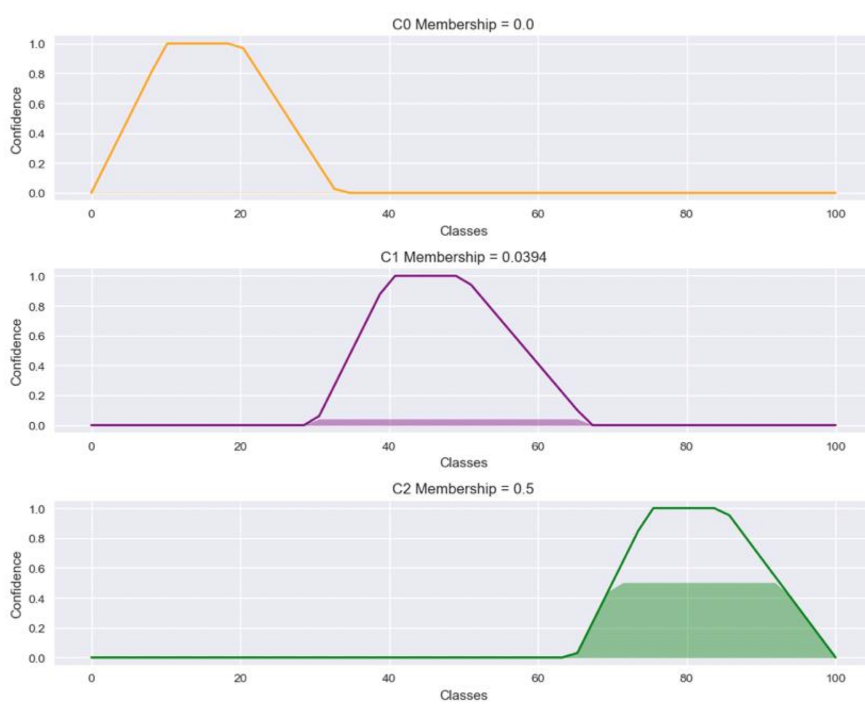

Figure 18. Assigned class membership for Index 70 using correlation min

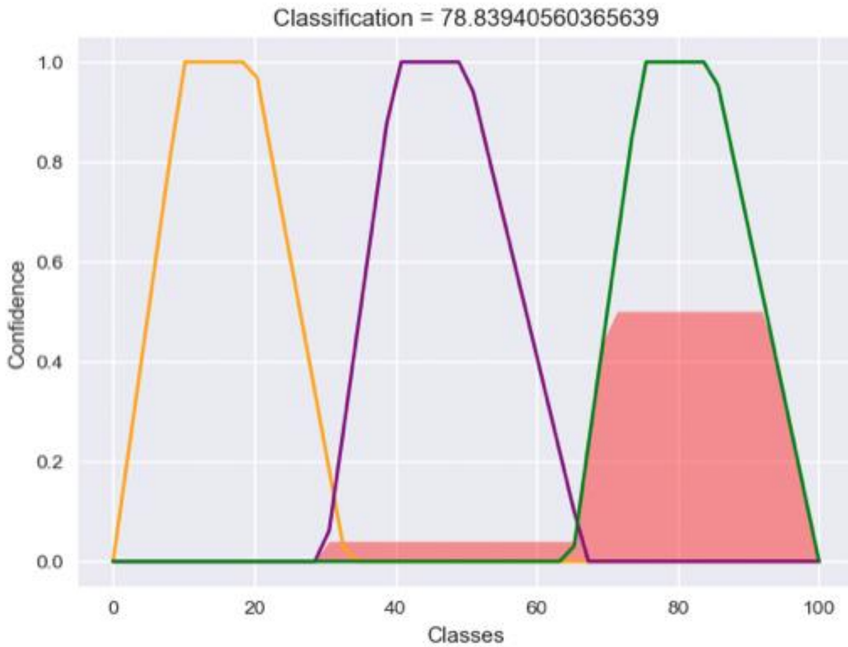

Figure 19. Classification of index 70 of the Iris data set into class 2 (wrong! ☹)

An advantage of the fuzzy logic system is its transparency. We just observed the end-to-end fuzzy inference process for an observation which the system fails to classify, which means we can begin to make an exception rule. We know from Figure 17 that there is some ambiguity between petal length being somewhat medium and somewhat large, so we may need to look at other features to classify this exception. This will probably require us to explore other domains, like sepal width and sepal length. This is further discussed in the Future Work section of the report.

Zadeh's FLS for performing logical operations

For this project, we provide a proof of concept for Zadeh's FLS using compositional rule of inference (Table 1) with Lukasiewicz and correlation-min operators, using Keller's example in Example 7.2 and 2.3 [1]. In this example, $X = 1,2,3,4, Y = a, b, c, d$. For A = SMALL and B = MEDIUM, we can consider both fuzzy sets as vectors over their domains.

$$A = SMALL = {}^{1.0}/_{1} + {}^{0.8}/_{2} + {}^{0.0}/_{3} + {}^{0.0}/_{4}$$
$$B = MEDIUM = {}^{0.0}/_{a} + {}^{0.5}/_{b} + {}^{1.0}/_{c} + {}^{0.5}/_{d} + {}^{0.0}/_{e}$$

Using these memberships, we calculate B' for two rules:

1. If A, then B
2. If NOT A, then B

For both rules, we follow the flowchart of Figure 8. We input Vectors A and B (or !A and B, for rule 2) into the fuzzy logic system, use either the Lukasiewicz or correlation min as the fuzzy relation operator followed by compositional rule of inference, and then we get a logical output.

Rule 1: If A, then B

For this rule, A and B do not change from above:

$$A = {}^{1.0}/_{1} + {}^{0.8}/_{2} + {}^{0.0}/_{3} + {}^{0.0}/_{4}$$
$$B = {}^{0.0}/_{a} + {}^{0.5}/_{b} + {}^{1.0}/_{c} + {}^{0.5}/_{d} + {}^{0.0}/_{e}$$

Rule 1 using Lukasiewicz

$$R_z = \begin{bmatrix} 0.0 & 0.5 & 1.0 & 0.5 & 0.0 \\ 0.2 & 0.7 & 1.0 & 0.7 & 0.2 \\ 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \end{bmatrix}$$

$$B' = A \circ \ col_i(R_z)|\forall i = \{\ 0.2, 0.7, 1.0, 0.7, 0.2\ \}$$

Note that B' does not equal B. That is because this is a fuzzy output, not a crisp output. This is not incorrect; it gives degrees of belonging rather than binary answers associated with modus ponens. This becomes clearer as we consider that the "shape" of B is roughly the "shape" of B' as in Figure 20. B is a crisp output, and B' is a softened, fuzzified output. Every value of B', or prediction, is centered around the mean of B. So Zadeh's system does not strictly follow logic, but facilitates fuzzy logic.

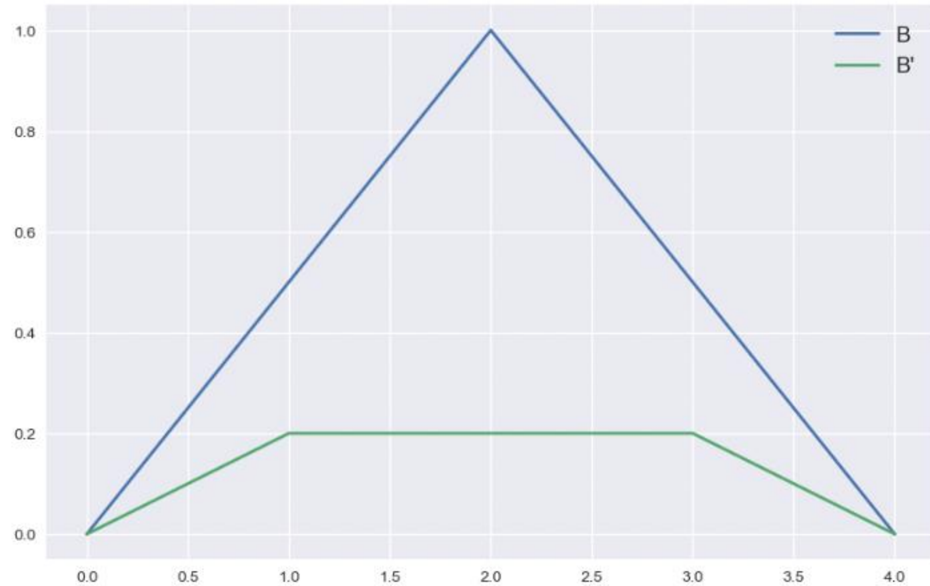

Figure 20. B and B' using Lukasiewicz relational operator for If A then B

Rule 1 using correlation min

$$R_z = \begin{bmatrix} 0.0 & 0.5 & 1.0 & 0.5 & 0.0 \\ 0.0 & 0.5 & 0.8 & 0.5 & 0.0 \\ 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \end{bmatrix}$$

$$B' = A \circ \; col_i(R) | \forall i = \{\, 0.0,0.5,1.0,0.05,0.0 \,\}$$

Note that correlation min does give B' equal to B. For This rule, using correlation min with the compositional rule of inference gives us the same logic output as modus ponens, but correlation min does not always follow true logic, as we will see in the following section.

Rule 2: If NOT A, then B

For this rule, A becomes A complement, and B is B. We use the fuzzy relation matrices, R_z, for Lukasiewicz and correlation min, which are calculated above.

$$A = A^c = {}^{0.0}/_{1} + {}^{0.2}/_{2} + {}^{1.0}/_{3} + {}^{1.0}/_{4}$$
$$B = {}^{0.0}/_{a} + {}^{0.5}/_{b} + {}^{1.0}/_{c} + {}^{0.5}/_{d} + {}^{0.0}/_{e}$$

Rule 2 using Luasiewicz

$$B_z{}' = A^c \circ \ col_i(R_z)|\forall i = \{ \ 1.0,1.0,1.0,1.0 \ \}$$

This output represents complete uncertainty. This shows how fuzzy logic manages to "encode" uncertainty. The semantic representation of this is, "If not A, then we don't know B."

Rule 2 using correlation min:

$$B_{cm}{}' = A^c(R_{cm})|\forall i = \{ \ 0.0,0.2,0.2,0.0 \ \}$$

Correlation min does not translate the logical equivalence of implication. Correlation min "breaks the rules of logic," producing an output set with very small memberships, indicating low correlation between antecedent and fact. This means the input is actually unrelated to the output, which is okay because this is not strictly a logical operator.

Conclusions and Future Work

This project serves as an introduction to an end-to-end fuzzy logic system, demonstrating handcrafted, explainable AI, especially for classification, but also for facilitating logical operations. We showed that, unlike genetic algorithms and neural networks, fuzzy systems give the developer complete control, since the developer makes the rules. This means the system is somewhat brittle, because it is only as good as the rules you give it. On the other hand, the system is completely transparent and allows for exception rules, so the algorithm can be tuned to specific applications. This has the effect that fuzzy logic systems tend not to be generalizable or scalable.

The implementation of the Mamdani system presented in this paper is a simple system which could be expanded to allow for more complexity. When we say this is a simple system, we mean the class memberships all exist on the same domain. This makes for relatively simplistic implementation, but it builds a bias into the system. Figure 13 shows an overlap between Class 0 and Class 1, as well as overlap between Class 1 and Class 2, but there is no overlap between Class 0 and Class 2. This works out fine for the Iris classification problem because we know there is no overlap between Class 0 and the other classes, but it still lacks the precision that a more complex system might offer. A more complicated and also more precise alternative is presented in Figure 21, where membership functions are defined on a confidence domain. In this system each class gets its own graph (only Class 0 is shown as an example), where memberships are defined as low, medium, and high confidence. A fuzzy inference system implemented with these consequence membership functions would allow us to encode more complex relationships among classes. Such a system would require more rules, which would be more cumbersome than the simple system, but takes advantage of the "white box" nature of the fuzzy logic system, in which we are able to make exception rules. If we implement a complex system like this, we need to be careful of overfitting, but we can find a smart way to get correct classifications almost all of the time.

For example, if we revisit the "problem child" of Figs 17-19, we identify the problem that membership antecedents for both Classes 1 and 2 fire. We might find a way to attribute higher confidence to the Class 1 membership so that it gets classified correctly. This might be tricky, since the Class 2 memberships fire fairly high for this observation, but we have the option to include more rules, so we might find a clever way to classify this observation.

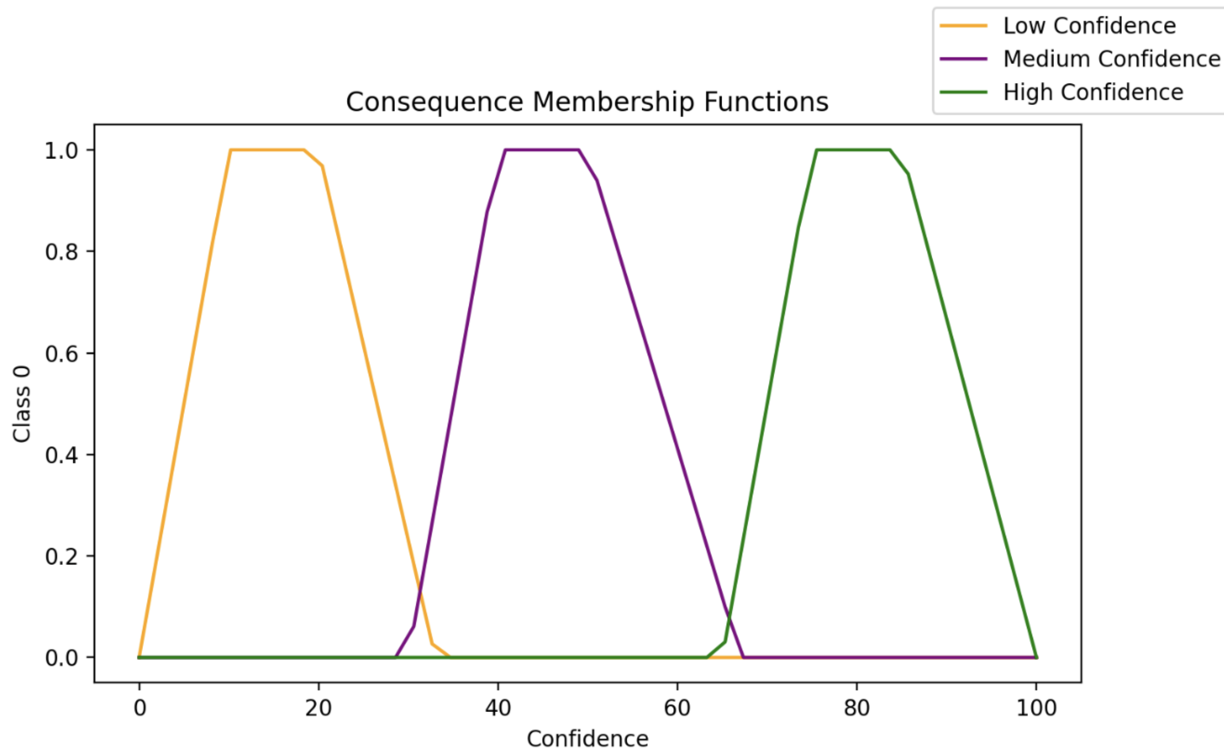

Figure 21. Membership functions defined on confidence domain

Still, the rules we created for this simple system output the correct classification 97.3% of the time. As mentioned before, four of the 150 observations were wrongly classified in the simple Mamdani system. In addition to considering a complex system, we might also visualize our "problem children" to gain some insight on what effective exception rules might be. We recreated the plots of Figure 9 with all data points muted in color except for the wrongly classified ones, which are displayed more prominently as stars. Unsurprisingly, these data points exist in spaces on the boundary between Class 1 and Class 2, and sometimes as outliers. Perhaps the graphs showing outliers will be more instructive for exception rule-making. For example, we could make an exception rule for Class 2 based on sepal length and sepal width.

This graph gives us a good place to start if we want to try to keep a simple system rather than change the algorithm significantly to something more complex, or perhaps we could do both.

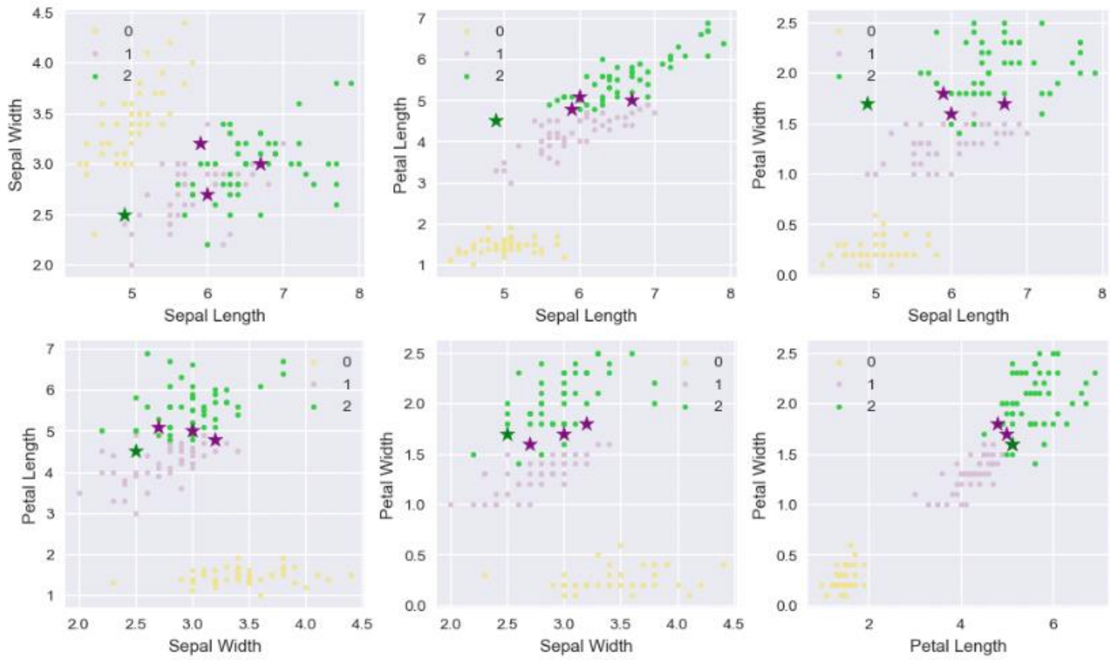

Figure 22. Visualization of the "problem children" in this project's implementation of Mamdani FLS

As for the Zadeh FLS, we showed that we can use fuzzy logic to extend modus ponens logic, but we can also break the rules of logic; that is, we can fail to satisfy crisp modus ponens and still contain truth in the sense that a fuzzy output can represent *degrees* of membership. Zadeh's compositional rule of inference allows us to consider uncertainty in problem solving. Future work would involve applying the Zadeh system to a real-world application, such as a control problem or perhaps to classify Iris or another dataset.

References

[1] J. M. Keller, D. Liu, and D. B. Fogel, *Fundamentals of Computational Intelligence*. John Wiley & Sons, 2016.

[2] A. P. Engelbrecht, *Computational Intelligence*. John Wiley & Sons, 2002.

[3] "Fuzzy Inference Process - MATLAB & Simulink," *www.mathworks.com*. https://www.mathworks.com/help/fuzzy/fuzzy-inference-process.html#FP346 (accessed Oct. 29, 2022).

[4] "The Iris Dataset," *scikit-learn*. *https://scikit-learn.org/stable/auto_examples/ datasets/ plot_iris_dataset.html*