University of Missouri

Project 2: Fuzzy Inference Systems

Student: Aaron Harrison

Student ID: 18185071

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF EQUATIONS

1 PROJECT DESCRIPTION

In this project a Mamdani Fuzzy Inference System is implemented to perform a variety of different tasks, such as regression and classification. First to understand how fuzzy inference systems work fuzzy sets, membership functions, and fuzzy logic will be discussed. Next each task is described.

1.1 FUZZY SETS

In a traditional or classical set an item is either in the set or it is not. This allows for no variation in interpretation between people. A fuzzy set allows for an item to be partially in a set and partially in another set. An example of this is shown in Figure 1 through temperature and how people would describe different levels through linguistic variables. The figure says that if it is below 30 degrees it is freezing, but in between 30 degrees and 50 it might be freezing or it might be cold. This represents the difference between different people's views of the word freezing. In a crisp set there would be no overlap; everyone would have to agree on what is cold and what is freezing. Each variable of freezing, cool, warm, and hot each have their own membership functions which describe how likely a temperature is to be in the set for that linguistic variable.

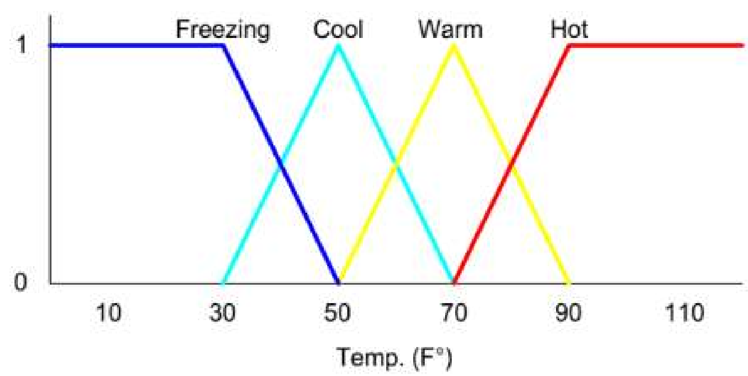


Figure 1. Example of a fuzzy set

1.2 MEMBERSHIP FUNCTIONS

Again, membership functions show how likely a certain value is to be in a certain set. There are many different types of membership functions, but the ones covered in this paper will be triangle, trapezoidal, and gaussian.

1.2.1 Triangle

A triangle membership function is shown in Figure 2. It is created using 3 points a, b, and c. A is the starting point and then goes linearly to b. B is the max of the function, or where the membership value equals one. Then B goes linearly to c. The equation is shown in Equation 1.

$$u(x) = \begin{cases} \dfrac{x-a}{b-a}, & a \leq x \leq b \\ \dfrac{c-x}{c-b}, & b \leq x \leq c \\ 0, & otherwise \end{cases}$$

Equation 1. Triangle membership function

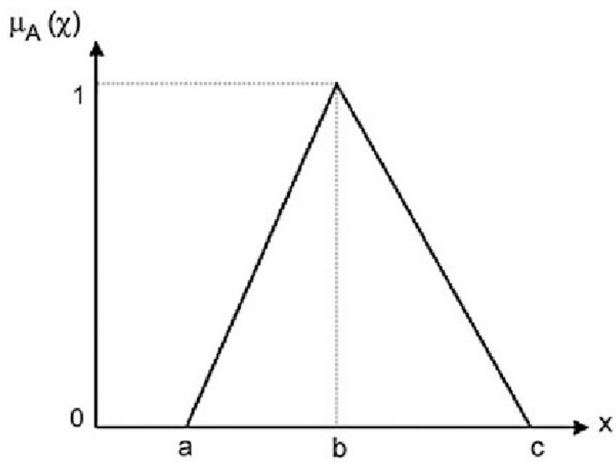


Figure 2. Triangle membership function

1.2.2 Trapezoidal

A trapezoidal membership function is shown in Figure 3. It is created using 4 points a, b, c, and d. A is once again the starting point and goes linearly to b. B is equal to one and extends horizontally to c which is also equal to one. Finally, c goes linearly to d. The equation is shown in Equation 2.

$$u(x) = \begin{cases} \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & b < x < c \\ \dfrac{d-x}{d-c}, & c \le x \le d \\ 0, & otherwise \end{cases}$$

Equation 2. Trapezoidal membership function

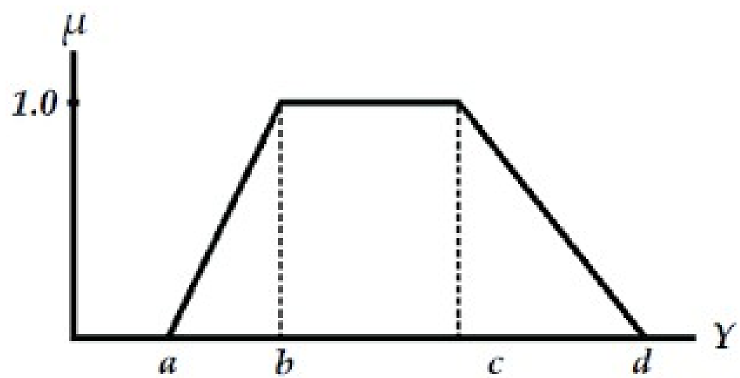


Figure 3. Trapezoidal membership function

1.2.3 Gaussian Membership Function

A gaussian membership function is shown in Figure 4. It is described with two main variables μ, the mean, and σ, the standard deviation. The max value of one is at the mean.

$$u(x) = e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Equation 3. Gaussian membership function

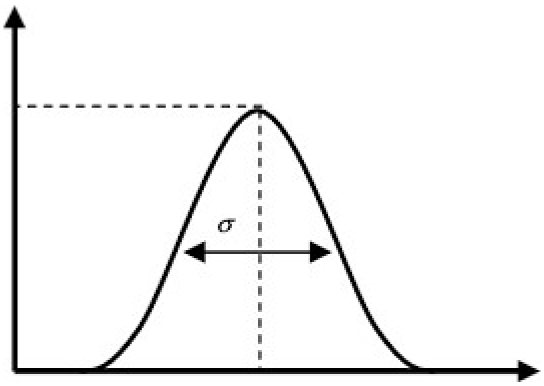


Figure 4. Gaussian membership function

1.3 FUZZY LOGIC

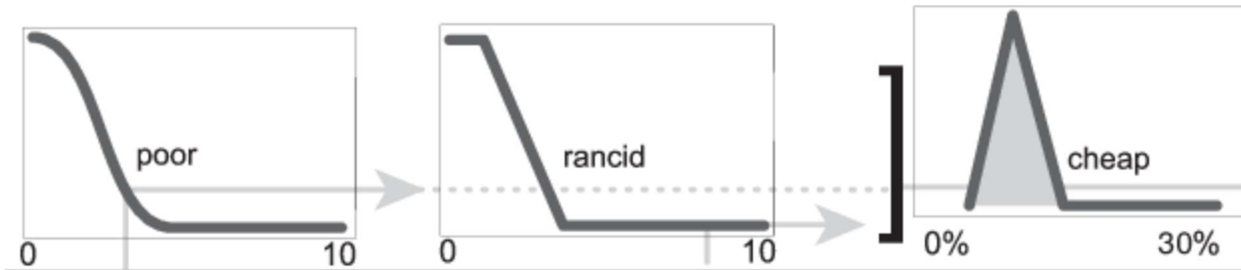
In classical or binary logic propositions are always true of false. This can create problems when there is uncertainty in a statement. Fuzzy logic creates the ability for logic to be on a spectrum of true or false to handle this uncertainty. To do this fuzzy logic makes use of fuzzy sets to display this uncertainty. For example, using Figure 1, if someone ask if it is hot when the temperature is 80 degrees, fuzzy logic will say there is a 50% chance it is hot. Which is a logical conclusion as some people would consider 80 degrees hot and others would not.

Fuzzy logic often takes the form of Equation 4. The beginning of the if statement is considered the antecedent or what is known to be true or false and causes the consequent. The consequent is the then part of the if then statement. It represents what the outcome of the antecedent is.

$$If\ U_1 is\ A_1\ and/or\ \ldots\ and/or\ U_n is\ A_n\ then\ V\ is\ B$$

Equation 4. Fuzzy logic

In fuzzy logic U_n is a variable, A_n is a fuzzy set, V is a variable, B is a fuzzy set. In Figure 5 an example of fuzzy logic is shown. Here if the service is poor or the food is rancid then the tip is cheap. Poor, rancid, and cheap are all represented as fuzzy sets.



Service is poor or Food is rancid then Tip is cheap.

U_1 is A_1 or U_2 is A_2 then V is B

Figure 5. Example of fuzzy logic

If a customer was to describe their service as a 3 and their food as a 3.5, how would the membership of the tip to be cheap be determined? This is represented in Equation 5 where A_1' and A_2' are fuzzy sets consisting of just one location equaling 1 and all else equaling 0.

$$If\ U_1\ is\ A_1'\ and/or\ U_2\ is\ A_2'\ then\ V\ is\ B'$$

Equation 5. Fuzzy logic use

B' is computed using Equation 6. First a fuzzy operation must be performed to combine the antecedents. For "or" statements it is the max value of the antecedents. For "and" statements the minimum is chosen. In the example above, a 3 rating for service corresponds to a 0.2 membership value, and a 3.5 for food corresponds to a 0.1 membership value. Since it is an "or" statement the max is taken.

Next implication is performed. There are many different implication operators, but the one used in this paper is correlation min. Correlation min results in taking the min of the combination of the antecedents and the relationship, which was computed as the original B. For the example described the output is Figure 6.

$$B'(y) = \sup_{x \in X}(A_1' \wedge \ldots \wedge A_2' \wedge R) \quad for\ \text{“and”}$$

$$B'(y) = \sup_{x \in X}(A_1' \vee \ldots \vee A_2' \wedge R) \quad for\ \text{“or”}$$

Equation 6. Computation of B' using correlation min

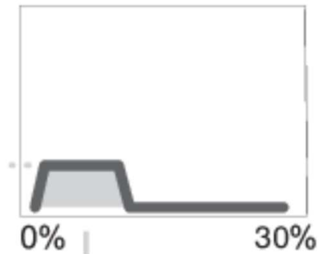


Figure 6. *Output of implication min*

1.4 Fuzzy Inference System

A fuzzy inference system is a method for reasoning created using fuzzy logic. A fuzzy inference system being used for control theory is shown in . The first step is to receive a crisp input from the controller. This input must go through fuzzification so the inference engine can work. The inference engine makes use of a set of fuzzy logic rules to create a fuzzy set as output. The fuzzy set output then goes through defuzzification to allow for a crisp output from the inference system.

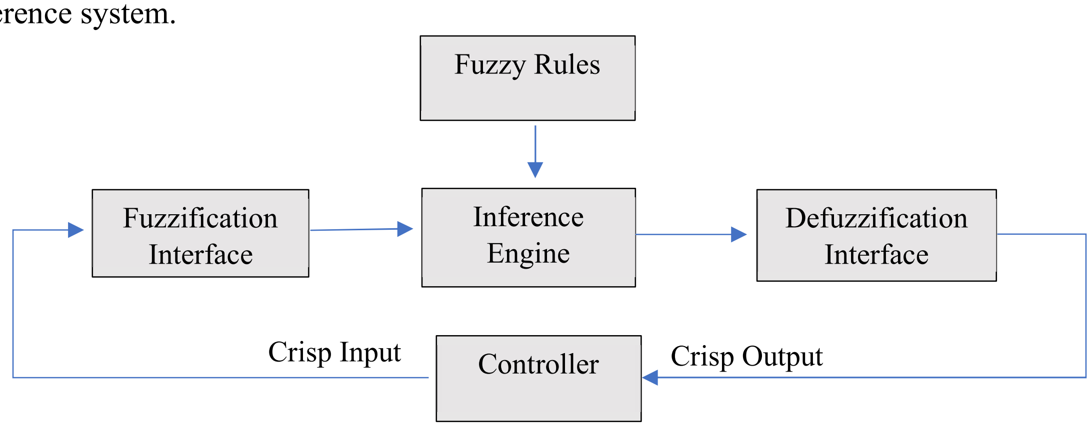


Figure 7. *Fuzzy inference system*

An example of a Mamdani fuzzy inference system is shown in Figure 8. To fuzzify inputs, each input is matched to the membership of the input to the rule. Then each rule is calculate using the fuzzy logic discussed in 1.3. The output of each rule is then combined using an aggregation method. There are many different aggregation methods, but only max and sum will be discussed. Max is where you take the max membership at each value, where as sum is where you sum the memberships at each value.

Finally, the last step is to defuzzify the output fuzzy set. The two defuzzification methods used are max and centroid. Max is where you take the value in the aggregated fuzzy set that has the highest membership. An issue can be caused when multiple values have the max membership. Centroid defuzzification finds the center of mass for the output fuzzy set. It is shown in Equation 7. Where $y(n)$ is the value and $B'(y_n)$ is the membership value of y_n.

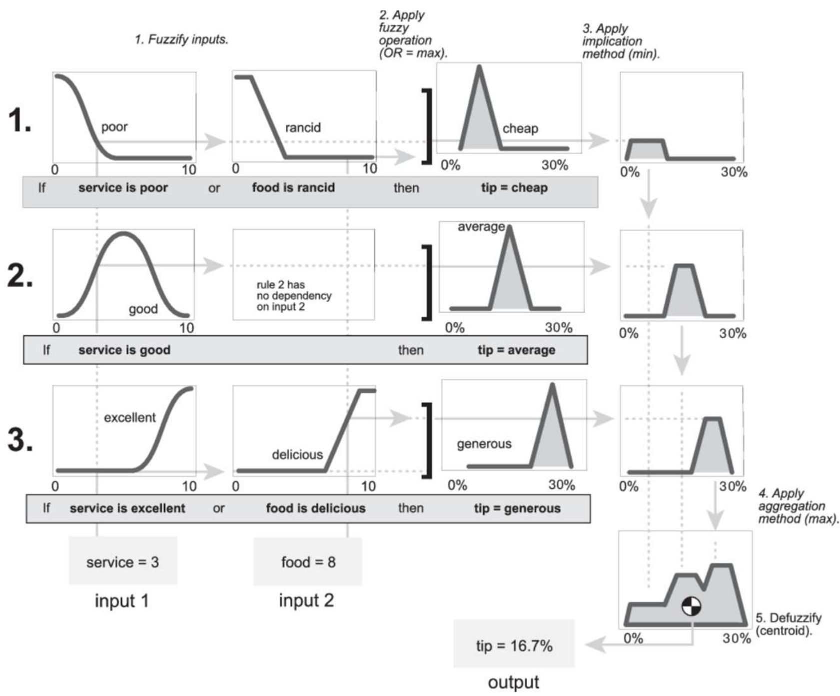


Figure 8. Mamdani fuzzy inference system

$$Y = \frac{\sum B'(y_n) * y_n}{\sum B'(y_n)}$$

Equation 7. Centroid defuzzification

2 Experiments and Results

This section of the report discusses the implementation of the Mamdani Fuzzy Inference System in python, the validation of it, and performing both classification and regression with it.

2.1 Implementation of Mamdani Fuzzy Inference System

The Mamdani Fuzzy Inference System was implemented in python through two separate classes. The first was the Rule class. This class creates the fuzzy logic discussed in 1.3. It allows for the easy creation of gaussian and trapezoidal antecedents and consequents. It also allows for the creation of custom consequents. To create a gaussian antecedent the function add_gaussian_antecedent(a, lingusted_variable, mean, standard_deviation). This adds the antecedent to the rule. For the Rule class only AND operations are allowed between antecedents. For trapezoidal and custom antecedents and all consequents, the process is very similar.

The second class is the Mamdani class which implements the fuzzy inference system discussed in 1.4. During the creation of this class many parameters are set including the rules, range, aggregation method, and defuzzification method. The rules are a list comprised of the Rule class. The range is the domain of the output fuzzy set. The aggregation method can be set to any function wanted, but the default is max. The defuzzification method can be set to centroid or max, but the default is centroid.

To make an inference with the Mamdani object the make_inference(inputs) function is called. The inputs variable takes a list of dictionaries defined as {“name”: U_n, “value”: A'_n}. Where U_n is the features name and A'_n is the features value. This function fuzzifies the inputs and performs the minimum operation of “and”. Next is performs the implication onto each rule. Next aggregation is performed to combine all fuzzy outputs from each rule. Then the system outputs the crisp value performed by the defuzzification method set on initialization.

2.2 VALIDATION OF FUZZY INFERENCE SYSTEM

For the validation of the fuzzy inference system a simple system was made for how a person should tip. A rule base consisting of four rules were added: if service is poor and food is rancid the tip low, if service is ok and food is ok, then tip medium, if service is great and food is great then tip good, if service is exquisite and food is exquisite then tip generous. The rule base appears in Figure 9.

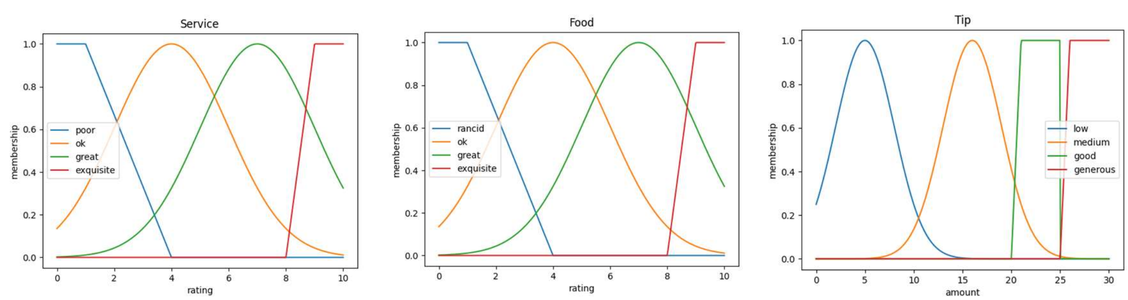


Figure 9. Validation Rule Base

The Mamdani class was implemented with the rule base, a range of [0, 30], max as the aggregation method, and centroid as the defuzzification method. For the first test values of 6 for service ranking and 4 for food ranking were chosen. After the value was fuzzified and the mins were selected the following min values were given: tip low - 0, tip medium - 0.606, tip good - 0.324, tip generous – 0. Next the implications were performed shown in Figure 10. Then aggregation was performed in Figure 11. Finally, defuzzification was performed using the centroid method producing the result of 17.0 %.

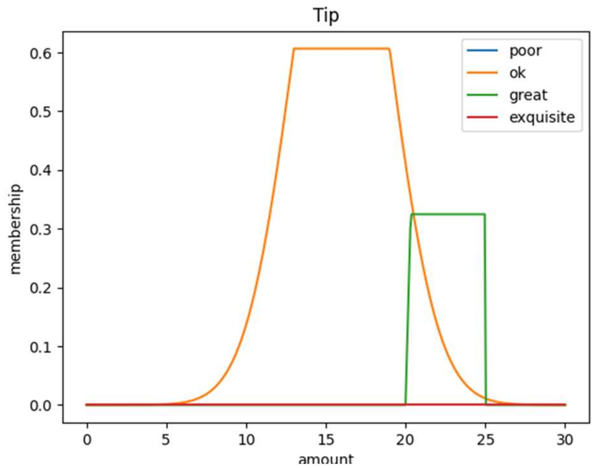


Figure 10. Implication of validation

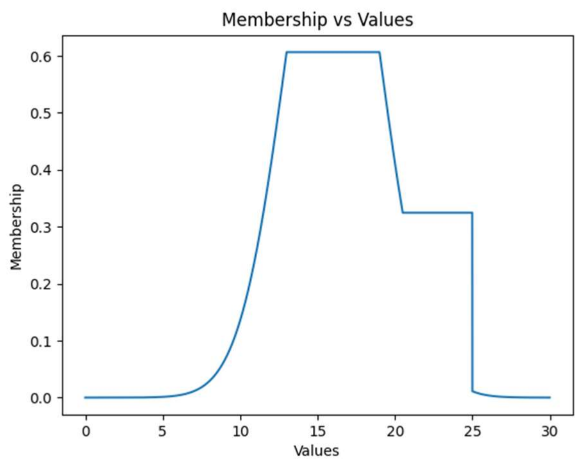


Figure 11. Aggregation of validation

Finally, the last step in validation was to run rankings of 1-10 of service against rankings of 1-10 of food and see the tip values. The results are displayed in Table 1. The results obtained are consistent with expected results.

Table 1. Service vs Food with Tip Percent

	1	2	3	4	5	6	7	8	9	10
1	7.46	7.46	7.83	9.14	16.0	16.0	16.0	16.0	16.0	16.0
2	7.46	9.03	9.52	11.03	16.0	16.0	16.0	16.0	16.0	16.0
3	7.83	9.52	10.74	12.21	16.04	16.04	16.04	16.06	16.12	16.32
4	9.14	11.03	12.21	12.89	16.26	16.26	16.31	16.46	16.87	18.59
5	16.0	16.0	16.04	16.26	16.83	16.85	17.0	17.44	18.53	20.29
6	16.0	16.0	16.04	16.26	16.85	17.64	17.89	18.53	19.75	21.2
7	16.0	16.0	16.04	16.31	17.0	17.89	18.51	19.22	20.41	21.2
8	16.0	16.0	16.06	16.46	17.44	18.53	19.22	19.45	20.41	21.2
9	16.0	16.0	16.12	16.87	18.53	19.75	20.41	20.41	20.41	21.2
10	16.0	16.0	16.32	18.59	20.29	21.2	21.2	21.2	21.2	24.92

2.3 CLASSIFICATION

An experiment of the fuzzy inference system was conducted to perform classification. The data set chosen was the iris dataset from the python library sklearn. The library consists of 3 different types of irises (Setosa, Versicolor, and Virginica) and 4 features for each iris (sepal length, sepal width, petal length, and petal width). The goal of the classification problem was to correctly identify each individual iris as the correct class. To perform this, the data was split into two categories. The first category, consisting of 70 % of the dataset, was for computing statistics, and the second category was for testing.

2.3.1 Computing statistics

The first step in the classification problem was to compute the statistics for each feature for each class in the dataset. Both the mean and the standard deviation were calculated and are displayed in Table 2.

Table 2. Statistics of iris features. mean(std)

	Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)
Setosa	4.96(0.33)	3.38(0.37)	1.46(0.18)	0.25(0.11)
Versicolor	5.86(0.52)	2.72(0.30)	4.21(0.49)	1.30(0.20)
Virginica	6.56(0.65)	2.99(0.31)	5.55(0.54)	2.01(0.293)

2.3.2 Visualization of Data

While statistics are very useful in determining what the rules should be, a better way might be to visualize the data. In Figure 12, each iris's data point is plotted for each feature and, each class is shown by a different color.

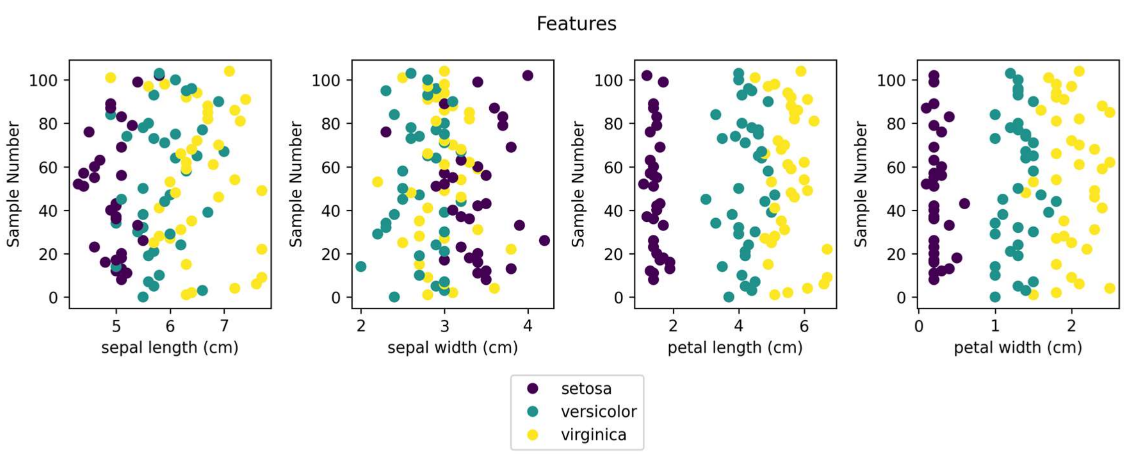


Figure 12. Iris data plotted.

2.3.3 Choosing of Rules

For the choosing of rules both the statistics and visualized data are looked at. Immediately it is apparent that the petal width would be a good feature for a rule set as the classes are separated decently well. For the formation of the first rule, a trapezoidal antecedent is added for the petal width being small that's points are a = 0, b = 0, c = 0.5, and d = 0.8. The consequent is a custom one that results in 1 only for class 0, which is setosa. The second rule has an antecedent for the petal width being medium. Here the fuzzy set is a gaussian distribution with the mean and standard deviation for the versicolor [1.30(0.20)]. The consequent is once again a custom one that results in 1 only for class 1, which is the versicolor. Finally, the third rule insist of a gaussian antecedent with the mean and standard deviation for the virginica [2.01(0.293)] and a custom consequent that results in 1 only for class 2, which is the virginica. The results of this rule base on the test data are displayed in Table 3. This resulted in 43 correct predictions and two incorrect predictions for an accuracy of 95 %.

Table 3. Iris confusion matrix

	Setosa Actual	**Versicolor Actual**	**Virginica Actual**
Setosa Predicted	19	0	0
Versicolor Predicted	0	11	2
Virginica Predicted	0	0	13

With this some of the predictions made are very close between versicolor and virginica as they have overlap. For example, in Figure 13 the inference system is almost similarly confident in versicolor and virginica, but choses virginica as it is about 0.03 percent higher.

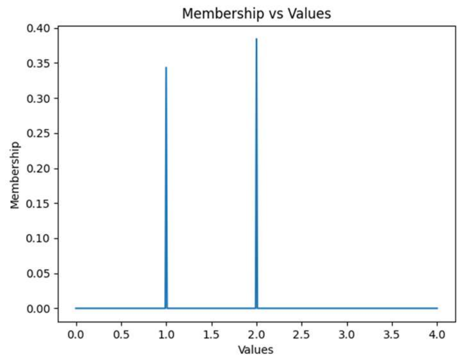


Figure 13. Figure of aggregated membership values

The second feature noticed is the petal length as the classes are separated decently well here as well. For the rule base it is implemented very similarly to the first rule based, but with the gaussians' means and standard deviations being the values for petal lengths, and the trapezoid for the setosa values being a = 0, b = 0, c = 2, and d = 2.5. Here this results in a perfect evaluation of the test data.

Due to the result evaluating to perfect for the test data everything was then tested again with the full iris data. In Figure 14, the confusion matrixes are shown for each feature rule set. In conclusion, the combination of the two results in the best performance at 96.7 %. While the single petal length performed close at 95.3 % and the petal width performed the worse, but still close at 94.6 %.

Petal Width

50	0	0
0	45	5
0	3	47

Petal Length

50	0	0
0	46	4
0	3	47

Petal Width and Petal Length

50	0	0
0	47	3
0	2	48

Figure 14. Confusion matrixes for entire data set

2.4 REGRESSION

An experiment of the fuzzy inference system was conducted to perform regression for prediction on fantasy football scores. To perform this game data was received through an API from Sports Radar. The desired data involved a players averages going into a game, the defense they were playing averages, and the players actual performance during the game. To achieve the desired data, multiple steps were taken such as combination of multiple files and filtering. Once the data was achieved the next step was to compute statistics. The 2022 season was picked as the data to pick statistics and the 2023 season was picked for testing.

2.4.1 Computing statistics

The statistics that were computed involved a player's yards per game and the defense's allowed yards per game. For the players' yards per game, they were separated into five categories: elite, good, average, ok, and poor. For the defense's allowed yards per game, they were separated into 15 categories. The first one was for crisp logic that separated by position. The next was fuzzy logic that separated into the same five categories as the players. For the regression problem, only the wide receivers are currently being evaluated. The results are display in Table 4 and Table 5.

Table 4. Player yards per game stats

Class	Mean	Standard Deviation
Elite	93.5	59.2
Good	59.2	8.6
Average	30.8	7.6
Ok	16.4	1.8
Poor	9.7	2.7

Table 5. Defense yards per game stats

Class	Mean	Standard Deviation
Elite	219.3	17.0
Good	179.4	7.7
Average	162.5	3.0
Ok	147.1	6.6
Poor	112.4	14.5

2.4.2 Determine Rules

The rules were chosen to be each combination of receiving yards per game and defensive yards per game. Each antecedent was the gaussian with mean and standard deviation for that statistic. To determine the consequent each player per game's data was evaluated. A random number was selected for both the offense and defense. Then the players average yards per game and the defenses yards per game were plugged into a gaussian representing each category to find the membership value a player had in each category. If the random number was less than or equal to the membership value for both the player and defense, then the actual score the player received was added to the consequent data for that category. The consequents were also represented as gaussians.

2.4.3 Evaluation

The evaluation was performed by looking at the scores the regression fuzzy inference system calculated and what a reputable source of ESPN calculated. The results are shown in Table 6. In most cases the number predicted is very similar to the one that ESPN predicted.

Table 6. ESPN vs FIS

Player	ESPN	FIS
Stefon Diggs	22.9	19.8
A.J. Brown	20.0	20.4
Chris Olave	14.9	14.9
Tyreek Hill	24.5	22.4
Michael Pittman Jr.	14.2	14.8
Romeo Doubs	9.0	10.7
Jonathan Mingo	8.5	7.9
DK Metcalf	14.7	15.2
Skyy Moore	5.8	10.1
Ja'Marr Chase	22.9	20.4

3 CONCLUSIONS AND FUTURE WORK

In conclusion, a Mamdani fuzzy inference system is a method of computing logical reasoning. The system uses foundations of fuzzy logic and fuzzy sets to make inferences. To make an inference the system takes in a crisp input and fuzzifies it. Then the fuzzy set is put through the inference engine which consists of fuzzy logic rules. The output of the inference engine is a fuzzy set that must be defuzzied into a crisp output.

This system implemented in this paper was first validated through a rigorous test rule set of a tip amount to give. Each step was broken down to make sure the system was computing correctly. The system also performed classification on the iris data set with high accuracy of up to 96.7 %. Finally, the system matched ESPN fantasy football predictions through regression.

The future work of this system will involve every possible combination of rules of the iris data set being tested to see which gives the best value. Also, the performance on a new dataset of iris consisting of the same classes and features.

New work on the NFL fantasy stats will involve testing the accuracy with more stats. These stats will include the number of targets, the number of receptions, the average yards per catch, the amount of 20+ yards per catch, and more. While the comparison to a reputable source is a good validation, it does not directly compare to the results achieved by the players. To fix this the data will also be validated by comparing the predicted value with the actual amounts of points scored. Other items that may be involved could be the addition of weather types, stadium information, and injuries.

Nevertheless, a Mamdani fuzzy inference system was implemented, validated, and used in classification and regression.