PREDICTING BANK LOAN ELIGIBILITY AND IDENTIFYING THE FACTORS THAT AFFECT OUR BANK LOAN ELIGBILITY USING FUZZY LOGIC AND FUZZY CLUSTERING

A THESIS SUBMITTED TO
THE FACULTY OF ACHITECTURE AND ENGINEERING
OF
EPOKA UNIVERSITY

BY

ESTREF KATILLARI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR BACHELOR DEGREE
IN COMPUTER ENGINEERING

JULY, 2021

Approval sheet of Thesis

This is to certify that we have read the thesis entitled **“PREDICTING BANK LOAN ELIGIBILITY AND INDETIFYING THE FACTORS THAT AFFECT OUR BANK LOAN ELIGBILITY USING FUZZY LOGIC AND FUZZY CLUSTERING”** and that in our opinion it is fully adequate, in scope and quality, as a thesis for Bachelor’s Degree of Computer Engineering.

……………………………………...

M.Sc. Enea Mançellari

Date: July 1, 2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: …………………………

Signature:

ABSTRACT

PREDICTING BANK LOAN ELIGIBILITY AND IDENTIFYING THE FACTORS THAT AFFECT OUR BANK LOAN ELIGBILITY USING FUZZY LOGIC AND FUZZY CLUSTERING

Katillari, Estref

B.Sc., Department of Computer Engineering

Supervisor: M.Sc. Enea Mançellari

Calculating risk to determine creditworthiness is a crucial but time-consuming procedure that the Risk Management Department (RMD) of a bank goes through every time an applicant submits their documents. This process has always been in need for a reliable decision-making system, and, in this thesis, my goal is to create a system that makes this process quick and to obtain results that are beneficial and as close to the decision that would be made by the Risk Management Department office as possible. By creating a Mamdani Fuzzy Inference System, the level of risk will be calculated by using five key components in an applicant's information. Furthermore, by using Fuzzy C-means clustering we will have a look at how certain information in an applicant's application affects the final decision made by the bank.

Keywords: bank loan, credit risk, Fuzzy Logic, Fuzzy Inference System, clustering

ABSTRAKT

PARASHIKIMI I PRANUESHMËRISË SË KREDISË BANKARE DHE IDENTIFIKIMI I FAKTORËVE QË NDIKOJNË NË PRANUESHMËRINË E KREDISË BANKARE DUKE PERDORUR LOGJIKËN FUZZY DHE FUZZY CLUSTERING

Katillari, Estref

B.Sc., Departamenti i Inxhinierisë Kompjuterike

Supervizor: M.Sc. Enea Mançellari

Kalkulimi i riskut për të përcaktuar vlerën e kreditit është një procedure e rëndësishme por që konsumon shumë kohë dhe që bëhet nga Departmaneti i Menaxhimit të Riskut (DMR) i bankës cdo here që një aplikant dorëzon dokumentat e tij. Ky process ka patur gjithmonë nevojë një sistem ndihmues dhe të mirëbesuar, dhe, në këtë studim qëllimi im është që të krijojë një sistem që e përshpejtonë këtë process dhe arrin rezultate përfituese dhe sa më të përafërta me ato reale të mara nga Departmaneti i Menaxhimit të Riskut. Më së tepërti, duke përdorur Fuzzy C-means clustering do të shohim se si informacioni personal i një aplikanti ndikon në vendimin e marrë nga banka.

Fjalët Kyçe: kredi, risku i kreditit, Logjika Fuzzy, Fuzzy Inference System, clustering

ACKNOWLEDGEMENTS

To Prof. Enea Mançellari, I would like to express my sincerest gratitude for his ongoing assistance and counsel as I wrote this thesis and for his tireless work during the completion of my bachelor studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FL	Fuzzy Logic
FIS	Fuzzy Inference System
FCM	Fuzzy C-means

CHAPTER 1

INTRODUCTION

1.1. Introduction

Bank loans have always been and will continue to be an important part of a person's economy, with 51.3 percent of Americans admitting to taking a loan at some point in their lives, whether it was a minor loan to buy a car, a student loan or a larger loan to buy a house, which is quite prevalent nowadays. [1]

When someone request for a loan, the bank must ensure that the person who will be receiving this loan is not a risk to the bank’s financial stability, therefore putting a lot of pressure on the bank employees who hold the responsibility of analyzing each person’s documentation in order to make the safest decision.

The bank employees aim to keep the credit risk at a manageable level and the process of credit analysis is a multilayered process that takes into consideration a wide range of factors that go beyond just the financial means of an individual and require personal experience to analyze.

1.1.1. Problem Development

As mentioned above, loan risk calculation is very imprecise in nature, and besides the calculation of the financial means of an individual, analyzing elements like credit history, the character of an applicant or the nature of the collateral are built upon linguistic variables, human expertise and personal experience.

Manual methods, such as the use of pen and paper in the survey process for possible new customers, are still used in operational approaches that are still in the process of applying for loans. As a result, consumer data can be erroneous and unproductive when establishing creditworthiness. As a result, a system is required to assist in the classification of loan eligibility using a method that can process their classification.

1.1.2. Motivation

The motivation of this study is to analyze how certain factors affect an applicant's bank loan eligibility and how they weight in on bank's decision. Additionally, I want to showcase how this lengthy process could be done with the help of a Fuzzy algorithm.

1.1.3. Objective of the study

The objective of this thesis is to create a Mamdani Fuzzy Inference System that is able to calculate the level of risk given seven key components retrieved from an applicant's documentation and to provide a closer look on how certain factors determine our loan eligbility.

1.2. Introduction to Fuzzy Logic

1.2.1. Fuzzy Logic

The traditional approach to reasoning and computing has always been done through Boolean logic, where a variable can either be true or false, which is a very precise and restricted method. Even though this technique has proven to be quite useful there are two key issues that arise:

1. A comprehensive description of a genuine system typically necessitates far more detailed data than a human being could ever identify or comprehend at the same time.
2. Real-life circumstances are rarely clear and deterministic, and therefore are difficult to be defined precisely [2].

The concept of vagueness and fuzziness introduces us to fuzzy logic, which, contrary to the traditional methodology supports partial truth, where the value of certainty if a value between 0 and 1. It makes it able for machines to deal with ambiguous and inaccurate data.

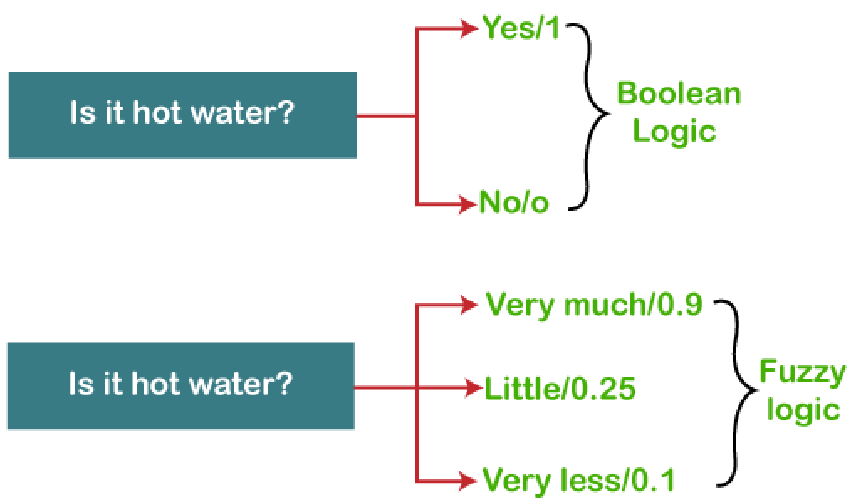


Figure 1. - Boolean Logic vs Fuzzy Logic

1.2.2. Fuzzy Clustering

Clustering is a machine learning technique that entails grouping data points into homogenous classes or clusters, with items in the same class being as similar as feasible and elements in other classes being as distinct as possible. Every approach has its own set of rules for determining similarity between data points. In practice, there are over a hundred different clustering techniques. Distance, connectedness, and intensity are some examples of values that can be employed as similarity measurements. Fuzzy clustering, also known as soft clustering, is a type of clustering in which each data point can be assigned to multiple clusters and every element has a probability of belonging to each cluster. To look at it another way, each element has a set of membership coefficients that correlate to the degree to which it is a member of a specific cluster. A numerical number ranging from 0 to 1 represents the degree to which an element belongs to a certain cluster. [3]

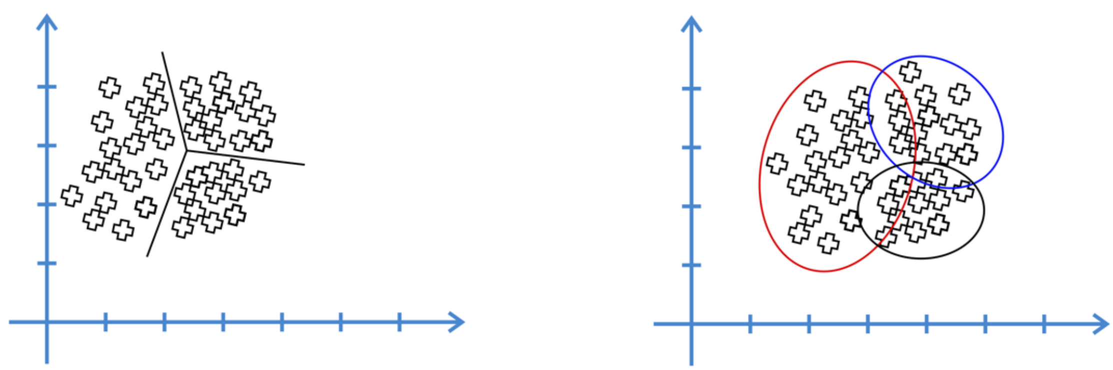


Figure 2. – Regular clustering vs Fuzzy clustering

1.2.3. Fuzzy C-means

As previously indicated, there are various clustering algorithms to choose from; however, I have chosen to employ the Fuzzy C-means, which is a well-known clustering methodology. It starts with a random initial guess for the cluster centers; that is the mean location of each cluster. It begins with a guess at the cluster centers, which is the average location of each cluster. Then, for each cluster, FCM assigns a random membership grade to each data point. FCM moves cluster centers to the correct location within a data set and finds the degree of membership in each cluster for each data point by iteratively updating the cluster centers and membership grades for each data point. This iteration seeks to minimize an objective function that represents the distance between any given data point and a cluster center, weighted by the data point's cluster membership. [4]

1.2.4. Fuzzy C-means algorithm

The fuzzy C-Means algorithm has as its objective to minimize the following:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{i,j}^{m} \left\| x_i - c_j \right\|^2$$

(Equation 1)

where:

I. m is a real number greater than 1.

II. $u_{i,j}$ defines the degree of membership of x_i in the cluster j

$$u_{i,j} = 1 / \sum_{k=1}^{C} \left(\left\| x_i - c_j \right\| / \left\| x_i - c_k \right\| \right)^{2/(m-1)}$$

(Equation 2)

III. c_j is the d-dimensional center of the cluster

$$c_j = \sum_{i=1}^{N} (u_{i,j}^{m} . x_i) / \sum_{i=1}^{N} u_{i,j}^{m}$$

IV. x_i is the i th of d-dimensional measured data

V. * is any norm expressing the similarity between any measured data and the center. [5]

The algorithm goes through the following steps:

1. Initialize $U = [u_{i,j}], U^{(0)}$
2. At the k[th]-step: calculate vectors $C^k = [c_j]$ with $U^{(k)}$
3. Update $u_{i,j}$
4. If $U^{(k+1)} - U^k < \sum$ stop; otherwise return to step 2.

1.2.5. Fuzzy sets and memberships

Boolean logic is based on Crisp Sets which are outlined by exact and well-defined features. This kind of set is a bi-valued set and an element either is a member or not. Contrary to Crisp Sets, Fuzzy Sets allow partial membership. The degree of membership, which can be any value from 0 to 1 (including both of them), determines this partial membership. If we take a look at Figure 3 we can see a clearer example of Fuzzy sets and the membership function regarding the linguistic variable "age". [6]

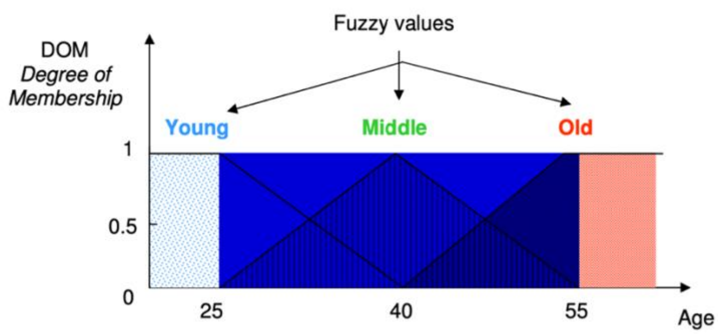


Figure 3. – Fuzzy triangular membership functions for the variable "age"

For example, if we say "John is 28" that means that he belongs to the linguistic variable "Young" with a degree of membership 0.8 and to the linguistic variable "Middle" with a degree of membership 0.3. If we say "Chloe is 45" that means that she belongs to the linguistic variable "Old" with a DOM 0.4 and to "Middle" with a degree of membership 0.6. On a crisp set the values 28 and 45 would belong fully to just one set, either young, middle or old.

1.2.6. The Mamdani Fuzzy Inference System

Fuzzy inference can be defined as a method for interpreting the values in an input vector and assigning values to the output vector based on a set of rules. During this process the inference system formulates the mapping from a given input to an output. [7]

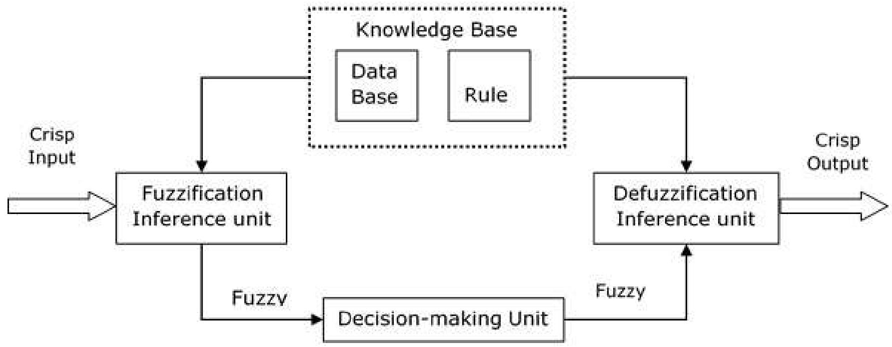


Figure 4. – The structure of a Fuzzy Inference System

The following sections will show all the steps needed to calculate an output.

1.2.6.1. Creating Fuzzy Rules

Fuzzy rules are a sequence of linguistic statements that describe how the FIS should regulate an input. Linguistic variables are a standard in fuzzy inference system. A fuzzy rule, known as an IF-THEN rule, is structured as the following [8]:

IF [variable1 is membership of function 1**] AND/OR [**variable1 is membership of function 1**]**

THEN [*$output_n$ is output membership $function_n$***]**

Examples of fuzzy rules would be:

if road is filled with snow ***then*** driving is dangerous

if service was below average ***and*** food was good ***then*** tip a little

1.2.6.2. Fuzzification

The process of fuzzification converts a real time crisp input into a fuzzy function with linguistic variables by determining the degree of membership, which as previously mentioned will be a number between 0 and 1. There are 6 main membership functions; trapezoidal, triangular, s-shape, z-shape, Gaussian and sigmoid. [9]

1.2.6.3. Knowledge Base

In the fuzzy inference system knowledge base represents a set of IF-then rules and linguistic that calculate the output based on the principles of fuzzy set theory. A database and a rule base comprise the knowledge base.

1.2.6.4. Fuzzy combinations (T-norms)

On section 6.1 we described creating fuzzy rules and the concept of using "and", "or" and less occasionally "or". The following section will show how we deal with fuzzy logic operator, which are analogous to their binary counterparts. [10]

Table 1 – Fuzzy Logic operators

Fuzzy Operator	Description
OR	In order to calculate the fuzzy logic operator "or" we will use the Zadeh technique which is the most frequently used technique. This methodology calculates the fuzzy "or" by taking the *maximum of two or more variables*. If we try, we can see that the fuzzy "or" can be used to compute binary "or".
AND	Contrary to the fuzzy "or", to calculate the fuzzy "and" we will take the *minimum of two or more*

	variables, a method that once again could be just as useful while dealing with binary "and".
NOT	The fuzzy "not" is equal to 1 minus the input variable.

1.2.6.5. Calculating the consequence of each fuzzy rule

In order to calculate the consequence of each fuzzy rule we compute the rule strength by using the T-norms mentioned on section 6.4 and we also clip the output membership function at the rule strength.

1.2.6.6. Output distribution

In order to obtain the fuzzy output distribution, we combine all the outputs of each fuzzy rule that we calculated at step 1.2.6.5.

1.2.6.7. Defuzzification

In several instances, including ours, we want a single crisp value as an output from our fuzzy interface system. The process of calculating a crisp value is called defuzzification and the methodology that we are going to apply is the "centroid" defuzzification technique which returns as an output the center of gravity of the output distribution that we calculated in 1.2.6.6. [11]

On the example shown on Figure 5 below, a Mamdani FIS is built to deicde how much we should tip on a restaurant. The image below shows all the steps that we explained above.

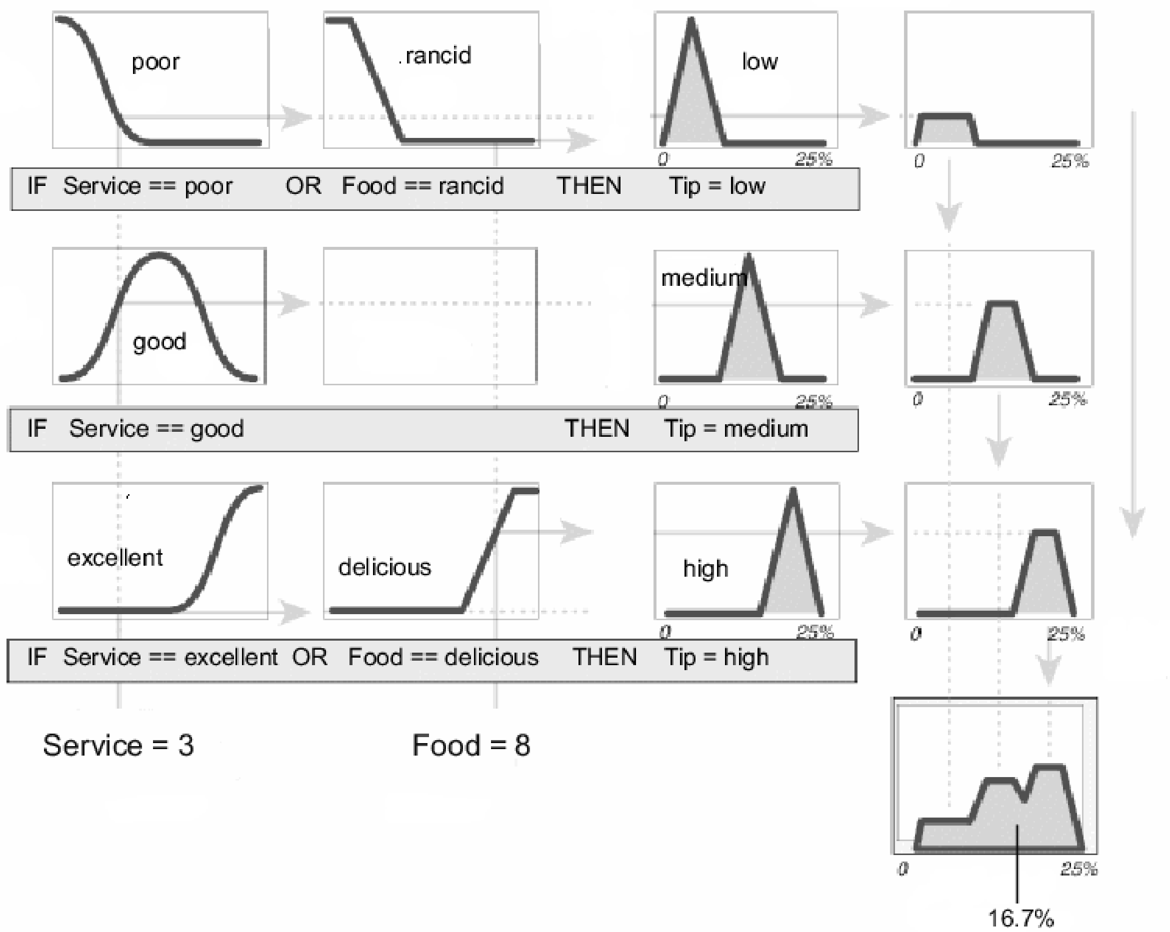


Figure 5. - Example of a Mamdani fuzzy interface system

1.3. Related Literature

Fuzzy logic has proven to be a promising and useful tool in several areas in business. Rico et al. [12] suggested that fuzzy logic can be applied in five main business areas including management control and measuring financial performance which requires risk calculation techniques.

In a research done by Ginting et al. [13] they use the Fuzzy Tsukamoto method, which uses monotonical membership functions, to model a system that predicts creditworthiness. After analysing the Bandung City Bank loan database, a 90% accuracy was obtained after comparing the decision made by the Fuzzy Inference System and the eligibility decision made by the actual bank.Another important reading in creation of this study was Abdulrahman et al. [14] which proposed that an in-built Fuzzy Logic system could be a good choice to model systems that mimic human behaviour, however considering this is a financial institution that is directly affected by changes in the economy the variables and the if-then fuzzy rules should be carefully selected so that the efficiency of the calculations doesn’t fade over time.

Darwish et al. [15] proved fuzzy logic techniques to be one of the most reliable and manageable ways to determine credit risk. In addition, this paper also provided an amazing insight at financial indicators used in a fuzzy inference system. Lastly, the need arises to create standard fuzzy rules and improved linguistic variables that can be implemented on every credit score prediction model. Sampath et al. [16] also suggested that due to the susceptivity of the topic, implementing scenarios that are generated by analysing historical data could obtain high accuracy. In the future, it is predicted that similar models will be created by using MCDM models that include fuzzy VIKOR, fuzzy ELECTRE and fuzzy PROMETHEE.

CHAPTER 2

METHODOLOGY

2.1. Materials

The dataset that I'm going to use while working on this thesis is created by Dream Housing Finance and is available on Kaggle [17]. After obtaining the data I did some changes by removing and adding columns. Firstly, I regenerated new Loan IDs for the first 166 records and also deleted the "Gender" column since it wasn't relevant to my work on this thesis. In addition, I added two new columns, the first one is a column that represents the ratio between the monthly debt that the applicant will owe to the debt, if the bank decides to hand the loan, and the applicant's monthly income. The same thing was done with the co-applicant if there is one. Knowing that the same amount of debt can have a different impact in two different people with different salaries, these columns will help us better visualize how the loan will affect each applicant's monthly budget. For my FCM I had to separate our dtaset in 2 parts, one that holds the information about applicant who were given the loan and the other about the ones who weren't given the loan

I am using Python programming language to create the Fuzzy Inference System. Since I have decided to work with Mamdani approach, the skfuzzy library in Python has that technique already built in.\

2.2. Method

Based on the content of the table which includes information about the applicants, a fuzzy inference system will be built.

Key information regarding the dataset:

Table 2 – Dataset Information

Key Name	Description
Loan ID	Unique Loan ID
Married	Is applicant married (1 for Yes, 0 for No)
Dependents	Number of dependents (0 for none, 1 for one dependent, 2 for two dependents, 3 for three or more dependents)
Education	Applicant's education. (1 for Graduate / 0 for Not Graduate)
Self Employed	Is applicant self-employed (1 for Yes, 0 for No)
Applicant Income	Applicant's income
Co-applicant Income	Co-applicant's income.
Loan Amount	Loan amount (in thousands)
Loan Amount Term	Loan term in months
Credit History	Whether credit history meets guidelines. (1 if it does, 0 if it doesn't)
Property Area	The area where the property is. (1 for rural, 2 for Semi-urban, 3 for Urban)
Monthly Payment / Applicant Income	Represent the ratio between the debt that the applicant has to pay to the bank and applicant's monthly income.
Monthly Payment / Co-applicant Income	Represent the ratio between the debt that the applicant has to pay to the bank and co-applicant's monthly income. (1 means that co-applicant has no income)
Loan Status	If the loan was granted or no. (1 for Yes, 0 for No)

The last column (highlighted in yellow) is the one we want to predict. After I build the Mamdani Fuzzy Inference System I will compare the Loan Status from the dataset with the results I obtain from the FIS. There are twelve factors that I will be using during the process of creating my fuzzy rules, shown in the table below:

Table 3 – Factors used in FIS

No.	**Factor**	
1.	*f_1 = Marital Status*	
2.	*f_2 = Dependents*	
3.	*f3 = Level of Education*	
4.	*f4 = Self-employed*	
5.	*f5 = Applicant Income*	
6.	*f6 = Co-applicant income*	
7.	*f7 = Loan Amount*	
8.	*f8 = Loan Amount Term*	
9.	*f9 = Credit History*	
10.	*f10 = Property Area*	
11.	*f11 = Monthly Payment / AI*	Monthly Payment = f7 * 1000 / f8 f11 = Monthly Payment / f5
12.	*f12= Monthly Payment / CAI*	Monthly Payment = f7 * 1000 / f8 f12 = Monthly Payment / f6

2.3. Linguistic variables and membership functions

As we can see from the dataset f1, f2, f3, f4, f9, f10 are crisp sets and their membership functions be linear while f5, f6, f7, f8, f11, f12 will be fuzzy sets with triangular membership functions. In the following tables the set of values for the crisp and the linguistic variables for the fuzzy sets will be displayed.

Table 4 - Set of values for *f1*

f1: Marital Status	
Married	**Not married**
1	0

Table 5 - Set of values for *f2*

f2: Dependents		
One	**Two**	**Three or more**
1	2	3

Table 6 – Set of values for *f3*

f3: Level of Education	
Graduate	**Non-Graduate**
1	0

Table 7- Set of values for *f4*

f4: Self-employed	
Yes	**No**
1	0

Table 8 – Set of values for *f9*

f9: Credit History	
Meets guidelines	**Doesn't meet guidelines**
1	0

Table 9 – Set of values for *f10*

f10: Property Area		
Rural	**Semi-urban**	**Urban**
1	2	3

The values of *f5* range from \$1000 to \$39999 with an average value of \$5376.

Table 10 – Set of values for *f5*

f5: Applicant Income			
Low	**Medium**	**High**	**Very High**
\$2000 and lower	\$3000 and lower	\$4800 and lower	\$39999 and lower

The values for *f6* range from \$0 to \$11300 with an average value of \$1614.

Table 11 – Set of values for *f6*

f6: Co-applicant Income			
Low	**Medium**	**High**	**Very High**
\$1100 and lower	\$1750 and lower	\$3200 and lower	\$11300 and lower

Table 12 – Set of values for *f7*

f7: Loan Amount			
Low	**Medium**	**High**	**Very High**
\$90000 and lower	\$150000 and lower	\$200000 and lower	\$600000 and lower

The values for f8 range from 60 months to 480 months with an average of 390 months.

Table 13 – Set of values for *f8*

f8: Loan Amount Term		
Short	**Medium**	**Long**
100 and lower	270 and lower	480 and lower

For f11 and f12 the lower the value the better and safer choice it is for the bank. However, for the f12, 0 doesn't mean the best (and it mathematically it isn't an accepted value), it means that the co-applicant doesn't have any income.

The values for *f11* range from 0.0085 to 0.3513 with an average of 0.1.

Table 14 – Set of values for *f11*

f11: Monthly Payment / AI			
Low	**Medium**	**High**	**Very High**
0.065 and lower (excluding 0)	0.085 and lower	0.15 and lower	0.3513 and lower

The values for *f12* range from 0.0423 to 0.555 with an average of 0.1806.

Table 15 – Set of values for *f12*

f12: Monthly Payment / CAI			
Low	**Medium**	**High**	**Very High**
0.085 and lower (excluding 0)	0.12 and lower	0.18 and lower	0.555 and lower

Table 16 – Set of values for *cr*

Credit risk			
Low	**Medium**	**High**	**Very High**
25% and lower	50% and lower	75% and lower	100% and lower

Membership function for *f1: Marital status*, since this is a crisp input, we are only interested in values 0 and 1, and as we can see for input 0 the membership degrees are

0.0 for 'Married' and 1.0 for 'Not married' and vice versa. We don't care about any values in between.

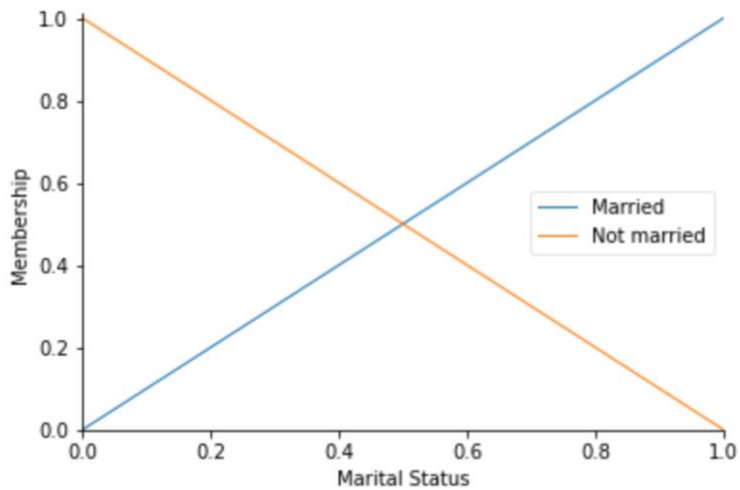


Figure 6. - Membership function for *f1*

Membership function for *f2: Dependents.* Once again this is a crisp so we follow the same rules. For *f2* there are only 3 possible inputs (1, 2 and 3) which if we check the function that is fully represented. The input value 1 has a membership degree of 1.0 for "One" and 0.0 for the others, the input value 2 has a membership degree of 1.0 for "Two" and 0.0 for the other linguistic variables while the input value 3 has a degree of 1.0 for the linguistic variable "Three or more" and 0.0 for the others.

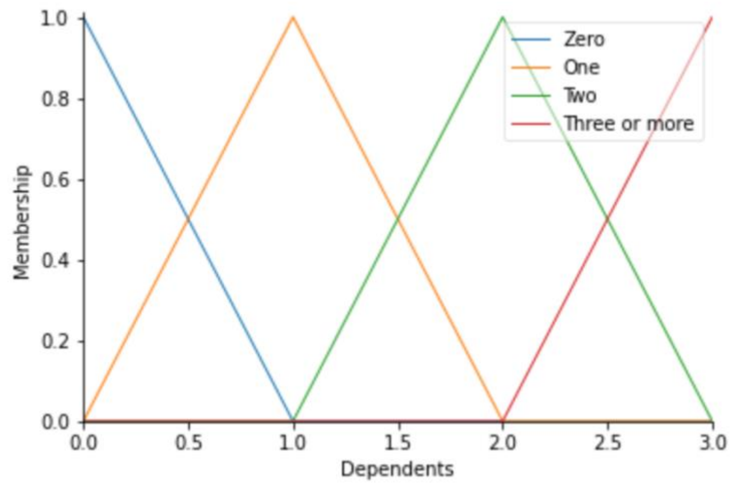


Figure 7. – Membership function for *f2*

The same rules and mindset will be applied for f3, f4, f9, f10 since all of them are crisp inputs.

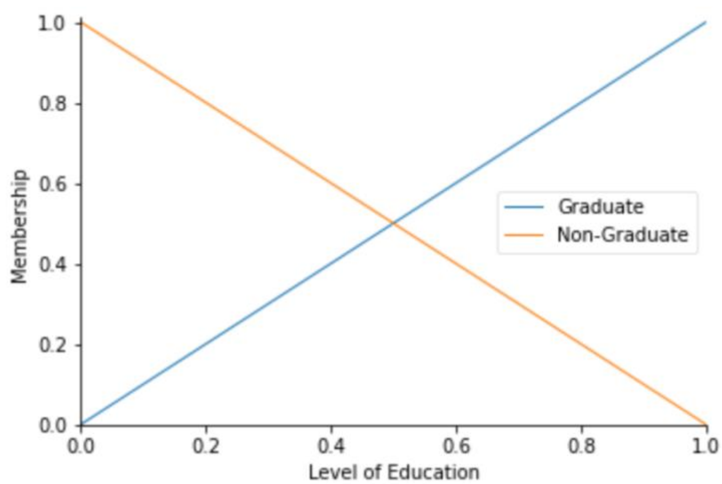


Figure 8. – Membership function for *f3*

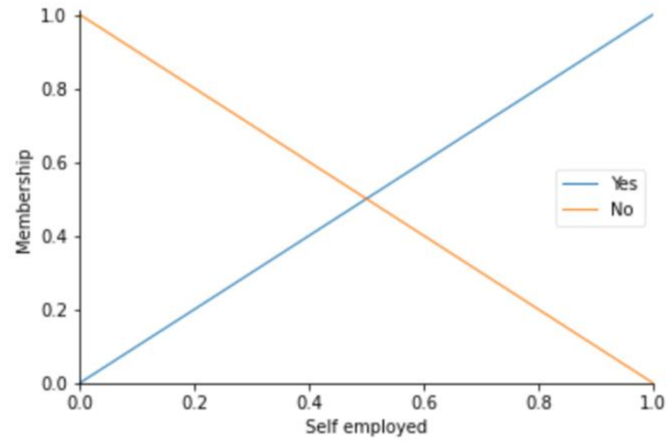


Figure 9. – Membership function for *f4*

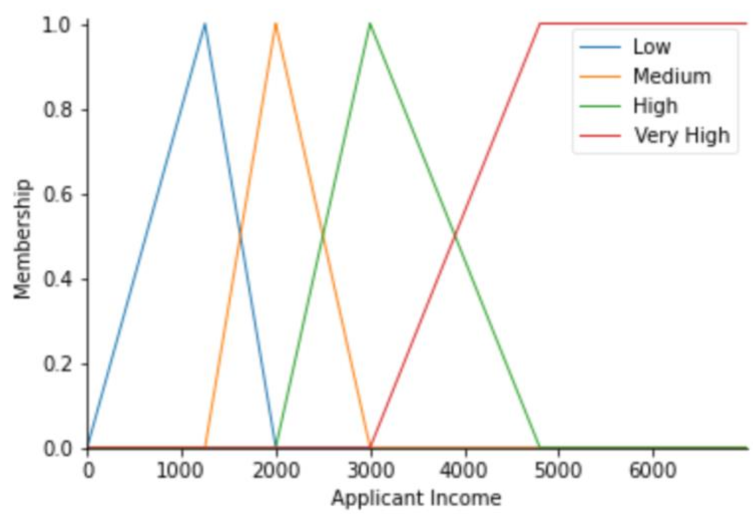


Figure 10. – Membership function for *f5*

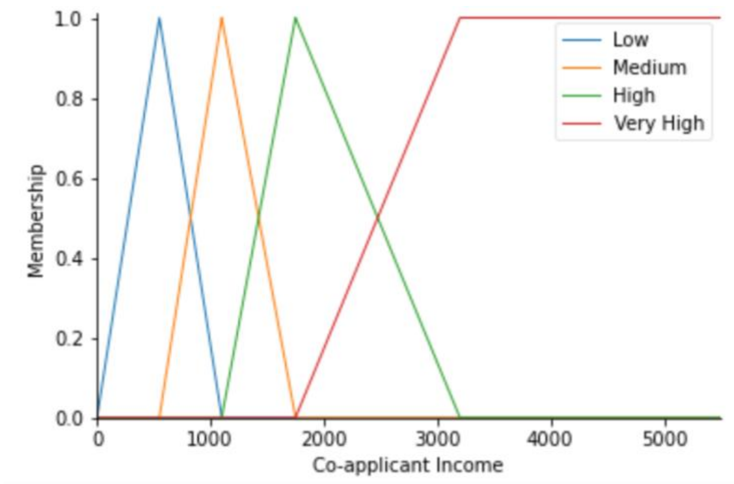


Figure 11. – Membership function for *f6*

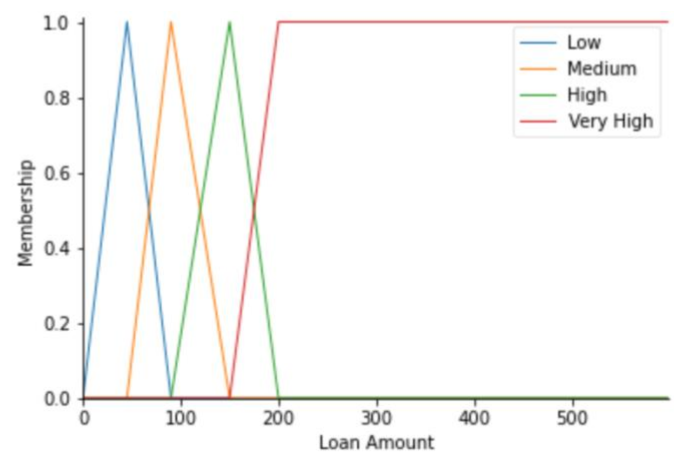


Figure 12. – Membership function for *f7* (in thousands)

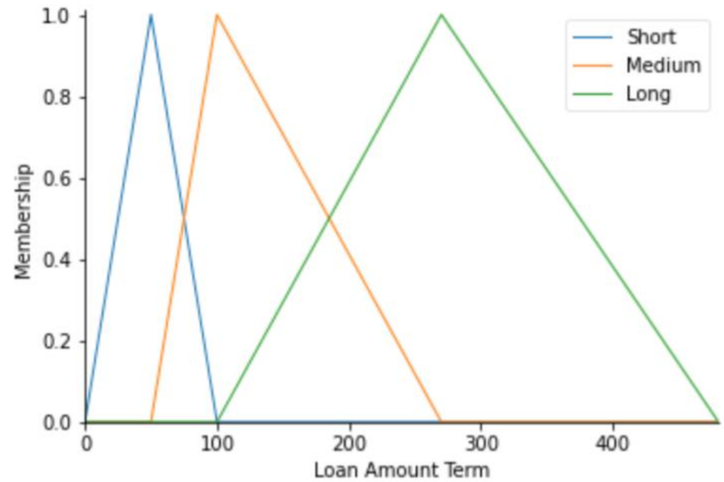


Figure 13. – Membership function for *f8*

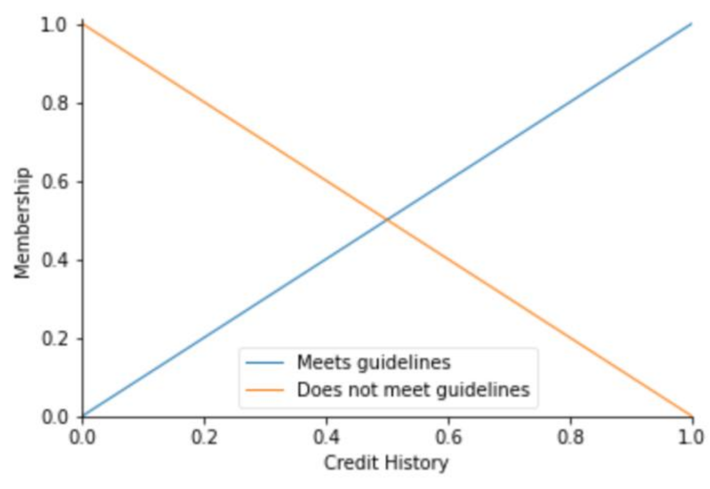


Figure 14. – Membership function for *f9*

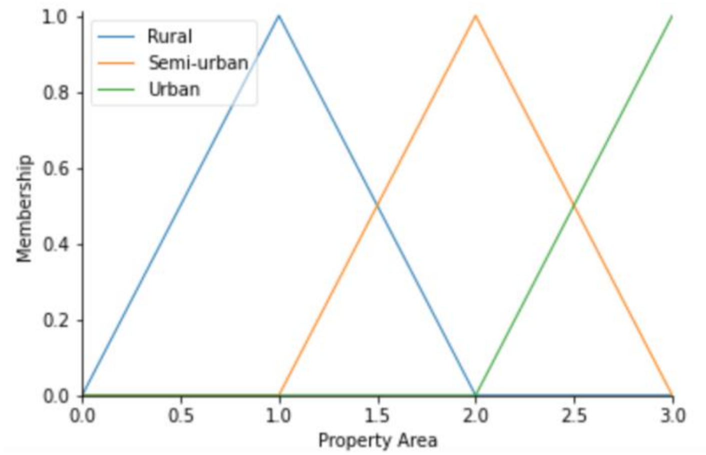


Figure 15. – Membership function for *f10*

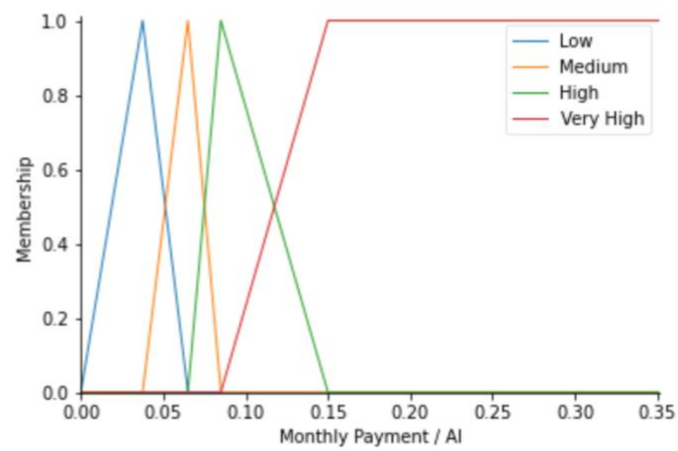


Figure 16. – Membership function for *f11*

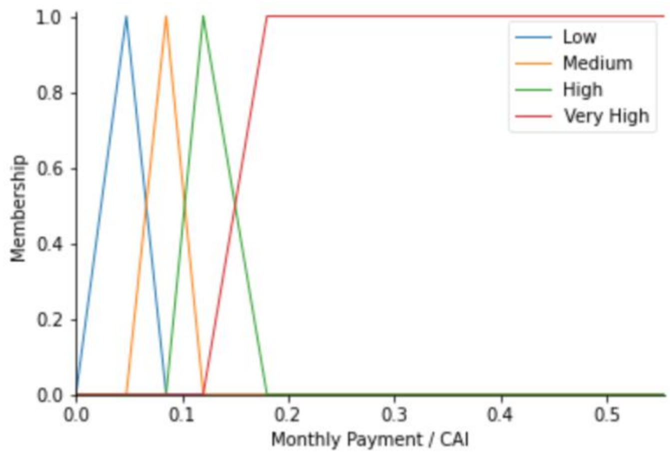


Figure 17. – Membership function for *f12*

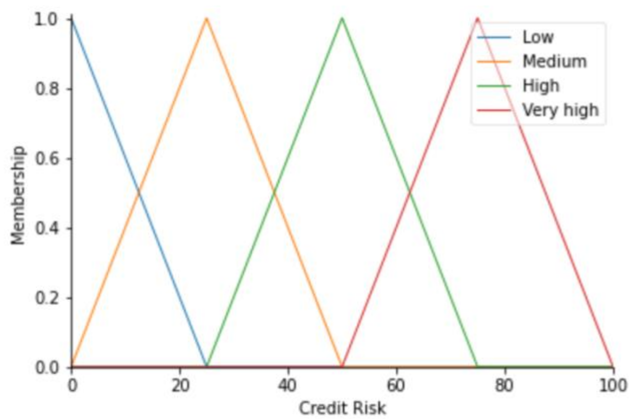


Figure 18. – Membership function for *Credit Risk*

2.3. Fuzzy Rules

For this particular study, there could be a total of 8^{12} (approximately 68719476736) where 8 if the number of linguistic variables and 12 is the total number of input variables. An example of how a fuzzy rule is written is shown in the example below.

IF *f1 = 'Married'* **AND** *f2 = 'Zero '* **AND** *f3 = 'Graduate'* **AND** *f4 = 'Yes '* **AND** *f5 = 'High'* **AND** *f6 = 'Low'* **AND** *f7 = 'Low'* **AND** *f8 = 'Long'* **AND** *f9 'Meets guidelines'* **AND** *f10 = 'Urban'* **AND** *f11 = 'Low'* **AND** *f12 = 'Very High'*
THEN *cr = 'Medium'*

The value *cr* represents the credit risk which is the crisp value output. Based on the cr value the bank will decide if the loan should be given or not.

2.4. Implementing the FIS and FCM using *skfuzzy* library

The twelve factors that we defined above will be hold by the antecedent meanwhile cr which is our crisp output value will be the consequent. An example of the antecedent and consequent is shown in listing 1.

```
f12= ctrl.Antecedent(np.arange(0, 0.555, 0.0001), 'Monthly Payment / CAI')

cr = ctrl.Consequent(np.arange(0, 101, 1), 'Credit Risk')
```

Listing 1 – Antecedent and consequent

For each of our rules we had to create a membership function as shown above. In order to do so I used a mix of triangular and trapezoidal functions, at some only one and some both as shown in listing 2. A triangular function is represented by *"fuzz.trimf"* meanwhile a trapezoidal one by the *"fuzz.trapmf"*

```
f11['Low'] = fuzz.trimf(f11.universe, [0, 0.0375, 0.065])

f11['Medium'] = fuzz.trimf(f11.universe, [0.0375, 0.065, 0.085])

f11['High'] = fuzz.trimf(f11.universe, [0.065, 0.085, 0.15])

f11['Very High'] = fuzz.trapmf(f11.universe, [0.085, 0.15, 0.3513, 0.3513])
```

Listing 2 – Creating membership functions

The "automf" function is used to assign our custom linguistic variables into our factors and after we decide on a set of rules that are going to best represent our case we use "output.input" to input values and then "output.compute()" to calculate the crisp value output. To calculate the FCM functions for our case I used the "fcmeans", "matplotlib", "seaborn", "pandas" and "numpy" libraries. After reading the dataset and and creating two dimensional arrays that hold the values of loan amount in one column and the other factors in the others, we had to calculate the centers and perform the clustering as shown in the listing below:

```
centers = fcm.centers
labels  = fcm.u.argmax(axis = 1)
f, axes = plt.subplots(1, 2)
scatter(x3[:,0], x3[:,1], ax = axes[0])
scatter(x3[:,0], x3[:,1], ax = axes[1], hue = labels)
scatter(centers[:,0], centers[:,1], ax = axes[1], marker = "s", color = 'r', s = 35)
f.suptitle('Relationship between Age and Organizational Culture', size = 10)
for ax in axes.flat:
   ax.set(xlabel = 'Applicant Income', ylabel = 'Loan Amount')
```

Listing 3 – FCM implementation

2.5. Fuzzy C-means clustering

For this project, in order to make clustering as useful as possible I sperated the dataset in two parts. The first one holds information about the applicants whose loans were approved, the second one holds information about the applicants whose loans were not approved. For each of these datasets I am going to perform the same clustering process, where I am going to look at the relationship between seven different factors and loan amount and compare them with one another to see how certain factors effect a person’s loan eligbility. I sperated the loan amount in three clusters: 0-100, 101-200, 201-400.

CHAPTER 3

RESULTS AND DISCUSSION

3.1. Fuzzy Inference System

After creating the FIS I created a total set of of four rules in order to test it. The rules go as shown below:

Rule 1:

IF *f1 = 'Married'* **AND** *f2 = 'Zero '* **AND** *f3 = 'Graduate'* **AND** *f4 = 'Yes '* **AND** *f5 = 'High'* **AND** *f6 = 'Low'* **AND** *f7 = 'Low'* **AND** *f8 = 'Long'* **AND** *f9 = 'Meets guidelines'* **AND** *f10 = 'Urban'* **AND** *f11 = 'Low'* **AND** *f12 = 'Very High'* **THEN** *cr = 'Medium'*

Rule 2:

IF *f1 = 'Not married'* **AND** *f2 = 'Zero '* **AND** *f3 = 'Graduate'* **AND** *f4 = 'No '* **AND** *f5 = 'High'* **AND** *f6 = 'Low'* **AND** *f7 = 'Medium'* **AND** *f8 = 'Long'* **AND** *f9 = 'Does not meet guidelines'* **AND** *f10 = 'Semi-urban'* **AND** *f11 = 'High'* **AND** *f12 = 'Very High'* **THEN** *cr = 'High'*

Rule 3:

IF *f1 = 'Married'* **AND** *f2 = 'One '* **AND** *f3 = 'Graduate'* **AND** *f4 = 'No '* **AND** *f5 = 'Very High'* **AND** *f6 = 'High'* **AND** *f7 = 'High'* **AND** *f8 = 'Long'* **AND** *f9 = 'Meets guidelines'* **AND** *f10 = 'Urban'* **AND** *f11 = 'Medium'* **AND** *f12 = 'Very High'* **THEN** *cr = 'Low'*

Rule 4:

IF *f1 = 'Not married'* **AND** *f2 = 'Zero '* **AND** *f3 = 'Graduate'* **AND** *f4 = 'No '* **AND** *f5 = 'High'* **AND** *f6 = 'Very High'* **AND** *f7 = 'High'* **AND** *f8 = 'Long'* **AND** *f9 = 'Meets guidelines'* **AND** *f10 = 'Urban'* **AND** *f11 = 'Very High'* **AND** *f12 = 'High'* **THEN** *cr = 'Very High'*

Each defined rules vizulazed below:

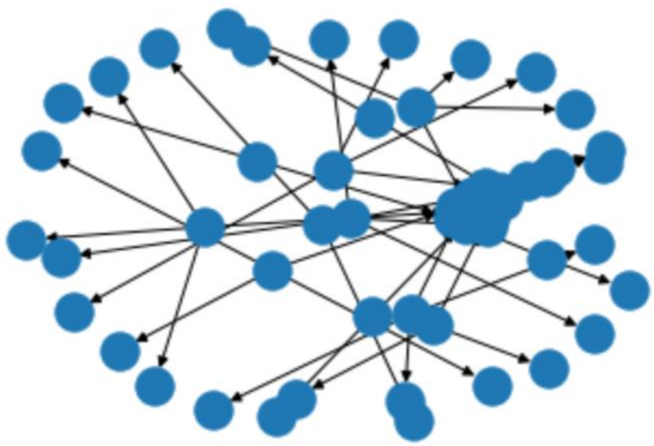


Figure 19. – Rule 1

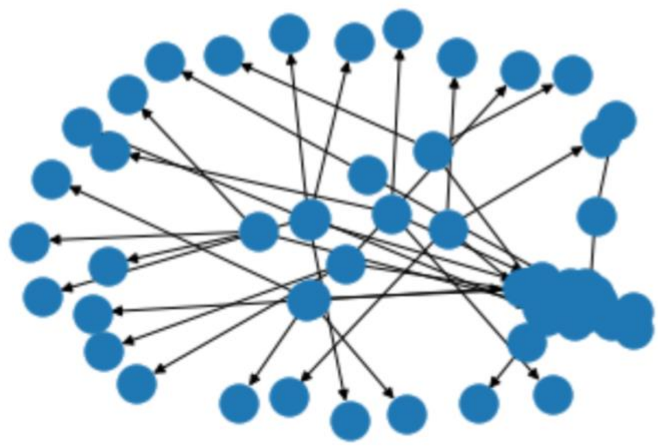


Figure 20. – Rule 2

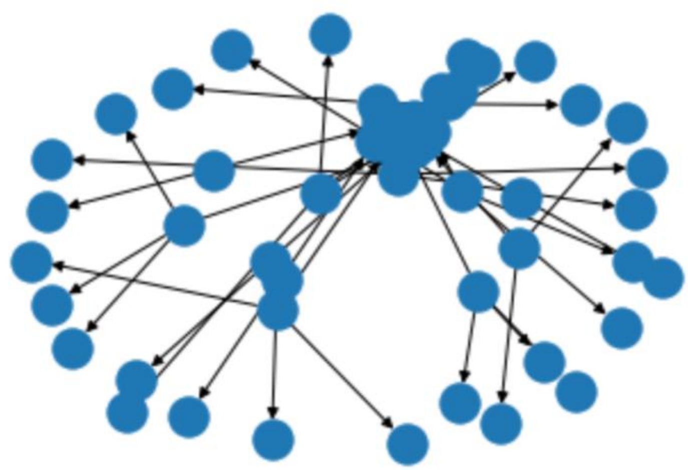


Figure 21. – Rule 3

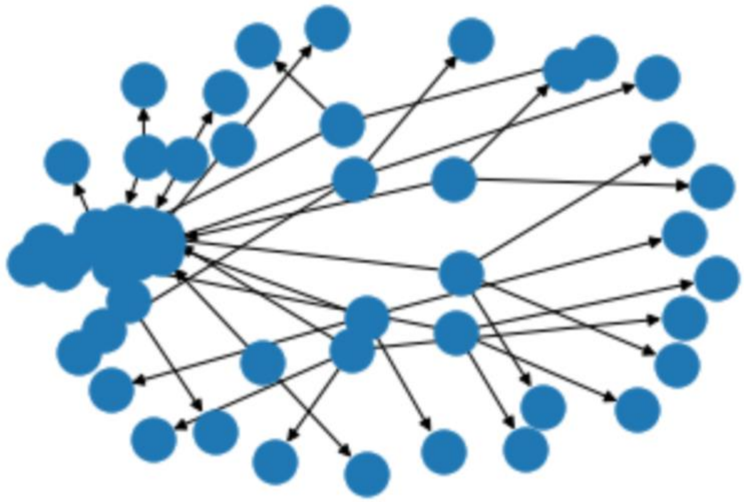


Figure 22. – Rule 4

In order to see how efficient our FIS is, I had to test it by using the information in our dataset. Particularly the information obtained by the applicants with Loan ID 1, 51, 45 and 61. Their input values and the cr output are displayed in the table below:

Table 17 – Input sets to test our fuzzy rules

INPUT												OUTPUT
f1	*f2*	*f3*	*f4*	*f5*	*f6*	*f7*	*f8*	*f9*	*f10*	*f11*	*f12*	*cr*
1	1	1	0	8080	2250	180	360	1	3	0.06188	0.2222	10.2083
1	0	1	1	3000	0	66	360	1	3	0.0611	1	25.004
0	0	1	0	4166	0	116	360	0	2	0.077345709	1	50.0036
0	0	1	0	3750	4750	176	360	1	3	0.1303	0.1209	74.999999

After I assigned the input values to test each rule, I obtained the credit risk. For credit risk lower than 50 the bank will give the applicant the loan and for credit risk higher htan 50 the bank will not. These outputs predict perfectly the last column showing that A

FIS can be quite a useful tool in modelling bank eligibility systems in a finance institution. The graph area of cr for each output is shown in the following figres.

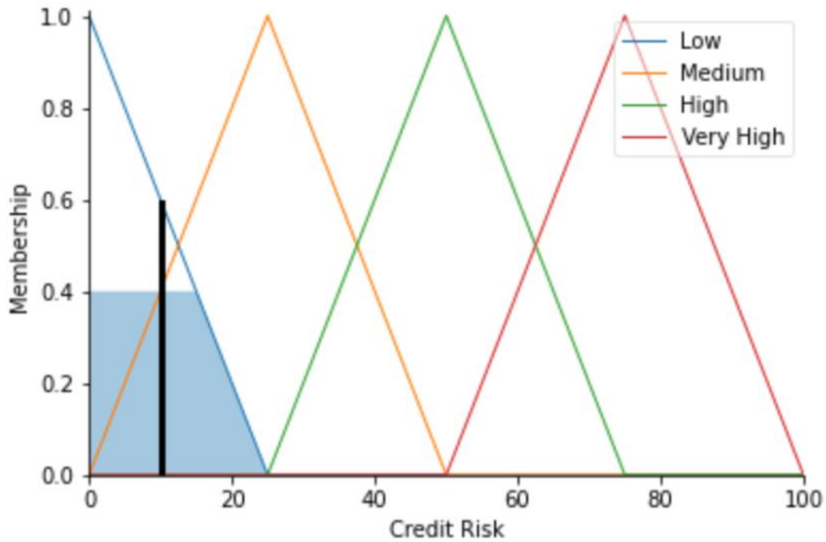


Figure 23. – Graph area for rule 3

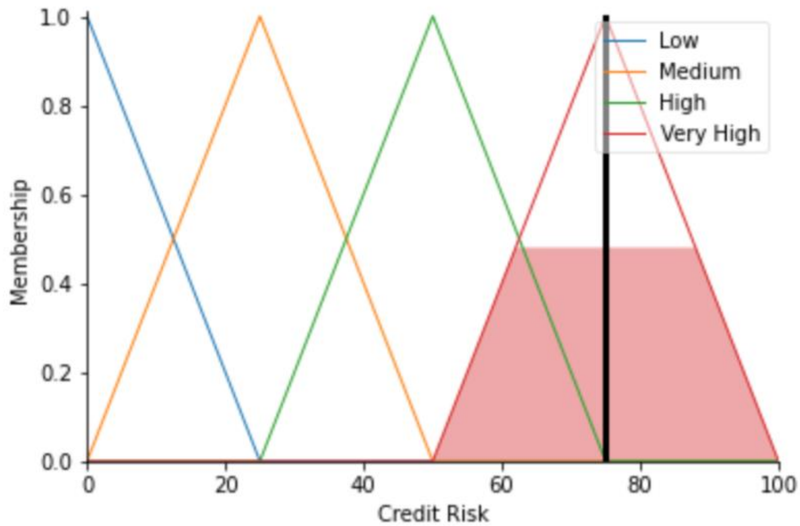


Figure 24. – Graph area for rule 4

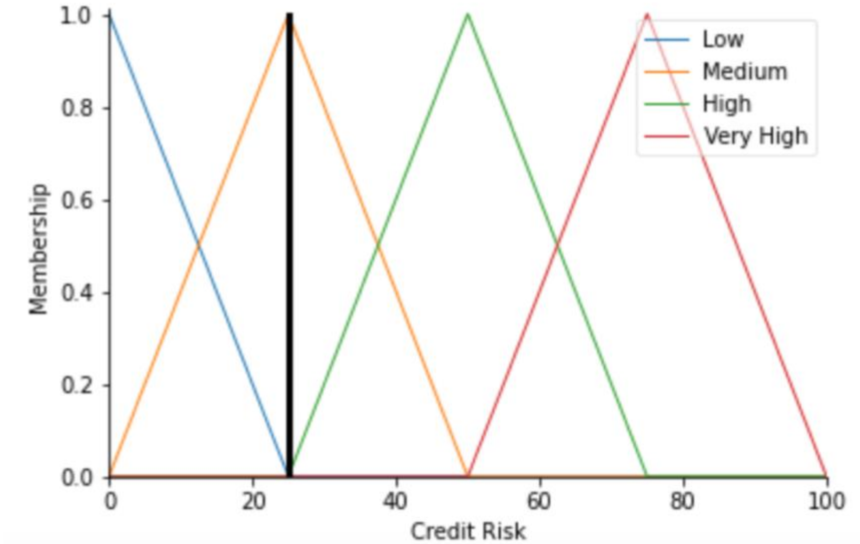


Figure 25. – Graph area for rule 1

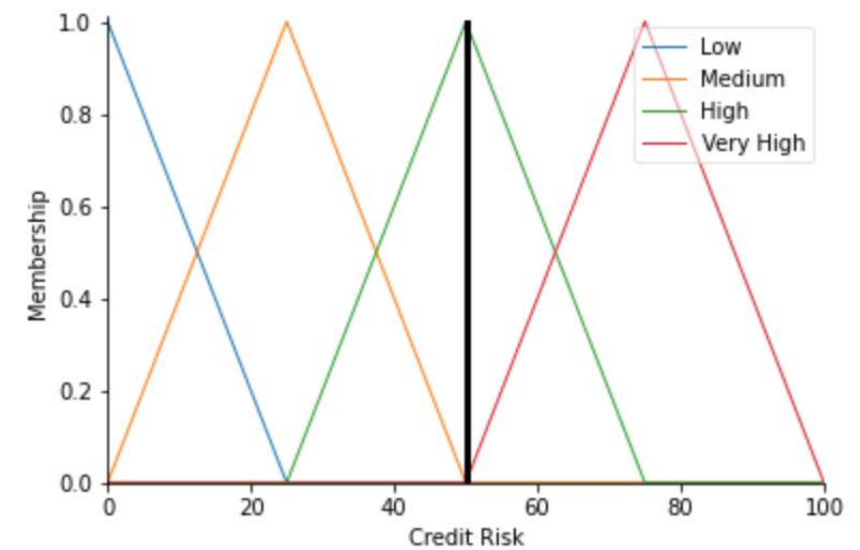


Figure 26. – Graph area for rule 2

3.2. Fuzzy C-means clustering

After dividing our dataset in two mini datasets and performing the FCM algorithm in both of them, the following figures show the results. Each of these graphs shows the relationship between one of the factors and the loan amount for each of the datasets.

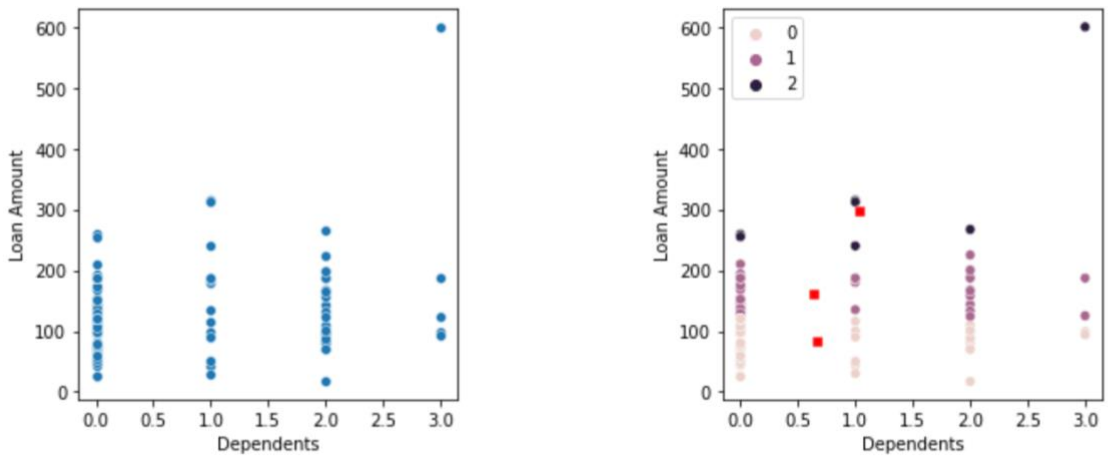


Figure 27 – Relationship between Dependents and Loan Amount; loan eligble applicants

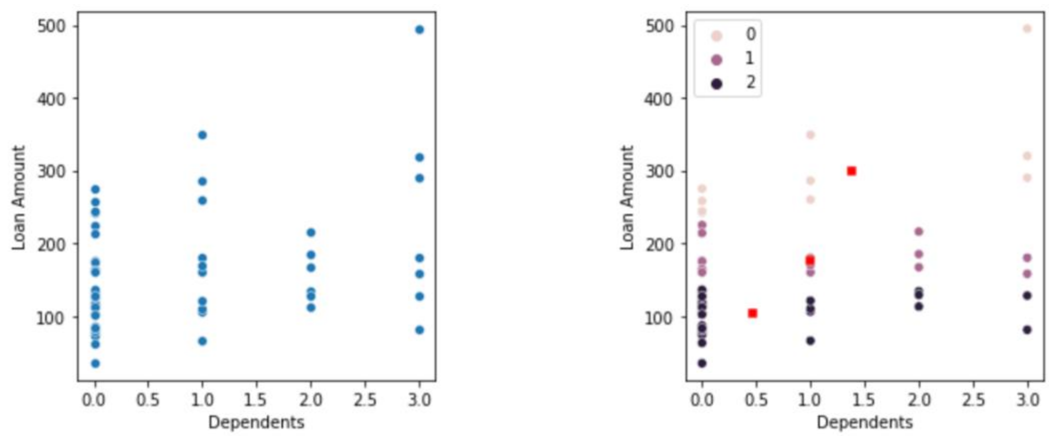


Figure 28 – Relationship between Dependents and Loan Amount; applicants not eligble for a loan

If we compare figure 25 and 26 we notice that for eligble applicants that have applied for a loan between the amount 0 to 100 we notice that the cluster center is a more on the right, for the amount 101 to 200 is pretty much the same while for the amount 201 to 300 the cluster center is more on the left for applicants who are not eligble. This lets us know that the number of dependents has a limited influence to the decision made by the bank.

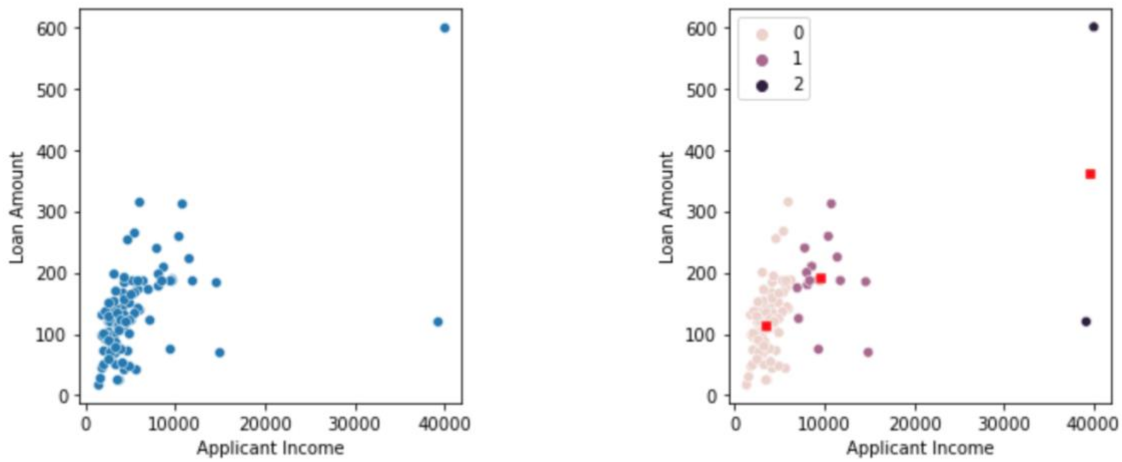


Figure 29 – Relationship between Applicant Income and Loan Amount; eligble applicants

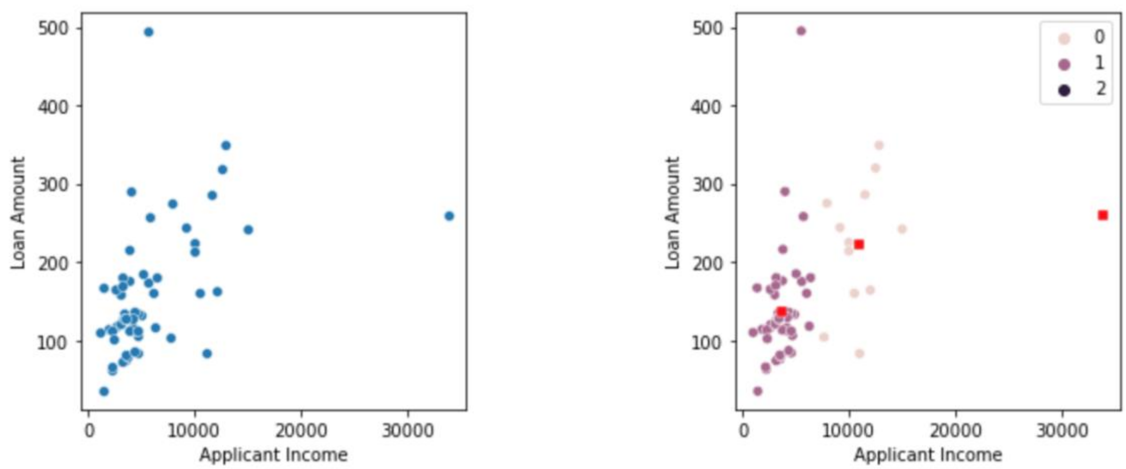


Figure 30 – Relationship between Applicant Income and Loan Amount; applicant not eligble for a loan

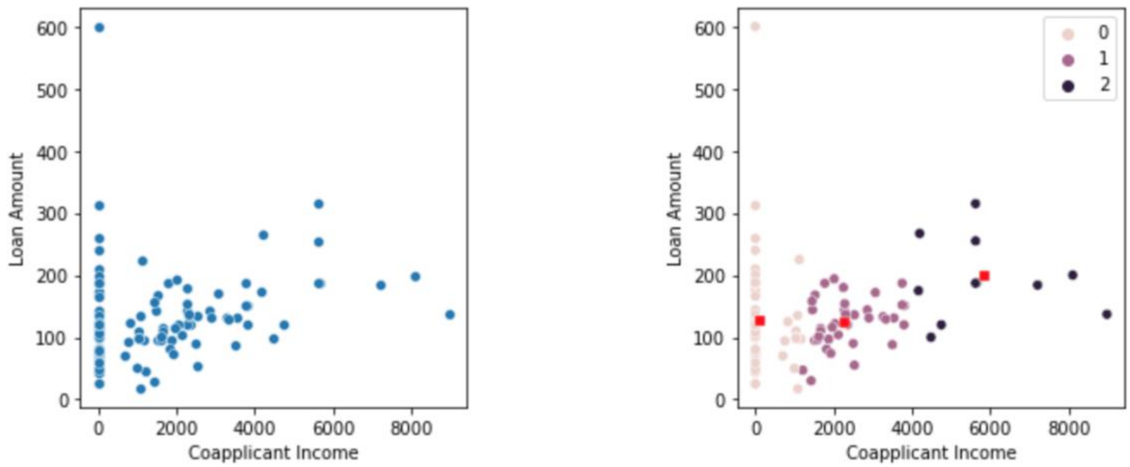


Figure 31 – Relationship between Copplicant Income and Loan Amount; eligble applicants

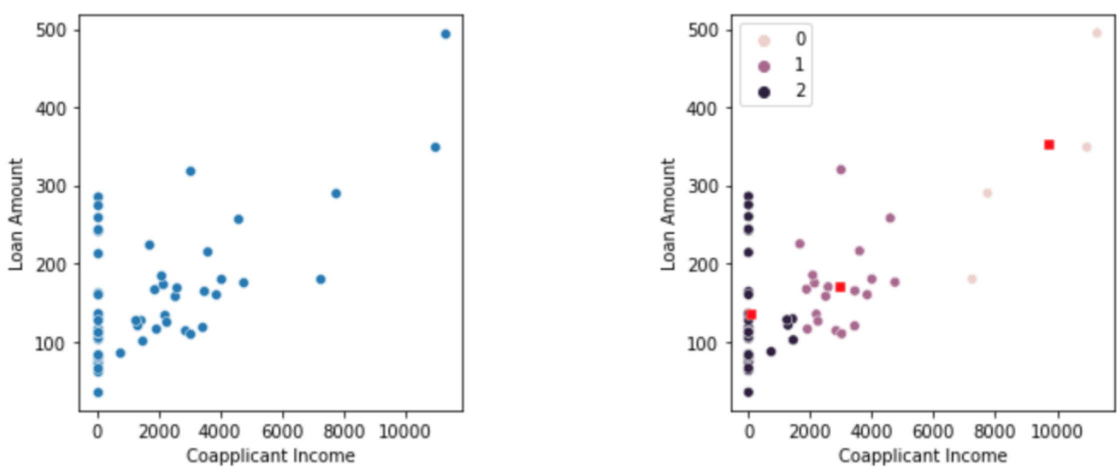


Figure 32 – Relationship between Copplicant Income and Loan Amount; applicants not eligble for a loan

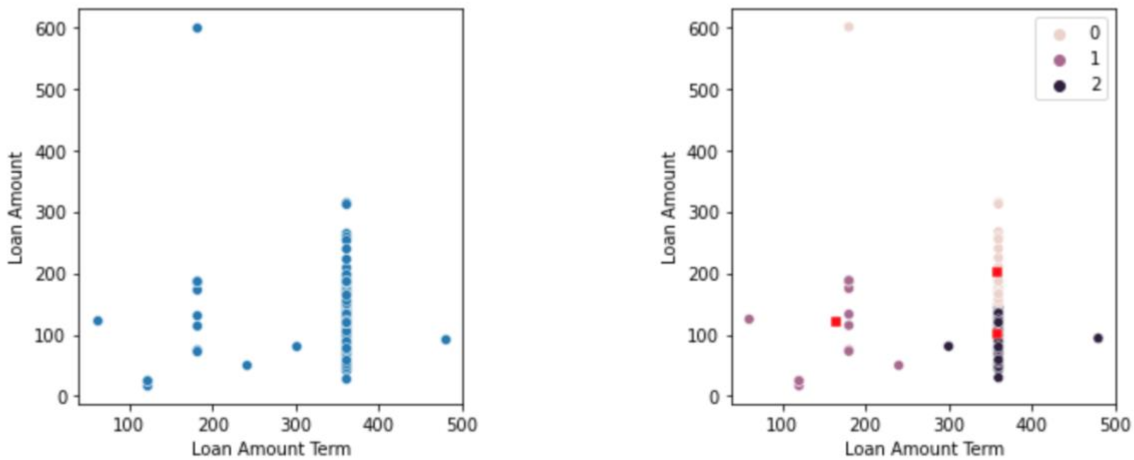


Figure 33 – Relationship between Loan Amount Term and Loan Amount; eligble applicants

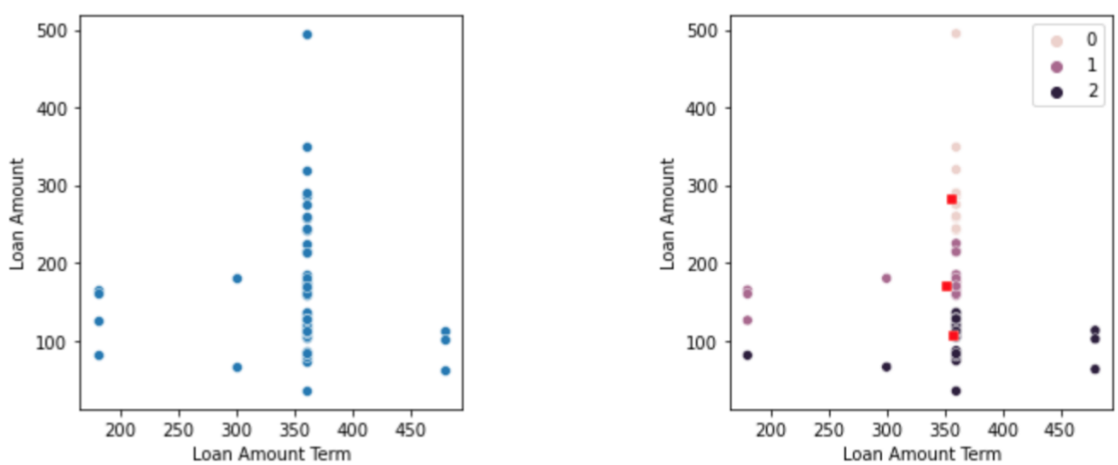


Figure 34 – Relationship between Loan Amount Term and Loan Amount; applicants not eligble for a loan

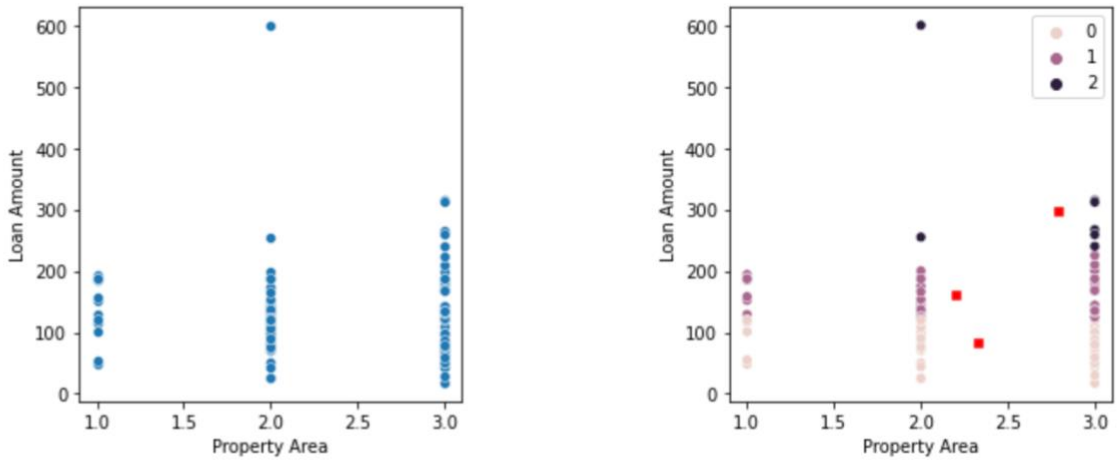


Figure 35 – Relationship between Property Area and Loan Amount; eligble applicants

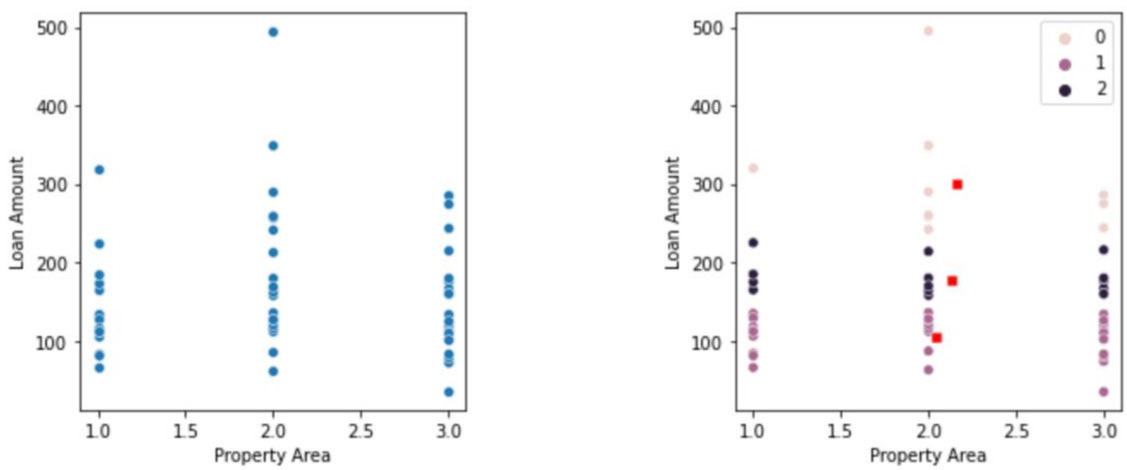


Figure 36 – Relationship between Property Area and Loan Amount; applicants not eligble for a loan

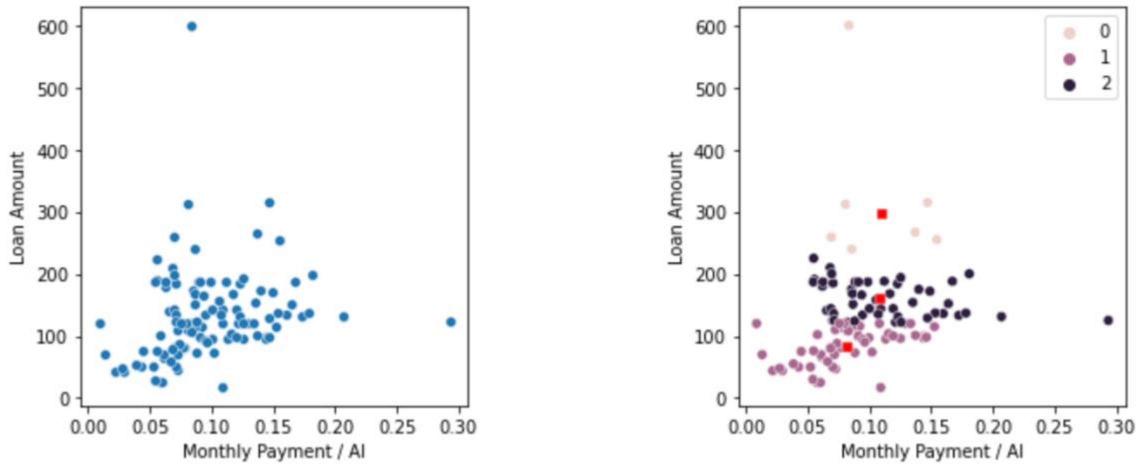


Figure 37 – Relationship between Monthly Payment / AI and Loan Amount; eligble applicants

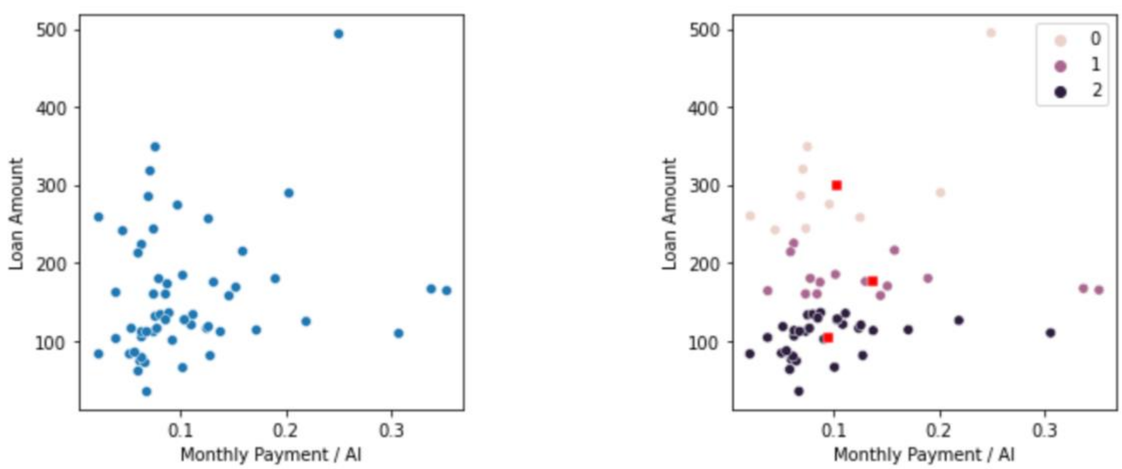


Figure 38 – Relationship between Monthly Payment / AI and Loan Amount; applicants not eligble for a loan

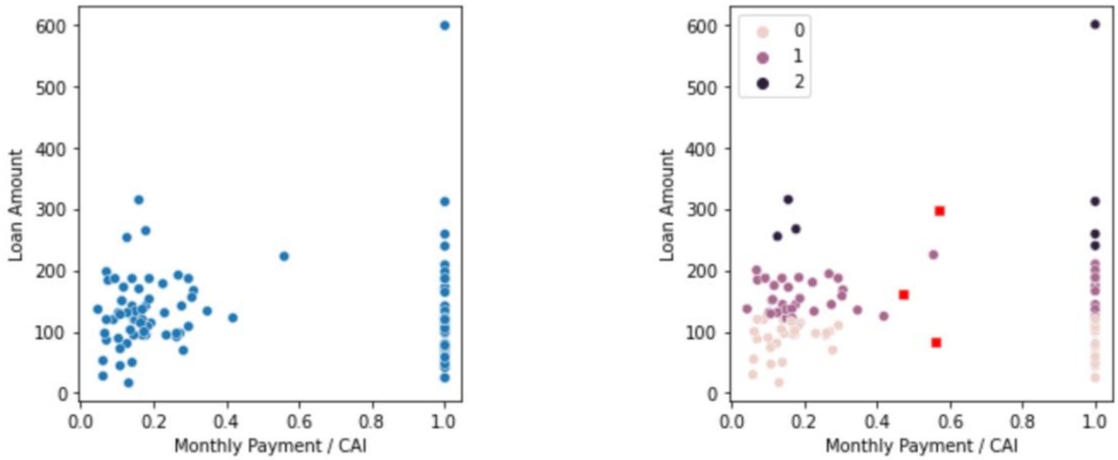


Figure 39 – Relationship between Monthly Payment / CAI and Loan Amount; eligble applicants

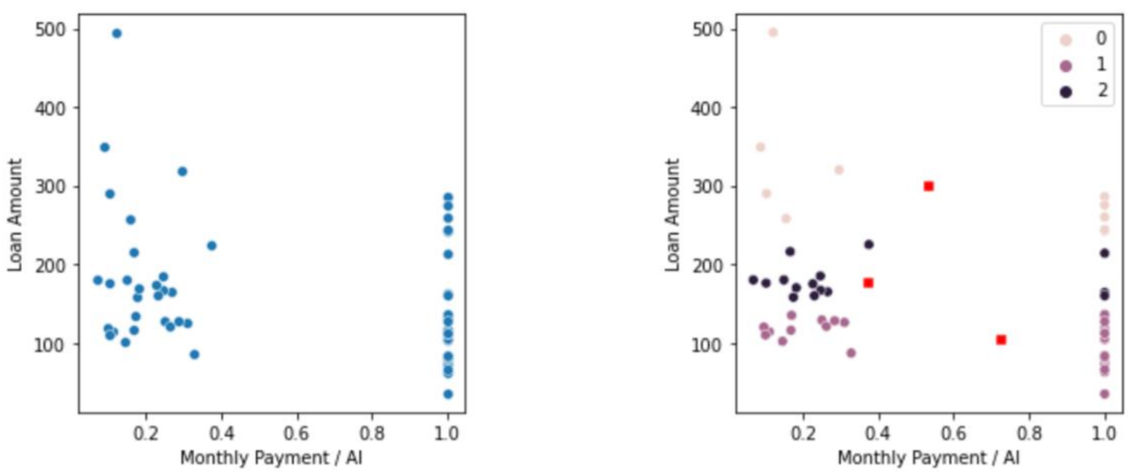


Figure 40 – Relationship between Monthly Payment / CAI and Loan Amount; applicants not eligble for a loan

After performing the fuzzy clustering using the FCM algorithm it was very evident that certain factors affect an applicant's loan eligibility more than others. The more evident factor is the debt to income ratio which we clearly see on figure 35 and 36. Property Area, displayed on figure 33 and 34 is also an important factor. A factor that was surprisingly not the most cruicial was the applicant's income. Surely it is very important for an applicant to have an income sufficient enough to cover their monthly debt however dhe debt to income ratio is a more useful factor.

CHAPTER 4

CONCLUSION

Bank loan eligibility and risk calculation are two key components that remain a concern of every financial institution. Having reliable risk evaluation systems is still a work in progress in almost every bank due to the sensitive manner of the topic, good credit risk evaluation means more profit for the bank.

Throughout this thesis, we saw how FL algorithm could be used to implement a system that could mimic human behaviour and way of thinking in order to make calculating risk an automated process. By using the right number of rules and studying real life scenarios and implementing these scenarious in our system we could obtain incredible results that would reflect the decision made by a human. By using a FCM we were able to check the weight that certain factors have in our application and how these factors will affect the decision made by the bank.

REFERENCES

[1] S. Anna, Personal Loan Statistics. Last visited on: May 27, 2022

Last visited at https://www.finder.com/personal-loans-statistics

[2] H.-J. Zimmermann, Fuzzy Set Theory- and Its Applications, Fourth Edition, 11-16 (2001).

[3] Fuzzy Logic (2004) Last visited on: May 29, 2022

Last visited at: reference.wolfram.com/legacy/applications/fuzzylogic/Manual/12.html

[4] E El-Khamy, Rowayda A Sadek, Mohamed A El-Khoreby, An efficient brain mass detection with adaptive clustered based fuzzy C-mean and thresholding, (2015).

[4] M. E. Karim, F. Yun, S. P. Madani, Fuzzy Cluster Analysis, (2010).

Last visited at: https://www.diva-portal.org/smash/get/diva2:829433/FULLTEXT01.pdf

[6] A. Shawkat, S. S. Abdul, Fuzzy Soft Multiset Theory, (2012)

Last visited at: https://www.hindawi.com/journals/aaa/2012/350603/

[7] S. A. Kalogirou, Fuzzy Inference, (2008)

Last visited at: https://www.sciencedirect.com/topics/engineering/fuzzy-inference

[8] M. Jarabaro, B. Boholi, Intelligent approaches for prediction of compressional, shear and Stoneley wave velocities from conventional well log data: A case study from the Sarvak carbonate reservoir in the Abadan Plain, (2007)

Last visited at: https://www.researchgate.net/figure/Six-types-of-fuzzy-membership-functions-triangular-A-z-shape-B-trapezoidal-C_fig2_223364345

[9] Princeton, Fuzzy Inferesce System. Last visited on: May 28, 2022 Last visited at: https://www.cs.princeton.edu/courses/archive/fall07/cos436/HIDDEN/Knapp/fuzzy004.htm#E7E50

[10] S. Marcin, Univeristy of Warsaw, Applied Fuzzy Logic, (2019)

Last visited at: https://www.mimuw.edu.pl/~szczuka/ls/Lecture5EN.pdf

[11] J. W. Mark, Creighton University, An introduction to Math of Uncertainty (2010)

Last visited at:

https://www.creighton.edu/fileadmin/user/CCAS/programs/fuzzy_math/docs/MOU.pdf

[12] Application on economy, business and administration. Last visited on: 7 June, 2022

Last visited at: https://search.proquest.com/docview/750061085?accountid=36765

[13] S. L. B. Ginting, The Application of Fuzzy Logic Method in the Debtors Eligibility Assessment System of Microfinance Institution, (2020)

[14] U. F. 1 Abdulrahman, J. K. Panford, and J. B. Hayfron-Acquah, Fuzzy Logic Approach to Credit Scoring for Micro Finance in Ghana: A Case Study of KWIQPLUS Money Lending, Int. J. Comput, (2014)

[15] N. R. Darwish, A. S. Abdelghany, A Fuzzy Logic Model for Credit Risk Rating of Egyptian Commercial Banks, (2016)

[16] S. Sampath, V. Kalaichelvi, Assessment of Mortgage Applications Using Fuzzy Logic, (2014)

[17] Loan Eligble Dataset Last accessed on: June 16, 2022

Last accessed at: https://www.kaggle.com/datasets/vikasukani/loan-eligible-dataset?select=loan-train.csv

[18] Fuzzy Logic Last accessed on: May 26, 2022

Last accessed at: https://www.javatpoint.com/fuzzy-logic

[19] M. E. Karim, F. Yun, S. P. Madani, Fuzzy Cluster Analysis, (2010).

Last visited at: https://www.diva-portal.org/smash/get/diva2:829433/FULLTEXT01.pdf

[20] Fuzzy Inference System Last accessed on: June 1, 2022

Last accessed at: https://www.tutorialspoint.com/fuzzy_logic/fuzzy_logic_inference_system.htm

[21] Example of a fuzzy inference system (2010) Last accessed on: June 2, 2022

Last accessed at: https://www.researchgate.net/figure/Example-of-inference-in-a-fuzzy-inference-system-adapted-from-MathWorks-2010_fig4_279510330

APPENDIX A

FIS and FCM code implementation

A.1. FIS implementation

The source code can be found here.

GraduationProject/FuzzyInferenceSystem.ipynb

```
import numpy as np

import skfuzzy as fuzz

from skfuzzy import control as ctrl

f1= ctrl.Antecedent(np.arange(0, 2, 1), 'Marital Status')

f2= ctrl.Antecedent(np.arange(0, 4, 1), 'Dependents')

f3= ctrl.Antecedent(np.arange(0, 2, 1), 'Level of Education')

f4= ctrl.Antecedent(np.arange(0, 2, 1), 'Self employed')

f9= ctrl.Antecedent(np.arange(0, 2, 1), 'Credit History')

f10= ctrl.Antecedent(np.arange(0, 4, 1), 'Property Area')

f5= ctrl.Antecedent(np.arange(0, 40000, 1), 'Applicant Income')

f6= ctrl.Antecedent(np.arange(0, 11301, 1), 'Co-applicant Income')

f7= ctrl.Antecedent(np.arange(0, 600, 1), 'Loan Amount')

f8= ctrl.Antecedent(np.arange(0, 481, 1), 'Loan Amount Term')

f11= ctrl.Antecedent(np.arange(0, 0.3514, 0.0001), 'Monthly Payment / AI')

f12= ctrl.Antecedent(np.arange(0, 0.555, 0.0001), 'Monthly Payment / CAI')

cr = ctrl.Consequent(np.arange(0, 101, 1), 'Credit Risk')
```

```
variable_1 = ['Married', 'Not married']

variable_2 = ['Zero','One', 'Two', 'Three or more']

variable_3 = ['Graduate','Non-Graduate']

variable_4 = ['Yes','No']

variable_5 = ['Meets guidelines', 'Does not meet guidelines']

variable_6 = ['Rural', 'Semi-urban', 'Urban']

variable_7 = ['Low', 'Medium', 'High', 'Very High']

variable_8 = ['Short', 'Medium', 'Long']


f1.automf(names = variable_1)

f2.automf(names = variable_2)

f3.automf(names = variable_3)

f4.automf(names = variable_4)

f5.automf(names = variable_7)

f6.automf(names = variable_7)

f7.automf(names = variable_7)

f8.automf(names = variable_8)

f9.automf(names = variable_5)

f10.automf(names = variable_6)

f11.automf(names = variable_7)

f12.automf(names = variable_7)
```

```
# Membership functions
# Factor 1: Marital Status
f1['Not married'] = fuzz.trimf(f1.universe, [0, 0, 0])
f1['Married'] = fuzz.trimf(f1.universe, [1, 1, 1])

# Factor 2: Dependents
f2['Zero'] = fuzz.trimf(f2.universe, [0, 0, 0])
f2['One'] = fuzz.trimf(f2.universe, [1, 1, 1])
f2['Two'] = fuzz.trimf(f2.universe, [2, 2, 2])
f2['Three or more'] = fuzz.trimf(f2.universe, [3, 3, 3])

# Factor 3: Level of Education

f3['Graduate'] = fuzz.trimf(f3.universe, [1, 1, 1])
f3['Non-Graduate'] = fuzz.trimf(f3.universe, [0, 0, 0])

# Factor 4: Self employed

f4['Yes'] = fuzz.trimf(f4.universe, [1, 1, 1])
f4['No'] = fuzz.trimf(f4.universe, [0, 0, 0])
# Factor 5: Applicant Income
f5['Low'] = fuzz.trimf(f5.universe, [0, 1250, 2000])
f5['Medium'] = fuzz.trimf(f5.universe, [1250, 2000, 3000])
f5['High'] = fuzz.trimf(f5.universe, [2000, 3000, 4800])
f5['Very High'] = fuzz.trapmf(f5.universe, [3000, 4800, 40000, 40000])
```

```
# Factor 6: Co-Applicant

f6['Low'] = fuzz.trimf(f6.universe, [0, 550, 1100])
f6['Medium'] = fuzz.trimf(f6.universe, [550, 1100, 1750])
f6['High'] = fuzz.trimf(f6.universe, [1100, 1750, 3200])
f6['Very High'] = fuzz.trapmf(f6.universe, [1750, 3200, 11301, 11301])

# Factor 7: Loan amount (in thousands)

f7['Low'] = fuzz.trimf(f7.universe, [0, 45, 90])
f7['Medium'] = fuzz.trimf(f7.universe, [45, 90, 150])
f7['High'] = fuzz.trimf(f7.universe, [90, 150, 200])
f7['Very High'] = fuzz.trapmf(f7.universe, [150, 200, 600, 600])

# Factor 8: Loan amount term

f8['Short'] = fuzz.trimf(f8.universe, [0, 50, 100])
f8['Medium'] = fuzz.trimf(f8.universe, [50, 100, 270])
f8['Long'] = fuzz.trimf(f8.universe, [100, 270, 480])

# Factor 9: Credit history

f9['Meets guidelines'] = fuzz.trimf(f9.universe, [1, 1, 1])
f9['Does not meet guidelines'] = fuzz.trimf(f9.universe, [0, 0, 0])
```

```
# Factor 10: Property area


f10['Rural'] = fuzz.trimf(f10.universe, [1, 1, 1])

f10['Semi-urban'] = fuzz.trimf(f10.universe, [2, 2, 2])

f10['Urban'] = fuzz.trimf(f10.universe, [3, 3, 3])


# Factor 11: Monthly Payment / AI


f11['Low'] = fuzz.trimf(f11.universe, [0, 0.0375, 0.065])

f11['Medium'] = fuzz.trimf(f11.universe, [0.0375, 0.065, 0.085])

f11['High'] = fuzz.trimf(f11.universe, [0.065, 0.085, 0.15])

f11['Very High'] = fuzz.trapmf(f11.universe, [0.085, 0.15, 0.3513, 0.3513])

# Factor 12: Monthly Payment / CAI


f12['Low'] = fuzz.trimf(f12.universe, [0, 0.0475, 0.085])

f12['Medium'] = fuzz.trimf(f12.universe, [0.0475, 0.085, 0.12])

f12['High'] = fuzz.trimf(f12.universe, [0.085, 0.12, 0.18])

f12['Very High'] = fuzz.trapmf(f12.universe, [0.12, 0.18, 0.555, 0.555])


# Credit risk

cr['Low'] = fuzz.trimf(cr.universe, [0, 0, 25])

cr['Medium'] = fuzz.trimf(cr.universe, [0, 25, 50])

cr['High'] = fuzz.trimf(cr.universe, [25, 50, 75])

cr['Very High'] = fuzz.trimf(cr.universe, [50, 75, 100])
```

```
rule1 = ctrl.Rule(f1['Married']
                  & f2['Zero']
                  & f3['Graduate']
                  & f4['Yes']
                  & f5['High']
                  & f6['Low']
                  & f7['Low']
                  & f8['Long']
                  & f9['Meets guidelines']
                  & f10['Urban']
                  & f11['Low']
                  & f12['Very High']
                 ,cr['Medium'])



rule2 = ctrl.Rule(f1['Not married']
                  & f2['Zero']
                  & f3['Graduate']
                  & f4['No']
                  & f5['High']
                  & f6['Low']
                  & f7['Medium']
                  & f8['Long']
```

```
                 & f9['Does not meet guidelines']
                 & f10['Semi-urban']
                 & f11['High']
                 & f12['Very High']
                ,cr['High'])



rule3 = ctrl.Rule(f1['Married']
                 & f2['One']
                 & f3['Graduate']
                 & f4['No']
                 & f5['Very High']
                 & f6['High']
                 & f7['High']
                 & f8['Long']
                 & f9['Meets guidelines']
                 & f10['Urban']
                 & f11['Medium']
                 & f12['Very High']
                ,cr['Low'])


rule4 = ctrl.Rule(f1['Not married']
                  & f2['Zero']
                  & f3['Graduate']
                  & f4['No']
```

```
          & f6['Very High']
          & f7['High']
          & f8['Long']
          & f9['Meets guidelines']
          & f10['Urban']
          & f11['Very High']
          & f12['High']
          ,cr['Very High'])

output_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4])
output = ctrl.ControlSystemSimulation(output_ctrl)
output.input['Marital Status'] = 0
output.input['Dependents'] = 0
output.input['Level of Education'] = 1
output.input['Self employed'] = 0
output.input['Applicant Income'] = 4166
output.input['Co-applicant Income'] = 1
output.input['Loan Amount'] = 116
output.input['Loan Amount Term'] = 360
output.input['Credit History'] = 0
output.input['Property Area'] = 2
output.input['Monthly Payment / AI'] = 0.0773
output.input['Monthly Payment / CAI'] = 1

output.compute()
print(output.output['Credit Risk'])
```

A.2. FCM implementation

The source code can be found here.

GraduationProject/FuzzyCmeans.ipynb

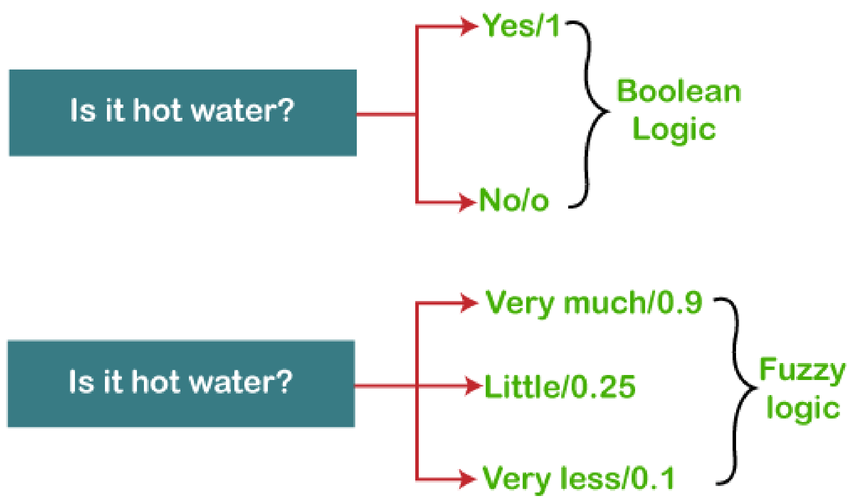
```
from fcmeans import FCM
from matplotlib import pyplot as plt
from seaborn import scatterplot as scatter
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore")
data = pd.read_csv('datasetEligble.csv')
mh1 = pd.DataFrame(data, columns = ['Dependents'])
mh2 = pd.DataFrame(data, columns = ['Applicant Income'])
mh3 = pd.DataFrame(data, columns = ['Coapplicant Income'])
mh4 = pd.DataFrame(data, columns = ['Loan Amount'])
mh5 = pd.DataFrame(data, columns = ['Loan Amount Term'])
mh6 = pd.DataFrame(data, columns = ['Property Area'])
mh7 = pd.DataFrame(data, columns = ['Monthly Payment / AI'])
mh8 = pd.DataFrame(data, columns = ['Monthly payment / CAI'])
mh_x1 = np.append(mh1, mh4, axis=1)
mh_x2 = np.append(mh2, mh4, axis=1)
mh_x3 = np.append(mh3, mh4, axis=1)
mh_x4 = np.append(mh5, mh4, axis=1)
mh_x5 = np.append(mh6, mh4, axis=1)
```

```
mh_x6 = np.append(mh7, mh4, axis=1)
mh_x7 = np.append(mh8, mh4, axis=1)



x1 = mh_x1
x2 = mh_x2
x3 = mh_x3
x4 = mh_x4
x5 = mh_x5
x6 = mh_x6
x7 = mh_x7

fcm = FCM(n_clusters = 3)
fcm.fit(x7)

centers = fcm.centers
labels  = fcm.u.argmax(axis = 1)



f, axes = plt.subplots(1, 2)
scatter(x7[:,0], x7[:,1], ax = axes[0])
scatter(x7[:,0], x7[:,1], ax = axes[1], hue = labels)

scatter(centers[:,0], centers[:,1], ax = axes[1], marker = "s", color = 'r', s = 35)
```

```
for ax in axes.flat:
    ax.set(xlabel = 'Monthly Payment / CAI', ylabel = 'Loan Amount')



plt.subplots_adjust(left=0.1,
                    bottom=0.1,
                    right=1.5,
                    top=0.9,
                    wspace=0.7,
                    hspace=0.4)


plt.show()
```

APPENDIX B

B.1. Figures



Boolean Logic vs Fuzzy Logic

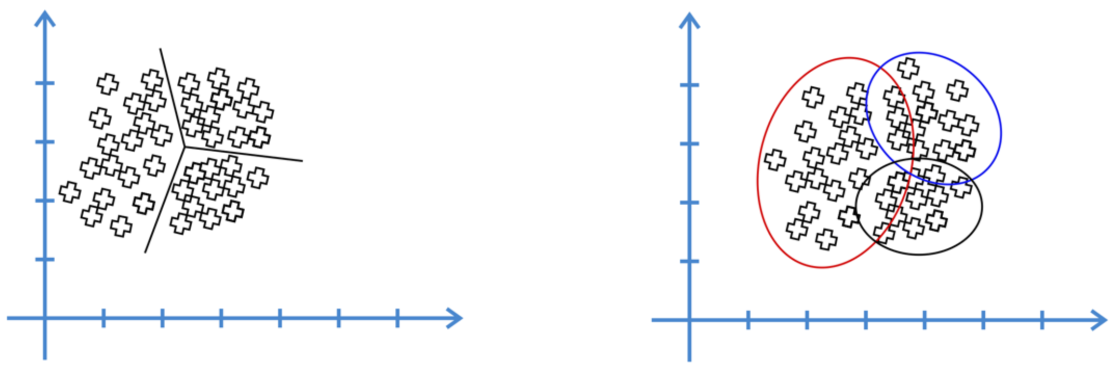
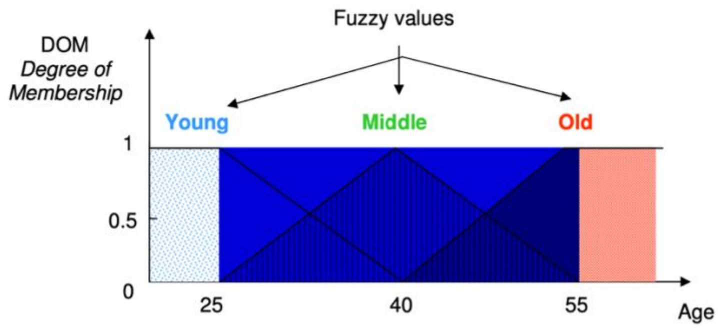
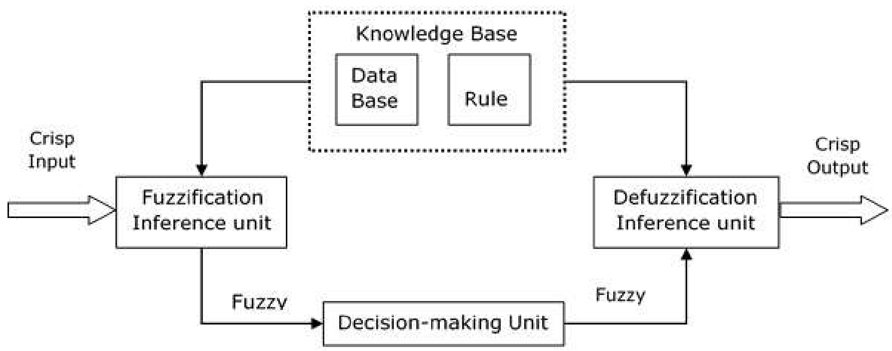


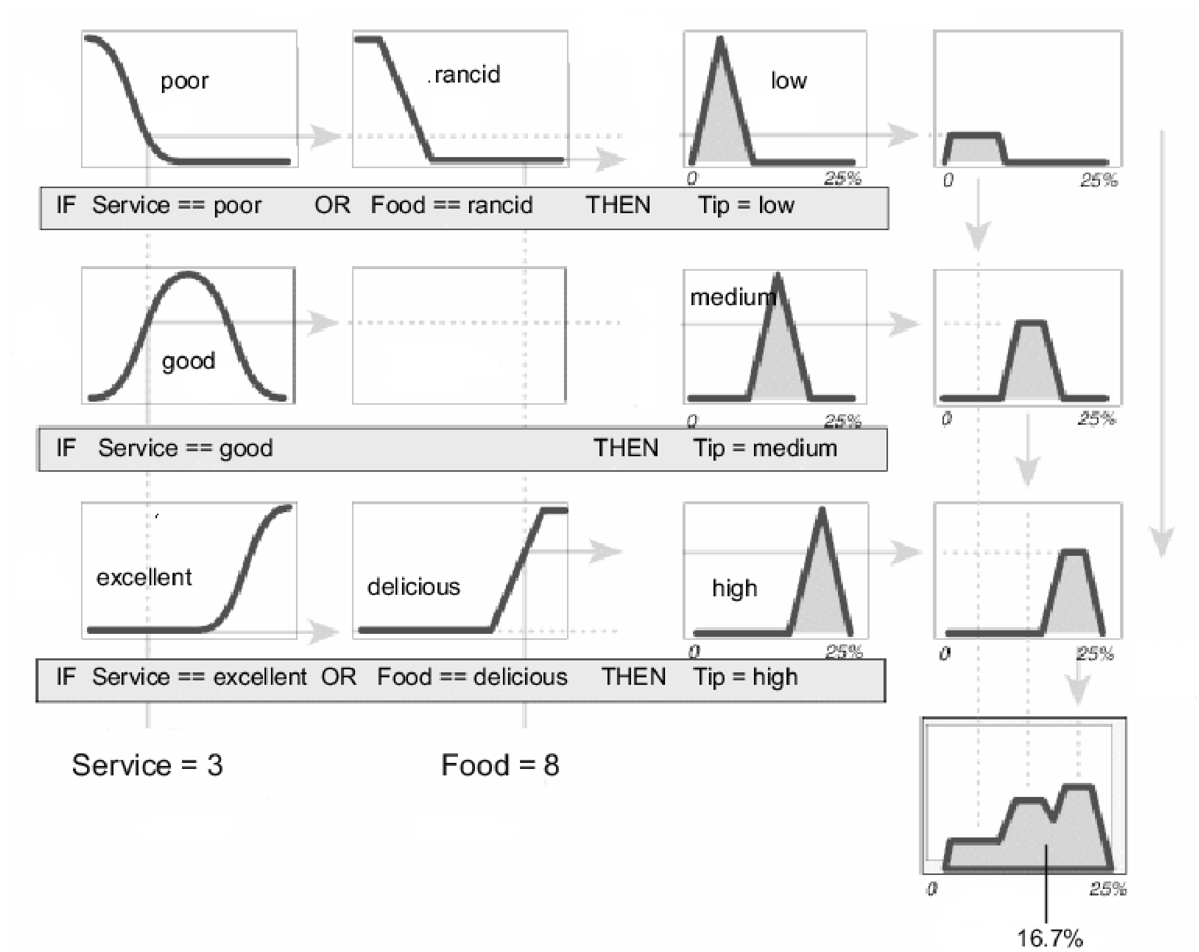
Figure 2. – Regular clustering vs Fuzzy clustering



Fuzzy triangular membership functions for the variable "age"

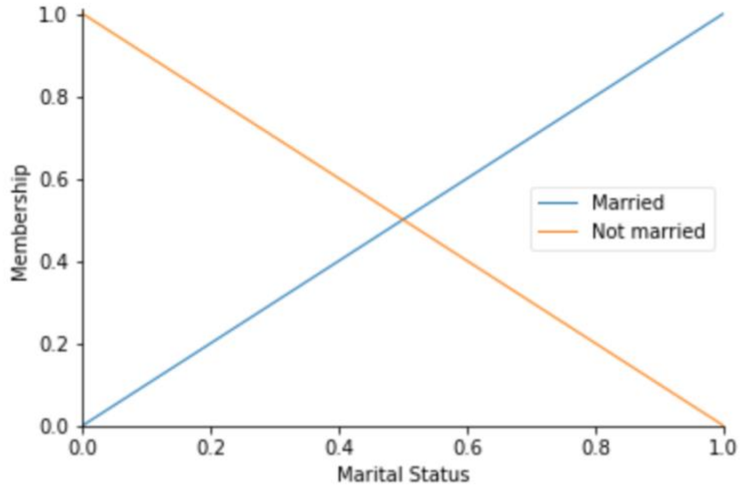


The structure of a Fuzzy Inference System

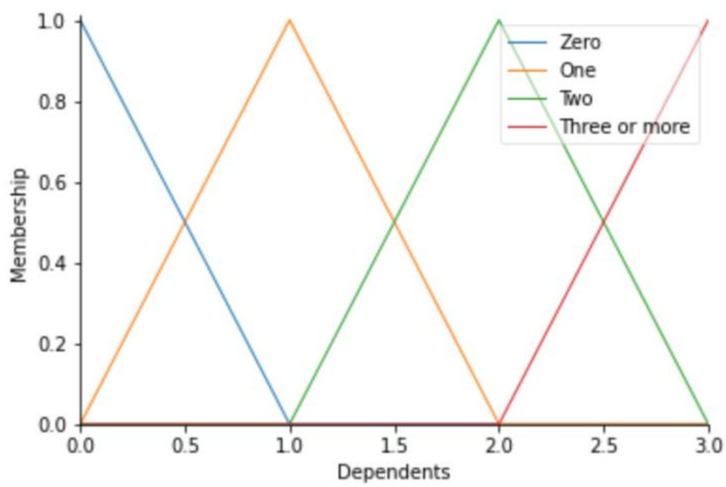


Example of a Mamdani fuzzy interface system

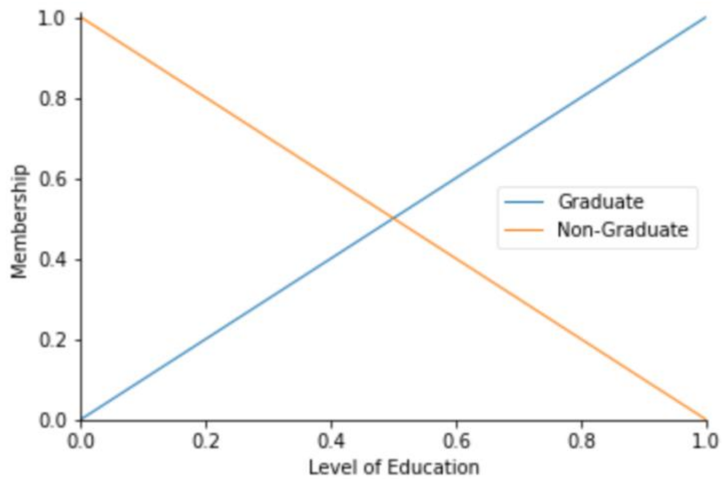
B.2. Membership Functions



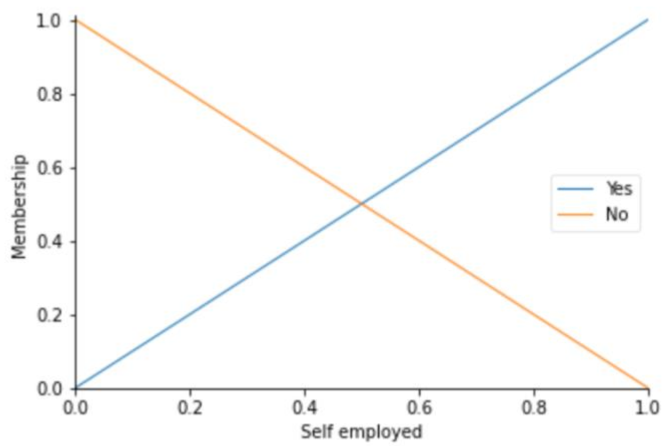
Membership function for *f1*



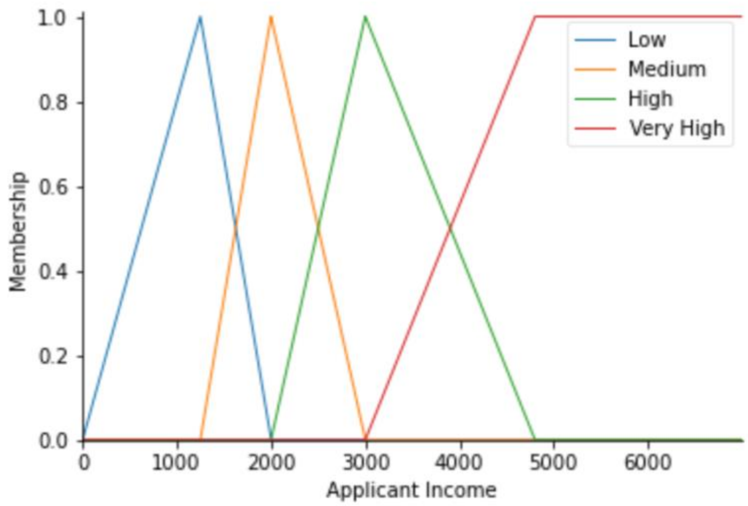
Membership function for *f2*



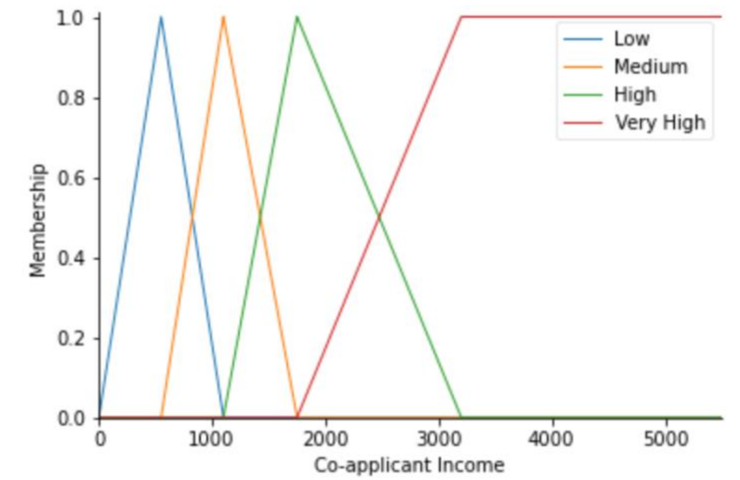
Membership function for *f3*



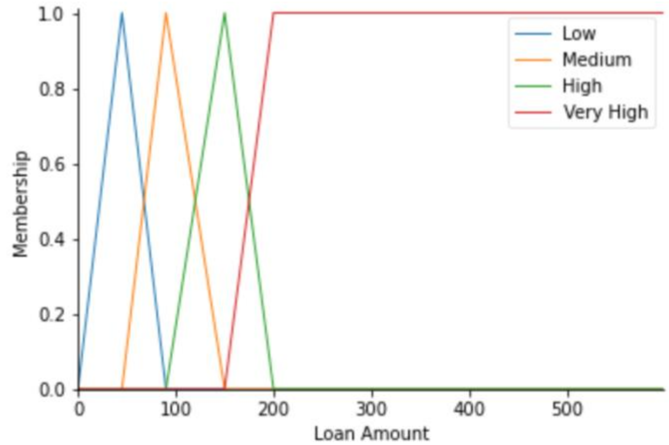
Membership function for *f4*



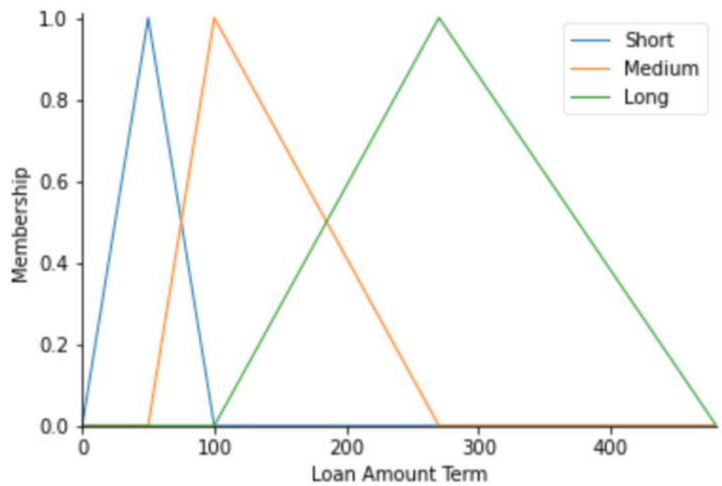
Membership function for *f5*



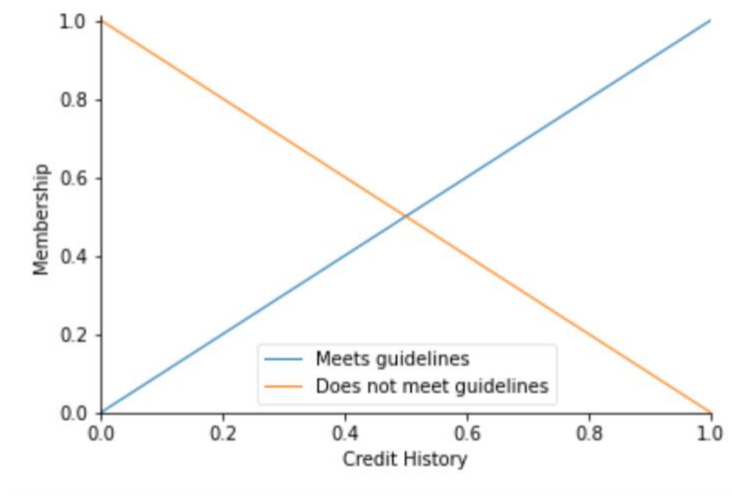
Membership function for *f6*



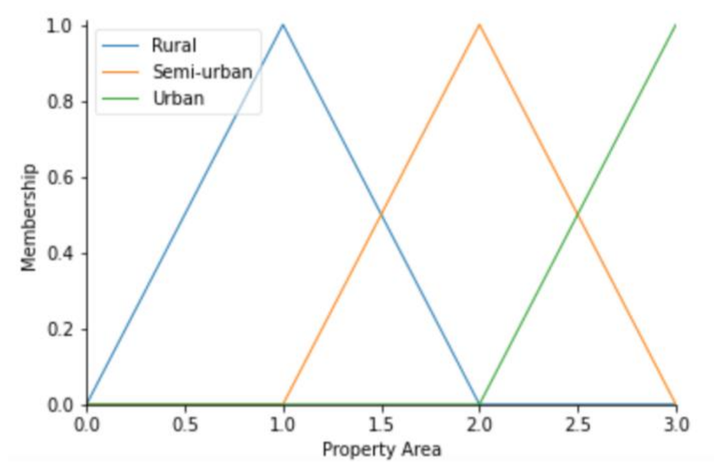
Membership function for *f7* (in thousands)



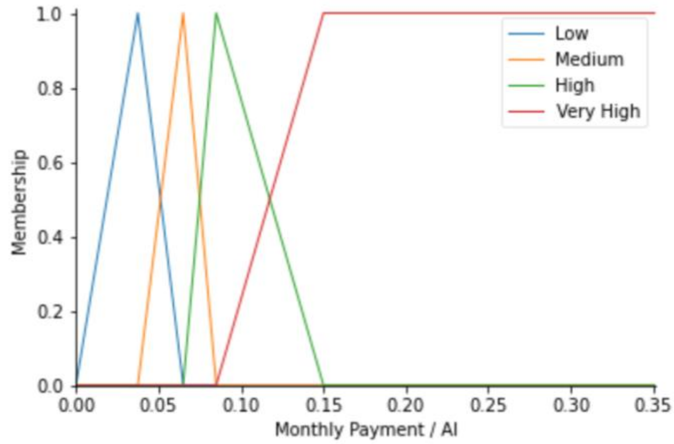
Membership function for *f8*



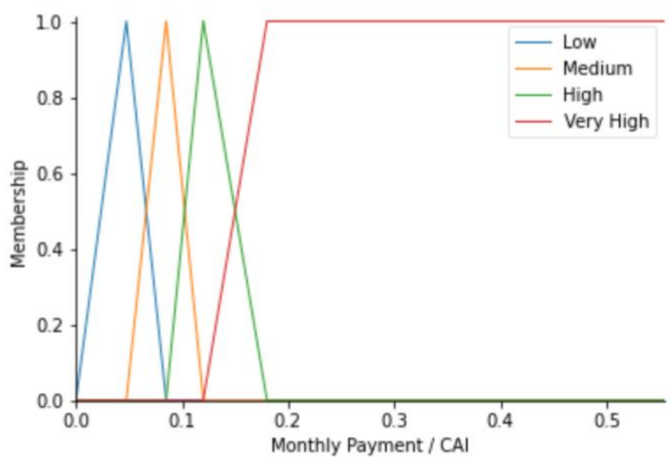
Membership function for *f9*



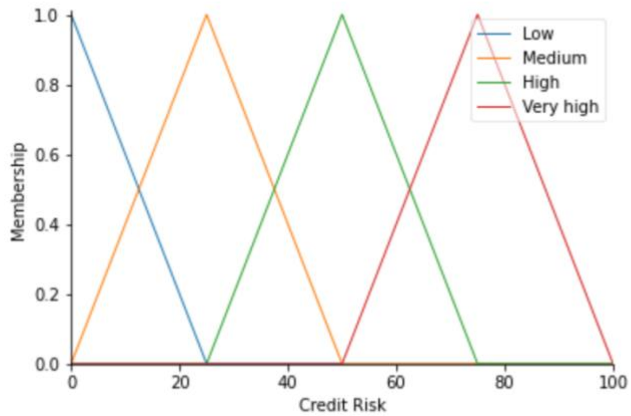
Membership function for *f10*



Membership function for *f11*

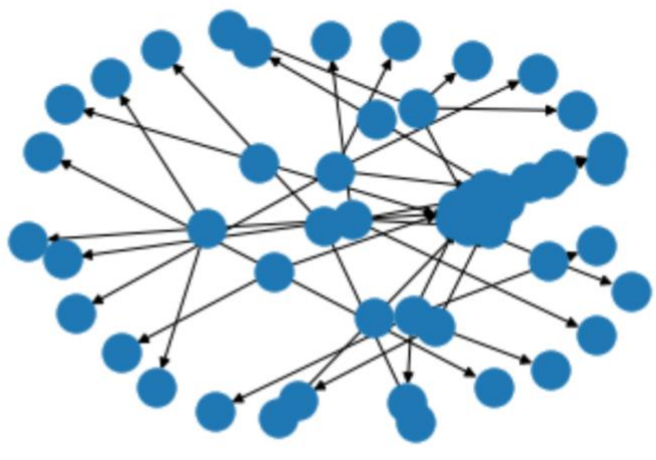


Membership function for *f12*

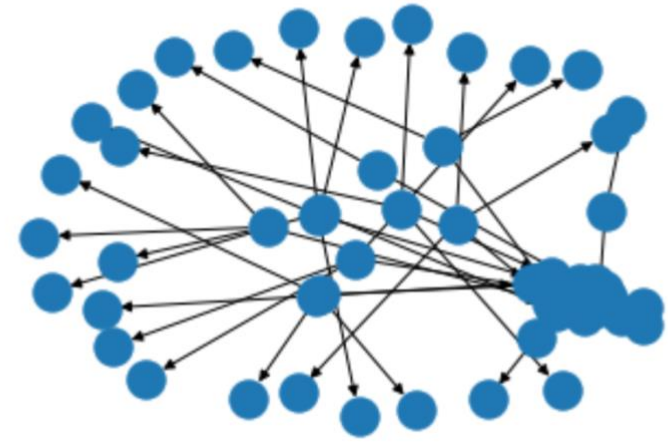


Membership function for *Credit Risk*

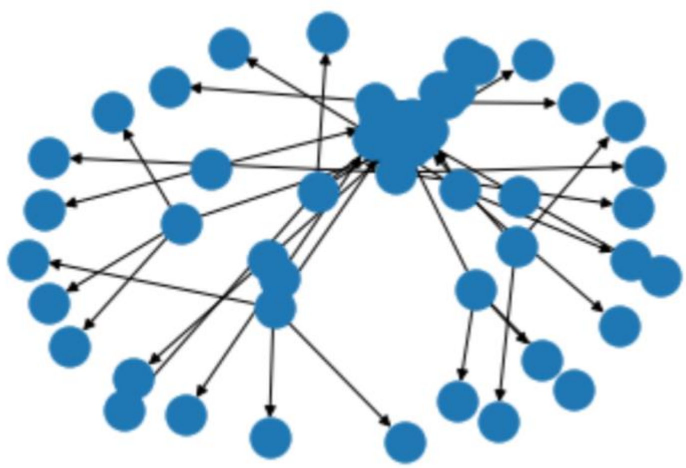
B.3. Visualized Rules



Rule 1



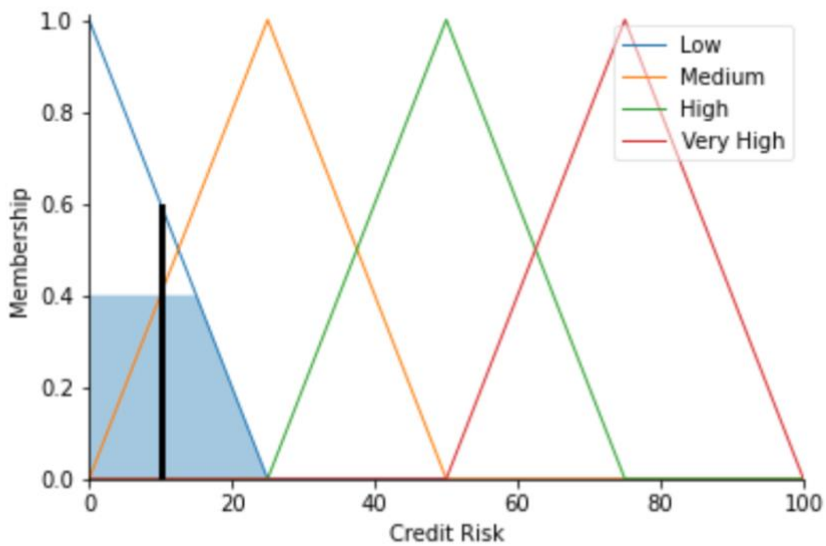
Rule 2



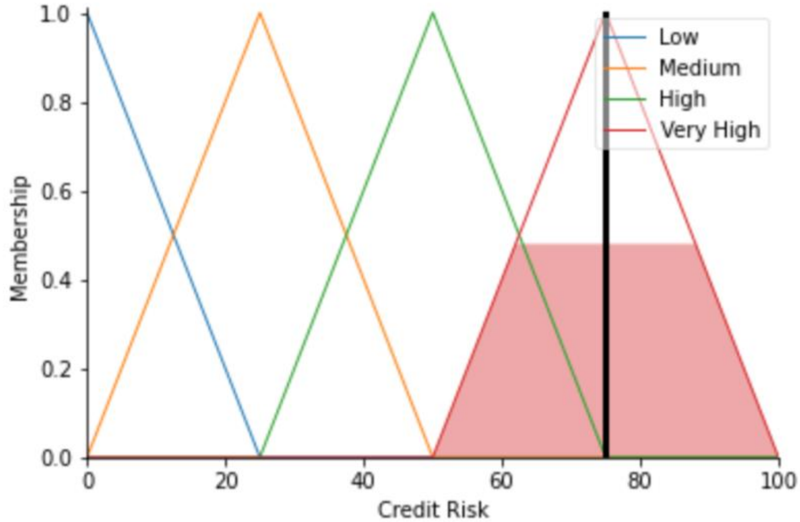
Rule 3

Rule 4

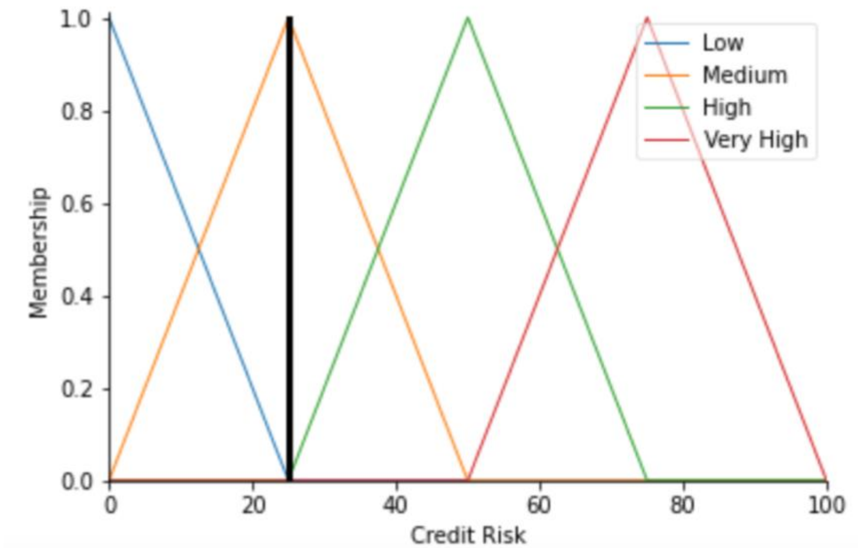
B.4. Grap area for rule output



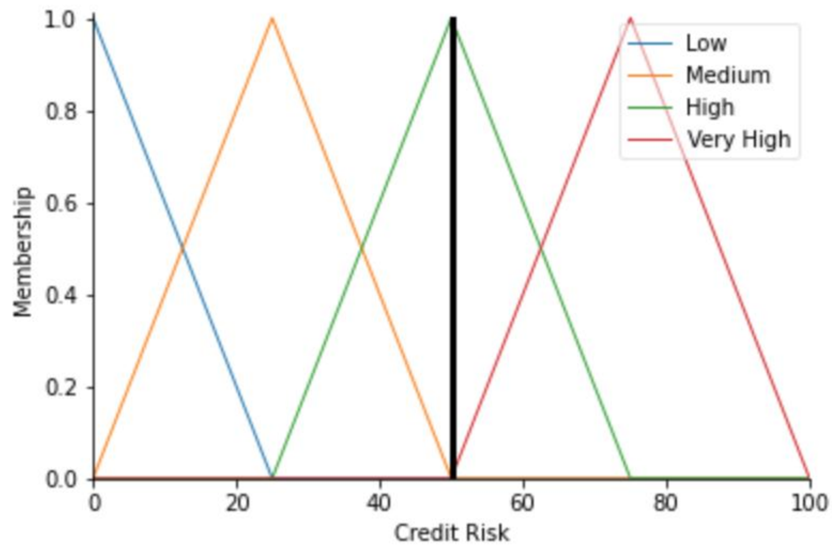
Graph area for rule 3



Graph area for rule 4

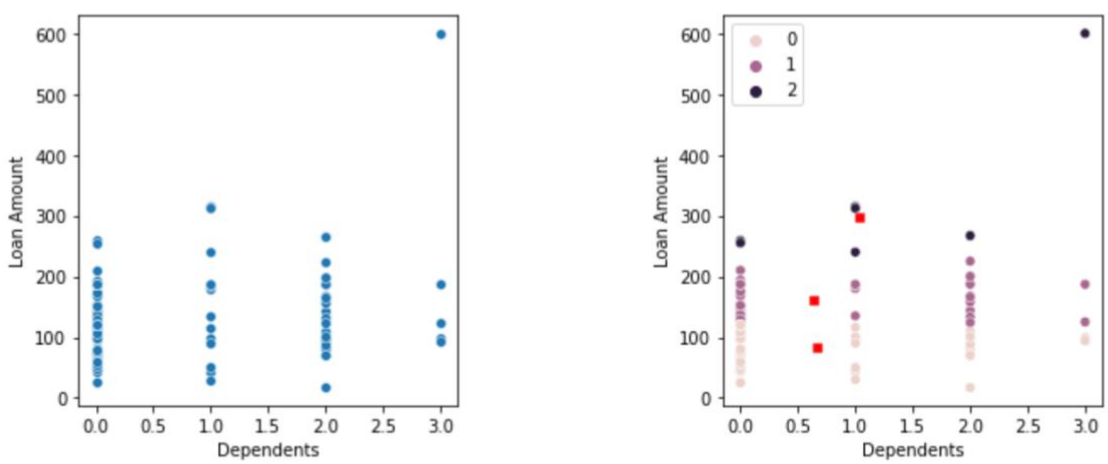


Graph area for rule 1

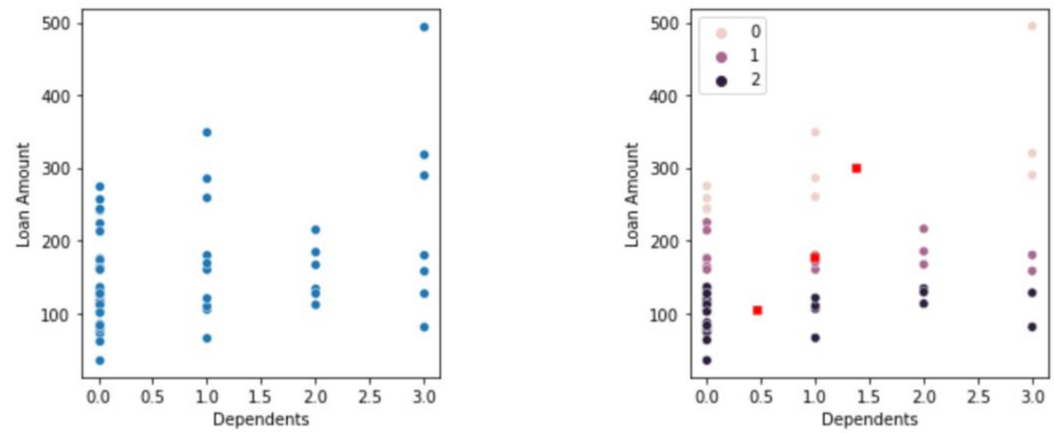


Graph area for rule 2

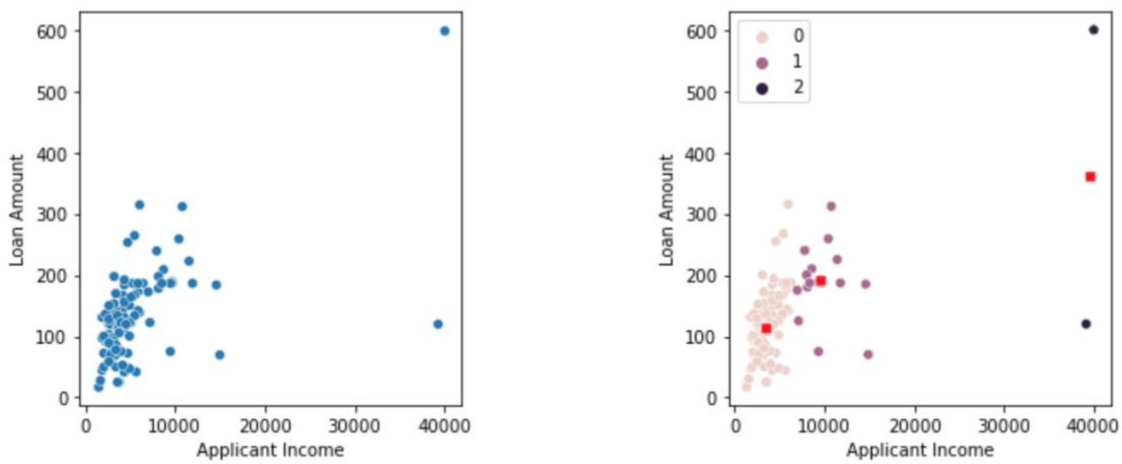
B.5. Clustering Outputs



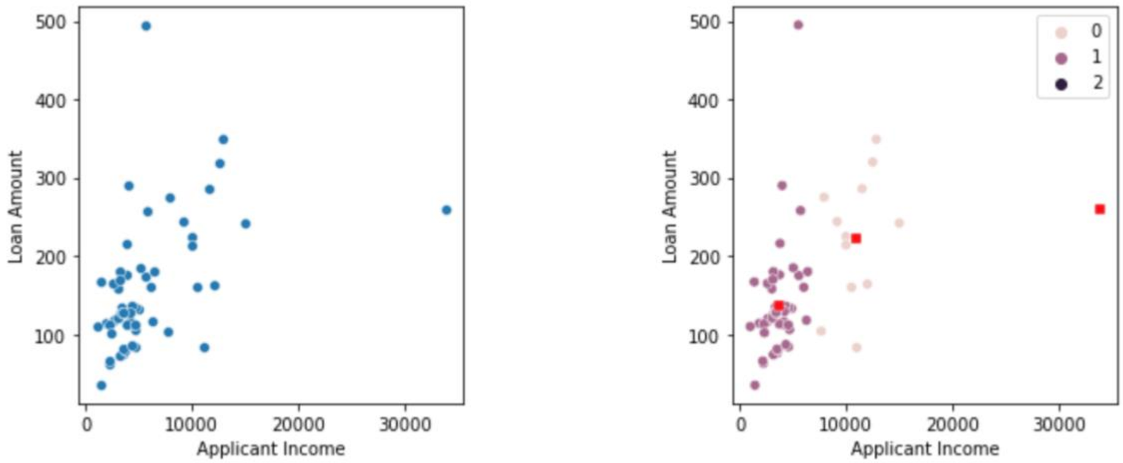
Relationship between Dependents and Loan Amount; loan eligble applicants



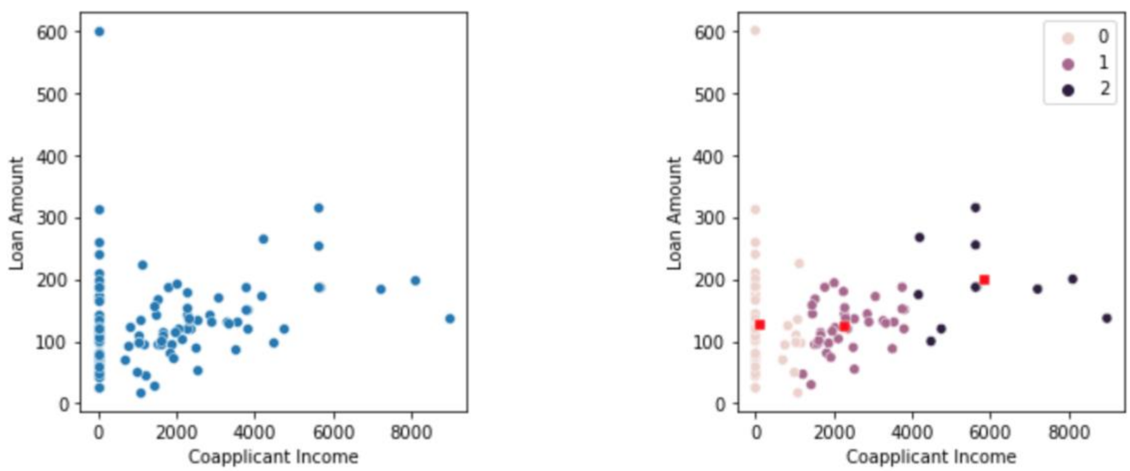
Relationship between Dependents and Loan Amount; applicants not eligble for a loan



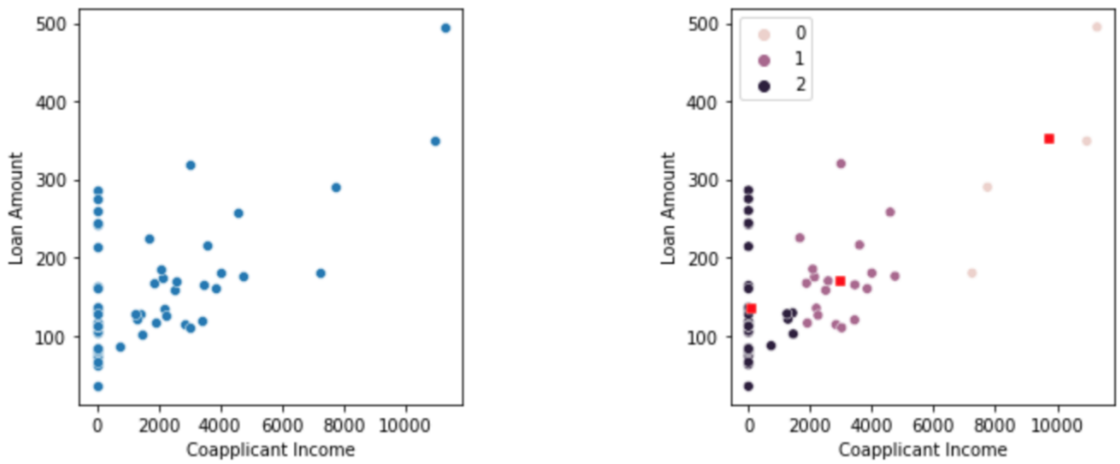
Relationship between Applicant Income and Loan Amount; eligble applicants



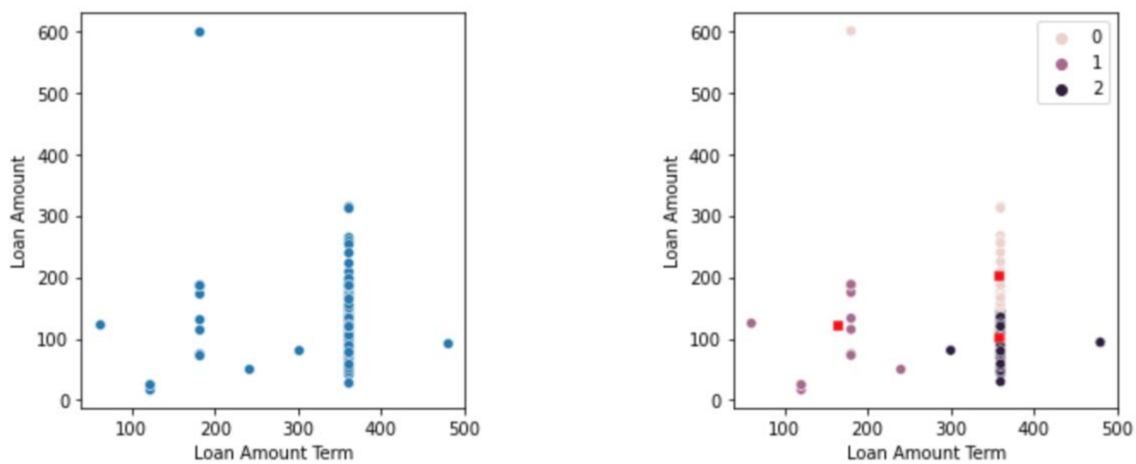
Relationship between Applicant Income and Loan Amount; applicant not eligble for a loan



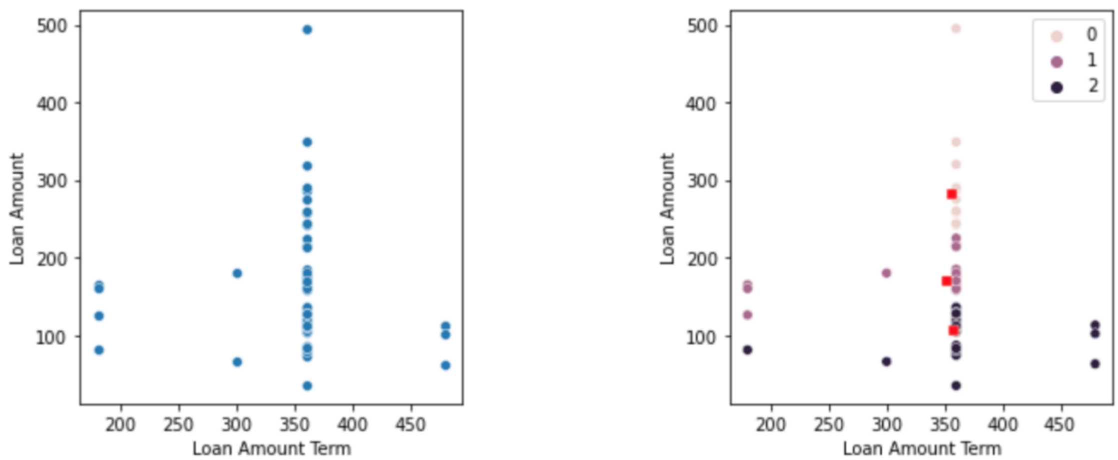
Relationship between Copplicant Income and Loan Amount; eligble applicants



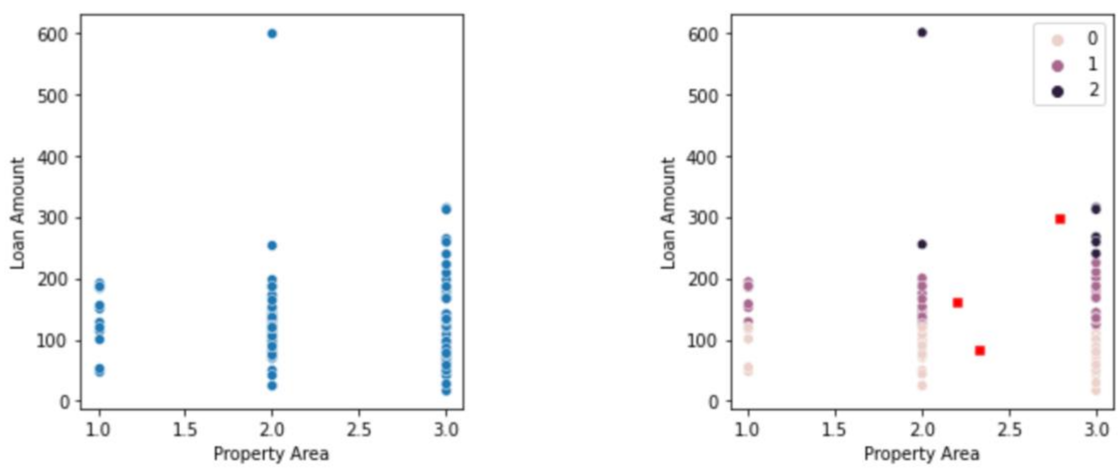
Relationship between Copplicant Income and Loan Amount; applicants not eligble for a loan



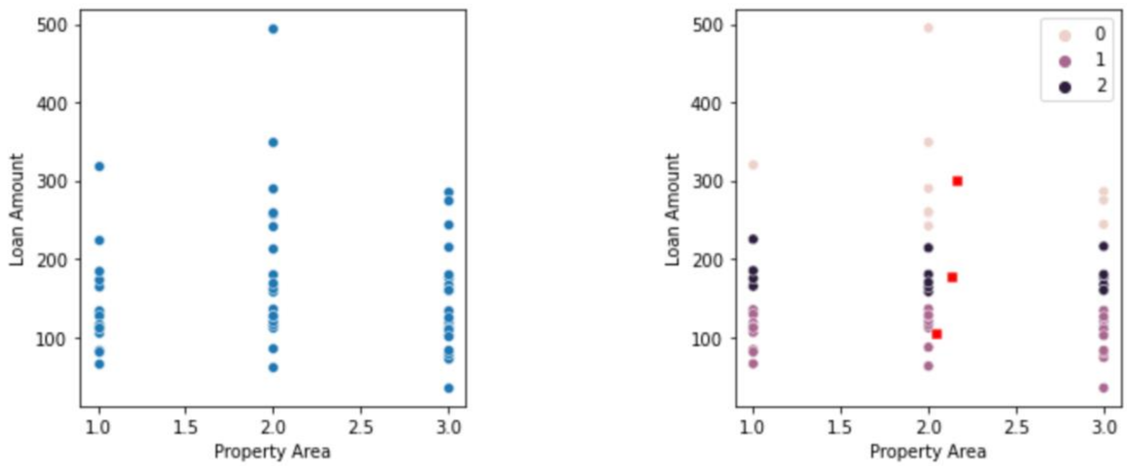
Relationship between Loan Amount Term and Loan Amount; eligble applicants



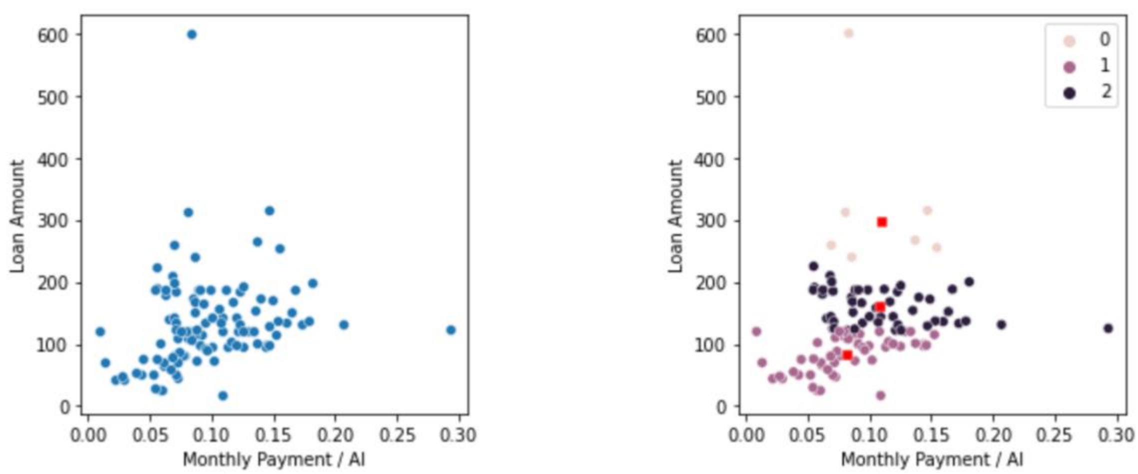
Relationship between Loan Amount Term and Loan Amount; applicants not eligble for a loan



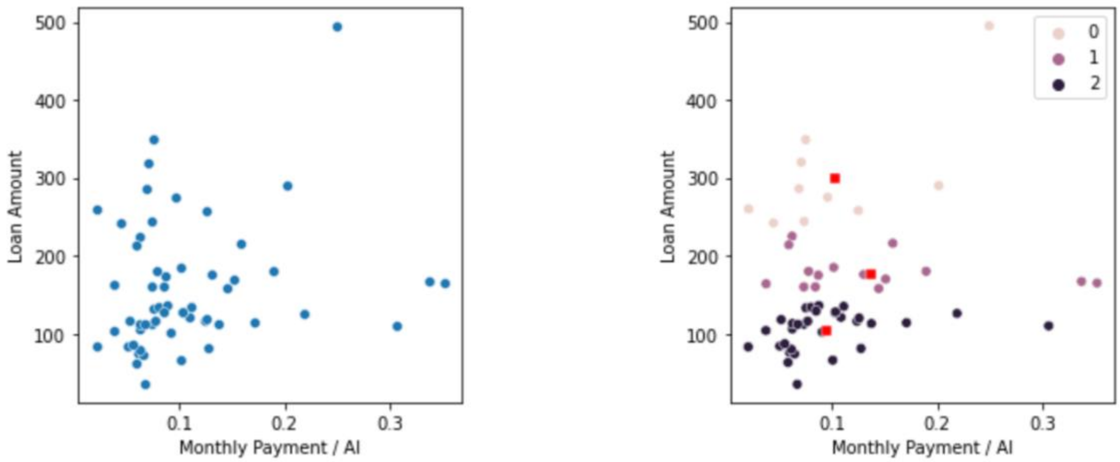
Relationship between Property Area and Loan Amount; eligble applicants



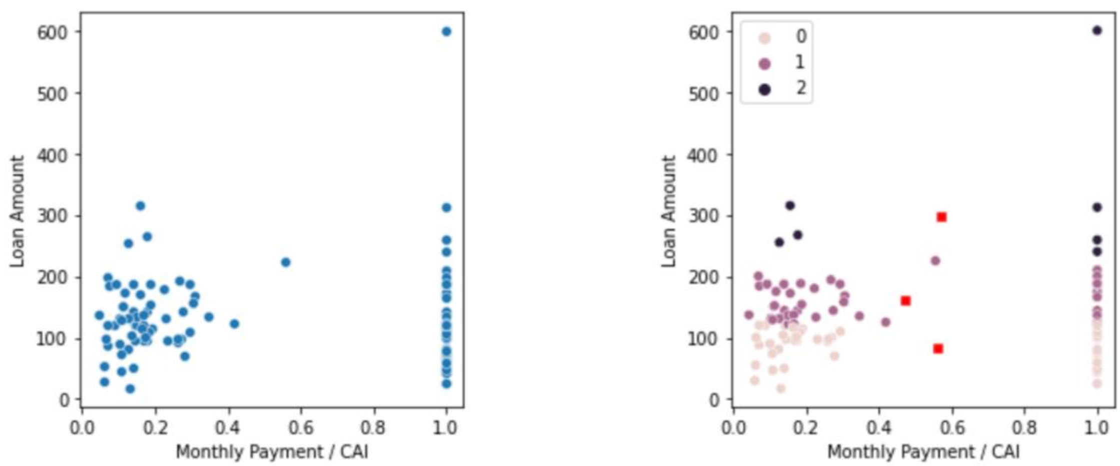
Relationship between Property Area and Loan Amount; applicants not eligble for a loan



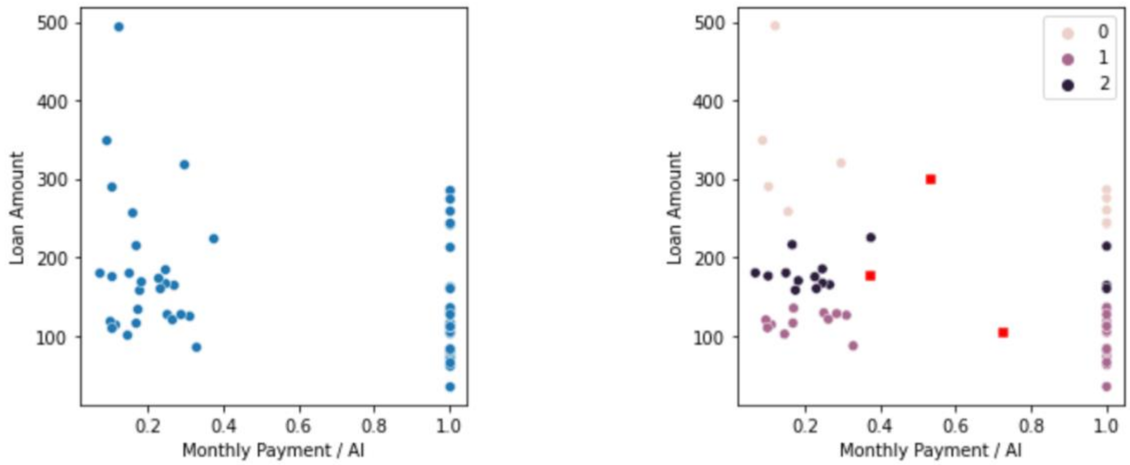
Relationship between Monthly Payment / AI and Loan Amount; eligble applicants



Relationship between Monthly Payment / AI and Loan Amount; applicants not eligble for a loan



Relationship between Monthly Payment / CAI and Loan Amount; eligble applicants



Relationship between Monthly Payment / CAI and Loan Amount; applicants not eligble for a loan