Fuzzy Systems

1. Problem definition:

We have ideated a problem statement where we predict the Quality of Ice-Cream taking into consideration various parameters relating to it and using the Fuzzy Logic to simulate such an environment. Nowadays, neural network technique is also implemented in food engineering to reduce production cost and to estimate the quality of food products. Ice cream is a mixture of different flavors, proteins, milk, water and emulsifiers. To produce the acceptable quality of ice cream different ingredients are mixed under standard conditions of temperature and pressure. The quality of the ice cream depends on its viscosity, overrun, melting rate and firmness. All these factors affect the overall quality of the ice cream.Viscosity, overrun, melting rate and firmness have been taken as input parameters and the overall quality of the ice cream has been taken as output. Fuzzy systems can work with vague information and can explain how to make decisions but they cannot automatically generate rules they use for decision making.

2. Fuzzification strategy:

The parameters of viscosity, overrun, melting rate and firmness are classified into three categories - Low, Medium and High. These categories are based on predefined values which are given to the system. According to the values, the category is decided and then the answer is decided based on the rulebase showcased in the next segment. The quality of ice-cream is divided into 3 categories based on the different values of the parameters. The categories are Poor, Good and Excellent. This label shows the overall quality that system provides based on the values of different parameters. This quality will be helpful in real time quality detection.

Parameter	Categories	Range
Viscosity	Low	0-18
	Medium	18-82
	High	82-100
Overrun	Low	0-22
	Medium	21-79
	High	79-100
Melting Rate	Low	0-20
	Medium	20-80
	High	80-100
Firmness	Low	0-29
	Medium	29-71
	High	71-100

3. Rulebase:

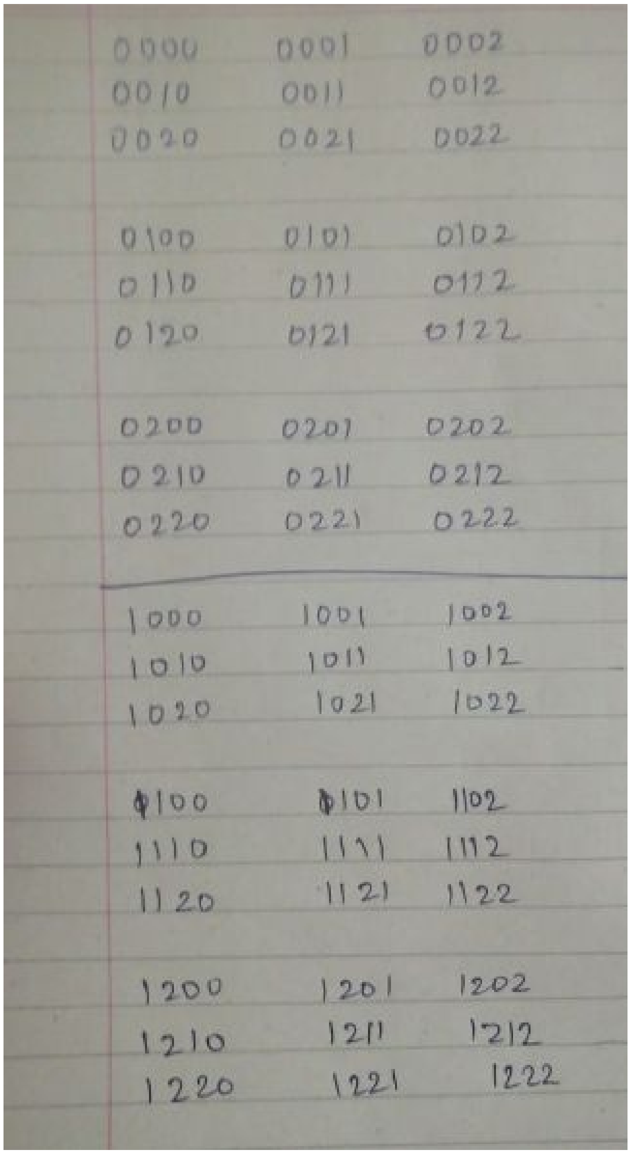
The rulebase for the problem is defined using a 4-dimensional matrix (3 * 3 * 3 * 3) as there are 3 categories for 4 input parameters and 3 categories for the quality predicted. So these lead to 81 different cases (permutations shown in the figure below) for input parameters.We use nested for loops for setting the rules for each input and output parameters.

The abbreviations used in the figure below are as follows:

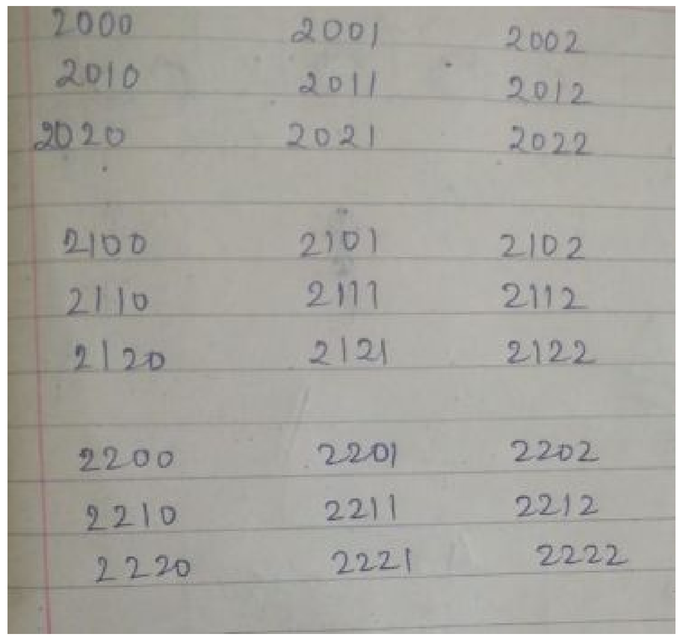
0 - Low

1 - Medium

2 - High



0000	0001	0002
0010	0011	0012
0020	0021	0022
0100	0101	0102
0110	0111	0112
0120	0121	0122
0200	0201	0202
0210	0211	0212
0220	0221	0222
1000	1001	1002
1010	1011	1012
1020	1021	1022
1100	1101	1102
1110	1111	1112
1120	1121	1122
1200	1201	1202
1210	1211	1212
1220	1221	1222



2000	2001	2002
2010	2011	2012
2020	2021	2022
2100	2101	2102
2110	2111	2112
2120	2121	2122
2200	2201	2202
2210	2211	2212
2220	2221	2222

We have defined some basic rules regarding rulebase formation and the output predicted, which are as follows:

Viscosity	**Overrun**	**Melting Rate**	**Firmness**	**Quality**
Low	Low	Low	High/Medium	Poor
Medium	Medium	Medium	Low/High	Good
High	High	High	Low/Medium	Excellent
Low	Low	Medium	Medium	Poor

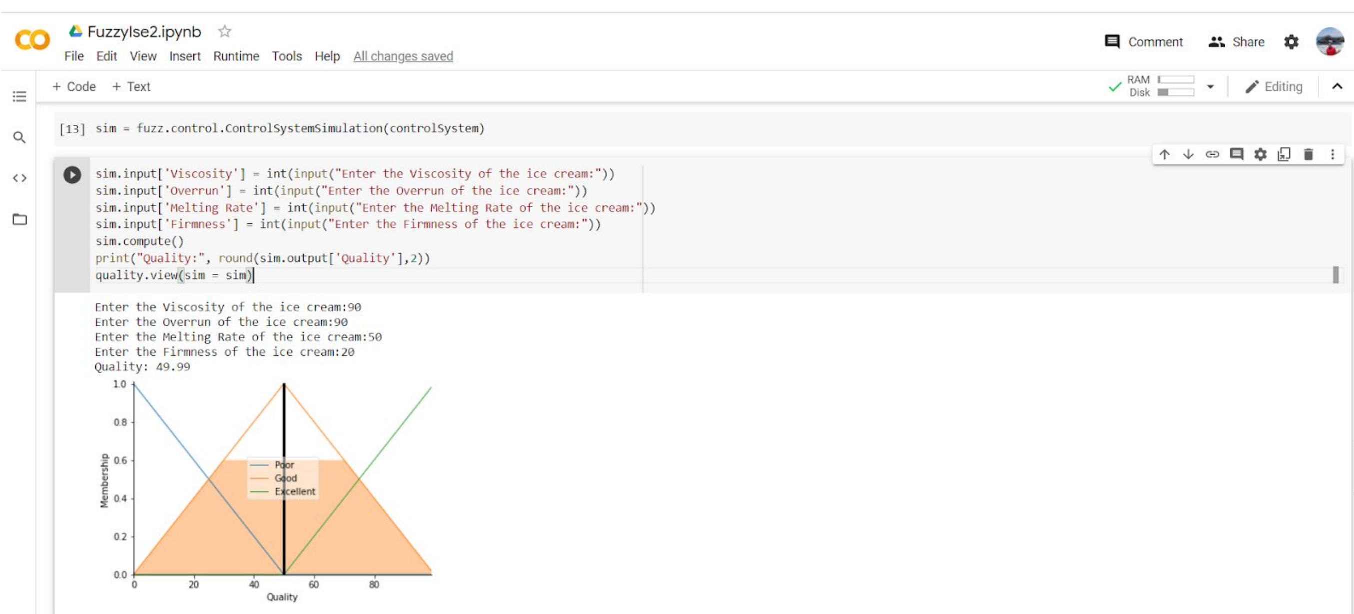
Medium	Medium	High	High	Good
Low	Low	High	High	Good
Low	Medium	High	High	Good
Low	Low	Medium	High	Good

We have used the AND operator to compute the Rules which means we are using the Fmin function. The output is then simulated based on these rules.

4. Defuzzification :

Quality	Poor	0-25
	Good	25-75
	Excellent	75-100

For Defuzzification, an object of control system is created where the antecedent is stored. This object is then passed to the Control System Simulator where the actual output is calculated and the output is given as a defuzzified value between 0-100(range defined). The output is also plotted and the corresponding category is shaded with its probability on the y-axis.

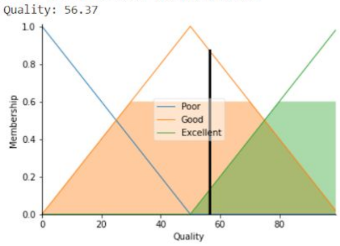


FuzzyIse2.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
[15] sim.input['Viscosity'] = int(input("Enter the Viscosity of the ice cream:"))
     sim.input['Overrun'] = int(input("Enter the Overrun of the ice cream:"))
     sim.input['Melting Rate'] = int(input("Enter the Melting Rate of the ice cream:"))
     sim.input['Firmness'] = int(input("Enter the Firmness of the ice cream:"))
     sim.compute()
     print("Quality:", round(sim.output['Quality'],2))
     quality.view(sim = sim)
```

```
Enter the Viscosity of the ice cream:20
Enter the Overrun of the ice cream:20
Enter the Melting Rate of the ice cream:90
Enter the Firmness of the ice cream:90
Quality: 56.37
```

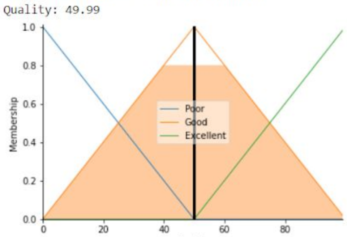


FuzzyIse2.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
[17] sim.input['Viscosity'] = int(input("Enter the Viscosity of the ice cream:"))
     sim.input['Overrun'] = int(input("Enter the Overrun of the ice cream:"))
     sim.input['Melting Rate'] = int(input("Enter the Melting Rate of the ice cream:"))
     sim.input['Firmness'] = int(input("Enter the Firmness of the ice cream:"))
     sim.compute()
     print("Quality:", round(sim.output['Quality'],2))
     quality.view(sim = sim)
```

```
Enter the Viscosity of the ice cream:20
Enter the Overrun of the ice cream:50
Enter the Melting Rate of the ice cream:50
Enter the Firmness of the ice cream:90
Quality: 49.99
```



Fuzzylse2.ipynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share

+ Code + Text

RAM Disk Editing

[14]

```
[18] sim.input['Viscosity'] = int(input("Enter the Viscosity of the ice cream:"))
     sim.input['Overrun'] = int(input("Enter the Overrun of the ice cream:"))
     sim.input['Melting Rate'] = int(input("Enter the Melting Rate of the ice cream:"))
     sim.input['Firmness'] = int(input("Enter the Firmness of the ice cream:"))
     sim.compute()
     print("Quality:", round(sim.output['Quality'],2))
     quality.view(sim = sim)
```

```
Enter the Viscosity of the ice cream:90
Enter the Overrun of the ice cream:90
Enter the Melting Rate of the ice cream:90
Enter the Firmness of the ice cream:20
Quality: 58.18
```