Experiment-9

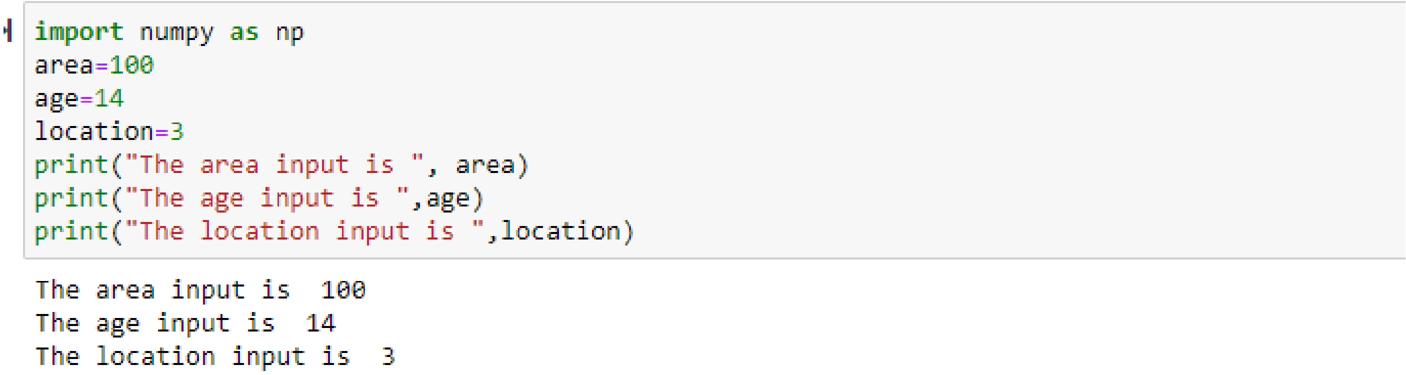
Objective: - Develop a fuzzy logic program for price prediction of a property based on inputs space, age, location.

- Use triangular, open left and open right membership function.
- Consider area from 10 – 1000 sq. feet.
- Consider age of the property from 1-100 years.
- Consider the location from 1- 10 points.
- Consider the price of the property from 0 – 10000000.

Software used: - Jupyter Notebook

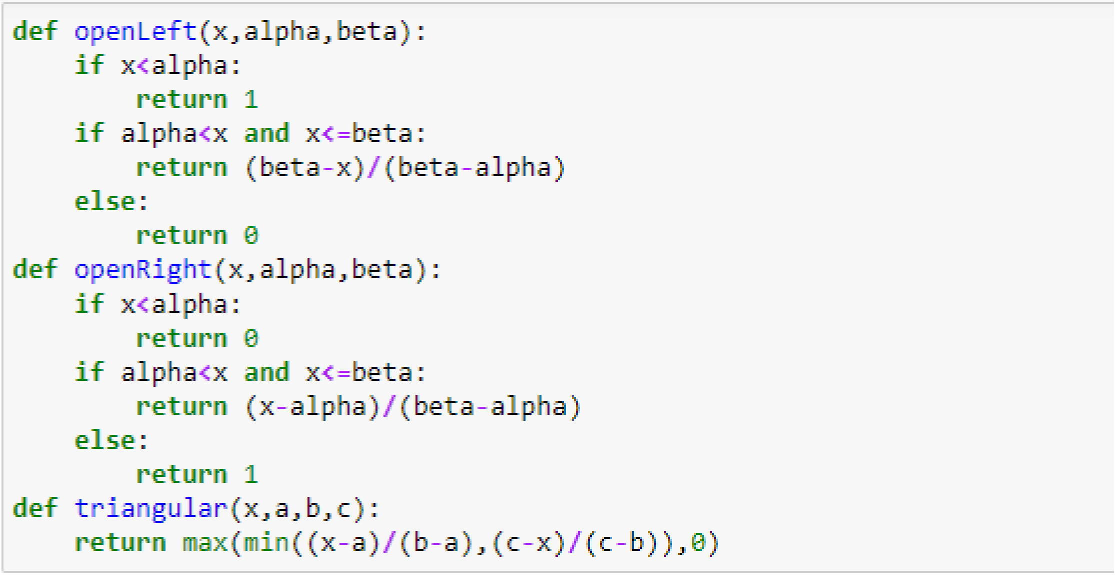
Theory :- Here we will be implementing the fuzzy function by taking three inputs space , age and location which will help to predict the price of the property which is our output.
Here are some abbreviations used:
LA : Low Area
MA : Medium Area
HA : High Area
LAG : Low Age
MAG : Medium Age
HAG : High Age
LL : Low Location
ML : Medium Location
HL : High Location
LP : Low Price
MP : Medium Price
HP : High Price

Code with simulated output: -

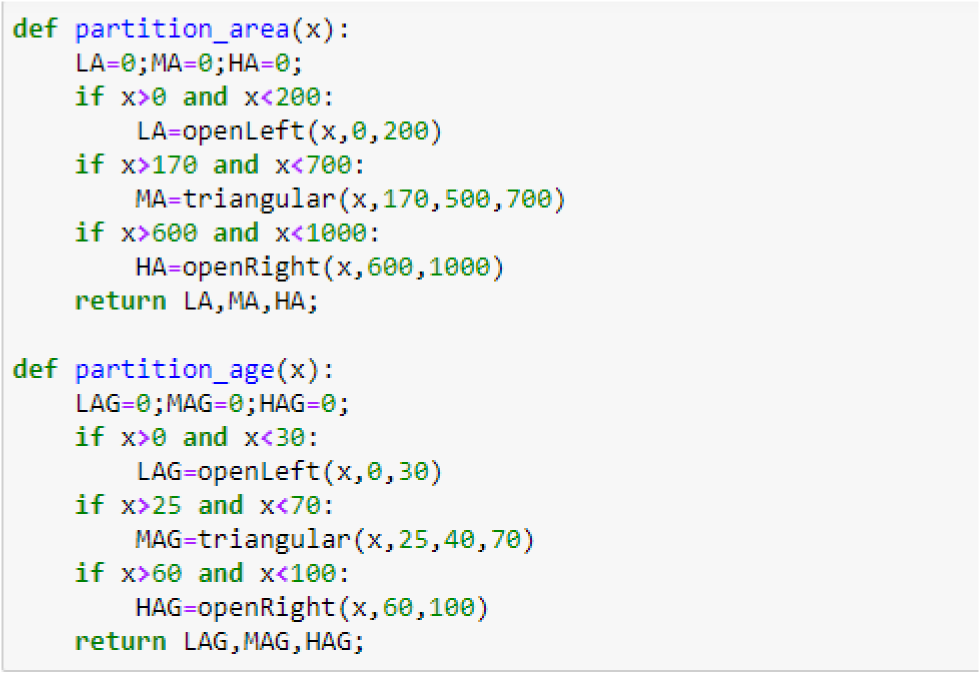


```python
import numpy as np
area=100
age=14
location=3
print("The area input is ", area)
print("The age input is ",age)
print("The location input is ",location)
```

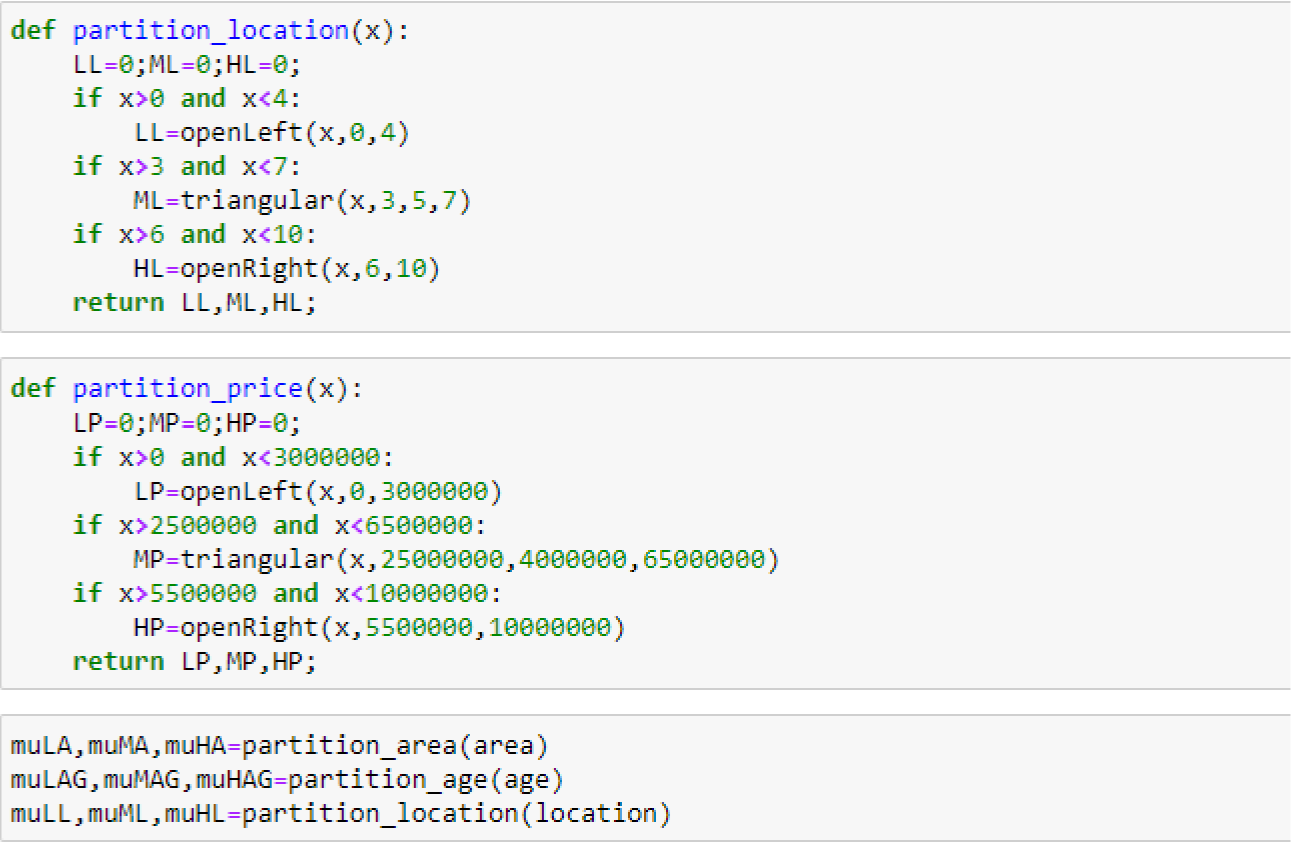
```
The area input is  100
The age input is  14
The location input is  3
```



```python
def openLeft(x,alpha,beta):
    if x<alpha:
        return 1
    if alpha<x and x<=beta:
        return (beta-x)/(beta-alpha)
    else:
        return 0
def openRight(x,alpha,beta):
    if x<alpha:
        return 0
    if alpha<x and x<=beta:
        return (x-alpha)/(beta-alpha)
    else:
        return 1
def triangular(x,a,b,c):
    return max(min((x-a)/(b-a),(c-x)/(c-b)),0)
```



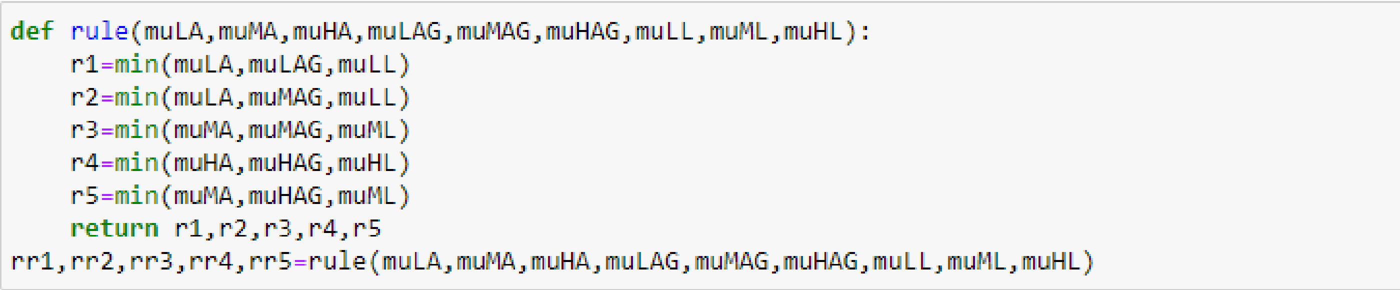
```python
def partition_area(x):
    LA=0;MA=0;HA=0;
    if x>0 and x<200:
        LA=openLeft(x,0,200)
    if x>170 and x<700:
        MA=triangular(x,170,500,700)
    if x>600 and x<1000:
        HA=openRight(x,600,1000)
    return LA,MA,HA;

def partition_age(x):
    LAG=0;MAG=0;HAG=0;
    if x>0 and x<30:
        LAG=openLeft(x,0,30)
    if x>25 and x<70:
        MAG=triangular(x,25,40,70)
    if x>60 and x<100:
        HAG=openRight(x,60,100)
    return LAG,MAG,HAG;
```



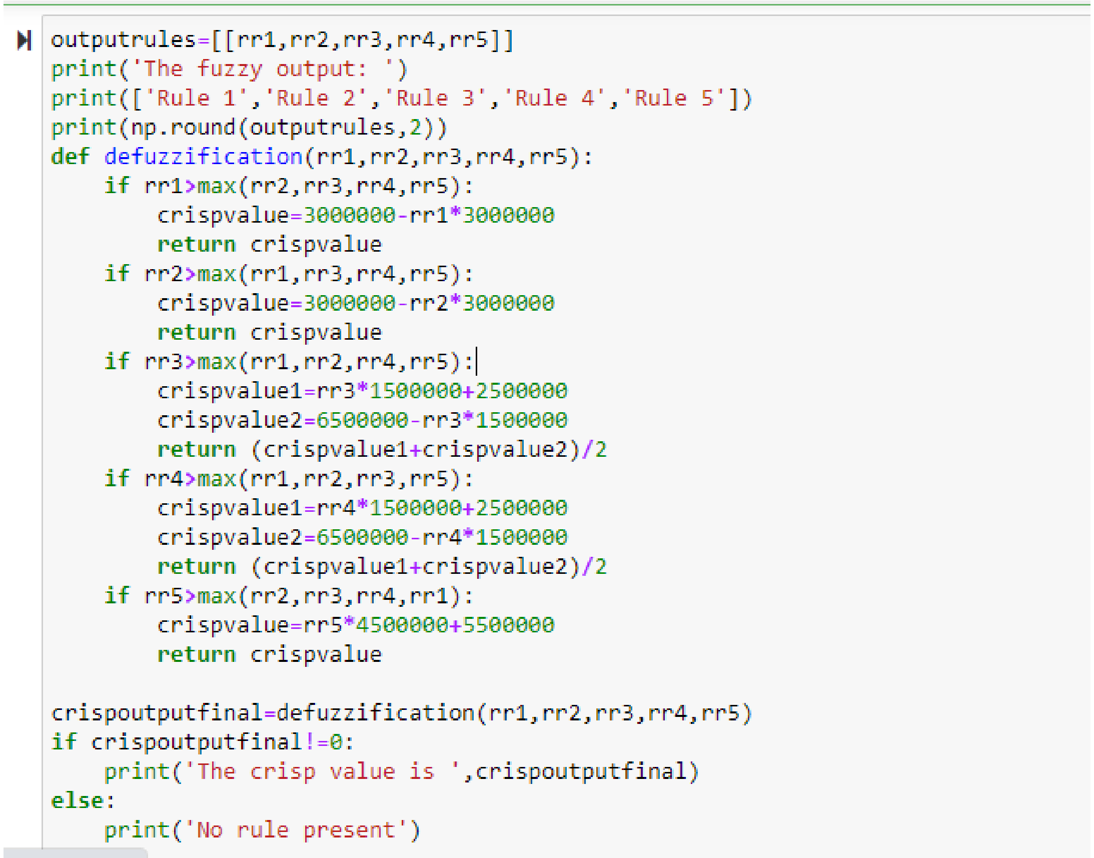
```python
def partition_location(x):
    LL=0;ML=0;HL=0;
    if x>0 and x<4:
        LL=openLeft(x,0,4)
    if x>3 and x<7:
        ML=triangular(x,3,5,7)
    if x>6 and x<10:
        HL=openRight(x,6,10)
    return LL,ML,HL;
```

```python
def partition_price(x):
    LP=0;MP=0;HP=0;
    if x>0 and x<3000000:
        LP=openLeft(x,0,3000000)
    if x>2500000 and x<6500000:
        MP=triangular(x,25000000,4000000,65000000)
    if x>5500000 and x<10000000:
        HP=openRight(x,5500000,10000000)
    return LP,MP,HP;
```

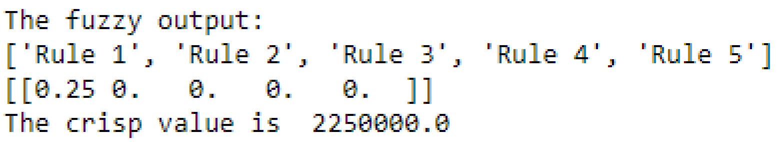
```python
muLA,muMA,muHA=partition_area(area)
muLAG,muMAG,muHAG=partition_age(age)
muLL,muML,muHL=partition_location(location)
```



```python
def rule(muLA,muMA,muHA,muLAG,muMAG,muHAG,muLL,muML,muHL):
    r1=min(muLA,muLAG,muLL)
    r2=min(muLA,muMAG,muLL)
    r3=min(muMA,muMAG,muML)
    r4=min(muHA,muHAG,muHL)
    r5=min(muMA,muHAG,muML)
    return r1,r2,r3,r4,r5
rr1,rr2,rr3,rr4,rr5=rule(muLA,muMA,muHA,muLAG,muMAG,muHAG,muLL,muML,muHL)
```



```
outputrules=[[rr1,rr2,rr3,rr4,rr5]]
print('The fuzzy output: ')
print(['Rule 1','Rule 2','Rule 3','Rule 4','Rule 5'])
print(np.round(outputrules,2))
def defuzzification(rr1,rr2,rr3,rr4,rr5):
    if rr1>max(rr2,rr3,rr4,rr5):
        crispvalue=3000000-rr1*3000000
        return crispvalue
    if rr2>max(rr1,rr3,rr4,rr5):
        crispvalue=3000000-rr2*3000000
        return crispvalue
    if rr3>max(rr1,rr2,rr4,rr5):
        crispvalue1=rr3*1500000+2500000
        crispvalue2=6500000-rr3*1500000
        return (crispvalue1+crispvalue2)/2
    if rr4>max(rr1,rr2,rr3,rr5):
        crispvalue1=rr4*1500000+2500000
        crispvalue2=6500000-rr4*1500000
        return (crispvalue1+crispvalue2)/2
    if rr5>max(rr2,rr3,rr4,rr1):
        crispvalue=rr5*4500000+5500000
        return crispvalue

crispoutputfinal=defuzzification(rr1,rr2,rr3,rr4,rr5)
if crispoutputfinal!=0:
    print('The crisp value is ',crispoutputfinal)
else:
    print('No rule present')
```



```
The fuzzy output:
['Rule 1', 'Rule 2', 'Rule 3', 'Rule 4', 'Rule 5']
[[0.25 0.   0.   0.   0.  ]]
The crisp value is  2250000.0
```

Conclusion: - We have predicted the price of the property using fuzzy approach where space, age and location is taken as input.