Koushik Sahu
118CS0597
Artificial Intelligence Lab – 8
17th November 2021, Wednesday

1.

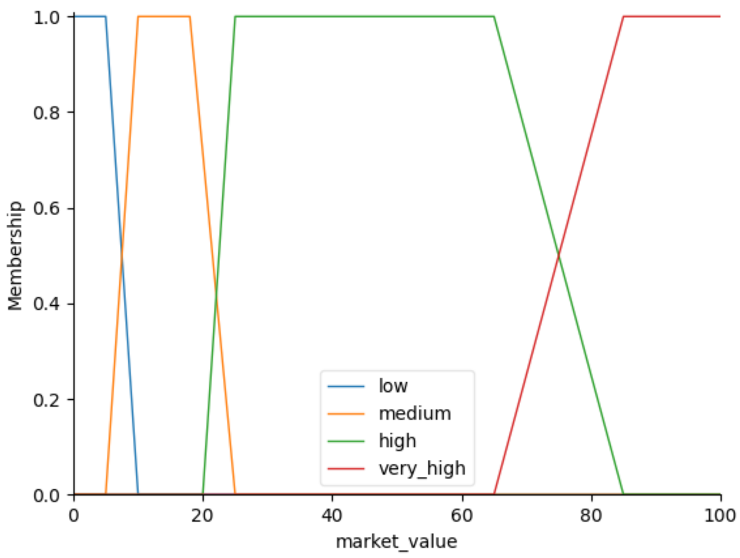
```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt

market_value =  ctrl.Antecedent(np.arange(0, 101, 1), 'market_value')
location =  ctrl.Antecedent(np.arange(0, 11, 0.5), 'location')
house = ctrl.Consequent(np.arange(0, 11, 1), 'house')

market_value['low'] = fuzz.trapmf(market_value.universe, [0, 0, 5, 10])
market_value['medium'] = fuzz.trapmf(market_value.universe, [5, 10, 18, 25])
market_value['high'] = fuzz.trapmf(market_value.universe, [20, 25, 65, 85])
market_value['very_high'] = fuzz.trapmf(market_value.universe, [65, 85, 100, 100])

location['bad'] = fuzz.trapmf(location.universe, [0, 0, 1.5, 4])
location['fair'] = fuzz.trapmf(location.universe, [2.5, 5, 6, 8.5])
location['excellent'] = fuzz.trapmf(location.universe, [6, 8.5, 10, 10])

house['very_low'] = fuzz.trimf(house.universe, [0, 0, 3])
house['low'] = fuzz.trimf(house.universe, [0, 3, 6])
house['medium'] = fuzz.trimf(house.universe, [2, 5, 8])
house['high'] = fuzz.trimf(house.universe, [4, 7, 10])
house['very_high'] = fuzz.trimf(house.universe, [7, 10, 10])

market_value.view()

location.view()

house.view()

he_rule1 = ctrl.Rule(market_value['low'], house['low'])
he_rule2 = ctrl.Rule(location['bad'], house['low'])

he_rule3 = ctrl.Rule(market_value['low'] & location['bad'], house['very_low'])
he_rule4 = ctrl.Rule(market_value['medium'] & location['bad'], house['low'])
he_rule5 = ctrl.Rule(market_value['high'] & location['bad'], house['medium'])
he_rule6 = ctrl.Rule(market_value['very_high'] & location['bad'], house['high'])

he_rule7 = ctrl.Rule(market_value['low'] & location['fair'], house['low'])
he_rule8 = ctrl.Rule(market_value['medium'] & location['fair'], house['medium'])
he_rule9 = ctrl.Rule(market_value['high'] & location['fair'], house['high'])
he_rule10 = ctrl.Rule(market_value['very_high'] & location['fair'], house['very_high'])

he_rule11 = ctrl.Rule(market_value['low'] & location['excellent'], house['medium'])
he_rule12 = ctrl.Rule(market_value['medium'] & location['excellent'], house['high'])
```

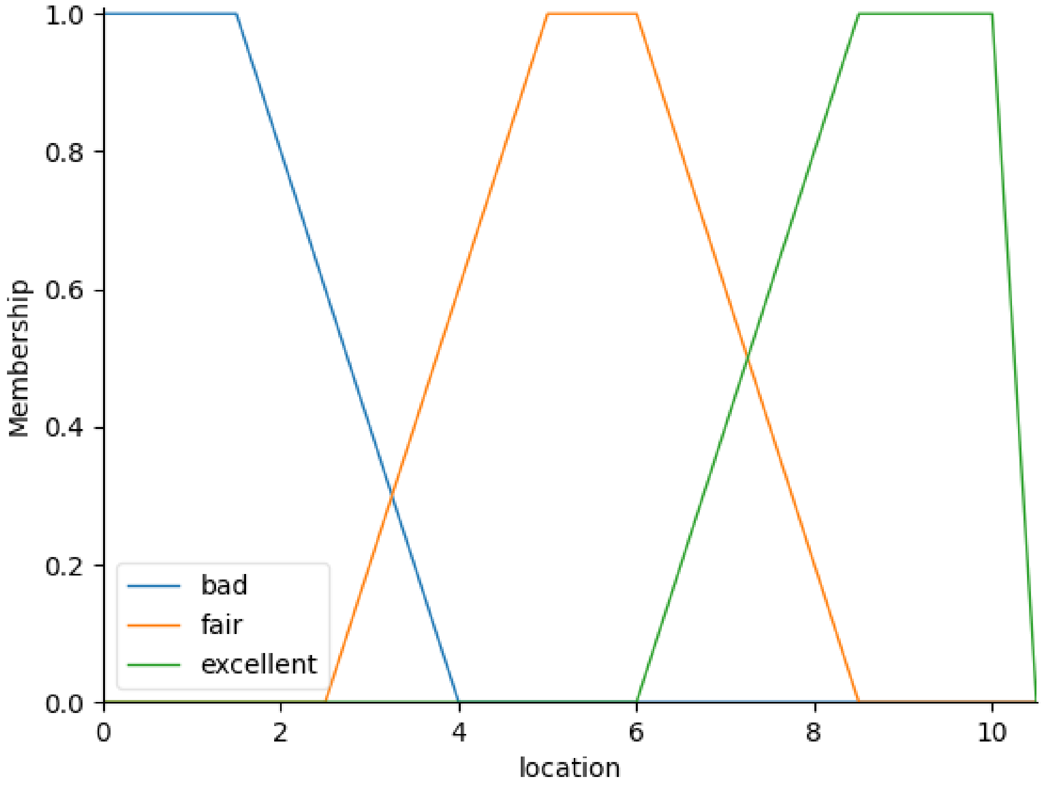
```
he_rule13 = ctrl.Rule(market_value['high'] & location['excellent'], house['very_high'])
he_rule14 = ctrl.Rule(market_value['very_high'] & location['excellent'], house['very_high'])

he_Rules = [he_rule1, he_rule2, he_rule3, he_rule4, he_rule5, he_rule6, he_rule7, he_rule8,
he_rule9, he_rule10, he_rule11, he_rule12, he_rule13, he_rule14]
house_ctrl = ctrl.ControlSystem(he_Rules)
house_eval = ctrl.ControlSystemSimulation(house_ctrl)

assets =  ctrl.Antecedent(np.arange(0, 1001, 1), 'assets')
income =  ctrl.Antecedent(np.arange(0, 101, 1), 'income')
applicant = ctrl.Consequent(np.arange(0, 11, 1), 'applicant')

assets['low'] = fuzz.trimf(assets.universe, [0, 0, 150])
assets['medium'] = fuzz.trapmf(assets.universe, [50, 250, 450, 650])
assets['high'] = fuzz.trapmf(assets.universe, [500, 700, 1000, 1000])

income['low'] = fuzz.trapmf(income.universe, [0, 0, 10, 25])
income['medium'] = fuzz.trimf(income.universe, [15, 35, 55])
income['high'] = fuzz.trimf(income.universe, [40, 60, 80])
income['very_high'] = fuzz.trapmf(income.universe, [60, 80, 100, 100])

applicant['low'] = fuzz.trapmf(applicant.universe, [0, 0, 2, 4])
applicant['medium'] = fuzz.trimf(applicant.universe, [2, 5, 8])
applicant['high'] = fuzz.trapmf(applicant.universe, [6, 8, 10, 10])

assets.view()

income.view()

applicant.view()

ae_rule1 = ctrl.Rule(assets['low'] & income['low'], applicant['low'])
ae_rule2 = ctrl.Rule(assets['low'] & income['medium'], applicant['low'])
ae_rule3 = ctrl.Rule(assets['low'] & income['high'], applicant['medium'])
ae_rule4 = ctrl.Rule(assets['low'] & income['very_high'], applicant['high'])

ae_rule5 = ctrl.Rule(assets['medium'] & income['low'], applicant['low'])
ae_rule6 = ctrl.Rule(assets['medium'] & income['medium'], applicant['medium'])
ae_rule7 = ctrl.Rule(assets['medium'] & income['high'], applicant['high'])
ae_rule8 = ctrl.Rule(assets['medium'] & income['very_high'], applicant['high'])

ae_rule9 = ctrl.Rule(assets['high'] & income['low'], applicant['medium'])
ae_rule10 = ctrl.Rule(assets['high'] & income['medium'], applicant['medium'])
ae_rule11 = ctrl.Rule(assets['high'] & income['high'], applicant['high'])
ae_rule12 = ctrl.Rule(assets['high'] & income['very_high'], applicant['high'])

ae_Rules = [ae_rule1, ae_rule2, ae_rule3, ae_rule4, ae_rule5, ae_rule6, ae_rule7, ae_rule8,
ae_rule9, ae_rule10, ae_rule11, ae_rule12]
applicant_ctrl = ctrl.ControlSystem(ae_Rules)
applicant_eval = ctrl.ControlSystemSimulation(applicant_ctrl)
```

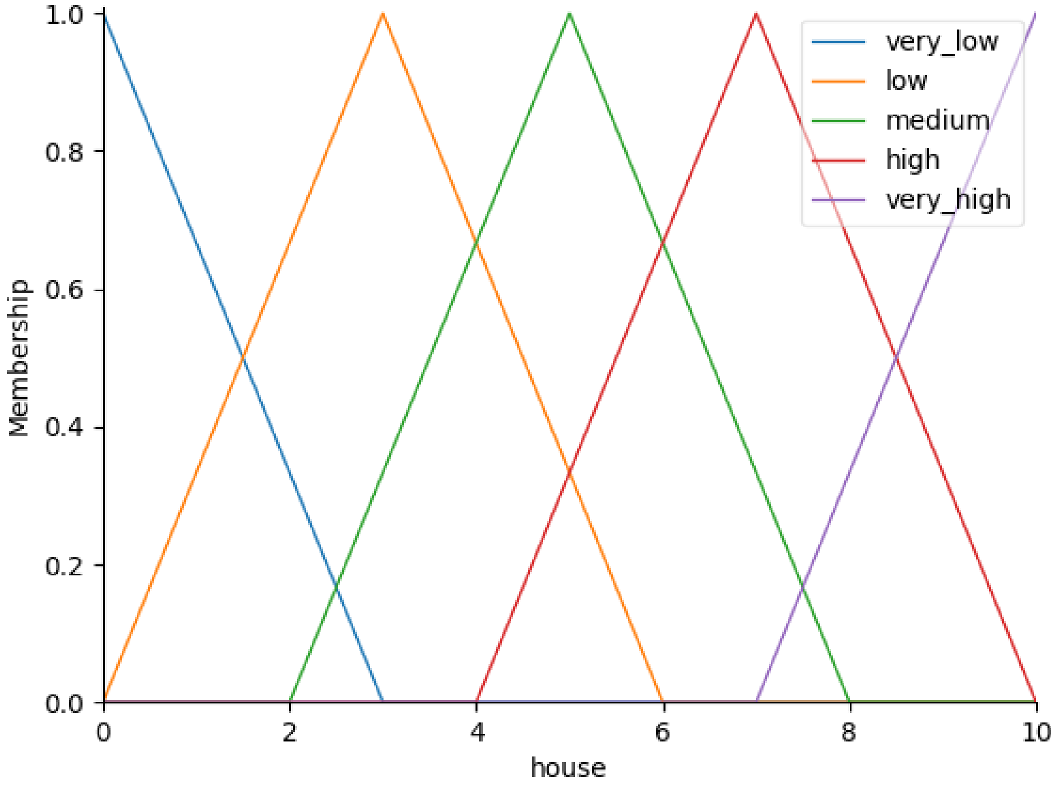
```
income =  ctrl.Antecedent(np.arange(0, 101, 1), 'income')
interest =  ctrl.Antecedent(np.arange(0, 11, 0.5), 'interest')
applicant = ctrl.Antecedent(np.arange(0, 11, 1), 'applicant')
house = ctrl.Antecedent(np.arange(0, 11, 1), 'house')
credit = ctrl.Consequent(np.arange(0, 11, 1), 'credit')

income['low'] = fuzz.trapmf(income.universe, [0, 0, 10, 25])
income['medium'] = fuzz.trimf(income.universe, [15, 35, 55])
income['high'] = fuzz.trimf(income.universe, [40, 60, 80])
income['very_high'] = fuzz.trapmf(income.universe, [60, 80, 100, 100])

interest['low'] = fuzz.trapmf(interest.universe, [0, 0, 2, 5])
interest['medium'] = fuzz.trapmf(interest.universe, [2, 4, 6, 8])
interest['high'] = fuzz.trapmf(interest.universe, [6, 8.5, 10, 10])

applicant['low'] = fuzz.trapmf(applicant.universe, [0, 0, 2, 4])
applicant['medium'] = fuzz.trimf(applicant.universe, [2, 5, 8])
applicant['high'] = fuzz.trapmf(applicant.universe, [6, 8, 10, 10])

house['very_low'] = fuzz.trimf(house.universe, [0, 0, 3])
house['low'] = fuzz.trimf(house.universe, [0, 3, 6])
house['medium'] = fuzz.trimf(house.universe, [2, 5, 8])
house['high'] = fuzz.trimf(house.universe, [4, 7, 10])
house['very_high'] = fuzz.trimf(house.universe, [7, 10, 10])

credit['very_low'] = fuzz.trimf(credit.universe, [0, 0, 3])
credit['low'] = fuzz.trimf(credit.universe, [0, 3, 6])
credit['medium'] = fuzz.trimf(credit.universe, [2, 5, 8])
credit['high'] = fuzz.trimf(credit.universe, [4, 7, 10])
credit['very_high'] = fuzz.trimf(credit.universe, [7, 10, 10])

income.view()

interest.view()

applicant.view()

house.view()

credit.view()

ce_rule1 = ctrl.Rule(income['low'] & interest['medium'], credit['very_low'])
ce_rule2 = ctrl.Rule(income['low'] & interest['high'], credit['very_low'])
ce_rule3 = ctrl.Rule(income['medium'] & interest['high'], credit['low'])
ce_rule4 = ctrl.Rule(applicant['low'], credit['very_low'])
ce_rule5 = ctrl.Rule(house['very_low'], credit['very_low'])

ce_rule6 = ctrl.Rule(applicant['medium'] & house['very_low'], credit['low'])
ce_rule7 = ctrl.Rule(applicant['medium'] & house['low'], credit['low'])
ce_rule8 = ctrl.Rule(applicant['medium'] & house['medium'], credit['medium'])
ce_rule9 = ctrl.Rule(applicant['medium'] & house['high'], credit['high'])
```

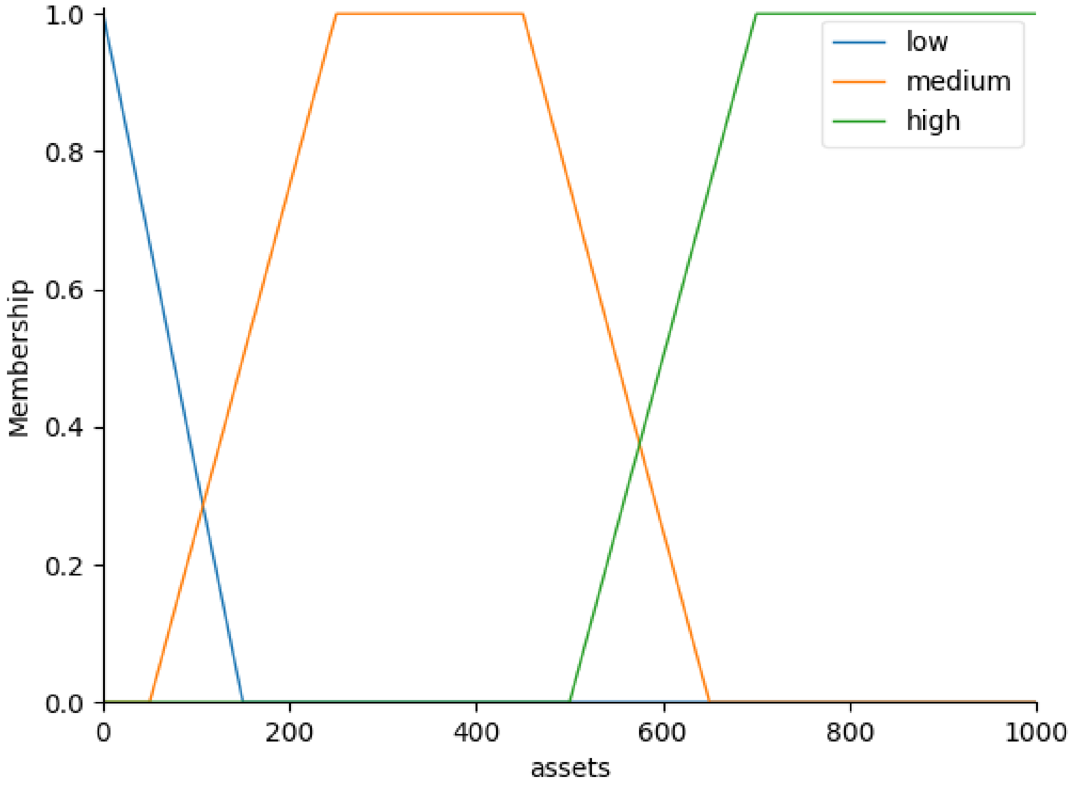
```
ce_rule10 = ctrl.Rule(applicant['medium'] & house['very_high'], credit['high'])

ce_rule11 = ctrl.Rule(applicant['high'] & house['very_low'], credit['low'])
ce_rule12 = ctrl.Rule(applicant['high'] & house['low'], credit['medium'])
ce_rule13 = ctrl.Rule(applicant['high'] & house['medium'], credit['high'])
ce_rule14 = ctrl.Rule(applicant['high'] & house['high'], credit['high'])
ce_rule15 = ctrl.Rule(applicant['high'] & house['very_high'], credit['very_high'])

ce_Rules = [ce_rule1, ce_rule2, ce_rule3, ce_rule4, ce_rule5, ce_rule6, ce_rule7, ce_rule8,
ce_rule9, ce_rule10, ce_rule11, ce_rule12, ce_rule13, ce_rule14, ce_rule15]
credit_ctrl = ctrl.ControlSystem(ce_Rules)
credit_eval = ctrl.ControlSystemSimulation(credit_ctrl)

house_eval.input['market_value'] = 60      # 0 to 100
house_eval.input['location'] = 7           # 0 to 10
applicant_eval.input['assets'] = 400       # 0 to 1000
applicant_eval.input['income'] = 70        # 0 to 100

house_eval.compute()
applicant_eval.compute()

credit_eval.input['income'] = 70           # 0 to 100
credit_eval.input['interest'] = 7          # 0 to 10
credit_eval.input['applicant'] = applicant_eval.output['applicant']
credit_eval.input['house'] = house_eval.output['house']

credit_eval.compute()
credit_eval.output['credit']

credit.view(sim=credit_eval)

plt.show()
```

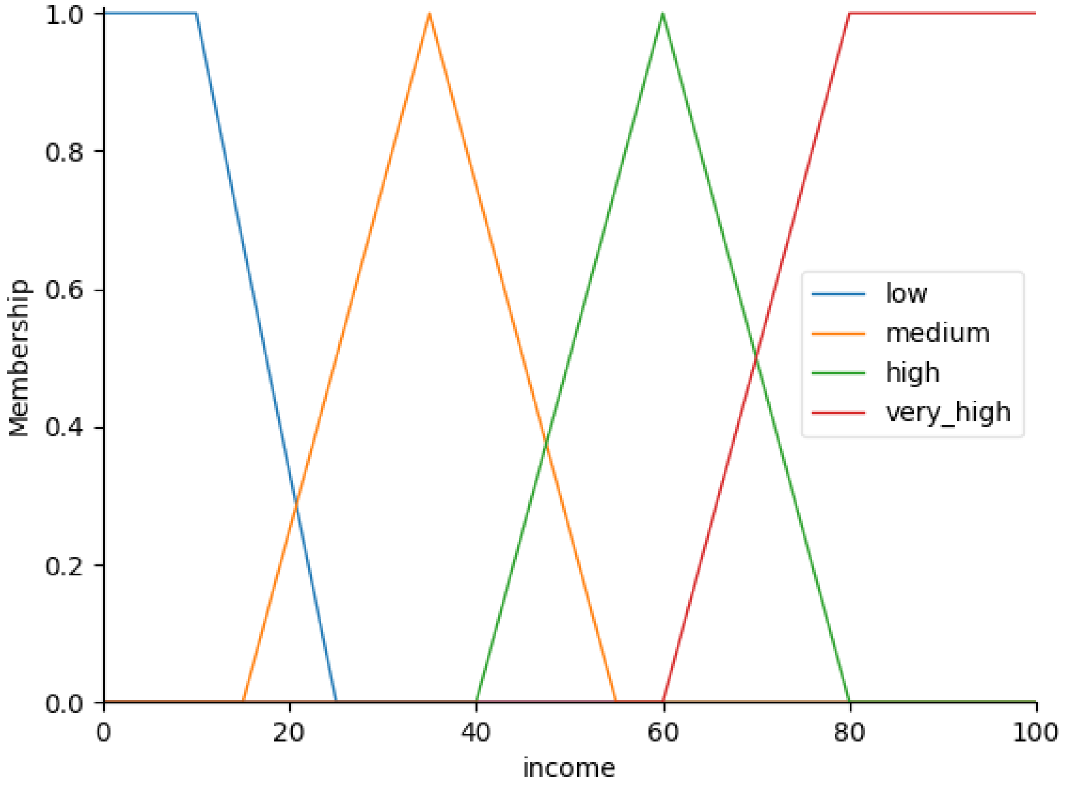
Output:

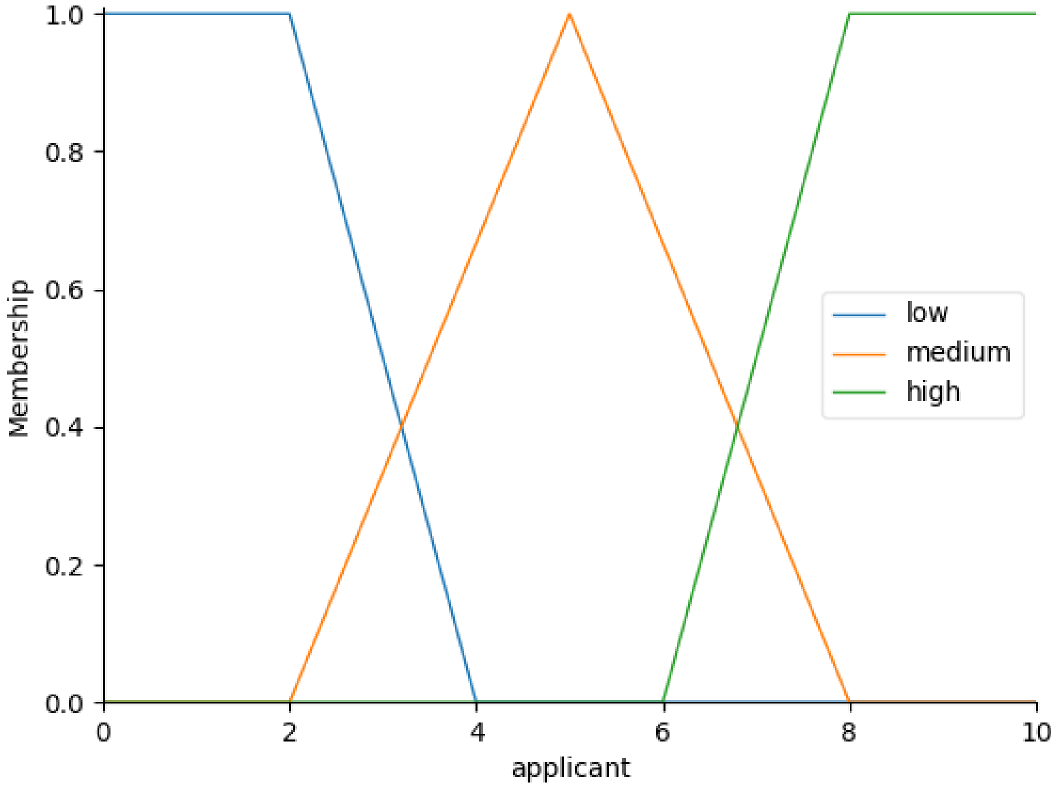


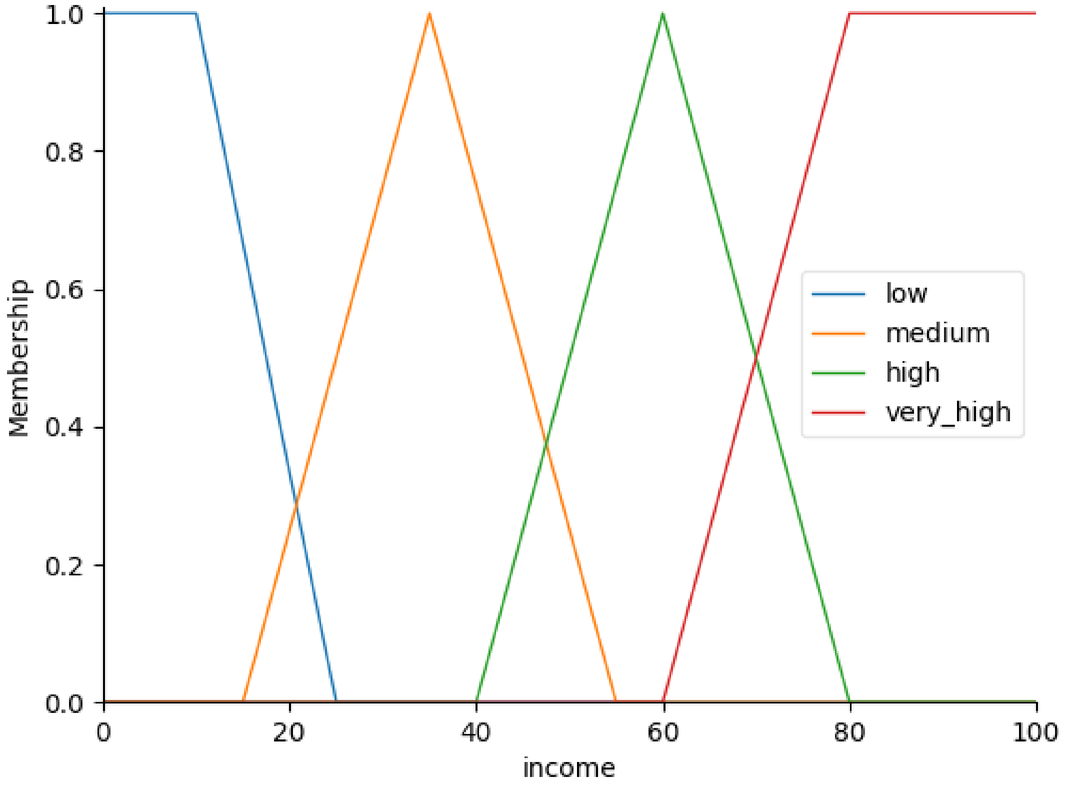

1.0
0.8
0.6
0.4
0.2
0.0
Membership
0
2
4
6
8
10
location
bad
fair
excellent

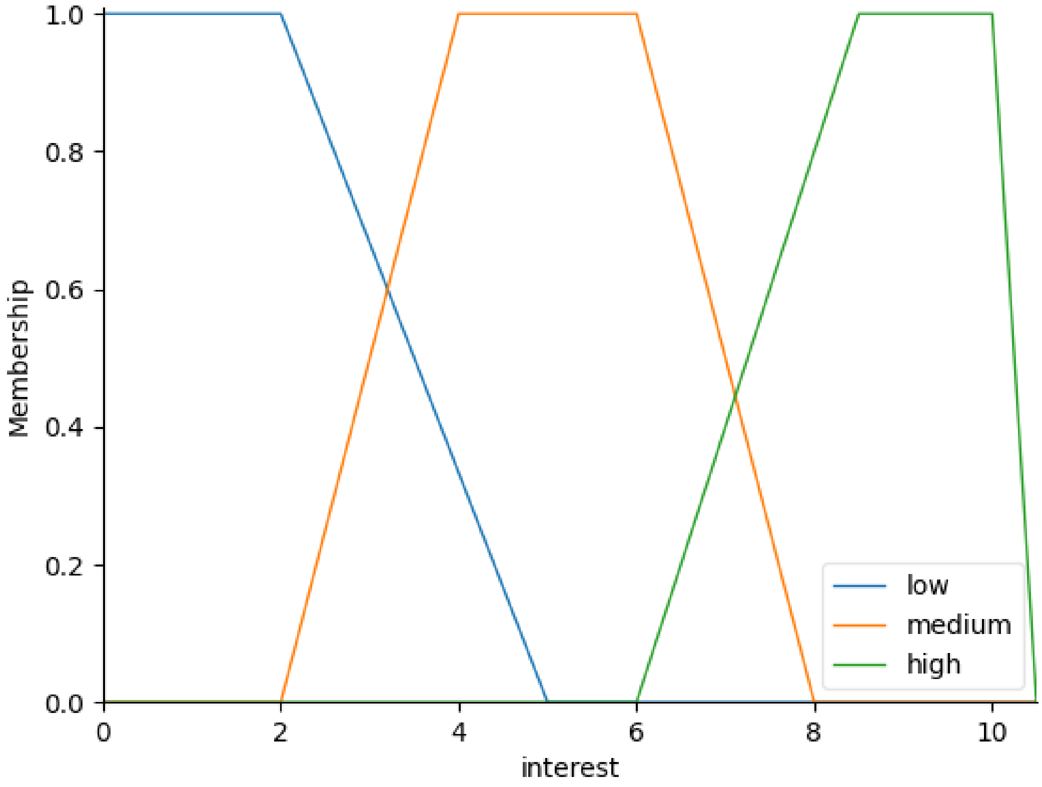

1.0
0.8
0.6
0.4
0.2
0.0
Membership
0
2
4
6
8
10
house
very_low
low
medium
high
very_high

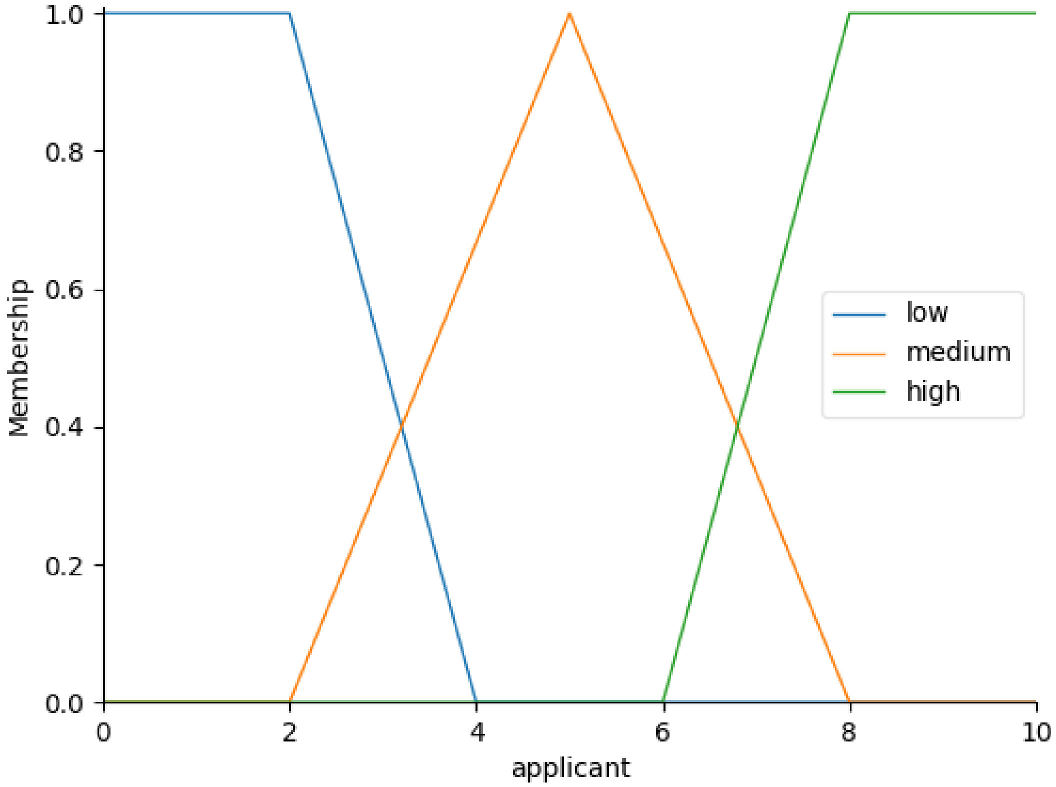

1.0
0.8
0.6
0.4
0.2
0.0
Membership
0
200
400
600
800
1000
assets
low
medium
high

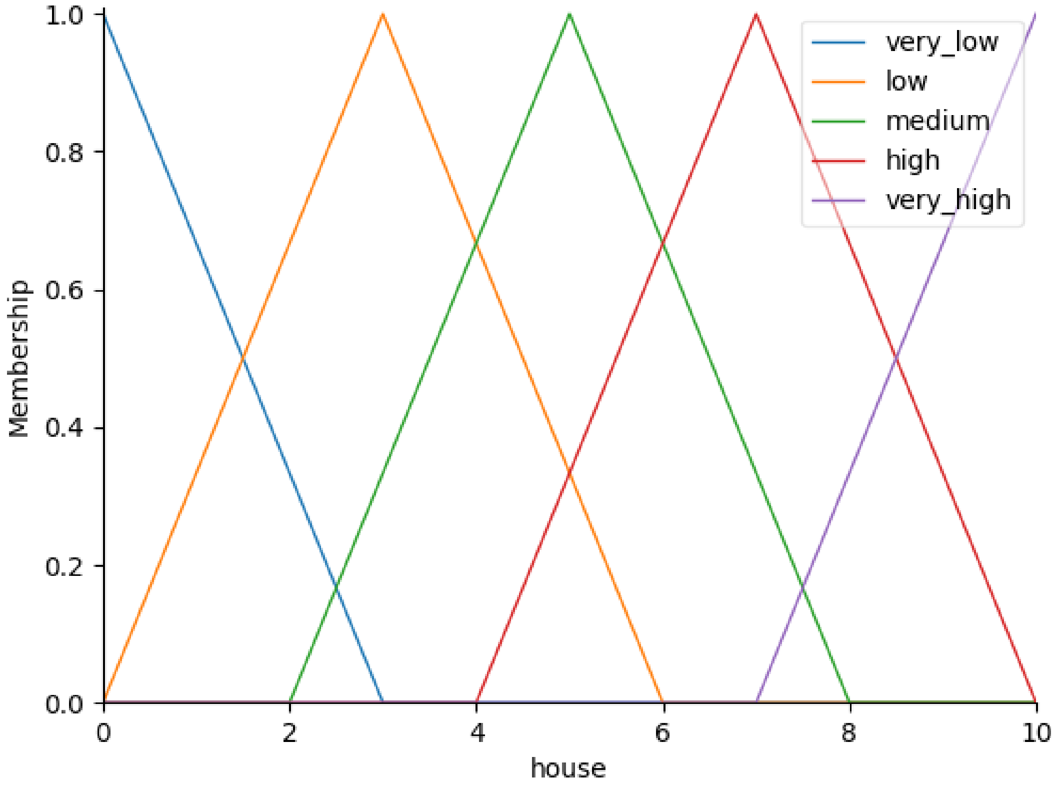

1.0
0.8
0.6
0.4
0.2
0.0
Membership
0
20
40
60
80
100
income
low
medium
high
very_high

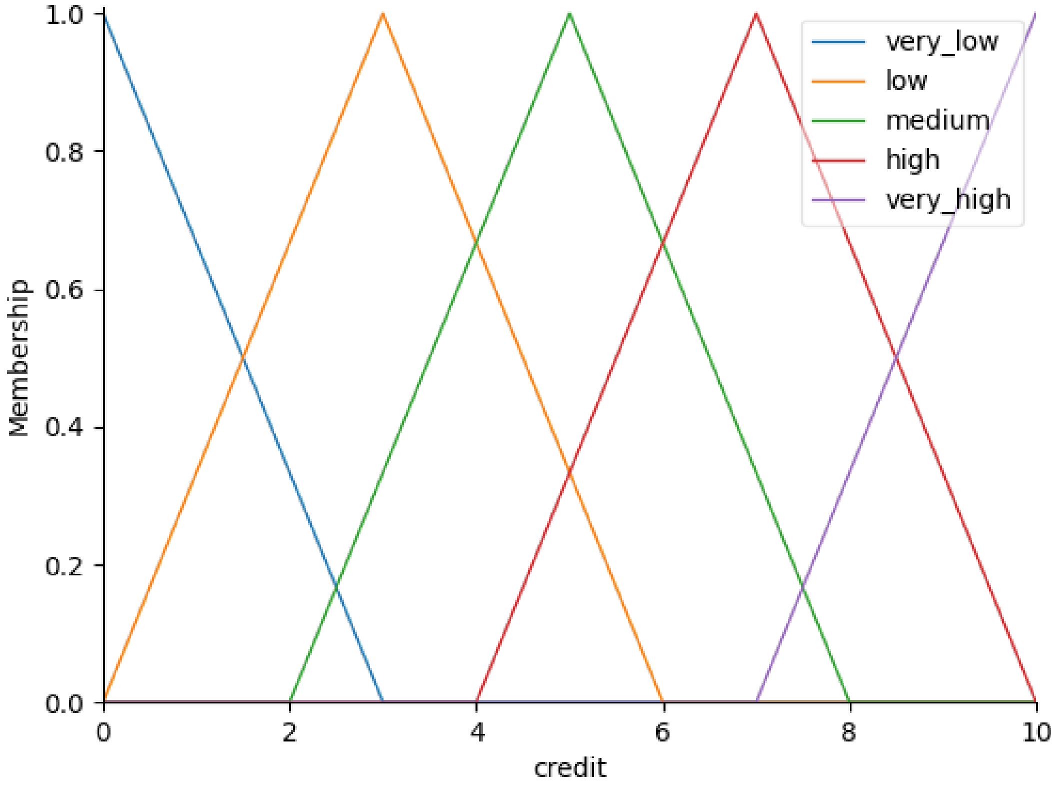

1.0
0.8
0.6
0.4
0.2
0.0
Membership
0
2
4
6
8
10
applicant
low
medium
high

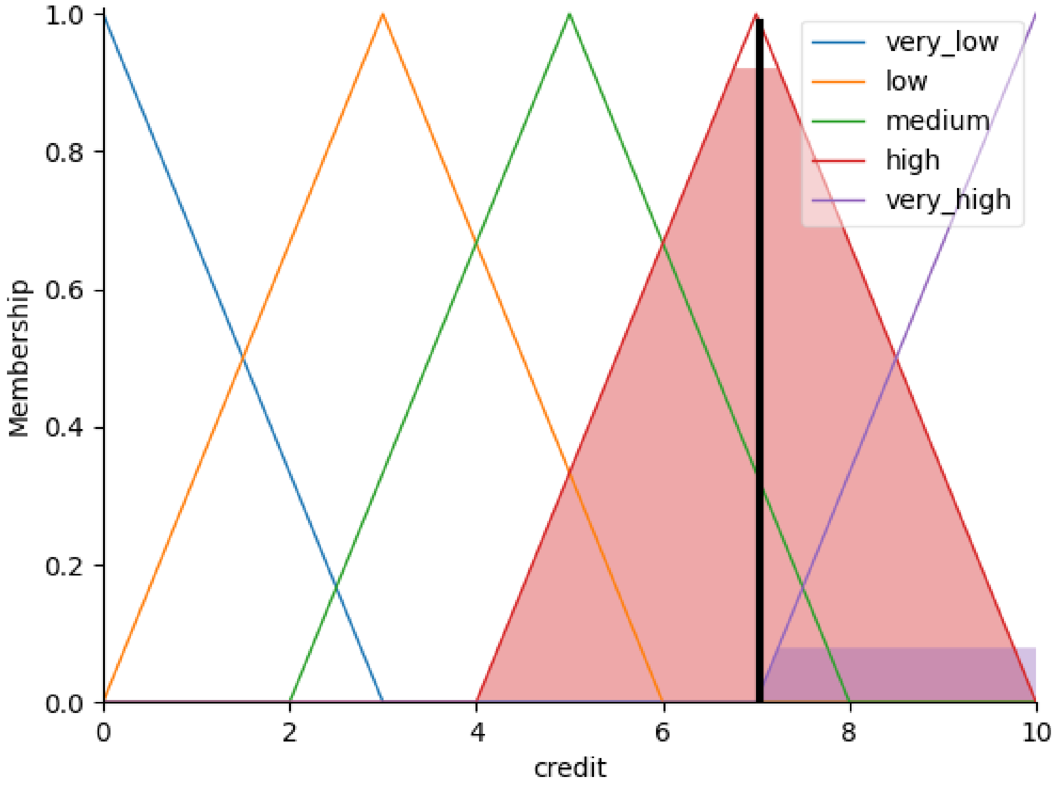

1.0
0.8
0.6
0.4
0.2
0.0
Membership
0
20
40
60
80
100
income
low
medium
high
very_high


1.0
0.8
0.6
0.4
0.2
0.0
Membership
0
2
4
6
8
10
interest
low
medium
high


1.0
0.8
0.6
0.4
0.2
0.0
Membership
0
2
4
6
8
10
applicant
low
medium
high


1.0
0.8
0.6
0.4
0.2
0.0
Membership
0
2
4
6
8
10
house
very_low
low
medium
high
very_high


1.0
0.8
0.6
0.4
0.2
0.0
Membership
0
2
4
6
8
10
credit
very_low
low
medium
high
very_high

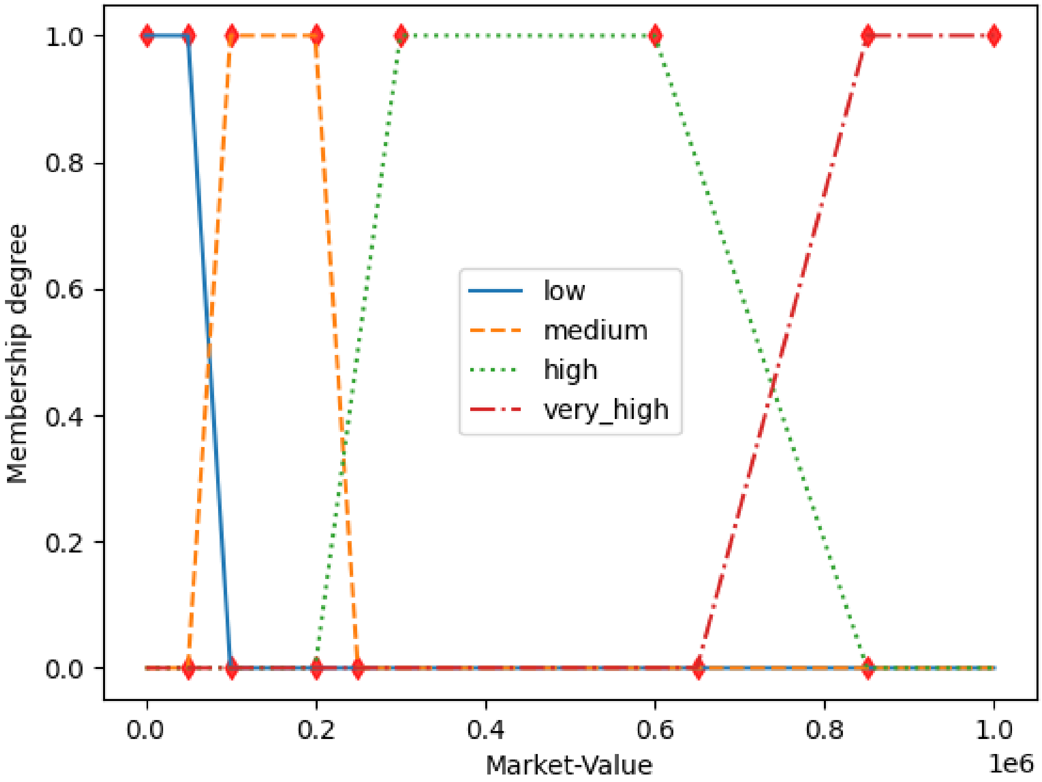


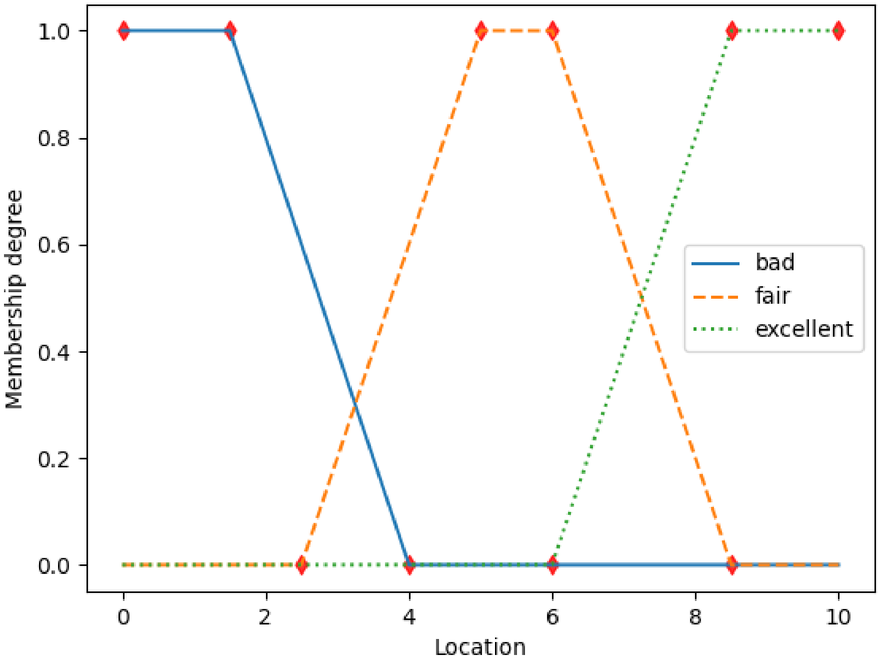
```
from simpful import *
FS = FuzzySystem()

S_1 = FuzzySet(points=[[0., 1.],[50000,1],  [100000., 0.]], term="low")
S_2 = FuzzySet(points=[[50000., 0.], [100000., 1.],[200000., 1.], [250000., 0.]], term="medium")
S_3 = FuzzySet(points=[[200000., 0.],[300000,1],[600000,1],  [850000., 0.]], term="high")
S_4 = FuzzySet(points=[[650000., 0.],  [850000., 1.], [1000000,1]], term="very_high")
FS.add_linguistic_variable("MarketValue", LinguisticVariable([S_1, S_2, S_3, S_4],
concept="Market-Value"))
FS.plot_variable("MarketValue")

S_1 = FuzzySet(points=[[0., 1.],[1.5,1],  [4., 0.]], term="bad")
S_2 = FuzzySet(points=[[2.5, 0.], [5., 1.],[6., 1.], [8.5, 0.]], term="fair")
S_3 = FuzzySet(points=[[6., 0.],[8.5,1],[10,1]], term="excellent")
FS.add_linguistic_variable("Location", LinguisticVariable([S_1, S_2, S_3], concept="Location"))
FS.plot_variable("Location")

# Define output crisp values
FS.set_crisp_output_value("very_low", 1)
FS.set_crisp_output_value("low", 3)
FS.set_crisp_output_value("medium", 5)
FS.set_crisp_output_value("high", 7)
FS.set_crisp_output_value("very_high", 9)

R1 = "IF (MarketValue IS low) THEN (House IS low)"
```

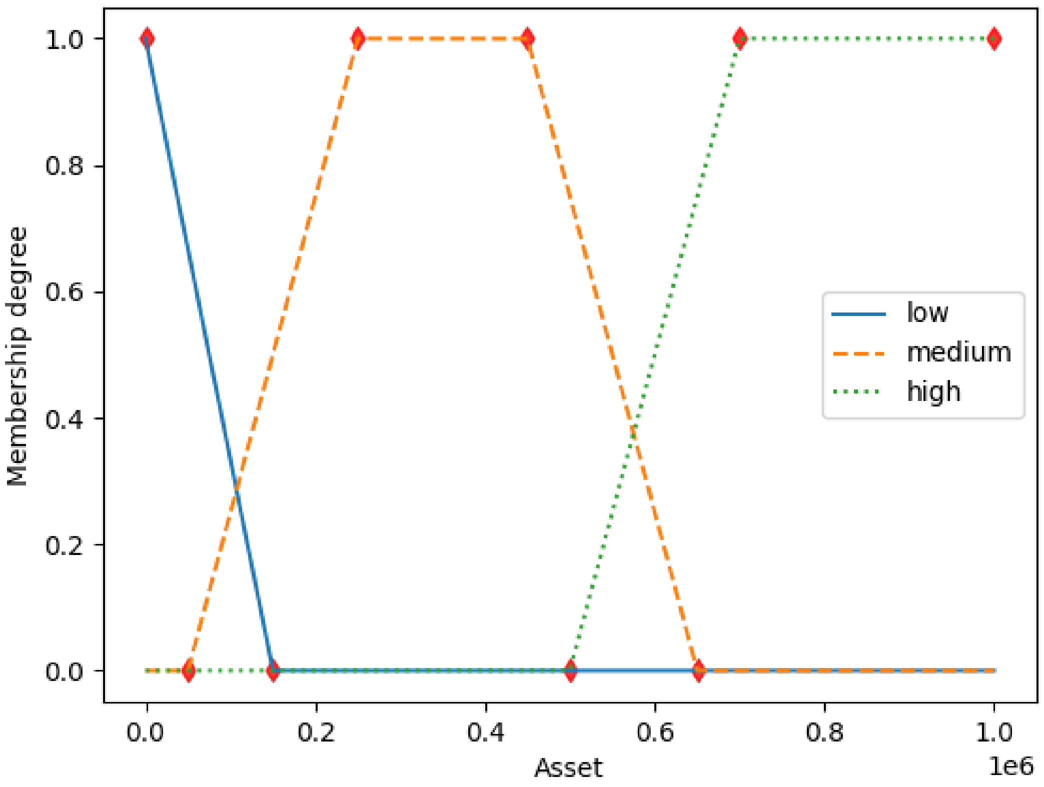
```
R2 = "IF (Location IS bad) THEN (House IS low)"
R3 = "IF (Location IS bad) AND (MarketValue IS low) THEN (House IS very_low)"
R4 = "IF (Location IS bad) AND (MarketValue IS medium) THEN (House IS low)"
R5 = "IF (Location IS bad) AND (MarketValue IS high) THEN (House IS medium)"
R6 = "IF (Location IS bad) AND (MarketValue IS very_high) THEN (House IS high)"
R7 = "IF (Location IS fair) AND (MarketValue IS low) THEN (House IS low)"
R8 = "IF (Location IS fair) AND (MarketValue IS medium) THEN (House IS medium)"
R9 = "IF (Location IS fair) AND (MarketValue IS high) THEN (House IS high)"
R10= "IF (Location IS fair) AND (MarketValue IS very_high) THEN (House IS very_high)"
R11= "IF (Location IS excellent) AND (MarketValue IS low) THEN (House IS medium)"
R12= "IF (Location IS excellent) AND (MarketValue IS medium) THEN (House IS high)"
R13= "IF (Location IS excellent) AND (MarketValue IS high) THEN (House IS very_high)"
R14= "IF (Location IS excellent) AND (MarketValue IS very_high) THEN (House IS very_high)"
FS.add_rules([R1, R2, R3, R4, R5, R6, R7, R9, R10, R11, R12, R13, R14, R8])

# Set antecedents values
FS.set_variable("MarketValue", 146000)
FS.set_variable("Location", 7.2)

# Perform Sugeno inference and print output
print(FS.Sugeno_inference(["House"]))

FS2 = FuzzySystem()

# Define fuzzy sets and linguistic variables
S_1 = FuzzySet(points=[[0., 1.],[150000,0]], term="low")
S_2 = FuzzySet(points=[[50000., 0.], [250000., 1.],[450000., 1.], [650000., 0.]], term="medium")
S_3 = FuzzySet(points=[[500000., 0.],[700000,1],[1000000,1]], term="high")
FS2.add_linguistic_variable("Asset", LinguisticVariable([S_1, S_2, S_3], concept="Asset"))
FS2.plot_variable("Asset")

S_1 = FuzzySet(points=[[0., 1.],[12500,1],  [25000., 0.]], term="low")
S_2 = FuzzySet(points=[[15000, 0.], [35000., 1.],[55000., 0.]], term="medium")
S_3 = FuzzySet(points=[[40000., 0.],[60000,1],[80000,0]], term="high")
S_4 = FuzzySet(points=[[60000., 0.],[80000,1],[100000,1]], term="very_high")
FS2.add_linguistic_variable("Income", LinguisticVariable([S_1, S_2, S_3, S_4], concept="Income"))
FS2.plot_variable("Income")

# Define output crisp values
FS2.set_crisp_output_value("low", 1.45)
FS2.set_crisp_output_value("medium", 5)
FS2.set_crisp_output_value("high", 8.55)

R1 = "IF (Asset IS low) AND (Income IS low) THEN (Applicant IS low)"
R2 = "IF (Asset IS low) AND (Income IS medium) THEN (Applicant IS low)"
R3 = "IF (Asset IS low) AND (Income IS high) THEN (Applicant IS medium)"
R4 = "IF (Asset IS low) AND (Income IS very_high) THEN (Applicant IS high)"
R5 = "IF (Asset IS medium) AND (Income IS low) THEN (Applicant IS low)"
R6 = "IF (Asset IS medium) AND (Income IS medium) THEN (Applicant IS medium)"
R7 = "IF (Asset IS medium) AND (Income IS high) THEN (Applicant IS high)"
R8 = "IF (Asset IS medium) AND (Income IS very_high) THEN (Applicant IS high)"
```

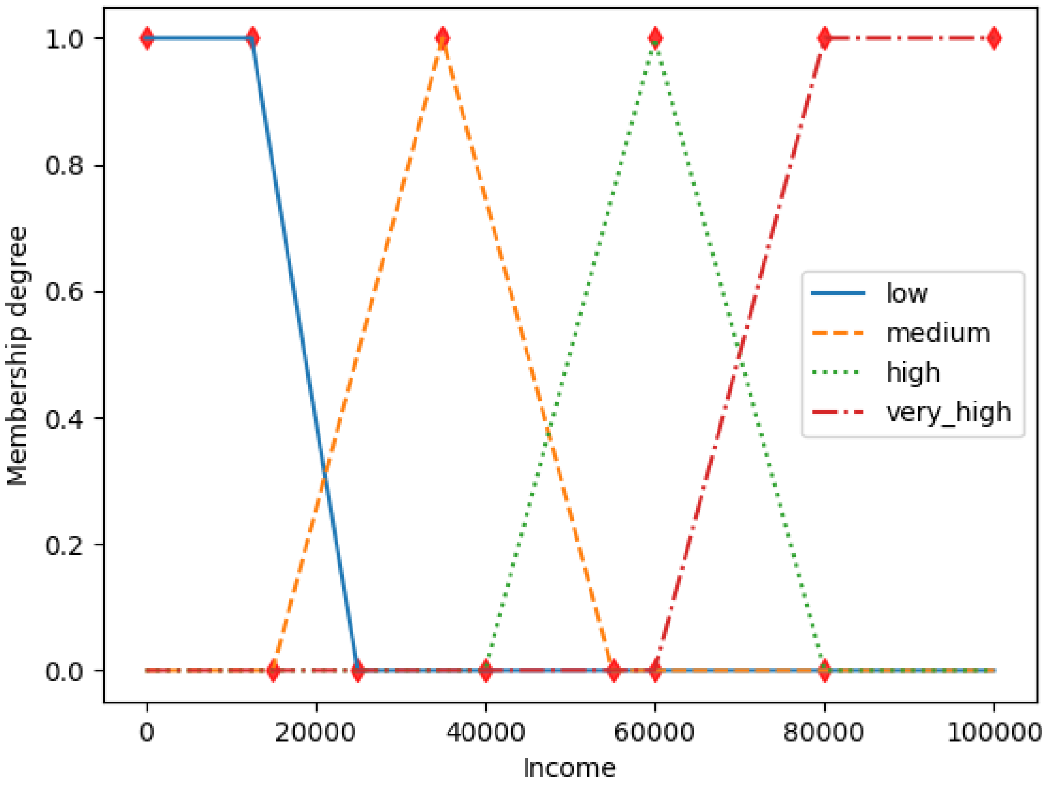
```
R9 = "IF (Asset IS high) AND (Income IS low) THEN (Applicant IS medium)"
R10= "IF (Asset IS high) AND (Income IS medium) THEN (Applicant IS medium)"
R11= "IF (Asset IS high) AND (Income IS high) THEN (Applicant IS high)"
R12= "IF (Asset IS high) AND (Income IS very_high) THEN (Applicant IS high)"
FS2.add_rules([R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12])

# Set antecedents values
FS2.set_variable("Asset", 520000)
FS2.set_variable("Income", 60000)

# Perform Sugeno inference and print output
print(FS2.Sugeno_inference(["Applicant"]))

# A simple fuzzy inference system for the tipping problem
# Create a fuzzy system object
FS3 = FuzzySystem()

S_1 = FuzzySet(points=[[0., 1.],[12500,1],  [25000., 0.]], term="low")
S_2 = FuzzySet(points=[[15000, 0.], [35000., 1.],[55000., 0.]], term="medium")
S_3 = FuzzySet(points=[[40000., 0.],[60000,1],[80000,0]], term="high")
S_4 = FuzzySet(points=[[60000., 0.],[80000,1],[100000,1]], term="very_high")
FS3.add_linguistic_variable("Income", LinguisticVariable([S_1, S_2, S_3, S_4], concept="Income"))
FS3.plot_variable("Income")

S_1 = FuzzySet(points=[[0., 1.],[2,1],  [5., 0.]], term="low")
S_2 = FuzzySet(points=[[2, 0.], [4., 1.], [6,1], [8., 0.]], term="medium")
S_3 = FuzzySet(points=[[6., 0.],[8.5,1],[10,1]], term="high")
FS3.add_linguistic_variable("Interest", LinguisticVariable([S_1, S_2, S_3], concept="Interest"))
FS3.plot_variable("Interest")

S_1 = FuzzySet(points=[[0., 1.],[2,1],  [4., 0.]], term="low")
S_2 = FuzzySet(points=[[2, 0.], [5., 1.], [8., 0.]], term="medium")
S_3 = FuzzySet(points=[[6., 0.],[8,1],[10,1]], term="high")
FS3.add_linguistic_variable("Applicant", LinguisticVariable([S_1, S_2, S_3], concept="Applicant"))
FS3.plot_variable("Applicant")

S_1 = FuzzySet(points=[[0., 1.],[3,0]], term="very_low")
S_2 = FuzzySet(points=[[0, 0.], [3., 1.],[6., 0.]], term="low")
S_3 = FuzzySet(points=[[2., 0.],[5,1],[8,0]], term="medium")
S_4 = FuzzySet(points=[[4., 0.],[7,1],[10,0]], term="high")
S_5 = FuzzySet(points=[[7., 0.],[10,1]], term="very_high")
FS3.add_linguistic_variable("House", LinguisticVariable([S_1, S_2, S_3, S_4,S_5],
concept="House"))
FS3.plot_variable("House")

# Define output crisp values

FS3.set_crisp_output_value("very_low", 1)
FS3.set_crisp_output_value("low", 3)
FS3.set_crisp_output_value("medium", 5)
FS3.set_crisp_output_value("high", 7)
```

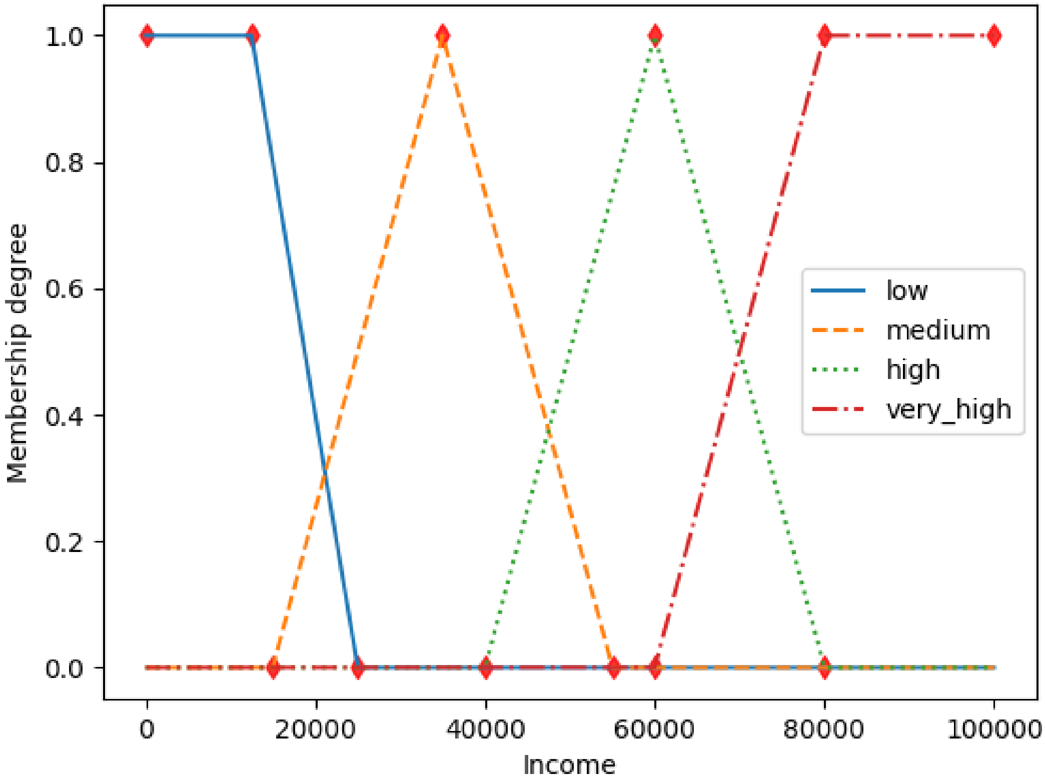
```
FS3.set_crisp_output_value("very_high", 9)

R1 = "IF (Income IS low) AND (Interest IS medium) THEN (Credit IS very_low)"
R2 = "IF (Income IS low) AND (Interest IS high) THEN (Credit IS very_low)"
R3 = "IF (Income IS medium) AND (Interest IS high) THEN (Credit IS low)"
R4 = "IF (Applicant IS low) THEN (Credit IS very_low)"
R5 = "IF (House IS very_low) THEN (Credit IS very_low)"
R6 = "IF (Applicant IS medium) AND (House IS very_low) THEN (Credit IS low)"
R7 = "IF (Applicant IS medium) AND (House IS low) THEN (Credit IS low)"
R8 = "IF (Applicant IS medium) AND (House IS medium) THEN (Credit IS medium)"
R9 = "IF (Applicant IS medium) AND (House IS high) THEN (Credit IS high)"
R10= "IF (Applicant IS medium) AND (House IS very_high) THEN (Credit IS high)"
R11= "IF (Applicant IS high) AND (House IS very_low) THEN (Credit IS low)"
R12= "IF (Applicant IS high) AND (House IS low) THEN (Credit IS medium)"
R13= "IF (Applicant IS high) AND (House IS medium) THEN (Credit IS high)"
R14= "IF (Applicant IS high) AND (House IS high) THEN (Credit IS high)"
R15= "IF (Applicant IS high) AND (House IS very_high) THEN (Credit IS very_high)"
FS3.add_rules([R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15])

# Set antecedents values
FS3.set_variable("Income", 28500)
FS3.set_variable("Interest", 2.8)
FS3.set_variable("Applicant", 5.2)
FS3.set_variable("House", 2.4)

# Perform Sugeno inference and print output
print(FS3.Sugeno_inference(["Credit"]))
```

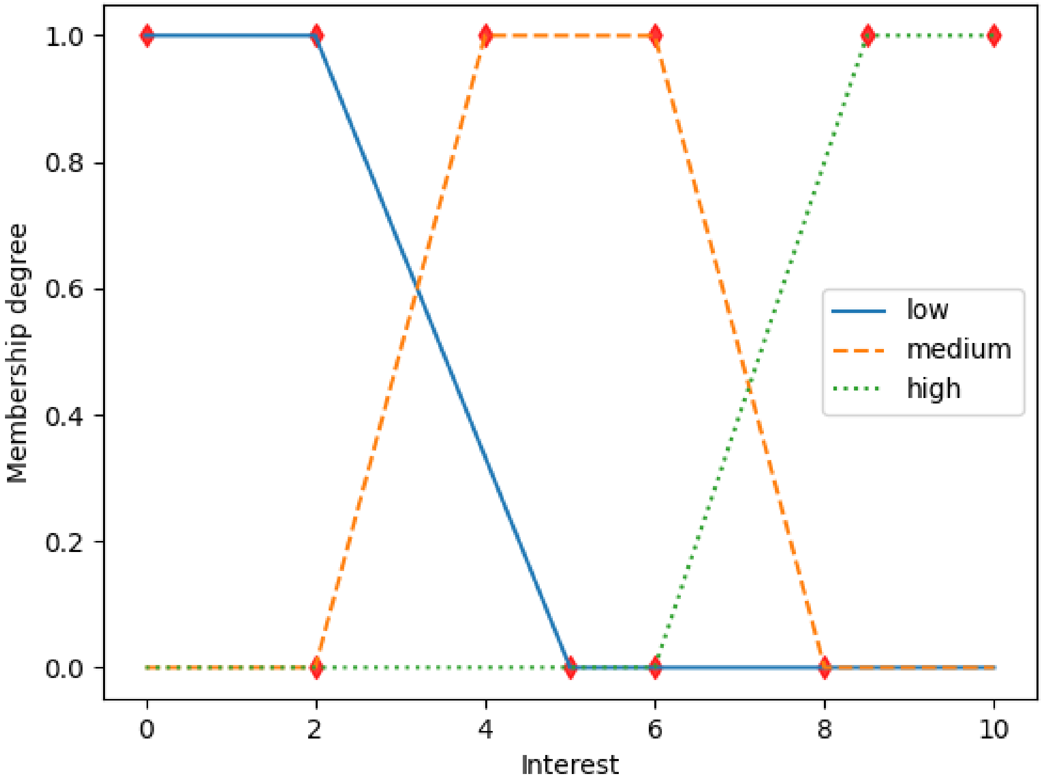

1.0
0.8
0.6
0.4
0.2
0.0
Membership degree
low
medium
high
very_high
0.0
0.2
0.4
0.6
0.8
1.0
Market-Value
1e6

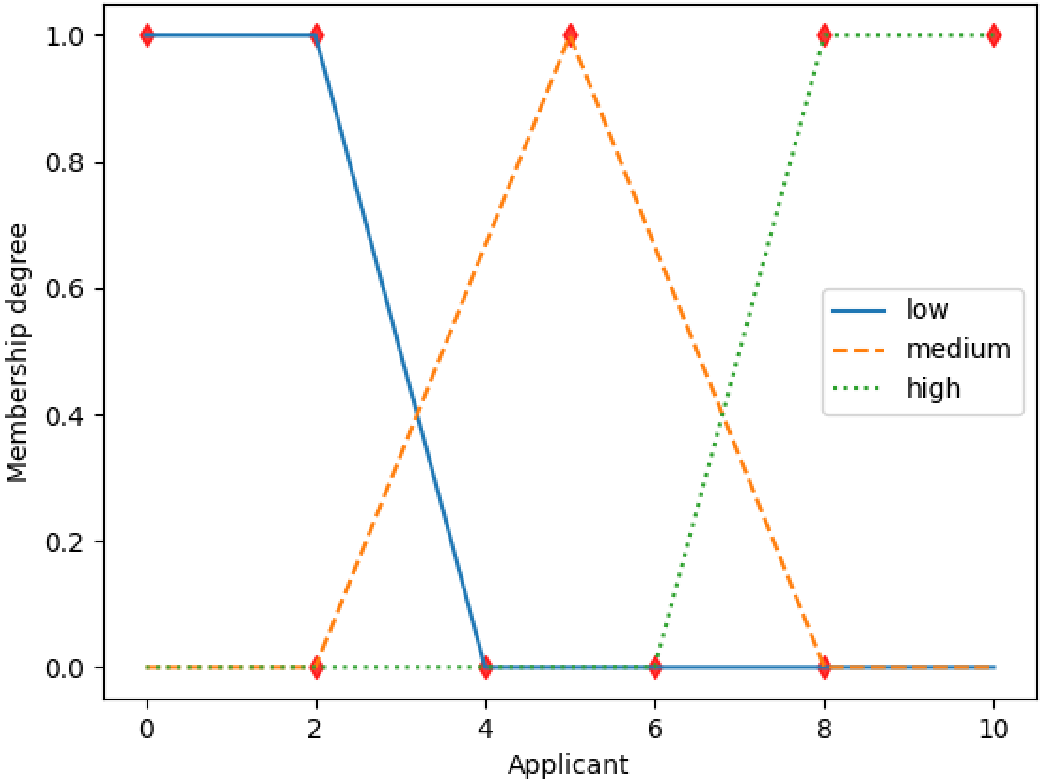

1.0
0.8
0.6
0.4
0.2
0.0
Membership degree
bad
fair
excellent
0
2
4
6
8
10
Location

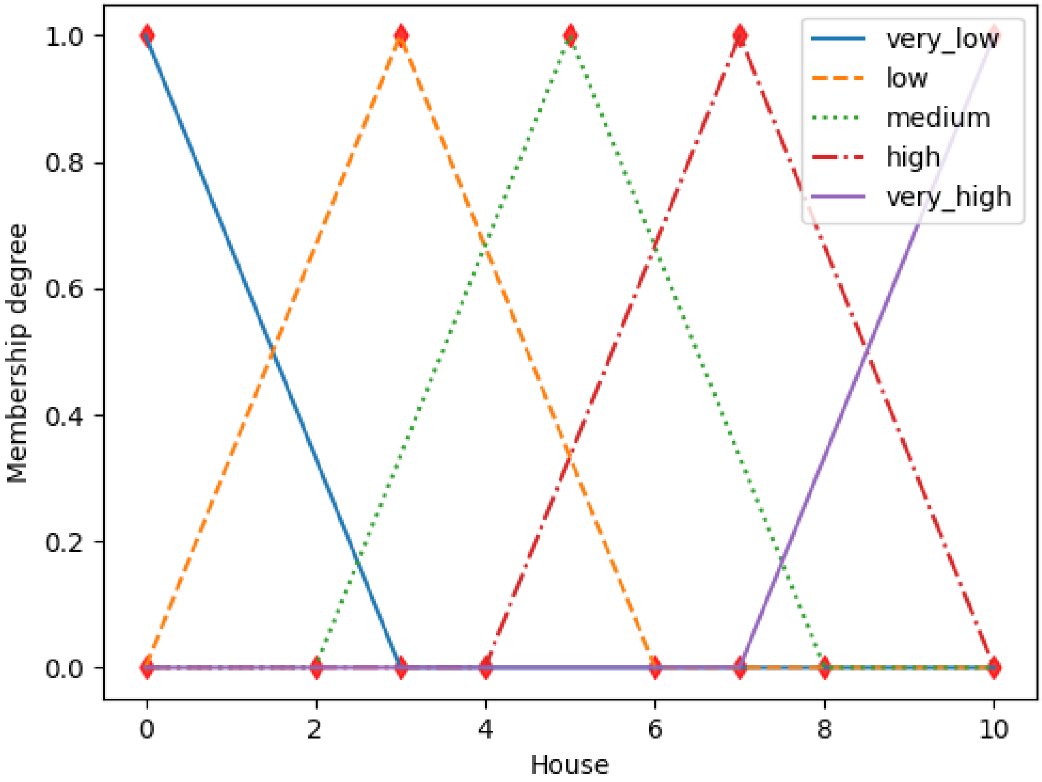

1.0
0.8
0.6
0.4
0.2
0.0
Membership degree
0.0
0.2
0.4
0.6
0.8
1.0
Asset
1e6
low
medium
high


1.0
0.8
0.6
0.4
0.2
0.0
Membership degree
0
20000
40000
60000
80000
100000
Income
low
medium
high
very_high


1.0
0.8
0.6
0.4
0.2
0.0
Membership degree
0
20000
40000
60000
80000
100000
Income
low
medium
high
very_high


1.0
0.8
0.6
0.4
0.2
0.0
Membership degree
0
2
4
6
8
10
Interest
low
medium
high

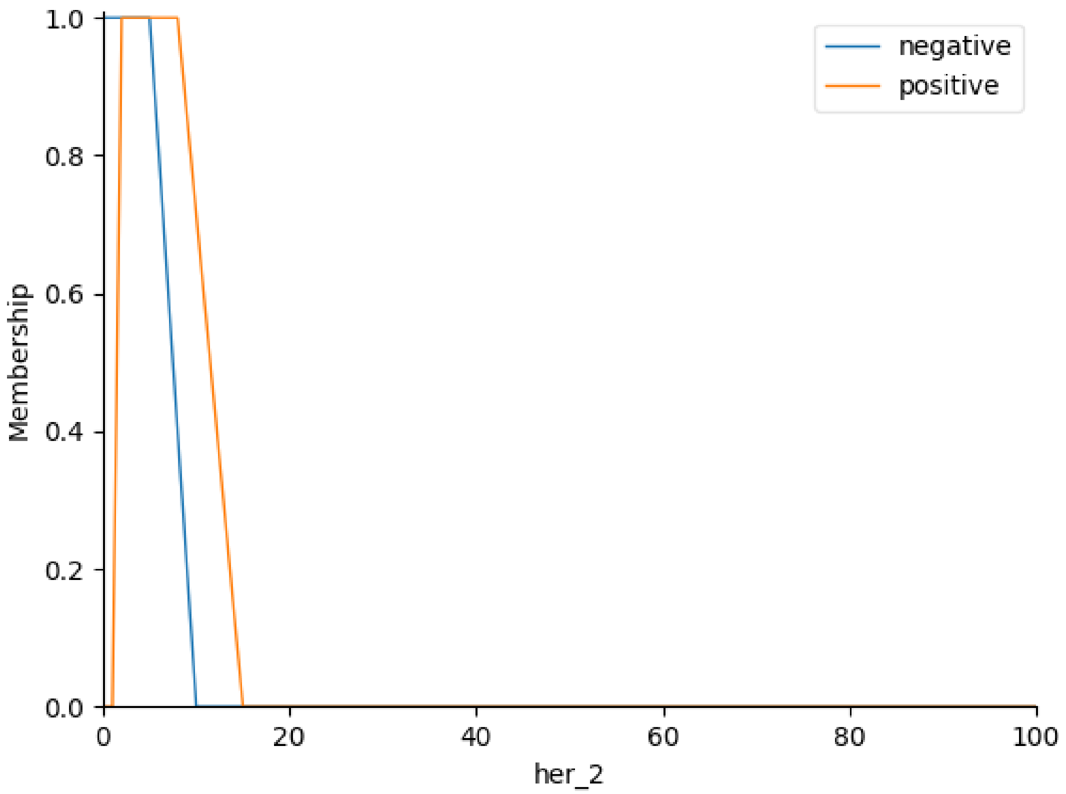

1.0
0.8
0.6
0.4
0.2
0.0
Membership degree
0
2
4
6
8
10
Applicant
low
medium
high

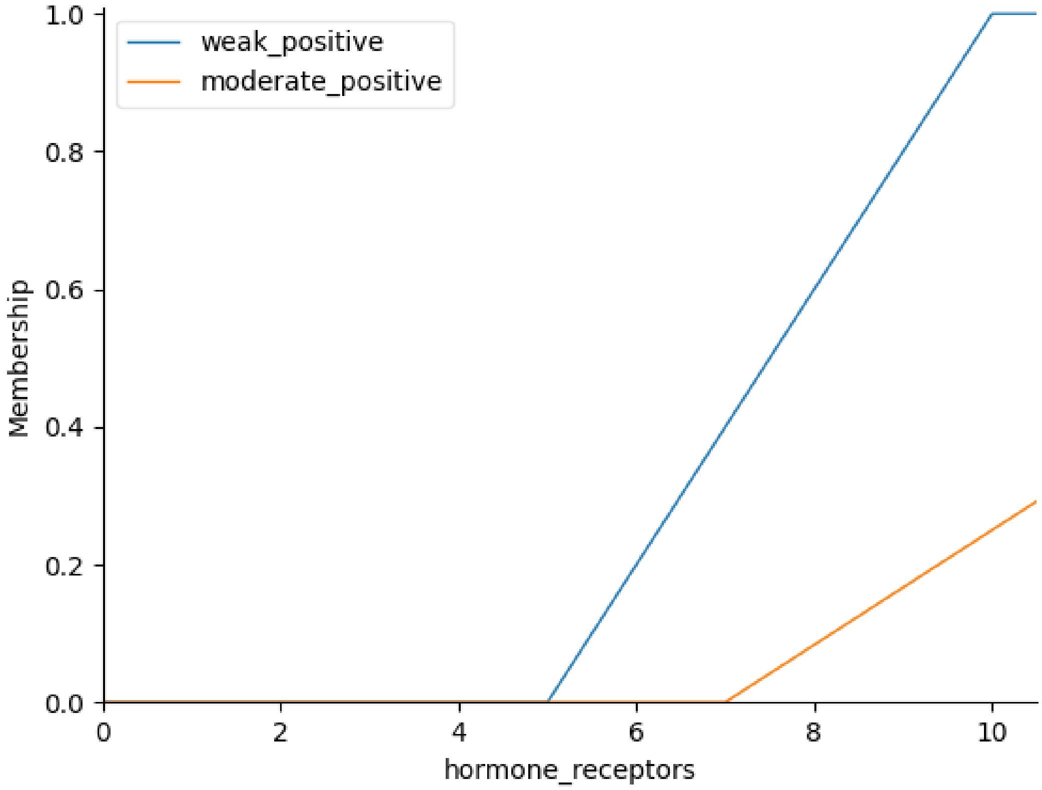

1.0
0.8
0.6
0.4
0.2
0.0
Membership degree
0
2
4
6
8
10
House
very_low
low
medium
high
very_high

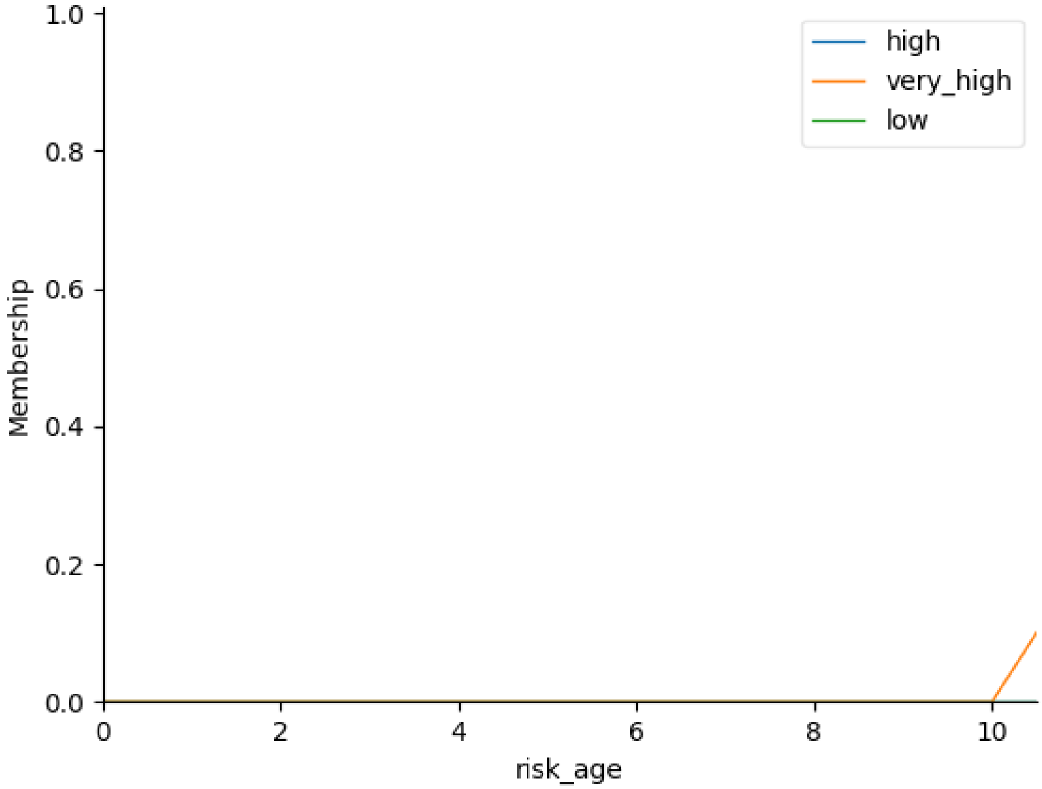
3.

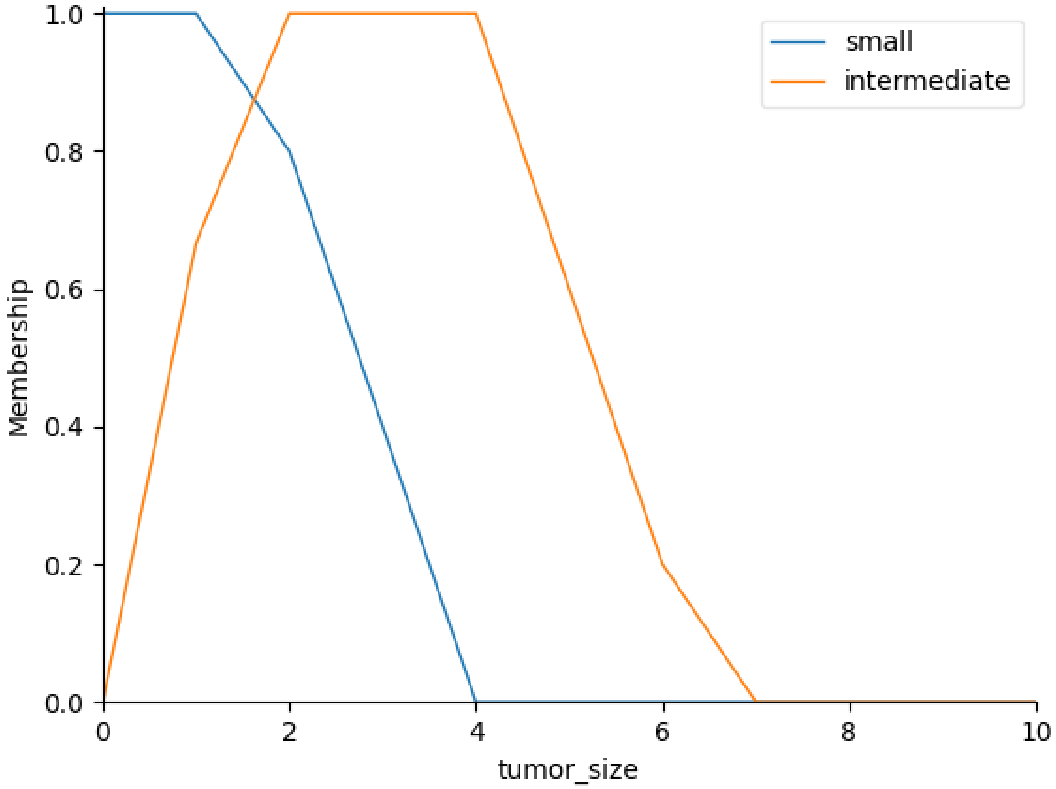
```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt

her_2 =  ctrl.Antecedent(np.arange(0, 101, 1), 'her_2')
hormone_receptors =  ctrl.Antecedent(np.arange(0, 11, 0.5), 'hormone_receptors')
risk_age =  ctrl.Antecedent(np.arange(0, 11, 0.5), 'risk_age')
grade = ctrl.Antecedent(np.arange(0, 11, 1), 'grade')
tumor_size = ctrl.Antecedent(np.arange(0, 11, 1), 'tumor_size')
lymph_node = ctrl.Antecedent(np.arange(0, 11, 1), 'lymph_node')

risk_status = ctrl.Consequent(np.arange(0, 11, 1), 'risk_status')

her_2['negative'] = fuzz.trapmf(her_2.universe, [0, 0, 5, 10])
her_2['positive'] = fuzz.trapmf(her_2.universe, [1, 2, 8, 15])

hormone_receptors['weak_positive'] = fuzz.trapmf(hormone_receptors.universe, [5, 10, 18, 25])
hormone_receptors['moderate_positive'] = fuzz.trapmf(hormone_receptors.universe, [7, 19, 21, 
40])

risk_age['high'] = fuzz.trapmf(risk_age.universe, [20, 25, 65, 85])
risk_age['very_high'] = fuzz.trapmf(risk_age.universe, [10, 15, 25, 45])
risk_age['low'] = fuzz.trapmf(risk_age.universe, [11,19,20,22])

grade['grade1'] = fuzz.trapmf(grade.universe, [65, 85, 100, 100])
grade['grade2'] = fuzz.trapmf(grade.universe, [35, 55, 70, 70])
grade['grade3'] = fuzz.trapmf(grade.universe, [12, 20, 60, 80])

tumor_size['small'] = fuzz.trapmf(tumor_size.universe, [0, 0, 1.5, 4])
tumor_size['intermediate'] = fuzz.trapmf(tumor_size.universe, [0, 1.5, 4, 6.5])

lymph_node['zero'] = fuzz.trapmf(lymph_node.universe, [2.5, 5, 6, 8.5])
lymph_node['intermediate_num'] = fuzz.trapmf(lymph_node.universe, [3.5, 6, 8, 8.5])
lymph_node['high_num'] = fuzz.trapmf(lymph_node.universe, [9.5, 20, 35, 40.5])

risk_status['low'] = fuzz.trimf(risk_status.universe, [0, 0, 3])
risk_status['intermediate'] = fuzz.trimf(risk_status.universe, [0, 3, 6])
risk_status['high'] = fuzz.trimf(risk_status.universe, [2, 5, 8])

her_2.view()

hormone_receptors.view()

risk_age.view()

grade.view()

tumor_size.view()

lymph_node.view()
```

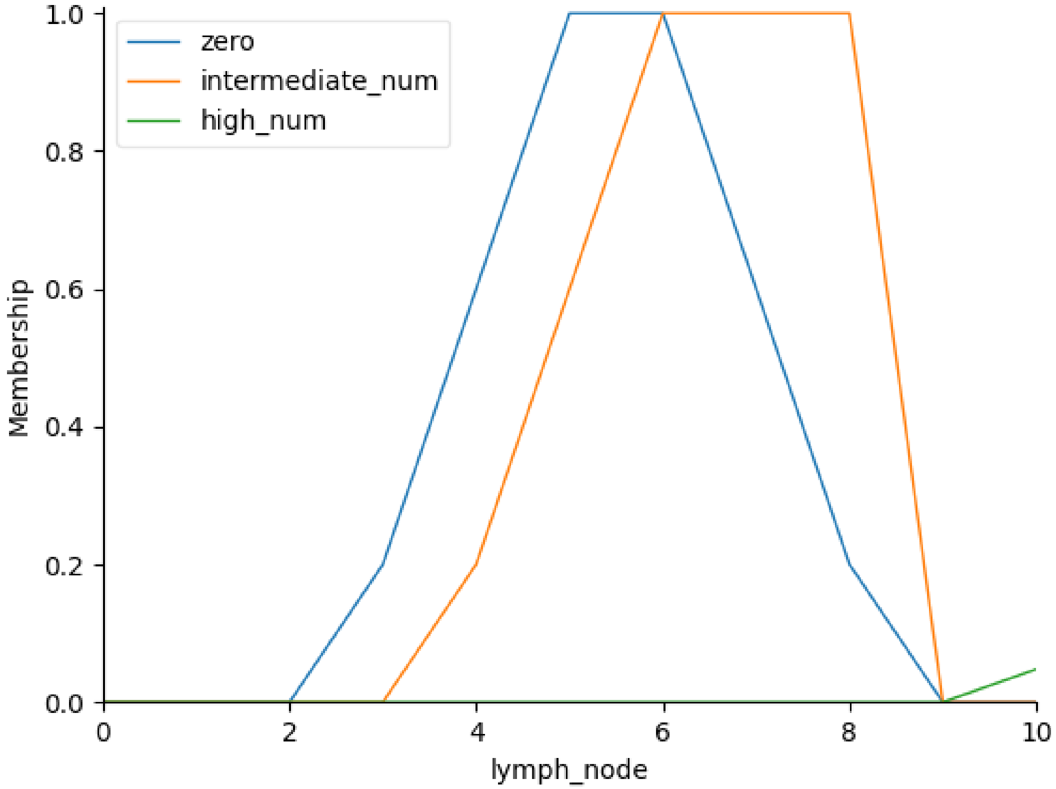
```
risk_status.view()

bc_rule1 = ctrl.Rule(lymph_node['high_num'],risk_status['high'])
bc_rule2 = ctrl.Rule(lymph_node['intermediate_num'],her_2['positive'],risk_status['high'])
bc_rule3 =
ctrl.Rule(lymph_node['intermediate_num'],her_2['negative'],risk_status['intermediate'])
bc_rule4 = ctrl.Rule(her_2['positive'],risk_status['intermediate'])
bc_rule5 = ctrl.Rule(risk_age['very_high'],risk_status['intermediate'])
bc_rule6 = ctrl.Rule(tumor_size['intermediate'],risk_status['intermediate'])
bc_rule7 = ctrl.Rule(grade['grade3'],risk_status['intermediate'])
bc_rule8 = ctrl.Rule(grade['grade3'],risk_status['intermediate'])
bc_rule9 = ctrl.Rule(grade['grade1'],her_2['negative'],lymph_node['zero'],risk_status['low'])

bc_Rules = [bc_rule1, bc_rule2, bc_rule1, bc_rule4, bc_rule5, bc_rule6, bc_rule7, bc_rule8,
bc_rule9]
Cancer_ctrl = ctrl.ControlSystem(bc_Rules)
cancer_eval = ctrl.ControlSystemSimulation(Cancer_ctrl)

plt.show()
```

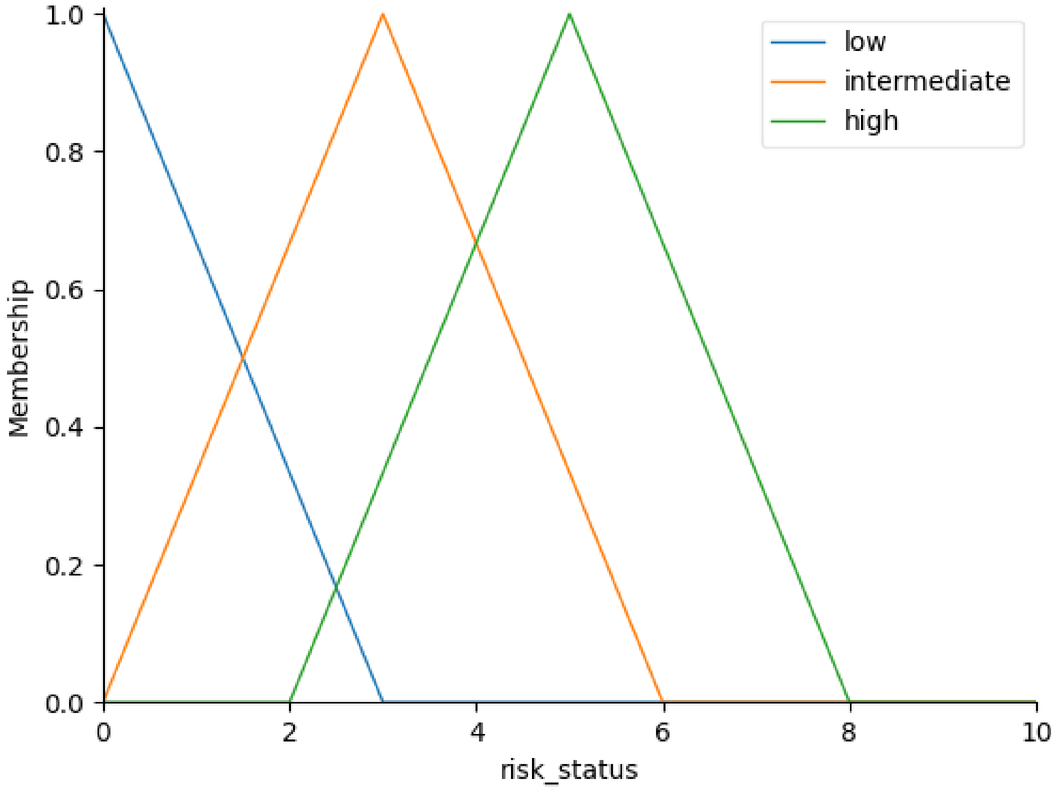



weak_positive
moderate_positive
Membership
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
hormone_receptors


high
very_high
low
Membership
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
risk_age


small
intermediate
Membership
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
tumor_size


zero
intermediate_num
high_num
Membership
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
lymph_node


1.0
0.8
0.6
0.4
0.2
0.0
Membership
0
2
4
6
8
10
risk_status
low
intermediate
high