

Trabalho EEL874 Inteligência Artificial

Aluno: Lucas Manoel de Assis Ventura Trogo DRE 122168184

O trabalho consiste em um sistema de recomendação de filmes usando lógica fuzzy, para criar um sistema de recomendação baseado em três inputs: avaliação do filme, popularidade do filme e preferência do gênero. O output será uma classificação de recomendação.

```
!pip install scikit-fuzzy
```

```
Requirement already satisfied: scikit-fuzzy in /usr/local/lib/python3.10/dist-packages (0.4.2)
Requirement already satisfied: numpy>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from scikit-fuzzy) (1.23.5)
Requirement already satisfied: scipy>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from scikit-fuzzy) (1.11.2)
Requirement already satisfied: networkx>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from scikit-fuzzy) (3.1)
```

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

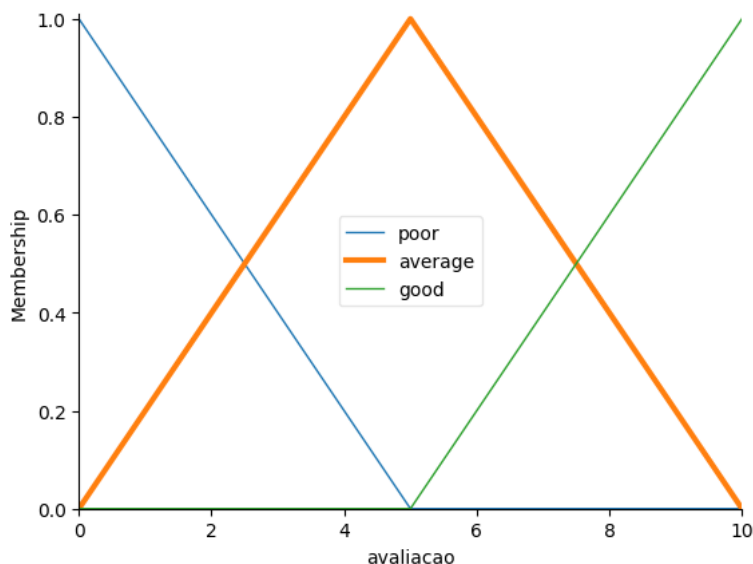
```
# Cria as variáveis de entrada
avaliacao = ctrl.Antecedent(np.arange(0, 11, 1), 'avaliacao')
popularidade = ctrl.Antecedent(np.arange(0, 11, 1), 'popularidade')
preferencia_genero = ctrl.Antecedent(np.arange(0, 11, 1), 'preferencia_genero')
```

```
# Cria a variável de saída
recomendacao = ctrl.Consequent(np.arange(0, 11, 1), 'recomendacao')
```

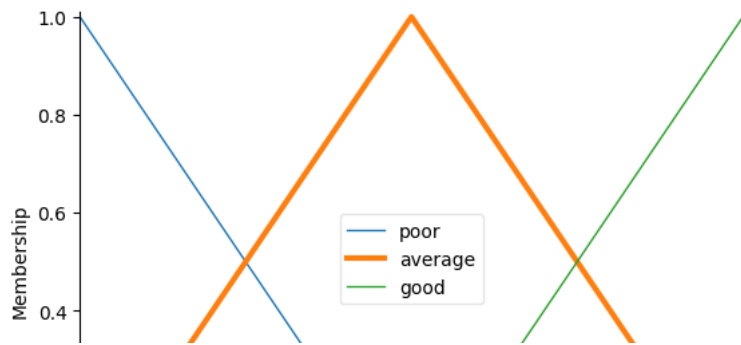
```
# Define funções de pertinência para as variáveis de entrada e saída
avaliacao.automf(3)
popularidade.automf(3)
preferencia_genero.automf(3)
```

```
recomendacao['baixa'] = fuzz.trimf(recomendacao.universe, [0, 0, 5])
recomendacao['média'] = fuzz.trimf(recomendacao.universe, [0, 5, 10])
recomendacao['alta'] = fuzz.trimf(recomendacao.universe, [5, 10, 10])
```

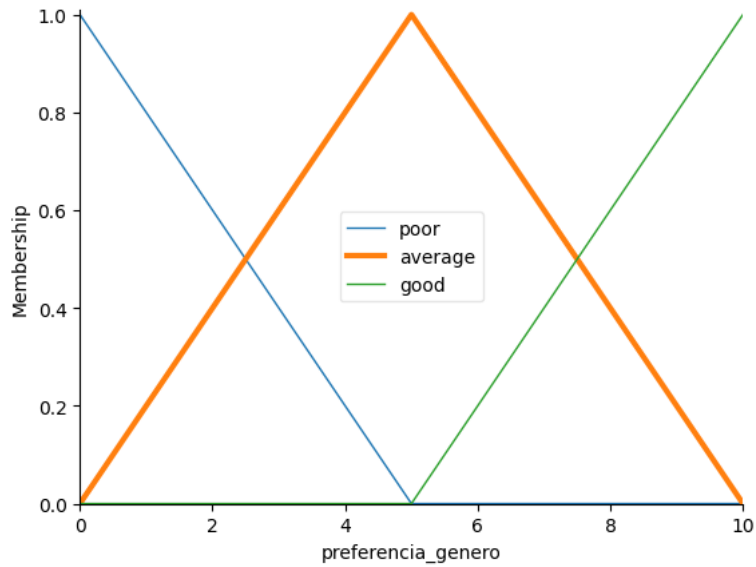
```
avaliacao['average'].view()
```



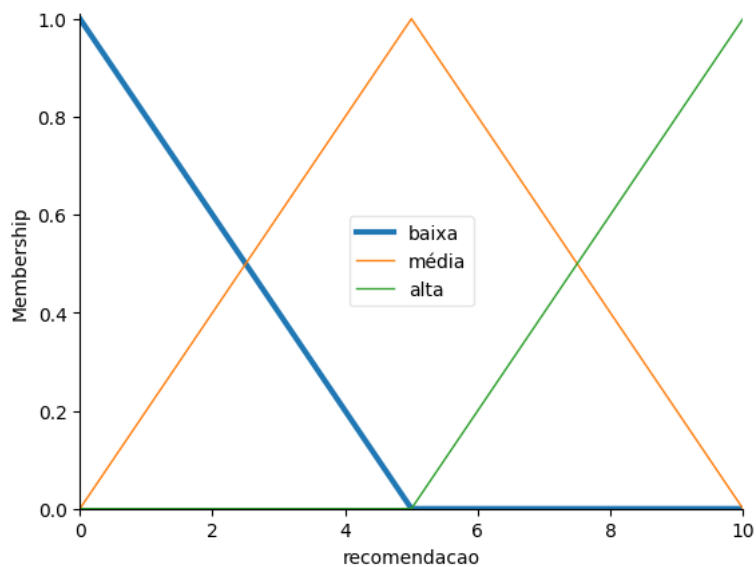
```
popularidade['average'].view()
```



```
preferencia_genero['average'].view()
```



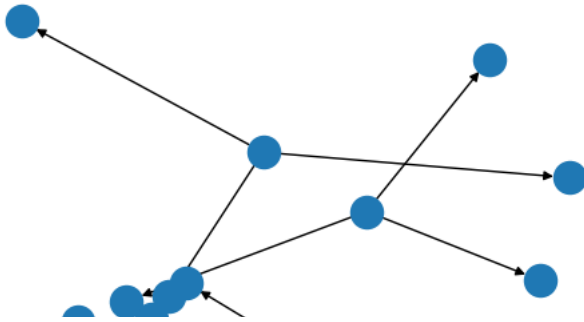
```
recomendacao['baixa'].view()
```



```
# Regras Fuzzy
regra1 = ctrl.Rule(avaliacao['poor'] | popularidade['poor'] | preferencia_genero['poor'], recomendacao['baixa'])
regra2 = ctrl.Rule(avaliacao['average'] & popularidade['average'] & preferencia_genero['average'], recomendacao['média'])
regra3 = ctrl.Rule(avaliacao['good'] | popularidade['good'] | preferencia_genero['good'], recomendacao['alta'])

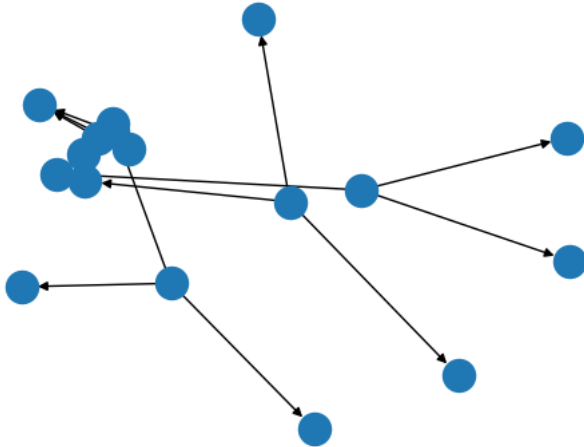
regra1.view()
```

(<Figure size 640x480 with 1 Axes>, <Axes: >)



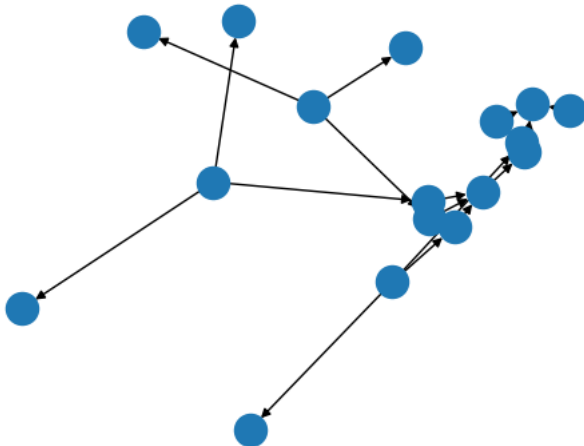
regra2.view()

(<Figure size 640x480 with 1 Axes>, <Axes: >)



regra3.view()

(<Figure size 640x480 with 1 Axes>, <Axes: >)



```
# Cria o sistema de controle
sistema_controle = ctrl.ControlSystem([regra1, regra2, regra3])
```

```
# Cria a simulação
simulacao = ctrl.ControlSystemSimulation(sistema_controle)
```

```
# Defina os valores de entrada
simulacao.input['avaliacao'] = 7.5
simulacao.input['popularidade'] = 6.0
simulacao.input['preferencia_genero'] = 8.0
```

```
# Calcule a recomendação
simulacao.compute()
```

```
# Obtenha o valor de recomendação
recomendacao_final = simulacao.output['recomendacao']

# Visualize a recomendação
print(recomendacao_final)
recomendacao.view(sim=simulacao)
```

