

UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
COORDENADORIA DE ENGENHARIA ELÉTRICA



Universidade Federal  
de São João del-Rei

Inteligência Artificial Aplicada - Trabalho Prático 2

Ana Carolina Rodrigues Almeida 140950050

João Victor Galdino Bouzon 160950007

Luis Fernando Macedo Innocencio 160950019

Sistemas de Inferência Fuzzy

São João del-Rei, Outubro de 2020.

## 1 OBJETIVOS

Iniciar o desenvolvimento das competências (conteúdos, habilidades e atitudes) necessárias para se tornar um profissional de IA no uso de algoritmos voltados para problemas reais a partir do estudo e da compreensão dos Sistemas de Inferência Fuzzy (FIS).

Os objetivos específicos são:

1. Modelar problemas do mundo real que envolvem incerteza ou variáveis linguísticas;
2. Entender como selecionar os conjuntos os quais essas variáveis podem ser "qualificadas" determinando o tipo de função que representa cada conjunto;
3. Entender os conceitos de fuzzificação, defuzzificação, funções de pertinências e regras de inferência.
4. Implementar um sistema de inferência fuzzy usando a linguagem Python.

## 2 RECURSOS E PRÉ-REQUISITOS

- Anaconda (Faça o download em <https://www.scilab.org/> e instale antes do momento síncrono);
- Spyder;
- Bibliotecas: skfuzzy e numpy.
- Iris dataset

## 3 ATIVIDADES

A atividade a ser entregue consiste na implementação de dois estudos de caso sendo um o sistema de controle Fuzzy para temperatura do chuveiro e o segundo um sistema de classificação para o Iris dataset.

### 3.1 Para o Estudo de caso 1:

1. Utilizando as rotinas de exemplo fornecidas, implemente um sistema de Inferência (controle) Fuzzy para a temperatura de um chuveiro, considerando duas variáveis de entrada e uma de saída;
2. Detalhe, no relatório, a implementação mostrando funções de pertinência de entrada e saída, tabela mostrando as regras de inferência escolhidas e processo de defuzzificação;
3. Escolha valores de entrada significativos para ilustrar o funcionamento do sistema;
4. Suba o código para o github ou gitlab e não esqueça de comentá-lo.

### 3.2 Para o Estudo de caso 2:

1. Utilizando as rotinas de exemplo fornecidas, implemente um sistema de Inferência (controle) Fuzzy que seja capaz de classificar os 3 tipos de flores Iris contidas no dataset a partir das suas 4 características;
2. Detalhe, no relatório, a implementação mostrando funções de pertinência de entrada e saída, tabela mostrando as regras de inferência escolhidas e processo de defuzzificação;

3. Divida o dataset iris em duas partes, um para treinamento e outro para verificar a porcentagem de acertos.
4. Use a estatística dos dados do dataset, ou seja, como os dados estão distribuídos por característica para cada tipo de flor, para determinar a forma e os parâmetros das funções de pertinência e as regras de inferência.
5. Utilize as labels contidas no dataset para verificar o índice de acertos.
6. Ajuste as funções de pertinência de modo a aumentar o índice de acertos acima de 90%. Relate o processo de ajuste adotado.
7. Suba o código para o github ou gitlab e não esqueça de comentá-lo.

#### 4 REFERÊNCIAS

Seguem alguns links para apoio a atividade

- <<https://pythonhosted.org/scikit-fuzzy/>>
- <[https://scikit-fuzzy.readthedocs.io/en/latest/userguide/getting\\_started.html](https://scikit-fuzzy.readthedocs.io/en/latest/userguide/getting_started.html)>
- <[https://scikit-learn.org/stable/auto\\_examples/datasets/plot\\_iris\\_dataset.html](https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html)>
- <<https://medium.com/@avulurivenkatasaireddy/exploratory-data-analysis-of-iris-data-set-using-python>>

## 5 METODOLOGIA

### Estudo de caso 1:

Para projetar o controlador de chuveiro com a logica fuzzy iniciamos escrevendo nossas variáveis de entrada (abertura da torneira e temperatura da água acima da ambiente) e saída (potência do chuveiro), bem como seus valores numéricos e nomenclaturas para a classificação fuzzy, esses dados estão disponíveis no arquivo *'info.txt'*.

Após definidas as variáveis foram elaboradas as regras de decisão, que relacionam as entradas com uma saída, para o nosso controlador específico as regras de decisão têm como objetivo manter a temperatura da água mediana, assim quando a temperatura estiver alta a potencia do chuveiro é diminuída, quando estiver baixa a potencia é aumentada, caso a temperatura esteja mediana a potência do chuveiro também seria mediana.

Na tabela de regras de decisão as saídas foram agrupadas para otimizar o numero de regras necessárias para cobrir todas as possibilidades, a tabela está disponível no arquivo *'RegrasDeDecisão.xlsx'*

Definidas as variáveis e as regras de decisão iniciamos a redação da rotina em Python usando os comandos usados no código exemplo dado pelo professor, tais como, *fuzz.trimf*, *automf*, *ControlSystemSimulation* e demais da biblioteca *Skfuzzy*.

Depois de simular e entrar com diferentes valores de abertura da torneira e temperatura da água pudemos determinar que a potência de saída dada pela rotina condiz com o esperado. Exemplos de resultados expostos no item 6 desde relatório.

### Estudo de caso 2:

Para o processo do estudo de caso 2 foi necessário de antemão conhecer os dados do banco de dados da íris, este possui quatro entradas, comprimento da flor, largura da flor, comprimento da pétala e largura da pétala. Analisando o banco de dados vimos que para diferenciar as espécies da planta não seria necessário usar as quatro variáveis visto que com apenas duas já seria possível fazer a distinção por haver uma disparidade nas dimensões entre cada espécie, estas duas seriam o comprimento e a largura da pétala.

Então estas variáveis foram divididas em nomenclaturas fuzzy, *pp*, *p*, *m*, *g*, *mg* e *gg* para a largura da pétala e *pp*, *p*, *m*, *g* e *mg* para o comprimento da pétala. Para a saída como só haviam 3 possíveis espécies as variáveis de saída são 0, 1 e 2 para condizer com a label do dataset.

A rotina de fuzzyficação e desfuzzyficação foi elaborada da mesma forma que no estudo de caso 1 acrescentando a leitura dos dados do data base da íris e a métrica *classification\_report* da biblioteca *sklearn.metrics*

## 6 RESULTADOS

### Estudo de caso 1:

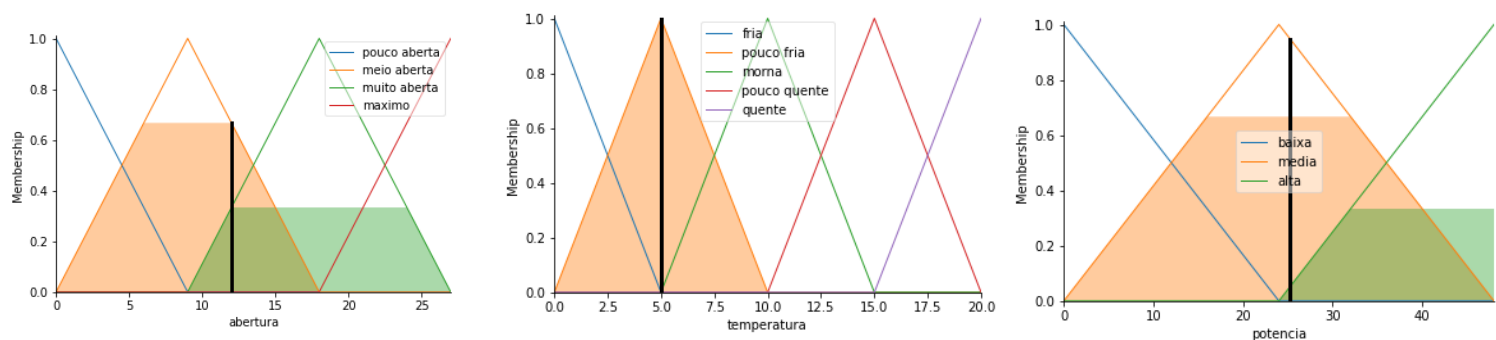


Figura 6.1 Abertura=12 e temperatura=5

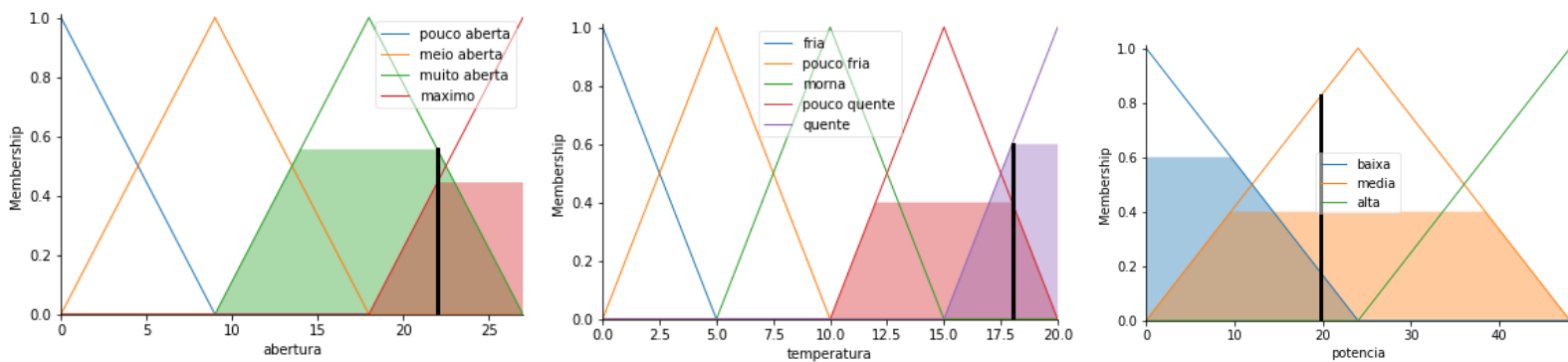


Figura 6.2 Abertura=22 e temperatura=18

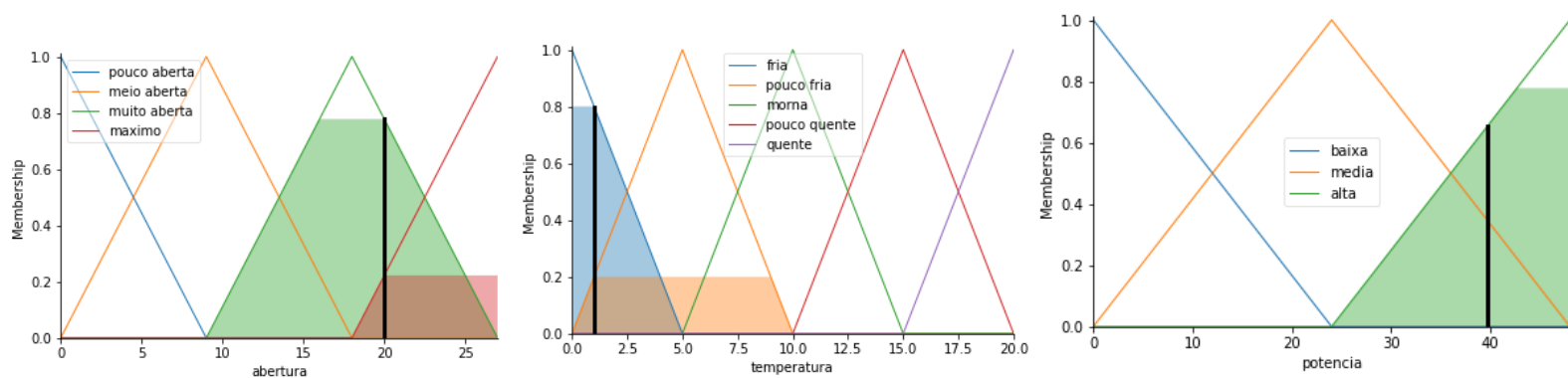


Figura 6.3 Abertura=20 e temperatura=1

## Estudo de caso 2:

36	5.4	3.4	1.3	0.2	0
37	4.9	3.6			
38	4.4	3			
39	5.1	3.4			
40	5	3.5	1.3	0.3	0
41	4.5	2.3	1.3	0.3	0
42	4.4	3.2	1.3	0.2	0

**Entradas**  
↓ ↓ ↓

**Saida esperada**

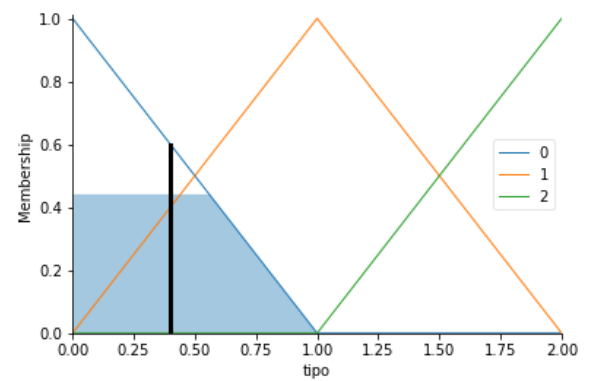
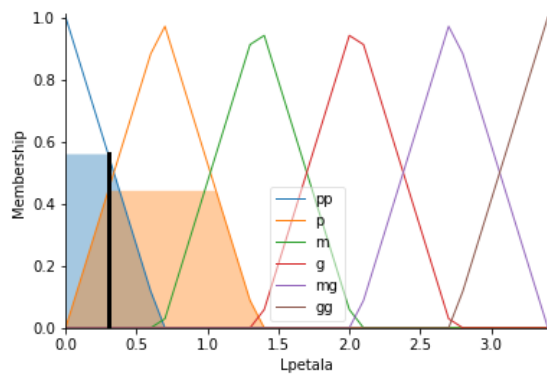
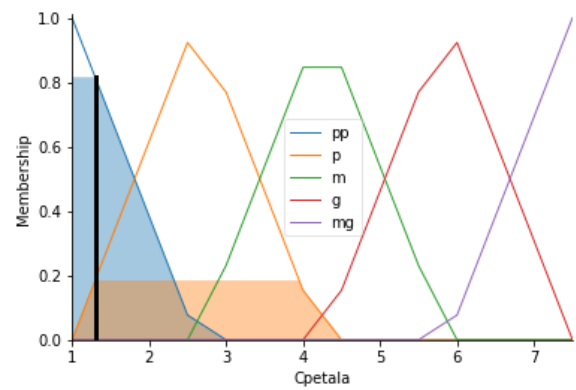


Figura 6.4 -Resposta da rotina à amostra número 40 do dataset

64	5.6	2.9	3.6	1.3	1
65	6.7	3.1			
66	5.6	3			
67	5.8	2.7			
68	6.2	2.2	4.5	1.5	1
69	5.6	2.5	3.9	1.1	1

**Entradas**  
↓ ↓ ↓

**Saida esperada**

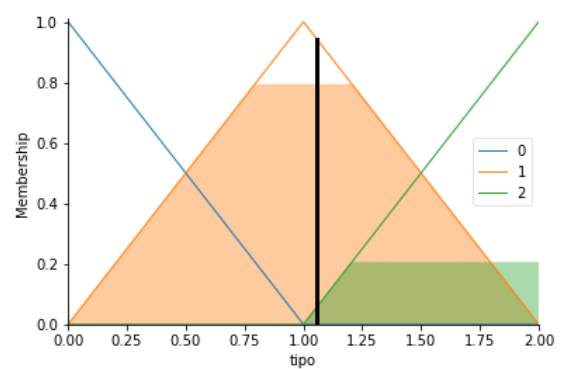
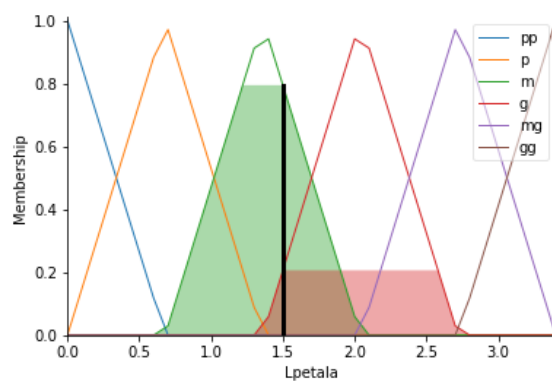
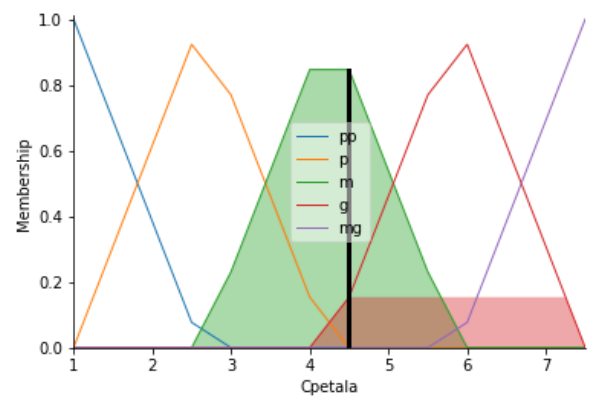


Figura 6.5 -Resposta da rotina à amostra número 68 do dataset

	precision	recall	f1-score	support
0	1.00	1.00	1.00	50
1	0.96	0.91	0.93	53
2	0.90	0.96	0.93	47
accuracy			0.95	150
macro avg	0.95	0.95	0.95	150
weighted avg	0.95	0.95	0.95	150

Figura 6.6 - Resultados entregues pela métrica *classification\_report*

## 7 DISCUSSÃO DOS RESULTADOS

Estudo de caso 1:

É possível verificar pelos plots gerados pela rotina que a potência de saída dada pelo controlador fuzzy condiz com o esperado definido nas regras de decisão.

Estudo de caso 2:

Neste caso como temos a base de dados com as variáveis de entrada e a saída esperada vemos nas figuras 6.4 e 6.5 exemplificações de entradas para a rotina e adquirindo a saída esperada. Já na figura 6.6 vemos o resultado da métrica que indica um índice de acertos de 95% para o nosso algoritmo de classificação.

## 8 CONCLUSÃO

A elaboração do algoritmo apresentou como desafios: o conhecimento de bibliotecas específicas para a lógica Fuzzy, dimensionamento das variáveis, atribuições de nomenclaturas fuzzy para caracterização de cada variável, regras da defuzzificação, entrada de dados para teste, simulação e visualização gráfica. Além de desenvolvimento de condicionamentos, recursividades com regras de decisão para validação do método utilizado dados da data base.

Para este mesmo dataset foi realizado semanas atrás o trabalho prático 1 que fez uso de diferentes rotinas de classificação/clusterização para os dados do estudo de caso 2. Nestas rotinas (K-means, CNN e DBSCAN) a taxa de acertos ficou entre 60 e 75%, já com a rotina discutida neste trabalho a taxa de acertos bate os 95%, provando ser mais precisa para este tipo de dataset.

SABER	INDAGAR	APRENDER	REFLETIR
O que eu sabia antes de fazer a atividade?	O que eu ainda preciso aprender?	O que eu aprendi?	Como eu aprendi? Quais estratégias utilizei e como posso usá-las em outras situações de aprendizagem?
Logica fuzzy teórica	Funções de Python	Algumas funções python (array, list, etc.)	Artigos Online
Métricas		Rotina base da logica fuzzy	
Uso de Bibliotecas Python		Plotagem de resultados	

Tabela 8.1 – Tabelar SIAR: Simulação de algoritmos de classificação e agrupamento.

## 9 NOVAS REFERÊNCIAS

<https://github.com/luisfernandoinn/Trabalho2IA>