FUZZY: DESENVOLVIMENTO DE UM SHELL PARA A CONSTRUÇÃO DE MODELOS DE SISTEMAS DIFUSOS

Beatriz Cristina Luiz [I]

Paulo Roberto Espindola de Oliveira Junior [II]

Orientador: Prof. Dr. Clávison Martinelli Zapelini[III]

Resumo: Este artigo apresenta o desenvolvimento de um Shell para a geração automática de sistemas Fuzzy, que possibilita a definição das variáveis linguísticas, os graus de pertinência dessas variáveis e as regras que serão utilizadas no processo de Fuzzificação e Defuzzificação. Foram realizados testes com uma turma de alunos do curso de Ciências da Computação da Unisul, que tinham como objetivo comparar a eficácia do Shell desenvolvido e um framework, já consolidado. Os resultados alcançados nesse processo comprovaram a eficácia do programa desenvolvido no tratamento de dados imprecisos e suporte ao apoio de decisão em sistemas fuzzy.

Palavras-chave: Sistema especialista, lógica fuzzy, Shell, Sistema de apoio à decisão.

Abstract: This article presents the development of a Shell for the automatic generation of Fuzzy systems, which allows the definition of linguistic variables, degrees of relevance of these variables and the rules that will be used in the process of Fuzzification and De-fuzzification. Tests were carried out with a group of students from the Computer Science course at Unisul, which aimed to compare the effectiveness of the developed Shell and an already consolidated framework. The results achieved in this process proved the effectiveness of the program developed in the treatment of inaccurate data and help for decision support in fuzzy systems.

Keywords: Expert Systems, Fuzzy Logic, Shell, Support Decision Systems.

1 INTRODUÇÃO

Na lógica tradicional, existem apenas dois possíveis valores para cada preposição, sendo eles verdadeiro (1) ou falso (0). Desta maneira uma contraposição que existe inferência imediata e, dessa forma, dada uma preposição outra é inferida. Contudo isto é diferente na lógica fuzzy. Nesta, uma preposição possui graus de inferência e não uma inferência absoluta. Ela admite valores verdade diferentes de verdadeiro (1) ou falso (0). A maior diferença entre a lógica tradicional (contraposição) e a lógica fuzzy é o seu objetivo. Simplificando, o conceito de lógica

[I] Acadêmica do curso de Ciência da Computação da Universidade do Sul de Santa Catarina – Unisul. E-mail: beatriz.luiz@unisul.br

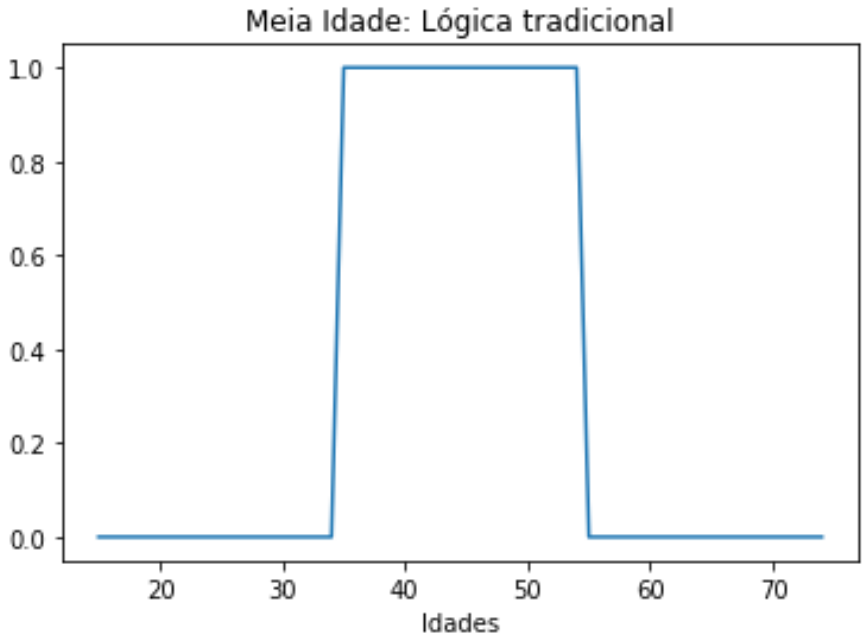
[II] Acadêmico do curso de Ciência da Computação da Universidade do Sul de Santa Catarina – Unisul. E-mail: paulo.oliveira37@unisul.br

[III] Professor do curso de Ciência da Computação da Universidade do Sul de Santa Catarina – Unisul. E-mail: clavison.zapelini@unisul.br

fuzzy pode ser aplicado em uma situação em que não é possível responder simplesmente "sim" ou "não", e nesse caso, “talvez” ou “quase” sejam mais apropriados. [RIGNEL, CHENCI e LUCAS, 2011]

Desta maneira se considerarmos a lógica tradicional para tratar o período da meia idade, uma pessoa com 34 anos só pertenceria a esse grupo após completar 35 anos. Da mesma forma, uma pessoa ao completar 56 anos deixaria de fazer parte desse grupo, conforme gráfico (gráfico 1). (MUKAIDONO, 2001).

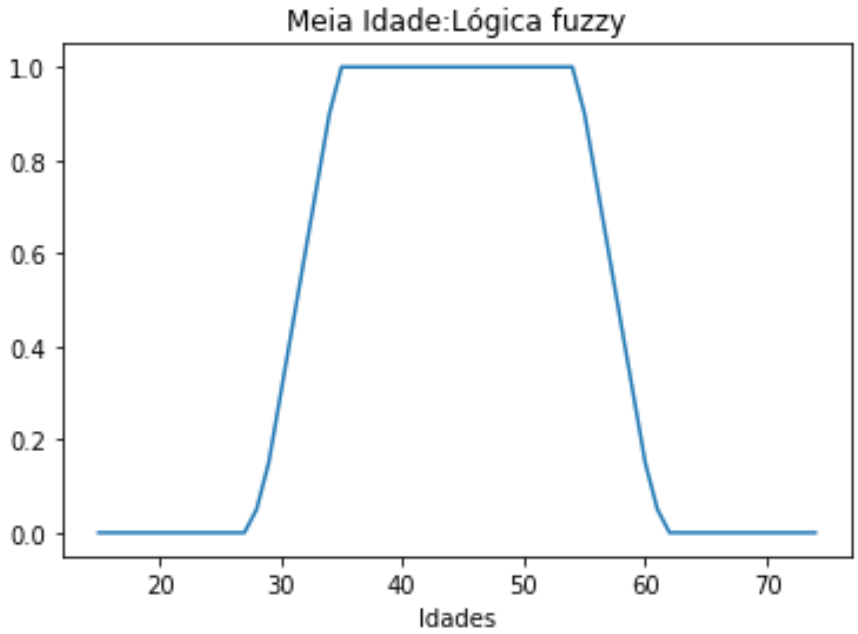
Gráfico 1 - Meia idade pela lógica tradicional.



Fonte: Elaboração do autor, 2020.

Todavia, se aplicarmos o mesmo problema utilizando a lógica fuzzy, uma pessoa de 34 anos teria um grau de pertencer ao grupo da meia idade. Nota-se que o grau de pertinência para que uma pessoa de 28 anos pertença ao grupo da meia idade é muito menor do que uma pessoa de 33 anos (gráfico 2). (MUKAIDONO, 2001)

Gráfico 2 - Grafico da meia idade pela lógica fuzzy.



Fonte: Elaboração do autor, 2020.

A lógica fuzzy admite valores entre 0 e 1. Sendo que uma pertinência de 0,5 representa meia verdade, quando 0,9 representa quase verdade e 0,1 quase falso. Diferente da lógica tradicional que só admite valores booleanos, ou seja, verdadeiro ou falso. (SILVA, 2005). E, a necessidade de tratar problemas de raciocínios aproximados ou multivalorados em vez do tradicional verdadeiro (1) ou falso (0), torna a técnica da lógica fuzzy ideal no tratamento de dados imprecisos. Ela permite interpretar fenômenos não quantitativos com graus de verdade em pertencer ou não a determinados grupos. A lógica difusa possui também a vantagem de ser facilmente expressada em termos humanos e auxilia na operação de tarefas com naturalidade. Zadeh, grande colaborador da lógica, em 1973 apresentou o princípio da incompatibilidade como parte de sua concepção:

> "À medida que a complexidade de um sistema aumenta, nossa habilidade para fazer afirmações precisas e que sejam significativas acerca deste sistema diminui até que um limiar é atingindo além do qual precisão e relevância tornam-se quase características mutuamente exclusivas" (WEBER e KLEIN, 2003, p. 23).

A lógica difusa pode ser uma ferramenta muito útil para resolver problemas de imprecisão. Para [POSSELT, FROZZA E MOLZ, 2011], a lógica difusa ainda é pouco disseminada como forma de auxiliar nos processos de algum negócio. Isso ocorre,

principalmente, pela dificuldade de modelagem e implementação. O projeto Fuzzy visa resolver a problemática de desenvolver um sistema inteligente do início ao fim a ser aplicado a um problema específico. A solução descrita nesse artigo, permite modelar uma ferramenta difusa para auxiliar o especialista de qualquer área, sendo expresso em termos humanos, e incentivar na identificação e resolução de problemas com naturalidade.

Segundo [FERRARI, ARGOUD e AZEVEDO, 2004], o Shell separa a máquina de inferência da parte do sistema que trata o problema especificamente. E o usuário deve se preocupar apenas em obter o conhecimento do especialista humano. Nesse artigo, é proposto uma ferramenta Shell que permite criar e modelar um sistema para resolução de um problema por um especialista de qualquer área, com treinamento e conhecimento dos conceitos aqui colocados, para auxiliar e incentivar as operações de tarefas diárias com naturalidade. Este projeto também tem o objetivo de incentivar e permitir a aplicação de conceitos de ciência da computação em outras áreas do conhecimento, podendo ser utilizado para a criação de protótipos inteligentes, de forma didática e de fácil manipulação pelo usuário final.

A ferramenta deve permitir ao especialista agregar informações próprias no seu protótipo e aumentar a eficiência e eficácia na análise e resolução de problemas.

Este artigo apresenta como solução para estes desafios um Shell para construção de sistemas difusos.

2 TRABALHOS RELACIONADOS

Existem vários trabalhos na linha de tratamento de problemas onde os valores de verdade podem variar entre 0 e 1. A maior dificuldade, no entanto, é a necessidade de desenvolver um sistema do completo e abrangente para resolver uma problemática específica. Listamos dois sistemas muito úteis no processo de apoio à decisão utilizando lógica fuzzy.

1.1 CREDSAPIENS

Sistema Especialista Difuso de apoio a análise de concessão de crédito. O CREDSAPIENS é um Sistema Especialista desenvolvido em JAVA Swing, que utiliza uma máquina de inferência difusa já existente. O autor desenvolveu um Shell que consulta essa máquina e é destinado a atender a lacuna no crédito de cobrança, para dar apoio ao especialista na tomada de decisão e efetuar a liberação de crédito adequada. O Sistema recebe o valor que o cliente solicitou e tem como resultado o valor sugerido pelo sistema. Através de gráficos, é possível visualizar o que o cliente solicitou, o que o sistema sugeriu e o que o especialista

liberou. O sistema mostrou resultados satisfatórios, tendo uma diferença de 9,3% entre o que o sistema sugeriu e o que o especialista de crédito liberou. [NETO, PEREIRA. 2011]

1.2 SISTEMA ESPECIALISTA DIFUSO APLICADO AO PROCESSO DE ANÁLISE QUÍMICA QUALITATIVA DE AMOSTRAS DE MINERAIS.

O Sistema propôs uma análise qualitativa, baseada na lógica difusa, de amostras de minerais com características de natureza imprecisa. O sistema mostrou uma performance mais satisfatória comparado a um sistema tradicional, em termos computacionais. Uma vez que se aproxima mais da perícia de um humano. [FERNANDES, 1996]

3 SOLUÇÃO PROPOSTA

A construção de um Shell para criação de modelos de sistemas difusos, que possibilite fácil configuração e que se aplique tanto a nível acadêmico, quanto corporativo, e que permita a utilização da lógica fuzzy como auxílio para propor soluções.

Especificamente, o sistema permite a criação de modelos de conjuntos difusos e - além de definir as variáveis - configurar os seus intervalos, visualizando-os em um gráfico, definir a saída do modelo, criar regras de inferência e, por fim, testar o modelo, possibilitando ajustes.

Explicaremos a utilização do sistema desenvolvido exemplificando como este pode auxiliar a determinar o melhor nível de irrigação de água a ser adotado, levando em consideração diversas variáveis como tipo de plantação, tipo do solo, temperatura, umidade e luz do sol. A aplicação desta solução é extremamente útil para o controle de qualidade e maximizar a colheita em plantações sensíveis a estas variações, como por exemplos as de uvas viníferas, utilizadas na produção de vinhos, que em algumas épocas do ano devem receber apenas uma gota de água por dia.

3.1 Base de dados

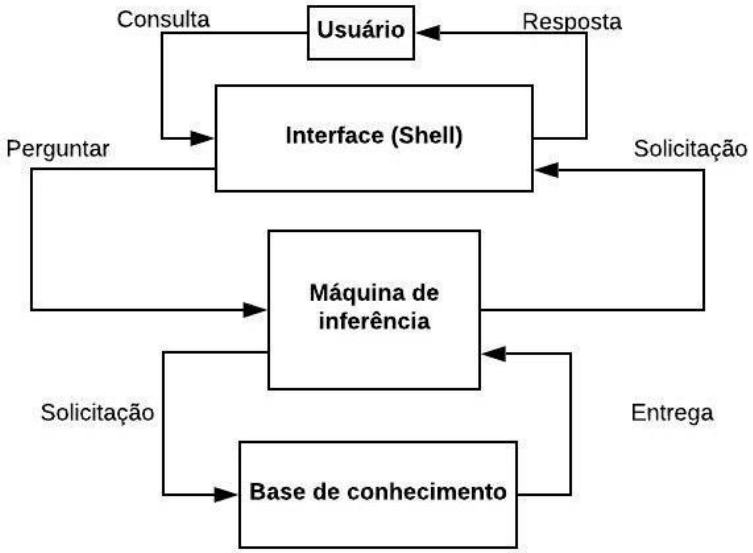
É necessário obter uma base de dados, ou seja, um especialista da área, nesse caso, um agricultor ou pesquisador da área de agricultura. Alguém que tenha conhecimento no assunto do problema. Requer do especialista os valores reais para cada variável.

O primeiro passo é determinar as variáveis de entrada do modelo, nesse caso, tipo de plantação, época do ano, temperatura, umidade e luz do sol. Após isso, determina-se as variáveis de saída. Nesse caso, a irrigação, ou seja, porcentagem de água liberada. A próxima

etapa é determinar as regras, ou seja, condições para irrigação ocorrer. Ex: Se tipo de plantação = Vinífera E Temperatura = Alta OU umidade = baixa Então Irrigação = Alta.

3.2 Testando o modelo

Figura 1 - Fluxograma do sistema Fuzzy



Fonte: Elaboração do autor, 2020.

Após o especialista preencher a base de dados do problema em questão, é possível testar o modelo.

Conforme mostra o fluxograma (figura 1), o usuário informa por meio de uma interface amigável os valores de cada entrada com base nas condições atuais. A interface envia estas informações para a máquina de inferência que consta a base de dados cadastrada pelo especialista.

Para cada campo da variável, será calculada a pertinência do ambiente daquela variável. Ou seja, é aplicada a função de pertinência trapezoidal. Essa função obtém o X (input do usuário) e o abcd (início suporte, início núcleo, fim do núcleo, fim do suporte).

Isto posto, são feitas as verificações:

- Primeiro, verifica se o X está dentro da range(abcd). Caso não esteja, ele retorna 0 como pertinência.
- Caso a primeira verificação retorne verdadeiro, verifica se X >= Início do Núcleo E x <= Fim do Núcleo, se for verdadeiro, retorna 1.

- Posteriormente, verifica se o início do suporte é menor que o início do núcleo, se o início do núcleo é menor do que o fim do núcleo, e se o fim do núcleo é menor que o fim do suporte. Ou seja, a <= b <= c <= d
- Se A for diferente de B, ele verifica se X está entre A e B e se o B é maior que A, então ele usa a função:

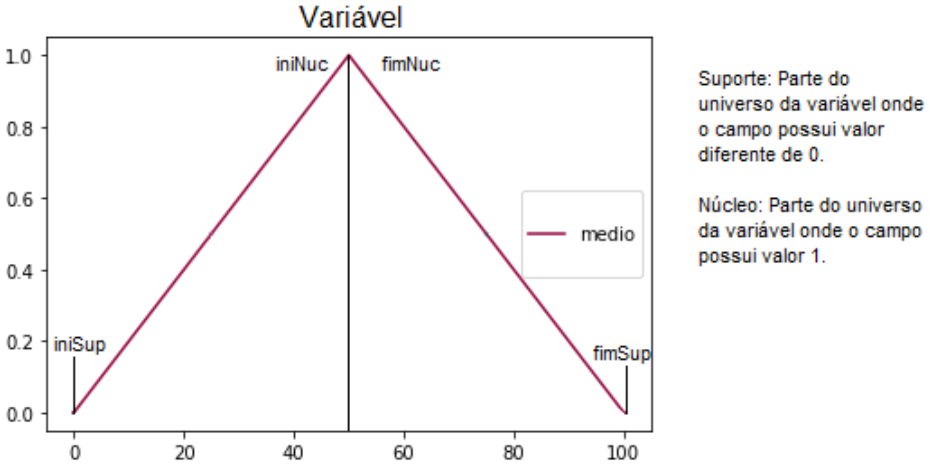
$$\frac{X - IniSup}{IniNucleo - IniSup}$$

Se não ele usa a função:

$$\frac{fimSuporte - X}{fimSuporte - fimNucleo}$$

Essas duas funções calculam a pertinência de cada campo para cada variável (gráfico 3).

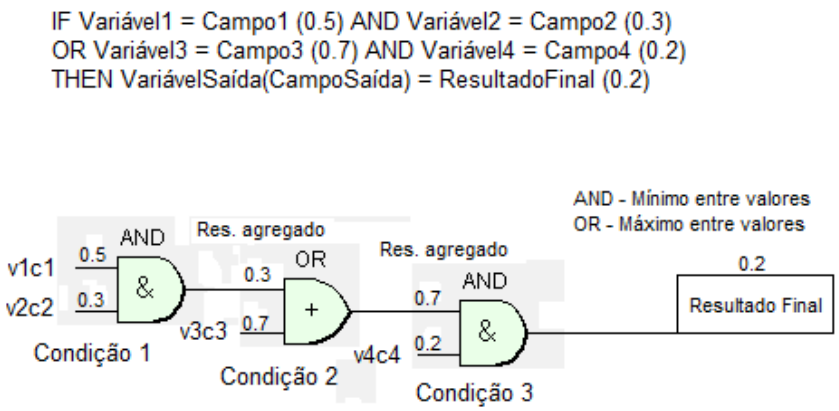
Grafico 3 - Suporte e núcleo de uma variável



Fonte: Elaboração do autor, 2020

Após calcular toda pertinência dos campos das variáveis, é feita a leitura das regras e o cálculo da pertinência da regra a partir das pertinências dos campos presentes nas regras e dos operadores lógicos AND e OR (Figura 2).

Figura 2 - Cálculo da pertinência do campo da variável de saída a partir da regra.



Fonte: Elaboração do autor, 2020

Ou seja,

Se tipo de plantação = Vinífera E Temperatura = Alta OU umidade = baixa Então Irrigação = Alta.

Utilizando os operadores And (E) e Or (OU) somos capazes de efetuar a agregação da pertinência dos campos, ou seja, o And utiliza o menor valor entre a pertinência agregada e a pertinência do próximo campo. A pertinência agregada é o valor de pertinência da regra até o ponto atual. A nova pertinência agregada é a união da pertinência dos campos das variáveis baseadas no operador da regra que as une, até que não existam mais novos campos na regra.

O método utilizado neste projeto é o método de Mamdani, onde E utiliza o menor valor (Min.) entre duas pertinências e OU utiliza o maior valor (Max.).

Exemplificando, suponhamos a seguinte regra:

Se tipo de plantação = Vinífera E Temperatura = Alta OU umidade = baixa Então Irrigação = Alto.

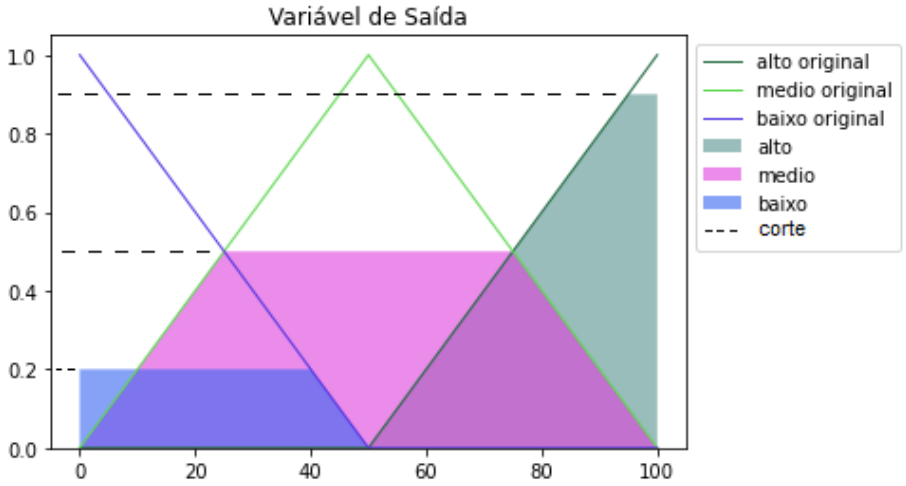
- Pertinência do tipo de plantação (Vinífera) = 1
- Pertinência da temperatura (Alta) = 0,9
- Pertinência da umidade (Baixa) = 0,6

A pertinência de agregação do tipo de plantação e temperatura é 0,9, pois o operador que as une é E.

A nova pertinência agregada baseada na última pertinência e o campo da variável Umidade é 0.9, pois o operador que as une é OU.

Isto posto, é extraído o valor de pertinência para as saídas das regras, esses valores são classificados e o algoritmo utiliza a maior pertinência para cada saída. A partir desses valores, é feito o corte vertical da parte superior de cada campo de saída (gráfico 4).

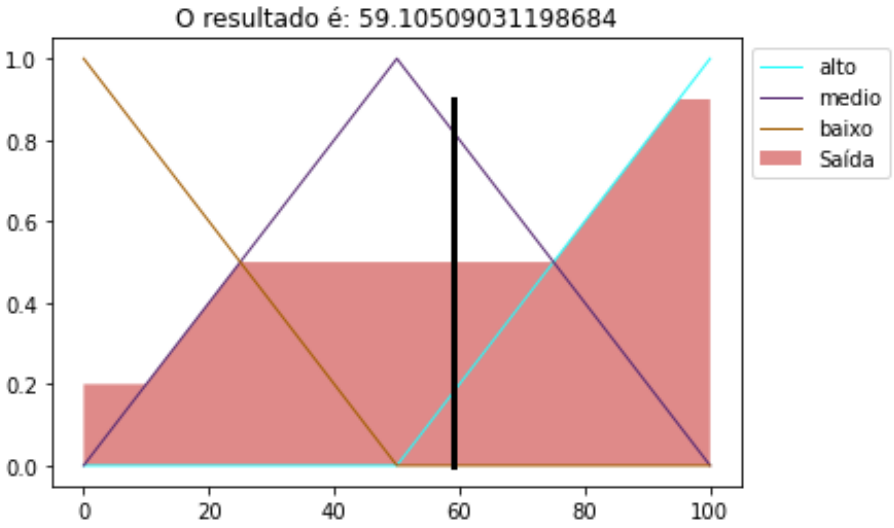
Gráfico 4 - Saída das variáveis



Fonte: Elaboração do autor, 2020

Posteriormente, é feita a agregação das funções de pertinência das saídas, gerando uma função só. Nesta etapa, é feita a Defuzzificação dessa nova função, a partir do método Centroid do eixo X (horizontal). E este é o valor da saída (gráfico 5). A máquina de inferência devolve este valor resultado para a interface que mostra ao usuário.

Gráfico 5 - Resultado.



Fonte: Elaboração do autor, 2020.

3.3 Interface

A aplicação possui uma interface web escrita em Django, que permite o cadastro dos modelos, visualização do gráfico com os intervalos de cada variável. Os modelos poderão ser

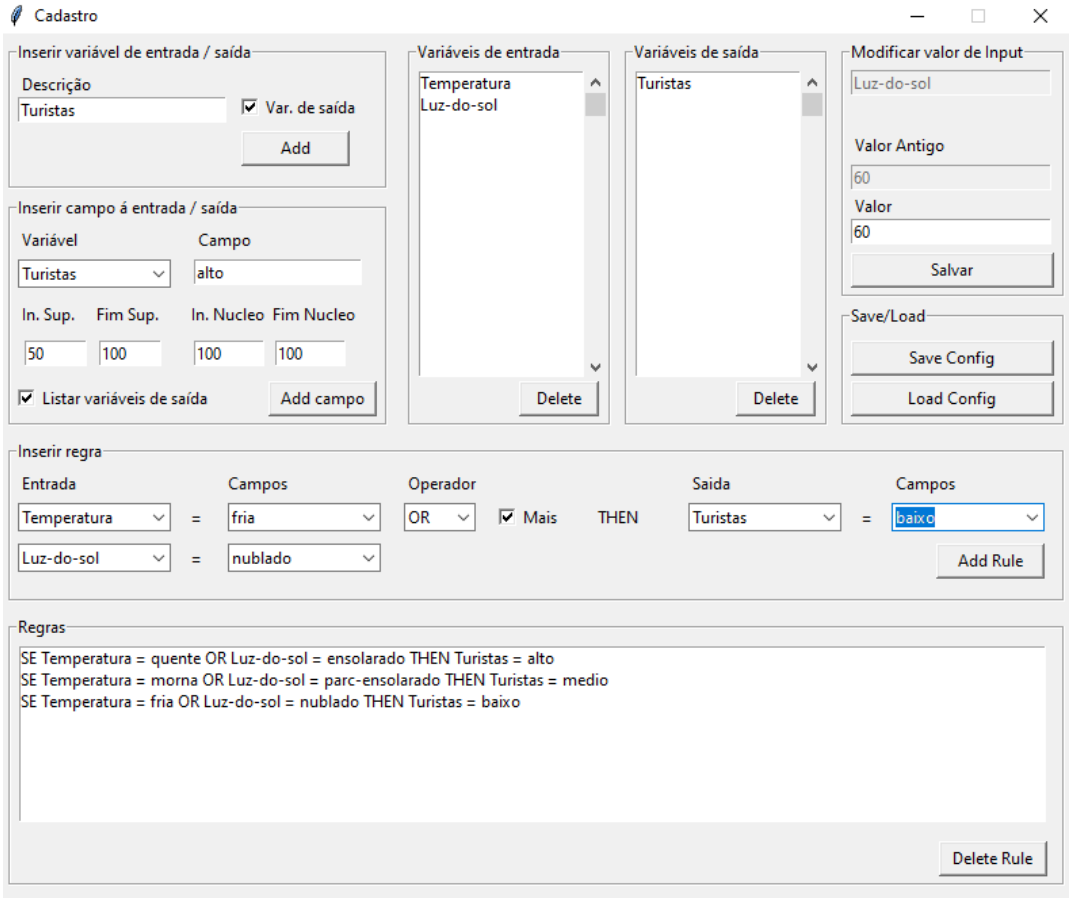
utilizados na web para testes para facilitar os ajustes, como também para uso final.

Qualquer modelo poderá ser alterado, permitindo assim adicionar novas variáveis, regras da base de conhecimento e, caso desejado, excluir o modelo.

4 Testes

Inicialmente, desenvolveu-se um protótipo na linguagem Python, com a interface gráfica Tkinter. Os dados foram cadastrados em uma única tela (Figura 3).

Figura 3 - Tela do protótipo



Fonte: Elaboração do autor, 2020.

Nesta figura, foi utilizado o seguinte exemplo: Seja um sistema difuso para predizer o número de turistas visitando um resort.

Variáveis de entrada:

- Temperatura (em graus Celsius)
- Luz do sol (expressa em uma porcentagem do máximo esperado de luz do sol)

Variáveis de saída:

- Quantidade estimada de turistas (expressa em porcentagem da capacidade do resort).

Base de conhecimento:

- Variáveis linguísticas de entrada:
    - Temperatura {fria, morna, quente}
    - Luz do sol {nublado, parcialmente ensolarado, ensolarado}
- Variáveis linguísticas de saída:
    - Turistas {baixo, médio, alto}

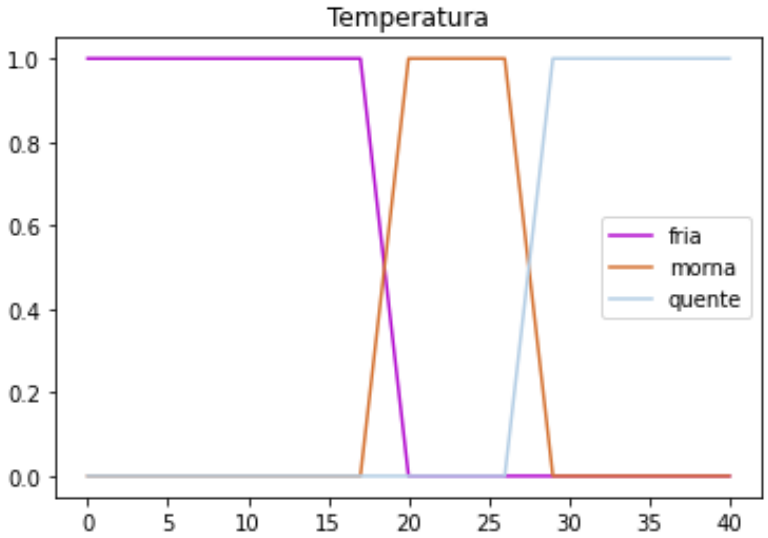
Regras (devem ser definidas por um especialista):

- Se temperatura é quente ou luz do sol é ensolarado então turistas é alto.
- Se temperatura é morna e luz do sol é parcialmente ensolarado então turistas é médio.
- Se temperatura é fria ou luz do sol é nublado então turistas é baixo.

Operadores de união e intersecção: máx. e min.

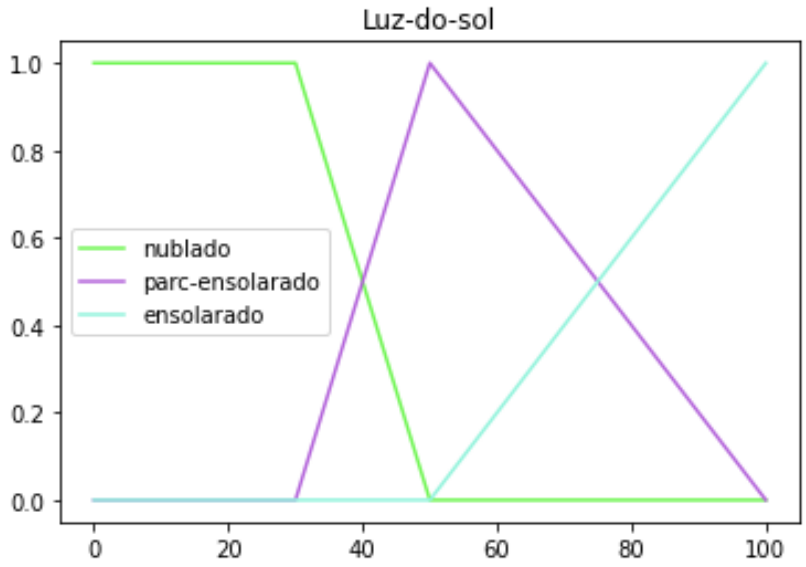
Suponhamos abaixo uma situação em que a temperatura está em 19 graus Celsius e a luz do sol em 60%.

Gráfico 6 – Temperatura



Elaboração do autor, 2020

Gráfico 7 – Luz do Sol

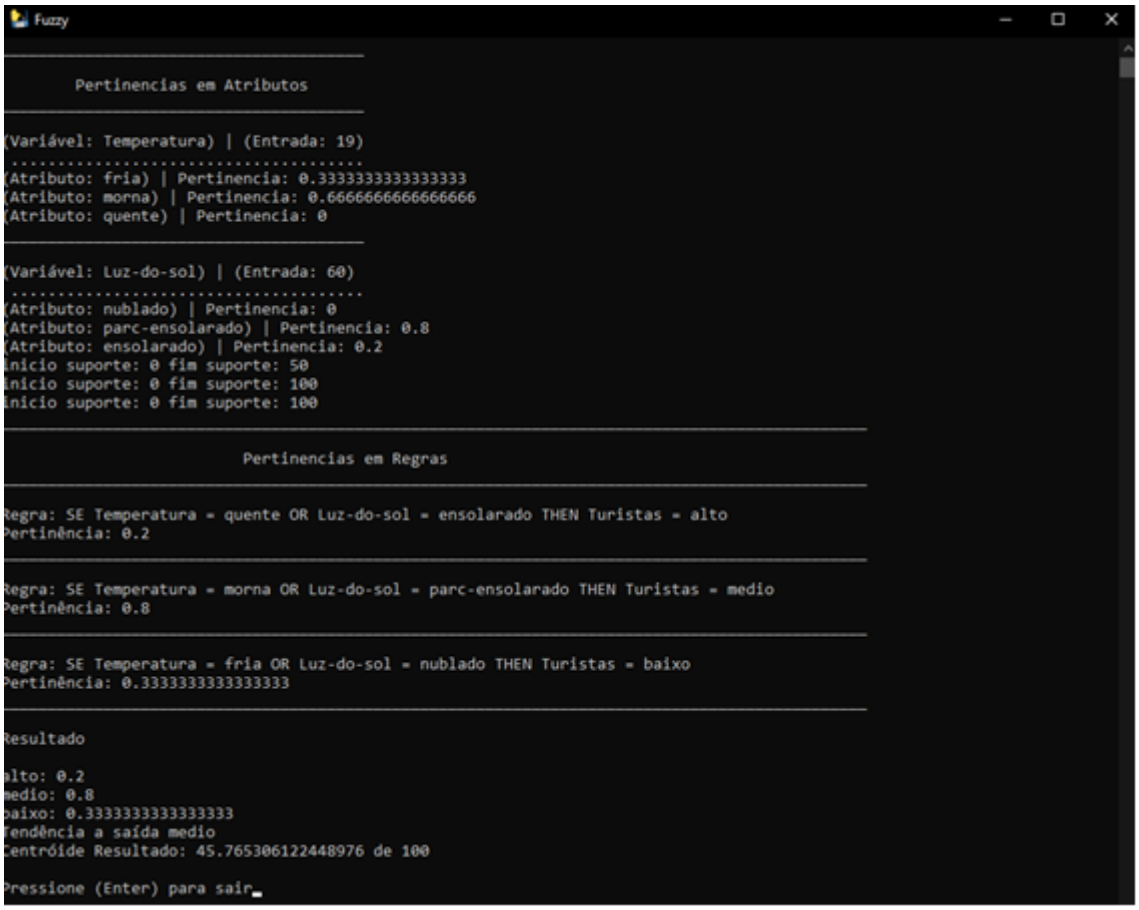


Elaboração do autor, 2020.

O usuário deve digitar as variáveis de entrada no campo “Descrição” e as cadastra clicando em “add”.

1. Para adicionar as variáveis de saída, é necessário marcar a checkbox “Var. de saída”. Em seguida digitar os valores no campo “Descrição” e clicar em “add”.
2. O usuário deve selecionar a variável cadastrada e preencher os campos de “suporte” e “núcleo” e clicar em “add campo”.
3. Realizado os primeiros passos deve-se cadastrar as regras. Selecionar a “entrada”, o “campo da entrada”, a “variável de saída” e o “campo da saída”. Caso deseja-se incluir mais uma variável de entrada, pode-se marcar a checkbox “Mais” e selecionar uma nova entrada e seu campo. O protótipo é limitado a dois campos de entrada. Preenchidos todos os campos, deve-se clicar em “Add Rule”.
4. Partindo de que todas as métricas estão devidamente cadastradas, deve-se clicar em “Save config”. Ao fazê-lo a ferramenta vai exportar um Json.

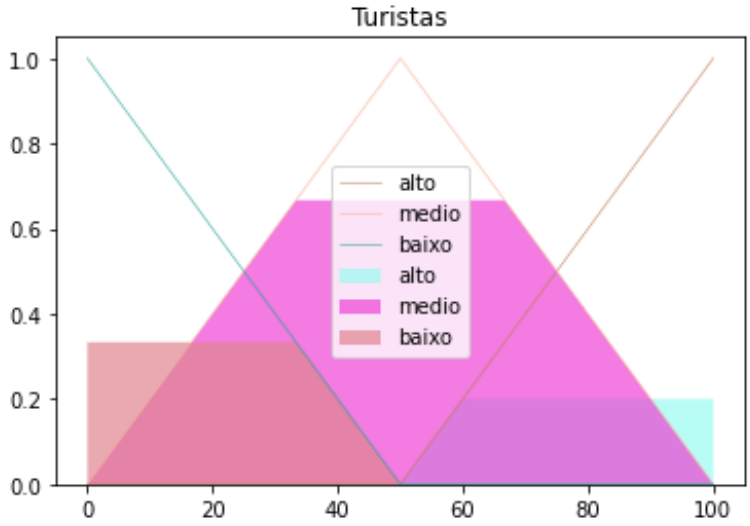
Figura 4 - Tela do resultado.



Elaboração do autor, 2020.

5. Após salvar a config Json, deve-se executar o arquivo “fuzzy.exe” que exibirá um terminal com o resultado calculado (figura 11). O arquivo “fuzzy.exe” verificará a pertinência de cada atributo para cada variável configurada. O sistema calcula a pertinência de cada regra para cada variável de saída da regra. Para realizar isso a solução desenvolvida utiliza a pertinência dos atributos e aplica um mecanismo de inferência. Esse mecanismo acopla múltiplas condições dos atributos com operadores e retorna um valor único para a regra. Ao finalizar o cálculo de todas as regras, chamado de processo de implicação, as que possuem maior pertinência para cada atributo da variável de saída são selecionadas. Por fim, é executada a agregação das saídas com o objetivo de calcular o centroide. O valor retornado por essas regras (de 0 a 1) vai definir o teto da curva no gráfico da variável de saída. Exemplo (gráfico 8):

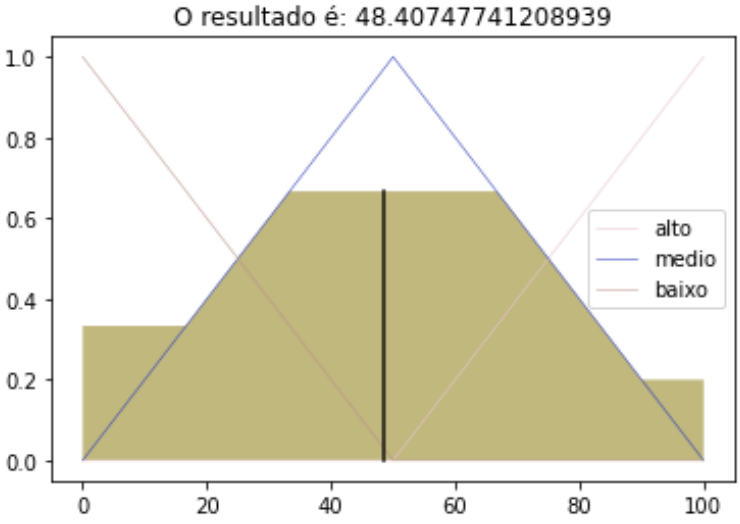
Gráfico 8 - Teto da curva.



Fonte: Elaboração do autor, 2020.

Neste momento é possível realizar a agregação das saídas e consequentemente o cálculo do centroide, onde é calculado o centro da gravidade. Com isso é exibido o valor no eixo x do centroide, que é o resultado deste exemplo (gráfico 9).

Gráfico 9 - Resultado



Fonte: Elaboração do autor, 2020.

Esse protótipo foi validado pela turma da disciplina Modelos Evolucionários e Tratamento de Incertezas, ofertada no semestre 2020/1 com 25 alunos. Para isso, encaminhou-se um trabalho para estes analisarem o framework e identificar pontos positivos e negativos da ferramenta. Paralelo a isto, os alunos analisaram outro framework já conhecido e desenvolvido em Java, o Jfuzzylogic, com o objetivo de realizar comparações.

Os estudantes avaliaram e testaram ambos modelos e contribuíram com suas impressões referentes ao protótipo desenvolvido. De acordo com estes estudantes seguem os pontos positivos e negativos da solução.

Prós:

- Menu é intuitivo e usual.
- Disposição dos objetos na tela torna o aprendizado rápido para realizar os testes.
- Sistema é preciso e eficiente.
- Possibilidade de salvar dados para uso posterior. Outro ponto positivo é a
- Facilidade em alterar os valores das variáveis de entrada.
- Execução é rápida e consistente.

Contras:

- Necessidade de abrir um outro aplicativo para ver os resultados
- Sem informações sobre os campos cadastrados
- Limitação a duas condições por regras. Outro ponto negativo é a
- Falta de um feedback pelo sistema quando os dados são inseridos com sucesso
- Os nomes dos campos poderiam não ser abreviados, por dificultar a leitura e entendimento deles. Sentiram
- Falta uma listagem de variável adicionada, pois na medida que vão preenchendo os dados, podem esquecer de alguma variável.

Além de listarem os prós e contras, os estudantes deixaram sugestões de melhoria para o sistema. Dentre essas, analisamos e selecionamos o que poderia ser utilizado no projeto final.

Pontos de melhoria:

- Limpar os campos ao finalizar o cadastro.
- Possibilitar a visualização e alteração de campos ao clicar em uma variável
- Incluir mensagens para o usuário quando um dado foi cadastrado
- Remover a abreviação das palavras
- Incluir gráficos na visualização de problemas com os valores das variáveis
- Definir o idioma do sistema.
- Exibir o resultado na interface gráfica

Após a realização desta experiência com os estudantes deu-se início ao projeto final. Dentre os ajustes necessários, foi definido que o sistema deveria ser hospedado em um site, sendo desenvolvida uma interface web para tal.

5 CONSIDERAÇÕES FINAIS

Conclui-se que, para tratar de casos com imprecisões, a lógica fuzzy torna-se uma ferramenta muito útil. Problemas complexos de diversas áreas da atualidade podem ser resolvidos utilizando essa tecnologia. Porém, a lógica fuzzy ainda é pouco disseminada pelo fato de possuir a necessidade de desenvolver um sistema complexo do início ao fim para resolver um problema específico. Nesse caso, desenvolver uma ferramenta Shell que possibilita construir um modelo de sistema e consultar uma máquina de inferência difusa resolve essa problemática e incentiva o uso da tecnologia por especialistas das mais diversas áreas, com treinamento e conhecimento dos conceitos aqui colocados. Por ser expressado em termos humanos, é intuitivo e permite resolver problemas dos mais complexos ao mais simples com naturalidade. A ferramenta descrita nesse artigo se mostrou eficaz para ser aplicada por um especialista, independente da área de atuação, mas com um problema de imprecisão. Isto posto, consideramos para implementações futuras, possibilitar ao usuário alterar as métricas diretamente no gráfico, de forma que altere o núcleo e suporte dos campos de maneira visual. Além disso, disponibilizar a ferramenta no domínio da UNISUL, permitindo professores, alunos e a comunidade em geral, utilizar a ferramenta. Entendemos também, a limitação no que se refere ás condições das regras, e consideramos implementar condições que permitam gerar regras mais complexas, pois atualmente a nossa solução permite tratamento das condições das regras de forma linear (vide Figura 2), o que pode não ser adequado em ambientes mais complexos, onde a ordem com que as condições são calculadas importa.

REFERÊNCIAS

[RIGNEL, CHENCI e LUCAS, 2011] RIGNEL, Diego G. S., CHENCI, Gabriel P., LUCAS, Carlos A. **Uma Introdução a Lógica Fuzzy.** Revista Eletrônica de Sistemas da Informação e Gestão Tecnológica. Vol. 01, 2011.

[MUKAIDONO, 2001] MUKAIDONO, Mazao. **Fuzzy Logic For Beginners.** Singapore: World Scientific, 2001.

[SILVA, 2005] SILVA, Renato Afonso Cota. **Inteligência artificial aplicada à ambientes de Engenharia de Software: Uma visão geral.** Universidade Federal de Viçosa, 2005.

[WEBER e KLEIN, 2003, p. 23] WEBER, Leo e KLEIN, Pedro Antonio Trierweiler. **Aplicação da Lógica Fuzzy em Software e Hardware.** Canoas: Ed. Ulbra, 2003.

[NETO e PEREIRA. 2011] NETO, Laudelino M. C., PEREIRA, Amanda V. **CREDSAPIENS. Sistema Especialista Difuso de apoio na análise de crédito.** Tubarão, 2011 (Artigo disponibilizado pelo autor em Maio 2019)

[FERNANDES, 1996] FERNANDES, Anita M. R. **Sistema Especialista Difuso aplicado ao processo de análise química qualitativa de amostras de minerais.** Fevereiro, 1996. Disponível em: https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/76453/103503.pdf?sequence=1&isAllowed=y (Acessado em Maio, 2019)

[FERRARI, ARGOUD e AZEVEDO, 2004] FERRARI, G. L.; ARGOUD, F. I. M.; AZEVEDO, F. M**. Shell para Desenvolvimento de Sistemas Especialistas Fuzzy – Estudo de Caso: Gastroenterologia, IV Workshop de Informática aplicada à Saúde** – CBComp (Outubro 2004 : Itajaí, Santa Catarina). Anais. Santa Catarina, 2004. p. 583-588.

[POSSELT, FROZZA, MOLZ, 2011] Posselt, E. L.; Frozza, R.; Molz, R. F.. Software Infuzzy 2011. **Programa de Mestrado em Sistemas e Processos Industriais** PPGSPI, UNISC, 2011. Disponível em: **http://www.unisc.br/ppgspi**

AGRADECIMENTOS

Beatriz

Aos meus avós (in memorian) pelo amor, carinho e apoio concedidos durante momentos significativos da minha vida. Saudades.

A minha filha Mirana, meu incentivo diário e motivo de toda minha dedicação.

A minha mãe Katia e minha tia Márcia, pelo amor, incentivo e apoio financeiro.

Ao meu namorado Thiago, pelo amor, pela paciência e por compreender minha ausência pelo tempo dedicado aos estudos.

Ao nosso orientador Clávison, pela confiança depositada na nossa proposta, pela dedicação do seu escasso tempo ao nosso projeto e pelo apoio incondicional.

Ao professor Max, por inspirar a elaboração desta proposta. Obrigado por nos manter motivados durante todo o processo.

A esta universidade, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior.

Em especial, aos meus colegas Paulo, Danilo e Gabriel pela amizade, pelo suporte e pelos anos de convívio durante a graduação.

Paulo

A minha família, por ter confiado em minha capacidade.

Aos meus amigos que me acompanharam durante os últimos 5 anos.

Ao nosso orientador Clávison, pela sua confiança depositada.

E ao nosso professor Max, por toda sua inspiração.