



PROTÓTIPO DE SISTEMA DE APOIO À DECISÃO COM LÓGICA FUZZY

Álvaro de Araújo
Ana Heloísa

ROTEIRO

- descrição do problema
- plataforma e ferramentas utilizadas
- algoritmos implementados
- bibliografia



DESCRIÇÃO DO PROBLEMA

O objetivo deste projeto é desenvolver um sistema de recomendação de compra de carros usados utilizando lógica fuzzy. O sistema será capaz de recomendar carros com base nas seguintes variáveis: eficiência de combustível, quilometragem, taxa e valor do carro.





Plataforma e ferramentas

- Python
- Pandas
- skit-fuzzy
- numpy
- matplotlib
- VSCode
- poetry
- MacOS e Linux Arch

Algoritmos implementados

Carregamento dos Dados: Os dados sobre os carros usados, incluindo preço, taxa, eficiência de combustível (km por litro) e quilometragem, são carregados a partir de um arquivo CSV para um DataFrame do pandas.

Definição das Variáveis Fuzzy: As variáveis fuzzy são definidas para representar as características dos carros, como preço, taxa, eficiência de combustível e quilometragem. Cada variável é dividida em conjuntos fuzzy que representam diferentes faixas de valores, como "baixo", "médio" e "alto".

Algoritmos implementados

Criação das Regras do Sistema de Controle: definidas regras de inferência fuzzy que relacionam as variáveis de entrada (preço, taxa, eficiência de combustível e quilometragem) com a variável de saída (benefício). As regras são formuladas de acordo com o conhecimento especializado sobre carros usados e a lógica fuzzy.

Criação do Sistema de Controle e Simulação: sistema de controle fuzzy é criado com base nas variáveis fuzzy e nas regras definidas. Em seguida, uma simulação do sistema de controle é realizada para cada entrada de carro no DataFrame, calculando o benefício da compra de acordo com as regras e variáveis fuzzy.

Algoritmos implementados

Visualização dos Resultados: Os resultados da simulação, ou seja, o benefício da compra de cada carro, são visualizados graficamente para facilitar a compreensão e análise.

Algoritmos implementados

Variáveis e funções de pertinência

```
# variáveis fuzzy
preco = ctrl.Antecedent(np.arange(df['preco'].min(), df['preco'].max(), 1), 'preco')
taxa = ctrl.Antecedent(np.arange(df['taxa'].min(), df['taxa'].max(), 1), 'taxa')
km_por_litro = ctrl.Antecedent(np.arange(df['km/l'].min(), df['km/l'].max(), 1), 'km_por_litro')
km_rodado = ctrl.Antecedent(np.arange(df['km'].min(), df['km'].max(), 1), 'km_rodado')
beneficio = ctrl.Consequent(np.arange(0, 11, 1), 'beneficio')

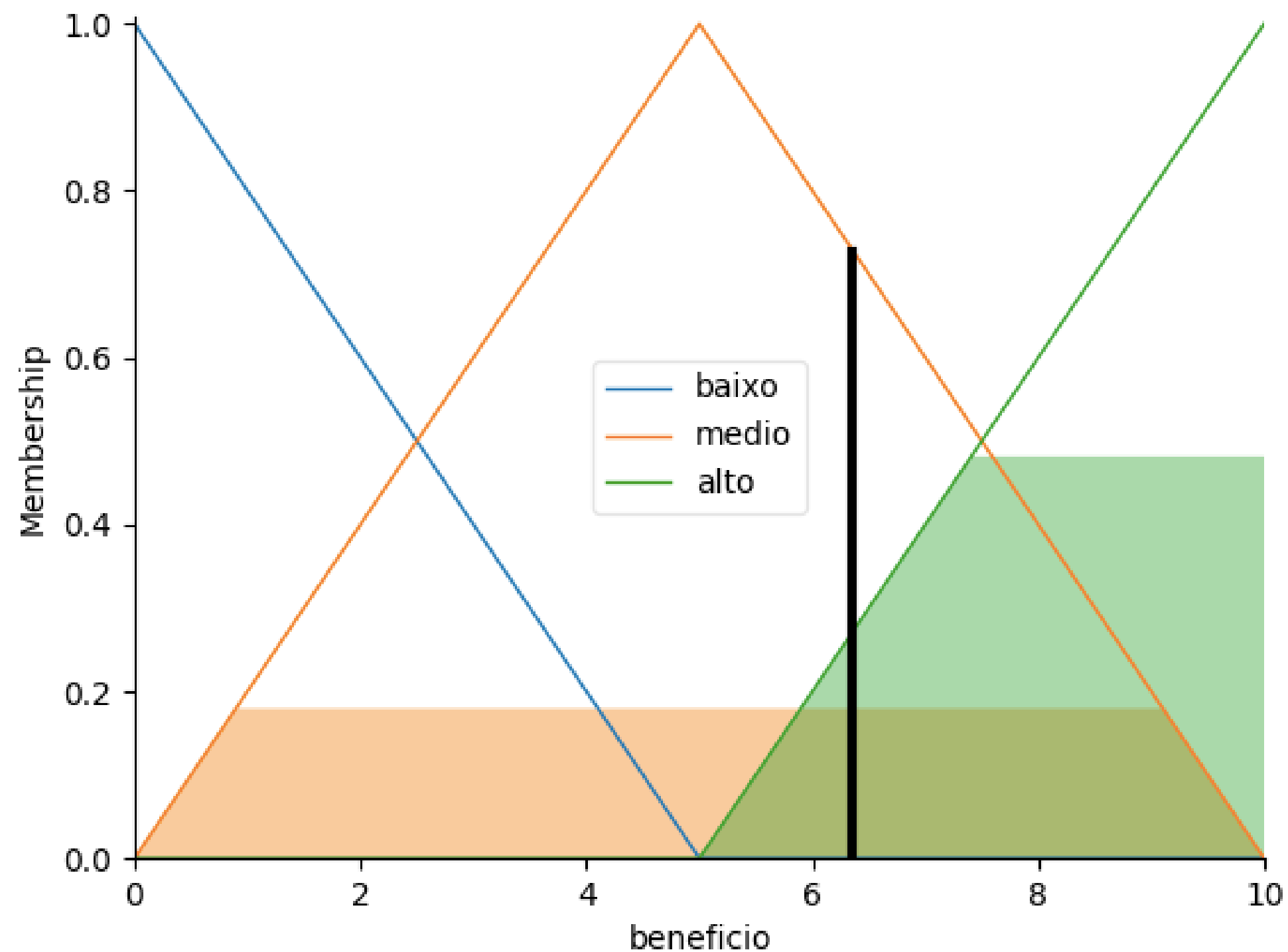
# funções de pertinência
preco.automf(number=3, names=['baixo', 'medio', 'alto'])
taxa.automf(number=3, names=['baixo', 'medio', 'alto'])
km_por_litro.automf(number=3, names=['alto', 'medio', 'baixo'])
km_rodado.automf(number=3, names=['baixo', 'medio', 'alto'])
beneficio.automf(number=3, names=['baixo', 'medio', 'alto'])
```


Algoritmos implementados

Regras

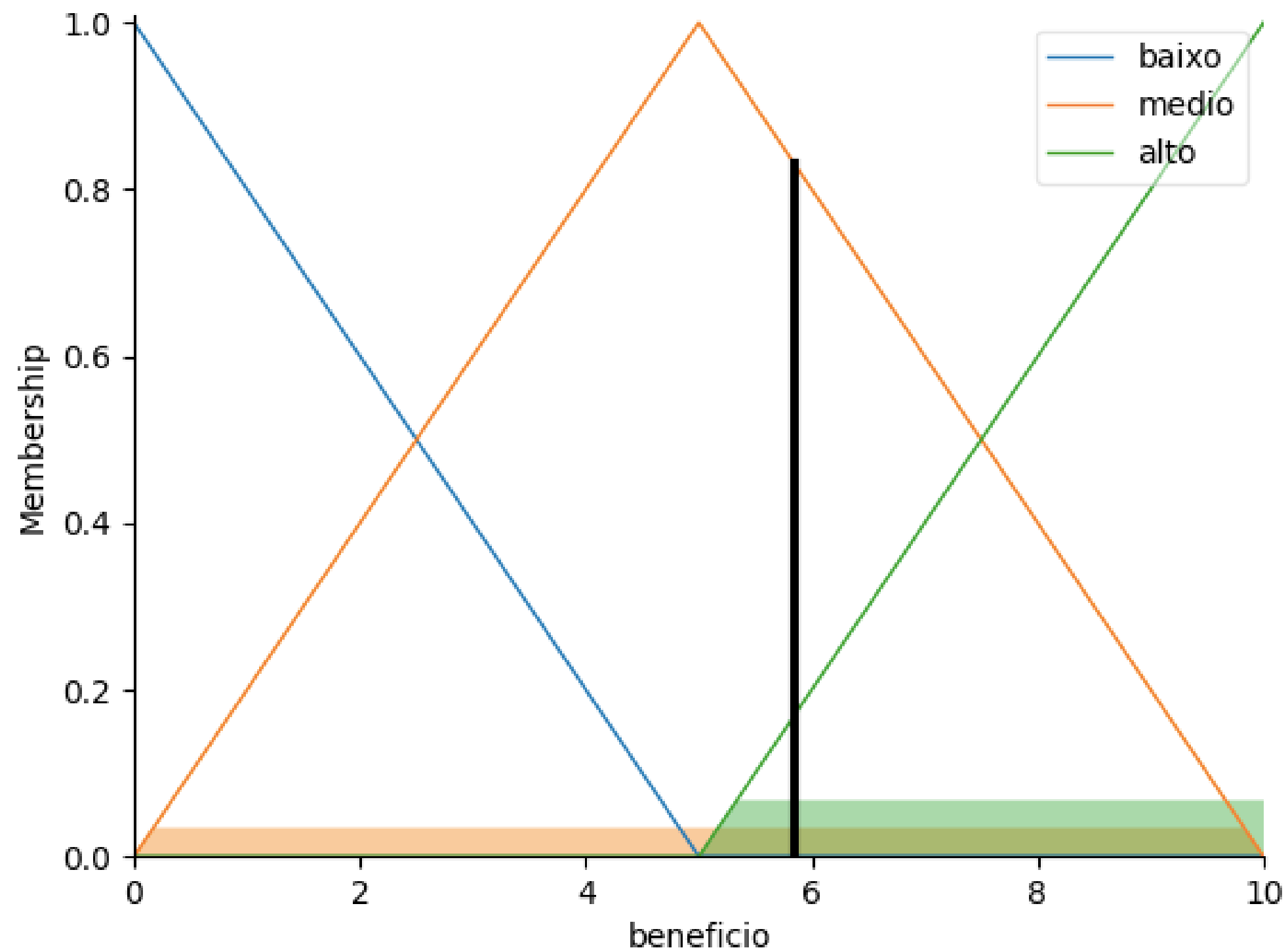
```
regras = [  
    ctrl.Rule(preco['baixo'] & taxa['baixo'] & km_por_litro['medio'] & km_rodado['alto'], beneficio['medio']),  
    ctrl.Rule(preco['baixo'] & taxa['baixo'] & km_por_litro['alto'] & km_rodado['baixo'], beneficio['alto']),  
    ctrl.Rule(preco['baixo'] & taxa['medio'] & km_por_litro['baixo'] & km_rodado['alto'], beneficio['medio']),  
    ctrl.Rule(preco['baixo'] & taxa['medio'] & km_por_litro['medio'] & km_rodado['alto'], beneficio['medio']),  
    ctrl.Rule(preco['baixo'] & taxa['alto'] & km_por_litro['medio'] & km_rodado['alto'], beneficio['baixo']),  
    ctrl.Rule(preco['baixo'] & taxa['alto'] & km_por_litro['baixo'] & km_rodado['alto'], beneficio['baixo']),  
    ctrl.Rule(preco['baixo'] & taxa['baixo'] & km_por_litro['baixo'] & km_rodado['medio'], beneficio['alto']),  
    ctrl.Rule(preco['alto'] & taxa['baixo'] & km_por_litro['baixo'] & km_rodado['medio'], beneficio['alto']),  
    ctrl.Rule(preco['alto'] & taxa['baixo'] & km_por_litro['alto'] & km_rodado['alto'], beneficio['baixo']),  
    ctrl.Rule(preco['alto'] & taxa['medio'] & km_por_litro['baixo'] & km_rodado['baixo'], beneficio['medio']),  
    ctrl.Rule(preco['alto'] & taxa['medio'] & km_por_litro['medio'] & km_rodado['medio'], beneficio['medio']),  
    ctrl.Rule(preco['alto'] & taxa['medio'] & km_por_litro['alto'] & km_rodado['alto'], beneficio['baixo']),  
    ctrl.Rule(preco['alto'] & taxa['alto'] & km_por_litro['baixo'] & km_rodado['baixo'], beneficio['baixo']),  
    ctrl.Rule(preco['alto'] & taxa['alto'] & km_por_litro['medio'] & km_rodado['medio'], beneficio['baixo']),  
    ctrl.Rule(preco['alto'] & taxa['alto'] & km_por_litro['alto'] & km_rodado['alto'], beneficio['baixo']),  
    ctrl.Rule(preco['medio'] & taxa['baixo'] & km_por_litro['baixo'] & km_rodado['baixo'], beneficio['alto']),  
    ctrl.Rule(preco['medio'] & taxa['baixo'] & km_por_litro['medio'] & km_rodado['baixo'], beneficio['alto']),  
    ctrl.Rule(preco['medio'] & taxa['baixo'] & km_por_litro['alto'] & km_rodado['medio'], beneficio['alto']),  
    ctrl.Rule(preco['medio'] & taxa['medio'] & km_por_litro['medio'] & km_rodado['medio'], beneficio['medio']),  
    ctrl.Rule(preco['medio'] & taxa['medio'] & km_por_litro['alto'] & km_rodado['alto'], beneficio['baixo']),  
    ctrl.Rule(preco['medio'] & taxa['alto'] & km_por_litro['baixo'] & km_rodado['medio'], beneficio['medio']),  
    ctrl.Rule(preco['medio'] & taxa['alto'] & km_por_litro['alto'] & km_rodado['alto'], beneficio['baixo']),  
]
```

Algoritmos implementados



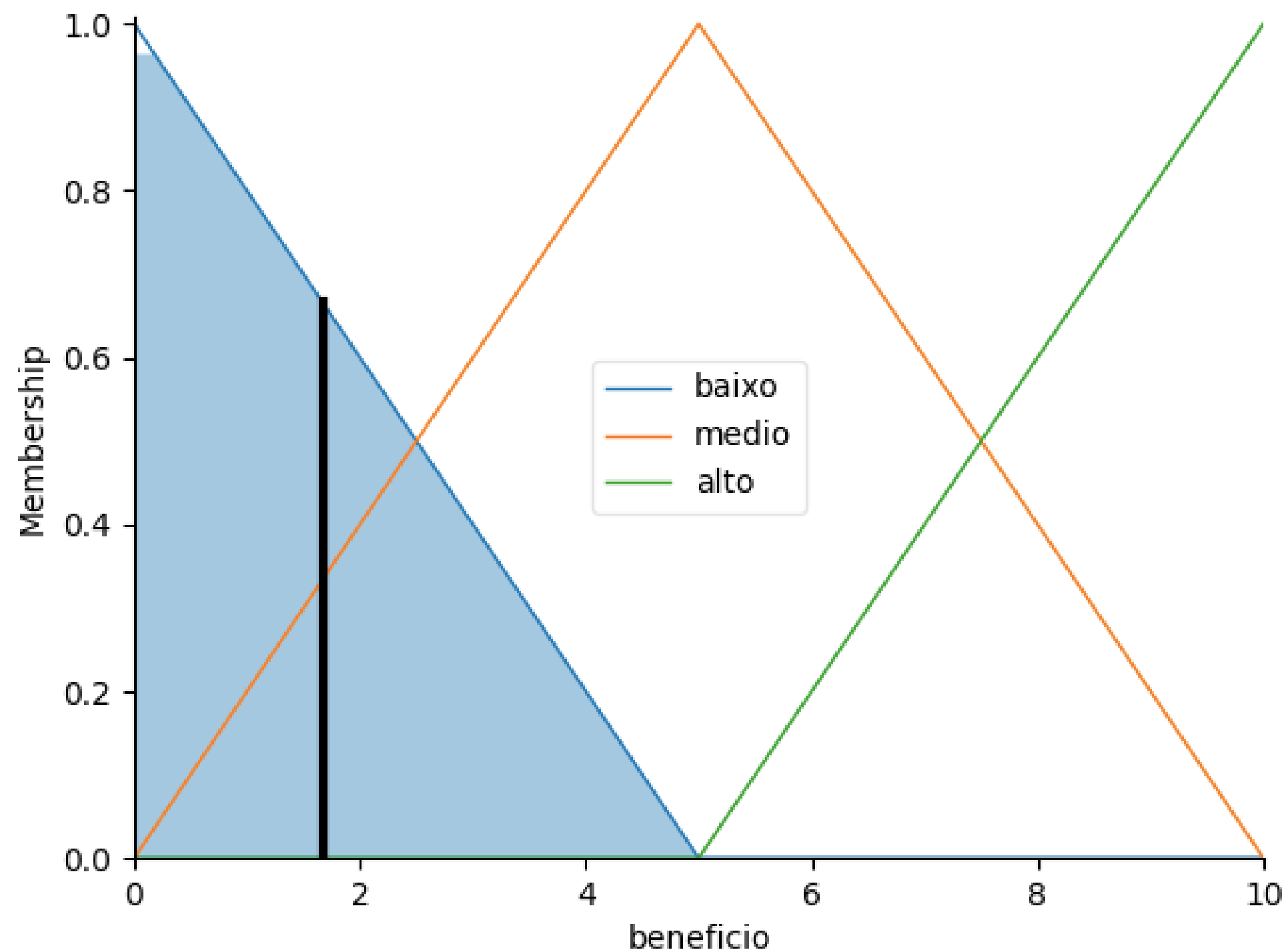
preço	taxa	km/L	km
12000	150	25	25659

Algoritmos implementados



preço	taxa	km/L	km
5000	270	14	15659

Algoritmos implementados



preço	taxa	km/L	km
90000	300	10	905659

A dark, moody background image featuring a laptop, a pair of sunglasses, and the number '15' in a large, light font.

BIBLIOGRAFIA

kaggle:

[https://www.kaggle.com/datasets/adh
urimquku/ford-car-price-prediction](https://www.kaggle.com/datasets/adh
urimquku/ford-car-price-prediction)

skit-fuzzy: [https://scikit-
fuzzy.readthedocs.io/en/latest/](https://scikit-
fuzzy.readthedocs.io/en/latest/)