

**Disciplina:** inteligência Artificial

**Acadêmico:** LUCAS BEZERRA MARTINS

**R.A:** 125111361589

**Acadêmico:** PAULO HENRIQUE BRITO

**R.A:** 125111377593

1. Uma descrição do funcionamento do algoritmo, histórico e características (2 ponto) e áreas onde o algoritmo pode ser aplicado com exemplos de problemas que podem ser resolvidos pelo mesmo (1 ponto);

A Teoria de Conjuntos Fuzzy foi concebida por L.A. Zadeh com o objetivo de fornecer uma ferramenta matemática para o tratamento de informações de caráter impreciso ou vago. A Lógica Fuzzy, baseada nessa teoria, foi inicialmente construída a partir dos conceitos já estabelecidos de lógica clássica; operadores foram definidos à semelhança dos tradicionalmente utilizados e outros foram introduzidos ao longo do tempo, muitas vezes por necessidades de caráter eminentemente prático.

Essa abordagem permite atribuir um valor de pertinência a cada faixa de avaliação para cada categoria (Aclamado, Regular, Odiado). O algoritmo Fuzzy, então, utiliza esses valores de pertinência para calcular a classificação final do filme em uma das três categorias.

Essa classificação Fuzzy pode ser útil em sistemas de recomendação de filmes, onde os usuários podem receber recomendações personalizadas com base em suas preferências e classificações anteriores. Também pode ser aplicada em análises de avaliações de produtos, avaliação de feedback de clientes, entre outros cenários em que seja necessário classificar avaliações em categorias específicas.

2. Demonstração do algoritmo (implementação) na resolução de um problema. O problema deve ser real, ou seja, deve utilizar uma base de dados real e que não esteja previamente “limpa” (5 pontos).

Fizemos 3 funções, cada uma delas aplica a regra Fuzzy para as categorias escolhidas (Odiado, Regular, Aclamado)

```
def hated(rating: float) -> float:
    if rating >= 3.0:
        return 0.0
    elif rating <= 1.5:
        return 1.0
    else:
        return (rating - 1.5) / 1.5

def regular(rating: float) -> float:
    if rating <= 1.5:
        return 0.0
    elif rating > 1.5 and rating < 2.5:
        return (rating - 1.5) / 1.5
    elif rating >= 2.5 and rating <= 3.5:
        return 1.0
    elif rating > 3.5 and rating < 4.5:
        return (rating - 3.5) / 1.5
    else:
        return 0.0

def loved(rating: float) -> float:
    if rating <= 2.0:
        return 0.0
    elif rating >= 4.0:
        return 1.0
    else:
        return (rating - 2.0) / 2.0
```

O resultado do algoritmo:

```
{
  "Dangerous Minds": [
    "3.0",
    [
      "50.0%",
      "100.0%",
      "0.0%"
    ]
  ]
}
```

A descrição do resultado:

```
{
  "NAME": [
    "RATING",
    [
      "LOVED",
      "REGULAR",
      "HATED"
    ]
  ]
}
```