

Your waifu is trash – A Fuzzy Recommendation System

LUCY MIYUKI MIYAGUSIKU NARITA *

WILLIAM GONÇALVES DA CRUZ †

Ciência da Computação – Graduação

*RA: 182851

†RA: 188671

I. INTRODUÇÃO

Representações de características de itens de natureza subjetiva, vaga e com relações imprecisas apresentam grandes desafios para sistemas de recomendações baseados em preferência de usuário.

Essas imprecisões não estocásticas que se dão devido a subjetividade e imprecisão dos dados devem ser levadas em conta na hora de prever a probabilidade de um dado usuário gostar ou não daquele item.

Dito isso, acreditamos que a lógica *Fuzzy* pode nos ajudar a tratar a inexatidão dos dados para esses casos.

II. TRABALHO PROPOSTO

Para este projeto, propomos um rascunho de um sistema de recomendações de itens para um dado perfil (usuário) baseado em lógica *Fuzzy*.

A ideia original surgiu de apresentarmos um sistema que programaticamente decide para qual lado o usuário faria um *swipe* no popular aplicativo de relacionamentos: *Tinder*. (esquerda: *dislike*, direita: *like*, cima: *super like*)

Entretanto, para o escopo deste projeto decidimos não nos basear em pessoas reais para fins de testes, por questões éticas envolvidas. Como alternativa, decidimos basear nosso sistemas em *Waifus*[1]: personagens fictícias, normalmente de um *anime* ou jogo, que são consideradas objetos de afeto para algum grupo ou indivíduo.

O código pode ser encontrado [neste repositório](#)¹.

A. Linguagem escolhida

A linguagem utilizada foi Python, decidida assim pois é conhecida pelo grupo, além de trazer facilidade em desenvolver com sua lib *skfuzzy*, ótimos desenhos de gráficos com *matplotlib* e simplicidade em realizar uma demonstração da recomendação com uma lib de janelas visuais, *tkinter*.

Todas as bibliotecas e suas versões utilizadas podem ser encontrados no arquivo `requirements.txt`.

B. Modelagem do problema

O trabalho proposto gira em torno de duas estruturas principais para a definição do problema: Usuário e Waifu. Somente a partir da relação entre elas é possível esclarecer o sistema de regras e inferências. Logo, primeiramente devemos definir essas estruturas de maneira mais formal.

1) *Waifu*: Na visão deste trabalho, uma waifu é modelada com visão a ter suas características mapeadas em dados quantitativos, baseados na cognição de como essa entidade é percebida pelo público. De forma a reduzir o problema, selecionamos uma pequena parcela de características e amostramos os dados para algumas waifus em questão. A Figura 1 apresenta um exemplo gráfico de como seria essa representação.

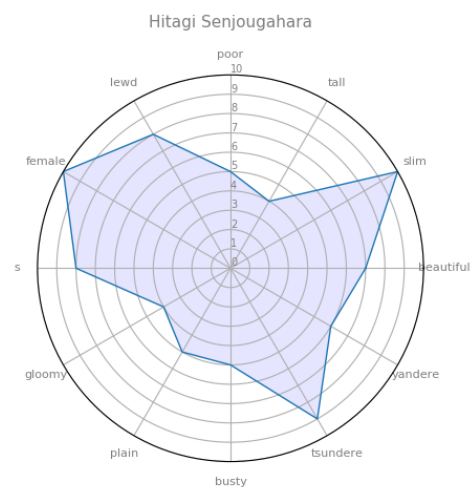


Figura 1. Hitagi Senjougahara Trait Chart

2) *Usuário*: A fim de relacionar usuário a *waifu* como é a intenção do trabalho, é preciso trabalhar com a definição de um perfil para o usuário. Assim, um perfil é definido pela afinidade que um usuário tem com a combinação de características de uma *waifu*, se há preferência alta ou baixa na existência daquela característica ou até mesmo inexistência para alguns casos.

Dessa forma, a relação entre um usuário e várias *waifus*

¹<https://github.com/lnarita/mc906a/tree/master/03>

pode ser descrita, onde a correlação entre as combinações do perfil e as características da própria *waifu* nos traz uma condição que pode ser avaliada na recomendação, que será melhor explicada adiante.

Por fim, o usuário também define como funcionam suas funções de pertinência, baseadas nas entradas de características. Elas podem variar de formato e intervalo, de forma a melhor descrever seu gosto.

Abaixo, segue um exemplo gráfico de um perfil para usuário:

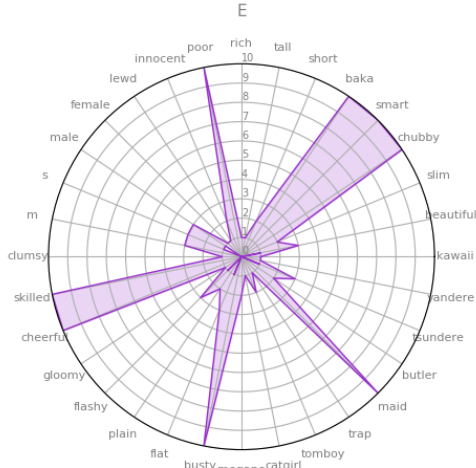


Figura 2. Exemplo relação do perfil de usuário com as características, nível E (menor preferência do usuário)

C. Sistemas de regras

Cada usuário deve necessariamente definir 3 regras para definir as seguintes possíveis categorias que uma *waifu* pode se encaixar:

- é best girl? (Maior preferência)
- é waifu? (Média preferência)
- é lixo? (Baixa preferência)

Essas regras devem utilizar qualquer número e combinações de características para mapear para um determinado nível de preferência (*likeness*).

D. Modelo de inferência

O modelo adotado foi Mamdani, pois é intuitivo e bem aceito, tornando-o mais simples de se trabalhar.

E. Modelo de recomendação

Tendo em vista todas as definições acima, a implementação da recomendação se baseia no resultado defuzzificado da operação de combinar um perfil de um usuário a uma *waifu* (afinidade). A partir dele, é calculado o pertencimento deste valor CRISP nos conjuntos de regras definidos para a afinidade. Assim, o conjunto ao qual houver maior pertencimento define a recomendação da *waifu* para aquele perfil.

Desta forma, a interação de nosso algoritmo com o tinder segue da seguinte maneira:

- Alta afinidade Super like
- Media afinidade Like
- Baixa afinidade Dislike

III. RESULTADOS E DISCUSSÃO

Para a análise da solução proposta, utilizaremos um perfil de usuário computado, que possui 3 (três) implementações distintas para comparação. Este mesmo perfil será mensurado com relação a 6 *waifus* distintas entre si, de modo a mensurar a resposta das implementações para todos os casos.

O perfil em questão é descrito no sistema como tendo uma alta preferência pelas características muito presentes de *tall*, *lewd*, *s*, *flashy*, *busty*, *beautiful*, *female*, juntamente baixa presença de características como *short*, *yandere*, *innocent*, *baka*, tendo nível médio de afinidade por *flat*.

Em contraste, sua baixa preferência é definida por altos índices das características *short*, *yandere*, *m*, *cheerful*, *baka*, *kawaii*, combinadas a baixa incidência de *lewd*, *flashy*.

Conjunto de waifus avaliadas

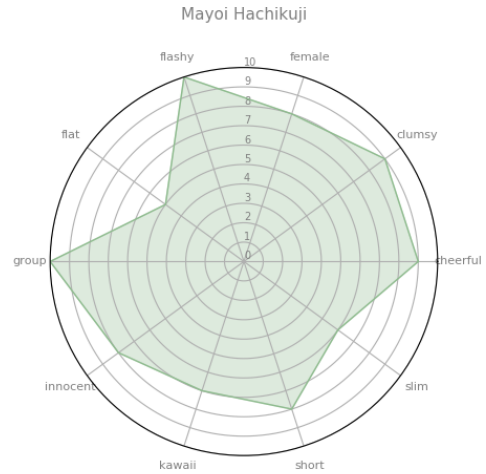


Figura 3. Waifu 1 - Trait chart
Tsubasa Hanekawa



Figura 4. Waifu 2 - Trait chart

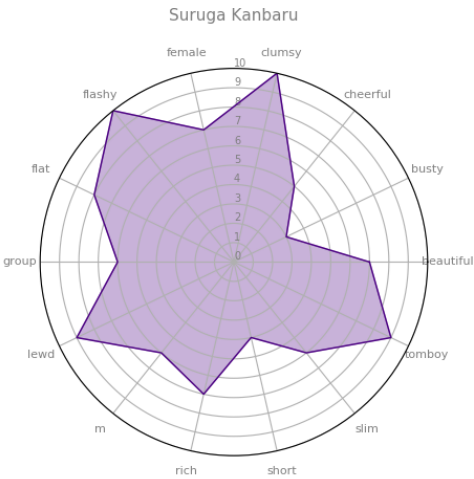


Figura 5. Waifu 3 - Trait chart
Nadeko Sengoku

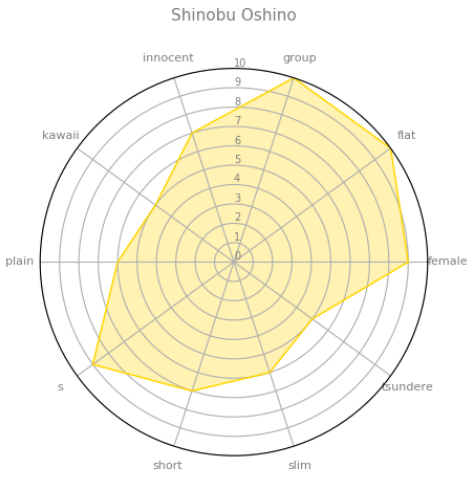


Figura 8. Waifu 6 - Trait chart

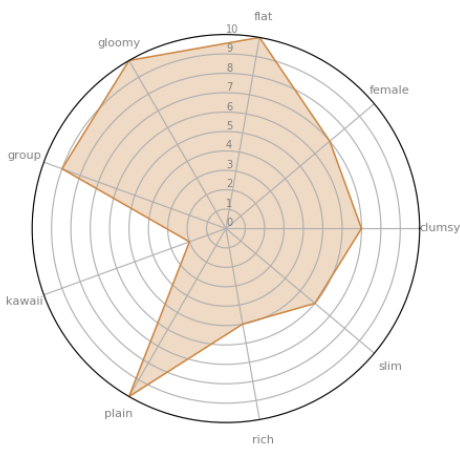


Figura 6. Waifu 4 - Trait chart
Hitagi Senjougahara

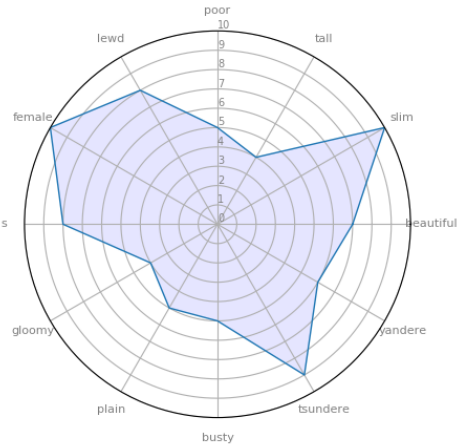


Figura 7. Waifu 5 - Trait chart

Primeira implementação

Sua definição funciona a partir de funções de *Auto-membership* para as combinações de características no perfil e também para a consequente de *likeness*.

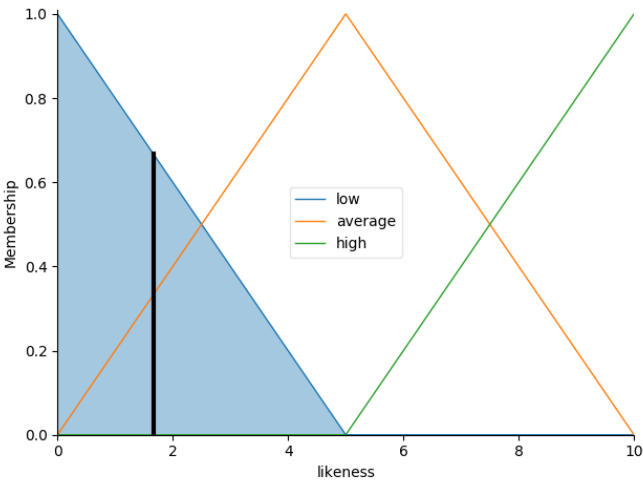


Figura 9. Resultado pertencimento da função de likeness - Waifu 1

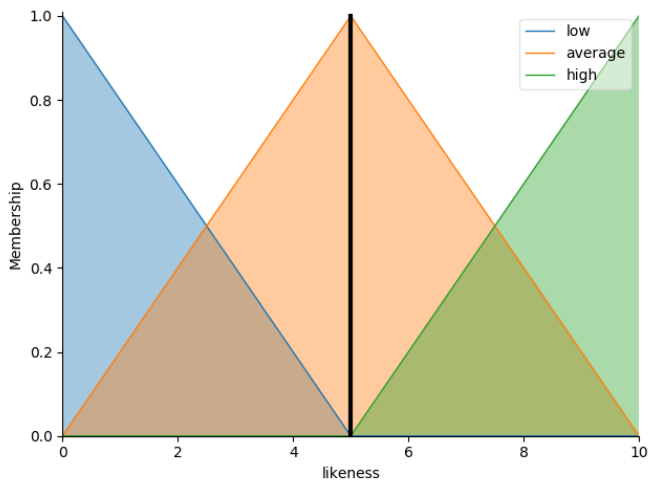


Figura 10. Resultado pertencimento da função de likeness - Waifu 2

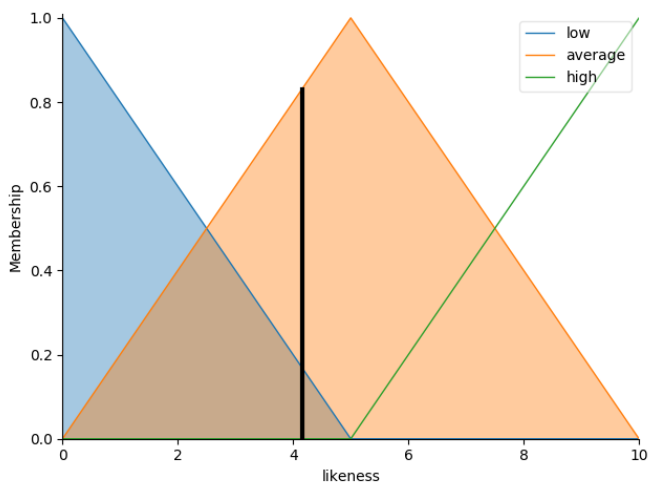


Figura 11. Resultado pertencimento da função de likeness - Waifu 3

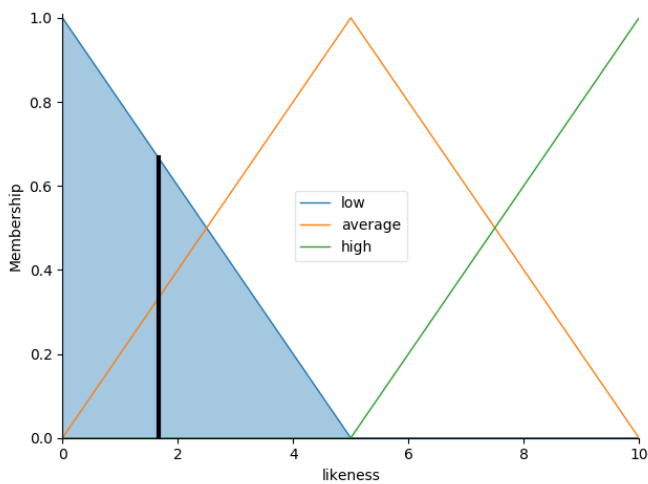


Figura 12. Resultado pertencimento da função de likeness - Waifu 4

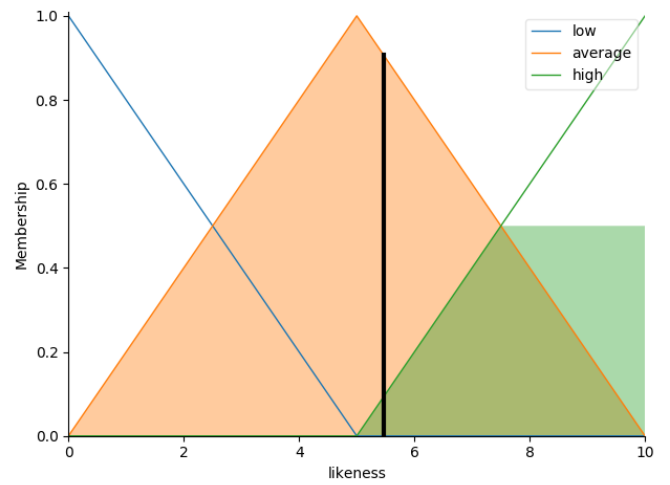


Figura 13. Resultado pertencimento da função de likeness - Waifu 5

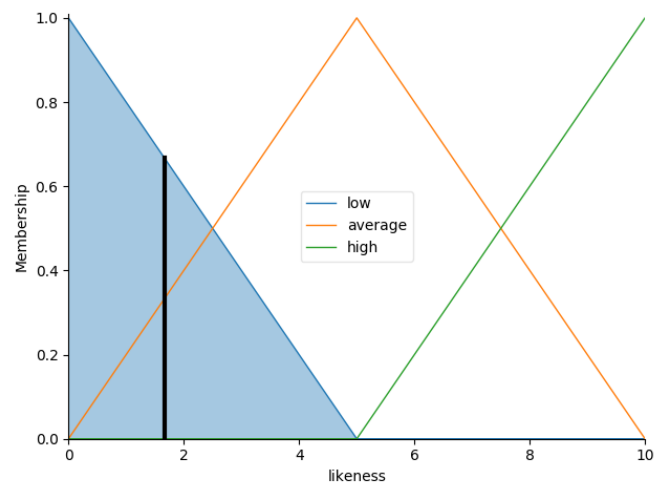


Figura 14. Resultado pertencimento da função de likeness - Waifu 6

Segunda implementação

Nessa implementação cada relação de característica com nível de preferência é definida por uma função trapezoidal, diferentemente do caso anterior. Porém, é mantida *Auto-membership* para a fuzzyficação da variável de *likeness*. Segue abaixo um exemplo para uma característica de como seria a função trapezoidal:

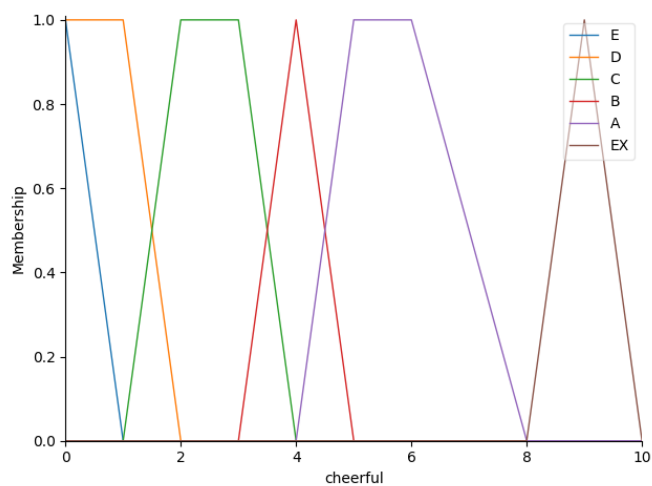


Figura 15. Exemplo de função de pertinência trapezoidal

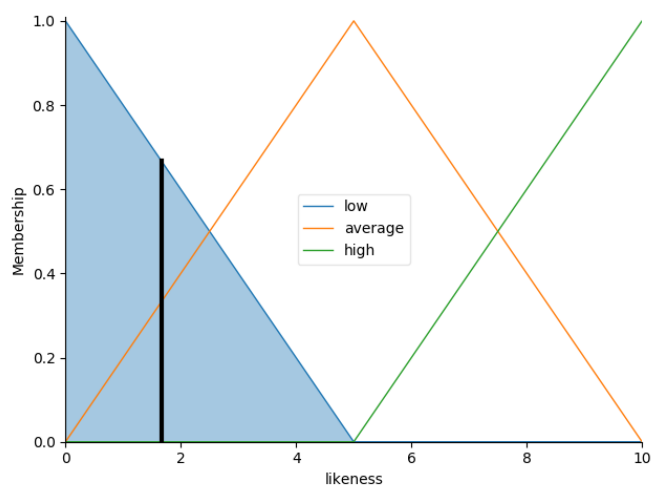


Figura 16. Resultado pertencimento da função de likeness - Waifu 1

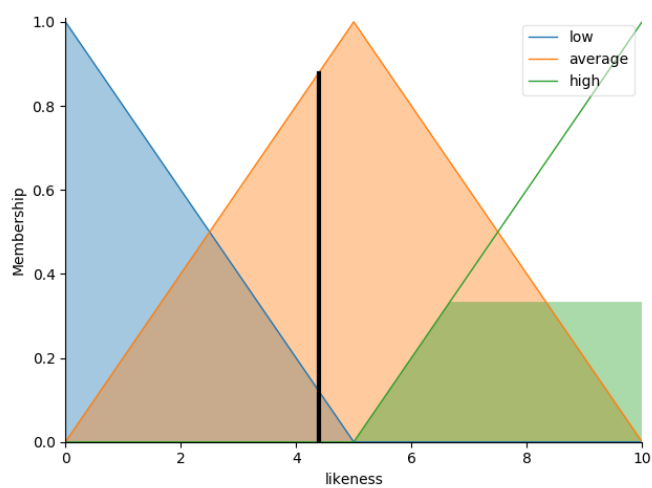


Figura 17. Resultado pertencimento da função de likeness - Waifu 2

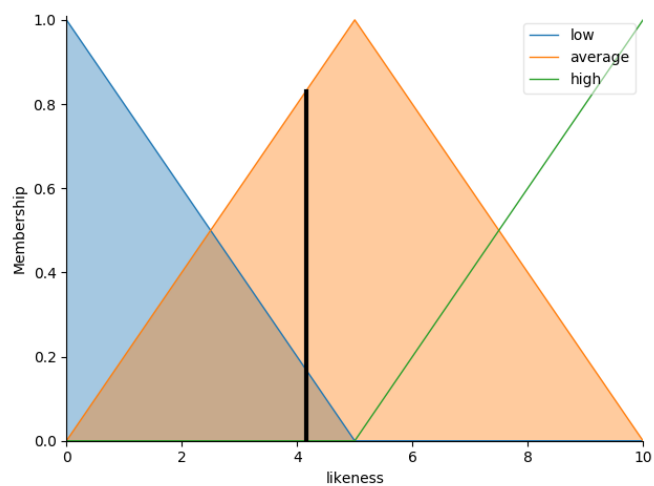


Figura 18. Resultado pertencimento da função de likeness - Waifu 3

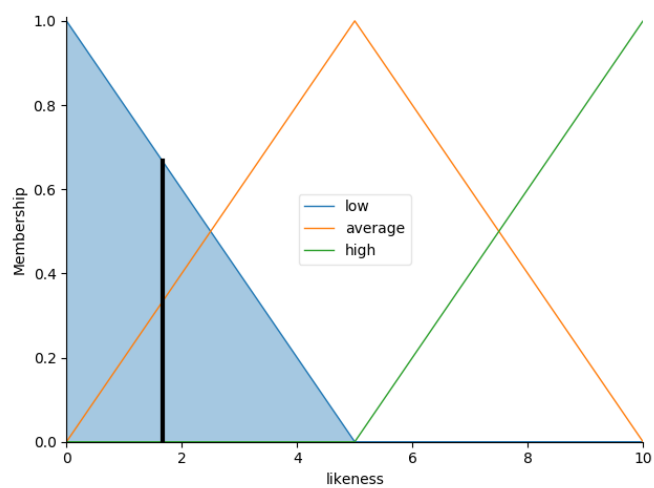


Figura 19. Resultado pertencimento da função de likeness - Waifu 4

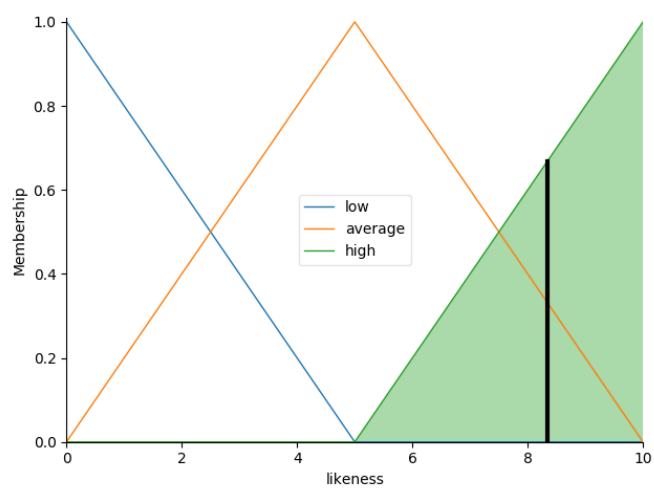


Figura 20. Resultado pertencimento da função de likeness - Waifu 5

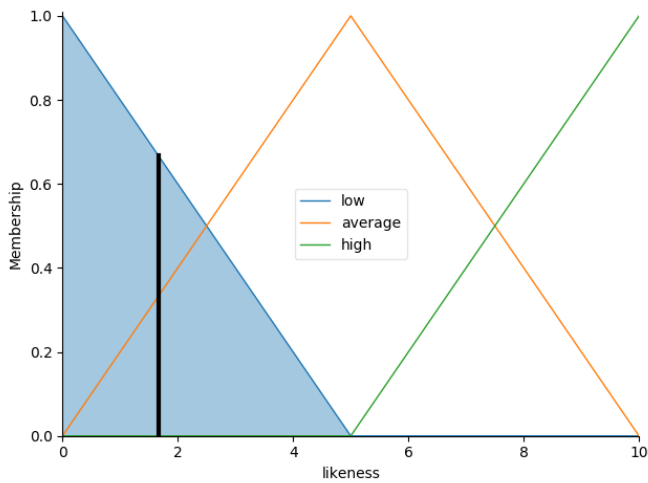


Figura 21. Resultado pertencimento da função de likeness - Waifu 6

Terceira implementação

Por último, nessa variação de implementação, as funções para as características se mantêm iguais em relação a primeira implmentação, com mudanças apenas no funcionamento da *likeness*. Esta por sua vez foi definida com 3 (três) funções de pertecimento diferentes. A baixa pertinência é uma função *Z*, a média, por uma função de *Bell*, e a alta, por uma função *Sigma*.

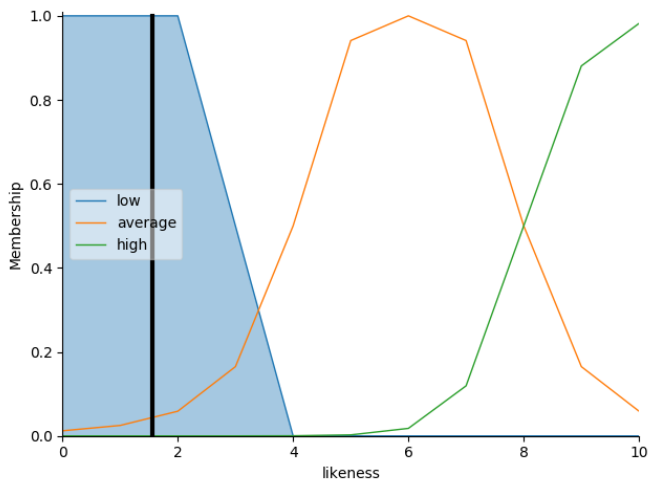


Figura 22. Resultado pertencimento da função de likeness - Waifu 1

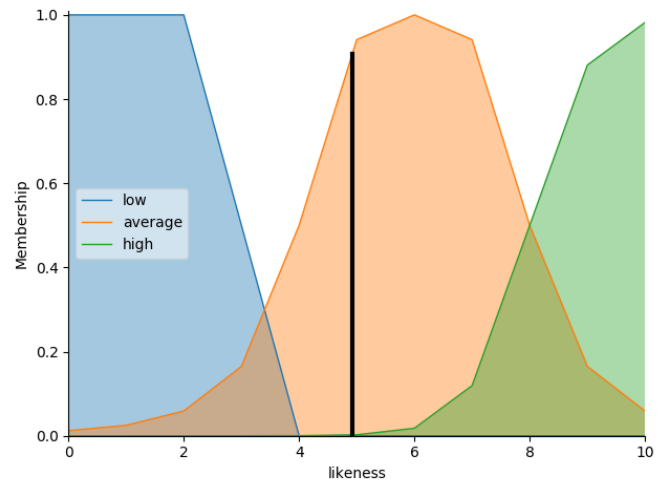


Figura 23. Resultado pertencimento da função de likeness - Waifu 2

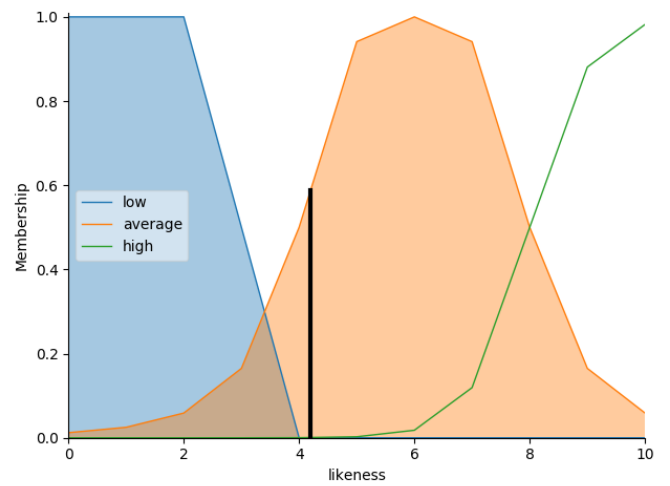


Figura 24. Resultado pertencimento da função de likeness - Waifu 3

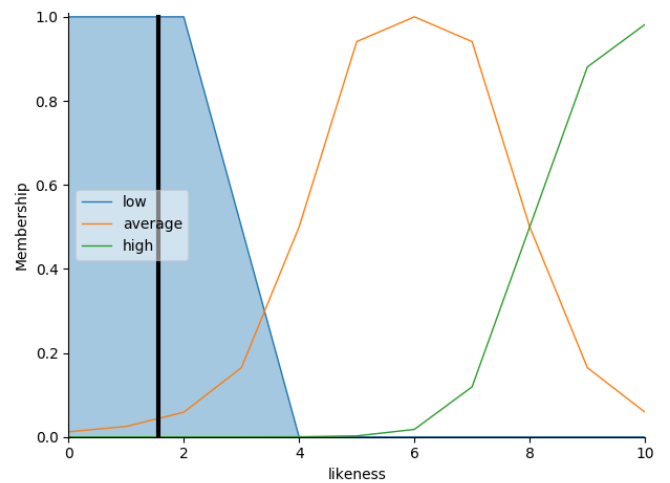


Figura 25. Resultado pertencimento da função de likeness - Waifu 4

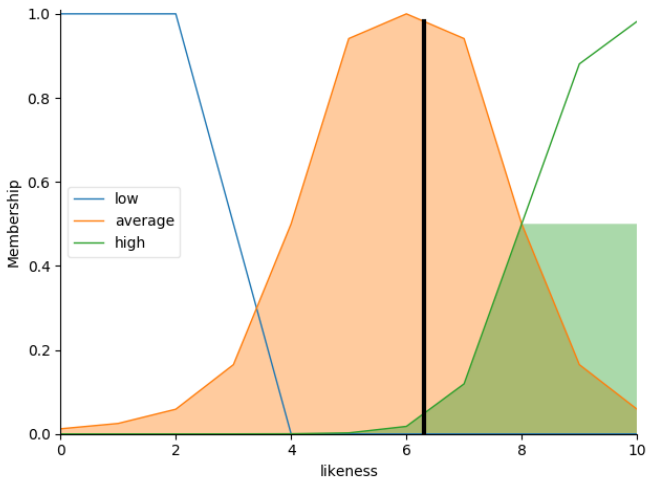


Figura 26. Resultado pertencimento da função de likeness - Waifu 5

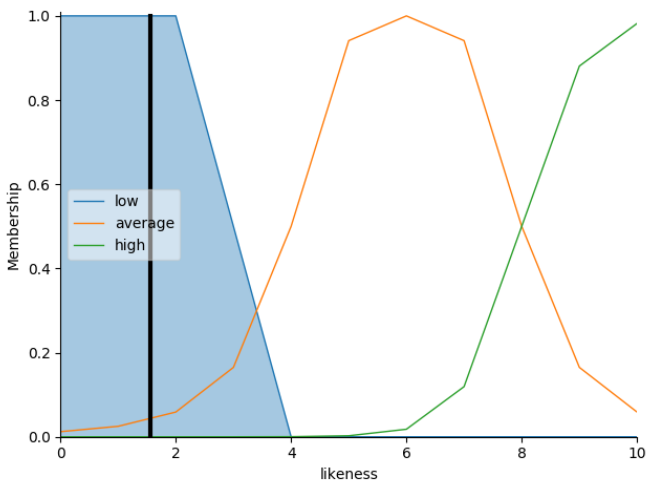


Figura 27. Resultado pertencimento da função de likeness - Waifu 6

IV. ANÁLISE

Começando por uma análise mais geral dos dados das *waifus* analisadas no processo, podemos alinhar nossas expectativas de, em um nível mais genérico das definições, quais seriam as *waifus* mais compatíveis com o esperado pelo perfil, dentre as avaliadas pela recomendação. Assim sendo, temos:

- *Waifu 1 (Figura 3) - Baixa relação:* Por possuir níveis elevados de *clumsy*, *short* e *kawaii* o algoritmo deve despriorizá-la.
- *Waifu 2 (Figura 4) - Média relação:* Neste caso, existem boas relações em certas características (*smart* e *busty*), porém ruins em outras (*cheerful*, *short*), logo o algoritmo deve ser mantê-la em médio nível de pertinência.
- *Waifu 3 (Figura 5) - Média relação:* Como o caso acima, se sai bem em alguns pontos (*flashy*, *lewd* e *beatiful*), entretando também possui divergências a intenção do usuário (*clumsy* e *plain*)

- *Waifu 4 (Figura 6) - Baixa relação:* Basicamente todos seus pontos predominantes são de baixa preferência do usuário, assim é esperado uma baixíssima pertinência.
- *Waifu 5 (Figura 7) - Alta relação:* Em contraste a quarta entrada, esta possui todas as características predominantes de acordo com as preferências do perfil, deste modo deverá ser a maior pertinência do conjunto.
- *Waifu 6 (Figura 8) - Baixa relação:* Possui muitas características em desacordo com o perfil, possivelmente será de prioridade.

Dito isso, podemos agora avaliar o resultado das implementações citadas, comparando-nas sempre ao nosso entendimento prévio descrito acima.

Com relação a primeira implementação, percebemos que esta é bem direta quanto a sua classificação, rebaixando bastante os valores de pertinência para os casos onde há baixa preferência (Figuras 9, 12, 14). Analisando por outro ângulo, aqui se sobressai um caso curioso para a figura 10, onde a mistura entre altas e baixas correlações leva o resultado ao seu valor médio absoluto, muito por conta de ser calculado pelo método do *centroid*, que a nivela ao centro do universo descrito. No outro caso de mistura para essa implementação (Figura 11) também fica clara essa relação, apontando mais para a baixa *likeness* pois seus pontos baixos acabam tendo um maior peso nessa implementação. Por último, o caso de melhor *likeness* tem seu valor mediado, apesar de aparecer na curva de alta pertinência, ao que aparenta ser também por conta da defuzzificação.

Por fim, essa solução transparece ser negativista em relação as características de pouca pertinência, sendo que nos casos analisados abaixa o resultado de *likeness*. Isso ocorre pois a função de *centroid* possui uma média ponderada que pode amplificar a importância de certas entradas, além da solução ser totalmente baseada em função triangulares, que são menos suaves em descrever resultados tão esparsos quantos os propostos por esse trabalho.

Em seguida, nos atentando a segunda implementação, podemos notar a influência de funções trapezoidais para nos resultados apresentados. As funções das *Waifus* 1, 4 e 6 (Figuras 16, 19 e 21 respectivamente) se mantêm como baixa pertinência, porém é perceptível a queda no valor defuzzificado. Já para as funções de média pertinência (Figuras 17 e 18) também acontece essa queda, destaque para a primeira delas, onde há perda para a alta *likeness*. Por último, a análise da *waifu* 5 (Figura 20) é interessante pois trás o contraponto onde ela ganhou um aumento significativo em sua pertinência, estando agora totalmente no grupo de alta *likeness*.

Todas essas mudanças podem ser explicadas pela nova forma trapezoidal das funções de relação entre perfil/características. Como seus intervalos se tornaram maiores, ambas qualidades de baixa e alta preferência ganharam pesos maiores simultaneamente, fazendo com que *centroids* que já puxavam o valor pra baixo continuassem o fazendo, enquanto aumentou o valor daqueles que possuíam altos pesos em características de alta preferência.

Logo em sequência temos a terceira e última implementação, que mistura 3 (três) funções diferentes para formar a *likeness*. Mantendo o padrão das implementações anteriores, os resultados para as *waifus* 1, 4 e 6 (Figuras 22, 25 e 27) continuam sendo a de baixa pertinência, mostrando um valor absoluto menor ainda para as respectivas *likenesses*. Em seguida, as funções que apresentam média pertinência (Figuras 23, 24 e 26) são altamente influenciadas pela escolha da função de *Bell* e tem valores defuzzificados diminuídos também. Ocorro pois esta função ocupa uma grande parcela do intervalo e tem seu pico em um ponto afastado do centro universal, desviando a resposta da recomendação para os pontos mais centrais.

Ou seja, a função de *Bell* neste caso age como um separador que impede o valor da *likeness* de crescer livremente, impondo uma barreira que mantém uma grande parte dos combinações em valores medianos. Esse comportamento pode ser bom para realmente separar os ótimos resultados, mas acaba englobando uma grande parcela de combinações em "mediano", o que talvez não seja o desejado em alguns sistemas. Em nosso caso particular, observamos isso nas combinações que entram na pertinência alta (Figuras 24 e 26), entretanto sofrem desta regulação para o centro.

Em vista de todos os resultados para as diferentes implementações, podemos inferir que existem certa continuidade na maneira como problema foi tratado. Foi observável que, independentemente das variações no algoritmo, as baixas equivalências entre perfil e *waifu* se mantém transparentes e padronizadas. A maior variação nesse aspecto foram justamente nas altas equivalências (*Waifu* 5 especificamente), onde cada implementação se comportou de uma maneira diferente e somente a segunda atingiu nossas expectativas iniciais (recomendá-la em alto nível de prioridade).

Ainda em nossa visão, todas as implementações foram assertivas em diferenciar os 3 (três) pólos de recomendação baseados na *likeness*, uma vez que todas elas possuem ao menos um gráfico em que é considerado certa função de pertencimento (Baixa, média ou alta). Mesmo que os resultados variem por causa da escolha de defuzzificação, consideramos que todas as implementações conseguem manifestar muito bem o significado das entradas inputadas.

V. CONCLUSÕES

Como pudemos verificar nos resultados, nosso sistema é capaz de representar e quantificar, ainda que com algumas limitações, as preferências de um perfil de usuário. Ele também abrange os 3 (três) possíveis polos de afinidade simplificados com que decidimos trabalhar.

Para a criação de um sistema de recomendação completo, precisaríamos ainda de uma forma de coletar e classificar os dados. Possivelmente de maneira colaborativa a fim de diluir o enviesamento da categorização dos mesmos. Para uma grande quantidade de itens e usuários, também deveríamos fazer uma filtragem das possíveis recomendações, formando grupos de usuários com preferências similares, a fim de aumentar as chances de acerto na predição, além da

qualidade dos *matches* e diminuir o número de amostras a serem processadas pelo recomendador.

+-----+

REFERÊNCIAS

- [1] "Waifu," English Wiktionary. [Online]. Available: <https://en.wiktionary.org/wiki/waifu> 1