

ÖZET

BULANIK MANTIK YÖNTEMLERİ KULLANILARAK GAZLI İÇECEKLERDE KARBONDİOKSİT MİKTARININ BASINÇ VE SICAKLIK DEĞERLERİNE GÖRE KONTROLÜ

Zekiye Beyza Kocamış - 20290267

Zeynep Heray - 20290258

Merve Başöz - 19290215

Bulanık Mantık günlük yaşantımızda kullandığımız, kavramlarını yaşantımızın birçok yerinde görmekte olduğumuz matematiksel bir disiplindir. Bu kavramlar yüksek, orta ve düşük değerleridir.

Araştırmalar Bulanık Mantık sistemleri kullanılarak elde edilen sonuçların, klasik yöntemlerle karşılaştırıldığında daha iyi sonuçlar verdiğini göstermiştir. Bununla birlikte bulanık mantık kavramı, kontrol sistemi ihtiyacı olarak ortaya çıkmış ve daha fazla teknolojiye tercih edilmeye başlanmıştır. Kontrol sistemleri tıp veya ekonomi gibi birçok alanda kullanılmaktadır.

Bu projede de, bulanık mantık sistemleri kullanılarak dünyadaki popüler içecekler arasında yükselmeye devam eden gazlı içeceklerin karbondioksit miktarı, sıcaklık ve basınç değerlerine bağlı olarak belirlenecektir.

Bu çalışmada **Mamdani çıkarım yöntemi** ve **Center of Area durulaştırma yöntemi** kullanılacaktır.

GİRİŞ

Günlük hayatta kullandığımız çoğu terim aslında bulanık bir yapıya sahiptir. Bir şeyi açıklarken, bir tanım yaparken ve daha birçok farklı durumda kullandığımız kelimeler bulanıklık içeren ifadelerdir. Örneğin; az, çok, biraz, fazla, sıcak, soğuk, genç, yaşlı, uzun, kısa, yavaş, hızlı gibi pek çok sözel terim bulanıklık içeren terimlerdir.

İnsanlar genel olarak bir olayı anlatırken kesinlik içeren sayısal ifadeler yerine bu tür kesinlik belirtmeyen terimler kullanırlar. Bir insanın yaş durumuna göre ona çocuk, genç, orta yaşlı veya yaşlı deriz. İçtiğimiz içeceğin sıcaklığına göre sıcak, ılık veya soğuk ifadelerini kullanırız. Güneşin durumuna göre havanın açık, bulutlu, güneşli olduğunu belirtiriz. Bütün bunlar insan beyninin belirsiz ve kesinlik içermeyen durumlarda nasıl davrandığına ve olayları nasıl değerlendirip, tanımlayıp, komut verdiğine dair birer örnektir.

Bulanık Mantığın temeli bulanık sistemlerin en temel elemanı olan bulanık kümeler sayesinde oluşmuştur. Bulanık kümeler ile ilgili ilk açıklama 1965 yılında Berkeley Üniversitesi öğretim üyesi olan Lotfi A. Zadeh tarafından ortaya atılmıştır. Yayımlanan bu makaleden sonra belirsizlik içeren sistemlerin incelenmesi yeni bir boyut kazanmıştır. Fakat makale 1965 yılında yayınlanmış olmasına rağmen bulanık küme kavramı ancak 1970'li yılların ikinci yarısından sonra kullanılmaya başlanmıştır.

Bulanık Mantıktaki kümelerin mantığı alışık olduğumuz klasik kümelerden oldukça farklıdır. Klasik küme yaklaşımında elemanlar bir kümeye ya aittir ya da değildir yani ya 0 ya da 1 değerlerini alabilirler. Oysa Bulanık Mantık yaklaşımında elemanların bir kümeye aitliği 0 ile 1 arasında değişebilir.

Bulanık Mantık sistemi 1980'li yıllarda Japonlar tarafından üretilen ürünlerde kullanılmaya başlanmıştır. Örneğin bulanık denetimli çamaşır makinesi; çamaşırın cinsine, miktarına, kirliliğine göre en etkili çamaşır

yıkama ve su kullanım programını seçebilmektir. Buna ek olarak elektrik süpürgesi, televizyon ve müzik aygıtlarında Bulanık Mantık denetim kullanılmaktadır. Bu örneklerden de görüldüğü gibi Bulanık Mantığın uygulama alanları oldukça geniştir ve bu uygulamalar ekonomik bir kontrol sağlamaya olanak tanımaktadır. Bu tür kullanımlardan sonra bulanık mantık sistemi hız kazanmıştır ve bu sayede günümüzde hemen hemen her alanda kullanılmaktadır.

Şimdi, Bulanık Mantık sistemlerinde kullanılan Bulanık Çıkarım sisteminden bahsedeceğiz.

Bulanık çıkarım, girdi vektöründeki değerleri yorumlayan ve bazı kural setlerine dayalı olarak çıktı vektörüne değerler atayan bir yöntemdir. Bulanık Çıkarım sistemi, Defuzzification interface, Fuzzification interface, Knowledge base ve Controlled system kısımlarından oluşur. Defuzzification interface, Fuzzification interface ve Knowledge base Bulanık Mantık sistemini oluşturan kısımlardır. Biz projemiz için Knowledge base ve Defuzzification'dan bahsedeceğiz.

Bulanık Mantık sistemlerinde, **Fuzzy Knowledge Base**, bulanık küme teorisine dayanan kuralların ve dilsel değişkenlerin gerçeklerini temsil eder, böylece bilgi tabanı sistemleri yaklaşık akıl yürütmeye izin verir.

Fuzzy Knowledge Base (FKB) sisteminin bilgi tabanı (knowledge base) database ve rule base olmak üzere iki bileşenden oluşur. FKB bulanık niteliklere ve kurallara (rules) sahip nesneler içerir. Bulanık sonuçlar çıkarılması için bir bulanık çıkarım yöntemi kullanılır.

Database'in Yapısı:

Bulanık sistemde hangi parametrelerin olacağı belirlenip parametrelere nasıl sorusu sorularak, parametrelerle ilgili bulanık kümeler bulunur. Bu bulanık kümelerin input parametresi ve output parametresi arasındaki ilişkiyi belirleyeceğimiz kural tabanı ile kural seti belirlenir.

Rule Base'in Yapısı:

Input ve output parametresi alınıp input ile alakalı bulanık kümeler belirlenir. Sonrasında, yapılan gözlemler ve deneyler neticesinde iki parametre arasındaki ilişki belirlenip bu ilişki üzerinden bir kural kümesi tayin edilir. Bu kural kümesi üzerinden de çıkarım yapılır.

En çok kullanılan kural IF-THEN kuralıdır.

Örneğin: *IF x is A THEN y is B*

Burada x ve y değişken, A ve B ise bulanık kümelerdir.

Fuzzy Rules, yeni öznitelikler türetmek veya farklı nesnelerin yalnızca kesin değil aynı zamanda bulanık özniteliklerini kullanarak bazı kısıtlamaları belirtmek için kullanılır. Kuralların değişkenleri, nesnelerin veya nesnelerin kendilerinin niteliklerini temsil eder.

Defuzzification, yani durulaştırma kümelenmiş bulanık kümenin çıktısından tek bir sayı elde etme işlemidir. Diğer bir deyişle, durulaştırma, bulanık bir kümeye dayalı olarak en iyi kesin değeri seçen bir karar verme algoritması ile gerçekleştirilir. Maksimum Ortalama (Mean of Maximum), Ağırlık Merkezi (Center of Area) ve Merkez Ortalama (Bisector of Area) yöntemleri dahil olmak üzere çeşitli durulaştırma biçimleri vardır.

Mean of Maksimum: En büyük üyelik değerini veren noktaların ortalamasını bulur.

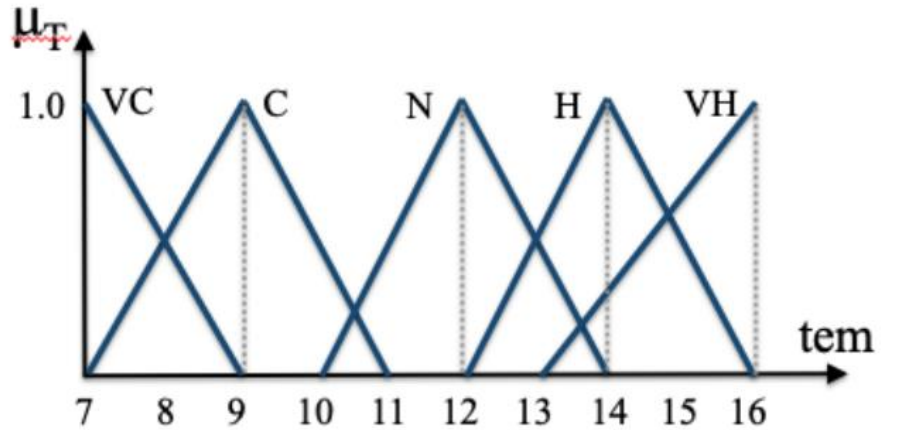
Center of Area: Üyelik değerlerinin ağırlıklı ortalamasını bulur. Başka bir deyişle ilgili alanın merkezi bulunur.

Bisector of Area: İlgili alanın tam olarak ikiye bölünebilmesi için gereken x değeri bulunur.

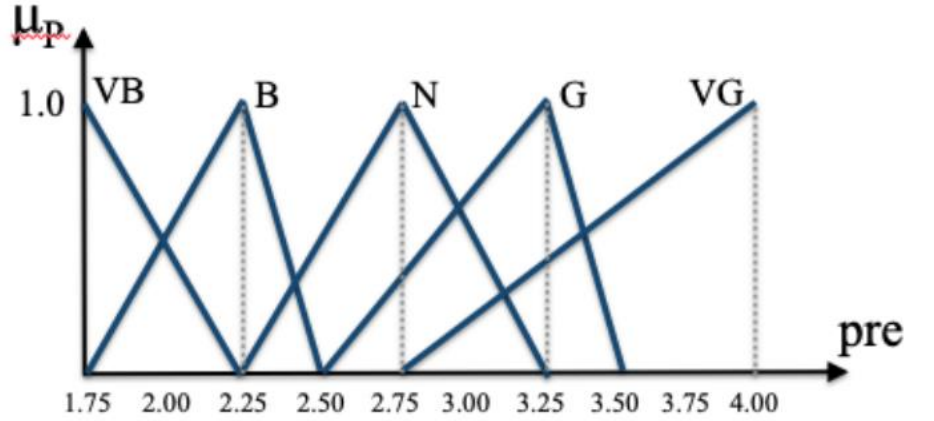
Bizim çalışmamızda Bulanık Mantık kullanılarak gazlı içeceklerdeki karbondioksit kontrolü yapılacaktır. Tüm dünyada tüketimi oldukça fazla olan gazlı içecekler bazı önemli özellikler taşımaktadır. Gazlı içecek üretiminde karbondioksitin kontrolü en önemli süreçlerden biridir.

Bu kontrol sisteminde gazlı içeceğin sıcaklık ve basınç değerleri parametre olarak alınacaktır. Bu iki değere göre bir tane çıkış değeri elde edilecektir ve bu da kullanılması gereken karbondioksit yüzdesinin ne olduğudur. Çalışmamızda **Mamdani çıkarım yöntemini** karbondioksit miktarı belirlemek, **Center of Area durulaştırma yöntemini** ise Mamdani çıkarımı ile elde edilen şeklin alanının ağırlık merkezini bulmak için kullanacağız.

1. Sıcaklık değeri:



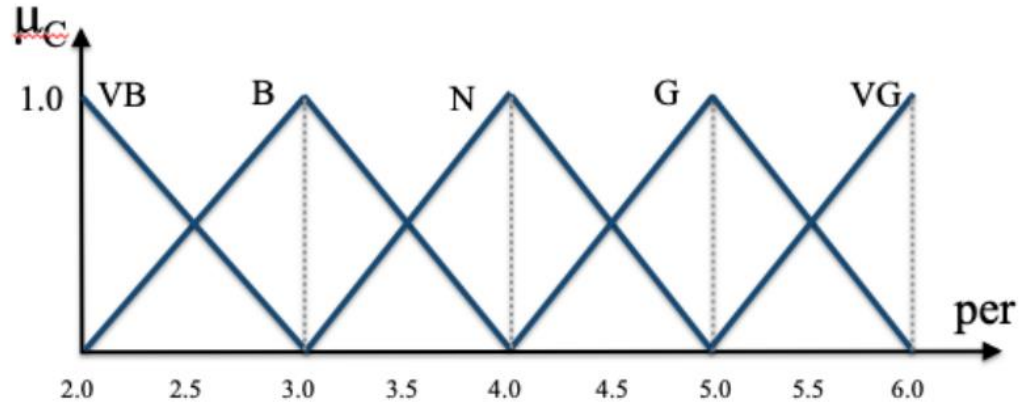
2. Basınç değeri:



Mamdani çıkarım yöntemi ile karbondioksit miktarını belirlemek için bir sıcaklık ve bir de basınç değeri alınacak. Alınan bu değerlerin, belirtilen sıcaklık ve basınç grafiklerinde karşılık buldukları üyelik değerleri belirlenecek. Mamdani çıkarım yöntemi gereğince, bulunan üyelik değerlerinden küçük olanı alınıp bu değer karbondioksit yüzdesi belirlemede kullanılan grafiğe yansıtılacak ve değere karşılık gelen üyelik değeri bulunacak. Verilen kural matrisine göre, bulduğumuz üyelik değeri işaretlememiz gereken değer aralığına yansıtılacak. En son olarak ise grafiğe yansıtılan bir ya da birden fazla üyelik değerinden büyük olan grafik alınacak.

Center of Area durulaştırma yönteminin kullanımı için ise bahsettiğimiz bu grafiğin merkezi, bir başka deyişle üyelik değerlerinin ağırlıklı ortalaması bulunacak.

3. Karbondioksit yüzdesi:



YAPI

Yazacağımız bulanık mantık sisteminde giriş parametrelerimiz sıcaklık ve basınç çıkış parametremiz ise karbondioksit yüzdesidir. Bütün parametrelerimiz için beşer adet değer verilmiştir

Sıcaklık: Verycold, Cold, Normal, Hot, Veryhot

Basınç: Verycad, Bad, Normal, Good, Verygood

Karbondioksit yüzdesi: Verybad, Bad, Normal, Good, Verygood

Bu değerler için ise 25 adet kural belirlenmiştir.

Kural 1 : Sıcaklık Verycold ve basınç Verybad ise karbondioksit yüzdesi Normal'dir.

Kural 2 : Sıcaklık Verycold ve basınç Bad ise karbondioksit yüzdesi Normal'dir.

Kural 3: Sıcaklık Verycold ve basınç Normal ise karbondioksit yüzdesi Good'dur.

Kural 4 : Sıcaklık Verycold ve basınç Good ise karbondioksit yüzdesi Verygood'dur.

Kural 5 : Sıcaklık Verycold ve basınç Verygood ise karbondioksit yüzdesi Verygood'dur.

Kural 6 : Sıcaklık Cold ve basınç Verybad ise karbondioksit yüzdesi Bad'dir.

Kural 7 : Sıcaklık Cold ve basınç Bad ise karbondioksit yüzdesi Good'dur.

Kural 8 : Sıcaklık Cold ve basınç Normal ise karbondioksit yüzdesi Good'dur.

Kural 9 : Sıcaklık Cold ve basınç Good ise karbondioksit yüzdesi Good'dur.

Kural 10 : Sıcaklık Cold ve basınç Verygood ise karbondioksit yüzdesi Verygood'dur.

Kural 11 : Sıcaklık Normal ve basınç Verybad ise karbondioksit yüzdesi Bad'dir.

Kural 12 : Sıcaklık Normal ve basınç Bad ise karbondioksit yüzdesi Normal'dir.

Kural 13 : Sıcaklık Normal ve basınç Normal ise karbondioksit yüzdesi Normal'dir.

Kural 14 : Sıcaklık Normal ve basınç Good ise karbondioksit yüzdesi Good'dur.

Kural 15 : Sıcaklık Normal ve basınç Verygood ise karbondioksit yüzdesi Verygood'dur.

Kural 16 : Sıcaklık Hot ve basınç Verybad ise karbondioksit yüzdesi Bad'dir.

Kural 17 : Sıcaklık Hot ve basınç Bad ise karbondioksit yüzdesi Bad'dir.

Kural 18 : Sıcaklık Hot ve basınç Normal ise karbondioksit yüzdesi Normal'dir.

Kural 19 : Sıcaklık Hot ve basınç Good ise karbondioksit yüzdesi Normal'dir.

Kural 20 : Sıcaklık Hot ve basınç Verygood ise karbondioksit yüzdesi Good'dur.

Kural 21 : Sıcaklık Veryhot ve basınç Verybad ise karbondioksit yüzdesi Verybad'dir.

Kural 22 : Sıcaklık Veryhot ve basınç Bad ise karbondioksit yüzdesi Bad'dir.

Kural 23 : Sıcaklık Veryhot ve basınç Normal ise karbondioksit yüzdesi Normal'dir.

Kural 24 : Sıcaklık Veryhot ve basınç Good ise karbondioksit yüzdesi Normal'dir.

Kural 25 : Sıcaklık Veryhot ve basınç Verygood ise karbondioksit yüzdesi Good'dur.

Programımızı yazarken Python dilini kullanıyoruz. Öncelikle kullanacağımız kütüphaneleri ekliyoruz.

```
import numpy as np
import skfuzzy as fuzz
import skfuzzy.membership as mf
```

Numpy, Scikit-Fuzzy kütüphanelerini ve üyelik fonksiyonları için 'membership' sınıfını ekliyoruz.

```
s_temp = np.arange(7, 17, 1)
s_pre = np.arange(1.75, 4.25, 0.25)
s_carb = np.arange(2.0, 6.5, 0.5)
```

Sıcaklık, basınç ve karbondioksit yüzdeleri için Numpy kütüphanesini arange metodu ile tanım aralıklarını belirliyoruz.

```
temp_VC = mf.trimf(s_temp, [7, 7, 9])
temp_C = mf.trimf(s_temp, [7, 9, 11])
temp_N = mf.trimf(s_temp, [10, 12, 14])
temp_H = mf.trimf(s_temp, [12, 14, 16])
temp_VH = mf.trimf(s_temp, [13, 16, 16])

pre_VB = mf.trimf(s_pre, [1.75, 1.75, 2.25])
pre_B = mf.trimf(s_pre, [1.75, 2.25, 2.50])
pre_N = mf.trimf(s_pre, [2.25, 2.75, 3.25])
pre_G = mf.trimf(s_pre, [2.50, 3.25, 3.50])
pre_VG = mf.trimf(s_pre, [2.75, 4.00, 4.00])

carb_VB = mf.trimf(s_carb, [2.0, 2.0, 3.0])
carb_B = mf.trimf(s_carb, [2.0, 3.0, 4.0])
carb_N = mf.trimf(s_carb, [3.0, 4.0, 5.0])
carb_G = mf.trimf(s_carb, [4.0, 5.0, 6.0])
carb_VG = mf.trimf(s_carb, [5.0, 6.0, 6.0])
```

Parametrelerimizin değerlerinin her biri için 'trimf' metodu ile üçgensel üyelik değeri oluşturuyoruz. 'trimf' metodunun ilk parametresi değişkenin tanım aralığı; ikinci parametre ise üçgensel sayının başlangıç, orta ve bitiş noktalarının gösteren bir listedir.

```
input_temp = 12
input_pre = 2.00
```

Girdi değerlerimizi belirliyoruz.

```
temp_fit_VC = fuzz.interp_membership(s_temp, temp_VC, input_temp)
temp_fit_C = fuzz.interp_membership(s_temp, temp_C, input_temp)
temp_fit_N = fuzz.interp_membership(s_temp, temp_N, input_temp)
temp_fit_H = fuzz.interp_membership(s_temp, temp_H, input_temp)
temp_fit_VH = fuzz.interp_membership(s_temp, temp_VH, input_temp)

pre_fit_VB = fuzz.interp_membership(s_pre, pre_VB, input_pre)
pre_fit_B = fuzz.interp_membership(s_pre, pre_B, input_pre)
pre_fit_N = fuzz.interp_membership(s_pre, pre_N, input_pre)
pre_fit_G = fuzz.interp_membership(s_pre, pre_G, input_pre)
pre_fit_VG = fuzz.interp_membership(s_pre, pre_VG, input_pre)
```

Belirlediğimiz girdi değeri için 'interp_membership' metodu ile üyelik değeri hesaplıyoruz.

'interp_membership' metodunun ilk parametresi değişken için tanım aralığını, ikinci parametresi ilgili bulanık kümenin tanım aralığını üçüncü parametre ise girdi değerini belirtir.

```
rule1 = np.fmin(np.fmin(temp_fit_VC, pre_fit_VB), carb_N)
rule2 = np.fmin(np.fmin(temp_fit_VC, pre_fit_B), carb_N)
rule3 = np.fmin(np.fmin(temp_fit_VC, pre_fit_N), carb_G)
rule4 = np.fmin(np.fmin(temp_fit_VC, pre_fit_G), carb_VG)
rule5 = np.fmin(np.fmin(temp_fit_VC, pre_fit_VG), carb_VG)

rule6 = np.fmin(np.fmin(temp_fit_C, pre_fit_VB), carb_B)
rule7 = np.fmin(np.fmin(temp_fit_C, pre_fit_B), carb_G)
rule8 = np.fmin(np.fmin(temp_fit_C, pre_fit_N), carb_G)
rule9 = np.fmin(np.fmin(temp_fit_C, pre_fit_G), carb_G)
rule10 = np.fmin(np.fmin(temp_fit_C, pre_fit_VG), carb_VG)

rule11 = np.fmin(np.fmin(temp_fit_N, pre_fit_VB), carb_B)
rule12 = np.fmin(np.fmin(temp_fit_N, pre_fit_B), carb_N)
rule13 = np.fmin(np.fmin(temp_fit_N, pre_fit_N), carb_N)
rule14 = np.fmin(np.fmin(temp_fit_N, pre_fit_G), carb_G)
rule15 = np.fmin(np.fmin(temp_fit_N, pre_fit_VG), carb_VG)

rule16 = np.fmin(np.fmin(temp_fit_H, pre_fit_VB), carb_B)
rule17 = np.fmin(np.fmin(temp_fit_H, pre_fit_B), carb_B)
rule18 = np.fmin(np.fmin(temp_fit_H, pre_fit_N), carb_N)
rule19 = np.fmin(np.fmin(temp_fit_H, pre_fit_G), carb_N)
rule20 = np.fmin(np.fmin(temp_fit_H, pre_fit_VG), carb_G)

rule21 = np.fmin(np.fmin(temp_fit_VH, pre_fit_VB), carb_VB)
rule22 = np.fmin(np.fmin(temp_fit_VH, pre_fit_B), carb_B)
rule23 = np.fmin(np.fmin(temp_fit_VH, pre_fit_N), carb_N)
rule24 = np.fmin(np.fmin(temp_fit_VH, pre_fit_G), carb_N)
rule25 = np.fmin(np.fmin(temp_fit_VH, pre_fit_VG), carb_G)
```

Kurallarımızda ve bağlacı kullandığımız için Numpy kütüphanesine ait 'fmin' metodu ile kurallarımızın her birini yazıyoruz. 'fmin' metodu listelerde elemanlar arası karşılaştırma yaparak minimum elemanları bulur.

```

out_VB = rule21
out_B = np.fmax(rule6, np.fmax(np.fmax(rule11, rule16), np.fmax(rule17, rule22)))
out_N = np.fmax(np.fmax(np.fmax(rule1, rule2), np.fmax(rule12, rule13)), np.fmax(np.fmax(rule18, rule19),
np.fmax(rule23, rule24)))
out_G = np.fmax(rule3, np.fmax(np.fmax(rule7, rule8), np.fmax(np.fmax(rule9, rule14), np.fmax(rule20, rule25))))
out_VG = np.fmax(np.fmax(rule4, rule5), np.fmax(rule10, rule15))

```

Girdiye göre hesaplanan kurallarda aynı çıktıyı veren kuralların Numpy kütüphanesine ait 'fmax' metodu ile birleşim kümelerini hesaplıyoruz. 'fmax' metodu listelerde elemanlar arası karşılaştırma yaparak maximum elemanları bulur.

```

out_carb = np.fmax(out_VB, np.fmax(np.fmax(out_B, out_N), np.fmax(out_G, out_VG)))
defuzzied = fuzz.defuzz(s_carb, out_carb, 'centroid')
print(defuzzied)

```

Çıkış kümesini(out_carb) yine 'fmax' metodu ile hesaplıyoruz ve durulaştırma fonksiyonuna gönderiyoruz. 'defuzz' metodu durulaştırma fonksiyonudur. Birinci parametresi çıkış değişkeninin değer aralığını, ikinci parametresi girdi değeriyle elde edilen alanı ve üçüncü parametre ise durulaştırma yaparken kullandığımız yöntemi belirtir. 'centroid' yöntemi ağırlık merkezi yöntemidir.

UYGULAMA

Girdi 1: sıcaklık = 8, basınç = 2.75

```
5.083333333333333
```

Girdi 2: sıcaklık = 12, basınç = 2

```
3.5
```

Girdi 3: sıcaklık = 10.5, basınç = 3.25

```
5.119047619047619
```

KAYNAKÇA

<https://iksadyayinevi.com/wp-content/uploads/2020/02/BULANIK-MANTIK-Y%C3%96NTEM%C4%B0-VE-UYGULAMALARI.pdf>

https://acikbilim.yok.gov.tr/bitstream/handle/20.500.12812/47801/yokAcikBilim_407690.pdf?sequence=-1

https://www.researchgate.net/publication/3336292_A_fuzzy_knowledge-based_system_for_intelligent_retrieval

<https://www.sciencedirect.com/topics/engineering/fuzzy-inference>

<https://www.sciencedirect.com/topics/engineering/defuzzification#:~:text=Defuzzification%20is%20the%20process%20of,based%20on%20a%20fuzzy%20set.>

<https://pythonhosted.org/scikit-fuzzy/>

<https://numpy.org/>