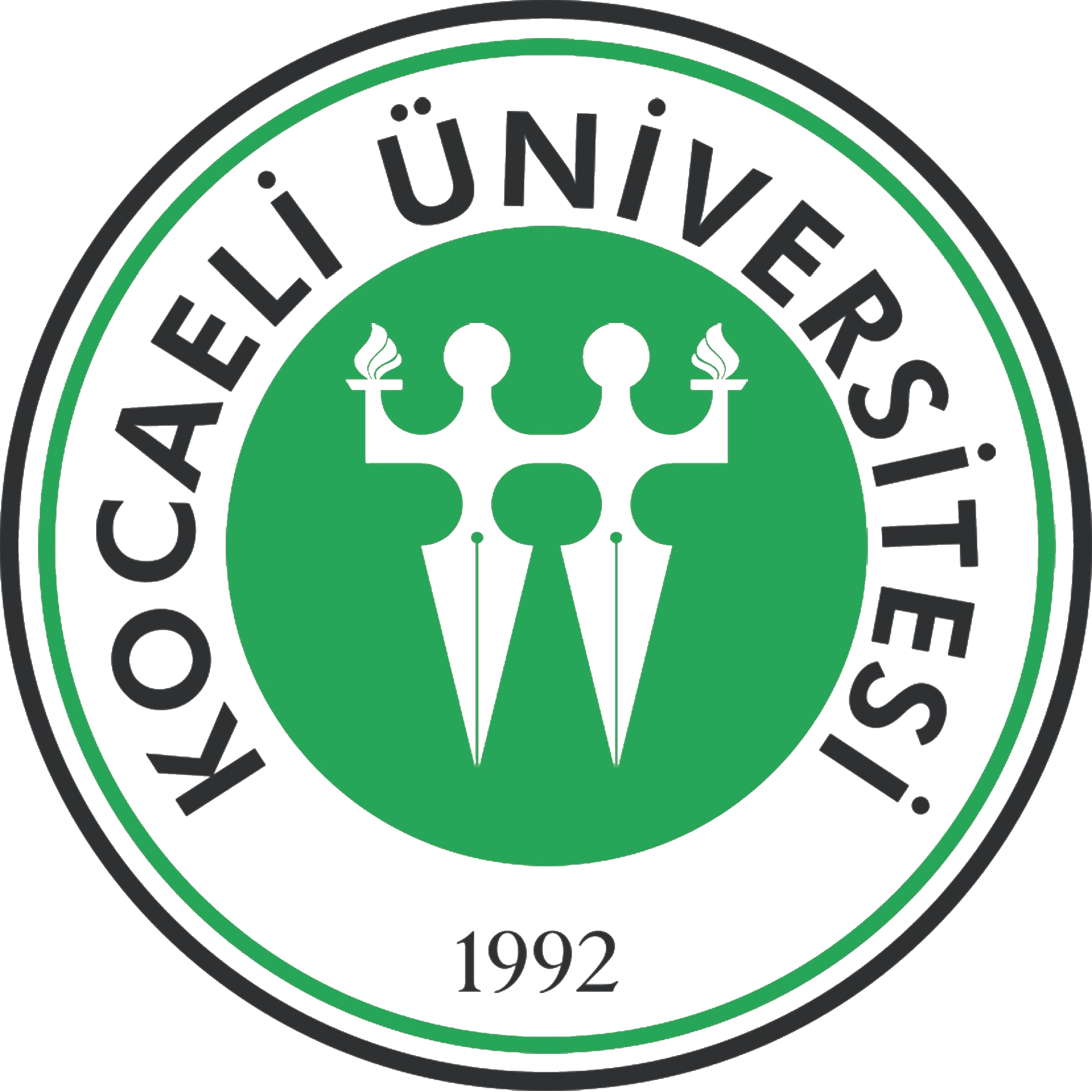
**BULANIK MANTIK YAKLAŞIMI İLE MEYVE SUYU KALİTE KONTROL UYGULAMASI**

**Bu çalışma, Kocaeli Üniversitesi Bilgisayar Mühendisliği öğrencilerinden Pınar Sarı (170202049),Veli Kaan Çetinel (170202081),Alper Aktepe (160202056) ,Hasan Basri Ersan (170202077) ve Yunus Durankuş (170202117) tarafından yapılmıştır.**

****

Kocaeli Üniversitesi,Bilgisayar Mühendisliği

**Özet**

Bu projede giriş olarak aldığımız; meyve suyu, normal su ve şeker değerlerine göre, meyve suyunun kalite oranı ölçülmektedir. Bu ölçüm bulanık mantık kuralları ile modellenerek gerçekleştirilmektedir.

# 1. Giriş

Yapmış olduğumuz “Bulanık Mantık Yaklaşımı ile Meyve Suyu Kalite Kontrolü Uygulaması” projesinde, Scikit-Fuzzy kütüphanesi kullanarak bulanık mantık tasarımını Python programlama dilinde tasarladık..Tasarım sonucunda, girilen değerlere göre meyve suyunun kalite değeri hakkında bize bilgi döndüren bir uygulama gerçekleştirdik.

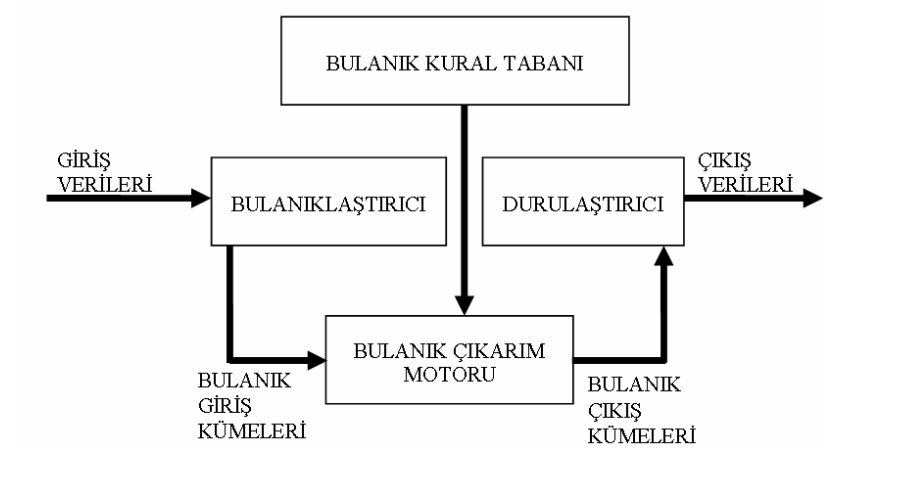
# 2.Temel Bilgiler

Projeyi gerçekleştirirken 2 farklı bilgisayar üzerinde 2 farklı ide ve python versiyonunda testler yapıldı.Projede NumPy,Scikit-Fuzzy ve Matplotlib kütüphanelerini kullandık.Matematiksel işlemler için NumPy,bulanık mantık işlemleri için Scikit-Fuzzy ve son olarak grafik ve görselleştirme için ise Matplotlib kütüphanesini kullandık. İlki Visual Studio Code ve python 2.7, diğeri ise Spyder(Anaconda) ve Python 3.7 dir.

**2.1** Bulanık Mantık nedir?

Bulanık mantık ilkeleri klasik mantık ilkeleri gibi çalışmaz. Bilgisayarlarda görülen ikili çalışma sistemi gibi değildir. Sorulan bir soru karşısında net bir cevap beklemez. Bulanık mantık ilkelerinde tamamen doğru ya da tamamen yanlış değerler yoktur. Bunun yerine yanlışa veya doğruya yakın kısmen doğru, kısmen yanlış gibi kesin olmayan belirsiz seçenekler de işin içine dahil edilir ve belli kurallar çerçevesinde derecelendirilir.

Bulanık mantık uygulamaları ile klasik mantık ilkelerinde bulunan ve kesinlik bildiren alternatifler esnetilir. Böylece daha belirsiz durumları da kapsayan modeller aracılığıyla alternatifler çoğaltılır ve değişkenler belli kurallar ışığında derecelendirilir. Kesin olmayan, birbirleriyle çelişkili ve muğlak gözüken değişkenlere ilişkin karmaşık ve çok boyutlu problemlerin çözümü ve yönetilebilir hale gelmesi hedeflenir.[1]



Şekil1: Bulanıklaştırma-durulaştırma birimli bulanık sistem [2]

Bir bulanık sistemi Şekil 1’de görüldüğü gibi giriş verileri, bulanıklaştırıcı, bulanık kural tabanı, bulanık çıkarım motoru, durulaştırıcı ve çıktı birimlerinden oluşmaktadır. Giriş verileri, sisteme verilecek olan girdileri ve bunlar hakkındaki tüm bilgileri içerir. Bu bilgiler sayısal veya sözel olabilir. Bulanıklaştırıcı, sayısal girdi verilerini sözel olarak tanımlanmış bulanık kümelerdeki üyelik derecelerine atayan birimdir.

Bulanık Kural Tabanı, veri tabanındaki girişleri çıkış değişkenlerine bağlayan mantıksal EĞER–İSE türünde yazılabilen kuralların içeren birimdir. Bu kuralların yazılmasında sadece girdi verileri ile çıktılar arasında olabilecek tüm ara (bulanık küme) bağlantıları düşünülür. Böylece, her bir kural girdi uzayının bir parçasını çıktı uzayına mantıksal olarak bağlar. Bulanık Çıkarım Motoru, bulanık kural tabanında giriş ve çıkış bulanık kümeleri arasında kurulmuş olan parça ilişkilerin hepsini bir arada toplayarak sistemin bir çıkışlı davranmasını temin eden işlemler topluluğunu içeren bir mekanizmadır. Bu motor her bir kuralın çıkarımlarını bir araya toplayarak tüm sistemin girdiler altında nasıl bir çıktı vereceğinin belirlenmesine yarar. Durulaştırıcı, bulanık işlemler sonucu elde edilen bulanık çıkarım sonuçlarını keskin sayısal çıkış değerlerine dönüştürür. Çıktı Birimi, bilgi ve bulanık kural tabanlarının bulanık çıkarım motoru vasıtası ile etkileşimi sonunda elde edilen çıktı değerlerinin topluluğunu belirtir.

**2.1.1 Bulanık Mantıkta Veri Tabanı Nedir**?

Veritabanları çok büyük boyuttaki veriyi saklayabilen ve gerektiğinde bunu kullanıcının hizmetine sunabilen veri depolama kaynaklarıdır. Bu kaynaklar çeşitli yazılımlar aracılığıyla yönetilirler. Yeryüzünde üretilen veri ve bilginin çok büyük bir kısmı veritabanlarında tutulur. Bu nedenle veritabanlarını daha etkin kullanma üzerine yapılan çalışmalar günden güne artmaktadır.Bulanık mantık uygulamalarında ise veri tabanı bulanık sisteme ait üyelik fonksiyonlarının tutulduğu alandır.

**2.1.2 Kural Tabanı Nedir**?

Bulanık kümeler arasındaki ilişkiler burada tutulur. Gelecek verilere göre hangi çıkarımların yapılacağıyla ilgili kurallar yine buradadır. Bu kısmı, bulanık sistemin anayasası gibi düşünebiliriz.

**2.1.3 Bulanıklaştırma Nedir**?

Sisteme yalın haliyle alınmış değerler, üyelik fonksiyonu kullanılarak, bulanık değerlere dönüştürülme işlemidir. Yani her bir giriş değerinin, bulanık kümeye/kümelere olan üyelik derecesi hesaplanır diyebiliriz.Bu işlemi gerçekleştiren birime bulanıklaştırıcı denir.

**2.1.4 Çıkarım Sistemi Nedir**?

Bu kısım, bilgi tabanıyla ortak çalışarak, kendisine gelen bulanık değerlerden sonuçlar çıkarmaya çalışır. Bu sonuçların neye göre ve nasıl çıkarılacağının bilgisi (ismiyle müsemma) kural tabanında tutulmaktadır.

**2.1.5 Durulaştırma Nedir**?

Çıkarım yapılmış veriler bu işleme kadar bulanık değer aralığında gelmektedir. Oysa bizim ihtiyacımız olan çıkış verilerinin bambaşka bir aralıkta olması gerekebilir. Durulaştırma ; gelen bulanık değerlerin, istediğimiz bir aralığa göre ölçeklenmesini sağlar.Bu işlemi gerçekleştiren birime durulaştırıcı denir.

**2.2** Scikit-Fuzzy kütüphanesi nedir?

Scikit-Fuzzy, bilimsel ve mühendislik temelli hesaplamalar için kullanışlı araçlar üreten **SciPy** topluluğu tarafından geliştirilen, Python’da yazılmış açık kaynak bir bulanık mantık kütüphanesidir. Bulanık mantıkla ilgili temel seviyedeki çoğu özelliği barındırmasının yanı sıra; bulanık kümeleme gibi ileri seviye makine öğrenmesi konuları için de kullanışlı araçlara sahiptir. Tamamen ücretsiz ve açık kaynak olmasından dolayı, MATLAB/Fuzzy Logic Toolbox gibi ücretli yazılımlar için iyi bir alternatiftir.

**2.3** Visual Studio Code nedir?

Visual Studio Code Microsoft, Linux ve Mac işletim sistemlerinde çalışabilen bir araçtır. Kurulum yapıldığında basit bir metin editörü görüntüsü ile karşımıza çıkan ürün, eklentiler sayesinde Node JS, Ruby, Python, C/C++, C#, Javascript gibi birçok programlama dilini desteklemektedir. Yani klasik Visual Studio ürünleri gibi her şeyi bünyesinde barındırmak yerine çekirdek bir yapı ile başlatılıp lazım olan parçaların eklentiler halinde ürüne eklenmesi sağlanarak farklılık kazanmıştır. Eklentilerin indirilebildiği bir market ortamı oluşturulmuştur. Ek olarak Visual Studio Code, hata ayıklama özellikleriyle, gelişmiş web ve bulut uygulamaları üstünde kodları düzenlemeye, yeniden tanımlamaya ve optimize etmeye yarar.

Kısacası Visual Studio Code. Linux, Mac OS ve Windows işletim sistemlerinde çalışan yaklaşık 30 programlama dili desteği sunan IDE ortamıdır.

**2.4** Spyder Anaconda nedir?

Anaconda, veri bilimi ve benzeri bilimsel uygulamalar için python kullanmak isteyenlere hazırlanmış tümleşik bir python dağıtımıdır. Veri bilimi, yapay zeka vb konularında sıkça kullanılan kütüphanelerin yanı sıra jupiter notebook ve spyder gibi araçları da barındırır. Programı [anaconda.org](http://anaconda.org/) adresinden kendi işletim sistemimize uygun versiyonu indirerek kurduk. Anacondayı kurduğumuzda sistemimizde python, jupiter notebook ve spyder da kurulmuş oldu.

**2.5** NumPy Kütüphanesi nedir?

**NumPy** (Numerical Python) bilimsel hesaplamaları hızlı bir şekilde yapmamızı sağlayan bir matematik kütüphanesidir. Numpy’ın temelini numpy dizileri oluşturur. Numpy dizileri python listelerine benzer fakat hız ve işlevsellik açısından python listelerinden daha kullanışlıdır.Ayrıca python listelerinden farklı olarak Numpy dizileri homojen yapıda olmalıdır yani dizi içindeki tüm elemanlar aynı veri tipinden olmalıdır.

**2.6** Matplotlib Kütüphanesi Nedir?

Matplotlib; veri görselleştirme işleminde kullandığımız temel python kütüphanesidir. 2 ve 3 boyutlu çizimler yapmamızı sağlar. Matplotlib genelde 2 boyutlu çizimlerde kullanılırken, 3 boyutlu çizimlerde başka kütüphanelerden yararlanılır.

Kütüphane içerisinde kullandığımız bazı fonksiyonlar ve işlevleri :

**plt.plot():** Grafiğin ana komutunu oluşturur. Bu komutun içine ilk önce x ekseni, sonrasında y ekseni yazılır. Grafikteki temel bazı değişkenleri değiştirmek istersek ilgili kodu bu fonksiyonun içerisine yazmamız gerekir. Örneğin grafiğin rengi, çizgi kalınlığı, aralıklar vs.

**figsize:**Grafiğin boyutunu belirler.

Yaygın olarak plt.figure(figsize=(8,6)) şeklinde kullanılır.Farklı kullanımları da mevcuttur. Parantez içindeki ilk yazılan değer grafiğin eni, ikincisi boyudur.

**plt.title():** Grafiğin başlığını oluşturur.

# 3.Geliştirilen Mimari

**3.1** Scikit-Fuzzy’nin çalışabilmesi için, aşağıdaki modüllerin ilgili versiyonlarını veya daha güncel olanlarını yükledik:

* NumPy >= 1.6
* SciPy >= 0.9
* NetworkX >= 1.9

**3.1.1** Pip(PIP install packages) üzerinden kurulum:

Spyder editöründe çalıştırabilmek için işletim sistemimizin terminalinde ve Anaconda Prompt arayüzünde sırasıyla aşağıdaki komutlar çalıştırıldı.

pip install -U scikit-fuzzy ve pip3 install -U scikit-fuzzy

**3.1.2**  Kaynak dosyası üzerinden kurulum:

Kaynak dosyaları GitHub üzerinden indirdikten sonra dosyaları arşivden çıkarıldı. Terminalden çıkarılan dosyanın dizinine geçiş yapıldı ve aşağıdaki komut çalıştırıldı:

python setup.py install

Kurulumlardan sonra, editör üzerinden aşağıdaki gibi scikit-Fuzzy çağrıldı:

import skfuzzy

Böylelikle bulanık mantık metodlarını kullanabilmemiz için gerekli kütüphaneyi, yükleme işlemini gerçekleştirmiş olduk.

**3.2** Tasarım:

Giriş ve çıkış değerleri aşağıdaki gibi tanımlanmıştır:

* Girişler: Meyve suyu(%), normal su (%), şeker(%)
* Çıkış: Meyve suyu kalitesi(%)

Giriş ve çıkış değerlerini dilsel olarak ifade edecek olursak:

* Meyve suyu: Az, orta, çok
* Normal su: Az, orta, çok
* Şeker: Az, orta, çok
* Çıkış: Az kaliteli, orta kalite, çok kaliteli

**3.3** Kural Tanım Aralıklarının Belirlenmesi

Belirlediğimiz konu üzerinde Bulanık Mantık modellemesini oluşturmamız için öncelikli olarak gerçek veriler üzerinde araştırmalar yapıldı. Bu araştırmalar sırasında bazı sabitler seçildi. Bu sabitler her ürünün 100 ml olması ve ürün türünün(şeftali, elma, portakal) aynı olmasına dikkat edildi. Araştırmamızda, her markada mutlaka olduğunu bildiğimiz ve epey veriye ulaşabileceğimizi düşündüğümüz, meyve olarak şeftaliyi tercih ettik ve markaların şeftaliyi tercih ettiği meyve suyu ürünleri üzerinde yoğunlaştık.

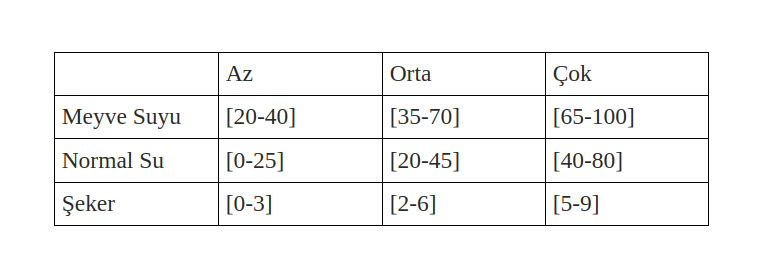
**3.3.1** Tanım aralığı belirlenirken referans aldığımız marka ürünleri:

* Cappy
* Pınar
* Tamek
* Dimes
* Meysu



Şekil 2: Cappy şeftali meyve suyu içerik örneği[3]

Bu markaların şeftali meyve sularının besin değerlerini açıkça paylaştığı internet sitelerinden incelediğimiz veriler sonucunda; meyve suyu, normal su ve şeker değerlerine ulaştık. (Ürünlerin içinde bulunan diğer katkı maddeleri ihlal edilmiştir.) Bu değerler sonucunda aşağıda görünen tabloda değer aralıkları belirlenmiştir.



Şekil 3: Kural tanım aralıkları.

**3.4**  Kurallar

3 Giriş için (meyve suyu, normal su ve şeker) oluşturulmuş 27 kural vardır. Bu kurallar aşağıda listelenmiştir:

KURAL 1:EĞER Meyve Suyu Az ve Su Az ve Şeker Az İSE ,O HALDE Kalite Azdır.

KURAL 2:EĞER Meyve Suyu Az ve Su Az ve Şeker Orta İSE ,O HALDE Kalite Azdır.

KURAL 3:EĞER Meyve Suyu Az ve Su Az ve Şeker Çok İSE ,O HALDE Kalite Azdır.

KURAL 4:EĞER Meyve Suyu Az ve Su Orta ve Şeker Az İSE ,O HALDE Kalite Azdır.

KURAL 5:EĞER Meyve Suyu Az ve Su Orta ve Şeker Orta İSE ,O HALDE Kalite Azdır.

KURAL 6:EĞER Meyve Suyu Az ve Su Orta ve Şeker Çok İSE ,O HALDE Kalite Azdır.

KURAL 7:EĞER Meyve Suyu Az ve Su Çok ve Şeker Az İSE ,O HALDE Kalite Azdır.

KURAL 8:EĞER Meyve Suyu Az ve Su Çok ve Şeker Orta İSE ,O HALDE Kalite Azdır.

KURAL 9:EĞER Meyve Suyu Az ve Su Çok ve Şeker Çok İSE ,O HALDE Kalite Azdır.

KURAL 10:EĞER Meyve Suyu Orta ve Su Az ve Şeker Az İSE ,O HALDE Kalite Azdır.

KURAL 11:EĞER Meyve Suyu Orta ve Su Az ve Şeker Orta İSE ,O HALDE Kalite Ortadır.

KURAl 12:EĞER Meyve Suyu Orta ve Su Az ve Çok İSE ,O HALDE Kalite Ortadır..

KURAL 13:EĞER Meyve Suyu Orta ve Su Orta ve Şeker Az İSE ,O HALDE Kalite Çoktur.

KURAL 14:EĞER Meyve Suyu Orta ve Su Orta ve Şeker Orta İSE ,O HALDE Kalite Çoktur.

KURAL 15:EĞER Meyve Suyu Orta ve Su Orta ve Şeker Çok İSE ,O HALDE Kalite Çoktur.

KURAL 16:EĞER Meyve Suyu Orta ve Su Çok ve Şeker Az İSE ,O HALDE Kalite Ortadır.

KURAL 17:EĞER Meyve Suyu Orta ve Su Çok ve Şeker Orta İSE ,O HALDE Kalite Ortadır.

KURAL 18:EĞER Meyve Suyu Orta ve Su Çok ve Şeker Çok İSE ,O HALDE Kalite Ortadır.

KURAL 19:EĞER Meyve Suyu Çok ve Su Az ve Şeker Az İSE ,O HALDE Kalite Çoktur..

KURAL 20:EĞER Meyve Suyu Çok ve Su Az ve Şeker Orta İSE ,O HALDE Kalite Çoktur.

KURAL 21:EĞER Meyve Suyu Çok ve Su Az ve Şeker Çok İSE ,O HALDE Kalite Çoktur.

KURAL 22:EĞER Meyve Suyu Çok ve Su Orta ve Şeker Az İSE ,O HALDE Kalite Çoktur.

KURAL 23:EĞER Meyve Suyu Çok ve Su Orta ve Şeker Orta İSE ,O HALDE Kalite Çoktur.

KURAL 24:EĞER Meyve Suyu Çok ve Su Orta ve Şeker Çok İSE ,O HALDE Kalite Çoktur.

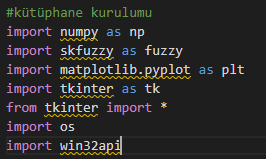
KURAL 25:EĞER Meyve Suyu Çok ve Su Çok ve Şeker Az İSE ,O HALDE Kalite Azdır.

KURAL 26:EĞER Meyve Suyu Çok ve Su Çok ve Şeker Orta İSE ,O HALDE Kalite Azdır.

KURAL 27:EĞER Meyve Suyu Çok ve Su Çok ve Şeker Çok İSE ,O HALDE Kalite Azdır.

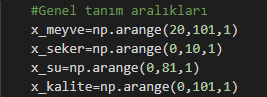
**3.5** Python ortamı

Öncelikle kullanacağımız kütüphaneleri ekledik:

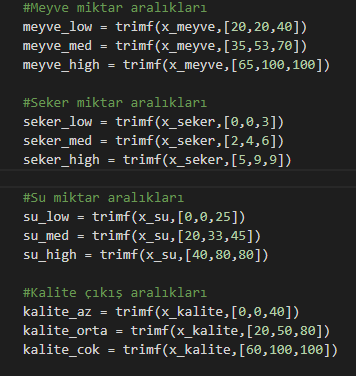


NumPy, Scikit-Fuzzy ve (görselleştirme için) Matplotlib kütüphanelerini ekliyoruz.

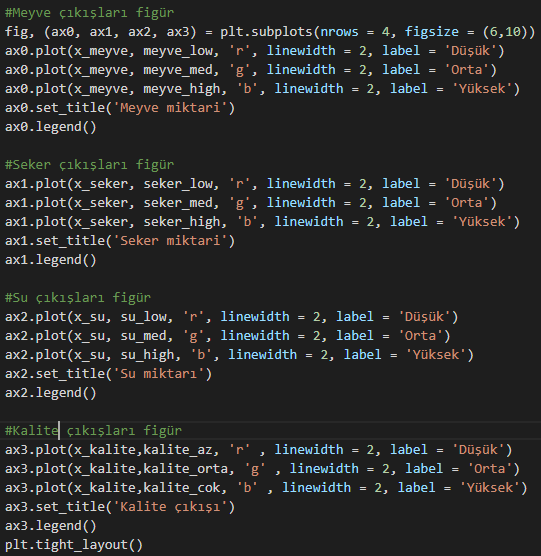
Meyve suyu, şeker, normal su ve kalite değişkenlerinin her biri için birer değişken üretiyoruz.NumPy kütüphanesinden “arange” metodu ile her bir değişken için önceden belirlenmiş olan tanım aralıklarını tanımlıyoruz.



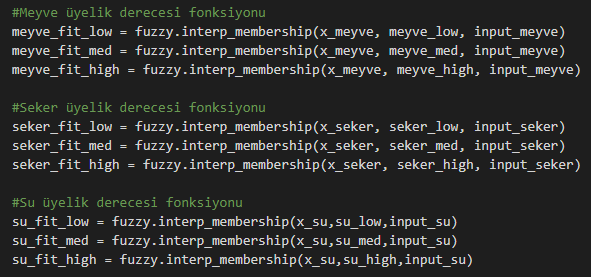
Şekil 3’de belirlediğimiz değerleri dikkate alarak, dilsel değerlerin her biri için, üyelik fonksiyonlarını oluşturuyoruz.

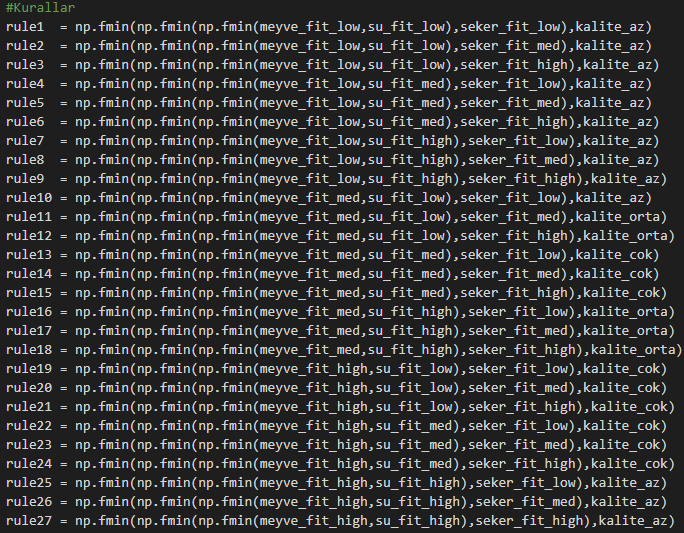


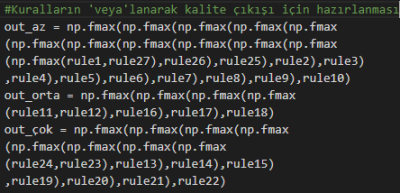
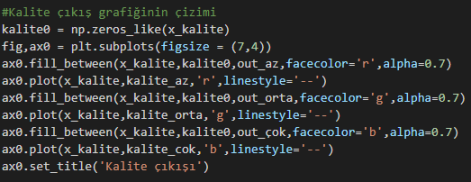
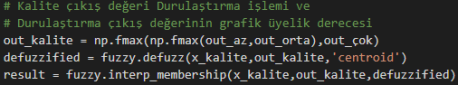
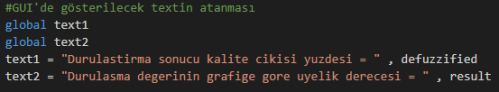
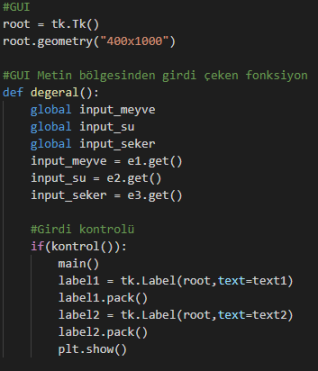
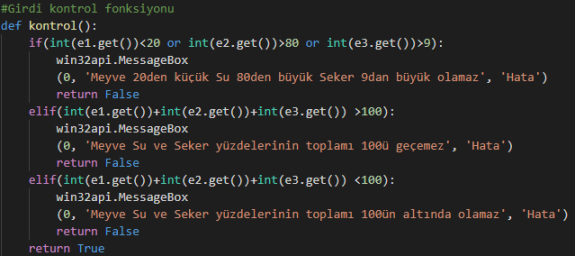
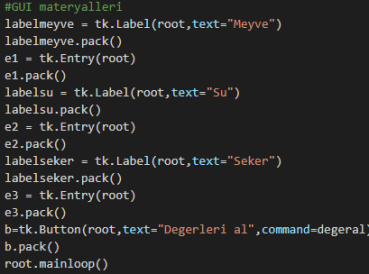
Meyve , şeker , su miktarlarının ve kalite çıkışının az orta ve çok eşik değerlerinin triangular membership function(Üçgen üyelik fonksiyonu) ‘da belirlenmesi



Oluşturduğumuz fonksiyonları, Matplotlib kütüphanesini kullanarak grafiğe dökelim.

Meyve şeker ve su için input değerlerinin eşik tanım değerleri üzerinde üyelik derecesinin hesaplanması

Bulanık mantık modelimizin kuralları

Kurallarımızdan elde ettiğimiz dizi sonuçlarından çıkışları aynı kalite çıkışını gösteren kuralların ‘veya’lanarak (max) kalite çıkışı eşik değerleri için atamasının yapılmasıElde ettiğimiz kalite eşik aralıklarına göre belirlenen kalite çıkış değerlerini de kullanarak ve fill\_between gibi matplotlib kütüphanesi fonksiyonları ile üyelik derecesini gösterebileceğimiz grafiği çizelim.Elde ettiğimiz kalite eşik çıkış değerlerinin tek bir kalite çıkış değerinde ‘veya’lanarak(max) bir kalite çıkış değeri elde etmemiz ve bu değeri kullanarak durulaştırma işlemi ile beraber gerçek yüzdelik değeri elde ediyoruz. Result ile birlikte ise gerçek duru değerimizin bulanıklaştırılarak grafik üzerinde üyelik derecesinin hesaplanması söz konusu.GUI Ekranında durulaştırma sonucu elde ettiğimiz ve o değerin grafik üzerinde ki üyelik derecelendirmesini göstermek için kullandığımız result değerini ekrana bastırmak için kullanacağımız text değerlerine atamasını yapıyoruz.GUI Ekranının oluşturulması ve input değerlerinin metin girme ekranındaki girilen değerlerden atanması ve girilen değerlerin kontrol edileceği if’e girmesi. If içerisinde ana işlemlerin ve çizim işlemlerinin yapılacağı main fonksiyonu çağırılıyor ve sonuçlar GUI’de metin olarak bastırılıyor. plt.show ile de çizimlerin grafiğie dökülmesi söz konusu.Kontrol fonksiyonunun içerisinde ki if koşulları ile mevcut girdilerin toplam değerinin istenilen değer aralıklarında olup olmadığı kontrol ediliyor. İstenilmeyen bir durum ile karşılaşılırsa hata mesajı verip kullanıcıyı yeni veriler girmeye yönlendiriyor.GUI ekranında gözükecek metin girme ekranı ve ekranın etiketi ve ekrandaki yerleri belirleniyor. Bu materyallerin üzerinde gözükecek metin atamaları yapılıyor ve degeral fonksiyonu ile input değerleri metin girme ekranlarından çekilip atanıyor.

# 4. Kullanıcı Kataloğu

Kullanıcı programı başlattığı zaman karşısına çıkan açılış ekranı:

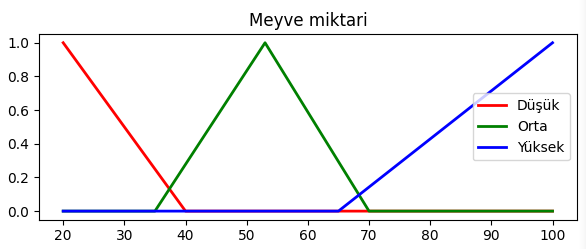
Bu ekranda meyve suyu (%), normal su (%) ve şeker miktarı (%) verilerini girer. Bu verileri girip “Degerleri al” butonuna bastıktan sonra program çalışır.

Program arka tarafta yazılan algoritmayı çalıştırır ve karşımıza 2 farklı pencere çıkar.

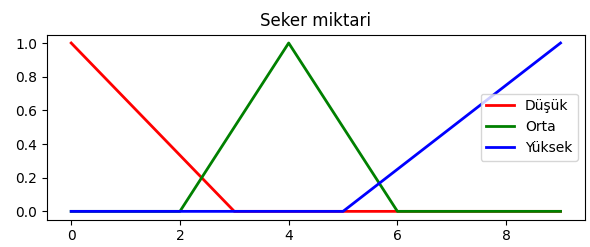
İstenilen uygun değerler girildikten ve butona basıldıktan sonra elde edilecek ekranlar aşağıdakiler gibidir.

Bu pencerelerden ilkinde 4 adet grafik vardır.

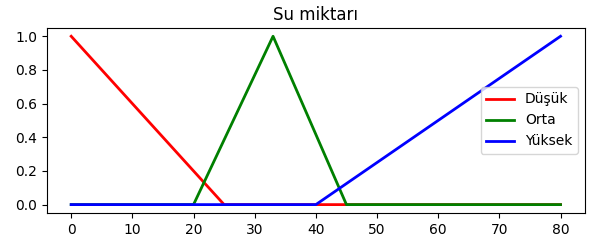
Meyve eşik değerlerinin grafiğe dökülmesi



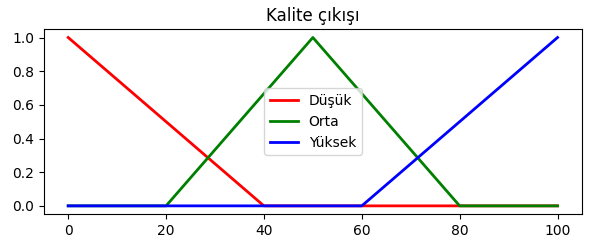
Şeker eşik değerlerinin grafiğe dökülmesi



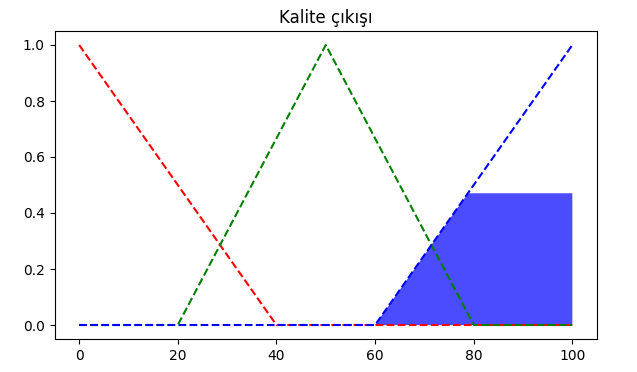
Su eşik değerlerinin grafiğe dökülmesi



Kalite eşik değerlerinin grafiğe dökülmesi



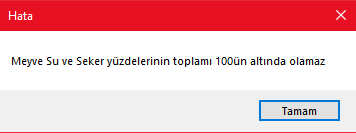
İkinci pencerede ise tek bir kalite çıkış grafiği vardır. Örnek:



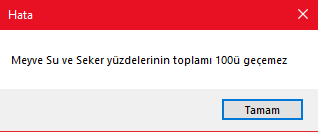
Örnek kalite çıkış grafiğine göre üyelik derecesinin ve kalite yüzdesinin hesaplandığı bölüm:



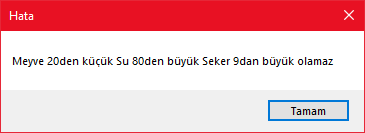
Eğer girdi değerleri istenilen koşullara uymazsa alınabilecek potansiyel hata ekranları bu şekildedir:



Değerler %lik olarak belirlendiği için 3 parametrenin toplam %lik değer %100 ün altında olamaz.



Değerler %lik olarak belirlendiği için 3 parametrenin toplam %lik değer %100 ün üstünde olamaz.



Meyve Su ve Şeker miktar eşik aralıkları sırasıyla

Meyve 20 -100

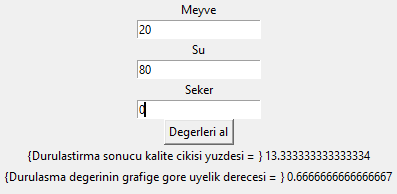
Su 0 - 80

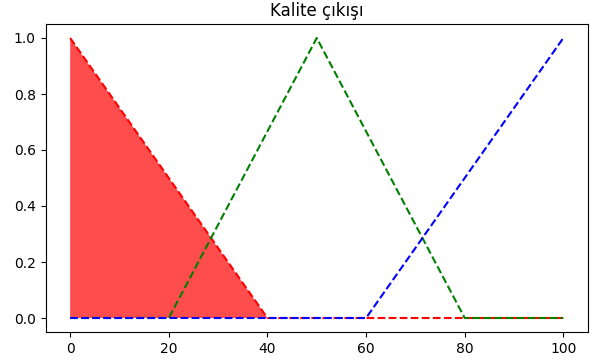
Şeker 0 - 9

Değerleri olduğundan ötürü bu değerler dışında bir değer girilemez.

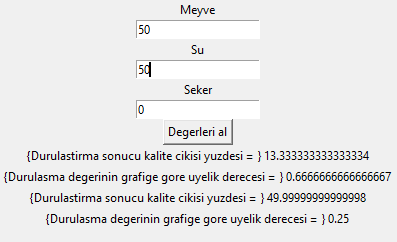
**5. Örnek Denemeler**

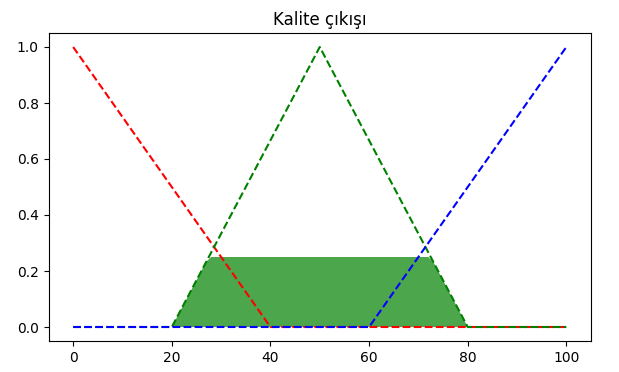
**Düşük kalite çıkışlı sınır giriş değerleriyle kalite çıkışı yüzdesi , üyelik derecesi ve kalite çıkışı grafiği:**

****

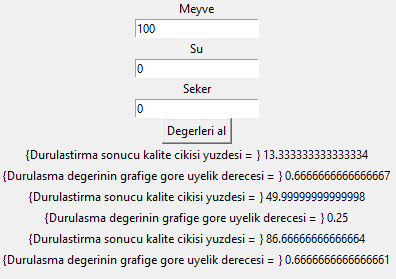
****

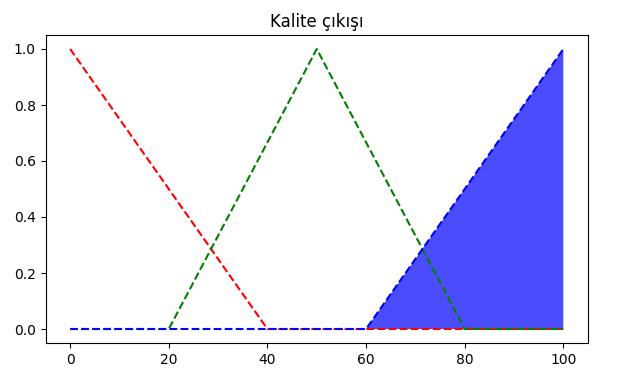
**Orta kaliteye yakın çıkışlı sınır giriş değerleriyle kalite çıkışı yüzdesi , üyelik derecesi ve kalite çıkışı grafiği:**

****

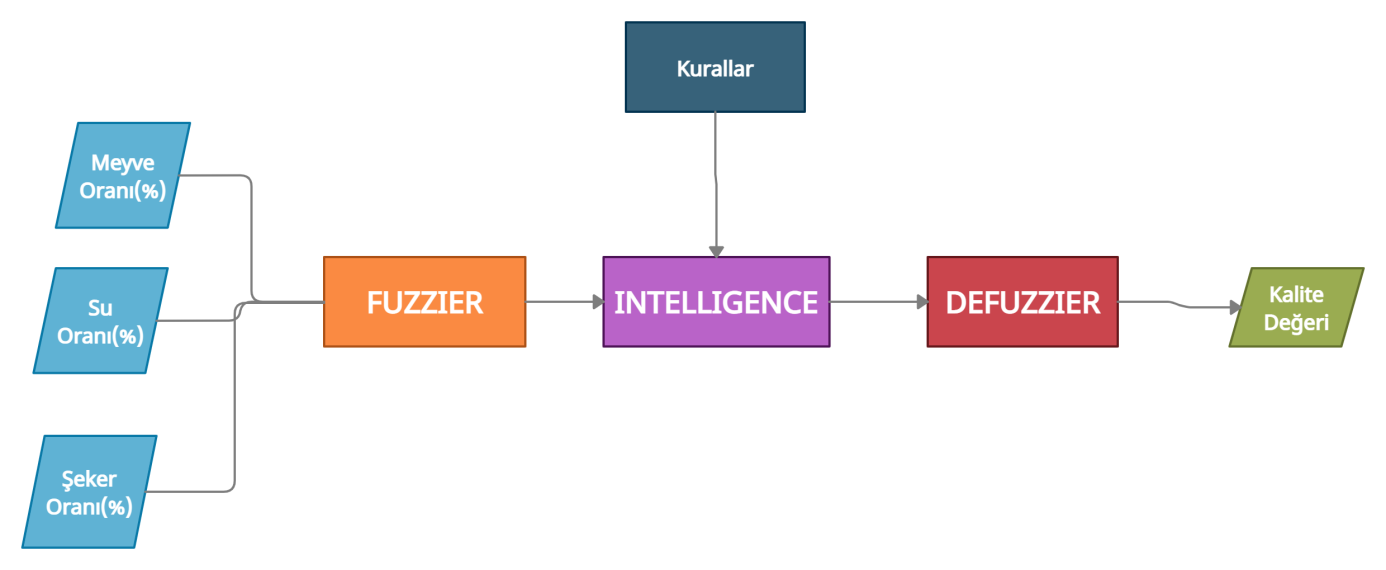
****

**Yüksek kalite çıkışlı sınır giriş değerleriyle kalite çıkışı yüzdesi , üyelik derecesi ve kalite çıkışı grafiği:**

****

****

**6. Akış Diyagramı**



# 

# 7. Sonuçlar

3 adet girdi(Meyve , Su , Şeker) elde ettikten sonra bu değerlerin tanım aralıkları üzerinde az orta veya çok olacak şekilde üyelik dereceleri hesaplanıyor. Bu tanım aralıklarının grafiklerinin matplotlib kütüphanesi ile çiziminden sonra input değerleri de kullanılarak üyelik fonksiyonuna gönderiliyor. Nihai olarak kalite çıkış grafiğimizde elimize bu input değerlerinin kural süzgecinden geçtikten sonra ve bu kuralların çıkışlarının toplam değeri belirlendikten sonra kalite çıkış grafiğinde üyelik derecesi ile beraber nihai çizim yapılabiliyor. Girdi değerleri GUI ekranında kontrol fonksiyonu ile kontrol edildikten sonra hata var ise hata ekranı verebiliyor. Çalışan değerler verildikten sonra giriş arayüzünün hemen altında hem nihai üyelik derecesi hem de durulanmış gerçek değer yüzdesi elde edilebiliyor.

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 8. Karşılaşılan Problemler ve Çözüm Yaklaşımları

1)Ski-fuzzy kütüphanesi import edilmesi aşamasında,pip install komutunu komut istemine yazdığımızda istediğimiz sonucu alamadık.Bu kütüphanenin python’un 3 ve üzeri sürümlerinde çalışmadığını farkettik.Daha sonra pip komutu yerine pip3 komutunu kullanarak Anaconda’nın kendi istemcisi olan Anaconda Prompt’ta çalıştırdıktan sonra bu sorun çözüme ulaştı.

2)Ödev isterleri doğrultusunda bulanık mantık üyelik fonksiyonları üçgenleştiren hazır fonksiyon olan trimf fonkisyonunu kullanmayacağımızı gördük.Bu problem doğrultusunda “trimf” ile aynı işleve sahip kendi fonksiyonumuzu oluşturduk.

3)Kuralların olabilecek ve olamayacak senaryolardaki sonuçlarında problem yaşadık.3 parametreli kural içersinde 27 kuralın ‘veya’lanmasında problem yaşadık çünkü ilk başta 9 kural olarak düşündük.Ancak daha sonra doğru sonuca 27 kural ile ulaştık.

# 

# 

# 

# 9. Kaynakça

[1]https://www.iienstitu.com/blog/bulanik-mantik-nedir-nasil-uygulanir

[2]DEÜ MÜHENDİSLİK FAKÜLTESİ MÜHENDİSLİK BİLİMLERİ DERGİSİ Cilt: 15 Sayı: 2 sh. 15-29 Ocak 2013

[3] http://www.cappy.com.tr/tr/cappy-cesitleri/cappy-bahce/

[4]<https://stackoverflow.com/questions/177287/alert-boxes-in-python>

[5]https://www.w3schools.com/python/python\_variables\_global.asp

[6]<https://www.javatpoint.com/python-tkinter-entry>

[7] [GitHub - sorend/fuzzy4j: Fuzzy logic library for Java](https://github.com/sorend/fuzzy4j)

[8] [GitHub - cschen1205/java-fuzzy-logic: Package provides java implementation of fuzzy logic controller](https://github.com/cschen1205/java-fuzzy-logic)

[9]https://pythonexamples.org/python-tkinter-button-click-call-function-2/#:~:text=Python%20Tkinter%20–%20Call%20Function%20on,when%20the%20button%20is%20clicked.

[10]https://realpython.com/python-gui-tkinter/#displaying-clickable-buttons-with-button-widgets

[11]<https://www.tutorialspoint.com/python/python_gui_programming.html>

[12] <https://www.javatpoint.com/fuzzy-logic>

[13] <https://www.geeksforgeeks.org/fuzzy-logic-introduction/>

[14] [Lec2012-3-159741-FuzzyLogic-v.2.pdf (massey.ac.nz)](https://www.massey.ac.nz/~nhreyes/MASSEY/159741/Lectures/Lec2012-3-159741-FuzzyLogic-v.2.pdf)

[15]<https://ahmetatasoglu98.medium.com/bulan%C4%B1k-mant%C4%B1k-3-bulan%C4%B1k-kurallar-ve-%C3%A7%C4%B1kar%C4%B1m-8f9d411080c>

[16]<https://www.researchgate.net/publication/264428657_Cig_Sut_Kalite_Degerlendirmesinde_Bulanik_Mantik_Yaklasimi>

[17]<http://www.saujs.sakarya.edu.tr/tr/download/article-file/192916>

[18]<http://jfuzzylogic.sourceforge.net/html/java.html>

[19]<https://github.com/amogorkon/fuzzylogic>

[20]<https://matplotlib.org>

[21][**https://studylibtr.com/doc/2333719/veri%CC%87tabanlarinda-geneti%CC%87k-algori%CC%87tma-ve-bulanik-mantik**](https://studylibtr.com/doc/2333719/veri%CC%87tabanlarinda-geneti%CC%87k-algori%CC%87tma-ve-bulanik-mantik)