

Implementacion

October 28, 2020

1 FUNCION MEMBRESIA TRIANGULAR

```
[8]: #Se importan las librerias necesarias
import numpy as np
import skfuzzy as sk
import matplotlib.pyplot as plt
%matplotlib inline

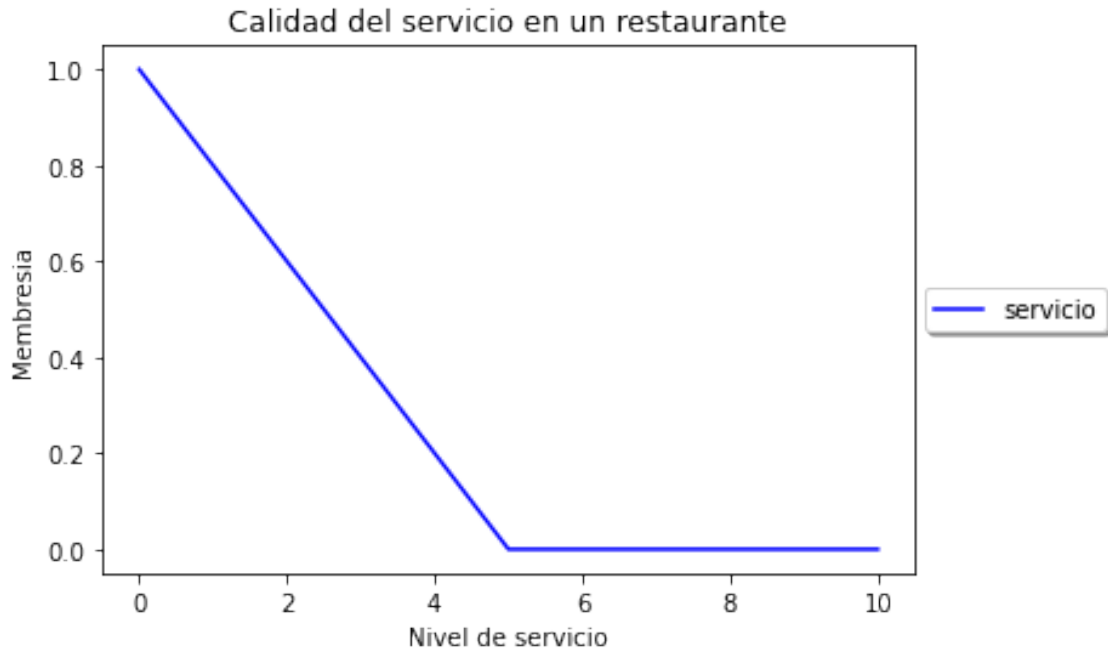
x = np.arange(0,11,1) #Se define un array x
calidad = sk.trimf(x,[0,0,0])#Se define un array para la funcion membresia de
    ↳ tipo triangular

#Graficación de la función membresia.
plt.figure()
#Se crea recuadro indicador de linea
plt.plot(x,calidad,'b',linewidth=1.5,label='servicio')
#Se define el titulo de la grafica
plt.title('Calidad del servicio en un restaurante')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Nivel de servicio')
#Ubicacion del recuadro con informacion de linea
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
    ↳ True,shadow=True)
plt.show()
```



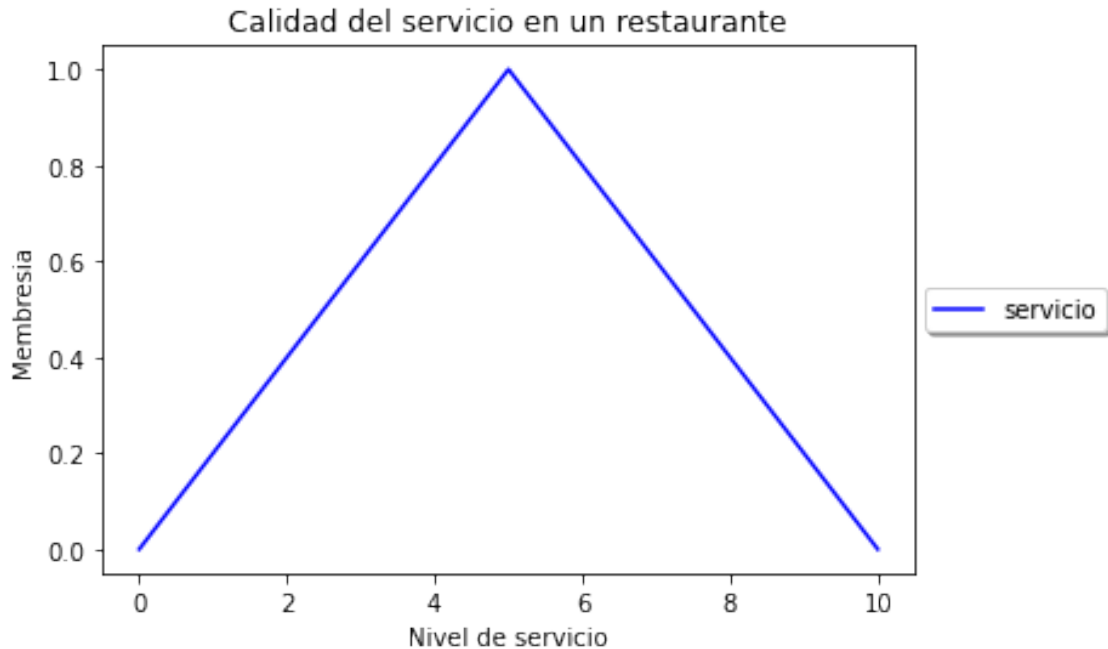
```
[9]: #Funcion membresia triangular modificando los valores del arreglo funcion
      ↪membresia
calidad = sk.trimf(x,[0,0,5])#Se define un array para la funcion membresia de
      ↪tipo triangular

#Graficación de la función membresia.
plt.figure()
#Se crea recuadro indicador de linea
plt.plot(x,calidad,'b',linewidth=1.5,label='servicio')
#Se define el titulo de la grafica
plt.title('Calidad del servicio en un restaurante')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Nivel de servicio')
#Ubicación del recuadro con informacion de linea
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
      ↪True,shadow=True)
plt.show()
```



```
[10]: #Funcion membresia triangular modificando los valores del arreglo funcion
      ↪membresia
calidad = sk.trimf(x,[0,5,10])#Se define un array para la funcion membresia de
      ↪tipo triangular

#Graficación de la función membresia.
plt.figure()
#Se crea recuadro indicador de linea
plt.plot(x,calidad,'b',linewidth=1.5,label='servicio')
#Se define el titulo de la grafica
plt.title('Calidad del servicio en un restaurante')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Nivel de servicio')
#Ubicación del recuadro con información de línea
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
      ↪True,shadow=True)
plt.show()
```



```
[11]: #Funcion membresia triangular modificando los valores del arreglo funcion
      ↪membresia
calidad = sk.trimf(x,[9,9,10])#Se define un array para la funcion membresia de
      ↪tipo triangular

#Graficación de la función membresia.
plt.figure()
#Se crea recuadro indicador de linea
plt.plot(x,calidad,'b',linewidth=1.5,label='servicio')
#Se define el titulo de la grafica
plt.title('Calidad del servicio en un restaurante')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Nivel de servicio')
#Ubicación del recuadro con informacion de linea
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
      ↪True,shadow=True)
plt.show()
```



```
[13]: #Funcion membresia triangular modificando los valores del arreglo funcion
      ↪membresia
calidad = sk.trimf(x,[10,10,10])#Se define un array para la funcion membresia
      ↪de tipo triangular

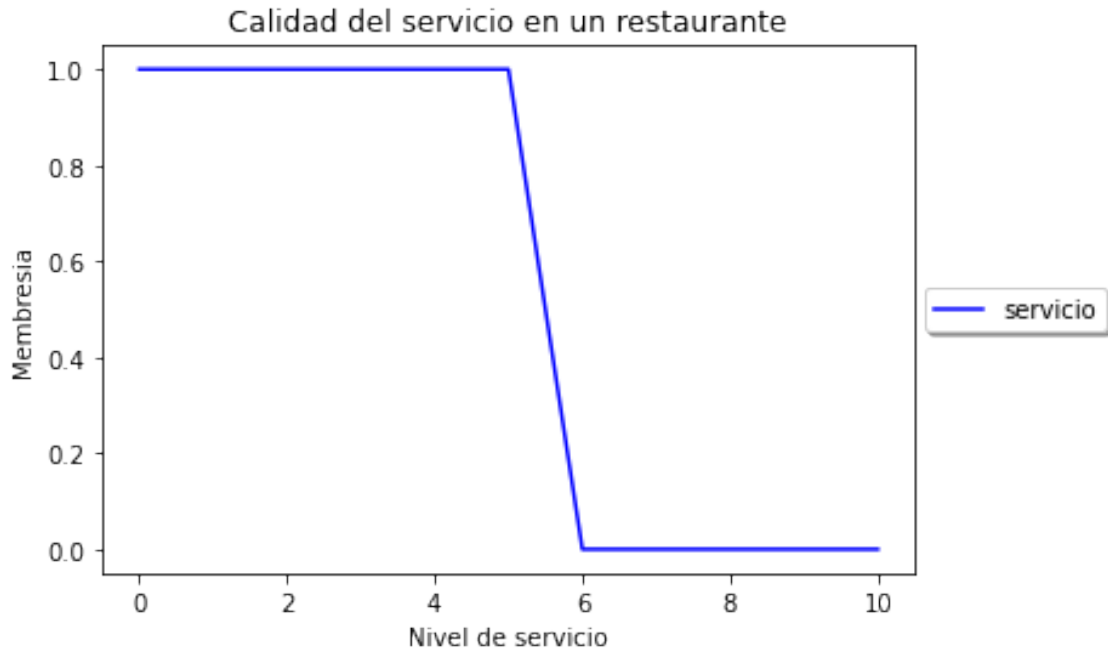
#Graficación de la función membresia.
plt.figure()
#Se crea recuadro indicador de linea
plt.plot(x,calidad,'b',linewidth=1.5,label='servicio')
#Se define el titulo de la grafica
plt.title('Calidad del servicio en un restaurante')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Nivel de servicio')
#Ubicación del recuadro con informacion de linea
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
      ↪True,shadow=True)
plt.show()
```



2 FUNCION MEMBRESIA TRAPEZOIDAL

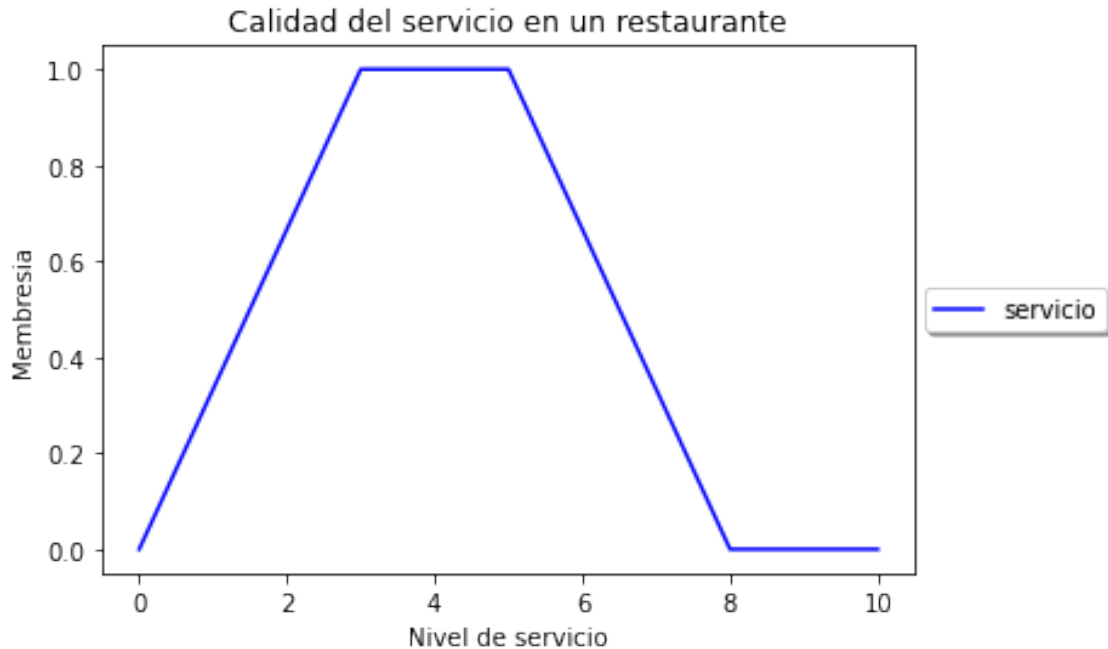
```
[15]: vd_trapezoidal = sk.trapmf(x,[0,0,5,5])#Se define un array para la funcion
      ↪membresia de tipo trapezoidal

      #Graficación de la función membresia.
      plt.figure()
      #Se crea recuadro indicador de linea
      plt.plot(x,vd_trapezoidal,'b',linewidth=1.5,label='servicio')
      #Se define el titulo de la grafica
      plt.title('Calidad del servicio en un restaurante')
      #Se asigna el nombre de la arista y de la grafica
      plt.ylabel('Membresia')
      #Se asigna el nombre de la arista x de la grafica
      plt.xlabel('Nivel de servicio')
      #Ubicación del recuadro con informacion de linea
      plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
      ↪True,shadow=True)
      plt.show()
```



```
[16]: #Funcion membresia trapezoidal modificando los valores del arreglo funcion
      ↪membresia
vd_trapezoidal = sk.trapmf(x,[0,3,5,8])#Se define un array para la funcion
      ↪membresia de tipo trapezoidal

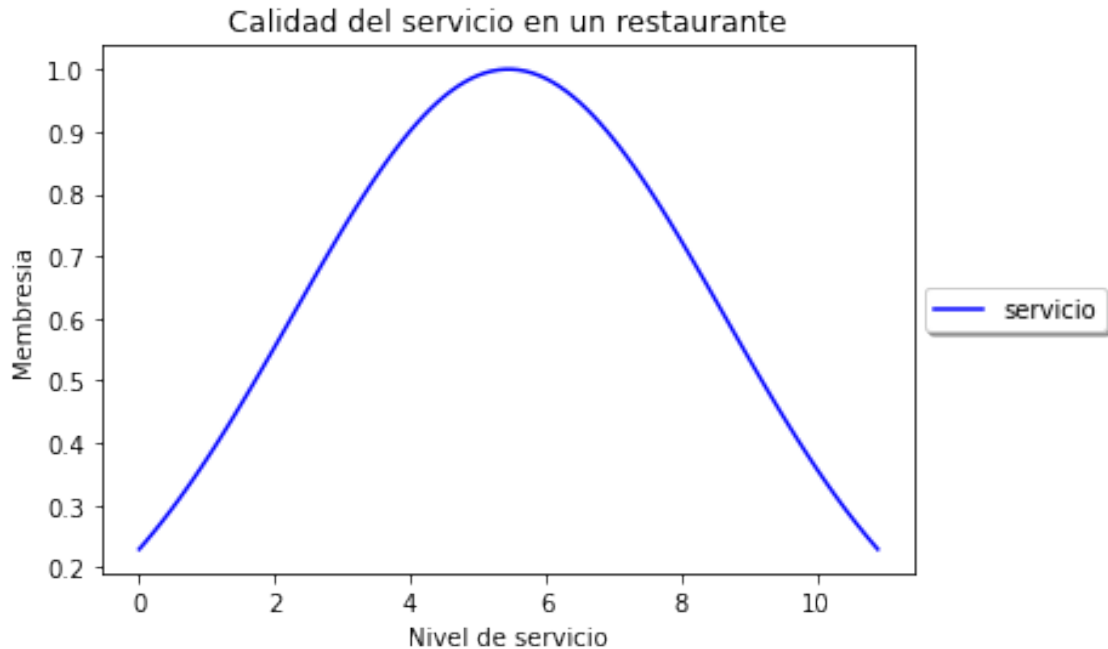
#Graficación de la función membresia.
plt.figure()
#Se crea recuadro indicador de linea
plt.plot(x,vd_trapezoidal,'b',linewidth=1.5,label='servicio')
#Se define el titulo de la grafica
plt.title('Calidad del servicio en un restaurante')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Nivel de servicio')
#Ubicación del recuadro con información de línea
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
      ↪True,shadow=True)
plt.show()
```



3 FUNCION GAUSSIANA

```
[18]: x = np.arange(0,11,0.1) #Se define un array x
vd_gaussiana = sk.gaussmf(x,np.mean(x),np.std(x))#Se define un array para la
    ↪funcion membresia gaussiana

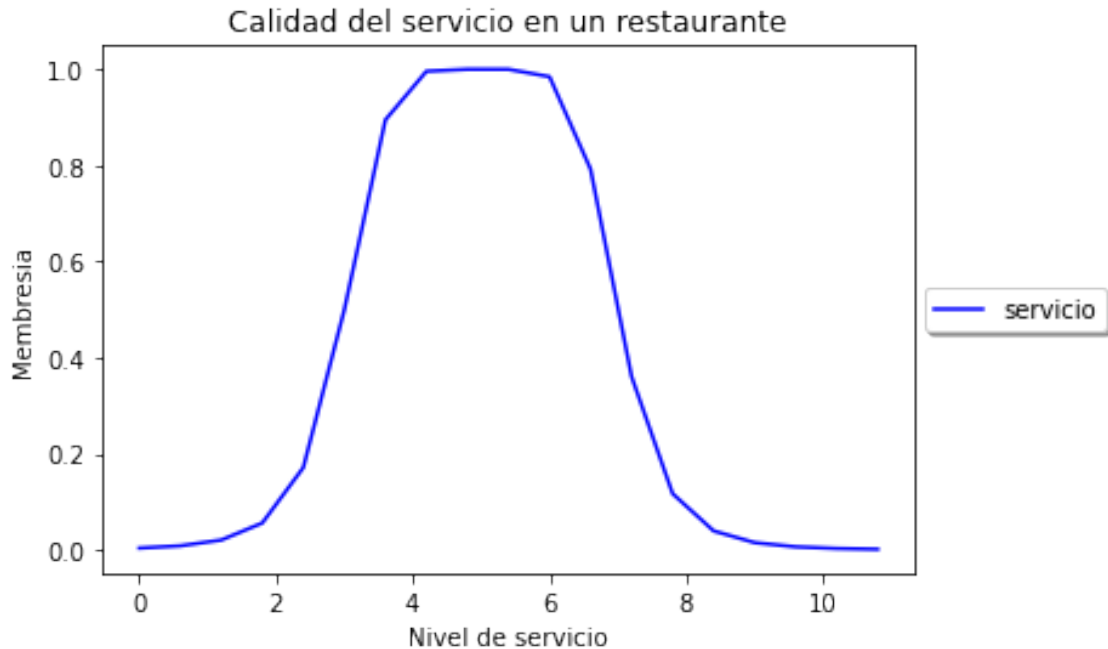
#Graficación de la función membresia.
plt.figure()
#Se crea recuadro indicador de linea
plt.plot(x,vd_gaussiana,'b',linewidth=1.5,label='servicio')
#Se define el titulo de la grafica
plt.title('Calidad del servicio en un restaurante')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Nivel de servicio')
#Ubicación del recuadro con información de línea
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
    ↪True,shadow=True)
plt.show()
```

4 FUNCION GAUSSIANA - BELL

```
[19]: x = np.arange(0,11,0.6) #Se define un array x
vd_gaussiana_bell = sk.gbellmf(x,2,3,5)#Se define un array para la funcion
      ↪membresia gaussiana - bell

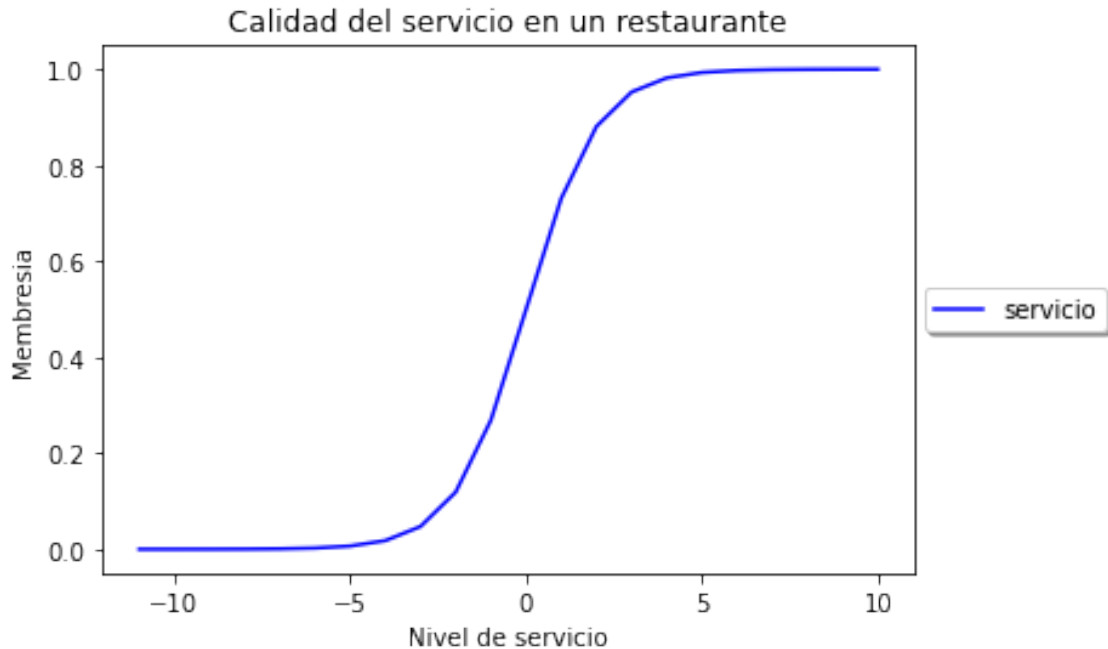
#Graficación de la función membresia.
plt.figure()
#Se crea recuadro indicador de linea
plt.plot(x,vd_gaussiana_bell,'b',linewidth=1.5,label='servicio')
#Se define el titulo de la grafica
plt.title('Calidad del servicio en un restaurante')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Nivel de servicio')
#Ubicación del recuadro con información de línea
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
      ↪True,shadow=True)
plt.show()
```



5 FUNCION SIGMOIDE

```
[20]: x = np.arange(-11,11,1) #Se define un array x
vd_sigmoide = sk.sigmf(x,0,1)#Se define un array para la funcion membresia
      ↪gaussiana

#Graficación de la función membresia.
plt.figure()
#Se crea recuadro indicador de linea
plt.plot(x,vd_sigmoide,'b',linewidth=1.5,label='servicio')
#Se define el titulo de la grafica
plt.title('Calidad del servicio en un restaurante')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Nivel de servicio')
#Ubicación del recuadro con información de línea
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
      ↪True,shadow=True)
plt.show()
```

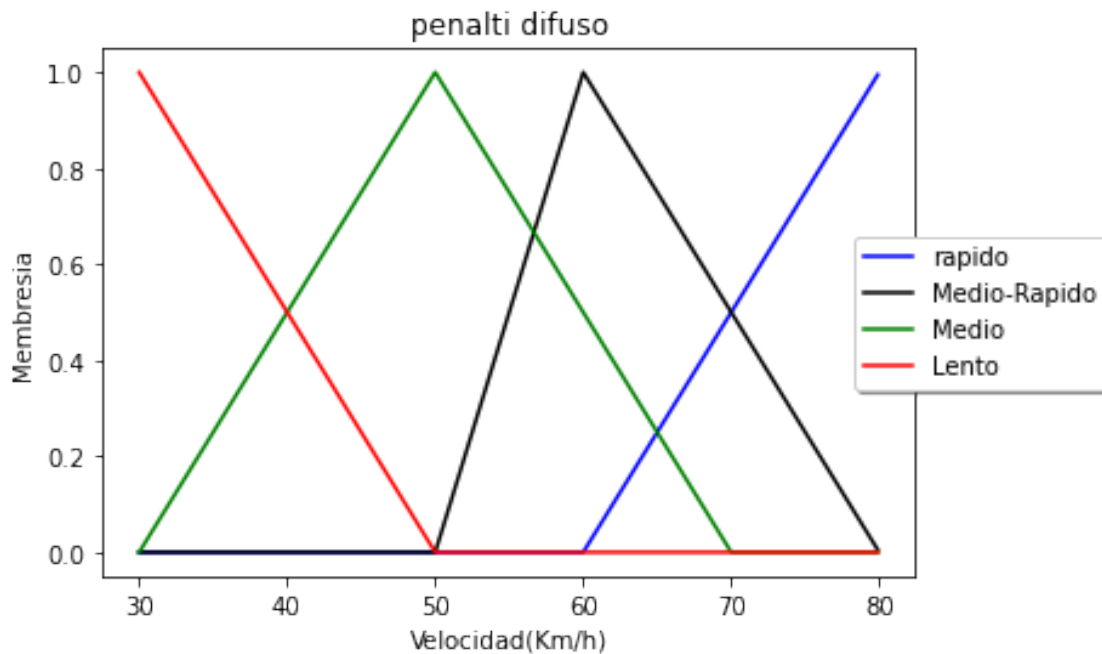


6 APLICACION FUTBOL

```
[22]: #Se definen los rangos de velocidad del balon de 30 a 80, con un aumento de 0.1
x = np.arange(30,80,0.1)
#Se definen las funciones membresia triangulares.
lento = sk.trimf(x, [30,30,50])
medio = sk.trimf(x, [30,50,70])
medio_rapido = sk.trimf(x, [50,60,80])
rapido = sk.trimf(x, [60,80,80])

#Se dibujan las funciones membresia
plt.figure()
#Se crea etiquetas indicadoras de linea
plt.plot(x,rapido,'b',linewidth=1.5,label='rapido')
plt.plot(x,medio_rapido,'k',linewidth=1.5,label='Medio-Rapido')
plt.plot(x,medio,'g',linewidth=1.5,label='Medio')
plt.plot(x,lento,'r',linewidth=1.5,label='Lento')
#Se define el titulo de la grafica
plt.title('penalti difuso')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Velocidad(Km/h)')
#Ubicacion del recuadro con informacion de linea
```

```
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
↳ True, shadow=True)
plt.show()
```

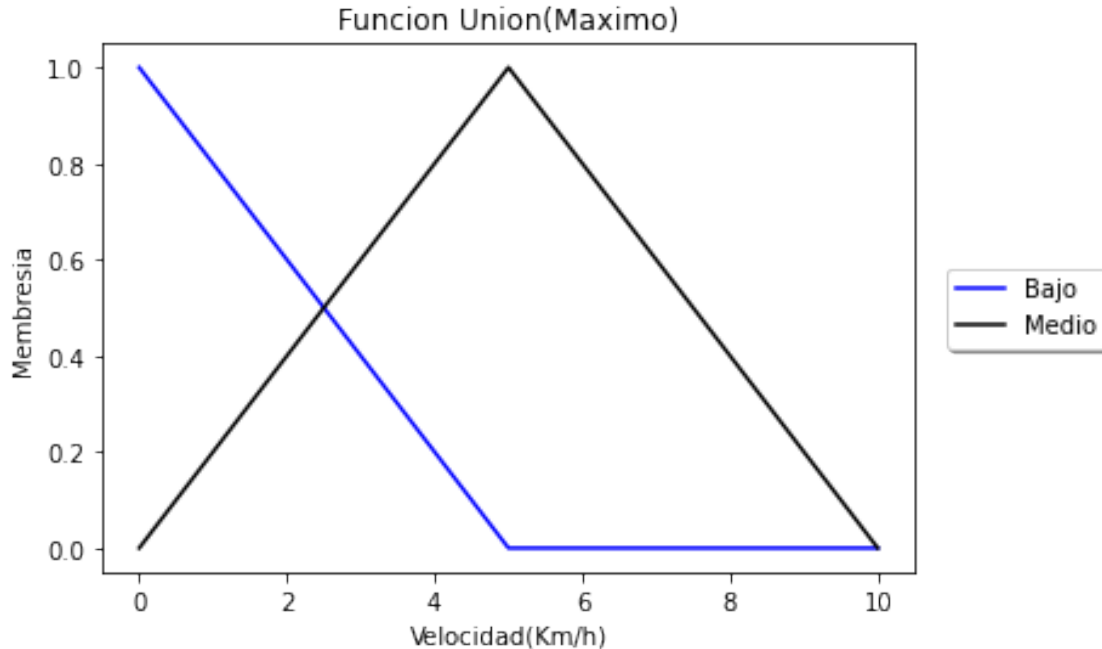


7 UNION

```
[23]: #Se definen los rangos de velocidad del balon de 0 a 11, con un aumento de 1
x = np.arange(0,11,1)
#Se definen las funciones membresia triangulares.
bajo = sk.trimf(x, [0,0,5])
medio = sk.trimf(x, [0,5,10])

#Se dibujan las funciones membresia
plt.figure()
#Se crea etiquetas indicadoras de linea
plt.plot(x,bajo,'b',linewidth=1.5,label='Bajo')
plt.plot(x,medio,'k',linewidth=1.5,label='Medio')
#Se define el titulo de la grafica
plt.title('Funcion Union(Maximo)')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Velocidad(Km/h)')
```

```
#Ubicacion del recuadro con informacion de linea
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
↳ True, shadow=True)
plt.show()
```



7.0.1 Graficacion de las lineas verticales por donde pasan los puntos

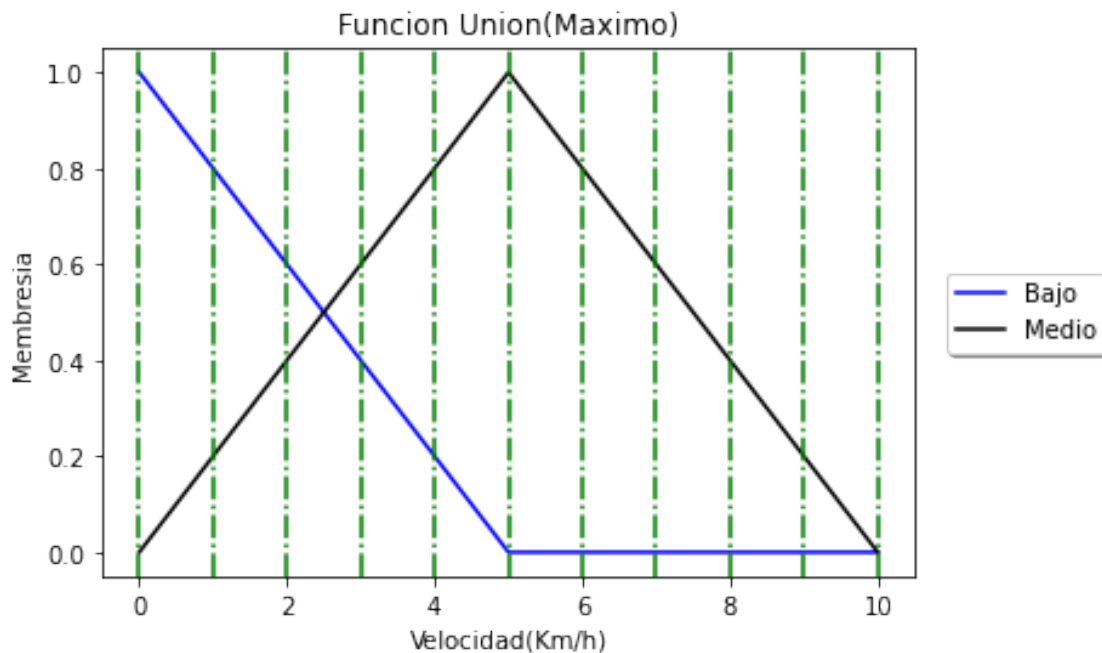
```
[27]: #Se definen los rangos de velocidad del balon de 0 a 11, con un aumento de 1
x = np.arange(0,11,1)
#Se definen las funciones membresia triangulares.
bajo = sk.trimf(x, [0,0,5])
medio = sk.trimf(x, [0,5,10])

#Se dibujan las funciones membresia
plt.figure()
#Se crea etiquetas indicadoras de linea
plt.plot(x,bajo,'b',linewidth=1.5,label='Bajo')
plt.plot(x,medio,'k',linewidth=1.5,label='Medio')
#Se define el titulo de la grafica
plt.title('Funcion Union(Maximo)')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Velocidad(Km/h)')
```

```

#Ubicacion del recuadro con informacion de linea
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
↳ True, shadow=True)
plt.axvline(x=0,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=1,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=2,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=3,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=4,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=5,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=6,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=7,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=8,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=9,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=10,ymin=0,ymax=10, color='g', linestyle='-.')
plt.show()

```



7.0.2 Graficacion de la intercepcion lineas verticales con las graficas superiores

```

[29]: #Se definen los rangos de velocidad del balon de 0 a 11, con un aumento de 1
x = np.arange(0,11,1)
#Se definen las funciones membresia triangulares.
bajo = sk.trimf(x, [0,0,5])
medio = sk.trimf(x, [0,5,10])

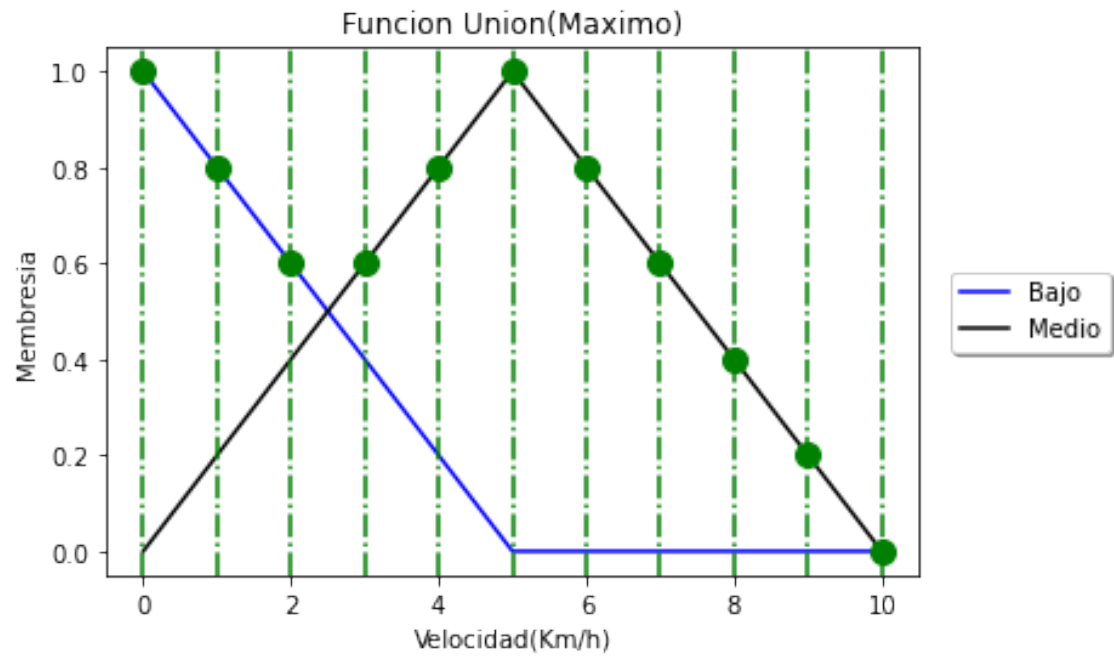
```

```

#Se dibujan las funciones membresia
plt.figure()
#Se crea etiquetas indicadoras de linea
plt.plot(x,bajo,'b',linewidth=1.5,label='Bajo')
plt.plot(x,medio,'k',linewidth=1.5,label='Medio')
#Se define el titulo de la grafica
plt.title('Funcion Union(Maximo)')
#Se asigna el nombre de la arista y de la grafica
plt.ylabel('Membresia')
#Se asigna el nombre de la arista x de la grafica
plt.xlabel('Velocidad(Km/h)')
#Ubicacion del recuadro con informacion de linea
plt.legend(loc='center right', bbox_to_anchor=(1.25,0.5), ncol = 1, fancybox=
↳ True, shadow=True)
plt.axvline(x=0,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=1,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=2,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=3,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=4,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=5,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=6,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=7,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=8,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=9,ymin=0,ymax=10, color='g', linestyle='-.')
plt.axvline(x=10,ymin=0,ymax=10, color='g', linestyle='-.')

plt.plot(0, 1, marker= 'o', markersize = 10, color = 'g')
plt.plot(1, 0.8, marker= 'o', markersize = 10, color = 'g')
plt.plot(2, 0.6, marker= 'o', markersize = 10, color = 'g')
plt.plot(3, 0.6, marker= 'o', markersize = 10, color = 'g')
plt.plot(4, 0.8, marker= 'o', markersize = 10, color = 'g')
plt.plot(5, 1, marker= 'o', markersize = 10, color = 'g')
plt.plot(6, 0.8, marker= 'o', markersize = 10, color = 'g')
plt.plot(7, 0.6, marker= 'o', markersize = 10, color = 'g')
plt.plot(8, 0.4, marker= 'o', markersize = 10, color = 'g')
plt.plot(9, 0.2, marker= 'o', markersize = 10, color = 'g')
plt.plot(10, 0, marker= 'o', markersize = 10, color = 'g')
plt.show()

```



[]: