ANEXOS

PROGRMAS:

VoiceRecognition_2.py

```
#-------------------------------------------------------------------------------
# Name:        Reconocimiento de voz por Tr. Wavelet (32-bits)
# Purpose:
#
# Author:      Emi
#
# Created:     18/02/2015
# Copyright:   (c) Emi 2015
# Licence:     <your licence>
#-------------------------------------------------------------------------------
import matplotlib
import matplotlib as mpl
import matplotlib.pyplot as plt
import pylab
from pylab import*
import dateutil
import pyparsing
import six
import sys
import os
import PIL
import random
import wave
#
import numpy
from numpy import*
from numpy import loadtxt
import numpy as np
import scipy
from scipy.io import wavfile
import scipy.misc
import pywt
import pymix
from pymix import mixture

##################################################################
#________________LECTURA AUDIO_COEFICIENTES WAVELET_________
#______________Coef. de aproximaci?n y de detalle___________
fs, dataarray = wavfile.read('AnitaEmi.wav')            #57344 samples
```

```
pywt.dwt(dataarray, 'haar', 'sym')
(cA, cD) = pywt.dwt(dataarray, 'haar', mode ='sym')
fs, dataarray2 = wavfile.read('AnitaNat.wav')              #59392 samples
pywt.dwt(dataarray2, 'haar', 'sym')
(cA2, cD2) = pywt.dwt(dataarray2, 'haar', mode='sym')
fs, dataarray3 = wavfile.read('AnitaMr.wav')
pywt.dwt(dataarray3, 'haar', 'sym')
(cA3, cD3) = pywt.dwt(dataarray3, 'haar', mode='sym')
fs, dataarray4 = wavfile.read('AnitaFabby.wav')
pywt.dwt(dataarray4, 'haar', 'sym')
(cA4, cD4) = pywt.dwt(dataarray4, 'haar', mode='sym')
fs, dataarray5 = wavfile.read('AnitaMama.wav')
pywt.dwt(dataarray5, 'haar', 'sym')
(cA5, cD5) = pywt.dwt(dataarray5, 'haar', mode='sym')
fs, dataarray6 = wavfile.read('AnitaPapa.wav')
pywt.dwt(dataarray6, 'haar', 'sym')
(cA6, cD6) = pywt.dwt(dataarray6, 'haar', mode='sym')

#_________________________MEZCLA_E.M.___________________________
########################### SEA?L 1 ############################
nCA = mixture.NormalDistribution(cA.mean(),cA.std())
nCD = mixture.NormalDistribution(cD.mean(),cD.std())
m = mixture.MixtureModel(2,[0.5,0.5], [nCA,nCD])
print "nCA: ",nCA
print "nCD: ",nCD
print m.pi
print ""
x = cA
x = np.array(x.tolist())
y = cD
y = np.array(y.tolist())
list = x.tolist()
list.extend(y.tolist())
dataCACD = np.asarray(list)
dataM = mixture.DataSet()
dataM.fromArray(dataCACD)     #57343 samples #(cA,cD)
m.EM(dataM,40,0.1)

############################ Nat
nCA2 = mixture.NormalDistribution(cA2.mean(),cA2.std())
nCD2 = mixture.NormalDistribution(cD2.mean(),cD2.std())
m2 = mixture.MixtureModel(2,[0.5,0.5], [nCA2,nCD2])
print ""
print "nCA2:",nCA2
print "nCD2:",nCD2
print m2.pi
print ""
```

```
x2 = cA2
x2 = np.array(x2.tolist())
y2 = cD2
y2 = np.array(y2.tolist())
list = x2.tolist()
list.extend(y2.tolist())
dataCA2CD2 = np.asarray(list)
dataM2 = mixture.DataSet()
dataM2.fromArray(dataCA2CD2)   #59391 samples
m2.EM(dataM2,40,0.1)
########################### MR
nCA3 = mixture.NormalDistribution(cA3.mean(),cA3.std())
nCD3 = mixture.NormalDistribution(cD3.mean(),cD3.std())
m3 = mixture.MixtureModel(2,[0.5,0.5], [nCA3,nCD3])
print "nCA: ",nCA3
print "nCD: ",nCD3
print m3.pi
print ""
x = cA3
x = np.array(x.tolist())
y = cD3
y = np.array(y.tolist())
list = x.tolist()
list.extend(y.tolist())
dataCA3CD3 = np.asarray(list)
dataM3 = mixture.DataSet()
dataM3.fromArray(dataCA3CD3)     #57343 samples #(cA,cD)
m3.EM(dataM3,40,0.1)
######################### FABBY
nCA4 = mixture.NormalDistribution(cA4.mean(),cA4.std())
nCD4 = mixture.NormalDistribution(cD4.mean(),cD4.std())
m4 = mixture.MixtureModel(2,[0.5,0.5], [nCA4,nCD4])
print "nCA: ",nCA4
print "nCD: ",nCD4
print m4.pi
print ""
x = cA4
x = np.array(x.tolist())
y = cD4
y = np.array(y.tolist())
list = x.tolist()
list.extend(y.tolist())
dataCA4CD4 = np.asarray(list)
dataM4 = mixture.DataSet()
dataM4.fromArray(dataCA4CD4)     #57343 samples #(cA,cD)
m4.EM(dataM4,40,0.1)
######################### Mama
```

```
nCA5 = mixture.NormalDistribution(cA5.mean(),cA5.std())
nCD5 = mixture.NormalDistribution(cD5.mean(),cD5.std())
m5 = mixture.MixtureModel(2,[0.5,0.5], [nCA5,nCD5])
print "nCA: ",nCA5
print "nCD: ",nCD5
print m5.pi
print ""
x = cA5
x = np.array(x.tolist())
y = cD5
y = np.array(y.tolist())
list = x.tolist()
list.extend(y.tolist())
dataCA5CD5 = np.asarray(list)
dataM5 = mixture.DataSet()
dataM5.fromArray(dataCA5CD5)     #57343 samples #(cA,cD)
m5.EM(dataM5,40,0.1)
############################## Papa
nCA6 = mixture.NormalDistribution(cA6.mean(),cA6.std())
nCD6 = mixture.NormalDistribution(cD6.mean(),cD6.std())
m6 = mixture.MixtureModel(2,[0.5,0.5], [nCA6,nCD6])
print "nCA: ",nCA6
print "nCD: ",nCD6
print m6.pi
print ""
x = cA6
x = np.array(x.tolist())
y = cD6
y = np.array(y.tolist())
list = x.tolist()
list.extend(y.tolist())
dataCA6CD6 = np.asarray(list)
dataM6 = mixture.DataSet()
dataM6.fromArray(dataCA6CD6)     #57343 samples #(cA,cD)
m6.EM(dataM6,40,0.1)
#__________________________ GRAFICA CURVA __________________________
###############_Coeficientes de Aproximaci?n_#####################
# ROJO EMI
plt.figure()
subplot(111,axisbg='#F1F1F1')
plt.title('Coeficientes de Aproximaci?n')
meanM = 500*nCA.mu
sigmaM = nCA.sigma
xM = np.linspace(-10000,10000,200)
plt.plot(xM,mlab.normpdf(xM,meanM,sigmaM),linewidth=1, color='#FF002A')
# AZUL SE?AL 2
subplot(111,axisbg='#F1F1F1')
```

```
meanM2 = 500*nCA2.mu #m2.pi[0]
sigmaM2 = nCA2.sigma #m2.pi[1]
xM2 = np.linspace(-10000,10000,200)
plt.plot(xM2,mlab.normpdf(xM2,meanM2,sigmaM2),linewidth=1,color='#00AAFF')
#AZUL MARINO
subplot(111,axisbg='#F1F1F1')
meanM3 = 500*nCA3.mu #m2.pi[0]
sigmaM3 = nCA3.sigma #m2.pi[1]
xM3 = np.linspace(-10000,10000,200)
plt.plot(xM3,mlab.normpdf(xM3,meanM3,sigmaM3),linewidth=1,color='#0101AA')
subplot(111,axisbg='#F1F1F1')
meanM4 = 500*nCA4.mu #m2.pi[0]
sigmaM4 = nCA4.sigma #m2.pi[1]
xM4 = np.linspace(-10000,10000,200)
plt.plot(xM4,mlab.normpdf(xM4,meanM4,sigmaM4),linewidth=1,color='#808080')
subplot(111,axisbg='#F1F1F1')
meanM5 = 500*nCA5.mu #m2.pi[0]
sigmaM5 = nCA5.sigma #m2.pi[1]
xM5 = np.linspace(-10000,10000,200)
plt.plot(xM5,mlab.normpdf(xM5,meanM5,sigmaM5),linewidth=1,color='#FF69B4')
subplot(111,axisbg='#F1F1F1')
meanM6 = 500*nCA6.mu #m2.pi[0]
sigmaM6 = nCA6.sigma #m2.pi[1]
xM6 = np.linspace(-10000,10000,200)
plt.plot(xM6,mlab.normpdf(xM6,meanM6,sigmaM6),linewidth=1,color='#32CD32')
plt.grid()
plt.show()

###################_Coeficientes de Detalle_#####################
# ROJO EMI
plt.figure(2)
subplot(111,axisbg='#F1F1F1')
plt.title('Coeficientes de Detalle')
meanM = 600*nCD.mu #m.pi[0]
sigmaM = nCD.sigma #m.pi[1]
xM = np.linspace(-600,600,200)
plt.plot(xM,mlab.normpdf(xM,meanM,sigmaM),linewidth=1, color='#FF002A')
# AZUL SE?AL 2
subplot(111,axisbg='#F1F1F1')
meanM2 = 600*nCD2.mu #m2.pi[0]
sigmaM2 = nCD2.sigma #m2.pi[1]
xM2 = np.linspace(-600,600,200)
plt.plot(xM2,mlab.normpdf(xM2,meanM2,sigmaM2),linewidth=1,color='#00AAFF')
#AZUL MARINO
subplot(111,axisbg='#F1F1F1')
meanM3 = 300*nCD3.mu #m2.pi[0]
sigmaM3 = nCD3.sigma #m2.pi[1]
```

```
xM3 = np.linspace(-600,600,200)
plt.plot(xM3,mlab.normpdf(xM3,meanM3,sigmaM3),linewidth=1,color='#0101AA')
subplot(111,axisbg='#F1F1F1')
meanM4 = 300*nCD4.mu #m2.pi[0]
sigmaM4 = nCD4.sigma #m2.pi[1]
xM4 = np.linspace(-600,600,200)
plt.plot(xM4,mlab.normpdf(xM4,meanM4,sigmaM4),linewidth=1,color='#808080')
subplot(111,axisbg='#F1F1F1')
meanM5 = 300*nCD5.mu #m2.pi[0]
sigmaM5 = nCD5.sigma #m2.pi[1]
xM5 = np.linspace(-600,600,200)
plt.plot(xM5,mlab.normpdf(xM5,meanM5,sigmaM5),linewidth=1,color='#FF69B4')
subplot(111,axisbg='#F1F1F1')
meanM6 = 300*nCD6.mu #m2.pi[0]
sigmaM6 = nCD6.sigma #m2.pi[1]
xM6 = np.linspace(-600,600,200)
plt.plot(xM6,mlab.normpdf(xM6,meanM6,sigmaM6),linewidth=1,color='#32CD32')
plt.grid()
plt.show()
```

DependenceTree_EXAMPLE.py

```
#-------------------------------------------------------------------------------
# Name:        module1
# Purpose:
#
# Author:      Emi
#
# Created:     23/03/2015
# Copyright:   (c) Emi 2015
# Licence:     <your licence>
#-------------------------------------------------------------------------------
import matplotlib
import matplotlib as mpl
import matplotlib.pyplot as plt
import pylab
from pylab import*
import dateutil
import pyparsing
import six
import sys
import os
import PIL
import random
```

```
import wave
#
import numpy
from numpy import*
from numpy import loadtxt
import numpy as np
import scipy
from scipy.io import wavfile
import scipy.misc
import pywt
import pymix
from pymix import mixture
##############################################################
#_________________LECTURA AUDIO_COEFICIENTES WAVELET_________
#______________Coef. de aproximaci?n y de detalle___________
fs, dataarray = wavfile.read('AnitaEmi.wav')            #57344 samples
pywt.dwt(dataarray, 'db10', 'sym')
(cA, cD) = pywt.dwt(dataarray, 'haar', mode ='sym')

fs, dataarray2 = wavfile.read('AnitaNat.wav')           #59392 samples
pywt.dwt(dataarray2, 'db10', 'sym')
(cA2, cD2) = pywt.dwt(dataarray2, 'haar', mode='sym')

fs, dataarray3 = wavfile.read('AnitaMr.wav')
pywt.dwt(dataarray3, 'haar', 'sym')
(cA3, cD3) = pywt.dwt(dataarray3, 'haar', mode='sym')

fs, dataarray4 = wavfile.read('AnitaFabby.wav')
pywt.dwt(dataarray4, 'haar', 'sym')
(cA4, cD4) = pywt.dwt(dataarray4, 'haar', mode='sym')

fs, dataarray5 = wavfile.read('AnitaMama.wav')
pywt.dwt(dataarray5, 'haar', 'sym')
(cA5, cD5) = pywt.dwt(dataarray5, 'haar', mode='sym')

fs, dataarray6 = wavfile.read('AnitaPapa.wav')
pywt.dwt(dataarray6, 'haar', 'sym')
(cA6, cD6) = pywt.dwt(dataarray6, 'haar', mode='sym')

################################################################################

#GRAFICACIÓN SEÑAL VOZ
"""spf = wave.open('AnitaEmi.wav','r')
subplot(211, axisbg='darkslategray')
#Extract RAW audio from WAV file
signal1 = spf.readframes(-1)
signal1 = np.fromstring(signal1,'Int16')
#IF Stereo
if spf.getnchannels() == 2:
    print ('Just mono files')
    sys.exit(0)
```

```
plt.figure(1)
plt.subplot(2,1,1)
subplot(211, axisbg='darkslategray')
plt.title('SIGNAL KEY-USER')
plt.ylabel("Amplitude")
plt.xlabel("Time")
plt.grid()
plt.plot(signal1,color='#FFFF00')
plt.draw()
spf = wave.open('AnitaNat.wav','r')
subplot(212, axisbg='darkslategray')
#Extract RAW audio from WAV file
signal2 = spf.readframes(-1)
signal2 = np.fromstring(signal2,'Int16')
#IF Stereo
if spf.getnchannels() == 2:
    print ('Just mono files')
    sys.exit(0)
plt.figure(1)
plt.subplot(2,1,2)
plt.title('SIGNAL COMPARISON-USER')
subplot(212, axisbg='darkslategray')
plt.ylabel("Amplitude")
plt.xlabel("Time")
plt.grid()
plt.plot(signal2,color='#FFFF00')
plt.draw()
"""
#################### DEPENDENCY TREES ###########################

#ARBOL 1
tree = {}
tree[0] = -1    #-1 denota ra?z
tree[1] = 0     # 0 denota siguiente rama, 1 la siguiente si existiera otro ?rbol.
n1 = mixture.ConditionalGaussDistribution(2,[cA.mean(), cD.mean()],[0,
5],[cA.std(),cD.std()],tree)

#ARBOL 2
tree2 = {}
tree2[0] = -1
tree2[1] = 0
#n2 = mixture.ProductDistribution([mixture.ConditionalGaussDistribution(2,[cA2.mean(),
cD2.mean()],[0.5, 0.5],[cA2.std(),cD2.std()],tree2)])
n2 = mixture.ConditionalGaussDistribution(2,[cA2.mean(), cD2.mean()],[0,
5],[cA2.std(),cD2.std()],tree2)

#ARBOL 3
tree3 = {}
tree3[0] = -1    #-1 denota ra?z
tree3[1] = 0     # 0 denota siguiente rama, 1 la siguiente si existiera otro ?rbol.
```

```
n3 = mixture.ConditionalGaussDistribution(2,[cA3.mean(), cD3.mean()],[0,
5],[cA3.std(),cD3.std()],tree3)

#ARBOL 4
tree4 = {}
tree4[0] = -1    #-1 denota ra?z
tree4[1] = 0     # 0 denota siguiente rama, 1 la siguiente si existiera otro ?rbol.
n4 = mixture.ConditionalGaussDistribution(2,[cA4.mean(), cD4.mean()],[0,
5],[cA4.std(),cD4.std()],tree4)

#ARBOL 5
tree5 = {}
tree5[0] = -1    #-1 denota ra?z
tree5[1] = 0     # 0 denota siguiente rama, 1 la siguiente si existiera otro ?rbol.
n5 = mixture.ConditionalGaussDistribution(2,[cA5.mean(), cD5.mean()],[0,
5],[cA5.std(),cD5.std()],tree5)

#ARBOL 6
tree6 = {}
tree6[0] = -1    #-1 denota ra?z
tree6[1] = 0     # 0 denota siguiente rama, 1 la siguiente si existiera otro ?rbol.
n6 = mixture.ConditionalGaussDistribution(2,[cA6.mean(), cD6.mean()],[0,
5],[cA6.std(),cD6.std()],tree6)

############################## CA ###################################

# ROJO EMI
plt.figure()
subplot(111,axisbg='#F1F1F1')
plt.title('Coeficientes de Aproximaci?n')
meanM = n1.mu[0] #Media CA emilio
sigmaM = n1.sigma[0] #sigma CA emilio pares /8
xM = np.linspace(-8000,8000,200)
plt.plot(xM,mlab.normpdf(xM,meanM,sigmaM),linewidth=1, color='#FF002A')

# AZUL SE?AL 2
subplot(111,axisbg='#F1F1F1')
meanM2 = n2.mu[0] #media CA Nat
sigmaM2 = n2.sigma[0] #sigma CA Nat
xM2 = np.linspace(-8000,8000,200)
plt.plot(xM2,mlab.normpdf(xM2,meanM2,sigmaM2),linewidth=1,color='#00AAFF')

#AZUL MARINO MR
subplot(111,axisbg='#F1F1F1')
meanM3 = n3.mu[0] #media CA Nat
sigmaM3 = n3.sigma[0] #sigma CA Nat
xM3 = np.linspace(-8000,8000,200)
plt.plot(xM3,mlab.normpdf(xM3,meanM3,sigmaM3),linewidth=1,color='#0101AA')

#GRIS FABBY
subplot(111,axisbg='#F1F1F1')
```

```
meanM4 = n4.mu[0] #media CA Nat
sigmaM4 = n4.sigma[0] #sigma CA Nat
xM4 = np.linspace(-8000,8000,200)
plt.plot(xM4,mlab.normpdf(xM4,meanM4,sigmaM4),linewidth=1,color='#808080')

#ROSA MAMA
subplot(111,axisbg='#F1F1F1')
meanM5 = n5.mu[0] #media CA Nat
sigmaM5 = n5.sigma[0] #sigma CA Nat
xM5 = np.linspace(-8000,8000,200)
plt.plot(xM5,mlab.normpdf(xM5,meanM5,sigmaM5),linewidth=1,color='#FF69B4')

#VERDE PAPA
subplot(111,axisbg='#F1F1F1')
meanM6 = n6.mu[0] #media CA Nat
sigmaM6 = n6.sigma[0] #sigma CA Nat
xM6 = np.linspace(-8000,8000,200)
plt.plot(xM6,mlab.normpdf(xM6,meanM6,sigmaM6),linewidth=1,color='#32CD32')

plt.grid()
plt.show()

#############################  CD  #######################################

# ROJO EMI
plt.figure()
subplot(111,axisbg='#F1F1F1')
plt.title('Coeficientes de Detalle')
meanM = n1.mu[1] #Media CD emilio
sigmaM = n1.sigma[1] #sigma CD emilio
xM = np.linspace(-8000,8000,200)
plt.plot(xM,mlab.normpdf(xM,meanM,sigmaM),linewidth=1, color='#FF002A')

# AZUL NAT
subplot(111,axisbg='#F1F1F1')
meanM2 = n2.mu[1] #media CD Nat
sigmaM2 = n2.sigma[1] #sigma CD Nat
xM2 = np.linspace(-8000,8000,200)
plt.plot(xM2,mlab.normpdf(xM2,meanM2,sigmaM2),linewidth=1,color='#00AAFF')

#AZUL MARINO MR
subplot(111,axisbg='#F1F1F1')
meanM3 = n3.mu[1] #Media CD emilio
sigmaM3 = n3.sigma[1] #sigma CD emilio
xM3 = np.linspace(-8000,8000,200)
plt.plot(xM3,mlab.normpdf(xM3,meanM3,sigmaM3),linewidth=1,color='#0101AA')

#GRIS FABBY
subplot(111,axisbg='#F1F1F1')
meanM4 = n4.mu[1] #Media CD emilio
sigmaM4 = n4.sigma[1] #sigma CD emilio
```

```
xM4 = np.linspace(-8000,8000,200)
plt.plot(xM4,mlab.normpdf(xM4,meanM4,sigmaM4),linewidth=1,color='#808080')

#ROSA MAMA
subplot(111,axisbg='#F1F1F1')
meanM5 = n5.mu[1] #Media CD emilio
sigmaM5 = n5.sigma[1] #sigma CD emilio
xM5 = np.linspace(-8000,8000,200)
plt.plot(xM5,mlab.normpdf(xM5,meanM5,sigmaM5),linewidth=1,color='#FF69B4')

#VERDE PAPA
subplot(111,axisbg='#F1F1F1')
meanM6 = n6.mu[1] #media CA Nat
sigmaM6 = n6.sigma[1] #sigma CA Nat
xM6 = np.linspace(-8000,8000,200)
plt.plot(xM6,mlab.normpdf(xM6,meanM6,sigmaM6),linewidth=1,color='#32CD32')

plt.grid()
plt.show()

############################### E. M. ####################################

#Expectation Maximization
m = mixture.MixtureModel(6,[0.4,0.2,0.1,0.1,0.1,0.1],[n1,n2,n3,n4,n5,n6])
#random.seed(1)
data = m.sampleDataSet(550)
m.modelInitialization(data)
m.EM(data,40,0.01)
print m

"""train = mixture.MixtureModel(2,[0.5,0.5],[n1,n2])
train.modelInitialization(data)
train.EM(data,40,0.01)
print m
print train
"""
#Clustering
clust = m.classify(data)

############################# CA ####################################
# ROJO EMI
plt.figure()
subplot(111,axisbg='#F1F1F1')
plt.title('Coeficientes de Aproximaci?n')
meanM = n1.mu[0] #Media CA emilio
sigmaM = n1.sigma[0] #sigma CA emilio pares /8
xM = np.linspace(-8000,8000,200)
plt.plot(xM,mlab.normpdf(xM,meanM,sigmaM),linewidth=1, color='#FF002A')
# AZUL SE?AL 2
subplot(111,axisbg='#F1F1F1')
meanM2 = n2.mu[0] #media CA Nat
```

```
sigmaM2 = n2.sigma[0] #sigma CA Nat
xM2 = np.linspace(-8000,8000,200)
plt.plot(xM2,mlab.normpdf(xM2,meanM2,sigmaM2),linewidth=1,color='#00AAFF')
#AZUL MARINO MR
subplot(111,axisbg='#F1F1F1')
meanM3 = n3.mu[0] #media CA Nat
sigmaM3 = n3.sigma[0] #sigma CA Nat
xM3 = np.linspace(-8000,8000,200)
plt.plot(xM3,mlab.normpdf(xM3,meanM3,sigmaM3),linewidth=1,color='#0101AA')
#GRIS FABBY
subplot(111,axisbg='#F1F1F1')
meanM4 = n4.mu[0] #media CA Nat
sigmaM4 = n4.sigma[0] #sigma CA Nat
xM4 = np.linspace(-8000,8000,200)
plt.plot(xM4,mlab.normpdf(xM4,meanM4,sigmaM4),linewidth=1,color='#808080')
#ROSA MAMA
subplot(111,axisbg='#F1F1F1')
meanM5 = n5.mu[0] #media CA Nat
sigmaM5 = n5.sigma[0] #sigma CA Nat
xM5 = np.linspace(-8000,8000,200)
plt.plot(xM5,mlab.normpdf(xM5,meanM5,sigmaM5),linewidth=1,color='#FF69B4')
#VERDE PAPA
subplot(111,axisbg='#F1F1F1')
meanM6 = n6.mu[0] #media CA Nat
sigmaM6 = n6.sigma[0] #sigma CA Nat
xM6 = np.linspace(-8000,8000,200)
plt.plot(xM6,mlab.normpdf(xM6,meanM6,sigmaM6),linewidth=1,color='#32CD32')

plt.grid()
plt.show()
##############################  CD  ##########################################
# ROJO EMI
plt.figure()
subplot(111,axisbg='#F1F1F1')
plt.title('Coeficientes de Detalle')
meanM = n1.mu[1] #Media CD emilio
sigmaM = n1.sigma[1] #sigma CD emilio
xM = np.linspace(-10000,10000,200)
plt.plot(xM,mlab.normpdf(xM,meanM,sigmaM),linewidth=1, color='#FF002A')
# AZUL SE?AL 2
subplot(111,axisbg='#F1F1F1')
meanM2 = n2.mu[1] #media CD Nat
sigmaM2 = n2.sigma[1] #sigma CD Nat
xM2 = np.linspace(-10000,10000,200)
plt.plot(xM2,mlab.normpdf(xM2,meanM2,sigmaM2),linewidth=1,color='#00AAFF')
#AZUL MARINO MR
subplot(111,axisbg='#F1F1F1')
meanM3 = n3.mu[1] #Media CD emilio
sigmaM3 = n3.sigma[1] #sigma CD emilio
xM3 = np.linspace(-8000,8000,200)
plt.plot(xM3,mlab.normpdf(xM3,meanM3,sigmaM3),linewidth=1,color='#0101AA')
```

```
#GRIS FABBY
subplot(111,axisbg='#F1F1F1')
meanM4 = n4.mu[1] #Media CD emilio
sigmaM4 = n4.sigma[1] #sigma CD emilio
xM4 = np.linspace(-8000,8000,200)
plt.plot(xM4,mlab.normpdf(xM4,meanM4,sigmaM4),linewidth=1,color='#808080')
#ROSA MAMA
subplot(111,axisbg='#F1F1F1')
meanM5 = n5.mu[1] #Media CD emilio
sigmaM5 = n5.sigma[1] #sigma CD emilio
xM5 = np.linspace(-8000,8000,200)
plt.plot(xM5,mlab.normpdf(xM5,meanM5,sigmaM5),linewidth=1,color='#FF69B4')
#VERDE PAPA
subplot(111,axisbg='#F1F1F1')
meanM6 = n6.mu[1] #media CA Nat
sigmaM6 = n6.sigma[1] #sigma CA Nat
xM6 = np.linspace(-8000,8000,200)
plt.plot(xM6,mlab.normpdf(xM6,meanM6,sigmaM6),linewidth=1,color='#32CD32')
plt.grid()
plt.show()
math.plotlib
```