ELM

May 9, 2022

1 Redes Neurais Artificiais

Nome: João Pedro Miranda Marques

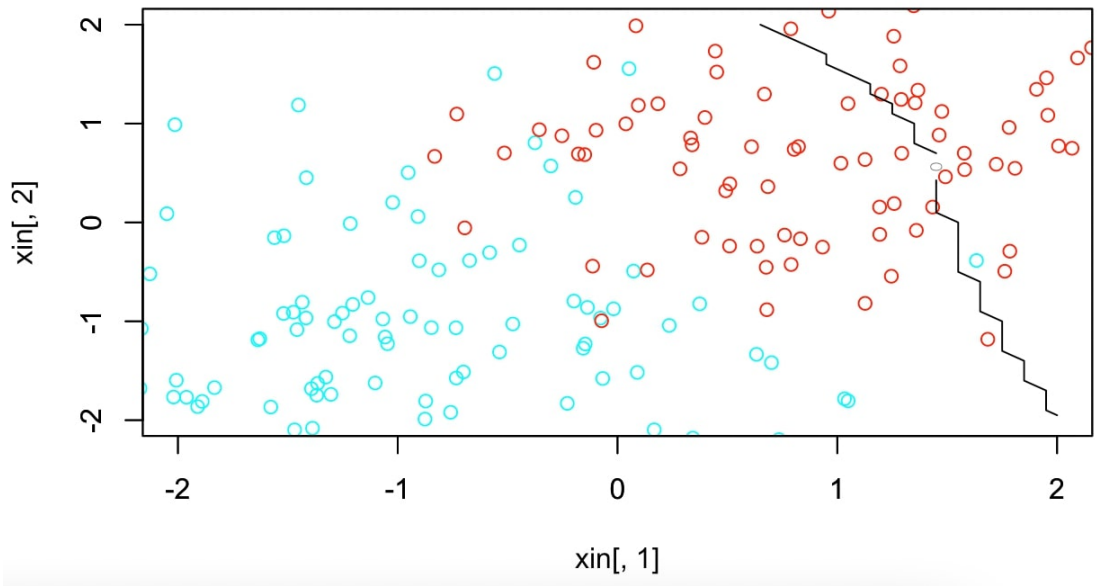
Matrícula: 2017050495

2 Extreme Learning Machine - ELM

As imagens abaixo foram criadas utilizando o codigo em R dado pelo professor para as funcoes de trainELM e YELM.

[ ]:
```
from IPython.display import Image
# 5 neuronios 2D
Image(filename='pics/pic1.jpg',width=800, height=400)
```

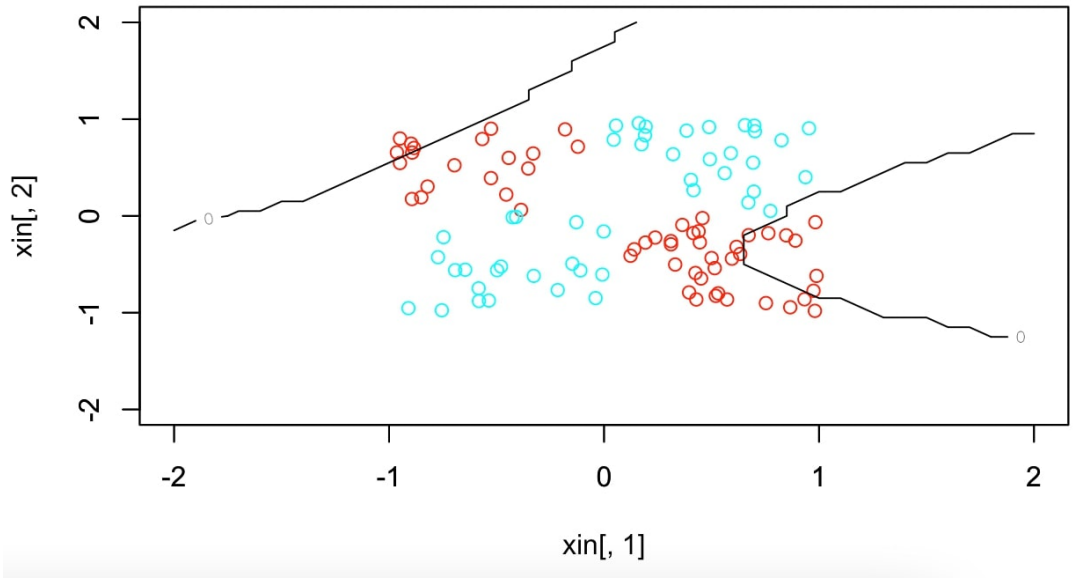
[ ]:



Classificação com 5 neuronios nesse problema ficou com baixa acurácia.

[ ]:
```
# 5 neuronios XOR
Image(filename='pics/pic2.jpg',width=800, height=400)
```

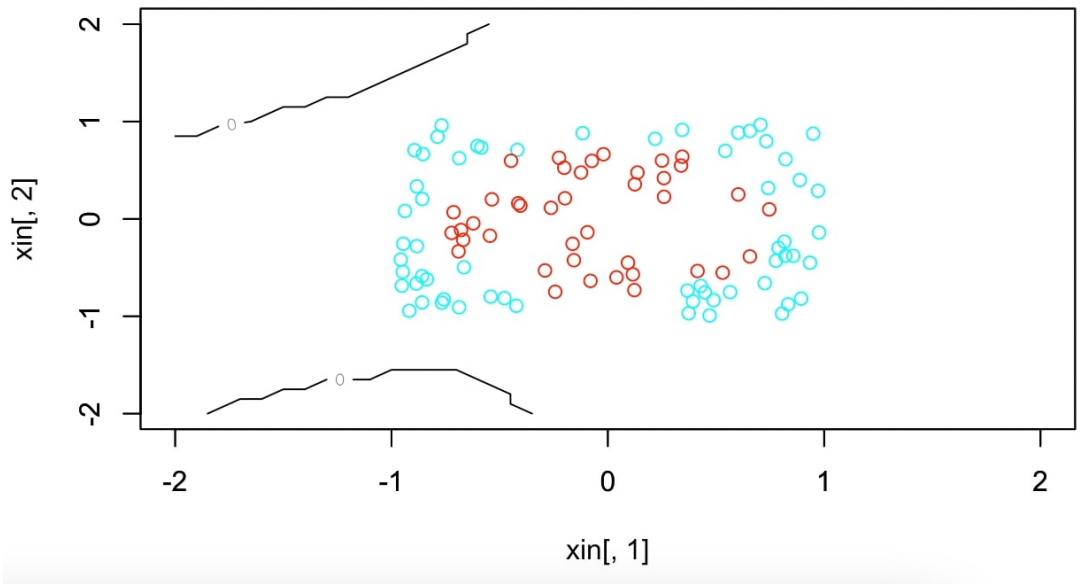
[ ]:



Classificação com 5 neuronios nesse problema ficou com baixa acurácia.

[ ]:
```
# 5 neuronios circles
Image(filename='pics/pic3.jpg',width=800, height=400)
```

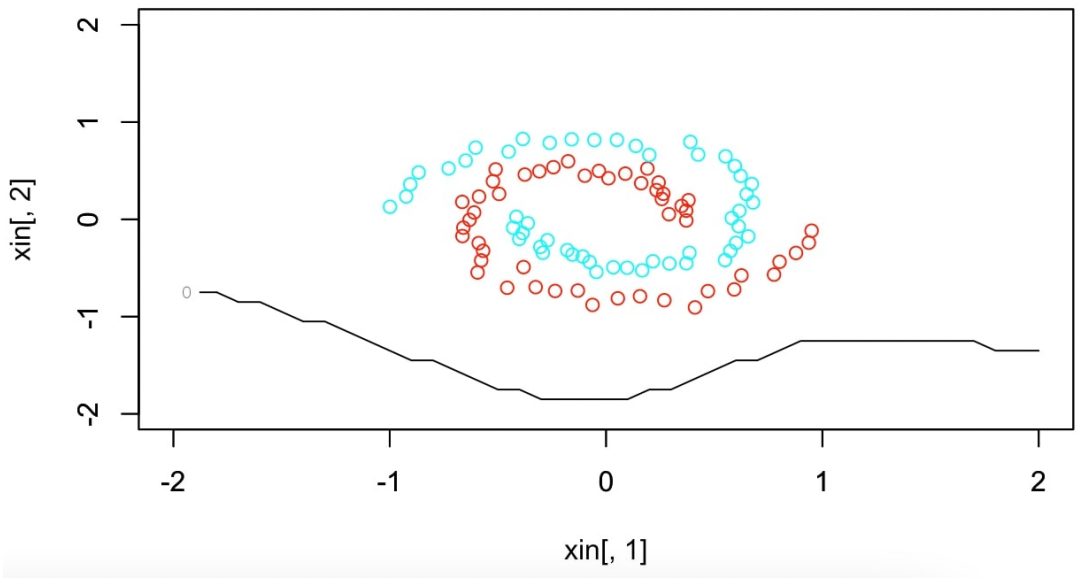
[ ]:



Classificação com 5 neuronios nesse problema ficou com baixa acurácia.

[ ]:
```
# 5 neuronios spirals
Image(filename='pics/pic4.jpg',width=800, height=400)
```

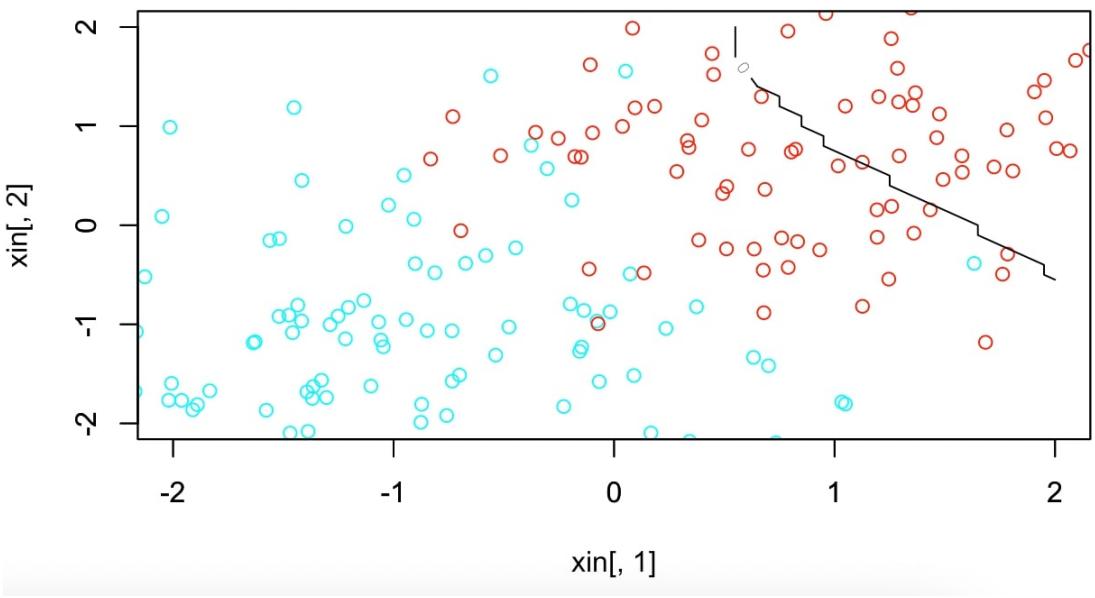
[ ]:



Classificação com 5 neuronios nesse problema ficou com baixa acurácia.

[ ]:
```
# 10 neuronios 2D
Image(filename='pics/pic5.jpg',width=800, height=400)
```

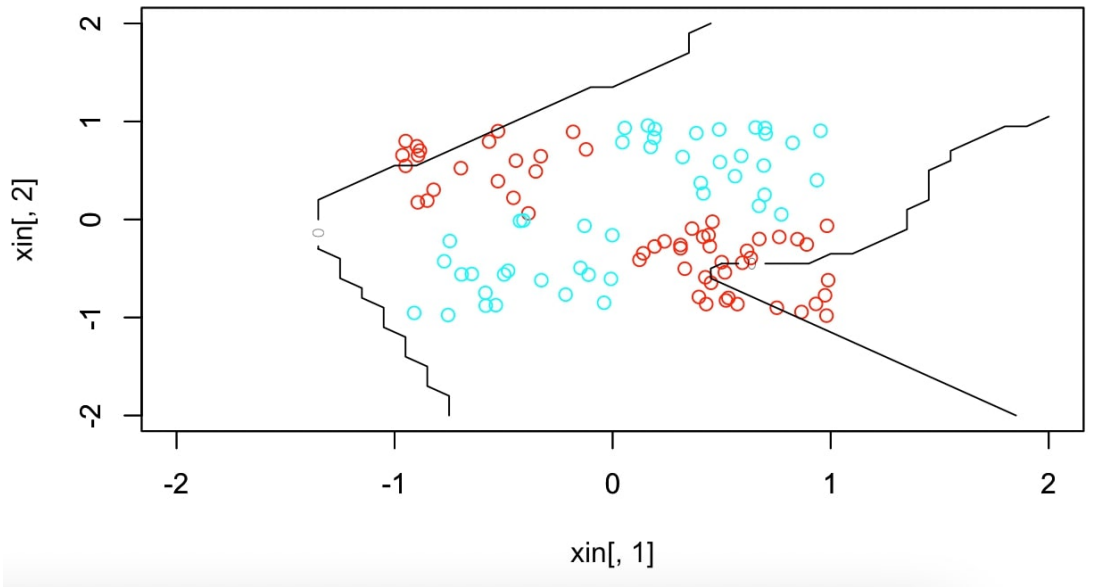
[ ]:



Classificação com 10 neuronios nesse problema ficou com baixa acurácia.

[ ]:
```
# 10 neuronios XOR
Image(filename='pics/pic6.jpg',width=800, height=400)
```

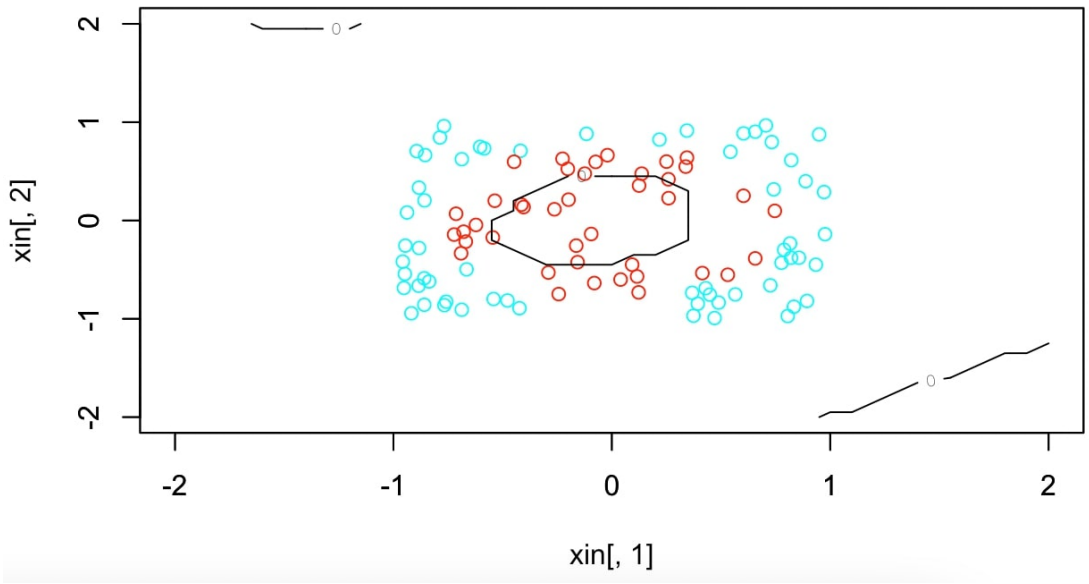
[ ]:



Classificação com 10 neuronios nesse problema ficou com baixa acurácia.

[ ]:
```
# 10 neuronios circles
Image(filename='pics/pic7.jpg',width=800, height=400)
```

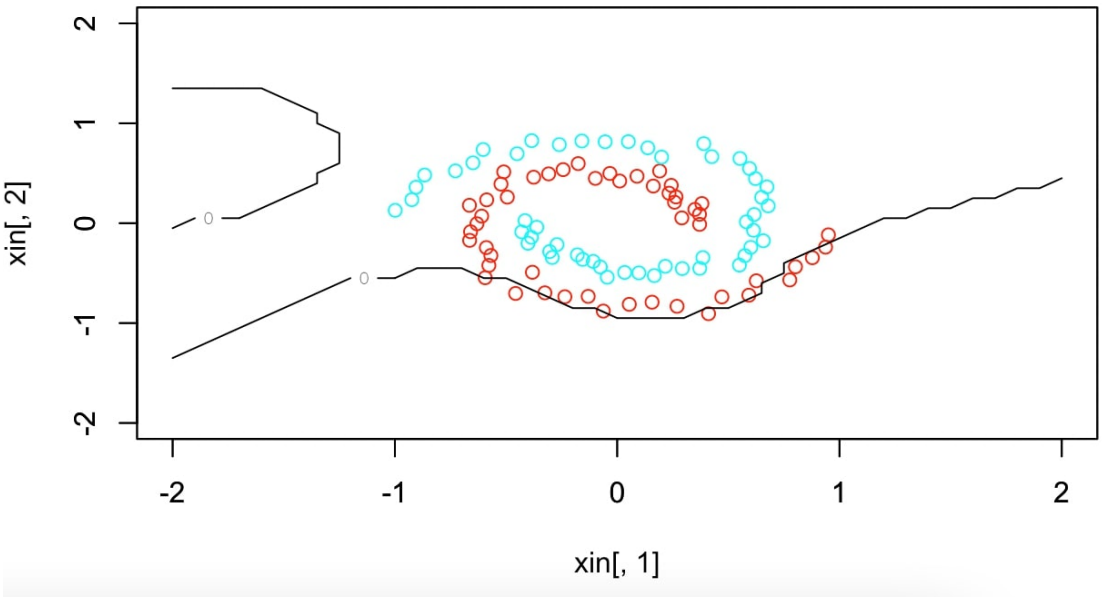
[ ]:



Classificação com 10 neuronios nesse problema ficou com baixa acurácia.

[ ]:
```
# 10 neuronios spirals
Image(filename='pics/pic8.jpg',width=800, height=400)
```

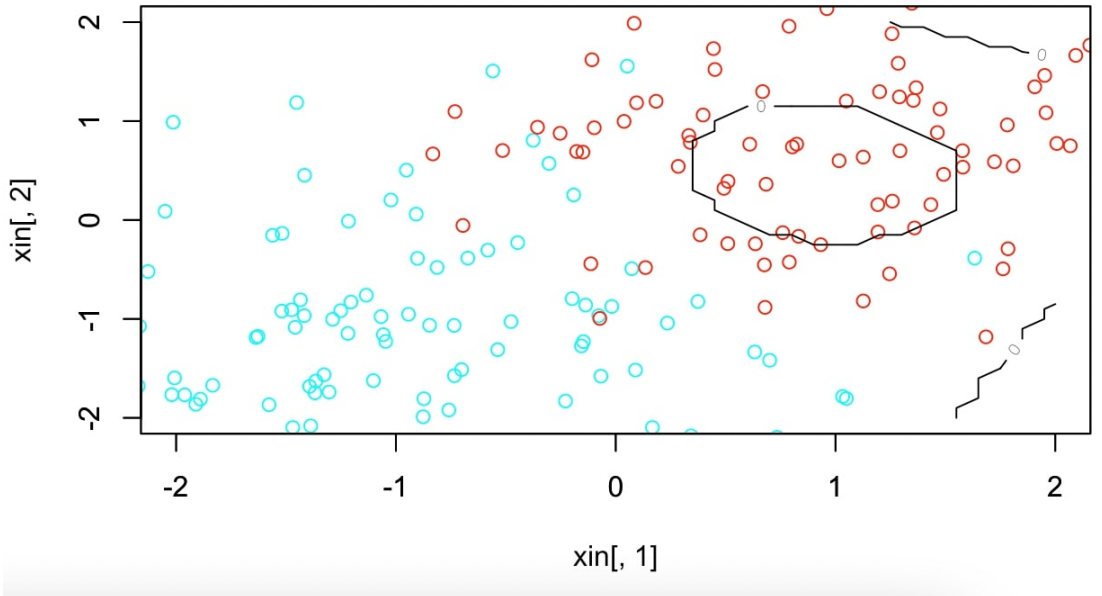
[ ]:



Classificação com 10 neuronios nesse problema ficou com baixa acurácia.

[ ]:
```
# 30 Neuronios 2D
Image(filename='pics/pic9.jpg',width=800, height=400)
```

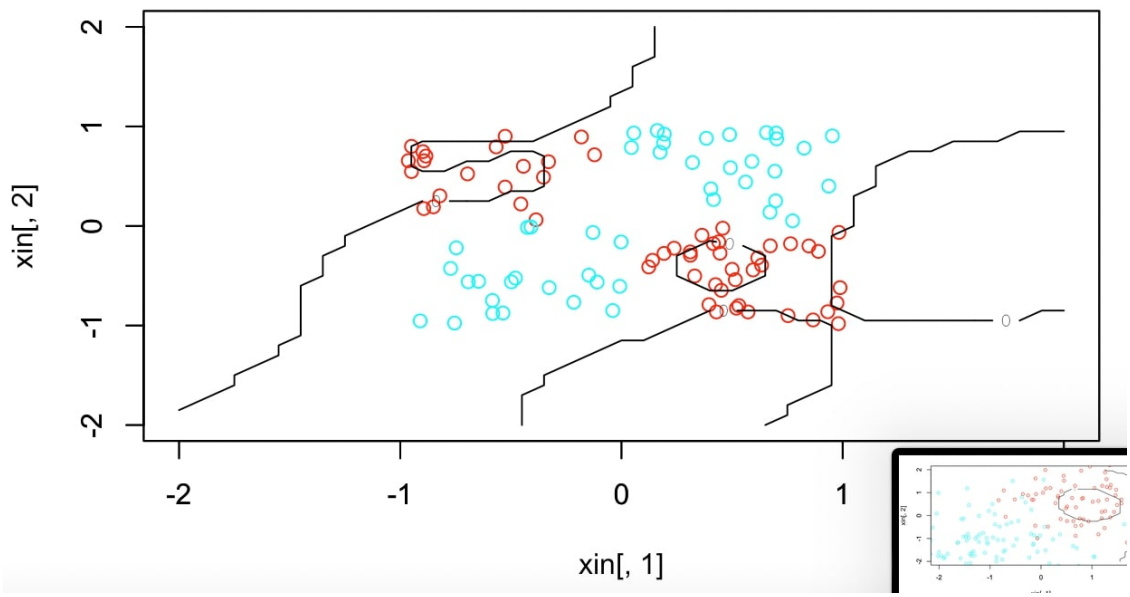
[ ]:



Classificação com 30 neuronios nesse problema ficou com overfitting.

[ ]:
```
# 30 Neuronios XOR
Image(filename='pics/pic10.jpg',width=800, height=400)
```

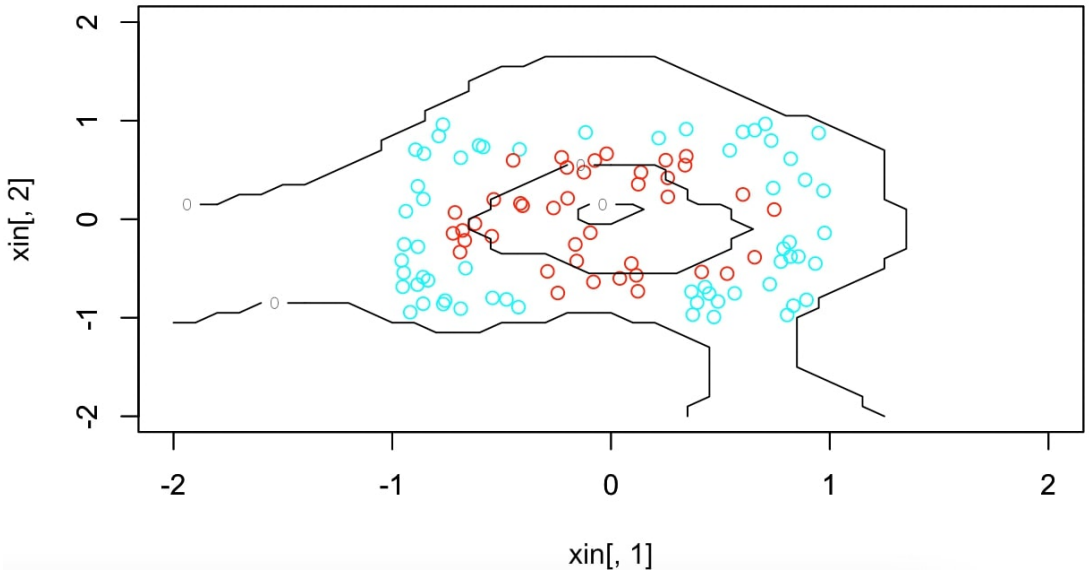
[ ]:



Classificação com 30 neuronios nesse problema ficou com acurácia aceitável.

[ ]:
```
# 30 Neuronios circles
Image(filename='pics/pic11.jpg',width=800, height=400)
```

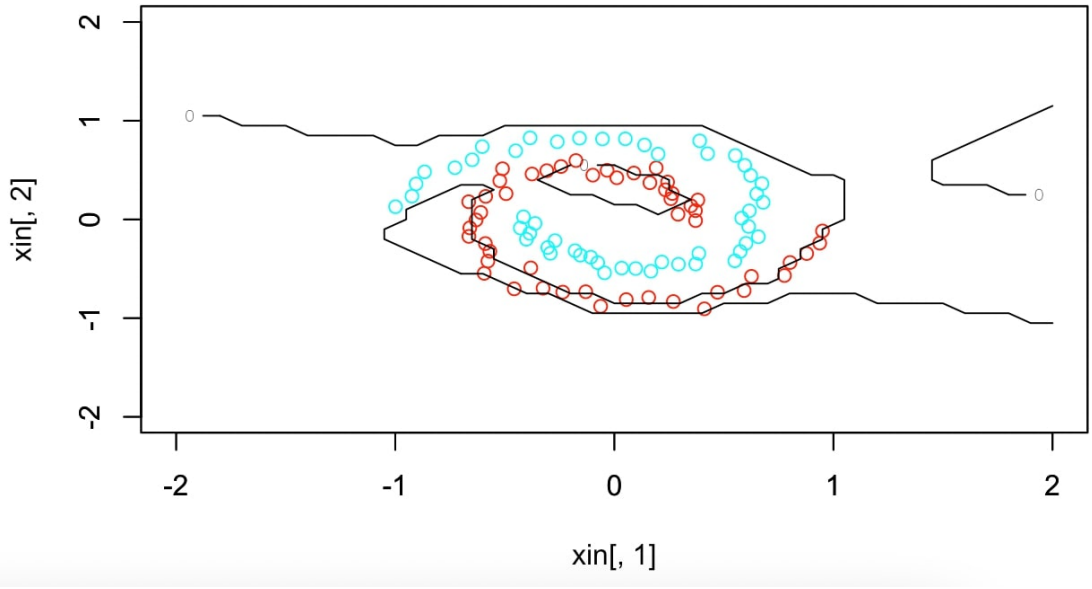
[ ]:



Classificação com 30 neuronios nesse problema ficou com acurácia aceitável.

[ ]:
```
# 30 Neuronios spirals
Image(filename='pics/pic12.jpg',width=800, height=400)
```

[ ]:



Classificação com 30 neuronios nesse problema ficou com boa acurácia.

Em Python, apesar das funções trainELM() e YELM() funcionarem corretamente, tive dificuldade na etapa de plotar no contour. Por isso os resultados graficos nao ficaram bons.

[ ]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

[ ]:
```
# Treinamento de uma rede ELM
def trainELM(xin, yin, nNeurons, par):
    xin = pd.DataFrame(xin)
    yin = pd.DataFrame(yin)

    nDimension = xin.shape[1]      # Dimensao de entrada.

    # Adiciona ou não um termo de polarização ao vetor de treinamento w.
    if par == 1:
        xin.insert(nDimension, nDimension, 1)
        # Z<-replicate(p, runif((n+1),-0.5,0.5))
        Z = [np.random.uniform(low=-0.5, high=0.5, size=nDimension+1) for _ in␣
↪range(nNeurons)]
```

```
    else:
        Z = [np.random.uniform(low=-0.5, high=0.5, size=nDimension) for _ in␣
↪range(nNeurons)]

    Z = pd.DataFrame(Z)
    Z = Z.T

    H = np.tanh(xin @ Z)

    W = ( np.linalg.pinv(H) @ yin)     #W<-pseudoinverse(H) %*% yin

    return [W,H,Z]
```

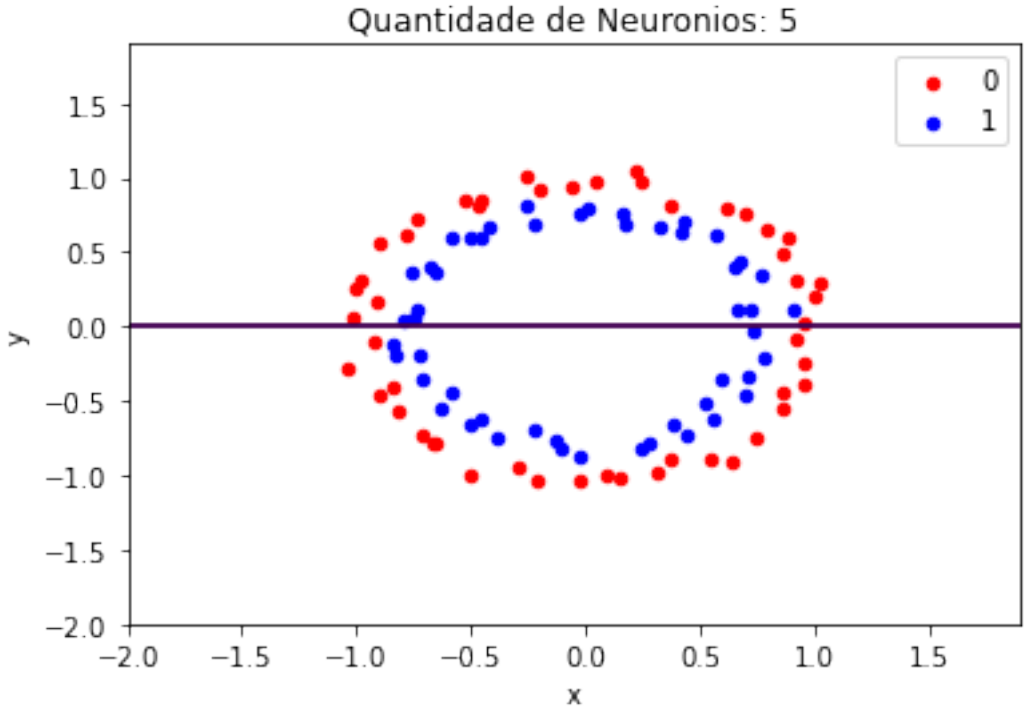
[ ]:
```
# Saída de uma rede ELM
def YELM(xin, Z, W, par):

    xin = pd.DataFrame(xin)
    Z = pd.DataFrame(Z)
    W = pd.DataFrame(W)

    nDimension = xin.shape[1]  # Dimensao de entrada.

    # Adiciona ou não termo de polarização
    if(par == 1):
        xin.insert(nDimension, nDimension, 1)
        # np.c_[ xin, np.ones(xin.shape[0]) ]

    # print("xin:", xin.shape)
    # print("Z:", Z.shape)
    H = np.tanh(xin @ Z)
    # print("H:", H.shape)
    # print("W:", W.shape)
    Yhat = np.sign(H @ W)

    return Yhat
```

[ ]:
```
from sklearn.datasets import make_circles, make_moons
from matplotlib import pyplot
from pandas import DataFrame

def plotContour(format, nNeurons):
    if(format == 1):
        X, y = make_circles(n_samples=100, noise=0.05)
    elif(format == 2):
        X, y = make_moons(n_samples=100, noise=0.05)
    # scatter plot, dots colored by class value
    df = DataFrame(dict(x=X[:,0], y=X[:,1], label=y))
```

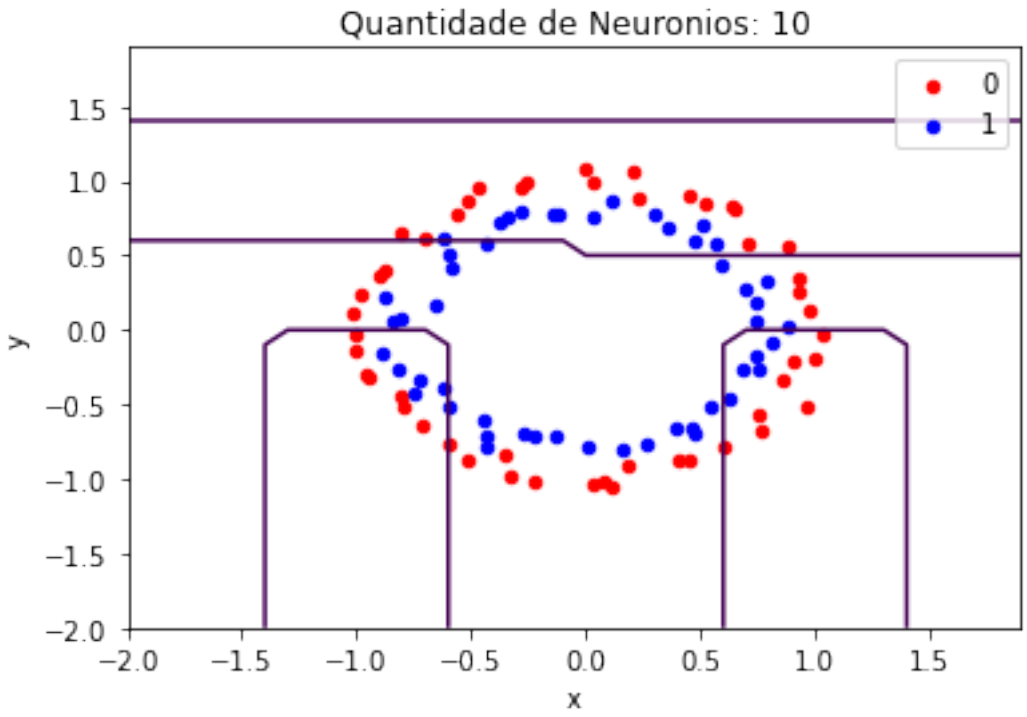
```
    colors = {0:'red', 1:'blue'}
    fig, ax = pyplot.subplots()
    grouped = df.groupby('label')
    for key, group in grouped:
        group.plot(ax=ax, kind='scatter', x='x', y='y', label=key,␣
↪color=colors[key])

    retlist = trainELM(X[:,:2], y, nNeurons, 1)

    W = retlist[0]
    H = retlist[1]
    Z = retlist[2]

    yt = YELM(X[:,:2], Z, W, 1)
    yt = pd.DataFrame(yt).to_numpy()

    #plotting contours
    seq = np.arange(-2,2,0.1)
    lseq = len(seq)

    Xtest = []
    Xtest = np.array(np.meshgrid(seq, seq))
    Xtest = np.reshape(Xtest, (lseq**2, 2))
    Ytest = YELM(Xtest, Z, W, 1)
    Ytest = pd.DataFrame(Ytest).to_numpy()
    Ytest = np.reshape(Ytest, (len(seq), len(seq)))

    shape = Ytest.shape
    result = np.zeros(shape)
    for x in range(0, shape[0]):
        for y in range(0, shape[1]):
            if Ytest[x, y] == 1:
                result[x, y] = 1

    plt.contour(seq, seq, result, 0)
    if(nNeurons == 5):
        plt.title('Quantidade de Neuronios: 5')
    elif(nNeurons == 10):
        plt.title('Quantidade de Neuronios: 10')
    elif(nNeurons == 30):
        plt.title('Quantidade de Neuronios: 30')
    plt.show()

plotContour(1,5)
plotContour(1,10)
plotContour(1,30)
plotContour(2,5)
```

```
plotContour(2,10)
plotContour(2,30)
```

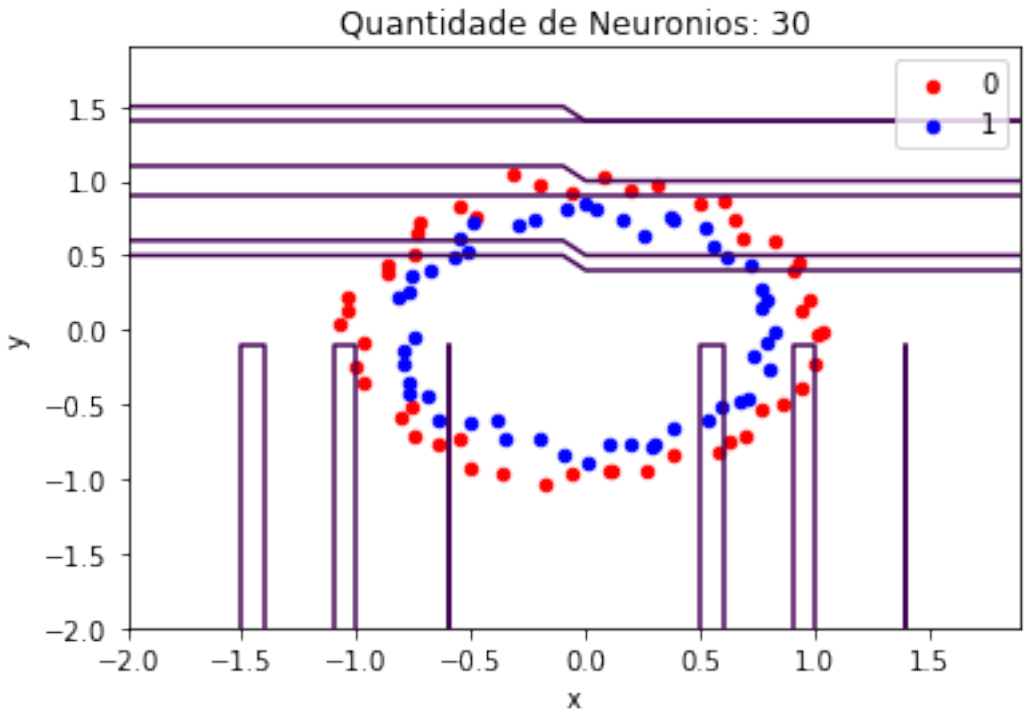
```
/var/folders/76/q41_l1rj2px1gtbm8fzg0tb00000gn/T/ipykernel_20105/3004038449.py:4
7: UserWarning: No contour levels were found within the data range.
  plt.contour(seq, seq, result, 0)
```



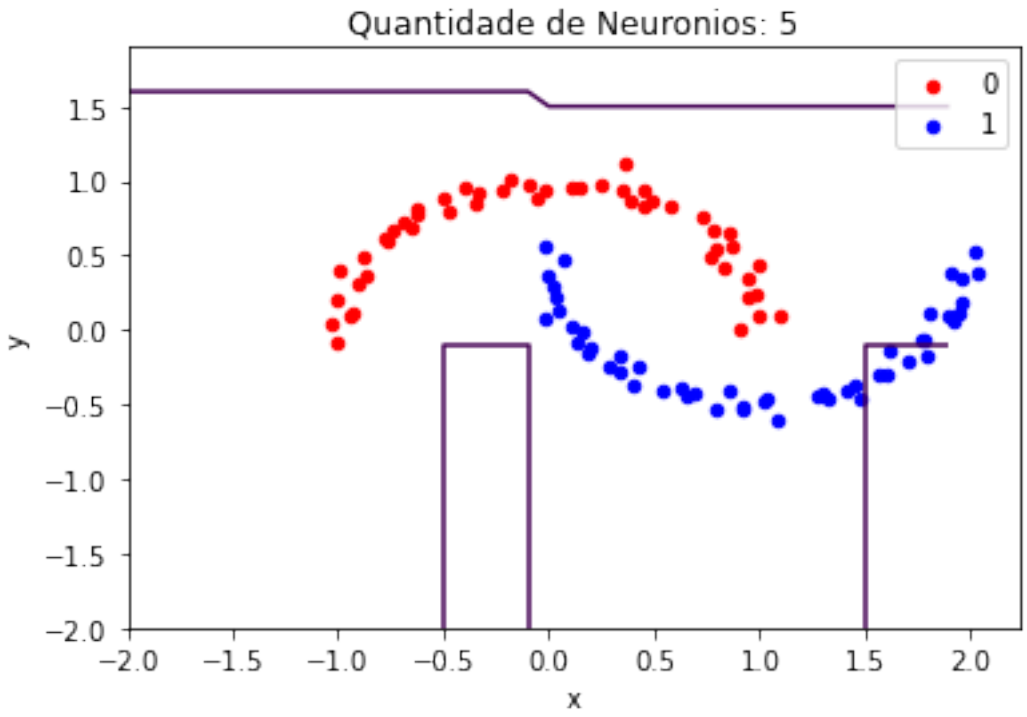

```
/var/folders/76/q41_l1rj2px1gtbm8fzg0tb00000gn/T/ipykernel_20105/3004038449.py:4
7: UserWarning: No contour levels were found within the data range.
  plt.contour(seq, seq, result, 0)
```



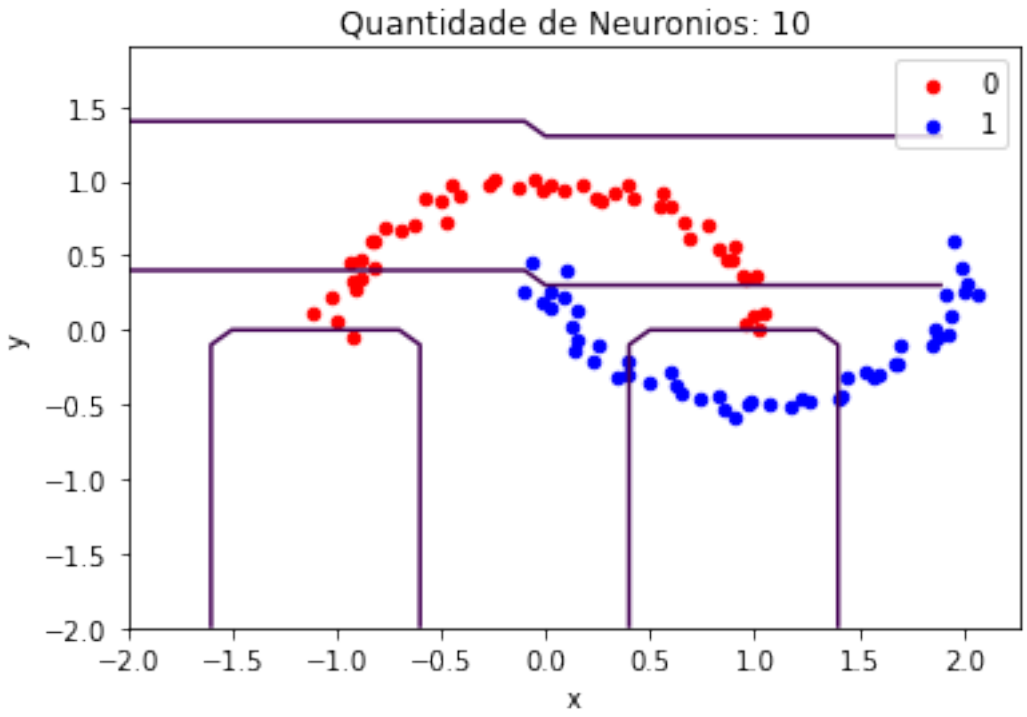

```
/var/folders/76/q41_l1rj2px1gtbm8fzg0tb00000gn/T/ipykernel_20105/3004038449.py:4
7: UserWarning: No contour levels were found within the data range.
  plt.contour(seq, seq, result, 0)
```



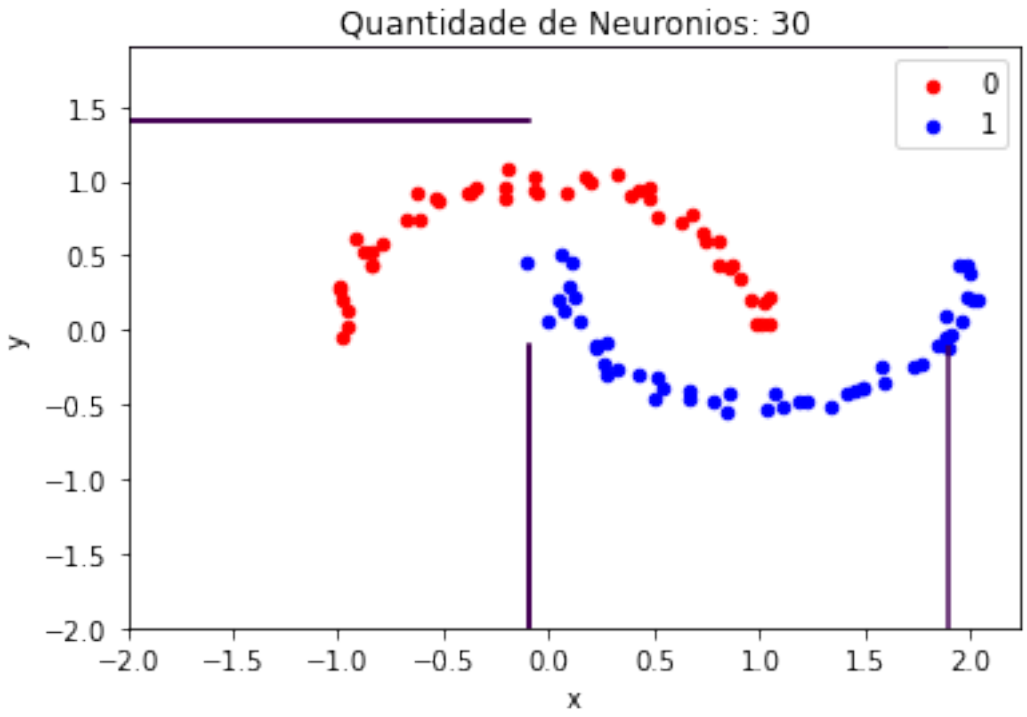

```
/var/folders/76/q41_l1rj2px1gtbm8fzg0tb00000gn/T/ipykernel_20105/3004038449.py:4
7: UserWarning: No contour levels were found within the data range.
  plt.contour(seq, seq, result, 0)
```




```
/var/folders/76/q41_l1rj2px1gtbm8fzg0tb00000gn/T/ipykernel_20105/3004038449.py:4
7: UserWarning: No contour levels were found within the data range.
  plt.contour(seq, seq, result, 0)
```




```
/var/folders/76/q41_l1rj2px1gtbm8fzg0tb00000gn/T/ipykernel_20105/3004038449.py:4
7: UserWarning: No contour levels were found within the data range.
  plt.contour(seq, seq, result, 0)
```


Quantidade de Neuronios: 30
0
1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
y
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x