Aprendizado de Máquina - MO444

Exercício 2

Aluno: Paulo Ricardo Finardi. RA: 144809

0 Preliminares

Para a solução do exercício **não foi utilizado** a função `GridsearchCV` da biblioteca `sklearn` (essa é a *única* restrição que consta no enunciado do exercício).

Foi utilizado `Python 3` na resolução dos exercícios. O código abaixo é o *header* de todos os itens.

```
import numpy as np
import pandas as pd
import csv
from sklearn import cross_validation
from sklearn.svm import SVC

# lendo os dados dada1.csv com o Pandas
data1 = pd.read_csv('data1.csv')

# excluindo a última coluna de data1
data1.set_index('clase').to_csv('data.csv', index=None)

# salvando um novo arquivo em formato csv
X = pd.read_csv('data.csv')

# convertendo os dados em numpy-array
data1 = data1.as_matrix(columns=None)
X = X.as_matrix(columns=None)

```

```
20 # y é o conjunto da classe (rótulo 0 ou 1)
21 y = data1[: , 166]
```

0.1 Estratégia

A solução deste exercício segue rigorosamente o enunciado do exercício. Com um pouco mais de detalhes, considere os comentários em cada trecho do enunciado.

Leia os dados do arquivo data1.csv A classe de cada dado é o valor da última coluna (0 ou 1).
Sem observações.

Treine um SVM com kernel RBF nos dados do arquivos.
Nesse trecho, realizamos o SVM em todo o conjunto de dados. Não existe nenhuma menção em tomar qualquer parcela para conjunto de treino/teste.

A validação externa deve ser 5-fold estratificado.
Utilizamos a função `cross_validation` da biblioteca `sklearn`. Não existe restrições ao uso dessa função.

Para cada conjunto de treino da validação externa faça um 3-fold para escolher os melhores hiperparametros para C (cost) e gamma.
Realizado laços individuais para cada conjunto de treino.

Faça um grid search de para o C nos valores 2**-5, 2**-2, 2**0, 2**2, e 2**5 e gamma nos valores 2**-15, 2**-10, 2**-5, 2**0, e 2**5
Processo realizado sem o uso da função `GridSearchCv`.

1) Qual a accuracia media (na validação de fora).
`Resposta:` Entendemos como "validação de fora" todo o conjunto dado. A acurácia média foi medida pela função ***score*** da biblioteca *sklearn*; essa função retorna a acurácia média. Realizamos o *cross-validation* estratificado com os hiperparâmetros C e *gamma* imutáveis (em *default*) e *kernel* = rbf. Nessas condições, obtemos **90.1%** de acurácia média.

2) Quais os valores de C e gamma a serem usados no classificador final (fazer o 3-fold no conjunto todo).
`Resposta:` Após o 3-fold em cada um dos 5 conjutos de treinamento (obtidos do conjunto inicial). O melhor valor para o cunjunto $\{C, gamma\}$ de hiperparâmetros é: $\{2^5, 2^{-5}\}$. Com esses valores em validação 3-fold para todo o conjunto, obtemos acurácia média **91.4%** (ganho de **1.3%** em relação a questão 1).

NÃO use funções prontas que ja fazer o grid search como GridSearchCV do sklearn ou o tune

do pacote e1070 do R. Neste exercicio eu quero que voces façam os loops explicitamente.
Não utilizado.

1 Código

O primeiro bloco de código refere-se a este trecho do enunciado: *Treine um SVM com kernel RBF nos dados do arquivos. A validação externa deve ser 5-fold estratificado.*

```
1  # criando um SVM classificador
2  svc = SVC(kernel='rbf')
3
4  # criando a validação cruzada estratificada
5  skf = cross_validation.StratifiedKFold(y, n_folds=5)
6
7  # treinando e avaliando
8  scores = [svc.fit(X[train], y[train]).score(X[test], y[test]) for train, test in skf]
9  score = [float(round(n, 3)) for n in scores]
10 print(score)
11
12 # média dan of scores
13 print('\nMean: %.3f' % np.mean(scores))
```

Respectivamente, os resultados das linhas 10 e 13:

```
[0.865, 0.885, 0.947, 0.884, 0.926]
Mean:  0.901
```

O segundo bloco de código refere-se a este trecho do enunciado: *Para cada conjunto de treino da validação externa faça um 3-fold para escolher os melhores hiperparâmetros para `C` e gamma.Faça um grid search de para o `C` nos valores 2^{-5}, 2^{-2}, 2^{0}, 2^{2}, e 2^{5} e gamma nos valores 2^{-15}, 2^{-10}, 2^{-5}, 2^{0}, e 2^{5}.*

```
1  # seleção individual de cada conjunto de treino/teste
2  mylist_skf = list(skf)
3  index = [None]*5
4  test  = [None]*5
5
6  for i in range(0,5):
```

```
7       index[i], test[i] = mylist_skf[i]
8
9   # conjuntos individuais de treino com o respectivo conjunto de classe.
10  X_15, y_15 = X[index[0]], y[index[0]] # conjunto treino/rotulo 1/5
11  X_25, y_25 = X[index[1]], y[index[1]] # conjunto treino/rotulo 2/5
12  X_35, y_35 = X[index[2]], y[index[2]] # conjunto treino/rotulo 3/5
13  X_45, y_45 = X[index[3]], y[index[3]] # conjunto treino/rotulo 4/5
14  X_55, y_55 = X[index[4]], y[index[4]] # conjunto treino/rotulo 5/5
15
16  # variação do hiperparâmetro C
17  c_range     = [np.power(2.0,i) for i in [-5, -2, 0, 2, 5]]
18
19  # variação do hiperparâmetro gamma
20  gamma_range = [np.power(2.0,i) for i in range(-15, 6, 5)]
```

Primeiro *loop para o Grid Search* conjunto 1/5

```
1   # conjunto 1 de 5
2   for c in c_range:
3       print('\n', 'Para C = %f'  % c)
4       for gamma in gamma_range:
5           print('Com gamma = %f' % gamma)
6           svc = SVC(C = c, gamma = gamma, kernel='rbf')
7           kf = cross_validation.KFold(len(X_15), n_folds=3) # 3-fold
8           scores = [ svc.fit(X_15[train], y_15[train]).score(X_15[test], y_15[test])
9                      for train, test in kf ]
10          score = [float(round(n, 3)) for n in scores]
11          print('Acurácia em 3-fold: ', score, '. Média da acurácia: %.3f' % np.mean(score))
```

O resultado da linha 11 é:

```
Para C = 0.031250
Com gamma = 0.000031
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 0.000977
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
```

```
Com gamma = 0.031250
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 1.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 32.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566

Para C = 0.250000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 0.000977
Acurácia em 3-fold:  [0.559, 0.591, 0.722] .  Média da acurácia:  0.624
Com gamma = 0.031250
Acurácia em 3-fold:  [0.567, 0.614, 0.659] .  Média da acurácia:  0.613
Com gamma = 1.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 32.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566

Para C = 1.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 0.000977
Acurácia em 3-fold:  [0.764, 0.803, 0.849] .  Média da acurácia:  0.805
Com gamma = 0.031250
Acurácia em 3-fold:  [0.882, 0.882, 0.897] .  Média da acurácia:  0.887
Com gamma = 1.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 32.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566

Para C = 4.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 0.000977
Acurácia em 3-fold:  [0.85, 0.827, 0.857] .  Média da acurácia:  0.845
Com gamma = 0.031250
Acurácia em 3-fold:  [0.89, 0.906, 0.897] .  Média da acurácia:  0.898
```

```
Com gamma = 1.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 32.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566

Para C = 32.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.787, 0.78, 0.81] .  Média da acurácia:  0.792
Com gamma = 0.000977
Acurácia em 3-fold:  [0.913, 0.898, 0.889] .  Média da acurácia:  0.900
Com gamma = 0.031250
Acurácia em 3-fold:  [0.89, 0.906, 0.897] .  Média da acurácia:  0.898
Com gamma = 1.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 32.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
```

Assim, para o conjunto (1/5) escolhemos C e *gamma* que possui o maior valor deacurácia média. Logo, $C = 32$ e $gamma = 2^{-5}$ implica em acurácia 89.8%.

```
1  # conjunto 2 de 5
2  for c in c_range:
3      print('\n', 'Para C = %f'  % c)
4      for gamma in gamma_range:
5          print('Com gamma = %f' % gamma)
6          svc = SVC(C = c, gamma = gamma, kernel='rbf')
7          kf = cross_validation.KFold(len(X_25), n_folds=3) # 3-fold
8          scores = [ svc.fit(X_25[train], y_25[train]).score(X_25[test], y_25[test])
9                     for train, test in kf ]
10         score = [float(round(n, 3)) for n in scores]
11         print('Acurácia em 3-fold: ', score, '. Média da acurácia: %.3f' % np.mean(score))
```

O resultado da linha 11 é:

```
Para C = 0.031250
Com gamma = 0.000031
```

```
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 0.000977
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 0.031250
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 1.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 32.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566

Para C = 0.250000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 0.000977
Acurácia em 3-fold:  [0.535, 0.575, 0.595] .  Média da acurácia:  0.568
Com gamma = 0.031250
Acurácia em 3-fold:  [0.567, 0.614, 0.627] .  Média da acurácia:  0.603
Com gamma = 1.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 32.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566

Para C = 1.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 0.000977
Acurácia em 3-fold:  [0.724, 0.858, 0.841] .  Média da acurácia:  0.808
Com gamma = 0.031250
Acurácia em 3-fold:  [0.858, 0.898, 0.913] .  Média da acurácia:  0.890
Com gamma = 1.000000
Acurácia em 3-fold:  [0.543, 0.583, 0.587] .  Média da acurácia:  0.571
Com gamma = 32.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566

Para C = 4.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
Com gamma = 0.000977
```

```
Acurácia em 3-fold:  [0.795, 0.85, 0.841] .  Média da acurácia:  0.829
Com gamma = 0.031250
Acurácia em 3-fold:  [0.89, 0.906, 0.897] .  Média da acurácia:  0.898
Com gamma = 1.000000
Acurácia em 3-fold:  [0.543, 0.583, 0.587] .  Média da acurácia:  0.571
Com gamma = 32.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566

Para C = 32.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.74, 0.835, 0.833] .  Média da acurácia:  0.803
Com gamma = 0.000977
Acurácia em 3-fold:  [0.898, 0.921, 0.865] .  Média da acurácia:  0.895
Com gamma = 0.031250
Acurácia em 3-fold:  [0.89, 0.906, 0.897] .  Média da acurácia:  0.898
Com gamma = 1.000000
Acurácia em 3-fold:  [0.543, 0.583, 0.587] .  Média da acurácia:  0.571
Com gamma = 32.000000
Acurácia em 3-fold:  [0.535, 0.575, 0.587] .  Média da acurácia:  0.566
```

Assim, para o conjunto (2/5) escolhemos C e *gamma* que possui o maior valor deacurácia média. Logo, $C = 32$ e $gamma = 2^{-5}$ implica em acurácia 89.8%.

```
1  # conjunto 3 de 5
2  for c in c_range:
3      print('\n', 'Para C = %f'  % c)
4      for gamma in gamma_range:
5          print('Com gamma = %f' % gamma)
6          svc = SVC(C = c, gamma = gamma, kernel='rbf')
7          kf = cross_validation.KFold(len(X_35), n_folds=3)
8          scores = [ svc.fit(X_35[train], y_35[train]).score(X_35[test], y_35[test])
9                     for train, test in kf ]
10         score = [float(round(n, 3)) for n in scores]
11         print('Acurácia em 3-fold: ', score, '. Média da acurácia: %.3f' % np.mean(score))
```

O resultado da linha 11 é:

```
Para C = 0.031250
Com gamma = 0.000031
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 0.000977
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 0.031250
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 1.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 32.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565

Para C = 0.250000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 0.000977
Acurácia em 3-fold:  [0.528, 0.591, 0.606] .  Média da acurácia:  0.575
Com gamma = 0.031250
Acurácia em 3-fold:  [0.535, 0.606, 0.614] .  Média da acurácia:  0.585
Com gamma = 1.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 32.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565

Para C = 1.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 0.000977
Acurácia em 3-fold:  [0.677, 0.803, 0.827] .  Média da acurácia:  0.769
Com gamma = 0.031250
Acurácia em 3-fold:  [0.866, 0.882, 0.929] .  Média da acurácia:  0.892
Com gamma = 1.000000
Acurácia em 3-fold:  [0.52, 0.598, 0.591] .  Média da acurácia:  0.570
Com gamma = 32.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565

Para C = 4.000000
```

```
Com gamma = 0.000031
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 0.000977
Acurácia em 3-fold:  [0.756, 0.85, 0.874] .  Média da acurácia:  0.827
Com gamma = 0.031250
Acurácia em 3-fold:  [0.866, 0.882, 0.929] .  Média da acurácia:  0.892
Com gamma = 1.000000
Acurácia em 3-fold:  [0.52, 0.598, 0.591] .  Média da acurácia:  0.570
Com gamma = 32.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565

Para C = 32.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.677, 0.803, 0.811] .  Média da acurácia:  0.764
Com gamma = 0.000977
Acurácia em 3-fold:  [0.858, 0.843, 0.913] .  Média da acurácia:  0.871
Com gamma = 0.031250
Acurácia em 3-fold:  [0.866, 0.882, 0.929] .  Média da acurácia:  0.892
Com gamma = 1.000000
Acurácia em 3-fold:  [0.52, 0.598, 0.591] .  Média da acurácia:  0.570
Com gamma = 32.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
```

Assim, para o conjunto (3/5) escolhemos C e *gamma* que possui o maior valor deacurácia média. Logo, $C = 32$ e $gamma = 2^{-5}$ implica em acurácia 89.2%.

```
# conjunto 4 de 5
for c in c_range:
    print('\n', 'Para C = %f'  % c)
    for gamma in gamma_range:
        print('Com gamma = %f' % gamma)
      svc = SVC(C = c, gamma = gamma, kernel='rbf')
        kf = cross_validation.KFold(len(X_45), n_folds=3)
        scores = [ svc.fit(X_45[train], y_45[train]).score(X_45[test], y_45[test])
                   for train, test in kf ]
```

```
10          score = [float(round(n, 3)) for n in scores]
11          print('Acurácia em 3-fold: ', score, '. Média da acurácia: %.3f' % np.mean(score))
```

O resultado da linha 11 é:

```
Para C = 0.031250
Com gamma = 0.000031
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 0.000977
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 0.031250
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 1.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 32.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565

Para C = 0.250000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 0.000977
Acurácia em 3-fold:  [0.512, 0.606, 0.598] .  Média da acurácia:  0.572
Com gamma = 0.031250
Acurácia em 3-fold:  [0.528, 0.614, 0.63] .  Média da acurácia:  0.591
Com gamma = 1.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 32.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565

Para C = 1.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 0.000977
Acurácia em 3-fold:  [0.709, 0.858, 0.85] .  Média da acurácia:  0.806
Com gamma = 0.031250
Acurácia em 3-fold:  [0.795, 0.913, 0.937] .  Média da acurácia:  0.882
Com gamma = 1.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
```

```
Com gamma = 32.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565

Para C = 4.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 0.000977
Acurácia em 3-fold:  [0.748, 0.843, 0.874] .  Média da acurácia:  0.822
Com gamma = 0.031250
Acurácia em 3-fold:  [0.843, 0.906, 0.937] .  Média da acurácia:  0.895
Com gamma = 1.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 32.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565

Para C = 32.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.709, 0.795, 0.85] .  Média da acurácia:  0.785
Com gamma = 0.000977
Acurácia em 3-fold:  [0.882, 0.858, 0.913] .  Média da acurácia:  0.884
Com gamma = 0.031250
Acurácia em 3-fold:  [0.843, 0.906, 0.937] .  Média da acurácia:  0.895
Com gamma = 1.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
Com gamma = 32.000000
Acurácia em 3-fold:  [0.512, 0.591, 0.591] .  Média da acurácia:  0.565
```

Assim, para o conjunto (4/5) escolhemos C e *gamma* que possui o maior valor deacurácia média. Logo, $C = 32$ e $gamma = 2^{-5}$ implica em acurácia 89.5%.

```
# conjunto 5 de 5
for c in c_range:
    print('\n', 'Para C = %f'  % c)
    for gamma in gamma_range:
        print('Com gamma = %f' % gamma)

```

```
7           svc = SVC(C = c, gamma = gamma, kernel='rbf')
8           kf = cross_validation.KFold(len(X_55), n_folds=3)
9           scores = [ svc.fit(X_55[train], y_55[train]).score(X_55[test], y_55[test])
10                     for train, test in kf ]
11          score = [float(round(n, 3)) for n in scores]
12          print('Acurácia em 3-fold: ', score, '. Média da acurácia: %.3f' % np.mean(score))
```

O resultado da linha 11 é:

```
Para C = 0.031250
Com gamma = 0.000031
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566
Com gamma = 0.000977
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566
Com gamma = 0.031250
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566
Com gamma = 1.000000
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566
Com gamma = 32.000000
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566

Para C = 0.250000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566
Com gamma = 0.000977
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566
Com gamma = 0.031250
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566
Com gamma = 1.000000
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566
Com gamma = 32.000000
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566

Para C = 1.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566
Com gamma = 0.000977
Acurácia em 3-fold:  [0.695, 0.843, 0.827] .  Média da acurácia:  0.788
```

```
Com gamma = 0.031250
Acurácia em 3-fold:  [0.828, 0.921, 0.913] .  Média da acurácia:  0.887
Com gamma = 1.000000
Acurácia em 3-fold:  [0.523, 0.583, 0.606] .  Média da acurácia:  0.571
Com gamma = 32.000000
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566

Para C = 4.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566
Com gamma = 0.000977
Acurácia em 3-fold:  [0.758, 0.85, 0.882] .  Média da acurácia:  0.830
Com gamma = 0.031250
Acurácia em 3-fold:  [0.867, 0.921, 0.929] .  Média da acurácia:  0.906
Com gamma = 1.000000
Acurácia em 3-fold:  [0.523, 0.583, 0.606] .  Média da acurácia:  0.571
Com gamma = 32.000000
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566

Para C = 32.000000
Com gamma = 0.000031
Acurácia em 3-fold:  [0.68, 0.811, 0.819] .  Média da acurácia:  0.770
Com gamma = 0.000977
Acurácia em 3-fold:  [0.844, 0.89, 0.874] .  Média da acurácia:  0.869
Com gamma = 0.031250
Acurácia em 3-fold:  [0.867, 0.921, 0.929] .  Média da acurácia:  0.906
Com gamma = 1.000000
Acurácia em 3-fold:  [0.523, 0.583, 0.606] .  Média da acurácia:  0.571
Com gamma = 32.000000
Acurácia em 3-fold:  [0.516, 0.583, 0.598] .  Média da acurácia:  0.566
```

Assim, para o conjunto (5/5) escolhemos C e *gamma* que possui o maior valor deacurácia média. Logo, $C = 32$ e $gamma = 2^{-5}$ implica em acurácia 90.6%.

No último bloco de código, refizemos o SVM com os hiperparâmetros $C = 32$, $gamma = 2^{-5}$ e $kernel$ = RBF. **Nessas condições obtemos um ganho de acurácia de** 1.4% **com relação ao parâmetros em *default*.**

```
1  # SVM nos mesmos dados iniciais, agora com hiperparâmetros turbinados
2  svc = SVC(kernel='rbf', C = 32, gamma = 2**-5)
3
4  # criando a validação cruzada estratificada (mesmo processo do primeiro bloco de código)
5  skf = cross_validation.StratifiedKFold(y, n_folds=5)
6
7  # treinando e avaliando
8  scores = [svc.fit(X[train], y[train]).score(X[test], y[test]) for train, test in skf]
9  score = [float(round(n, 3)) for n in scores]
10 print(score)
11
12 # média dan of scores
13 print('\nMean: %.3f' % np.mean(scores))
```

Respectivamente, os resultados das linhas 10 e 13:

```
[0.911, 0.905, 0.924]
Mean:  0.914
```