SVM

April 29, 2025

[65]:

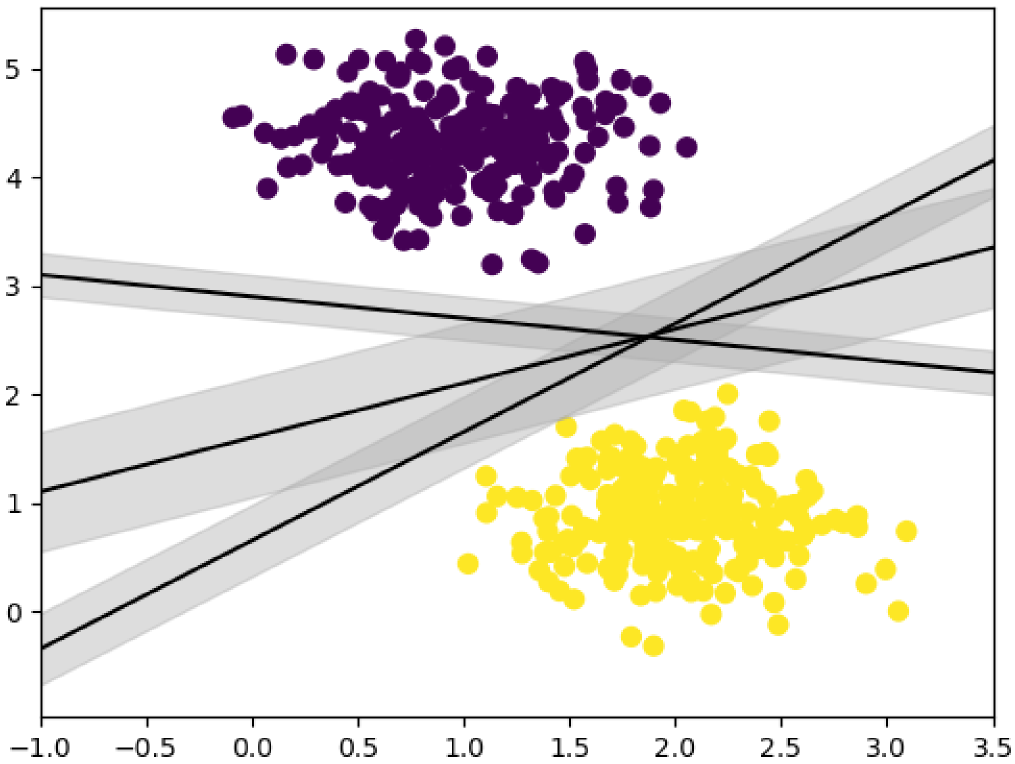
```
from sklearn.datasets import make_blobs
```

[66]:

```
X, Y = make_blobs(n_samples=500, centers=2,
                 random_state=0, cluster_std=0.4)
```

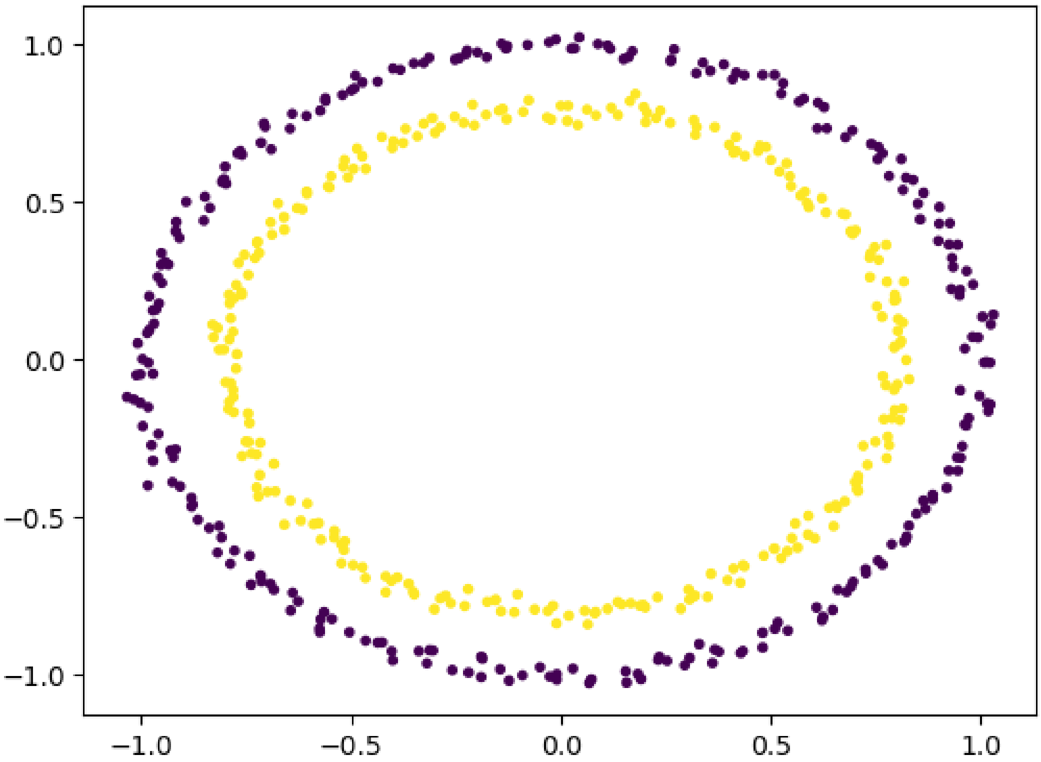
[14]:

```
import numpy as np
xfit = np.linspace(-1, 3.5)
plt.scatter(X[:,0], X[:,1], c=Y, s=50)
for m, b, d in [(1, 0.65, 0.33), (0.5, 1.6, 0.55), (-0.2, 2.9, 0.2)]:
    yfit = m * xfit + b
    plt.plot(xfit, yfit, '-k')
    plt.fill_between(xfit, yfit - d, yfit + d, edgecolor='none',
    color='#AAAAAA', alpha=0.4)

plt.xlim(-1, 3.5);
plt.show()
```



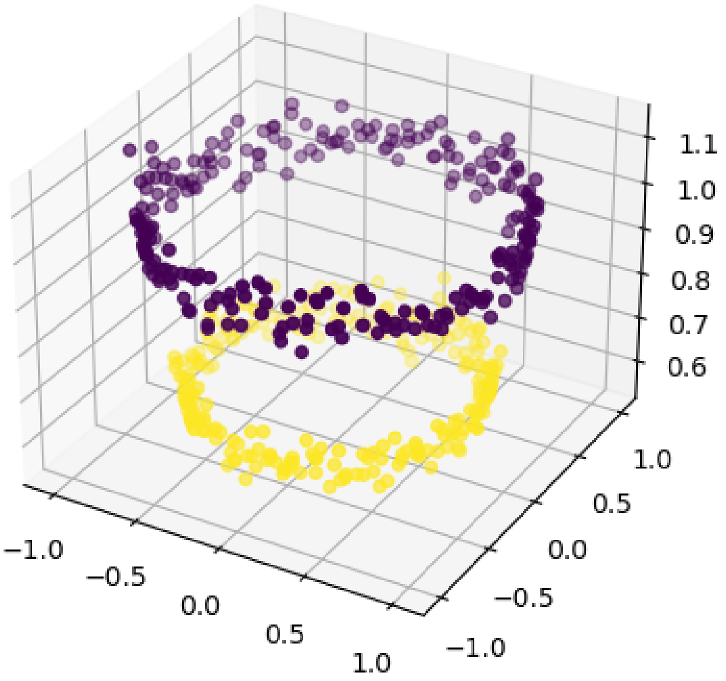
[15]:

```
# importing libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_circles
from mpl_toolkits.mplot3d import Axes3D

# generating data
X, Y = make_circles(n_samples = 500, noise = 0.02)

# visualizing data
plt.scatter(X[:, 0], X[:, 1], c = Y, marker = '.')
plt.show()
```



$$Z = X^2 + Y^2$$

[19]:
```
X1 = X[:, 0].reshape((-1, 1))
X2 = X[:, 1].reshape((-1, 1))
X3 = (X1**2 + X2**2)
X = np.hstack((X, X3))

# visualizing data in higher dimension
fig = plt.figure()
axes = fig.add_subplot(111, projection = '3d')
axes.scatter(X1, X2, X1**2 +X2**2, c = Y , depthshade = True)
plt.show()
```

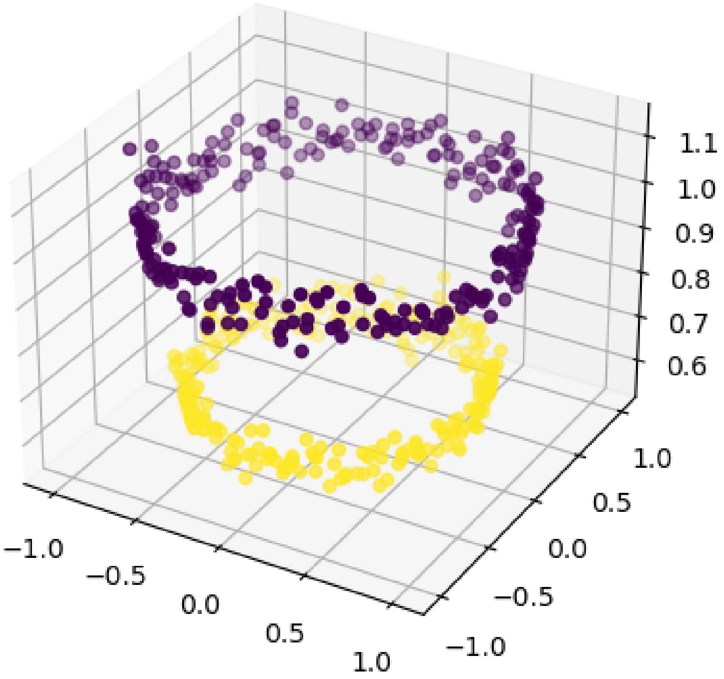


[25]:

```
from sklearn import svm
import matplotlib.pyplot as plt

svc=svm.SVC(kernel='linear')
svc.fit(X,Y)
w=svc.coef_
b=svc.intercept_

x1 = X[:, 0].reshape((-1, 1))
x2 = X[:, 1].reshape((-1, 1))
x1,x2=np.meshgrid(x1,x2)
x3=-(w[0][0]*x1+w[0][1]*x2+b)/w[0][2]

fig = plt.figure()
axes2 = fig.add_subplot(111, projection = '3d')
axes2.scatter(X1, X2, X1**2 + X2**2, c = Y, depthshade = True)
axes1 = fig.gca(projection='3d')
axes1.plot_surface(x1, x2, x3, alpha = 0.01)
plt.show()
```

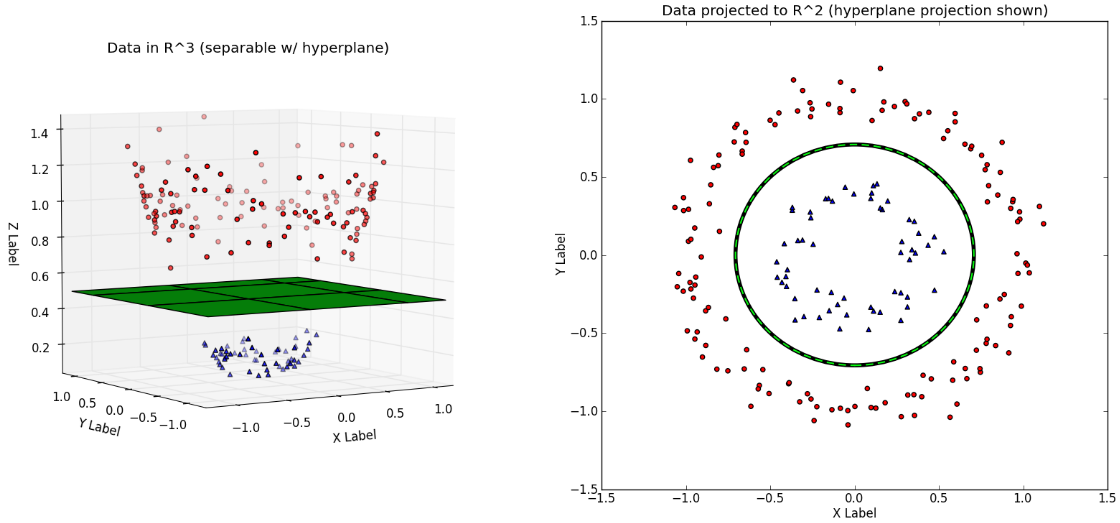
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In [25], line 17
```

```
     15 axes2 = fig.add_subplot(111, projection = '3d')
     16 axes2.scatter(X1, X2, X1**2 + X2**2, c = Y, depthshade = True)
---> 17 axes1 = fig.gca(projection='3d')
     18 axes1.plot_surface(x1, x2, x3, alpha = 0.01)
     19 plt.show()

TypeError: FigureBase.gca() got an unexpected keyword argument 'projection'
```



[26]:
```
from IPython.display import Image
Image(filename='data_2d_to_3d_hyperplane.png')
```

[26]:



[27]:
```
import pandas as pd
import seaborn as sns
from sklearn.datasets import load_breast_cancer
```

[29]:
```
cancer = load_breast_cancer()
```

[30]:
```
cancer.keys()
```

[30]:
```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names',
'filename', 'data_module'])
```

[31]:
```
print(cancer['DESCR'])
```

```
.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

**Data Set Characteristics:**

    :Number of Instances: 569

    :Number of Attributes: 30 numeric, predictive attributes and the class

    :Attribute Information:
        - radius (mean of distances from center to points on the perimeter)
        - texture (standard deviation of gray-scale values)
        - perimeter
        - area
        - smoothness (local variation in radius lengths)
        - compactness (perimeter^2 / area - 1.0)
        - concavity (severity of concave portions of the contour)
        - concave points (number of concave portions of the contour)
        - symmetry
        - fractal dimension ("coastline approximation" - 1)

        The mean, standard error, and "worst" or largest (mean of the three
        worst/largest values) of these features were computed for each image,
        resulting in 30 features.  For instance, field 0 is Mean Radius, field
        10 is Radius SE, field 20 is Worst Radius.

        - class:
                - WDBC-Malignant
                - WDBC-Benign

    :Summary Statistics:

    ===================================== ====== ======
                                           Min    Max
    ===================================== ====== ======
    radius (mean):                        6.981  28.11
    texture (mean):                       9.71   39.28
    perimeter (mean):                     43.79  188.5
    area (mean):                          143.5  2501.0
    smoothness (mean):                    0.053  0.163
    compactness (mean):                   0.019  0.345
    concavity (mean):                     0.0    0.427
    concave points (mean):                0.0    0.201
    symmetry (mean):                      0.106  0.304
    fractal dimension (mean):             0.05   0.097
    radius (standard error):              0.112  2.873
    texture (standard error):             0.36   4.885
    perimeter (standard error):           0.757  21.98
    area (standard error):                6.802  542.2
    smoothness (standard error):          0.002  0.031
    compactness (standard error):         0.002  0.135
    concavity (standard error):           0.0    0.396
    concave points (standard error):      0.0    0.053
    symmetry (standard error):            0.008  0.079
    fractal dimension (standard error):   0.001  0.03
    radius (worst):                       7.93   36.04
    texture (worst):                      12.02  49.54
    perimeter (worst):                    50.41  251.2
    area (worst):                         185.2  4254.0
    smoothness (worst):                   0.071  0.223
    compactness (worst):                  0.027  1.058
    concavity (worst):                    0.0    1.252
    concave points (worst):               0.0    0.291
```

```
    symmetry (worst):                     0.156  0.664
    fractal dimension (worst):            0.055  0.208
    ===================================== ====== ======

    :Missing Attribute Values: None

    :Class Distribution: 212 - Malignant, 357 - Benign

    :Creator:  Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian

    :Donor: Nick Street

    :Date: November, 1995

This is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic) datasets.
https://goo.gl/U2Uwz2

Features are computed from a digitized image of a fine needle
aspirate (FNA) of a breast mass.  They describe
characteristics of the cell nuclei present in the image.

Separating plane described above was obtained using
Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree
Construction Via Linear Programming." Proceedings of the 4th
Midwest Artificial Intelligence and Cognitive Science Society,
pp. 97-101, 1992], a classification method which uses linear
programming to construct a decision tree.  Relevant features
were selected using an exhaustive search in the space of 1-4
features and 1-3 separating planes.

The actual linear program used to obtain the separating plane
in the 3-dimensional space is that described in:
[K. P. Bennett and O. L. Mangasarian: "Robust Linear
Programming Discrimination of Two Linearly Inseparable Sets",
Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

ftp ftp.cs.wisc.edu
cd math-prog/cpo-dataset/machine-learn/WDBC/

.. topic:: References

   - W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction
     for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on
     Electronic Imaging: Science and Technology, volume 1905, pages 861-870,
     San Jose, CA, 1993.
   - O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and
```

```
     prognosis via linear programming. Operations Research, 43(4), pages
570-577,
     July-August 1995.
   - W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning
techniques
     to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77
(1994)
     163-171.
```

[32]:
```
df = pd.DataFrame(cancer['data'], columns=cancer['feature_names'])
```

[67]:
```
df.head()
```

[67]:
```
   mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
0        17.99         10.38          122.80     1001.0          0.11840
1        20.57         17.77          132.90     1326.0          0.08474
2        19.69         21.25          130.00     1203.0          0.10960
3        11.42         20.38           77.58      386.1          0.14250
4        20.29         14.34          135.10     1297.0          0.10030

   mean compactness  mean concavity  mean concave points  mean symmetry  \
0           0.27760          0.3001              0.14710         0.2419
1           0.07864          0.0869              0.07017         0.1812
2           0.15990          0.1974              0.12790         0.2069
3           0.28390          0.2414              0.10520         0.2597
4           0.13280          0.1980              0.10430         0.1809

   mean fractal dimension  …  worst radius  worst texture  worst perimeter  \
0                 0.07871  …         25.38          17.33           184.60
1                 0.05667  …         24.99          23.41           158.80
2                 0.05999  …         23.57          25.53           152.50
3                 0.09744  …         14.91          26.50            98.87
4                 0.05883  …         22.54          16.67           152.20

   worst area  worst smoothness  worst compactness  worst concavity  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
2      1709.0            0.1444             0.4245           0.4504
3       567.7            0.2098             0.8663           0.6869
4      1575.0            0.1374             0.2050           0.4000

   worst concave points  worst symmetry  worst fractal dimension
0                0.2654          0.4601                  0.11890
1                0.1860          0.2750                  0.08902
2                0.2430          0.3613                  0.08758
3                0.2575          0.6638                  0.17300
4                0.1625          0.2364                  0.07678
```

```
[5 rows x 30 columns]
```

[33]:

```
df.describe().T
```

[33]:

```
                         count        mean         std         min  \
mean radius              569.0   14.127292    3.524049    6.981000
mean texture             569.0   19.289649    4.301036    9.710000
mean perimeter           569.0   91.969033   24.298981   43.790000
mean area                569.0  654.889104  351.914129  143.500000
mean smoothness          569.0    0.096360    0.014064    0.052630
mean compactness         569.0    0.104341    0.052813    0.019380
mean concavity           569.0    0.088799    0.079720    0.000000
mean concave points      569.0    0.048919    0.038803    0.000000
mean symmetry            569.0    0.181162    0.027414    0.106000
mean fractal dimension   569.0    0.062798    0.007060    0.049960
radius error             569.0    0.405172    0.277313    0.111500
texture error            569.0    1.216853    0.551648    0.360200
perimeter error          569.0    2.866059    2.021855    0.757000
area error               569.0   40.337079   45.491006    6.802000
smoothness error         569.0    0.007041    0.003003    0.001713
compactness error        569.0    0.025478    0.017908    0.002252
concavity error          569.0    0.031894    0.030186    0.000000
concave points error     569.0    0.011796    0.006170    0.000000
symmetry error           569.0    0.020542    0.008266    0.007882
fractal dimension error  569.0    0.003795    0.002646    0.000895
worst radius             569.0   16.269190    4.833242    7.930000
worst texture            569.0   25.677223    6.146258   12.020000
worst perimeter          569.0  107.261213   33.602542   50.410000
worst area               569.0  880.583128  569.356993  185.200000
worst smoothness         569.0    0.132369    0.022832    0.071170
worst compactness        569.0    0.254265    0.157336    0.027290
worst concavity          569.0    0.272188    0.208624    0.000000
worst concave points     569.0    0.114606    0.065732    0.000000
worst symmetry           569.0    0.290076    0.061867    0.156500
worst fractal dimension  569.0    0.083946    0.018061    0.055040

                                25%         50%          75%         max
mean radius               11.700000   13.370000    15.780000    28.11000
mean texture              16.170000   18.840000    21.800000    39.28000
mean perimeter            75.170000   86.240000   104.100000   188.50000
mean area                420.300000  551.100000   782.700000  2501.00000
mean smoothness            0.086370    0.095870     0.105300     0.16340
mean compactness           0.064920    0.092630     0.130400     0.34540
mean concavity             0.029560    0.061540     0.130700     0.42680
mean concave points        0.020310    0.033500     0.074000     0.20120
mean symmetry              0.161900    0.179200     0.195700     0.30400
mean fractal dimension     0.057700    0.061540     0.066120     0.09744
radius error               0.232400    0.324200     0.478900     2.87300
texture error              0.833900    1.108000     1.474000     4.88500
perimeter error            1.606000    2.287000     3.357000    21.98000
area error                17.850000   24.530000    45.190000   542.20000
smoothness error           0.005169    0.006380     0.008146     0.03113
compactness error          0.013080    0.020450     0.032450     0.13540
concavity error            0.015090    0.025890     0.042050     0.39600
concave points error       0.007638    0.010930     0.014710     0.05279
symmetry error             0.015160    0.018730     0.023480     0.07895
fractal dimension error    0.002248    0.003187     0.004558     0.02984
worst radius              13.010000   14.970000    18.790000    36.04000
worst texture             21.080000   25.410000    29.720000    49.54000
worst perimeter           84.110000   97.660000   125.400000   251.20000
worst area               515.300000  686.500000  1084.000000  4254.00000
worst smoothness           0.116600    0.131300     0.146000     0.22260
worst compactness          0.147200    0.211900     0.339100     1.05800
worst concavity            0.114500    0.226700     0.382900     1.25200
worst concave points       0.064930    0.099930     0.161400     0.29100
worst symmetry             0.250400    0.282200     0.317900     0.66380
worst fractal dimension    0.071460    0.080040     0.092080     0.20750
```

[34]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 30 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   mean radius              569 non-null    float64
 1   mean texture             569 non-null    float64
 2   mean perimeter           569 non-null    float64
 3   mean area                569 non-null    float64
 4   mean smoothness          569 non-null    float64
 5   mean compactness         569 non-null    float64
 6   mean concavity           569 non-null    float64
 7   mean concave points      569 non-null    float64
 8   mean symmetry            569 non-null    float64
 9   mean fractal dimension   569 non-null    float64
 10  radius error             569 non-null    float64
 11  texture error            569 non-null    float64
 12  perimeter error          569 non-null    float64
 13  area error               569 non-null    float64
 14  smoothness error         569 non-null    float64
 15  compactness error        569 non-null    float64
 16  concavity error          569 non-null    float64
 17  concave points error     569 non-null    float64
 18  symmetry error           569 non-null    float64
 19  fractal dimension error  569 non-null    float64
 20  worst radius             569 non-null    float64
 21  worst texture            569 non-null    float64
 22  worst perimeter          569 non-null    float64
 23  worst area               569 non-null    float64
 24  worst smoothness         569 non-null    float64
 25  worst compactness        569 non-null    float64
 26  worst concavity          569 non-null    float64
 27  worst concave points     569 non-null    float64
 28  worst symmetry           569 non-null    float64
 29  worst fractal dimension  569 non-null    float64
dtypes: float64(30)
memory usage: 133.5 KB
```

[35]:

```
cancer['target']
```

[35]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,
       1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1,
       1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0,
       0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0,
       0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
       1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1,
       1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1,
       1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1])
```

[39]:

```
df_target =pd.DataFrame(cancer['target'],columns=['Cancer'])
```

[40]:

```
df_target.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 569 entries, 0 to 568
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Cancer  569 non-null    int64
dtypes: int64(1)
memory usage: 4.6 KB
```

[45]:
```
df_target.sum()
```

[45]:
```
Cancer    357
dtype: int64
```

[53]:
```
569-357
```

[53]:
```
212
```

[76]:
```
df.shape
```

[76]:
```
(569, 30)
```

[68]:
```
from sklearn.preprocessing import StandardScaler
```

[69]:
```
std=StandardScaler()
```

[79]:
```
std.fit(df)
```

[79]:
```
StandardScaler()
```

[80]:
```
X_std=std.transform(df)
```

[81]:
```
X_std
```

[81]:
```
array([[ 1.09706398, -2.07333501,  1.26993369, …,  2.29607613,
         2.75062224,  1.93701461],
       [ 1.82982061, -0.35363241,  1.68595471, …,  1.0870843 ,
        -0.24388967,  0.28118999],
       [ 1.57988811,  0.45618695,  1.56650313, …,  1.95500035,
         1.152255  ,  0.20139121],
       …,
       [ 0.70228425,  2.0455738 ,  0.67267578, …,  0.41406869,
        -1.10454895, -0.31840916],
       [ 1.83834103,  2.33645719,  1.98252415, …,  2.28998549,
         1.91908301,  2.21963528],
       [-1.80840125,  1.22179204, -1.81438851, …, -1.74506282,
        -0.04813821, -0.75120669]])
```

[82]:
```
X_std.shape
```

[82]:

```
(569, 30)
```

[83]:

```
from sklearn.model_selection import train_test_split
```

[84]:

```
X_treino,X_teste,Y_treino,Y_teste = train_test_split(X_std,np.ravel(df_target),
                                                     test_size=0.3,␣
 ↪random_state=101)
```

[85]:

```
from sklearn.svm import SVC
model=SVC()
model.fit(X_treino, Y_treino)
```

[85]:

```
SVC()
```

[86]:

```
previsao = model.predict(X_teste)
```

[87]:

```
from sklearn.metrics import classification_report, confusion_matrix
```

[88]:

```
print(confusion_matrix(Y_teste, previsao))
```

```
[[ 63   3]
 [  1 104]]
```

[90]:

```
print(classification_report(Y_teste, previsao))
```

```
              precision    recall  f1-score   support

           0       0.98      0.95      0.97        66
           1       0.97      0.99      0.98       105

    accuracy                           0.98       171
   macro avg       0.98      0.97      0.98       171
weighted avg       0.98      0.98      0.98       171
```

[91]:

```
param_grid = {'C': [0.1,1, 10, 100, 1000], 'gamma': [1,0.1,0.01,0.001,0.0001],
              'kernel': ['rbf']}

from sklearn.model_selection import GridSearchCV
```

[92]:

```
grid = GridSearchCV(SVC(),param_grid,refit=True, verbose=3)
```

[93]:

```
grid.fit(X_treino,Y_treino)
```

```
Fitting 5 folds for each of 25 candidates, totalling 125 fits
[CV 1/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
[CV 2/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
[CV 3/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.625 total time=   0.0s
[CV 4/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.633 total time=   0.0s
```

```
[CV 5/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.633 total time=   0.0s
[CV 1/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.925 total time=   0.0s
[CV 2/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.950 total time=   0.0s
[CV 3/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.900 total time=   0.0s
[CV 4/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.962 total time=   0.0s
[CV 5/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.949 total time=   0.0s
[CV 1/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.912 total time=   0.0s
[CV 2/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.963 total time=   0.0s
[CV 3/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.975 total time=   0.0s
[CV 4/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.987 total time=   0.0s
[CV 5/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.962 total time=   0.0s
[CV 1/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.688 total time=   0.0s
[CV 2/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.688 total time=   0.0s
[CV 3/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.688 total time=   0.0s
[CV 4/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.684 total time=   0.0s
[CV 5/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.709 total time=   0.0s
[CV 1/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.637 total time=   0.0s
[CV 2/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.637 total time=   0.0s
[CV 3/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.625 total time=   0.0s
[CV 4/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.633 total time=   0.0s
[CV 5/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.633 total time=   0.0s
[CV 1/5] END …C=1, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
[CV 2/5] END …C=1, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
[CV 3/5] END …C=1, gamma=1, kernel=rbf;, score=0.625 total time=   0.0s
[CV 4/5] END …C=1, gamma=1, kernel=rbf;, score=0.633 total time=   0.0s
[CV 5/5] END …C=1, gamma=1, kernel=rbf;, score=0.633 total time=   0.0s
[CV 1/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.950 total time=   0.0s
[CV 2/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.950 total time=   0.0s
[CV 3/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.963 total time=   0.0s
[CV 4/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.975 total time=   0.0s
[CV 5/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.987 total time=   0.0s
[CV 1/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.950 total time=   0.0s
[CV 2/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.988 total time=   0.0s
[CV 3/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.988 total time=   0.0s
[CV 4/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.987 total time=   0.0s
[CV 5/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.975 total time=   0.0s
[CV 1/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.912 total time=   0.0s
[CV 2/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.963 total time=   0.0s
[CV 3/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.975 total time=   0.0s
[CV 4/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.987 total time=   0.0s
[CV 5/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.962 total time=   0.0s
[CV 1/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.688 total time=   0.0s
[CV 2/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.700 total time=   0.0s
[CV 3/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.700 total time=   0.0s
[CV 4/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.696 total time=   0.0s
[CV 5/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.709 total time=   0.0s
[CV 1/5] END …C=10, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
[CV 2/5] END …C=10, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
```

```
[CV 3/5] END …C=10, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
[CV 4/5] END …C=10, gamma=1, kernel=rbf;, score=0.633 total time=   0.0s
[CV 5/5] END …C=10, gamma=1, kernel=rbf;, score=0.633 total time=   0.0s
[CV 1/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.950 total time=   0.0s
[CV 2/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.963 total time=   0.0s
[CV 3/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.963 total time=   0.0s
[CV 4/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.987 total time=   0.0s
[CV 5/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.987 total time=   0.0s
[CV 1/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.963 total time=   0.0s
[CV 2/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.975 total time=   0.0s
[CV 3/5] END …C=10, gamma=0.01, kernel=rbf;, score=1.000 total time=   0.0s
[CV 4/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.975 total time=   0.0s
[CV 5/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.987 total time=   0.0s
[CV 1/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.950 total time=   0.0s
[CV 2/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.988 total time=   0.0s
[CV 3/5] END …C=10, gamma=0.001, kernel=rbf;, score=1.000 total time=   0.0s
[CV 4/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.987 total time=   0.0s
[CV 5/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.975 total time=   0.0s
[CV 1/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.912 total time=   0.0s
[CV 2/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.963 total time=   0.0s
[CV 3/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.975 total time=   0.0s
[CV 4/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.987 total time=   0.0s
[CV 5/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.949 total time=   0.0s
[CV 1/5] END …C=100, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
[CV 2/5] END …C=100, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
[CV 3/5] END …C=100, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
[CV 4/5] END …C=100, gamma=1, kernel=rbf;, score=0.633 total time=   0.0s
[CV 5/5] END …C=100, gamma=1, kernel=rbf;, score=0.633 total time=   0.0s
[CV 1/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.950 total time=   0.0s
[CV 2/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.963 total time=   0.0s
[CV 3/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.975 total time=   0.0s
[CV 4/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.975 total time=   0.0s
[CV 5/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.987 total time=   0.0s
[CV 1/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.975 total time=   0.0s
[CV 2/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.938 total time=   0.0s
[CV 3/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.975 total time=   0.0s
[CV 4/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.949 total time=   0.0s
[CV 5/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.975 total time=   0.0s
[CV 1/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.950 total time=   0.0s
[CV 2/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.975 total time=   0.0s
[CV 3/5] END …C=100, gamma=0.001, kernel=rbf;, score=1.000 total time=   0.0s
[CV 4/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.975 total time=   0.0s
[CV 5/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.987 total time=   0.0s
[CV 1/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.950 total time=   0.0s
[CV 2/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.988 total time=   0.0s
[CV 3/5] END …C=100, gamma=0.0001, kernel=rbf;, score=1.000 total time=   0.0s
[CV 4/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.987 total time=   0.0s
[CV 5/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.975 total time=   0.0s
```

```
[CV 1/5] END …C=1000, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
[CV 2/5] END …C=1000, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
[CV 3/5] END …C=1000, gamma=1, kernel=rbf;, score=0.637 total time=   0.0s
[CV 4/5] END …C=1000, gamma=1, kernel=rbf;, score=0.633 total time=   0.0s
[CV 5/5] END …C=1000, gamma=1, kernel=rbf;, score=0.633 total time=   0.0s
[CV 1/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.950 total time=   0.0s
[CV 2/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.963 total time=   0.0s
[CV 3/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.975 total time=   0.0s
[CV 4/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.975 total time=   0.0s
[CV 5/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.987 total time=   0.0s
[CV 1/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.950 total time=   0.0s
[CV 2/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.950 total time=   0.0s
[CV 3/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.950 total time=   0.0s
[CV 4/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.937 total time=   0.0s
[CV 5/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.975 total time=   0.0s
[CV 1/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.975 total time=   0.0s
[CV 2/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.963 total time=   0.0s
[CV 3/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.975 total time=   0.0s
[CV 4/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.949 total time=   0.0s
[CV 5/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.975 total time=   0.0s
[CV 1/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.950 total time=   0.0s
[CV 2/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.975 total time=   0.0s
[CV 3/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=1.000 total time=   0.0s
[CV 4/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.975 total time=   0.0s
[CV 5/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.987 total time=   0.0s
```

```
[93]: GridSearchCV(estimator=SVC(),
                   param_grid={'C': [0.1, 1, 10, 100, 1000],
                               'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
                               'kernel': ['rbf']},
                   verbose=3)
```

```
[94]: grid.best_params_
```

```
[94]: {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}
```

```
[95]: grid.best_estimator_
```

```
[95]: SVC(C=10, gamma=0.01)
```

```
[96]: grid_previsao=grid.predict(X_teste)
```

```
[97]: print(confusion_matrix(Y_teste, grid_previsao))
```

```
[[ 64   2]
 [  0 105]]
```

```
[64]: print(classification_report(Y_teste, grid_previsao))
```

```
              precision    recall  f1-score   support

           0       0.94      0.89      0.91        66
           1       0.94      0.96      0.95       105

    accuracy                           0.94       171
   macro avg       0.94      0.93      0.93       171
weighted avg       0.94      0.94      0.94       171
```

[98]:

```
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state = 2)
```

[99]:

```
X_treino_res, y_treino_res = sm.fit_resample(X_treino, Y_treino.ravel())
```

[100]:

```
X_treino_res.shape
```

[100]:

```
(504, 30)
```

[101]:

```
y_treino_res.shape
```

[101]:

```
(504,)
```

[103]:

```
sum(y_treino_res==1)
```

[103]:

```
252
```

[104]:

```
sum(y_treino_res==0)
```

[104]:

```
252
```

[105]:

```
from sklearn.svm import SVC
model2=SVC()
model2.fit(X_treino_res, y_treino_res)
previsao_2 = model2.predict(X_teste)
print(confusion_matrix(Y_teste, previsao_2))
print(classification_report(Y_teste, previsao_2))
```

```
[[ 63   3]
 [  2 103]]
              precision    recall  f1-score   support

           0       0.97      0.95      0.96        66
           1       0.97      0.98      0.98       105

    accuracy                           0.97       171
   macro avg       0.97      0.97      0.97       171
weighted avg       0.97      0.97      0.97       171
```

[106]:

```
grid.fit(X_treino_res,y_treino_res)
```

```
Fitting 5 folds for each of 25 candidates, totalling 125 fits
[CV 1/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.495 total time=   0.0s
[CV 2/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.495 total time=   0.0s
[CV 3/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.495 total time=   0.0s
[CV 4/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.495 total time=   0.0s
[CV 5/5] END …C=0.1, gamma=1, kernel=rbf;, score=0.850 total time=   0.0s
[CV 1/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.921 total time=   0.0s
[CV 2/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.911 total time=   0.0s
[CV 3/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.911 total time=   0.0s
[CV 4/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.901 total time=   0.0s
[CV 5/5] END …C=0.1, gamma=0.1, kernel=rbf;, score=0.950 total time=   0.0s
[CV 1/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.950 total time=   0.0s
[CV 2/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.960 total time=   0.0s
[CV 3/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.970 total time=   0.0s
[CV 4/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.941 total time=   0.0s
[CV 5/5] END …C=0.1, gamma=0.01, kernel=rbf;, score=0.980 total time=   0.0s
[CV 1/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.871 total time=   0.0s
[CV 2/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.921 total time=   0.0s
[CV 3/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.901 total time=   0.0s
[CV 4/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.891 total time=   0.0s
[CV 5/5] END …C=0.1, gamma=0.001, kernel=rbf;, score=0.900 total time=   0.0s
[CV 1/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.495 total time=   0.0s
[CV 2/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.495 total time=   0.0s
[CV 3/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.495 total time=   0.0s
[CV 4/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.495 total time=   0.0s
[CV 5/5] END …C=0.1, gamma=0.0001, kernel=rbf;, score=0.620 total time=   0.0s
[CV 1/5] END …C=1, gamma=1, kernel=rbf;, score=0.733 total time=   0.0s
[CV 2/5] END …C=1, gamma=1, kernel=rbf;, score=0.723 total time=   0.0s
[CV 3/5] END …C=1, gamma=1, kernel=rbf;, score=0.762 total time=   0.0s
[CV 4/5] END …C=1, gamma=1, kernel=rbf;, score=0.950 total time=   0.0s
[CV 5/5] END …C=1, gamma=1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 1/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 2/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.950 total time=   0.0s
[CV 3/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.960 total time=   0.0s
[CV 4/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.960 total time=   0.0s
[CV 5/5] END …C=1, gamma=0.1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 1/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.970 total time=   0.0s
[CV 2/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.980 total time=   0.0s
[CV 3/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.980 total time=   0.0s
[CV 4/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.970 total time=   0.0s
[CV 5/5] END …C=1, gamma=0.01, kernel=rbf;, score=0.980 total time=   0.0s
[CV 1/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.950 total time=   0.0s
[CV 2/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.960 total time=   0.0s
[CV 3/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.970 total time=   0.0s
[CV 4/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.941 total time=   0.0s
[CV 5/5] END …C=1, gamma=0.001, kernel=rbf;, score=0.980 total time=   0.0s
```

```
[CV 1/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.871 total time=   0.0s
[CV 2/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.921 total time=   0.0s
[CV 3/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.901 total time=   0.0s
[CV 4/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.861 total time=   0.0s
[CV 5/5] END …C=1, gamma=0.0001, kernel=rbf;, score=0.900 total time=   0.0s
[CV 1/5] END …C=10, gamma=1, kernel=rbf;, score=0.743 total time=   0.0s
[CV 2/5] END …C=10, gamma=1, kernel=rbf;, score=0.762 total time=   0.0s
[CV 3/5] END …C=10, gamma=1, kernel=rbf;, score=0.792 total time=   0.0s
[CV 4/5] END …C=10, gamma=1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 5/5] END …C=10, gamma=1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 1/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.980 total time=   0.0s
[CV 2/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 3/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 4/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.980 total time=   0.0s
[CV 5/5] END …C=10, gamma=0.1, kernel=rbf;, score=0.980 total time=   0.0s
[CV 1/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.980 total time=   0.0s
[CV 2/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.960 total time=   0.0s
[CV 3/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.990 total time=   0.0s
[CV 4/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.970 total time=   0.0s
[CV 5/5] END …C=10, gamma=0.01, kernel=rbf;, score=0.980 total time=   0.0s
[CV 1/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.970 total time=   0.0s
[CV 2/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.980 total time=   0.0s
[CV 3/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.980 total time=   0.0s
[CV 4/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.960 total time=   0.0s
[CV 5/5] END …C=10, gamma=0.001, kernel=rbf;, score=0.980 total time=   0.0s
[CV 1/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.941 total time=   0.0s
[CV 2/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.950 total time=   0.0s
[CV 3/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.970 total time=   0.0s
[CV 4/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.941 total time=   0.0s
[CV 5/5] END …C=10, gamma=0.0001, kernel=rbf;, score=0.980 total time=   0.0s
[CV 1/5] END …C=100, gamma=1, kernel=rbf;, score=0.743 total time=   0.0s
[CV 2/5] END …C=100, gamma=1, kernel=rbf;, score=0.762 total time=   0.0s
[CV 3/5] END …C=100, gamma=1, kernel=rbf;, score=0.792 total time=   0.0s
[CV 4/5] END …C=100, gamma=1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 5/5] END …C=100, gamma=1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 1/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.980 total time=   0.0s
[CV 2/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 3/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.960 total time=   0.0s
[CV 4/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.980 total time=   0.0s
[CV 5/5] END …C=100, gamma=0.1, kernel=rbf;, score=0.980 total time=   0.0s
[CV 1/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.980 total time=   0.0s
[CV 2/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.970 total time=   0.0s
[CV 3/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.970 total time=   0.0s
[CV 4/5] END …C=100, gamma=0.01, kernel=rbf;, score=0.960 total time=   0.0s
[CV 5/5] END …C=100, gamma=0.01, kernel=rbf;, score=1.000 total time=   0.0s
[CV 1/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.980 total time=   0.0s
[CV 2/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.960 total time=   0.0s
[CV 3/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.990 total time=   0.0s
```

```
[CV 4/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.970 total time=   0.0s
[CV 5/5] END …C=100, gamma=0.001, kernel=rbf;, score=0.980 total time=   0.0s
[CV 1/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.970 total time=   0.0s
[CV 2/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.980 total time=   0.0s
[CV 3/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.980 total time=   0.0s
[CV 4/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.960 total time=   0.0s
[CV 5/5] END …C=100, gamma=0.0001, kernel=rbf;, score=0.980 total time=   0.0s
[CV 1/5] END …C=1000, gamma=1, kernel=rbf;, score=0.743 total time=   0.0s
[CV 2/5] END …C=1000, gamma=1, kernel=rbf;, score=0.762 total time=   0.0s
[CV 3/5] END …C=1000, gamma=1, kernel=rbf;, score=0.792 total time=   0.0s
[CV 4/5] END …C=1000, gamma=1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 5/5] END …C=1000, gamma=1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 1/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.980 total time=   0.0s
[CV 2/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.970 total time=   0.0s
[CV 3/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.960 total time=   0.0s
[CV 4/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.980 total time=   0.0s
[CV 5/5] END …C=1000, gamma=0.1, kernel=rbf;, score=0.980 total time=   0.0s
[CV 1/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.970 total time=   0.0s
[CV 2/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.970 total time=   0.0s
[CV 3/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.960 total time=   0.0s
[CV 4/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.970 total time=   0.0s
[CV 5/5] END …C=1000, gamma=0.01, kernel=rbf;, score=0.980 total time=   0.0s
[CV 1/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.970 total time=   0.0s
[CV 2/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.960 total time=   0.0s
[CV 3/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.970 total time=   0.0s
[CV 4/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.970 total time=   0.0s
[CV 5/5] END …C=1000, gamma=0.001, kernel=rbf;, score=0.970 total time=   0.0s
[CV 1/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.980 total time=   0.0s
[CV 2/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.960 total time=   0.0s
[CV 3/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.990 total time=   0.0s
[CV 4/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.970 total time=   0.0s
[CV 5/5] END ..C=1000, gamma=0.0001, kernel=rbf;, score=0.980 total time=   0.0s
```

```
[106]: GridSearchCV(estimator=SVC(),
                    param_grid={'C': [0.1, 1, 10, 100, 1000],
                                'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
                                'kernel': ['rbf']},
                    verbose=3)
```

```
[107]: grid_previsao_2=grid.predict(X_teste)
```

```
[110]: grid.best_params_
```

```
[110]: {'C': 100, 'gamma': 0.01, 'kernel': 'rbf'}
```

```
[108]: print(classification_report(Y_teste, grid_previsao_2))
```

```
              precision    recall  f1-score   support
```

```
           0       0.93      0.95      0.94        66
           1       0.97      0.95      0.96       105

    accuracy                           0.95       171
   macro avg       0.95      0.95      0.95       171
weighted avg       0.95      0.95      0.95       171
```

[109]:

```
print(confusion_matrix(Y_teste, grid_previsao_2))
```

```
[[ 63   3]
 [  5 100]]
```

[ ]:

```

```