

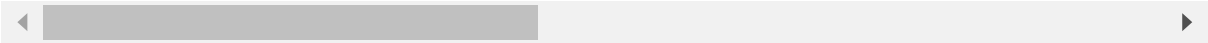
In [49]:

```
#support vector machine
from sklearn.datasets import load_breast_cancer
import pandas as pd
dataset=load_breast_cancer()
df=pd.DataFrame(dataset['data'],columns=dataset['feature_names'])
df['target']=dataset['target']
df.head()
```

Out[49]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809

5 rows × 31 columns



dataset

```
{ 'data': array([[1.799e+01, 1.038e+01, 1.228e+02, ..., 2.654e-01, 4.601e-01,
1.189e-01],
[2.057e+01, 1.777e+01, 1.329e+02, ..., 1.860e-01, 2.750e-01,
8.902e-02],
[1.969e+01, 2.125e+01, 1.300e+02, ..., 2.430e-01, 3.613e-01,
8.758e-02],
...,
[1.660e+01, 2.808e+01, 1.083e+02, ..., 1.418e-01, 2.218e-01,
7.820e-02],
[2.060e+01, 2.933e+01, 1.401e+02, ..., 2.650e-01, 4.087e-01,
1.240e-01],
[7.760e+00, 2.454e+01, 4.792e+01, ..., 0.000e+00, 2.871e-01,
7.039e-02]]),
'target': array([0, 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, 0, 0, 0, 0, 0, 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, 0, 1, 1, 0, 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, 0, 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, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1,
0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0,
0, 0, 1, 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, 0, 1])),
'frame': None,
'target_names': array(['malignant', 'benign'], dtype='<U9'),
'DESCR': '.. _breast_cancer_dataset:\n\nBreast cancer wisconsin (diagnostic) dataset\n-----\n\n**Data Set Characteristics:**\n\nNumber of Instances: 569\n\nNumber of Attributes: 30 numeric, predictive attributes and the class\n\nAttribute Information:\n- radius (mean of distances from center to points on the perimeter)\n- texture (standard deviation of gray-scale values)\n- perimeter\n- area\n- smoothness (local variation in radius lengths)\n- compactness (perimeter^2 / area - 1.0)\n- concavity (severity of concave portions of the contour)\n- concave points (number of concave portions of the contour)\n- symmetry\n- fractal dimension ("coastline approximation" - 1)\n\nThe 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 Rad
ius SE, field 20 is Worst Radius.\n\n      - class:\n      - WDB
C-Malignant\n      - WDBC-Benign\n\n      :Summary Statistics:\n\n
===== \n
Min      Max\n      ===== \n      rad
ius (mean):      6.981  28.11\n      texture (mean):
9.71  39.28\n      perimeter (mean):      43.79  188.5\n      ar
ea (mean):      143.5  2501.0\n      smoothness (mean):
0.053  0.163\n      compactness (mean):      0.019  0.345\n      co
ncavity (mean):      0.0  0.427\n      concave points (mean):
0.0  0.201\n      symmetry (mean):      0.106  0.304\n      fr
actal dimension (mean):      0.05  0.097\n      radius (standard erro
r):      0.112  2.873\n      texture (standard error):      0.3
6  4.885\n      perimeter (standard error):      0.757  21.98\n      area
(standard error):      6.802  542.2\n      smoothness (standard erro
r):      0.002  0.031\n      compactness (standard error):      0.002
0.135\n      concavity (standard error):      0.0  0.396\n      concave p
oints (standard error):      0.0  0.053\n      symmetry (standard error):
0.008  0.079\n      fractal dimension (standard error):      0.001  0.03\n      rad
ius (worst):      7.93  36.04\n      texture (worst):
12.02  49.54\n      perimeter (worst):      50.41  251.2\n      ar
ea (worst):      185.2  4254.0\n      smoothness (worst):
0.071  0.223\n      compactness (worst):      0.027  1.058\n      co
ncavity (worst):      0.0  1.252\n      concave points (wors
t):      0.0  0.291\n      symmetry (worst):      0.
156  0.664\n      fractal dimension (worst):      0.055  0.208\n      ====
===== \n\n      :Missing Attribute Va
lues: None\n\n      :Class Distribution: 212 - Malignant, 357 - Benign\n\n
:Creator: Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian\n\n
:Donor: Nick Street\n\n      :Date: November, 1995\n\nThis is a copy of UCI ML
Breast Cancer Wisconsin (Diagnostic) datasets.\nhttps://goo.gl/U2Uwz2\n\nFea
tures are computed from a digitized image of a fine needle\naspirate (FNA) o
f a breast mass. They describe\ncharacteristics of the cell nuclei present
in the image.\n\nSeparating plane described above was obtained using\nMultis
urface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree\nConstruction Via
Linear Programming." Proceedings of the 4th\nMidwest Artificial Intelligence
and Cognitive Science Society,\npp. 97-101, 1992], a classification method w
hich uses linear\nprogramming to construct a decision tree. Relevant featur
es\nwere selected using an exhaustive search in the space of 1-4\nfeatures a
nd 1-3 separating planes.\n\nThe actual linear program used to obtain the se
parating plane\nin the 3-dimensional space is that described in:\n[K. P. Ben
nett and O. L. Mangasarian: "Robust Linear\nProgramming Discrimination of Tw
o Linearly Inseparable Sets",\nOptimization Methods and Software 1, 1992, 23
-34].\n\nThis database is also available through the UW CS ftp server:\n\nft
p ftp.cs.wisc.edu\ncd math-prog/cpo-dataset/machine-learn/WDBC/\n\n.. topi
c:: References\n\n      - W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nucle
ar feature extraction \n      for breast tumor diagnosis. IS&T/SPIE 1993 Inte
rnational Symposium on \n      Electronic Imaging: Science and Technology, vo
lume 1905, pages 861-870,\n      San Jose, CA, 1993.\n      - O.L. Mangasarian,
W.N. Street and W.H. Wolberg. Breast cancer diagnosis and \n      prognosis v
ia linear programming. Operations Research, 43(4), pages 570-577, \n      Jul
y-August 1995.\n      - W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machin
e learning techniques\n      to diagnose breast cancer from fine-needle aspir
ates. Cancer Letters 77 (1994) \n      163-171.',
'feature_names': array(['mean radius', 'mean texture', 'mean perimeter', 'm
ean area',
      'mean smoothness', 'mean compactness', 'mean concavity',
      'mean concave points', 'mean symmetry', 'mean fractal dimension',
      'radius error', 'texture error', 'perimeter error', 'area error',
      'smoothness error', 'compactness error', 'concavity error',
      'concave points error', 'symmetry error',

```

```
'fractal dimension error', 'worst radius', 'worst texture',
'worst perimeter', 'worst area', 'worst smoothness',
'worst compactness', 'worst concavity', 'worst concave points',
'worst symmetry', 'worst fractal dimension'], dtype='<U23'),
'filename': 'C:\\Users\\91920\\anaconda3\\lib\\site-packages\\sklearn\\data
sets\\data\\breast_cancer.csv'}
```

In [51]:

```
from sklearn.model_selection import train_test_split
data=dataset['data']
target=dataset['target']
x_train,x_test,y_train,y_test=train_test_split(data,target,test_size=0.20,random_state=1)
```

In [52]:

```
print('train data of x_train is: ',x_train.shape)
print('train data of x_test is: ',x_test.shape)
print('train data of y_train is: ',y_train.shape)
print('train data of y_test is: ',y_test.shape)
```

```
train data of x_train is: (455, 30)
train data of x_test is: (114, 30)
train data of y_train is: (455,)
train data of y_test is: (114,)
```

In [53]:

```
from sklearn.svm import SVC
#model=SVC(kernel='linear') #95.4
#model=SVC(C=3,kernel='linear') #95.61
#model=SVC(C=1,kernel='poly') #89.47
#model=SVC(C=1,kernel='sigmoid') #42.15
model=SVC(C=4,kernel='rbf') #90.35
model
```

Out[53]:

```
SVC(C=4)
```

In [54]:

```
model.fit(x_train,y_train)
```

Out[54]:

```
SVC(C=4)
```

In [55]:

```
predicteddata=model.predict(x_test)
print(predicteddata)
```

```
[1 1 1 0 0 0 0 1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 0 1 1 0 0 0 0 1 0 0 1 1 0
 1 1 1 1 1 1 1 0 1 1 1 0 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0
 1 0 1 1 1 0 1 0 1 0 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1
 1 1 1]
```

In [56]:

y_test

Out[56]:

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

In [57]:

```
from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_test,predicteddata)
print("Accuracy of model is : ",accuracy)
```

Accuracy of model is : 0.9035087719298246

In [58]:

```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,predicteddata)
cm
```

Out[58]:

```
array([[32, 10],
       [ 1, 71]], dtype=int64)
```

In [59]:

```
from sklearn.metrics import classification_report
cr=classification_report(y_test,predicteddata)
print(cr)
```

	precision	recall	f1-score	support
0	0.97	0.76	0.85	42
1	0.88	0.99	0.93	72
accuracy			0.90	114
macro avg	0.92	0.87	0.89	114
weighted avg	0.91	0.90	0.90	114

In [60]:

```
# c = 0.01 , 0.1 , 1 , 3 , 7 , 10
# gamma = 0.001 , 0.01 , 0.1 , 1 , 10

# kernal = rbf , poly , linear

from sklearn.model_selection import GridSearchCV
#grid_values = {'gamma':[0.01, 0.1, 1, 10, 100], 'C':[0.01, 0.1, 1, 10, 100]}
param_grid={'C':[0.01 , 0.1 , 1 , 3 , 7 , 10], 'kernel':['rbf' , 'poly', 'linear'] }
grid= GridSearchCV(SVC(),param_grid,cv =5)
```

In [61]:

```
grid.fit(x_train,y_train)
```

Out[61]:

```
GridSearchCV(cv=5, estimator=SVC(),
              param_grid={'C': [0.01, 0.1, 1, 3, 7, 10],
                           'kernel': ['rbf', 'poly', 'linear']})
```

In [62]:

```
grid.best_estimator_
```

Out[62]:

```
SVC(C=0.01, kernel='linear')
```

In [64]:

```
grid.best_score_
```

Out[64]:

```
0.956043956043956
```

In []: