Dataset 2

December 9, 2022

[ ]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.utils import resample
from sklearn.preprocessing  import  scale
from sklearn.decomposition  import PCA
from sklearn import metrics

from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.model_selection import validation_curve
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
import matplotlib.pyplot as plt
import matplotlib.colors as colors
import seaborn as sns
import os
```

[ ]:

```
csv0= pd.read_csv("/home/linux-partition/Desktop/SVM/0.csv")
csv1= pd.read_csv("/home/linux-partition/Desktop/SVM/1.csv")
csv2= pd.read_csv("/home/linux-partition/Desktop/SVM/2.csv")
csv3= pd.read_csv("/home/linux-partition/Desktop/SVM/3.csv")
```

[ ]:

```
allFiles=['/home/linux-partition/Desktop/SVM/0.csv', '/home/linux-partition/
 ↪Desktop/SVM/1.csv', '/home/linux-partition/Desktop/SVM/2.csv', '/home/
 ↪linux-partition/Desktop/SVM/3.csv']
list = []
for file in allFiles:
    read = pd.read_csv(file, header = None)
    list.append(read)
data = pd.concat(list)
```

[ ]:

```
data.info
```

[ ]:

```
<bound method DataFrame.info of          0     1     2     3      4      5       6
7     8     9  …    55  \
0     26.0  4.0  5.0  8.0  -1.0 -13.0 -109.0 -66.0  -9.0  2.0  … -28.0
1    -47.0 -6.0 -5.0 -7.0  13.0  -1.0   35.0 -10.0  10.0 -4.0  … -25.0
2    -19.0 -8.0 -8.0 -8.0 -21.0  -6.0  -79.0  12.0   0.0  5.0  … -83.0
3      2.0  3.0  0.0  2.0   0.0  22.0  106.0 -14.0 -16.0 -2.0  … -38.0
4      6.0  0.0  0.0 -2.0 -14.0  10.0  -51.0   5.0   7.0  0.0  …  38.0
…    …  …  …  …   …   …    …   …   …  …  …   …
2917  -3.0 -1.0 -1.0 -1.0 -28.0  20.0    5.0   0.0  -5.0  0.0  …  -3.0
2918 -13.0 -5.0 -4.0 -3.0  -4.0 -24.0  -10.0  -8.0  20.0  9.0  …   6.0
2919  -1.0 -3.0 -1.0  1.0  30.0  38.0   -1.0  36.0 -10.0  1.0  …  14.0
2920   1.0  4.0  4.0  5.0   9.0 -10.0    4.0   1.0  -2.0 -1.0  … -16.0
2921  -2.0  4.0  2.0 -4.0  12.0   3.0   -2.0   9.0  -8.0 -2.0  …   2.0

        56   57   58    59    60    61     62    63  64
0     61.0  4.0  8.0   5.0   4.0  -7.0  -59.0  16.0   0
1     47.0  6.0  6.0   5.0  13.0  21.0  111.0  15.0   0
2      7.0  7.0  1.0  -8.0   7.0  21.0  114.0  48.0   0
3    -11.0  4.0  7.0  11.0  33.0  39.0  119.0  43.0   0
4    -35.0 -8.0  2.0   6.0 -13.0 -24.0 -112.0 -69.0   0
…    …  …  …   …   …   …    …   …  ..
2917   1.0  4.0  3.0   4.0 -51.0 -49.0    5.0  -9.0   3
2918  -3.0 -3.0 -3.0  -5.0  -4.0 -45.0  -12.0 -15.0   3
2919  -8.0 -4.0 -4.0  -4.0 -21.0 -29.0   -5.0   0.0   3
2920  -3.0  0.0 -3.0  -5.0 -36.0 -90.0    3.0   5.0   3
2921   1.0  0.0 -1.0  -2.0 -30.0  64.0   11.0   5.0   3

[11678 rows x 65 columns]>
```

```
[ ]: data.isnull().sum().head
```

```
[ ]: <bound method NDFrame.head of 0     0
     1     0
     2     0
     3     0
     4     0
          ..
     60    0
     61    0
     62    0
     63    0
     64    0
     Length: 65, dtype: int64>
```

```
[ ]: order= data[64].unique()
     print(order)
```

```
[0 1 2 3]
```

[ ]:

```
data[64].value_counts().sort_values(ascending=False)
```

[ ]:

```
2    2943
3    2922
0    2910
1    2903
Name: 64, dtype: int64
```

[ ]:

```
data.head()
```

[ ]:

```
      0    1    2    3     4     5      6     7     8    9  …    55    56  \
0  26.0  4.0  5.0  8.0  -1.0 -13.0 -109.0 -66.0  -9.0  2.0  … -28.0  61.0
1 -47.0 -6.0 -5.0 -7.0  13.0  -1.0   35.0 -10.0  10.0 -4.0  … -25.0  47.0
2 -19.0 -8.0 -8.0 -8.0 -21.0  -6.0  -79.0  12.0   0.0  5.0  … -83.0   7.0
3   2.0  3.0  0.0  2.0   0.0  22.0  106.0 -14.0 -16.0 -2.0  … -38.0 -11.0
4   6.0  0.0  0.0 -2.0 -14.0  10.0  -51.0   5.0   7.0  0.0  …  38.0 -35.0

    57   58    59    60    61     62    63  64
0  4.0  8.0   5.0   4.0  -7.0  -59.0  16.0   0
1  6.0  6.0   5.0  13.0  21.0  111.0  15.0   0
2  7.0  1.0  -8.0   7.0  21.0  114.0  48.0   0
3  4.0  7.0  11.0  33.0  39.0  119.0  43.0   0
4 -8.0  2.0   6.0 -13.0 -24.0 -112.0 -69.0   0

[5 rows x 65 columns]
```

[ ]:

```
y = data[64]
X = data.drop(columns=64)
```

[ ]:

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import Normalizer,normalize
scaler = MinMaxScaler(feature_range=(0, 1))
X = scaler.fit_transform(X)
X =pd.DataFrame(X)
```

[ ]:

```
X.head()
```

[ ]:

```
         0         1         2         3         4         5         6   \
0  0.625551  0.556701  0.567164  0.638462  0.563380  0.437751  0.074510
1  0.303965  0.505155  0.417910  0.523077  0.629108  0.485944  0.639216
2  0.427313  0.494845  0.373134  0.515385  0.469484  0.465863  0.192157
3  0.519824  0.551546  0.492537  0.592308  0.568075  0.578313  0.917647
4  0.537445  0.536082  0.492537  0.561538  0.502347  0.530120  0.301961

         7         8         9   …       54        55        56        57  \
0  0.244094  0.426160  0.555556  …  0.584314  0.413223  0.728395  0.566524
1  0.464567  0.506329  0.529915  …  0.090196  0.425620  0.670782  0.575107
```

```
2  0.551181  0.464135  0.568376  …  0.000000  0.185950  0.506173  0.579399
3  0.448819  0.396624  0.538462  …  0.290196  0.371901  0.432099  0.566524
4  0.523622  0.493671  0.547009  …  0.737255  0.685950  0.333333  0.515021

         58     59        60        61        62        63
0  0.720000  0.632  0.502347  0.474510  0.270588  0.557769
1  0.693333  0.632  0.544601  0.584314  0.937255  0.553785
2  0.626667  0.528  0.516432  0.584314  0.949020  0.685259
3  0.706667  0.680  0.638498  0.654902  0.968627  0.665339
4  0.640000  0.640  0.422535  0.407843  0.062745  0.219124

[5 rows x 64 columns]
```

[ ]:
```
X_scaled = scale(X)
X_scaled =pd.DataFrame(X_scaled)
X_scaled.head()
```

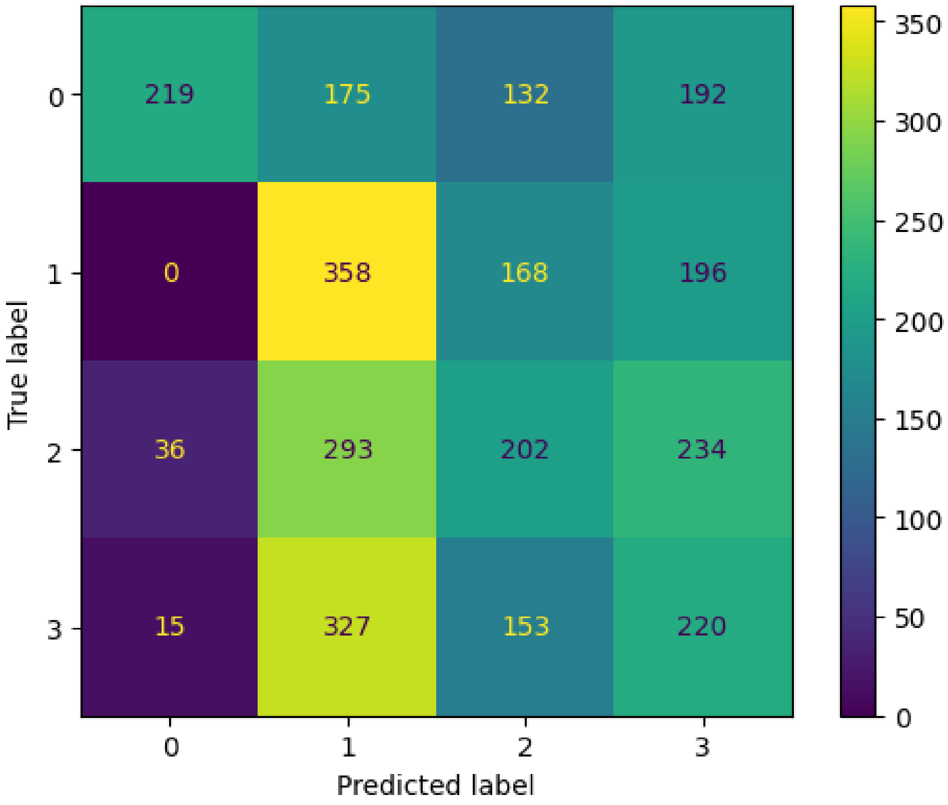
[ ]:
```
          0         1         2         3         4         5         6  \
0  1.428445  0.401724  1.150179  1.173139 -0.047110 -0.482211 -4.293826
1 -2.503492 -0.448155 -0.853937 -0.842622  0.737220 -0.017247  1.445765
2 -0.995352 -0.618131 -1.455172 -0.977006 -1.167581 -0.210982 -3.098077
3  0.135753  0.316736  0.148121  0.366835  0.008913  0.873935  4.275702
4  0.351202  0.061772  0.148121 -0.170702 -0.775416  0.408971 -1.982046

          7         8         9  …        54        55        56        57  \
0 -4.240470 -0.459858  0.224160  …  0.894454 -1.785637  3.396947  0.394898
1 -0.606050  0.588529 -0.282565  … -4.087375 -1.587727  2.627872  0.561499
2  0.821758  0.036746  0.477522  … -4.996756 -5.414002  0.430515  0.644800
3 -0.865651 -0.846106 -0.113656  … -2.070920 -2.445340 -0.558296  0.394898
4  0.367456  0.422994  0.055252  …  2.436448  2.568400 -1.876710 -0.604712

         58        59        60        61        62        63
0  1.764490  0.772653  0.232428 -0.259303 -2.278454  1.069524
1  1.362038  0.772653  0.736892  0.836588  4.451912  1.005130
2  0.355910 -0.987888  0.400583  0.836588  4.570683  3.130128
3  1.563264  1.585211  1.857924  1.541089  4.768635  2.808159
4  0.557136  0.908079 -0.720448 -0.924666 -4.376745 -4.403956

[5 rows x 64 columns]
```

[ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X_scaled,y,random_state=40)
```

[ ]:
```
linear_svm = SVC(kernel='linear')
linear_svm.fit(X_train, y_train)
y_pred_test = linear_svm.predict(X_test)
y_pred_train = linear_svm.predict(X_train)
```

```
print("train_accuracy:", metrics.accuracy_score(y_true=y_train,␣
 ↪y_pred=y_pred_train), "\n")
print("test_accuracy:", metrics.accuracy_score(y_true=y_test,␣
 ↪y_pred=y_pred_test), "\n")
ConfusionMatrixDisplay.from_predictions(y_test, y_pred_test)
plt.show()
```

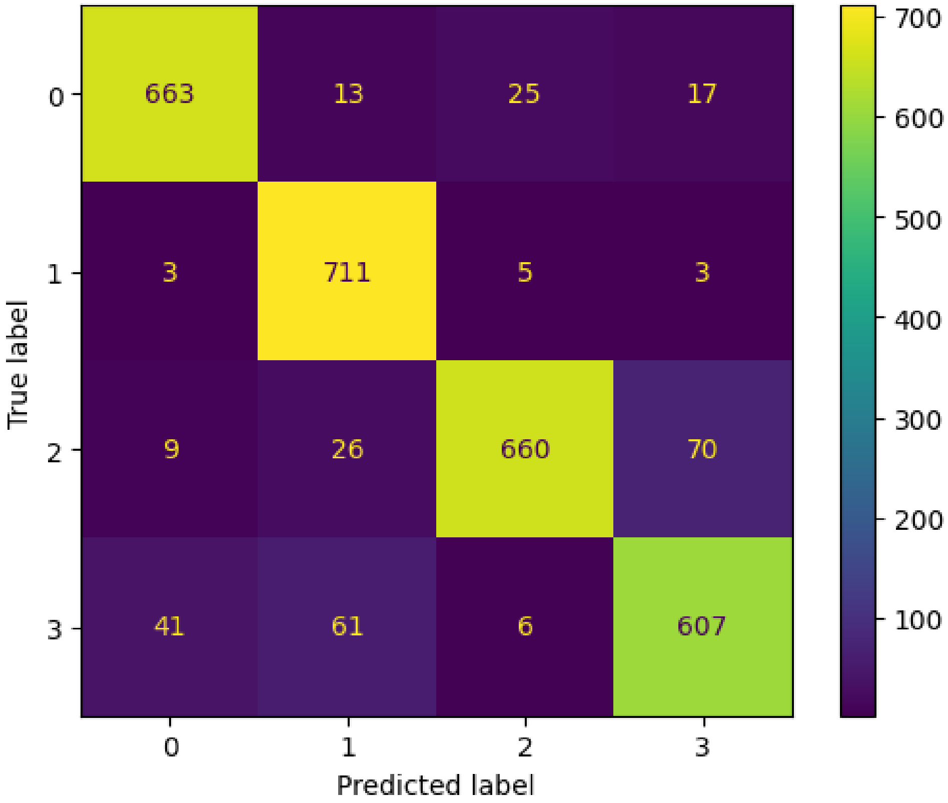
```
train_accuracy: 0.39027175154144783

test_accuracy: 0.3421232876712329
```



```
[ ]: rbf_svm = SVC(kernel='rbf')
     rbf_svm.fit(X_train, y_train)
     y_pred_test = rbf_svm.predict(X_test)
     y_pred_train = rbf_svm.predict(X_train)
     print("train_accuracy:", metrics.accuracy_score(y_true=y_train,␣
      ↪y_pred=y_pred_train), "\n")
     print("test_accuracy:", metrics.accuracy_score(y_true=y_test,␣
      ↪y_pred=y_pred_test), "\n")
     ConfusionMatrixDisplay.from_predictions(y_test, y_pred_test)
```

```
plt.show()
```

```
train_accuracy: 0.9513587577072391

test_accuracy: 0.9044520547945205
```



[ ]:
```
param_grid = [ {
    'C' :[4,6,10,20,25],
    "gamma" : ['scale',0.01,5],
    'kernel' :['rbf']},
                ]

model = SVC(kernel="rbf")

optimal_parameters =GridSearchCV(
    model, param_grid,
    cv=5,
    scoring= 'accuracy',
    verbose=10,
    n_jobs= -5,
    return_train_score=True
)
optimal_parameters.fit(X_train,y_train)

print(optimal_parameters.best_params_)
print(optimal_parameters.best_score_)
```

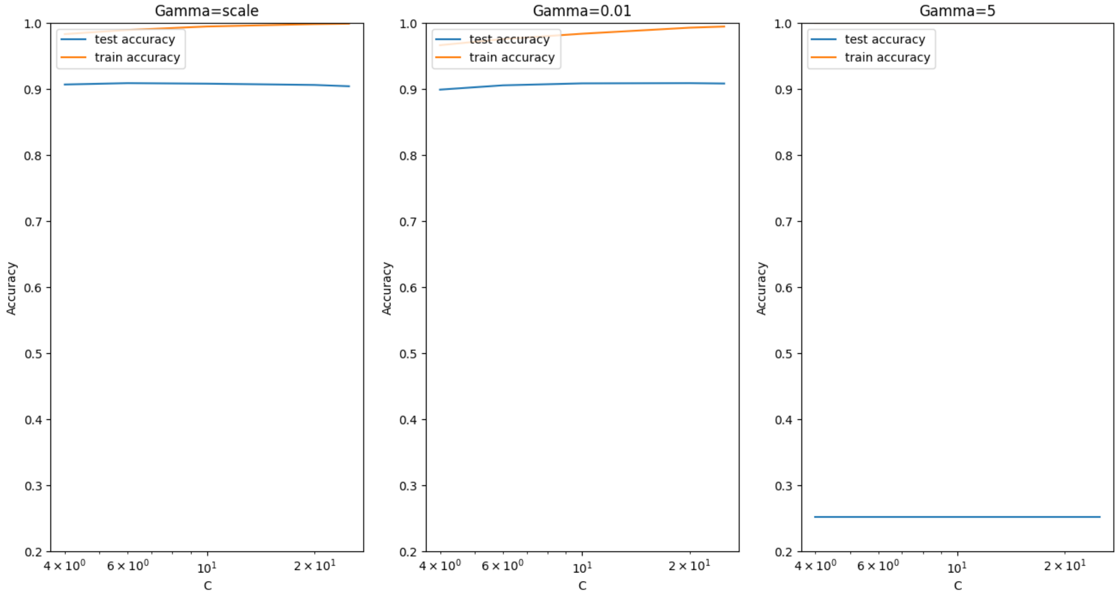
```
Fitting 5 folds for each of 15 candidates, totalling 75 fits
[CV 2/5; 1/15] START C=4, gamma=scale, kernel=rbf…
[CV 2/5; 2/15] START C=4, gamma=0.01, kernel=rbf…
[CV 1/5; 1/15] START C=4, gamma=scale, kernel=rbf…
[CV 3/5; 1/15] START C=4, gamma=scale, kernel=rbf…
[CV 2/5; 3/15] START C=4, gamma=5, kernel=rbf…
[CV 4/5; 1/15] START C=4, gamma=scale, kernel=rbf…
[CV 5/5; 1/15] START C=4, gamma=scale,
kernel=rbf…[CV 1/5; 2/15] START C=4, gamma=0.01,
kernel=rbf…

[CV 4/5; 2/15] START C=4, gamma=0.01, kernel=rbf…
[CV 3/5; 2/15] START C=4, gamma=0.01, kernel=rbf…
[CV 5/5; 2/15] START C=4, gamma=0.01, kernel=rbf…
[CV 1/5; 3/15] START C=4, gamma=5, kernel=rbf…
[CV 5/5; 1/15] END C=4, gamma=scale, kernel=rbf;, score=(train=0.984,
test=0.903) total time=   5.3s
[CV 3/5; 3/15] START C=4, gamma=5, kernel=rbf…
[CV 3/5; 1/15] END C=4, gamma=scale, kernel=rbf;, score=(train=0.983,
test=0.901) total time=   6.8s
[CV 4/5; 3/15] START C=4, gamma=5, kernel=rbf…
[CV 1/5; 1/15] END C=4, gamma=scale, kernel=rbf;, score=(train=0.983,
test=0.917) total time=   6.6s
[CV 5/5; 3/15] START C=4, gamma=5, kernel=rbf…
[CV 3/5; 2/15] END C=4, gamma=0.01, kernel=rbf;, score=(train=0.968, test=0.894)
total time=   7.2s
[CV 1/5; 4/15] START C=6, gamma=scale, kernel=rbf…
[CV 1/5; 2/15] END C=4, gamma=0.01, kernel=rbf;, score=(train=0.965, test=0.907)
total time=   8.0s
[CV 2/5; 4/15] START C=6, gamma=scale, kernel=rbf…
[CV 4/5; 2/15] END C=4, gamma=0.01, kernel=rbf;, score=(train=0.966, test=0.895)
total time=   7.9s
[CV 3/5; 4/15] START C=6, gamma=scale, kernel=rbf…
[CV 2/5; 1/15] END C=4, gamma=scale, kernel=rbf;, score=(train=0.982,
test=0.909) total time=  10.1s
[CV 4/5; 4/15] START C=6, gamma=scale, kernel=rbf…
[CV 2/5; 2/15] END C=4, gamma=0.01, kernel=rbf;, score=(train=0.965, test=0.903)
total time=  10.9s
[CV 5/5; 4/15] START C=6, gamma=scale, kernel=rbf…
[CV 5/5; 2/15] END C=4, gamma=0.01, kernel=rbf;, score=(train=0.968, test=0.897)
```

```
total time=  11.7s
[CV 1/5; 5/15] START C=6, gamma=0.01, kernel=rbf…
[CV 4/5; 1/15] END C=4, gamma=scale, kernel=rbf;, score=(train=0.984,
test=0.903) total time=  11.8s
[CV 2/5; 5/15] START C=6, gamma=0.01, kernel=rbf…
[CV 1/5; 4/15] END C=6, gamma=scale, kernel=rbf;, score=(train=0.989,
test=0.920) total time=   7.3s
[CV 3/5; 5/15] START C=6, gamma=0.01, kernel=rbf…
[CV 4/5; 4/15] END C=6, gamma=scale, kernel=rbf;, score=(train=0.990,
test=0.907) total time=   6.6s
[CV 4/5; 5/15] START C=6, gamma=0.01, kernel=rbf…
[CV 2/5; 4/15] END C=6, gamma=scale, kernel=rbf;, score=(train=0.990,
test=0.912) total time=   7.1s
[CV 5/5; 5/15] START C=6, gamma=0.01, kernel=rbf…
[CV 3/5; 4/15] END C=6, gamma=scale, kernel=rbf;, score=(train=0.990,
test=0.904) total time=   6.9s
[CV 1/5; 6/15] START C=6, gamma=5, kernel=rbf…
[CV 1/5; 5/15] END C=6, gamma=0.01, kernel=rbf;, score=(train=0.975, test=0.919)
total time=   7.8s
[CV 2/5; 6/15] START C=6, gamma=5, kernel=rbf…
[CV 3/5; 5/15] END C=6, gamma=0.01, kernel=rbf;, score=(train=0.975, test=0.900)
total time=   2.3s
[CV 3/5; 6/15] START C=6, gamma=5, kernel=rbf…
[CV 2/5; 5/15] END C=6, gamma=0.01, kernel=rbf;, score=(train=0.974, test=0.908)
total time=   8.7s
[CV 4/5; 6/15] START C=6, gamma=5, kernel=rbf…
[CV 5/5; 4/15] END C=6, gamma=scale, kernel=rbf;, score=(train=0.990,
test=0.902) total time=   9.9s
[CV 5/5; 6/15] START C=6, gamma=5, kernel=rbf…
[CV 5/5; 3/15] END C=4, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  11.1s
[CV 1/5; 7/15] START C=10, gamma=scale, kernel=rbf…
[CV 4/5; 5/15] END C=6, gamma=0.01, kernel=rbf;, score=(train=0.976, test=0.904)
total time=   6.7s
[CV 2/5; 7/15] START C=10, gamma=scale, kernel=rbf…
[CV 5/5; 5/15] END C=6, gamma=0.01, kernel=rbf;, score=(train=0.978, test=0.898)
total time=   9.2s
[CV 3/5; 7/15] START C=10, gamma=scale, kernel=rbf…
[CV 1/5; 7/15] END C=10, gamma=scale, kernel=rbf;, score=(train=0.994,
test=0.918) total time=   3.5s
[CV 4/5; 7/15] START C=10, gamma=scale, kernel=rbf…
[CV 1/5; 3/15] END C=4, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  28.4s
[CV 5/5; 7/15] START C=10, gamma=scale, kernel=rbf…
[CV 2/5; 3/15] END C=4, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  31.5s
[CV 1/5; 8/15] START C=10, gamma=0.01, kernel=rbf…
[CV 2/5; 7/15] END C=10, gamma=scale, kernel=rbf;, score=(train=0.995,
test=0.909) total time=   8.3s
[CV 2/5; 8/15] START C=10, gamma=0.01, kernel=rbf…
[CV 3/5; 7/15] END C=10, gamma=scale, kernel=rbf;, score=(train=0.996,
test=0.905) total time=   6.3s
[CV 3/5; 8/15] START C=10, gamma=0.01, kernel=rbf…
[CV 3/5; 6/15] END C=6, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  11.9s
[CV 4/5; 8/15] START C=10, gamma=0.01, kernel=rbf…
[CV 4/5; 3/15] END C=4, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  26.1s
[CV 5/5; 8/15] START C=10, gamma=0.01, kernel=rbf…
[CV 4/5; 7/15] END C=10, gamma=scale, kernel=rbf;, score=(train=0.995,
test=0.909) total time=   7.9s
[CV 1/5; 9/15] START C=10, gamma=5, kernel=rbf…
[CV 3/5; 3/15] END C=4, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  28.6s
[CV 2/5; 9/15] START C=10, gamma=5, kernel=rbf…
[CV 1/5; 8/15] END C=10, gamma=0.01, kernel=rbf;, score=(train=0.983,
test=0.922) total time=   4.0s
[CV 3/5; 9/15] START C=10, gamma=5, kernel=rbf…
[CV 5/5; 7/15] END C=10, gamma=scale, kernel=rbf;, score=(train=0.995,
test=0.900) total time=   8.6s
[CV 4/5; 9/15] START C=10, gamma=5, kernel=rbf…
[CV 1/5; 6/15] END C=6, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  18.3s
[CV 5/5; 9/15] START C=10, gamma=5, kernel=rbf…
[CV 4/5; 8/15] END C=10, gamma=0.01, kernel=rbf;, score=(train=0.984,
test=0.905) total time=   2.8s
[CV 1/5; 10/15] START C=20, gamma=scale, kernel=rbf…
[CV 2/5; 8/15] END C=10, gamma=0.01, kernel=rbf;, score=(train=0.983,
test=0.914) total time=   8.8s
[CV 2/5; 10/15] START C=20, gamma=scale, kernel=rbf…
[CV 1/5; 10/15] END C=20, gamma=scale, kernel=rbf;, score=(train=0.998,
test=0.914) total time=   2.6s
[CV 3/5; 10/15] START C=20, gamma=scale, kernel=rbf…
[CV 3/5; 8/15] END C=10, gamma=0.01, kernel=rbf;, score=(train=0.984,
test=0.902) total time=   9.0s
[CV 4/5; 10/15] START C=20, gamma=scale, kernel=rbf…
[CV 5/5; 8/15] END C=10, gamma=0.01, kernel=rbf;, score=(train=0.985,
test=0.901) total time=   7.6s
[CV 5/5; 10/15] START C=20, gamma=scale, kernel=rbf…
[CV 5/5; 6/15] END C=6, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  23.3s
[CV 1/5; 11/15] START C=20, gamma=0.01, kernel=rbf…
[CV 2/5; 10/15] END C=20, gamma=scale, kernel=rbf;, score=(train=0.998,
test=0.909) total time=   4.8s
[CV 2/5; 11/15] START C=20, gamma=0.01, kernel=rbf…
[CV 2/5; 6/15] END C=6, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
```

```
total time=  27.7s
[CV 3/5; 11/15] START C=20, gamma=0.01, kernel=rbf…
[CV 4/5; 6/15] END C=6, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  27.6s
[CV 4/5; 11/15] START C=20, gamma=0.01, kernel=rbf…
[CV 4/5; 9/15] END C=10, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=   8.7s
[CV 5/5; 11/15] START C=20, gamma=0.01, kernel=rbf…
[CV 3/5; 10/15] END C=20, gamma=scale, kernel=rbf;, score=(train=0.999,
test=0.904) total time=   8.4s
[CV 1/5; 12/15] START C=20, gamma=5, kernel=rbf…
[CV 4/5; 10/15] END C=20, gamma=scale, kernel=rbf;, score=(train=0.998,
test=0.907) total time=   7.7s
[CV 2/5; 12/15] START C=20, gamma=5, kernel=rbf…
[CV 1/5; 11/15] END C=20, gamma=0.01, kernel=rbf;, score=(train=0.992,
test=0.922) total time=   5.8s
[CV 3/5; 12/15] START C=20, gamma=5, kernel=rbf…
[CV 2/5; 11/15] END C=20, gamma=0.01, kernel=rbf;, score=(train=0.993,
test=0.913) total time=   3.4s
[CV 4/5; 12/15] START C=20, gamma=5, kernel=rbf…
[CV 5/5; 10/15] END C=20, gamma=scale, kernel=rbf;, score=(train=0.998,
test=0.897) total time=  10.0s
[CV 5/5; 12/15] START C=20, gamma=5, kernel=rbf…
[CV 1/5; 9/15] END C=10, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  14.7s
[CV 1/5; 13/15] START C=25, gamma=scale, kernel=rbf…
[CV 3/5; 9/15] END C=10, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  14.1s
[CV 2/5; 13/15] START C=25, gamma=scale, kernel=rbf…
[CV 5/5; 11/15] END C=20, gamma=0.01, kernel=rbf;, score=(train=0.993,
test=0.895) total time=   3.1s
[CV 3/5; 13/15] START C=25, gamma=scale, kernel=rbf…
[CV 4/5; 11/15] END C=20, gamma=0.01, kernel=rbf;, score=(train=0.993,
test=0.913) total time=   6.9s
[CV 4/5; 13/15] START C=25, gamma=scale, kernel=rbf…
[CV 3/5; 11/15] END C=20, gamma=0.01, kernel=rbf;, score=(train=0.995,
test=0.902) total time=   8.5s
[CV 5/5; 13/15] START C=25, gamma=scale, kernel=rbf…
[CV 1/5; 13/15] END C=25, gamma=scale, kernel=rbf;, score=(train=0.999,
test=0.912) total time=   4.9s
[CV 1/5; 14/15] START C=25, gamma=0.01, kernel=rbf…
[CV 3/5; 13/15] END C=25, gamma=scale, kernel=rbf;, score=(train=0.999,
test=0.900) total time=   7.1s
[CV 2/5; 14/15] START C=25, gamma=0.01, kernel=rbf…
[CV 2/5; 13/15] END C=25, gamma=scale, kernel=rbf;, score=(train=0.999,
test=0.911) total time=   9.2s
[CV 3/5; 14/15] START C=25, gamma=0.01, kernel=rbf…
[CV 4/5; 13/15] END C=25, gamma=scale, kernel=rbf;, score=(train=0.999,
test=0.905) total time=   6.6s
[CV 4/5; 14/15] START C=25, gamma=0.01, kernel=rbf…
[CV 4/5; 12/15] END C=20, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=   8.8s
[CV 5/5; 14/15] START C=25, gamma=0.01, kernel=rbf…
[CV 1/5; 14/15] END C=25, gamma=0.01, kernel=rbf;, score=(train=0.994,
test=0.920) total time=   3.6s
[CV 1/5; 15/15] START C=25, gamma=5, kernel=rbf…
[CV 2/5; 9/15] END C=10, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  28.2s
[CV 2/5; 15/15] START C=25, gamma=5, kernel=rbf…
[CV 5/5; 9/15] END C=10, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  25.8s
[CV 3/5; 15/15] START C=25, gamma=5, kernel=rbf…
[CV 3/5; 14/15] END C=25, gamma=0.01, kernel=rbf;, score=(train=0.996,
test=0.904) total time=   2.2s
[CV 4/5; 15/15] START C=25, gamma=5, kernel=rbf…
[CV 5/5; 13/15] END C=25, gamma=scale, kernel=rbf;, score=(train=0.999,
test=0.894) total time=   9.5s
[CV 5/5; 15/15] START C=25, gamma=5, kernel=rbf…
[CV 1/5; 12/15] END C=20, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  13.6s
[CV 4/5; 14/15] END C=25, gamma=0.01, kernel=rbf;, score=(train=0.994,
test=0.914) total time=   4.4s
[CV 5/5; 14/15] END C=25, gamma=0.01, kernel=rbf;, score=(train=0.994,
test=0.897) total time=   3.5s
[CV 2/5; 14/15] END C=25, gamma=0.01, kernel=rbf;, score=(train=0.995,
test=0.906) total time=   7.8s
[CV 3/5; 12/15] END C=20, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  21.5s
[CV 2/5; 12/15] END C=20, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  21.2s
[CV 4/5; 15/15] END C=25, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=   7.3s
[CV 5/5; 12/15] END C=20, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  24.5s
[CV 2/5; 15/15] END C=25, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  11.4s
[CV 5/5; 15/15] END C=25, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  11.7s
[CV 3/5; 15/15] END C=25, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  13.4s
[CV 1/5; 15/15] END C=25, gamma=5, kernel=rbf;, score=(train=1.000, test=0.252)
total time=  14.7s
{'C': 6, 'gamma': 'scale', 'kernel': 'rbf'}
0.9089965551322271
```

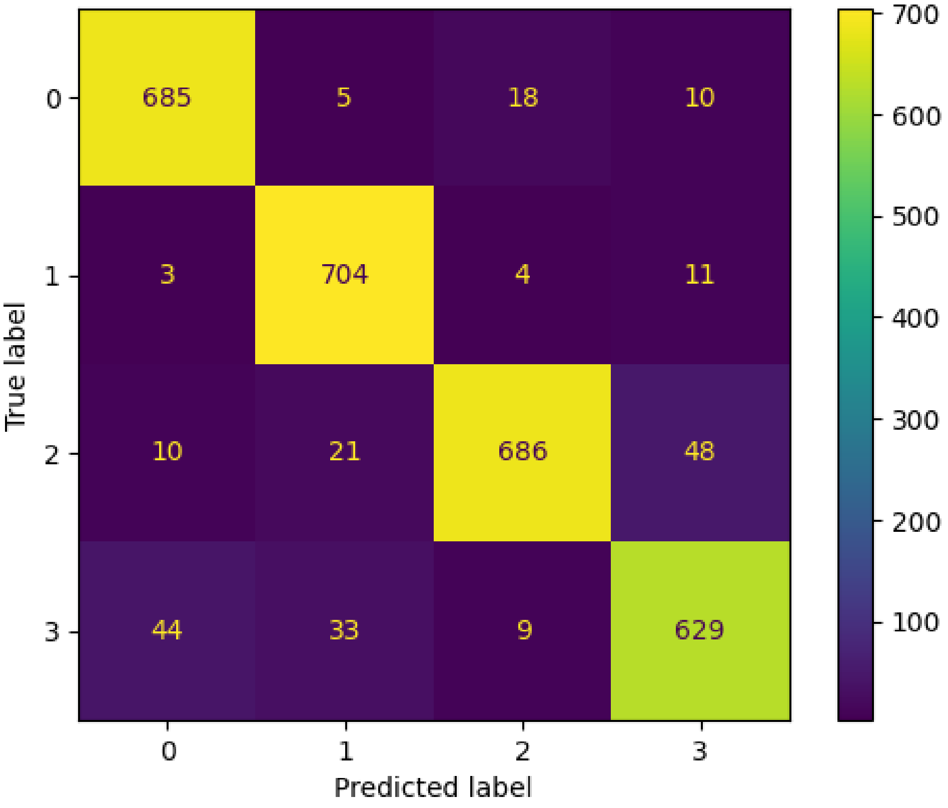
[ ]:

```
cv_results = pd.DataFrame(optimal_parameters.cv_results_)

# converting C to numeric type for plotting on x-axis
cv_results['param_C'] = cv_results['param_C'].astype('int')


plt.figure(figsize=(16,8))

plt.subplot(131)
gamma_scale = cv_results[cv_results['param_gamma']=='scale']

plt.plot(gamma_scale["param_C"], gamma_scale["mean_test_score"])
plt.plot(gamma_scale["param_C"], gamma_scale["mean_train_score"])
plt.xlabel('C')
plt.ylabel('Accuracy')
plt.title("Gamma=scale")
plt.ylim([0.20, 1])
plt.legend(['test accuracy', 'train accuracy'], loc='upper left')
plt.xscale('log')


plt.subplot(132)
gamma_001 = cv_results[cv_results['param_gamma']==0.01]

plt.plot(gamma_001["param_C"], gamma_001["mean_test_score"])
plt.plot(gamma_001["param_C"], gamma_001["mean_train_score"])
plt.xlabel('C')
plt.ylabel('Accuracy')
plt.title("Gamma=0.01")
plt.ylim([0.20, 1])
plt.legend(['test accuracy', 'train accuracy'], loc='upper left')
plt.xscale('log')



plt.subplot(133)
gamma_5 = cv_results[cv_results['param_gamma']==5]

plt.plot(gamma_5["param_C"], gamma_5["mean_test_score"])
plt.plot(gamma_5["param_C"], gamma_5["mean_train_score"])
plt.xlabel('C')
plt.ylabel('Accuracy')
plt.title("Gamma=5")
plt.ylim([0.20, 1])
plt.legend(['test accuracy', 'train accuracy'], loc='upper left')
plt.xscale('log')
```



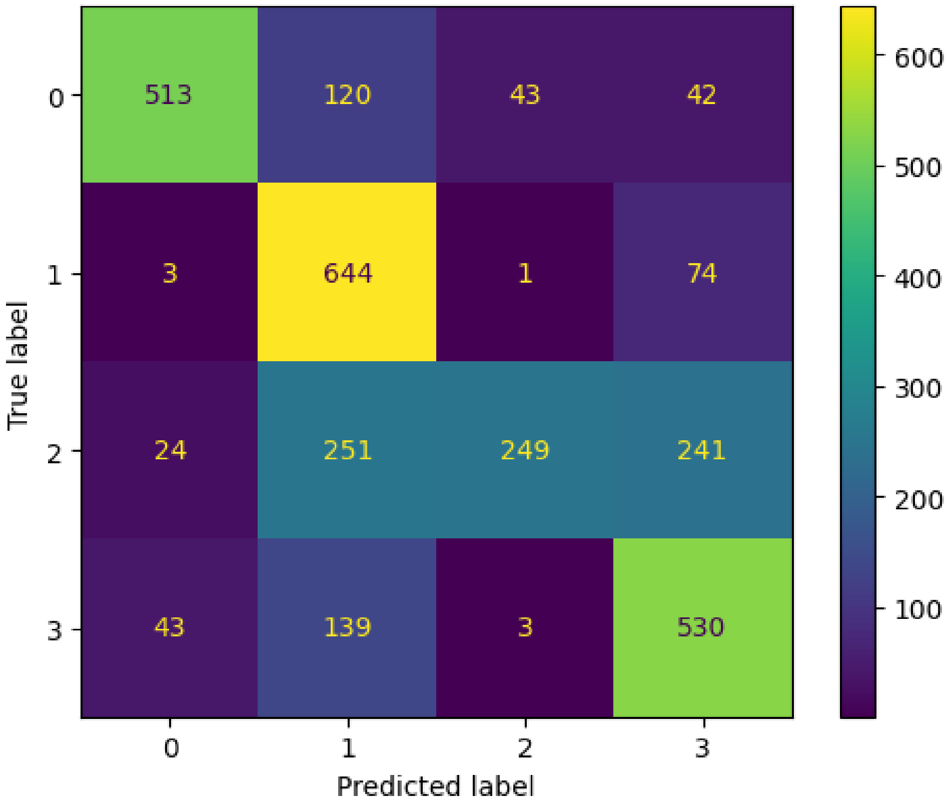
```
[ ]: better_rbf_svm = SVC(C = 6, gamma = 'scale',kernel='rbf')
     better_rbf_svm.fit(X_train, y_train)
     y_pred_test = better_rbf_svm.predict(X_test)
     y_pred_train = better_rbf_svm.predict(X_train)
     print("train_accuracy:", metrics.accuracy_score(y_true=y_train,␣
       ↪y_pred=y_pred_train), "\n")
     print("test_accuracy:", metrics.accuracy_score(y_true=y_test,␣
       ↪y_pred=y_pred_test), "\n")
     ConfusionMatrixDisplay.from_predictions(y_test, y_pred_test)
     plt.show()
```

```
train_accuracy: 0.9892669559260105

test_accuracy: 0.9260273972602739
```



```
[ ]: from sklearn.neighbors import KNeighborsClassifier
     knn_model =KNeighborsClassifier(n_neighbors=3)
     knn_model.fit(X_train, y_train)
     y_pred_test = knn_model.predict(X_test)
     y_pred_train = knn_model.predict(X_train)
     print("Train accuracy: ",metrics.accuracy_score(y_true=y_train,␣
      ↪y_pred=y_pred_train),"\n")
     print("Test accuracy:", metrics.accuracy_score(y_true=y_test,␣
      ↪y_pred=y_pred_test), "\n")
     ConfusionMatrixDisplay.from_predictions(y_test, y_pred_test)
     plt.show()
```

```
Train accuracy:  0.7990408769125371

Test accuracy: 0.663013698630137
```



[ ]:

```
model= KNeighborsClassifier()
param_grid = [{'n_neighbors': np.arange(1, 40)}]
optimal_parameters =GridSearchCV(
    model, param_grid,
    cv=5,
    scoring= 'accuracy',
    verbose=10,
    n_jobs= -1,
    return_train_score=True
)
optimal_parameters.fit(X_train,y_train)

print(optimal_parameters.best_params_)
print(optimal_parameters.best_score_)
```

```
Fitting 5 folds for each of 39 candidates, totalling 195 fits
[CV 1/5; 1/39] START n_neighbors=1…
[CV 2/5; 1/39] START n_neighbors=1…
[CV 3/5; 1/39] START n_neighbors=1…
[CV 5/5; 1/39] START n_neighbors=1…
[CV 4/5; 1/39] START n_neighbors=1…
```

```
[CV 1/5; 2/39] START n_neighbors=2…
[CV 2/5; 2/39] START n_neighbors=2…
[CV 3/5; 2/39] START n_neighbors=2…
[CV 4/5; 2/39] START n_neighbors=2…
[CV 5/5; 2/39] START n_neighbors=2…
[CV 1/5; 3/39] START n_neighbors=3…
[CV 2/5; 3/39] START n_neighbors=3…
[CV 4/5; 3/39] START n_neighbors=3…
[CV 3/5; 3/39] START n_neighbors=3…
[CV 5/5; 3/39] START n_neighbors=3…
[CV 1/5; 4/39] START n_neighbors=4…
[CV 1/5; 2/39] END n_neighbors=2;, score=(train=0.852, test=0.650) total time=
0.2s
[CV 2/5; 4/39] START n_neighbors=4…
[CV 1/5; 3/39] END n_neighbors=3;, score=(train=0.797, test=0.659) total time=
0.2s
[CV 3/5; 4/39] START n_neighbors=4…
[CV 2/5; 2/39] END n_neighbors=2;, score=(train=0.852, test=0.667) total time=
0.2s
[CV 4/5; 1/39] END n_neighbors=1;, score=(train=1.000, test=0.648) total time=
0.2s
[CV 4/5; 4/39] START n_neighbors=4…
[CV 5/5; 4/39] START n_neighbors=4…
[CV 3/5; 1/39] END n_neighbors=1;, score=(train=1.000, test=0.637) total time=
0.2s
[CV 1/5; 5/39] START n_neighbors=5…
[CV 5/5; 1/39] END n_neighbors=1;, score=(train=1.000, test=0.628) total time=
0.2s
[CV 2/5; 5/39] START n_neighbors=5…
[CV 2/5; 3/39] END n_neighbors=3;, score=(train=0.792, test=0.654) total time=
0.2s
[CV 3/5; 5/39] START n_neighbors=5…
[CV 2/5; 1/39] END n_neighbors=1;, score=(train=1.000, test=0.651) total time=
0.2s
[CV 4/5; 5/39] START n_neighbors=5…
[CV 1/5; 1/39] END n_neighbors=1;, score=(train=1.000, test=0.654) total time=
0.3s
[CV 3/5; 2/39] END n_neighbors=2;, score=(train=0.850, test=0.642) total time=
0.2s
[CV 5/5; 5/39] START n_neighbors=5…
[CV 1/5; 6/39] START n_neighbors=6…
[CV 5/5; 2/39] END n_neighbors=2;, score=(train=0.856, test=0.639) total time=
0.2s
[CV 2/5; 6/39] START n_neighbors=6…
[CV 4/5; 2/39] END n_neighbors=2;, score=(train=0.857, test=0.653) total time=
0.3s
[CV 3/5; 6/39] START n_neighbors=6…
[CV 4/5; 3/39] END n_neighbors=3;, score=(train=0.790, test=0.660) total time=
0.2s
[CV 4/5; 6/39] START n_neighbors=6…
[CV 5/5; 3/39] END n_neighbors=3;, score=(train=0.795, test=0.651) total time=
0.2s
[CV 3/5; 3/39] END n_neighbors=3;, score=(train=0.791, test=0.649) total time=
0.2s
[CV 5/5; 6/39] START n_neighbors=6…
[CV 1/5; 7/39] START n_neighbors=7…
[CV 2/5; 4/39] END n_neighbors=4;, score=(train=0.764, test=0.655) total time=
0.2s
[CV 2/5; 7/39] START n_neighbors=7…
[CV 3/5; 4/39] END n_neighbors=4;, score=(train=0.761, test=0.640) total time=
0.2s
[CV 5/5; 4/39] END n_neighbors=4;, score=(train=0.764, test=0.637) total time=
0.2s
[CV 3/5; 7/39] START n_neighbors=7…
[CV 1/5; 4/39] END n_neighbors=4;, score=(train=0.772, test=0.656) total time=
0.2s
[CV 4/5; 7/39] START n_neighbors=7…
[CV 4/5; 4/39] END n_neighbors=4;, score=(train=0.762, test=0.658) total time=
0.2s
[CV 5/5; 7/39] START n_neighbors=7…
[CV 1/5; 8/39] START n_neighbors=8…
[CV 2/5; 5/39] END n_neighbors=5;, score=(train=0.750, test=0.653) total time=
0.2s
[CV 2/5; 8/39] START n_neighbors=8…
[CV 3/5; 5/39] END n_neighbors=5;, score=(train=0.747, test=0.643) total time=
0.2s
[CV 3/5; 8/39] START n_neighbors=8…
[CV 1/5; 6/39] END n_neighbors=6;, score=(train=0.730, test=0.648) total time=
0.2s
[CV 4/5; 8/39] START n_neighbors=8…
[CV 5/5; 5/39] END n_neighbors=5;, score=(train=0.751, test=0.637) total time=
0.2s
[CV 5/5; 8/39] START n_neighbors=8…
[CV 1/5; 5/39] END n_neighbors=5;, score=(train=0.756, test=0.659) total time=
0.2s
[CV 1/5; 9/39] START n_neighbors=9…
[CV 3/5; 6/39] END n_neighbors=6;, score=(train=0.722, test=0.630) total time=
0.2s
[CV 4/5; 5/39] END n_neighbors=5;, score=(train=0.747, test=0.664) total time=
0.2s
[CV 2/5; 9/39] START n_neighbors=9…
[CV 3/5; 9/39] START n_neighbors=9…
[CV 2/5; 6/39] END n_neighbors=6;, score=(train=0.724, test=0.651) total time=
0.2s
[CV 4/5; 9/39] START n_neighbors=9…
[CV 4/5; 6/39] END n_neighbors=6;, score=(train=0.721, test=0.658) total time=
0.2s
[CV 5/5; 9/39] START n_neighbors=9…
[CV 3/5; 7/39] END n_neighbors=7;, score=(train=0.716, test=0.630) total time=
0.2s
[CV 1/5; 10/39] START n_neighbors=10…
[CV 2/5; 7/39] END n_neighbors=7;, score=(train=0.711, test=0.646) total time=
0.2s
[CV 2/5; 10/39] START n_neighbors=10…
[CV 4/5; 7/39] END n_neighbors=7;, score=(train=0.713, test=0.646) total time=
0.2s
[CV 3/5; 10/39] START n_neighbors=10…
[CV 5/5; 6/39] END n_neighbors=6;, score=(train=0.725, test=0.632) total time=
0.2s
[CV 4/5; 10/39] START n_neighbors=10…
[CV 5/5; 7/39] END n_neighbors=7;, score=(train=0.717, test=0.643) total time=
0.2s
[CV 5/5; 10/39] START n_neighbors=10…
[CV 3/5; 8/39] END n_neighbors=8;, score=(train=0.702, test=0.627) total time=
0.2s
[CV 1/5; 11/39] START n_neighbors=11…
[CV 2/5; 8/39] END n_neighbors=8;, score=(train=0.702, test=0.647) total time=
0.2s
[CV 2/5; 11/39] START n_neighbors=11…
[CV 1/5; 7/39] END n_neighbors=7;, score=(train=0.722, test=0.649) total time=
0.2s
[CV 3/5; 11/39] START n_neighbors=11…
[CV 4/5; 8/39] END n_neighbors=8;, score=(train=0.698, test=0.638) total time=
0.2s
[CV 4/5; 11/39] START n_neighbors=11…
[CV 1/5; 8/39] END n_neighbors=8;, score=(train=0.708, test=0.647) total time=
0.3s
[CV 5/5; 11/39] START n_neighbors=11…
[CV 5/5; 8/39] END n_neighbors=8;, score=(train=0.705, test=0.636) total time=
0.2s
[CV 1/5; 12/39] START n_neighbors=12…
[CV 1/5; 9/39] END n_neighbors=9;, score=(train=0.702, test=0.646) total time=
0.3s
[CV 2/5; 12/39] START n_neighbors=12…
[CV 3/5; 9/39] END n_neighbors=9;, score=(train=0.695, test=0.626) total time=
0.2s
[CV 4/5; 9/39] END n_neighbors=9;, score=(train=0.695, test=0.640) total time=
0.2s
[CV 3/5; 12/39] START n_neighbors=12…
[CV 4/5; 12/39] START n_neighbors=12…
[CV 2/5; 9/39] END n_neighbors=9;, score=(train=0.694, test=0.638) total time=
0.2s
[CV 5/5; 12/39] START n_neighbors=12…
[CV 5/5; 9/39] END n_neighbors=9;, score=(train=0.696, test=0.639) total time=
0.2s
[CV 1/5; 13/39] START n_neighbors=13…
[CV 1/5; 10/39] END n_neighbors=10;, score=(train=0.696, test=0.640) total time=
0.2s
[CV 2/5; 13/39] START n_neighbors=13…
[CV 3/5; 10/39] END n_neighbors=10;, score=(train=0.681, test=0.625) total time=
0.2s
[CV 3/5; 13/39] START n_neighbors=13…
[CV 2/5; 10/39] END n_neighbors=10;, score=(train=0.687, test=0.638) total time=
0.2s
[CV 4/5; 13/39] START n_neighbors=13…
[CV 4/5; 10/39] END n_neighbors=10;, score=(train=0.681, test=0.629) total time=
0.2s
[CV 5/5; 13/39] START n_neighbors=13…
[CV 1/5; 11/39] END n_neighbors=11;, score=(train=0.691, test=0.640) total time=
0.2s
[CV 1/5; 14/39] START n_neighbors=14…
[CV 5/5; 10/39] END n_neighbors=10;, score=(train=0.685, test=0.632) total time=
0.2s
[CV 2/5; 11/39] END n_neighbors=11;, score=(train=0.682, test=0.637) total time=
0.2s
[CV 2/5; 14/39] START n_neighbors=14…
[CV 4/5; 11/39] END n_neighbors=11;, score=(train=0.676, test=0.629) total time=
0.2s
[CV 3/5; 14/39] START n_neighbors=14…
[CV 4/5; 14/39] START n_neighbors=14…
[CV 3/5; 11/39] END n_neighbors=11;, score=(train=0.680, test=0.627) total time=
0.2s
[CV 5/5; 14/39] START n_neighbors=14…
[CV 2/5; 12/39] END n_neighbors=12;, score=(train=0.676, test=0.633) total time=
0.2s
[CV 1/5; 15/39] START n_neighbors=15…
[CV 5/5; 11/39] END n_neighbors=11;, score=(train=0.679, test=0.633) total time=
0.2s
[CV 2/5; 15/39] START n_neighbors=15…
[CV 4/5; 12/39] END n_neighbors=12;, score=(train=0.666, test=0.625) total time=
0.2s
[CV 3/5; 15/39] START n_neighbors=15…
[CV 1/5; 12/39] END n_neighbors=12;, score=(train=0.680, test=0.634) total time=
0.2s
[CV 4/5; 15/39] START n_neighbors=15…
[CV 5/5; 12/39] END n_neighbors=12;, score=(train=0.664, test=0.626) total time=
0.2s
[CV 5/5; 15/39] START n_neighbors=15…
[CV 3/5; 12/39] END n_neighbors=12;, score=(train=0.666, test=0.622) total time=
0.3s
[CV 1/5; 16/39] START n_neighbors=16…
[CV 1/5; 13/39] END n_neighbors=13;, score=(train=0.675, test=0.633) total time=
0.2s
[CV 2/5; 16/39] START n_neighbors=16…
[CV 3/5; 13/39] END n_neighbors=13;, score=(train=0.666, test=0.626) total time=
0.2s
[CV 3/5; 16/39] START n_neighbors=16…
[CV 4/5; 13/39] END n_neighbors=13;, score=(train=0.665, test=0.623) total time=
0.2s[CV 2/5; 13/39] END n_neighbors=13;, score=(train=0.670, test=0.638) total
time=   0.2s

[CV 4/5; 16/39] START n_neighbors=16…
[CV 1/5; 14/39] END n_neighbors=14;, score=(train=0.668, test=0.627) total time=
0.2s
[CV 5/5; 16/39] START n_neighbors=16…
[CV 1/5; 17/39] START n_neighbors=17…
[CV 5/5; 13/39] END n_neighbors=13;, score=(train=0.665, test=0.629) total time=
0.2s
[CV 2/5; 17/39] START n_neighbors=17…
[CV 4/5; 14/39] END n_neighbors=14;, score=(train=0.656, test=0.616) total time=
0.2s
[CV 3/5; 17/39] START n_neighbors=17…
[CV 2/5; 14/39] END n_neighbors=14;, score=(train=0.663, test=0.625) total time=
0.2s
[CV 4/5; 17/39] START n_neighbors=17…
[CV 3/5; 14/39] END n_neighbors=14;, score=(train=0.660, test=0.616) total time=
0.2s
[CV 5/5; 17/39] START n_neighbors=17…
[CV 5/5; 14/39] END n_neighbors=14;, score=(train=0.656, test=0.626) total time=
0.2s
[CV 1/5; 18/39] START n_neighbors=18…
[CV 1/5; 15/39] END n_neighbors=15;, score=(train=0.662, test=0.627) total time=
0.2s
[CV 2/5; 18/39] START n_neighbors=18…
[CV 2/5; 15/39] END n_neighbors=15;, score=(train=0.659, test=0.625) total time=
0.2s
[CV 3/5; 15/39] END n_neighbors=15;, score=(train=0.658, test=0.611) total time=
0.2s
[CV 3/5; 18/39] START n_neighbors=18…
[CV 4/5; 18/39] START n_neighbors=18…
[CV 5/5; 15/39] END n_neighbors=15;, score=(train=0.657, test=0.621) total time=
0.2s
[CV 5/5; 18/39] START n_neighbors=18…
[CV 4/5; 15/39] END n_neighbors=15;, score=(train=0.653, test=0.613) total time=
0.2s
[CV 1/5; 19/39] START n_neighbors=19…
[CV 1/5; 16/39] END n_neighbors=16;, score=(train=0.653, test=0.621) total time=
0.2s
[CV 2/5; 19/39] START n_neighbors=19…
[CV 2/5; 16/39] END n_neighbors=16;, score=(train=0.655, test=0.619) total time=
0.2s
[CV 3/5; 19/39] START n_neighbors=19…
[CV 3/5; 16/39] END n_neighbors=16;, score=(train=0.651, test=0.611) total time=
0.2s
[CV 4/5; 19/39] START n_neighbors=19…
[CV 1/5; 17/39] END n_neighbors=17;, score=(train=0.652, test=0.619) total time=
0.2s
[CV 5/5; 19/39] START n_neighbors=19…
[CV 3/5; 17/39] END n_neighbors=17;, score=(train=0.650, test=0.608) total time=
0.2s
[CV 1/5; 20/39] START n_neighbors=20…
[CV 4/5; 16/39] END n_neighbors=16;, score=(train=0.644, test=0.615) total time=
0.2s
[CV 2/5; 20/39] START n_neighbors=20…
[CV 2/5; 17/39] END n_neighbors=17;, score=(train=0.651, test=0.627) total time=
0.2s
[CV 3/5; 20/39] START n_neighbors=20…
[CV 5/5; 16/39] END n_neighbors=16;, score=(train=0.647, test=0.613) total time=
0.3s
[CV 4/5; 20/39] START n_neighbors=20…
[CV 4/5; 17/39] END n_neighbors=17;, score=(train=0.641, test=0.612) total time=
0.2s
[CV 5/5; 20/39] START n_neighbors=20…
[CV 5/5; 17/39] END n_neighbors=17;, score=(train=0.647, test=0.615) total time=
0.2s
[CV 1/5; 21/39] START n_neighbors=21…
[CV 1/5; 18/39] END n_neighbors=18;, score=(train=0.647, test=0.616) total time=
0.2s
[CV 2/5; 21/39] START
n_neighbors=21…[CV 2/5; 18/39] END
n_neighbors=18;, score=(train=0.644, test=0.618) total time=   0.2s

[CV 3/5; 21/39] START n_neighbors=21…
[CV 5/5; 18/39] END n_neighbors=18;, score=(train=0.640, test=0.612) total time=
0.2s
[CV 4/5; 21/39] START n_neighbors=21…
[CV 3/5; 18/39] END n_neighbors=18;, score=(train=0.643, test=0.601) total time=
0.2s
[CV 5/5; 21/39] START n_neighbors=21…
[CV 1/5; 19/39] END n_neighbors=19;, score=(train=0.642, test=0.615) total time=
0.2s
[CV 4/5; 18/39] END n_neighbors=18;, score=(train=0.638, test=0.611) total time=
0.2s
[CV 1/5; 22/39] START n_neighbors=22…
[CV 2/5; 22/39] START n_neighbors=22…
[CV 2/5; 19/39] END n_neighbors=19;, score=(train=0.641, test=0.619) total time=
0.2s
[CV 3/5; 22/39] START n_neighbors=22…
```

```
[CV 4/5; 19/39] END n_neighbors=19;, score=(train=0.634, test=0.605) total time=
0.2s
[CV 4/5; 22/39] START n_neighbors=22…
[CV 3/5; 19/39] END n_neighbors=19;, score=(train=0.637, test=0.600) total time=
0.2s
[CV 5/5; 22/39] START n_neighbors=22…
[CV 1/5; 20/39] END n_neighbors=20;, score=(train=0.638, test=0.614) total time=
0.2s
[CV 1/5; 23/39] START n_neighbors=23…
[CV 5/5; 19/39] END n_neighbors=19;, score=(train=0.639, test=0.617) total time=
0.2s
[CV 2/5; 23/39] START n_neighbors=23…
[CV 2/5; 20/39] END n_neighbors=20;, score=(train=0.635, test=0.612) total time=
0.2s
[CV 3/5; 23/39] START n_neighbors=23…
[CV 3/5; 20/39] END n_neighbors=20;, score=(train=0.637, test=0.599) total time=
0.2s
[CV 4/5; 23/39] START n_neighbors=23…
[CV 5/5; 20/39] END n_neighbors=20;, score=(train=0.633, test=0.613) total time=
0.2s
[CV 5/5; 23/39] START n_neighbors=23…
[CV 4/5; 20/39] END n_neighbors=20;, score=(train=0.628, test=0.600) total time=
0.3s
[CV 1/5; 24/39] START n_neighbors=24…
[CV 1/5; 21/39] END n_neighbors=21;, score=(train=0.636, test=0.607) total time=
0.2s
[CV 2/5; 24/39] START n_neighbors=24…
[CV 3/5; 21/39] END n_neighbors=21;, score=(train=0.634, test=0.597) total time=
0.2s
[CV 3/5; 24/39] START n_neighbors=24…
[CV 4/5; 21/39] END n_neighbors=21;, score=(train=0.626, test=0.595) total time=
0.2s
[CV 4/5; 24/39] START n_neighbors=24…
[CV 5/5; 21/39] END n_neighbors=21;, score=(train=0.630, test=0.607) total time=
0.2s
[CV 5/5; 24/39] START n_neighbors=24…
[CV 2/5; 21/39] END n_neighbors=21;, score=(train=0.632, test=0.611) total time=
0.2s
[CV 1/5; 25/39] START n_neighbors=25…
[CV 2/5; 22/39] END n_neighbors=22;, score=(train=0.629, test=0.606) total time=
0.2s
[CV 2/5; 25/39] START n_neighbors=25…
[CV 1/5; 22/39] END n_neighbors=22;, score=(train=0.631, test=0.604) total time=
0.3s
[CV 3/5; 25/39] START n_neighbors=25…
[CV 3/5; 22/39] END n_neighbors=22;, score=(train=0.631, test=0.594) total time=
0.3s
[CV 4/5; 25/39] START n_neighbors=25…
```

```
[CV 5/5; 22/39] END n_neighbors=22;, score=(train=0.626, test=0.605) total time=
0.2s
[CV 5/5; 25/39] START n_neighbors=25…
[CV 4/5; 22/39] END n_neighbors=22;, score=(train=0.623, test=0.595) total time=
0.2s
[CV 1/5; 26/39] START n_neighbors=26…
[CV 1/5; 23/39] END n_neighbors=23;, score=(train=0.628, test=0.608) total time=
0.2s
[CV 2/5; 26/39] START n_neighbors=26…
[CV 2/5; 23/39] END n_neighbors=23;, score=(train=0.627, test=0.608) total time=
0.2s
[CV 3/5; 26/39] START n_neighbors=26…
[CV 3/5; 23/39] END n_neighbors=23;, score=(train=0.628, test=0.595) total time=
0.2s
[CV 4/5; 26/39] START n_neighbors=26…
[CV 4/5; 23/39] END n_neighbors=23;, score=(train=0.620, test=0.594) total time=
0.2s
[CV 5/5; 26/39] START n_neighbors=26…
[CV 5/5; 23/39] END n_neighbors=23;, score=(train=0.624, test=0.607) total time=
0.2s
[CV 1/5; 27/39] START n_neighbors=27…
[CV 3/5; 24/39] END n_neighbors=24;, score=(train=0.624, test=0.592) total time=
0.2s
[CV 2/5; 27/39] START n_neighbors=27…
[CV 2/5; 24/39] END n_neighbors=24;, score=(train=0.622, test=0.603) total time=
0.2s
[CV 3/5; 27/39] START n_neighbors=27…
[CV 4/5; 24/39] END n_neighbors=24;, score=(train=0.618, test=0.595) total time=
0.2s
[CV 4/5; 27/39] START n_neighbors=27…
[CV 1/5; 24/39] END n_neighbors=24;, score=(train=0.627, test=0.604) total time=
0.2s
[CV 5/5; 27/39] START n_neighbors=27…
[CV 5/5; 24/39] END n_neighbors=24;, score=(train=0.619, test=0.603) total time=
0.2s
[CV 1/5; 28/39] START n_neighbors=28…
[CV 1/5; 25/39] END n_neighbors=25;, score=(train=0.624, test=0.603) total time=
0.2s
[CV 2/5; 28/39] START n_neighbors=28…
[CV 2/5; 25/39] END n_neighbors=25;, score=(train=0.622, test=0.611) total time=
0.2s
[CV 3/5; 28/39] START n_neighbors=28…
[CV 3/5; 25/39] END n_neighbors=25;, score=(train=0.622, test=0.594) total time=
0.2s
[CV 4/5; 28/39] START n_neighbors=28…
[CV 4/5; 25/39] END n_neighbors=25;, score=(train=0.618, test=0.595) total time=
0.3s
[CV 5/5; 28/39] START n_neighbors=28…
```

```
[CV 2/5; 26/39] END n_neighbors=26;, score=(train=0.618, test=0.610) total time=
0.2s
[CV 1/5; 29/39] START n_neighbors=29…
[CV 1/5; 26/39] END n_neighbors=26;, score=(train=0.621, test=0.602) total time=
0.3s
[CV 2/5; 29/39] START
n_neighbors=29…[CV 5/5; 25/39] END
n_neighbors=25;, score=(train=0.618, test=0.603) total time=   0.3s

[CV 3/5; 29/39] START n_neighbors=29…
[CV 5/5; 26/39] END n_neighbors=26;, score=(train=0.617, test=0.596) total time=
0.2s
[CV 4/5; 29/39] START n_neighbors=29…
[CV 3/5; 26/39] END n_neighbors=26;, score=(train=0.619, test=0.592) total time=
0.2s
[CV 5/5; 29/39] START n_neighbors=29…
[CV 4/5; 26/39] END n_neighbors=26;, score=(train=0.612, test=0.592) total time=
0.2s
[CV 1/5; 30/39] START n_neighbors=30…
[CV 2/5; 27/39] END n_neighbors=27;, score=(train=0.614, test=0.602) total time=
0.2s
[CV 2/5; 30/39] START n_neighbors=30…
[CV 1/5; 27/39] END n_neighbors=27;, score=(train=0.620, test=0.602) total time=
0.2s
[CV 3/5; 30/39] START n_neighbors=30…
[CV 3/5; 27/39] END n_neighbors=27;, score=(train=0.617, test=0.592) total time=
0.2s
[CV 4/5; 30/39] START n_neighbors=30…
[CV 4/5; 27/39] END n_neighbors=27;, score=(train=0.613, test=0.585) total time=
0.2s
[CV 5/5; 30/39] START n_neighbors=30…
[CV 1/5; 28/39] END n_neighbors=28;, score=(train=0.617, test=0.600) total time=
0.2s
[CV 1/5; 31/39] START n_neighbors=31…
[CV 5/5; 27/39] END n_neighbors=27;, score=(train=0.613, test=0.603) total time=
0.3s
[CV 2/5; 31/39] START n_neighbors=31…
[CV 2/5; 28/39] END n_neighbors=28;, score=(train=0.611, test=0.598) total time=
0.3s
[CV 3/5; 31/39] START n_neighbors=31…
[CV 3/5; 28/39] END n_neighbors=28;, score=(train=0.612, test=0.592) total time=
0.2s
[CV 4/5; 31/39] START n_neighbors=31…
[CV 4/5; 28/39] END n_neighbors=28;, score=(train=0.609, test=0.585) total time=
0.2s
[CV 5/5; 31/39] START n_neighbors=31…
[CV 5/5; 28/39] END n_neighbors=28;, score=(train=0.610, test=0.591) total time=
0.2s
```

```
[CV 1/5; 32/39] START n_neighbors=32…
[CV 1/5; 29/39] END n_neighbors=29;, score=(train=0.613, test=0.596) total time=
0.2s
[CV 2/5; 32/39] START n_neighbors=32…
[CV 2/5; 29/39] END n_neighbors=29;, score=(train=0.611, test=0.599) total time=
0.2s
[CV 3/5; 32/39] START n_neighbors=32…
[CV 3/5; 29/39] END n_neighbors=29;, score=(train=0.611, test=0.588) total time=
0.2s
[CV 4/5; 32/39] START n_neighbors=32…
[CV 4/5; 29/39] END n_neighbors=29;, score=(train=0.608, test=0.580) total time=
0.2s
[CV 5/5; 32/39] START n_neighbors=32…
[CV 2/5; 30/39] END n_neighbors=30;, score=(train=0.608, test=0.603) total time=
0.2s
[CV 1/5; 33/39] START n_neighbors=33…
[CV 1/5; 30/39] END n_neighbors=30;, score=(train=0.610, test=0.595) total time=
0.3s
[CV 5/5; 30/39] END n_neighbors=30;, score=(train=0.608, test=0.592) total time=
0.2s
[CV 2/5; 33/39] START n_neighbors=33…
[CV 5/5; 29/39] END n_neighbors=29;, score=(train=0.610, test=0.593) total time=
0.3s
[CV 3/5; 33/39] START n_neighbors=33…
[CV 4/5; 33/39] START n_neighbors=33…
[CV 4/5; 30/39] END n_neighbors=30;, score=(train=0.606, test=0.582) total time=
0.2s
[CV 5/5; 33/39] START n_neighbors=33…
[CV 3/5; 30/39] END n_neighbors=30;, score=(train=0.609, test=0.588) total time=
0.2s
[CV 1/5; 34/39] START n_neighbors=34…
[CV 1/5; 31/39] END n_neighbors=31;, score=(train=0.607, test=0.588) total time=
0.2s
[CV 2/5; 34/39] START n_neighbors=34…
[CV 2/5; 31/39] END n_neighbors=31;, score=(train=0.609, test=0.597) total time=
0.2s
[CV 3/5; 34/39] START n_neighbors=34…
[CV 3/5; 31/39] END n_neighbors=31;, score=(train=0.606, test=0.586) total time=
0.2s
[CV 4/5; 34/39] START n_neighbors=34…
[CV 5/5; 31/39] END n_neighbors=31;, score=(train=0.603, test=0.591) total time=
0.3s
[CV 5/5; 34/39] START n_neighbors=34…
[CV 4/5; 31/39] END n_neighbors=31;, score=(train=0.602, test=0.581) total time=
0.2s
[CV 1/5; 35/39] START n_neighbors=35…
[CV 1/5; 32/39] END n_neighbors=32;, score=(train=0.606, test=0.588) total time=
0.2s
```

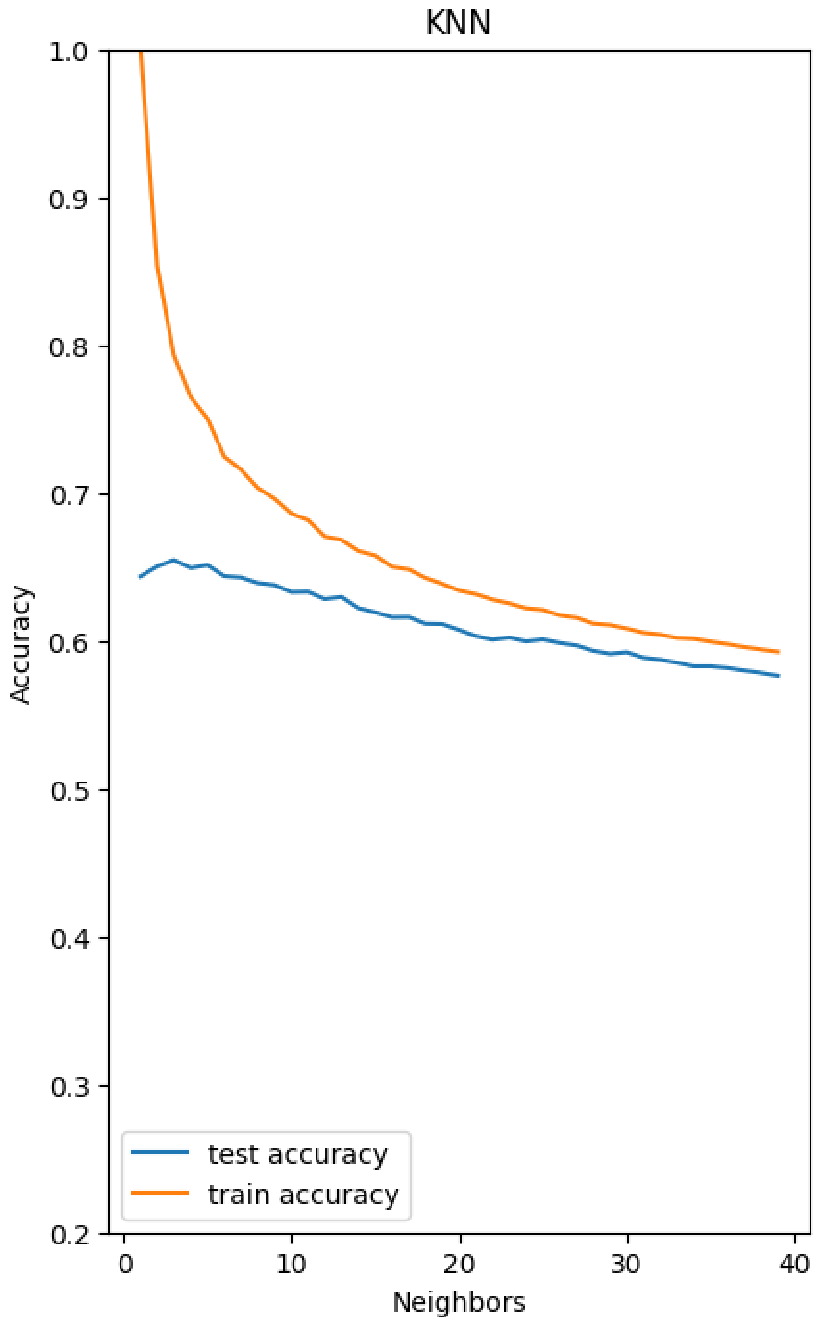
```
[CV 2/5; 35/39] START n_neighbors=35…
[CV 3/5; 32/39] END n_neighbors=32;, score=(train=0.603, test=0.582) total time=
0.3s
[CV 3/5; 35/39] START n_neighbors=35…
[CV 2/5; 32/39] END n_neighbors=32;, score=(train=0.605, test=0.599) total time=
0.3s
[CV 4/5; 35/39] START n_neighbors=35…
[CV 4/5; 32/39] END n_neighbors=32;, score=(train=0.603, test=0.579) total time=
0.3s
[CV 5/5; 35/39] START n_neighbors=35…
[CV 5/5; 32/39] END n_neighbors=32;, score=(train=0.603, test=0.588) total time=
0.2s
[CV 1/5; 36/39] START n_neighbors=36…
[CV 1/5; 33/39] END n_neighbors=33;, score=(train=0.605, test=0.586) total time=
0.2s
[CV 2/5; 36/39] START n_neighbors=36…
[CV 3/5; 33/39] END n_neighbors=33;, score=(train=0.602, test=0.583) total time=
0.2s
[CV 3/5; 36/39] START n_neighbors=36…
[CV 4/5; 33/39] END n_neighbors=33;, score=(train=0.598, test=0.576) total time=
0.2s
[CV 4/5; 36/39] START n_neighbors=36…
[CV 2/5; 33/39] END n_neighbors=33;, score=(train=0.605, test=0.597) total time=
0.3s
[CV 5/5; 36/39] START n_neighbors=36…
[CV 5/5; 33/39] END n_neighbors=33;, score=(train=0.599, test=0.584) total time=
0.2s
[CV 1/5; 37/39] START n_neighbors=37…
[CV 1/5; 34/39] END n_neighbors=34;, score=(train=0.605, test=0.587) total time=
0.2s
[CV 2/5; 37/39] START n_neighbors=37…
[CV 2/5; 34/39] END n_neighbors=34;, score=(train=0.603, test=0.593) total time=
0.3s
[CV 3/5; 37/39] START n_neighbors=37…
[CV 3/5; 34/39] END n_neighbors=34;, score=(train=0.601, test=0.580) total time=
0.3s
[CV 4/5; 37/39] START n_neighbors=37…
[CV 5/5; 34/39] END n_neighbors=34;, score=(train=0.600, test=0.581) total time=
0.2s
[CV 5/5; 37/39] START n_neighbors=37…
[CV 2/5; 35/39] END n_neighbors=35;, score=(train=0.601, test=0.593) total time=
0.3s
[CV 1/5; 38/39] START n_neighbors=38…
[CV 1/5; 35/39] END n_neighbors=35;, score=(train=0.604, test=0.588) total time=
0.3s
[CV 2/5; 38/39] START n_neighbors=38…
[CV 4/5; 34/39] END n_neighbors=34;, score=(train=0.597, test=0.572) total time=
0.3s
```

```
[CV 3/5; 38/39] START n_neighbors=38…
[CV 3/5; 35/39] END n_neighbors=35;, score=(train=0.599, test=0.580) total time=
0.2s
[CV 4/5; 38/39] START n_neighbors=38…
[CV 4/5; 35/39] END n_neighbors=35;, score=(train=0.596, test=0.569) total time=
0.3s
[CV 5/5; 38/39] START n_neighbors=38…
[CV 1/5; 36/39] END n_neighbors=36;, score=(train=0.601, test=0.590) total time=
0.2s
[CV 1/5; 39/39] START n_neighbors=39…
[CV 5/5; 35/39] END n_neighbors=35;, score=(train=0.597, test=0.584) total time=
0.2s
[CV 2/5; 39/39] START n_neighbors=39…
[CV 2/5; 36/39] END n_neighbors=36;, score=(train=0.601, test=0.593) total time=
0.2s
[CV 3/5; 39/39] START n_neighbors=39…
[CV 4/5; 36/39] END n_neighbors=36;, score=(train=0.596, test=0.568) total time=
0.2s
[CV 4/5; 39/39] START n_neighbors=39…
[CV 3/5; 36/39] END n_neighbors=36;, score=(train=0.596, test=0.578) total time=
0.2s
[CV 5/5; 39/39] START n_neighbors=39…
[CV 5/5; 36/39] END n_neighbors=36;, score=(train=0.595, test=0.579) total time=
0.2s
[CV 2/5; 37/39] END n_neighbors=37;, score=(train=0.598, test=0.591) total time=
0.2s
[CV 1/5; 37/39] END n_neighbors=37;, score=(train=0.599, test=0.589) total time=
0.3s
[CV 3/5; 37/39] END n_neighbors=37;, score=(train=0.594, test=0.574) total time=
0.2s
[CV 1/5; 38/39] END n_neighbors=38;, score=(train=0.596, test=0.585) total time=
0.3s
[CV 2/5; 38/39] END n_neighbors=38;, score=(train=0.597, test=0.590) total time=
0.2s
[CV 5/5; 37/39] END n_neighbors=37;, score=(train=0.594, test=0.582) total time=
0.3s
[CV 4/5; 37/39] END n_neighbors=37;, score=(train=0.594, test=0.564) total time=
0.3s
[CV 5/5; 38/39] END n_neighbors=38;, score=(train=0.594, test=0.580) total time=
0.3s
[CV 4/5; 38/39] END n_neighbors=38;, score=(train=0.592, test=0.563) total time=
0.2s
[CV 3/5; 38/39] END n_neighbors=38;, score=(train=0.592, test=0.574) total time=
0.3s
[CV 3/5; 39/39] END n_neighbors=39;, score=(train=0.590, test=0.573) total time=
0.2s
[CV 1/5; 39/39] END n_neighbors=39;, score=(train=0.595, test=0.584) total time=
0.2s
```

```
[CV 2/5; 39/39] END n_neighbors=39;, score=(train=0.595, test=0.587) total time=
0.2s
[CV 5/5; 39/39] END n_neighbors=39;, score=(train=0.592, test=0.575) total time=
0.2s
[CV 4/5; 39/39] END n_neighbors=39;, score=(train=0.591, test=0.562) total time=
0.2s
{'n_neighbors': 3}
0.6546017409490729
```

[ ]:

```python
cv_results = pd.DataFrame(optimal_parameters.cv_results_)
# print(cv_results)
# # converting C to numeric type for plotting on x-axis
cv_results['param_n_neighbors']=cv_results['param_n_neighbors'].astype('int')
plt.figure(figsize=(16,8))

plt.subplot(131)
plt.plot(cv_results["param_n_neighbors"], cv_results["mean_test_score"])
plt.plot(cv_results["param_n_neighbors"], cv_results["mean_train_score"])
plt.xlabel('Neighbors')
plt.ylabel('Accuracy')
plt.title("KNN")
plt.ylim([0.20, 1])
plt.legend(['test accuracy', 'train accuracy'], loc='lower left')
plt.show()
```

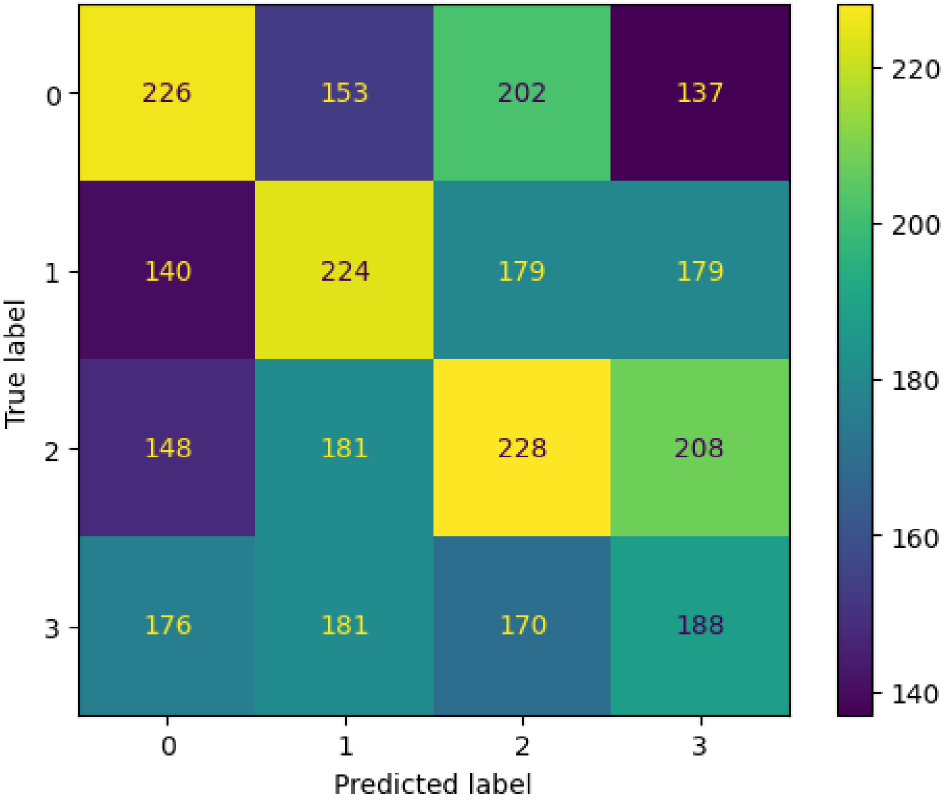


[ ]:

```
from sklearn.neighbors import NearestCentroid

ncc_model = NearestCentroid()
ncc_model.fit(X_train, y_train)
y_pred_test = ncc_model.predict(X_test)
y_pred_train = ncc_model.predict(X_train)
```

```
print("Train accuracy: ",metrics.accuracy_score(y_true=y_train,␣
 ↪y_pred=y_pred_train),"\n")
print("Test accuracy:", metrics.accuracy_score(y_true=y_test,␣
 ↪y_pred=y_pred_test), "\n")
print(f"Test Set Score : {ncc_model.score(X_test, y_test) * 100} %")
ConfusionMatrixDisplay.from_predictions(y_test, y_pred_test)
plt.show()
```

```
Train accuracy:  0.33375199817309886

Test accuracy: 0.2965753424657534

Test Set Score : 29.65753424657534 %
```


True label
Predicted label