Assignment 1A - Question 2

Classification

In [1]:

```
import pandas
import numpy
import matplotlib.pyplot as plt
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import plot_confusion_matrix
from sklearn.svm import SVC, NuSVC
from sklearn.multiclass import OneVsRestClassifier, OneVsOneClassifier
from scipy.stats import norm
from sklearn import tree
```

Import files

Files imported using pandas.read_csv

Check values using .head()

In [2]:

```
train = pandas.read_csv('CAB420_Assessment_1A_Data/Data/Q2/training.csv')
test = pandas.read_csv('CAB420_Assessment_1A_Data/Data/Q2/testing.csv')
train.head()
```

Out[2]:

	class	b1	b2	b3	b4	b5	b6	b7	b8	b9	...	pred_minus_obs_H_b9	pred_minus_ob
0	d	39	36	57	91	59	101	93	27	60	...	-2.36	
1	h	84	30	57	112	51	98	92	26	62	...	-2.26	
2	s	53	25	49	99	51	93	84	26	58	...	-1.46	
3	s	59	26	49	103	47	92	82	25	56	...	2.68	
4	d	57	49	66	103	64	106	114	28	59	...	-2.94	

5 rows × 28 columns

In [3]:

```
test.head()
```

Out[3]:

	class	b1	b2	b3	b4	b5	b6	b7	b8	b9	...	pred_minus_obs_H_b9	pred_minus_ol
0	d	67	51	68	115	69	111	136	31	67	...	-9.17	
1	s	67	28	51	99	50	97	82	26	59	...	-2.25	
2	s	63	26	50	95	49	91	81	26	57	...	-0.44	
3	d	63	42	63	97	66	108	111	28	59	...	-2.34	
4	s	46	27	50	83	51	90	76	26	56	...	1.25	

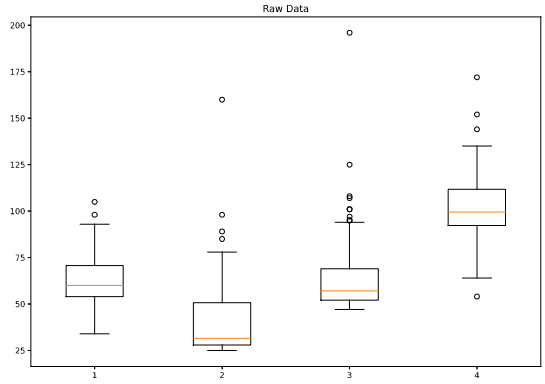
5 rows × 28 columns

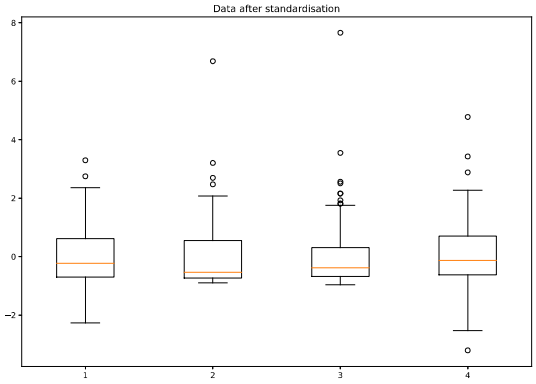
Plot data for train and test to Box Plot and standardised Box Plot

Code taken from Week 3 Example 1

In [4]:

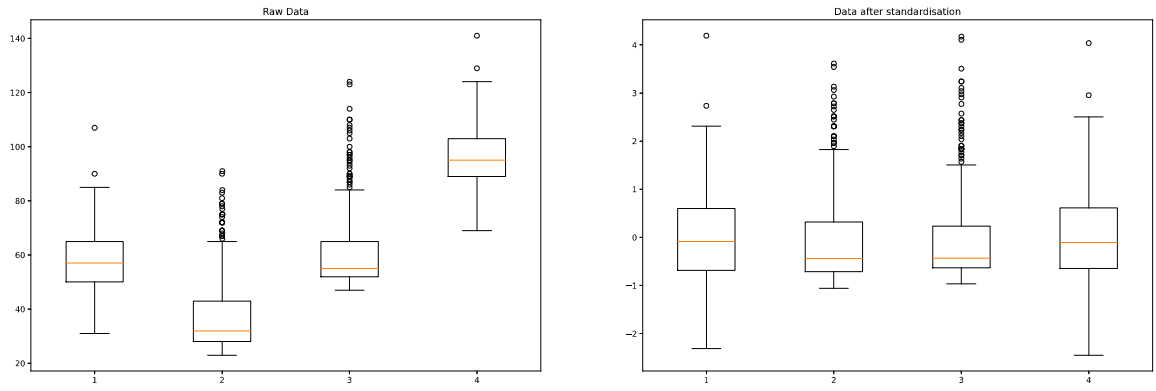
```
# get the data, we'll just grab 2 dimensions
X = train.iloc[:, 1:5].to_numpy()
train.iloc[:, 0] = train.iloc[:, 0].astype("category")
Y = train.iloc[:, 0].cat.codes.to_numpy()

# plot box plot for the data
fig = plt.figure(figsize=[25, 8])
ax = fig.add_subplot(1, 2, 1)
ax.boxplot(X)
ax.set_title('Raw Data')

# standardise data
mu = numpy.mean(X, 0)
sigma = numpy.std(X, 0)
X = (X - mu) / sigma

# box plot after standardisation
ax = fig.add_subplot(1, 2, 2)
ax.boxplot(X)
ax.set_title('Data after standardisation');

X_train = X
Y_train = Y
```





In [5]:

```
# get the data, we'll just grab 2 dimensions
X = test.iloc[:, 1:5].to_numpy()
test.iloc[:, 0] = test.iloc[:, 0].astype("category")
Y = test.iloc[:, 0].cat.codes.to_numpy()

# plot box plot for the data
fig = plt.figure(figsize=[25, 8])
ax = fig.add_subplot(1, 2, 1)
ax.boxplot(X)
ax.set_title('Raw Data')

# standardise data
mu = numpy.mean(X, 0)
sigma = numpy.std(X, 0)
X = (X - mu) / sigma

# box plot after standardisation
ax = fig.add_subplot(1, 2, 2)
ax.boxplot(X)
ax.set_title('Data after standardisation');
```



Split test data into testing and validation sets

Split testing set (50%) and validation set (50%)

Code taken from Week 3 Example 1

In [6]:

```
X_test, X_val, Y_test, Y_val = train_test_split(X, Y, test_size=0.7, random_state=0)
```

Check train and test data to confirm size

In [7]:

```
X_train.shape
```

Out[7]:

```
(198, 4)
```

In [8]:

```
X_test.shape
```

Out[8]:

```
(97, 4)
```

Evaluate models using confusion matrix

Code taken from Week 3 Example 2

In [9]:

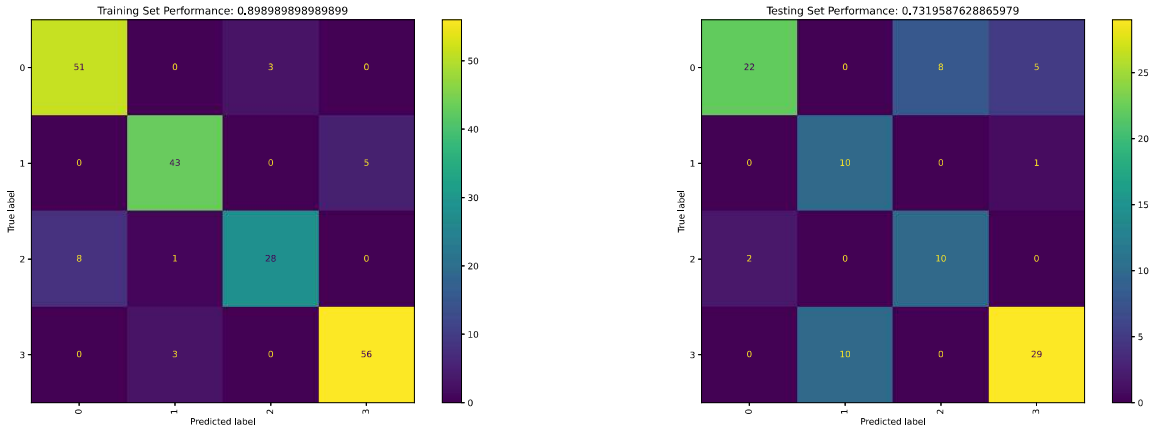
```
# function to do our eval for us, this is quite simple and will
# - create a figure
# - draw a confusion matrix for the trainign data in a sub-fig on the left
# - draw a confusion matrix for the testing data in a sub-fig on the right
# - compute the overall classification accuracy on the testing data
# this has simply been created as we're going to do this for each test that we run
def eval_model(model, X_train, Y_train, X_test, Y_test):
    fig = plt.figure(figsize=[25, 8])
    ax = fig.add_subplot(1, 2, 1)
    conf = plot_confusion_matrix(model, X_train, Y_train, normalize=None, xticks_rotati
on='vertical', ax=ax)
    pred = model.predict(X_train)
    conf.ax_.set_title('Training Set Performance: ' + str(sum(pred == Y_train)/len(Y_tr
ain)));
    ax = fig.add_subplot(1, 2, 2)
    conf = plot_confusion_matrix(model, X_test, Y_test, normalize=None, xticks_rotation
='vertical', ax=ax)
    pred = model.predict(X_test)
    conf.ax_.set_title('Testing Set Performance: ' + str(sum(pred == Y_test)/len(Y_test
)));
```

Evaluate basic SVC model

Code taken from Week 3 Example 2

In [10]:

```
svm = SVC()
svm.fit(X_train, Y_train)
eval_model(svm, X_train, Y_train, X_test, Y_test)
```



Set param grid to find best values

Code taken from Week 3 Example 2

used training data

In [11]:

```
param_grid = [
  {'C': [0.1, 1, 10, 100, 1000], 'kernel': ['linear'],'class_weight':['balanced','None'
]},
  {'C': [0.1, 1, 10, 100, 1000], 'gamma': [0.1, 0.01, 0.001, 0.0001, 'scale'], 'kernel'
: ['rbf'],'class_weight':['balanced','None']},
  {'C': [0.1, 1, 10, 100, 1000], 'degree': [3, 4, 5, 6], 'kernel': ['poly'],'class_weig
ht':['balanced','None']},
 ]
svm = SVC()

grid_search = GridSearchCV(svm, param_grid)
grid_search.fit(X_val, Y_val)
grid_search.cv_results_
```

```
alidation.py", line 593, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
```

```
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
```

```
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'
```

```
  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
```

```
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
```

```
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_search.py:918: UserWarning: One or more of the t
est scores are non-finite: [0.78917874        nan 0.79806763        nan 0.
77584541        nan
 0.78454106        nan 0.76705314        nan 0.68830918 0.50028986
 0.45188406 0.45188406 0.77603865        nan        nan        nan
        nan        nan 0.78927536 0.69710145 0.51333333 0.45188406
 0.78937198        nan        nan        nan        nan        nan
 0.77169082 0.78057971 0.70144928 0.44811594 0.77188406        nan
        nan        nan        nan        nan 0.75874396 0.78492754
 0.78492754 0.70144928 0.70618357        nan        nan        nan
        nan        nan 0.73681159 0.78048309 0.77613527 0.78937198
 0.71024155        nan        nan        nan        nan        nan
 0.57835749 0.55661836 0.56531401 0.54347826        nan        nan
        nan        nan 0.70995169 0.60898551 0.63545894 0.58724638
        nan        nan        nan        nan 0.78502415 0.63545894
 0.66618357 0.61777778        nan        nan        nan        nan
 0.7457971  0.67101449 0.73642512 0.62647343        nan        nan
        nan        nan 0.74144928 0.65294686 0.67082126 0.63555556
        nan        nan        nan        nan]
  warnings.warn(
```

Out[11]:

```
, --, --,
                   --, --, --, --, --, --, --, --, --, --, --, --, --, -
-,
                   --, --, --, --],
             mask=[ True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    True,  True, False, False, False, False, False, Fals
e,
                   False, False, False, False, False, False, False, Fals
e,
                   False, False, False, False, False, False, False, Fals
e,
                   False, False, False, False, False, False, False, Fals
e,
                   False, False, False, False, False, False, False, Fals
e,
                   False, False, False, False, False, False, False, Fals
e,
                   False, False, False, False,  True,  True,  True,  Tru
e,
                    True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    True,  True,  True,  True],
       fill_value='?',
            dtype=object),
 'param_degree': masked_array(data=[--, --, --, --, --, --, --, --, --, -
-, --, --, --, --,
                   --, --, --, --, --, --, --, --, --, --, --, --, --, -
-,
                   --, --, --, --, --, --, --, --, --, --, --, --, --, -
-,
                   --, --, --, --, --, --, --, --, --, --, --, --, --, -
-,
                   --, --, --, --, 3, 4, 5, 6, 3, 4, 5, 6, 3, 4, 5, 6, 3,
                   4, 5, 6, 3, 4, 5, 6, 3, 4, 5, 6, 3, 4, 5, 6, 3, 4, 5,
                   6, 3, 4, 5, 6, 3, 4, 5, 6],
             mask=[ True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    True,  True,  True,  True, False, False, False, Fals
e,
                   False, False, False, False, False, False, False, Fals
```

```
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False],
        fill_value='?',
             dtype=object),
 'params': [{'C': 0.1, 'class_weight': 'balanced', 'kernel': 'linear'},
  {'C': 0.1, 'class_weight': 'None', 'kernel': 'linear'},
  {'C': 1, 'class_weight': 'balanced', 'kernel': 'linear'},
  {'C': 1, 'class_weight': 'None', 'kernel': 'linear'},
  {'C': 10, 'class_weight': 'balanced', 'kernel': 'linear'},
  {'C': 10, 'class_weight': 'None', 'kernel': 'linear'},
  {'C': 100, 'class_weight': 'balanced', 'kernel': 'linear'},
  {'C': 100, 'class_weight': 'None', 'kernel': 'linear'},
  {'C': 1000, 'class_weight': 'balanced', 'kernel': 'linear'},
  {'C': 1000, 'class_weight': 'None', 'kernel': 'linear'},
  {'C': 0.1, 'class_weight': 'balanced', 'gamma': 0.1, 'kernel': 'rbf'},
  {'C': 0.1, 'class_weight': 'balanced', 'gamma': 0.01, 'kernel': 'rbf'},
  {'C': 0.1, 'class_weight': 'balanced', 'gamma': 0.001, 'kernel': 'rbf'},
  {'C': 0.1, 'class_weight': 'balanced', 'gamma': 0.0001, 'kernel': 'rb
f'},
  {'C': 0.1, 'class_weight': 'balanced', 'gamma': 'scale', 'kernel': 'rb
f'},
  {'C': 0.1, 'class_weight': 'None', 'gamma': 0.1, 'kernel': 'rbf'},
  {'C': 0.1, 'class_weight': 'None', 'gamma': 0.01, 'kernel': 'rbf'},
  {'C': 0.1, 'class_weight': 'None', 'gamma': 0.001, 'kernel': 'rbf'},
  {'C': 0.1, 'class_weight': 'None', 'gamma': 0.0001, 'kernel': 'rbf'},
  {'C': 0.1, 'class_weight': 'None', 'gamma': 'scale', 'kernel': 'rbf'},
  {'C': 1, 'class_weight': 'balanced', 'gamma': 0.1, 'kernel': 'rbf'},
  {'C': 1, 'class_weight': 'balanced', 'gamma': 0.01, 'kernel': 'rbf'},
  {'C': 1, 'class_weight': 'balanced', 'gamma': 0.001, 'kernel': 'rbf'},
  {'C': 1, 'class_weight': 'balanced', 'gamma': 0.0001, 'kernel': 'rbf'},
  {'C': 1, 'class_weight': 'balanced', 'gamma': 'scale', 'kernel': 'rbf'},
  {'C': 1, 'class_weight': 'None', 'gamma': 0.1, 'kernel': 'rbf'},
  {'C': 1, 'class_weight': 'None', 'gamma': 0.01, 'kernel': 'rbf'},
  {'C': 1, 'class_weight': 'None', 'gamma': 0.001, 'kernel': 'rbf'},
  {'C': 1, 'class_weight': 'None', 'gamma': 0.0001, 'kernel': 'rbf'},
  {'C': 1, 'class_weight': 'None', 'gamma': 'scale', 'kernel': 'rbf'},
  {'C': 10, 'class_weight': 'balanced', 'gamma': 0.1, 'kernel': 'rbf'},
  {'C': 10, 'class_weight': 'balanced', 'gamma': 0.01, 'kernel': 'rbf'},
  {'C': 10, 'class_weight': 'balanced', 'gamma': 0.001, 'kernel': 'rbf'},
  {'C': 10, 'class_weight': 'balanced', 'gamma': 0.0001, 'kernel': 'rbf'},
  {'C': 10, 'class_weight': 'balanced', 'gamma': 'scale', 'kernel': 'rb
f'},
  {'C': 10, 'class_weight': 'None', 'gamma': 0.1, 'kernel': 'rbf'},
  {'C': 10, 'class_weight': 'None', 'gamma': 0.01, 'kernel': 'rbf'},
  {'C': 10, 'class_weight': 'None', 'gamma': 0.001, 'kernel': 'rbf'},
  {'C': 10, 'class_weight': 'None', 'gamma': 0.0001, 'kernel': 'rbf'},
  {'C': 10, 'class_weight': 'None', 'gamma': 'scale', 'kernel': 'rbf'},
  {'C': 100, 'class_weight': 'balanced', 'gamma': 0.1, 'kernel': 'rbf'},
  {'C': 100, 'class_weight': 'balanced', 'gamma': 0.01, 'kernel': 'rbf'},
  {'C': 100, 'class_weight': 'balanced', 'gamma': 0.001, 'kernel': 'rbf'},
  {'C': 100, 'class_weight': 'balanced', 'gamma': 0.0001, 'kernel': 'rb
f'},
  {'C': 100, 'class_weight': 'balanced', 'gamma': 'scale', 'kernel': 'rb
f'},
  {'C': 100, 'class_weight': 'None', 'gamma': 0.1, 'kernel': 'rbf'},
```

```
    {'C': 100, 'class_weight': 'None', 'gamma': 0.01, 'kernel': 'rbf'},
    {'C': 100, 'class_weight': 'None', 'gamma': 0.001, 'kernel': 'rbf'},
    {'C': 100, 'class_weight': 'None', 'gamma': 0.0001, 'kernel': 'rbf'},
    {'C': 100, 'class_weight': 'None', 'gamma': 'scale', 'kernel': 'rbf'},
    {'C': 1000, 'class_weight': 'balanced', 'gamma': 0.1, 'kernel': 'rbf'},
    {'C': 1000, 'class_weight': 'balanced', 'gamma': 0.01, 'kernel': 'rbf'},
    {'C': 1000, 'class_weight': 'balanced', 'gamma': 0.001, 'kernel': 'rb
f'},
    {'C': 1000, 'class_weight': 'balanced', 'gamma': 0.0001, 'kernel': 'rb
f'},
    {'C': 1000, 'class_weight': 'balanced', 'gamma': 'scale', 'kernel': 'rb
f'},
    {'C': 1000, 'class_weight': 'None', 'gamma': 0.1, 'kernel': 'rbf'},
    {'C': 1000, 'class_weight': 'None', 'gamma': 0.01, 'kernel': 'rbf'},
    {'C': 1000, 'class_weight': 'None', 'gamma': 0.001, 'kernel': 'rbf'},
    {'C': 1000, 'class_weight': 'None', 'gamma': 0.0001, 'kernel': 'rbf'},
    {'C': 1000, 'class_weight': 'None', 'gamma': 'scale', 'kernel': 'rbf'},
    {'C': 0.1, 'class_weight': 'balanced', 'degree': 3, 'kernel': 'poly'},
    {'C': 0.1, 'class_weight': 'balanced', 'degree': 4, 'kernel': 'poly'},
    {'C': 0.1, 'class_weight': 'balanced', 'degree': 5, 'kernel': 'poly'},
    {'C': 0.1, 'class_weight': 'balanced', 'degree': 6, 'kernel': 'poly'},
    {'C': 0.1, 'class_weight': 'None', 'degree': 3, 'kernel': 'poly'},
    {'C': 0.1, 'class_weight': 'None', 'degree': 4, 'kernel': 'poly'},
    {'C': 0.1, 'class_weight': 'None', 'degree': 5, 'kernel': 'poly'},
    {'C': 0.1, 'class_weight': 'None', 'degree': 6, 'kernel': 'poly'},
    {'C': 1, 'class_weight': 'balanced', 'degree': 3, 'kernel': 'poly'},
    {'C': 1, 'class_weight': 'balanced', 'degree': 4, 'kernel': 'poly'},
    {'C': 1, 'class_weight': 'balanced', 'degree': 5, 'kernel': 'poly'},
    {'C': 1, 'class_weight': 'balanced', 'degree': 6, 'kernel': 'poly'},
    {'C': 1, 'class_weight': 'None', 'degree': 3, 'kernel': 'poly'},
    {'C': 1, 'class_weight': 'None', 'degree': 4, 'kernel': 'poly'},
    {'C': 1, 'class_weight': 'None', 'degree': 5, 'kernel': 'poly'},
    {'C': 1, 'class_weight': 'None', 'degree': 6, 'kernel': 'poly'},
    {'C': 10, 'class_weight': 'balanced', 'degree': 3, 'kernel': 'poly'},
    {'C': 10, 'class_weight': 'balanced', 'degree': 4, 'kernel': 'poly'},
    {'C': 10, 'class_weight': 'balanced', 'degree': 5, 'kernel': 'poly'},
    {'C': 10, 'class_weight': 'balanced', 'degree': 6, 'kernel': 'poly'},
    {'C': 10, 'class_weight': 'None', 'degree': 3, 'kernel': 'poly'},
    {'C': 10, 'class_weight': 'None', 'degree': 4, 'kernel': 'poly'},
    {'C': 10, 'class_weight': 'None', 'degree': 5, 'kernel': 'poly'},
    {'C': 10, 'class_weight': 'None', 'degree': 6, 'kernel': 'poly'},
    {'C': 100, 'class_weight': 'balanced', 'degree': 3, 'kernel': 'poly'},
    {'C': 100, 'class_weight': 'balanced', 'degree': 4, 'kernel': 'poly'},
    {'C': 100, 'class_weight': 'balanced', 'degree': 5, 'kernel': 'poly'},
    {'C': 100, 'class_weight': 'balanced', 'degree': 6, 'kernel': 'poly'},
    {'C': 100, 'class_weight': 'None', 'degree': 3, 'kernel': 'poly'},
    {'C': 100, 'class_weight': 'None', 'degree': 4, 'kernel': 'poly'},
    {'C': 100, 'class_weight': 'None', 'degree': 5, 'kernel': 'poly'},
    {'C': 100, 'class_weight': 'None', 'degree': 6, 'kernel': 'poly'},
    {'C': 1000, 'class_weight': 'balanced', 'degree': 3, 'kernel': 'poly'},
    {'C': 1000, 'class_weight': 'balanced', 'degree': 4, 'kernel': 'poly'},
    {'C': 1000, 'class_weight': 'balanced', 'degree': 5, 'kernel': 'poly'},
    {'C': 1000, 'class_weight': 'balanced', 'degree': 6, 'kernel': 'poly'},
    {'C': 1000, 'class_weight': 'None', 'degree': 3, 'kernel': 'poly'},
    {'C': 1000, 'class_weight': 'None', 'degree': 4, 'kernel': 'poly'},
    {'C': 1000, 'class_weight': 'None', 'degree': 5, 'kernel': 'poly'},
    {'C': 1000, 'class_weight': 'None', 'degree': 6, 'kernel': 'poly'}],
  'split0_test_score': array([0.86956522,        nan, 0.82608696,        na
n, 0.80434783,
               nan, 0.80434783,        nan, 0.73913043,        nan,
        0.69565217, 0.5       , 0.47826087, 0.47826087, 0.80434783,
```

```
                nan,         nan,         nan,         nan,         nan,
         0.84782609, 0.69565217, 0.47826087, 0.47826087, 0.82608696,
                nan,         nan,         nan,         nan,         nan,
         0.7826087 , 0.82608696, 0.69565217, 0.15217391, 0.76086957,
                nan,         nan,         nan,         nan,         nan,
         0.76086957, 0.7826087 , 0.82608696, 0.69565217, 0.69565217,
                nan,         nan,         nan,         nan,         nan,
         0.69565217, 0.7826087 , 0.80434783, 0.82608696, 0.67391304,
                nan,         nan,         nan,         nan,         nan,
         0.67391304, 0.54347826, 0.58695652, 0.56521739,         nan,
                nan,         nan,         nan, 0.7826087 , 0.67391304,
         0.69565217, 0.63043478,         nan,         nan,         nan,
                nan, 0.7826087 , 0.67391304, 0.67391304, 0.63043478,
                nan,         nan,         nan,         nan, 0.69565217,
         0.63043478, 0.7826087 , 0.67391304,         nan,         nan,
                nan,         nan, 0.69565217, 0.73913043, 0.65217391,
         0.73913043,         nan,         nan,         nan,         nan]),
  'split1_test_score': array([0.69565217,         nan, 0.7173913 ,         na
n, 0.73913043,
                nan, 0.76086957,         nan, 0.76086957,         nan,
         0.65217391, 0.5       , 0.43478261, 0.43478261, 0.73913043,
                nan,         nan,         nan,         nan,         nan,
         0.7173913 , 0.67391304, 0.56521739, 0.43478261, 0.7173913 ,
                nan,         nan,         nan,         nan,         nan,
         0.7173913 , 0.69565217, 0.69565217, 0.56521739, 0.69565217,
                nan,         nan,         nan,         nan,         nan,
         0.67391304, 0.7173913 , 0.69565217, 0.69565217, 0.60869565,
                nan,         nan,         nan,         nan,         nan,
         0.69565217, 0.76086957, 0.7173913 , 0.69565217, 0.69565217,
                nan,         nan,         nan,         nan,         nan,
         0.63043478, 0.65217391, 0.65217391, 0.60869565,         nan,
                nan,         nan,         nan, 0.76086957, 0.65217391,
         0.67391304, 0.65217391,         nan,         nan,         nan,
                nan, 0.76086957, 0.67391304, 0.69565217, 0.7173913 ,
                nan,         nan,         nan,         nan, 0.69565217,
         0.65217391, 0.73913043, 0.65217391,         nan,         nan,
                nan,         nan, 0.69565217, 0.63043478, 0.7173913 ,
         0.63043478,         nan,         nan,         nan,         nan]),
  'split2_test_score': array([0.86956522,         nan, 0.89130435,         na
n, 0.89130435,
                nan, 0.91304348,         nan, 0.91304348,         nan,
         0.7826087 , 0.43478261, 0.41304348, 0.41304348, 0.84782609,
                nan,         nan,         nan,         nan,         nan,
         0.84782609, 0.7826087 , 0.45652174, 0.41304348, 0.84782609,
                nan,         nan,         nan,         nan,         nan,
         0.86956522, 0.84782609, 0.7826087 , 0.45652174, 0.86956522,
                nan,         nan,         nan,         nan,         nan,
         0.84782609, 0.89130435, 0.86956522, 0.7826087 , 0.80434783,
                nan,         nan,         nan,         nan,         nan,
         0.82608696, 0.84782609, 0.84782609, 0.86956522, 0.82608696,
                nan,         nan,         nan,         nan,         nan,
         0.56521739, 0.56521739, 0.56521739, 0.54347826,         nan,
                nan,         nan,         nan, 0.7173913 , 0.65217391,
         0.65217391, 0.58695652,         nan,         nan,         nan,
                nan, 0.82608696, 0.67391304, 0.73913043, 0.65217391,
                nan,         nan,         nan,         nan, 0.80434783,
         0.73913043, 0.7826087 , 0.7173913 ,         nan,         nan,
                nan,         nan, 0.7826087 , 0.7173913 , 0.69565217,
         0.63043478,         nan,         nan,         nan,         nan]),
  'split3_test_score': array([0.77777778,         nan, 0.8       ,         na
n, 0.71111111,
```

```
                nan, 0.73333333,        nan, 0.71111111,        nan,
        0.64444444, 0.53333333, 0.46666667, 0.46666667, 0.73333333,
                nan,        nan,        nan,        nan,        nan,
        0.8       , 0.64444444, 0.53333333, 0.46666667, 0.8       ,
                nan,        nan,        nan,        nan,        nan,
        0.75555556, 0.8       , 0.64444444, 0.55555556, 0.75555556,
                nan,        nan,        nan,        nan,        nan,
        0.73333333, 0.8       , 0.77777778, 0.64444444, 0.73333333,
                nan,        nan,        nan,        nan,        nan,
        0.73333333, 0.75555556, 0.77777778, 0.8       , 0.66666667,
                nan,        nan,        nan,        nan,        nan,
        0.51111111, 0.51111111, 0.51111111, 0.48888889,        nan,
                nan,        nan,        nan, 0.66666667, 0.53333333,
        0.55555556, 0.53333333,        nan,        nan,        nan,
                nan, 0.77777778, 0.55555556, 0.6       , 0.55555556,
                nan,        nan,        nan,        nan, 0.77777778,
        0.68888889, 0.68888889, 0.53333333,        nan,        nan,
                nan,        nan, 0.73333333, 0.57777778, 0.64444444,
        0.55555556,        nan,        nan,        nan,        nan]),
 'split4_test_score': array([0.73333333,        nan, 0.75555556,        na
n, 0.73333333,
                nan, 0.71111111,        nan, 0.71111111,        nan,
        0.66666667, 0.53333333, 0.46666667, 0.46666667, 0.75555556,
                nan,        nan,        nan,        nan,        nan,
        0.73333333, 0.68888889, 0.53333333, 0.46666667, 0.75555556,
                nan,        nan,        nan,        nan,        nan,
        0.73333333, 0.73333333, 0.68888889, 0.51111111, 0.77777778,
                nan,        nan,        nan,        nan,        nan,
        0.77777778, 0.73333333, 0.75555556, 0.68888889, 0.68888889,
                nan,        nan,        nan,        nan,        nan,
        0.73333333, 0.75555556, 0.73333333, 0.75555556, 0.68888889,
                nan,        nan,        nan,        nan,        nan,
        0.51111111, 0.51111111, 0.51111111, 0.51111111,        nan,
                nan,        nan,        nan, 0.62222222, 0.53333333,
        0.6       , 0.53333333,        nan,        nan,        nan,
                nan, 0.77777778, 0.6       , 0.62222222, 0.53333333,
                nan,        nan,        nan,        nan, 0.75555556,
        0.64444444, 0.68888889, 0.55555556,        nan,        nan,
                nan,        nan, 0.8       , 0.6       , 0.64444444,
        0.62222222,        nan,        nan,        nan,        nan]),
 'mean_test_score': array([0.78917874,        nan, 0.79806763,        nan,
0.77584541,
                nan, 0.78454106,        nan, 0.76705314,        nan,
        0.68830918, 0.50028986, 0.45188406, 0.45188406, 0.77603865,
                nan,        nan,        nan,        nan,        nan,
        0.78927536, 0.69710145, 0.51333333, 0.45188406, 0.78937198,
                nan,        nan,        nan,        nan,        nan,
        0.77169082, 0.78057971, 0.70144928, 0.44811594, 0.77188406,
                nan,        nan,        nan,        nan,        nan,
        0.75874396, 0.78492754, 0.78492754, 0.70144928, 0.70618357,
                nan,        nan,        nan,        nan,        nan,
        0.73681159, 0.78048309, 0.77613527, 0.78937198, 0.71024155,
                nan,        nan,        nan,        nan,        nan,
        0.57835749, 0.55661836, 0.56531401, 0.54347826,        nan,
                nan,        nan,        nan, 0.70995169, 0.60898551,
        0.63545894, 0.58724638,        nan,        nan,        nan,
                nan, 0.78502415, 0.63545894, 0.66618357, 0.61777778,
                nan,        nan,        nan,        nan, 0.7457971 ,
        0.67101449, 0.73642512, 0.62647343,        nan,        nan,
                nan,        nan, 0.74144928, 0.65294686, 0.67082126,
        0.63555556,        nan,        nan,        nan,        nan]),
```

```
 'std_test_score': array([0.07059728,        nan, 0.05966213,        nan,
0.06555753,
               nan, 0.07138586,        nan, 0.07535266,        nan,
        0.05028835, 0.03598641, 0.02422328, 0.02422328, 0.04372725,
               nan,        nan,        nan,        nan,        nan,
        0.05525979, 0.04624175, 0.03987378, 0.02422328, 0.04733412,
               nan,        nan,        nan,        nan,        nan,
        0.05362589, 0.05729754, 0.0448584 , 0.15289   , 0.05617043,
               nan,        nan,        nan,        nan,        nan,
        0.05682326, 0.06127265, 0.05956771, 0.0448584 , 0.0636981 ,
               nan,        nan,        nan,        nan,        nan,
        0.04771265, 0.03511994, 0.04737315, 0.05969732, 0.05883447,
               nan,        nan,        nan,        nan,        nan,
        0.06489949, 0.05199863, 0.05270476, 0.04184144,        nan,
               nan,        nan,        nan, 0.05915585, 0.06227771,
        0.05103572, 0.04877323,        nan,        nan,        nan,
               nan, 0.0218195 , 0.04914884, 0.05012435, 0.06673622,
               nan,        nan,        nan,        nan, 0.04376118,
        0.03915846, 0.04193459, 0.07054291,        nan,        nan,
               nan,        nan, 0.04331952, 0.06409563, 0.03008169,
        0.05892025,        nan,        nan,        nan,        nan]),
 'rank_test_score': array([  5,  69,   1,  70,  14,  71,   9,  72,  17,  7
3,  29,  46,  47,
         47,  13,  74,  75,  76,  77,  78,   4,  28,  45,  47,   2,  79,
         80,  81,  82,  84,  16,  10,  26,  50,  15,  99,  85,  86,  87,
         88,  18,   7,   7,  26,  25,  89,  90,  91,  92,  67,  21,  11,
         12,   2,  23,  94,  95,  96,  97,  98,  41,  43,  42,  44,  68,
         83,  66,  52,  24,  39,  36,  40,  57,  62,  63,  64,   6,  35,
         32,  38,  58,  56,  55,  51,  19,  30,  22,  37,  54,  59,  60,
         61,  20,  33,  31,  34,  65,  53,  93, 100], dtype=int32)}
```

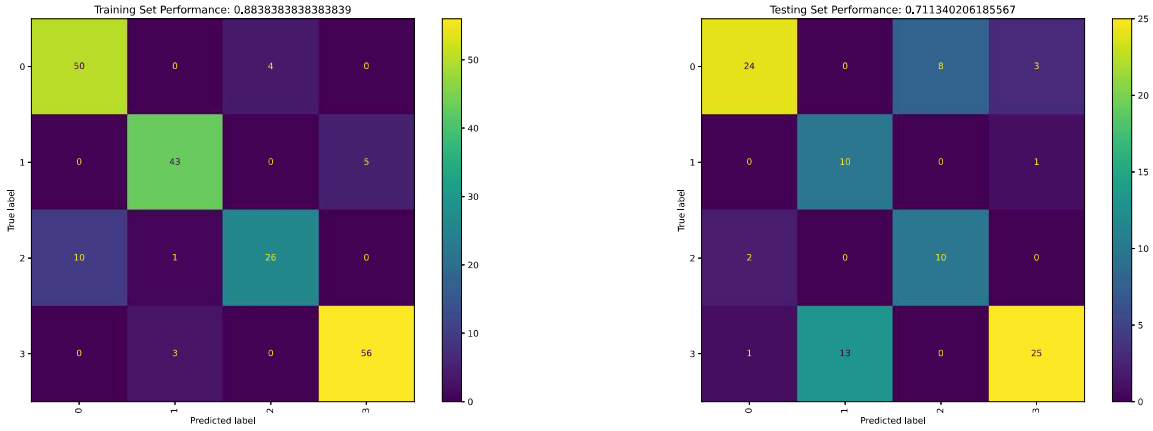
Use best values found from param grid as values for SVC

Code taken from Week 3 Example 2

In [12]:

```
best_system = numpy.argmin(grid_search.cv_results_['rank_test_score'])
params = grid_search.cv_results_['params'][best_system]
print(params)
svm = SVC().set_params(**params)
svm.fit(X_train, Y_train)
eval_model(svm, X_train, Y_train, X_test, Y_test)
```

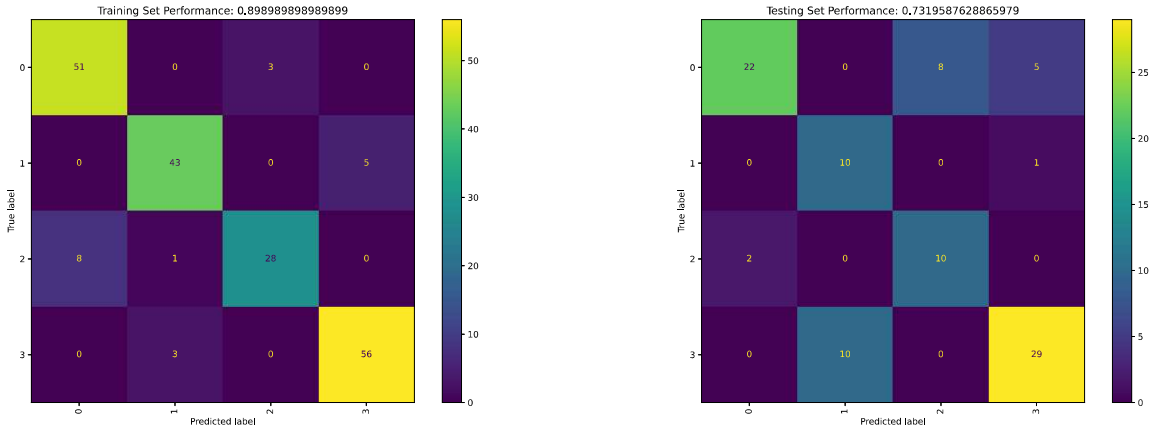
```
{'C': 1, 'class_weight': 'balanced', 'kernel': 'linear'}
```



validate using the validation set

In [13]:

```
svm2 = SVC()
svm2.fit(X_train, Y_train)
eval_model(svm2, X_train, Y_train, X_test, Y_test)
```

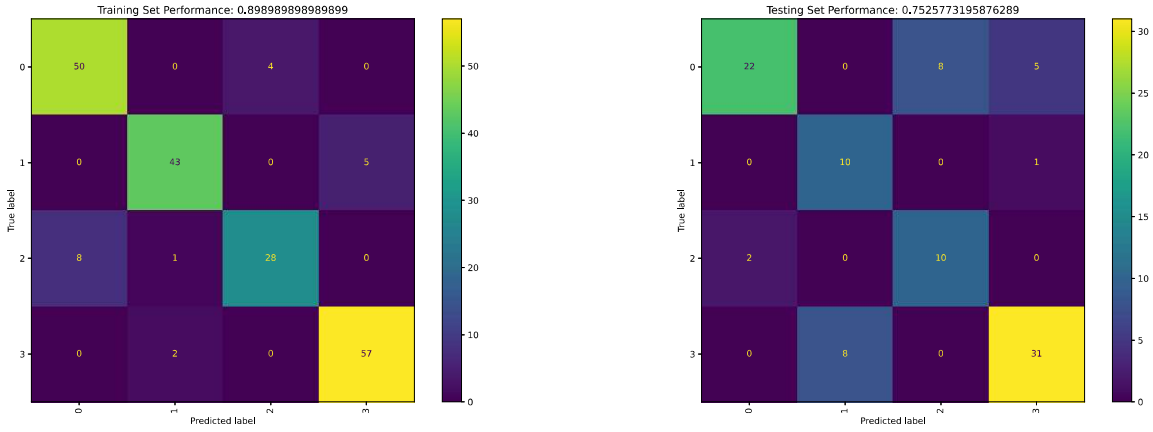


Evaluate model using NuSVC

Code taken from Week 3 Example 2

In [14]:

```
nu_svm = NuSVC()
nu_svm.fit(X_train, Y_train)
eval_model(nu_svm, X_train, Y_train, X_test, Y_test)
```



In [15]:

```
param_grid = [
  {'nu':[0.1,0.15,0.2,0.25], 'kernel': ['linear'],'class_weight':['balanced','None']},
  {'nu':[0.1,0.15,0.2,0.25], 'gamma': [0.15, 0.125, 0.1, 0.075, 0.05, 0.0025], 'kernel'
: ['rbf'],'class_weight':['balanced','None']},
  {'nu':[0.1,0.15,0.2,0.25], 'degree': [3, 4, 5, 6], 'kernel': ['poly'],'class_weight':
['balanced','None']},
 ]
svm = NuSVC()

grid_search = GridSearchCV(svm, param_grid)
grid_search.fit(X_val, Y_val)
grid_search.cv_results_
```

```
ck (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
```

```
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
```

```
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
```

```
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
```

```
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
```

```
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 173, in fit
    y = self._validate_targets(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\svm\_base.py", line 555, in _validate_targets
    self.class_weight_ = compute_class_weight(self.class_weight,
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\validation.py", line 63, in inner_f
    return f(*args, **kwargs)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\utils\class_weight.py", line 63, in compute_class_weight
    raise ValueError("class_weight must be dict, 'balanced', or None,"
ValueError: class_weight must be dict, 'balanced', or None, got: 'None'

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_search.py:918: UserWarning: One or more of the t
est scores are non-finite: [0.47806763 0.71884058 0.7236715  0.78898551
nan        nan
        nan        nan 0.74135266 0.74985507 0.7847343  0.78888889
 0.74966184 0.75855072 0.77594203 0.78444444 0.73246377 0.74550725
 0.77140097 0.78009662 0.7057971  0.73671498 0.78463768 0.8021256
 0.68425121 0.75874396 0.78038647 0.79352657 0.62318841 0.7715942
 0.80202899 0.77584541        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan 0.68454106 0.76251208 0.77603865 0.77594203
 0.57487923 0.62695652 0.60995169 0.61400966 0.67082126 0.66599034
 0.74956522 0.75835749 0.59169082 0.60492754 0.51700483 0.62251208
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan]
  warnings.warn(
```

Out[15]:

```
e, False, False, False, False, False,
                    False, False, False, False, False, False, False, Fals
e],
        fill_value='?',
             dtype=object),
 'param_nu': masked_array(data=[0.1, 0.15, 0.2, 0.25, 0.1, 0.15, 0.2, 0.2
5, 0.1, 0.15,
                    0.2, 0.25, 0.1, 0.15, 0.2, 0.25, 0.1, 0.15, 0.2, 0.25,
                    0.1, 0.15, 0.2, 0.25, 0.1, 0.15, 0.2, 0.25, 0.1, 0.15,
                    0.2, 0.25, 0.1, 0.15, 0.2, 0.25, 0.1, 0.15, 0.2, 0.25,
                    0.1, 0.15, 0.2, 0.25, 0.1, 0.15, 0.2, 0.25, 0.1, 0.15,
                    0.2, 0.25, 0.1, 0.15, 0.2, 0.25, 0.1, 0.15, 0.2, 0.25,
                    0.1, 0.15, 0.2, 0.25, 0.1, 0.15, 0.2, 0.25, 0.1, 0.15,
                    0.2, 0.25, 0.1, 0.15, 0.2, 0.25, 0.1, 0.15, 0.2, 0.25,
                    0.1, 0.15, 0.2, 0.25, 0.1, 0.15, 0.2, 0.25],
              mask=[False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e],
        fill_value='?',
             dtype=object),
 'param_gamma': masked_array(data=[--, --, --, --, --, --, --, --, 0.15,
0.15, 0.15, 0.15,
                    0.125, 0.125, 0.125, 0.125, 0.1, 0.1, 0.1, 0.1, 0.075,
                    0.075, 0.075, 0.075, 0.05, 0.05, 0.05, 0.05, 0.0025,
                    0.0025, 0.0025, 0.0025, 0.15, 0.15, 0.15, 0.15, 0.125,
                    0.125, 0.125, 0.125, 0.1, 0.1, 0.1, 0.1, 0.075, 0.075,
                    0.075, 0.075, 0.05, 0.05, 0.05, 0.05, 0.0025, 0.0025,
                    0.0025, 0.0025, --, --, --, --, --, --, --, --, --, -
-,
                    --, --, --, --, --, --, --, --, --, --, --, --, --, -
-,
                    --, --, --, --, --, --, --, --],
              mask=[ True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
```

```
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                     True,  True,  True,  True,  True,  True,  True,  Tru
e,
                     True,  True,  True,  True,  True,  True,  True,  Tru
e,
                     True,  True,  True,  True,  True,  True,  True,  Tru
e,
                     True,  True,  True,  True,  True,  True,  True,  Tru
e],
             fill_value='?',
                  dtype=object),
 'param_degree': masked_array(data=[--, --, --, --, --, --, --, --, --, --
-, --, --, --, --,
                    --, --, --, --, --, --, --, --, --, --, --, --, --, --
-,
                    --, --, --, --, --, --, --, --, --, --, --, --, --, --
-,
                    --, --, --, --, --, --, --, --, --, --, --, --, --, --
-,
                    3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6, 3, 3,
                    3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6],
                mask=[ True,  True,  True,  True,  True,  True,  True,  Tru
e,
                     True,  True,  True,  True,  True,  True,  True,  Tru
e,
                     True,  True,  True,  True,  True,  True,  True,  Tru
e,
                     True,  True,  True,  True,  True,  True,  True,  Tru
e,
                     True,  True,  True,  True,  True,  True,  True,  Tru
e,
                     True,  True,  True,  True,  True,  True,  True,  Tru
e,
                     True,  True,  True,  True,  True,  True,  True,  Tru
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e],
             fill_value='?',
                  dtype=object),
  'params': [{'class_weight': 'balanced', 'kernel': 'linear', 'nu': 0.1},
   {'class_weight': 'balanced', 'kernel': 'linear', 'nu': 0.15},
   {'class_weight': 'balanced', 'kernel': 'linear', 'nu': 0.2},
   {'class_weight': 'balanced', 'kernel': 'linear', 'nu': 0.25},
   {'class_weight': 'None', 'kernel': 'linear', 'nu': 0.1},
   {'class_weight': 'None', 'kernel': 'linear', 'nu': 0.15},
   {'class_weight': 'None', 'kernel': 'linear', 'nu': 0.2},
   {'class_weight': 'None', 'kernel': 'linear', 'nu': 0.25},
   {'class_weight': 'balanced', 'gamma': 0.15, 'kernel': 'rbf', 'nu': 0.1},
   {'class_weight': 'balanced', 'gamma': 0.15, 'kernel': 'rbf', 'nu': 0.1
5},
```

```
    {'class_weight': 'balanced', 'gamma': 0.15, 'kernel': 'rbf', 'nu': 0.2},
    {'class_weight': 'balanced', 'gamma': 0.15, 'kernel': 'rbf', 'nu': 0.2
5},
    {'class_weight': 'balanced', 'gamma': 0.125, 'kernel': 'rbf', 'nu': 0.
1},
    {'class_weight': 'balanced', 'gamma': 0.125, 'kernel': 'rbf', 'nu': 0.1
5},
    {'class_weight': 'balanced', 'gamma': 0.125, 'kernel': 'rbf', 'nu': 0.
2},
    {'class_weight': 'balanced', 'gamma': 0.125, 'kernel': 'rbf', 'nu': 0.2
5},
    {'class_weight': 'balanced', 'gamma': 0.1, 'kernel': 'rbf', 'nu': 0.1},
    {'class_weight': 'balanced', 'gamma': 0.1, 'kernel': 'rbf', 'nu': 0.15},
    {'class_weight': 'balanced', 'gamma': 0.1, 'kernel': 'rbf', 'nu': 0.2},
    {'class_weight': 'balanced', 'gamma': 0.1, 'kernel': 'rbf', 'nu': 0.25},
    {'class_weight': 'balanced', 'gamma': 0.075, 'kernel': 'rbf', 'nu': 0.
1},
    {'class_weight': 'balanced', 'gamma': 0.075, 'kernel': 'rbf', 'nu': 0.1
5},
    {'class_weight': 'balanced', 'gamma': 0.075, 'kernel': 'rbf', 'nu': 0.
2},
    {'class_weight': 'balanced', 'gamma': 0.075, 'kernel': 'rbf', 'nu': 0.2
5},
    {'class_weight': 'balanced', 'gamma': 0.05, 'kernel': 'rbf', 'nu': 0.1},
    {'class_weight': 'balanced', 'gamma': 0.05, 'kernel': 'rbf', 'nu': 0.1
5},
    {'class_weight': 'balanced', 'gamma': 0.05, 'kernel': 'rbf', 'nu': 0.2},
    {'class_weight': 'balanced', 'gamma': 0.05, 'kernel': 'rbf', 'nu': 0.2
5},
    {'class_weight': 'balanced', 'gamma': 0.0025, 'kernel': 'rbf', 'nu': 0.
1},
    {'class_weight': 'balanced', 'gamma': 0.0025, 'kernel': 'rbf', 'nu': 0.1
5},
    {'class_weight': 'balanced', 'gamma': 0.0025, 'kernel': 'rbf', 'nu': 0.
2},
    {'class_weight': 'balanced', 'gamma': 0.0025, 'kernel': 'rbf', 'nu': 0.2
5},
    {'class_weight': 'None', 'gamma': 0.15, 'kernel': 'rbf', 'nu': 0.1},
    {'class_weight': 'None', 'gamma': 0.15, 'kernel': 'rbf', 'nu': 0.15},
    {'class_weight': 'None', 'gamma': 0.15, 'kernel': 'rbf', 'nu': 0.2},
    {'class_weight': 'None', 'gamma': 0.15, 'kernel': 'rbf', 'nu': 0.25},
    {'class_weight': 'None', 'gamma': 0.125, 'kernel': 'rbf', 'nu': 0.1},
    {'class_weight': 'None', 'gamma': 0.125, 'kernel': 'rbf', 'nu': 0.15},
    {'class_weight': 'None', 'gamma': 0.125, 'kernel': 'rbf', 'nu': 0.2},
    {'class_weight': 'None', 'gamma': 0.125, 'kernel': 'rbf', 'nu': 0.25},
    {'class_weight': 'None', 'gamma': 0.1, 'kernel': 'rbf', 'nu': 0.1},
    {'class_weight': 'None', 'gamma': 0.1, 'kernel': 'rbf', 'nu': 0.15},
    {'class_weight': 'None', 'gamma': 0.1, 'kernel': 'rbf', 'nu': 0.2},
    {'class_weight': 'None', 'gamma': 0.1, 'kernel': 'rbf', 'nu': 0.25},
    {'class_weight': 'None', 'gamma': 0.075, 'kernel': 'rbf', 'nu': 0.1},
    {'class_weight': 'None', 'gamma': 0.075, 'kernel': 'rbf', 'nu': 0.15},
    {'class_weight': 'None', 'gamma': 0.075, 'kernel': 'rbf', 'nu': 0.2},
    {'class_weight': 'None', 'gamma': 0.075, 'kernel': 'rbf', 'nu': 0.25},
    {'class_weight': 'None', 'gamma': 0.05, 'kernel': 'rbf', 'nu': 0.1},
    {'class_weight': 'None', 'gamma': 0.05, 'kernel': 'rbf', 'nu': 0.15},
    {'class_weight': 'None', 'gamma': 0.05, 'kernel': 'rbf', 'nu': 0.2},
    {'class_weight': 'None', 'gamma': 0.05, 'kernel': 'rbf', 'nu': 0.25},
    {'class_weight': 'None', 'gamma': 0.0025, 'kernel': 'rbf', 'nu': 0.1},
    {'class_weight': 'None', 'gamma': 0.0025, 'kernel': 'rbf', 'nu': 0.15},
    {'class_weight': 'None', 'gamma': 0.0025, 'kernel': 'rbf', 'nu': 0.2},
    {'class_weight': 'None', 'gamma': 0.0025, 'kernel': 'rbf', 'nu': 0.25},
```

```
   {'class_weight': 'balanced', 'degree': 3, 'kernel': 'poly', 'nu': 0.1},
   {'class_weight': 'balanced', 'degree': 3, 'kernel': 'poly', 'nu': 0.15},
   {'class_weight': 'balanced', 'degree': 3, 'kernel': 'poly', 'nu': 0.2},
   {'class_weight': 'balanced', 'degree': 3, 'kernel': 'poly', 'nu': 0.25},
   {'class_weight': 'balanced', 'degree': 4, 'kernel': 'poly', 'nu': 0.1},
   {'class_weight': 'balanced', 'degree': 4, 'kernel': 'poly', 'nu': 0.15},
   {'class_weight': 'balanced', 'degree': 4, 'kernel': 'poly', 'nu': 0.2},
   {'class_weight': 'balanced', 'degree': 4, 'kernel': 'poly', 'nu': 0.25},
   {'class_weight': 'balanced', 'degree': 5, 'kernel': 'poly', 'nu': 0.1},
   {'class_weight': 'balanced', 'degree': 5, 'kernel': 'poly', 'nu': 0.15},
   {'class_weight': 'balanced', 'degree': 5, 'kernel': 'poly', 'nu': 0.2},
   {'class_weight': 'balanced', 'degree': 5, 'kernel': 'poly', 'nu': 0.25},
   {'class_weight': 'balanced', 'degree': 6, 'kernel': 'poly', 'nu': 0.1},
   {'class_weight': 'balanced', 'degree': 6, 'kernel': 'poly', 'nu': 0.15},
   {'class_weight': 'balanced', 'degree': 6, 'kernel': 'poly', 'nu': 0.2},
   {'class_weight': 'balanced', 'degree': 6, 'kernel': 'poly', 'nu': 0.25},
   {'class_weight': 'None', 'degree': 3, 'kernel': 'poly', 'nu': 0.1},
   {'class_weight': 'None', 'degree': 3, 'kernel': 'poly', 'nu': 0.15},
   {'class_weight': 'None', 'degree': 3, 'kernel': 'poly', 'nu': 0.2},
   {'class_weight': 'None', 'degree': 3, 'kernel': 'poly', 'nu': 0.25},
   {'class_weight': 'None', 'degree': 4, 'kernel': 'poly', 'nu': 0.1},
   {'class_weight': 'None', 'degree': 4, 'kernel': 'poly', 'nu': 0.15},
   {'class_weight': 'None', 'degree': 4, 'kernel': 'poly', 'nu': 0.2},
   {'class_weight': 'None', 'degree': 4, 'kernel': 'poly', 'nu': 0.25},
   {'class_weight': 'None', 'degree': 5, 'kernel': 'poly', 'nu': 0.1},
   {'class_weight': 'None', 'degree': 5, 'kernel': 'poly', 'nu': 0.15},
   {'class_weight': 'None', 'degree': 5, 'kernel': 'poly', 'nu': 0.2},
   {'class_weight': 'None', 'degree': 5, 'kernel': 'poly', 'nu': 0.25},
   {'class_weight': 'None', 'degree': 6, 'kernel': 'poly', 'nu': 0.1},
   {'class_weight': 'None', 'degree': 6, 'kernel': 'poly', 'nu': 0.15},
   {'class_weight': 'None', 'degree': 6, 'kernel': 'poly', 'nu': 0.2},
   {'class_weight': 'None', 'degree': 6, 'kernel': 'poly', 'nu': 0.25}],
  'split0_test_score': array([0.36956522, 0.7173913 , 0.7173913 , 0.8260869
6,        nan,
              nan,        nan,        nan, 0.73913043, 0.73913043,
        0.82608696, 0.86956522, 0.80434783, 0.73913043, 0.80434783,
        0.86956522, 0.69565217, 0.69565217, 0.7826087 , 0.84782609,
        0.76086957, 0.7173913 , 0.82608696, 0.84782609, 0.80434783,
        0.7826087 , 0.80434783, 0.84782609, 0.73913043, 0.80434783,
        0.80434783, 0.80434783,        nan,        nan,        nan,
              nan,        nan,        nan,        nan,        nan,
              nan,        nan,        nan,        nan,        nan,
              nan,        nan,        nan,        nan,        nan,
              nan,        nan,        nan,        nan,        nan,
              nan, 0.54347826, 0.84782609, 0.7826087 , 0.80434783,
        0.52173913, 0.63043478, 0.39130435, 0.39130435, 0.65217391,
        0.69565217, 0.73913043, 0.76086957, 0.63043478, 0.65217391,
        0.43478261, 0.60869565,        nan,        nan,        nan,
              nan,        nan,        nan,        nan,        nan,
              nan,        nan,        nan,        nan,        nan,
              nan,        nan,        nan]),
  'split1_test_score': array([0.32608696, 0.76086957, 0.73913043, 0.7608695
7,        nan,
              nan,        nan,        nan, 0.7173913 , 0.73913043,
        0.76086957, 0.7826087 , 0.73913043, 0.7826087 , 0.76086957,
        0.7826087 , 0.7173913 , 0.73913043, 0.7826087 , 0.7826087 ,
        0.65217391, 0.76086957, 0.7826087 , 0.80434783, 0.54347826,
        0.73913043, 0.73913043, 0.76086957, 0.36956522, 0.73913043,
        0.82608696, 0.76086957,        nan,        nan,        nan,
              nan,        nan,        nan,        nan,        nan,
              nan,        nan,        nan,        nan,        nan,
```

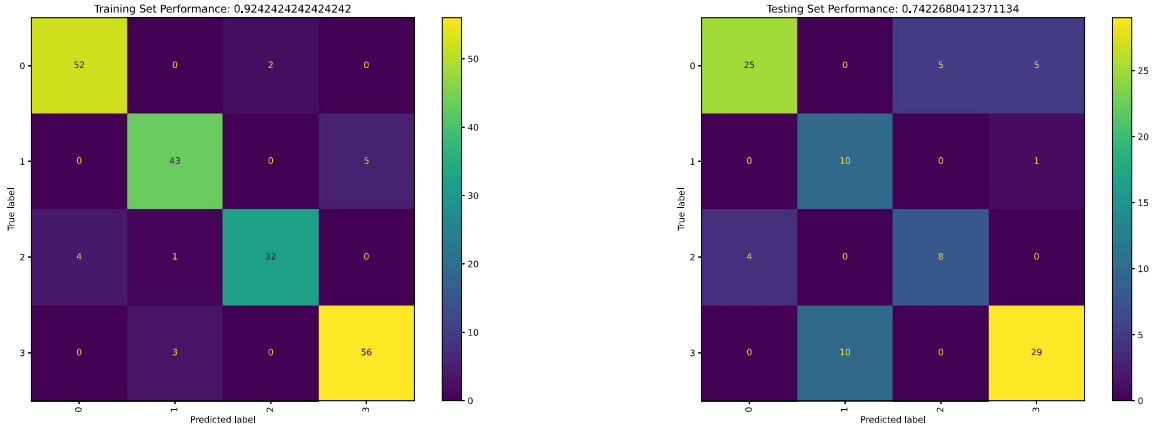
```
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan, 0.67391304, 0.73913043, 0.76086957, 0.80434783,
          0.58695652, 0.63043478, 0.65217391, 0.76086957, 0.65217391,
          0.67391304, 0.76086957, 0.84782609, 0.54347826, 0.60869565,
          0.60869565, 0.63043478,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan]),
   'split2_test_score': array([0.73913043, 0.7826087 , 0.7173913 , 0.8913043
  5,        nan,
                nan,        nan,        nan, 0.73913043, 0.80434783,
          0.84782609, 0.84782609, 0.7826087 , 0.80434783, 0.84782609,
          0.84782609, 0.7826087 , 0.82608696, 0.86956522, 0.84782609,
          0.7826087 , 0.76086957, 0.84782609, 0.86956522, 0.69565217,
          0.76086957, 0.86956522, 0.84782609, 0.67391304, 0.84782609,
          0.91304348, 0.86956522,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan, 0.76086957, 0.84782609, 0.84782609, 0.80434783,
          0.54347826, 0.67391304, 0.7173913 , 0.69565217, 0.76086957,
          0.7826087 , 0.84782609, 0.76086957, 0.69565217, 0.63043478,
          0.63043478, 0.69565217,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan]),
   'split3_test_score': array([0.42222222, 0.64444444, 0.73333333, 0.7333333
  3,        nan,
                nan,        nan,        nan, 0.73333333, 0.68888889,
          0.73333333, 0.68888889, 0.66666667, 0.71111111, 0.73333333,
          0.68888889, 0.73333333, 0.71111111, 0.71111111, 0.66666667,
          0.64444444, 0.71111111, 0.73333333, 0.75555556, 0.71111111,
          0.75555556, 0.75555556, 0.77777778, 0.68888889, 0.71111111,
          0.75555556, 0.71111111,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan, 0.75555556, 0.66666667, 0.73333333, 0.71111111,
          0.57777778, 0.55555556, 0.66666667, 0.57777778, 0.66666667,
          0.62222222, 0.68888889, 0.68888889, 0.55555556, 0.53333333,
          0.24444444, 0.6       ,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan]),
   'split4_test_score': array([0.53333333, 0.68888889, 0.71111111, 0.7333333
  3,        nan,
                nan,        nan,        nan, 0.77777778, 0.77777778,
          0.75555556, 0.75555556, 0.75555556, 0.75555556, 0.73333333,
          0.73333333, 0.73333333, 0.75555556, 0.71111111, 0.75555556,
          0.68888889, 0.73333333, 0.73333333, 0.73333333, 0.66666667,
          0.75555556, 0.73333333, 0.73333333, 0.64444444, 0.75555556,
          0.71111111, 0.73333333,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan,        nan,        nan,        nan,        nan,
                nan, 0.68888889, 0.71111111, 0.75555556, 0.75555556,
          0.64444444, 0.64444444, 0.62222222, 0.64444444, 0.62222222,
```

```
        0.55555556, 0.71111111, 0.73333333, 0.53333333, 0.6       ,
        0.66666667, 0.57777778,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan]),
 'mean_test_score': array([0.47806763, 0.71884058, 0.7236715 , 0.78898551,
nan,
               nan,        nan,        nan, 0.74135266, 0.74985507,
        0.7847343 , 0.78888889, 0.74966184, 0.75855072, 0.77594203,
        0.78444444, 0.73246377, 0.74550725, 0.77140097, 0.78009662,
        0.7057971 , 0.73671498, 0.78463768, 0.8021256 , 0.68425121,
        0.75874396, 0.78038647, 0.79352657, 0.62318841, 0.7715942 ,
        0.80202899, 0.77584541,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan, 0.68454106, 0.76251208, 0.77603865, 0.77594203,
        0.57487923, 0.62695652, 0.60995169, 0.61400966, 0.67082126,
        0.66599034, 0.74956522, 0.75835749, 0.59169082, 0.60492754,
        0.51700483, 0.62251208,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan]),
 'std_test_score': array([0.14778207, 0.0495314 , 0.01066747, 0.06136856,
nan,
               nan,        nan,        nan, 0.01987522, 0.03923276,
        0.04416696, 0.06501202, 0.04713788, 0.03259692, 0.04434498,
        0.0678371 , 0.0286297 , 0.04539459, 0.05857846, 0.06730452,
        0.05631781, 0.02101088, 0.0468598 , 0.05204367, 0.08414377,
        0.01399333, 0.051107  , 0.0465502 , 0.13046734, 0.04868971,
        0.0683143 , 0.05624566,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan, 0.07861256, 0.07339218, 0.03916632, 0.03752152,
        0.04196063, 0.03907182, 0.11357799, 0.12654644, 0.04729328,
        0.07570195, 0.05486749, 0.0519025 , 0.06220106, 0.04011745,
        0.15793789, 0.04027676,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan]),
 'rank_test_score': array([44, 29, 28,  4, 61, 62, 63, 64, 25, 21,  6,  5,
22, 19, 12,  8, 27,
        24, 16, 10, 30, 26,  7,  1, 32, 18,  9,  3, 36, 15,  2, 14, 65, 6
6,
        67, 68, 69, 70, 71, 72, 74, 87, 75, 59, 77, 78, 79, 80, 81, 82, 8
3,
        84, 85, 86, 60, 73, 31, 17, 11, 12, 42, 35, 39, 38, 33, 34, 23, 2
0,
        41, 40, 43, 37, 52, 53, 54, 55, 56, 49, 51, 58, 50, 57, 48, 47, 4
6,
        45, 76, 88], dtype=int32)}
```

In [16]:

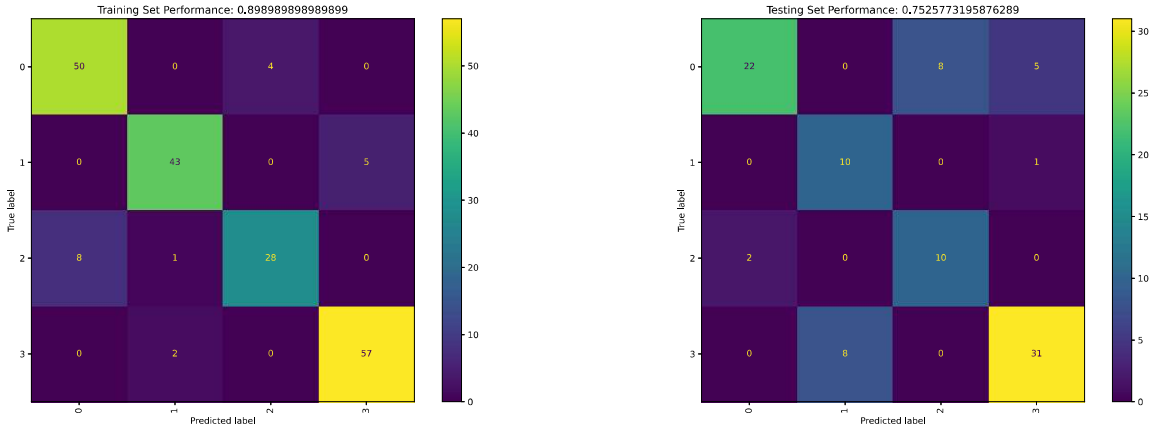
```
best_system = numpy.argmin(grid_search.cv_results_['rank_test_score'])
params = grid_search.cv_results_['params'][best_system]
print(params)
svm = NuSVC().set_params(**params)
svm.fit(X_train, Y_train)
eval_model(svm, X_train, Y_train, X_test, Y_test)
```

```
{'class_weight': 'balanced', 'gamma': 0.075, 'kernel': 'rbf', 'nu': 0.25}
```



In [17]:

```
svm2 = NuSVC()
svm2.fit(X_train, Y_train)
eval_model(svm2, X_train, Y_train, X_test, Y_test)
```

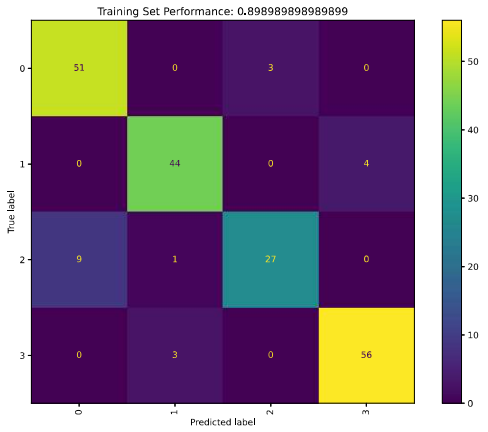


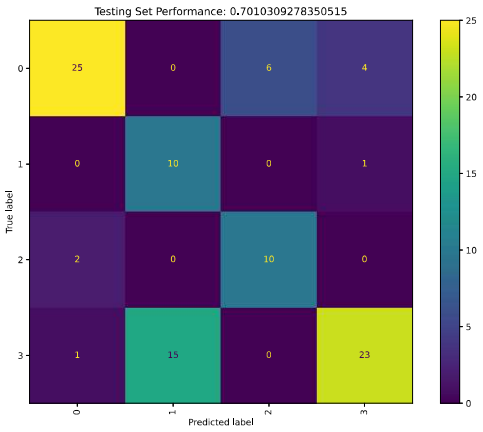
Evaluate K Nearest Neighbours model

Code taken from Week 3 Example 2

In [18]:

```
cknn = KNeighborsClassifier()
cknn.fit(X_train, Y_train)
eval_model(cknn, X_train, Y_train, X_test, Y_test)
```





Find best values for K Nearest Neighbours

Code taken from Week 3 Example 2

In [19]:

```
param_grid = [
   {'n_neighbors' : list(range(1,21)), 'weights' : ['uniform', 'distance']}
 ]
cknn = KNeighborsClassifier()

grid_search = GridSearchCV(cknn, param_grid)
grid_search.fit(X_val, Y_val)
grid_search.cv_results_
```

Out[19]:

```
{'mean_fit_time': array([0.00059571, 0.00059752, 0.00049624, 0.00049596,
0.00039687,
        0.00039687, 0.00049572, 0.00039635, 0.00029793, 0.00050139,
        0.00039697, 0.00049596, 0.00049605, 0.00039678, 0.00049596,
        0.00049596, 0.00039668, 0.00039635, 0.00039663, 0.00049601,
        0.00059533, 0.00049605, 0.00049834, 0.00049586, 0.00029821,
        0.00029764, 0.00039678, 0.00049577, 0.00050197, 0.00049605,
        0.00049601, 0.00039673, 0.00049109, 0.00049605, 0.0003026 ,
        0.00039668, 0.00039692, 0.00049553, 0.00050139, 0.0004961 ]),
 'std_fit_time': array([1.98650990e-04, 1.97542844e-04, 1.90734863e-07, 2.
33601546e-07,
        1.98435805e-04, 1.99102931e-04, 4.62310777e-07, 1.98173580e-04,
        2.43257653e-04, 1.06698641e-05, 1.98483546e-04, 1.78416128e-07,
        1.16800773e-07, 1.98388465e-04, 1.78416128e-07, 3.81469727e-07,
        1.98340552e-04, 1.98174383e-04, 1.98316832e-04, 1.90734863e-07,
        1.98364430e-04, 4.42200589e-07, 4.50957886e-06, 3.16297988e-07,
        2.43492094e-04, 2.43024613e-04, 1.98389955e-04, 4.42200589e-07,
        1.11661376e-05, 1.60614158e-05, 7.00804637e-07, 1.98364315e-04,
        9.87110505e-06, 4.42200589e-07, 2.47245028e-04, 1.98340953e-04,
        1.98460198e-04, 3.56832255e-07, 1.87427940e-05, 3.81469727e-07]),
 'mean_score_time': array([0.00208306, 0.00089035, 0.00198379, 0.00079374,
0.00188451,
        0.00068941, 0.00168638, 0.00059566, 0.00208774, 0.00068974,
        0.00168638, 0.00059524, 0.00168624, 0.00069437, 0.00158739,
        0.00059509, 0.0017858 , 0.00109167, 0.00198402, 0.0007936 ,
        0.00178542, 0.00099211, 0.00178299, 0.00069447, 0.00188389,
        0.00099187, 0.00178585, 0.00089321, 0.00197792, 0.00068927,
        0.00188451, 0.00089278, 0.00178576, 0.00069447, 0.00197883,
        0.00099211, 0.00188484, 0.00079913, 0.00207286, 0.00079379]),
 'std_score_time': array([1.98006766e-04, 1.97264345e-04, 4.15696997e-07,
2.42887359e-04,
        1.98388522e-04, 2.47196430e-04, 2.43296384e-04, 1.98197558e-04,
        3.69796504e-04, 2.46831843e-04, 2.43004207e-04, 1.98292811e-04,
        2.43120820e-04, 2.42965102e-04, 1.98721964e-04, 1.98007168e-04,
        3.96752454e-04, 1.98388580e-04, 1.16800773e-07, 2.42965102e-04,
        2.42867756e-04, 5.51978917e-07, 2.41304884e-04, 2.42596002e-04,
        1.98436378e-04, 8.63773106e-06, 2.43218252e-04, 1.98175072e-04,
        1.10909494e-05, 2.47304548e-04, 5.77762068e-04, 1.98555680e-04,
        2.43335099e-04, 2.42692660e-04, 1.01168170e-05, 2.78041453e-07,
        1.98555852e-04, 2.47434583e-04, 1.91214519e-04, 2.43121148e-04]),
 'param_n_neighbors': masked_array(data=[1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6,
6, 7, 7, 8, 8, 9, 9,
                   10, 10, 11, 11, 12, 12, 13, 13, 14, 14, 15, 15, 16, 1
6,
                   17, 17, 18, 18, 19, 19, 20, 20],
             mask=[False, False, False, False, False, False, False, Fals
e,
                   False, False, False, False, False, False, False, Fals
e,
                   False, False, False, False, False, False, False, Fals
e,
                   False, False, False, False, False, False, False, Fals
e,
                   False, False, False, False, False, False, False, Fals
e],
        fill_value='?',
             dtype=object),
 'param_weights': masked_array(data=['uniform', 'distance', 'uniform', 'di
stance',
```

```
                   'uniform', 'distance', 'uniform', 'distance',
                   'uniform', 'distance', 'uniform', 'distance',
                   'uniform', 'distance', 'uniform', 'distance',
                   'uniform', 'distance', 'uniform', 'distance',
                   'uniform', 'distance', 'uniform', 'distance',
                   'uniform', 'distance', 'uniform', 'distance',
                   'uniform', 'distance', 'uniform', 'distance',
                   'uniform', 'distance', 'uniform', 'distance',
                   'uniform', 'distance', 'uniform', 'distance'],
              mask=[False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e,
                    False, False, False, False, False, False, False, Fals
e],
        fill_value='?',
             dtype=object),
 'params': [{'n_neighbors': 1, 'weights': 'uniform'},
  {'n_neighbors': 1, 'weights': 'distance'},
  {'n_neighbors': 2, 'weights': 'uniform'},
  {'n_neighbors': 2, 'weights': 'distance'},
  {'n_neighbors': 3, 'weights': 'uniform'},
  {'n_neighbors': 3, 'weights': 'distance'},
  {'n_neighbors': 4, 'weights': 'uniform'},
  {'n_neighbors': 4, 'weights': 'distance'},
  {'n_neighbors': 5, 'weights': 'uniform'},
  {'n_neighbors': 5, 'weights': 'distance'},
  {'n_neighbors': 6, 'weights': 'uniform'},
  {'n_neighbors': 6, 'weights': 'distance'},
  {'n_neighbors': 7, 'weights': 'uniform'},
  {'n_neighbors': 7, 'weights': 'distance'},
  {'n_neighbors': 8, 'weights': 'uniform'},
  {'n_neighbors': 8, 'weights': 'distance'},
  {'n_neighbors': 9, 'weights': 'uniform'},
  {'n_neighbors': 9, 'weights': 'distance'},
  {'n_neighbors': 10, 'weights': 'uniform'},
  {'n_neighbors': 10, 'weights': 'distance'},
  {'n_neighbors': 11, 'weights': 'uniform'},
  {'n_neighbors': 11, 'weights': 'distance'},
  {'n_neighbors': 12, 'weights': 'uniform'},
  {'n_neighbors': 12, 'weights': 'distance'},
  {'n_neighbors': 13, 'weights': 'uniform'},
  {'n_neighbors': 13, 'weights': 'distance'},
  {'n_neighbors': 14, 'weights': 'uniform'},
  {'n_neighbors': 14, 'weights': 'distance'},
  {'n_neighbors': 15, 'weights': 'uniform'},
  {'n_neighbors': 15, 'weights': 'distance'},
  {'n_neighbors': 16, 'weights': 'uniform'},
  {'n_neighbors': 16, 'weights': 'distance'},
  {'n_neighbors': 17, 'weights': 'uniform'},
  {'n_neighbors': 17, 'weights': 'distance'},
  {'n_neighbors': 18, 'weights': 'uniform'},
  {'n_neighbors': 18, 'weights': 'distance'},
  {'n_neighbors': 19, 'weights': 'uniform'},
  {'n_neighbors': 19, 'weights': 'distance'},
  {'n_neighbors': 20, 'weights': 'uniform'},
  {'n_neighbors': 20, 'weights': 'distance'}],
```

```
 'split0_test_score': array([0.76086957, 0.76086957, 0.7173913 , 0.7608695
7, 0.82608696,
        0.82608696, 0.80434783, 0.7826087 , 0.82608696, 0.80434783,
        0.84782609, 0.84782609, 0.86956522, 0.86956522, 0.84782609,
        0.86956522, 0.86956522, 0.89130435, 0.84782609, 0.86956522,
        0.86956522, 0.86956522, 0.84782609, 0.86956522, 0.84782609,
        0.89130435, 0.86956522, 0.86956522, 0.82608696, 0.89130435,
        0.80434783, 0.86956522, 0.7826087 , 0.86956522, 0.7826087 ,
        0.86956522, 0.7826087 , 0.82608696, 0.7826087 , 0.80434783]),
 'split1_test_score': array([0.7173913 , 0.7173913 , 0.76086957, 0.7173913
, 0.76086957,
        0.73913043, 0.7826087 , 0.73913043, 0.76086957, 0.76086957,
        0.73913043, 0.76086957, 0.76086957, 0.76086957, 0.7173913 ,
        0.76086957, 0.76086957, 0.76086957, 0.7173913 , 0.76086957,
        0.69565217, 0.76086957, 0.76086957, 0.73913043, 0.76086957,
        0.7826087 , 0.73913043, 0.73913043, 0.76086957, 0.7826087 ,
        0.7826087 , 0.80434783, 0.76086957, 0.7826087 , 0.80434783,
        0.76086957, 0.76086957, 0.7826087 , 0.82608696, 0.76086957]),
 'split2_test_score': array([0.89130435, 0.89130435, 0.86956522, 0.8913043
5, 0.86956522,
        0.86956522, 0.84782609, 0.84782609, 0.86956522, 0.84782609,
        0.80434783, 0.86956522, 0.84782609, 0.84782609, 0.84782609,
        0.82608696, 0.84782609, 0.86956522, 0.84782609, 0.82608696,
        0.84782609, 0.86956522, 0.82608696, 0.82608696, 0.84782609,
        0.82608696, 0.82608696, 0.82608696, 0.86956522, 0.84782609,
        0.84782609, 0.82608696, 0.84782609, 0.82608696, 0.80434783,
        0.82608696, 0.80434783, 0.82608696, 0.80434783, 0.82608696]),
 'split3_test_score': array([0.64444444, 0.64444444, 0.64444444, 0.6444444
4, 0.68888889,
        0.68888889, 0.73333333, 0.71111111, 0.68888889, 0.73333333,
        0.71111111, 0.68888889, 0.73333333, 0.71111111, 0.68888889,
        0.71111111, 0.68888889, 0.68888889, 0.66666667, 0.73333333,
        0.66666667, 0.71111111, 0.66666667, 0.66666667, 0.62222222,
        0.62222222, 0.66666667, 0.66666667, 0.66666667, 0.68888889,
        0.64444444, 0.66666667, 0.64444444, 0.62222222, 0.62222222,
        0.62222222, 0.64444444, 0.62222222, 0.62222222, 0.6       ]),
 'split4_test_score': array([0.71111111, 0.71111111, 0.68888889, 0.7111111
1, 0.73333333,
        0.71111111, 0.71111111, 0.73333333, 0.75555556, 0.71111111,
        0.73333333, 0.75555556, 0.77777778, 0.75555556, 0.75555556,
        0.77777778, 0.77777778, 0.8       , 0.73333333, 0.75555556,
        0.77777778, 0.73333333, 0.75555556, 0.75555556, 0.73333333,
        0.75555556, 0.73333333, 0.73333333, 0.75555556, 0.75555556,
        0.75555556, 0.75555556, 0.77777778, 0.75555556, 0.75555556,
        0.75555556, 0.75555556, 0.75555556, 0.73333333, 0.75555556]),
 'mean_test_score': array([0.74502415, 0.74502415, 0.73623188, 0.74502415,
0.77574879,
        0.76695652, 0.77584541, 0.76280193, 0.78019324, 0.77149758,
        0.76714976, 0.78454106, 0.7978744 , 0.78898551, 0.77149758,
        0.78908213, 0.78898551, 0.8021256 , 0.7626087 , 0.78908213,
        0.77149758, 0.78888889, 0.77140097, 0.77140097, 0.76241546,
        0.77555556, 0.76695652, 0.76695652, 0.77574879, 0.79323671,
        0.76695652, 0.78444444, 0.76270531, 0.77120773, 0.75381643,
        0.7668599 , 0.74956522, 0.76251208, 0.75371981, 0.74937198]),
 'std_test_score': array([0.08207182, 0.08207182, 0.07668905, 0.08207182,
0.06465378,
        0.06930009, 0.04906558, 0.0484192 , 0.06230648, 0.04923651,
        0.0509143 , 0.06600597, 0.05210283, 0.05987953, 0.06581476,
        0.05449789, 0.06463862, 0.07352562, 0.07298019, 0.05074405,
        0.08028049, 0.06773216, 0.06350215, 0.07055284, 0.08377619,
        0.08934435, 0.07209413, 0.07209413, 0.06908476, 0.07073691,
```

```
        0.06829667, 0.06940023, 0.06611788, 0.0839733 , 0.06819956,
        0.08383312, 0.05533002, 0.07512262, 0.07258373, 0.07923377]),
 'rank_test_score': array([37, 37, 40, 37, 13, 26, 12, 28, 11, 16, 22,  9,
2,  6, 17,  4,  6,
         1, 30,  4, 18,  8, 19, 19, 32, 15, 23, 23, 14,  3, 23, 10, 29, 2
1,
        33, 27, 35, 31, 34, 36], dtype=int32)}
```

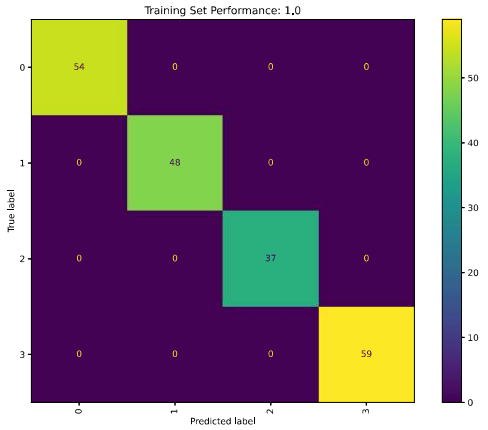
Evaluate K Nearest Neighbours using best values

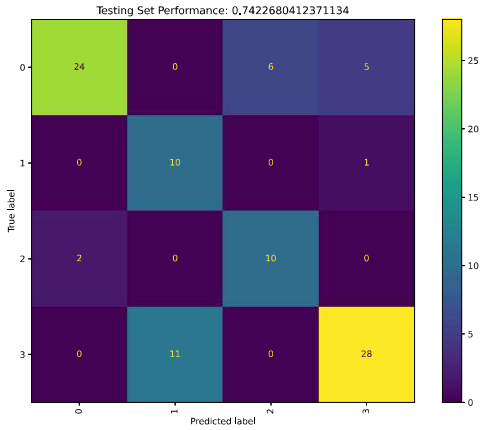
Code taken from Week 3 Example 2

In [20]:

```
best_system = numpy.argmin(grid_search.cv_results_['rank_test_score'])
params = grid_search.cv_results_['params'][best_system]
print(params)
cknn = KNeighborsClassifier().set_params(**params)
cknn.fit(X_train, Y_train)
eval_model(cknn, X_train, Y_train, X_test, Y_test)
```

```
{'n_neighbors': 9, 'weights': 'distance'}
```

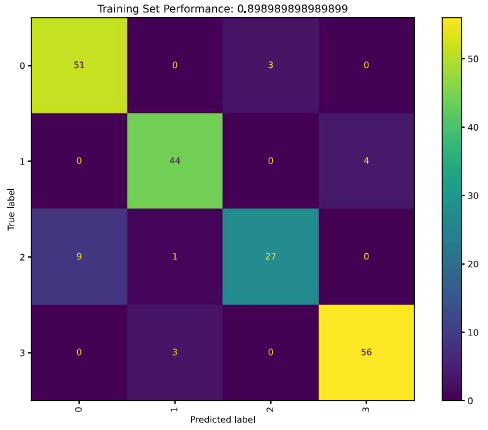


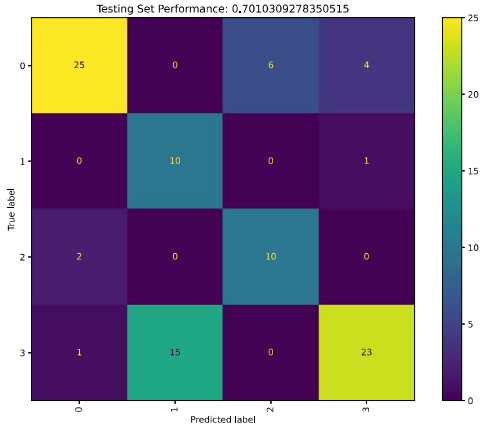


validate

In [21]:

```
cknn2 = KNeighborsClassifier()
cknn2.fit(X_train, Y_train)
eval_model(cknn2, X_train, Y_train, X_test, Y_test)
```



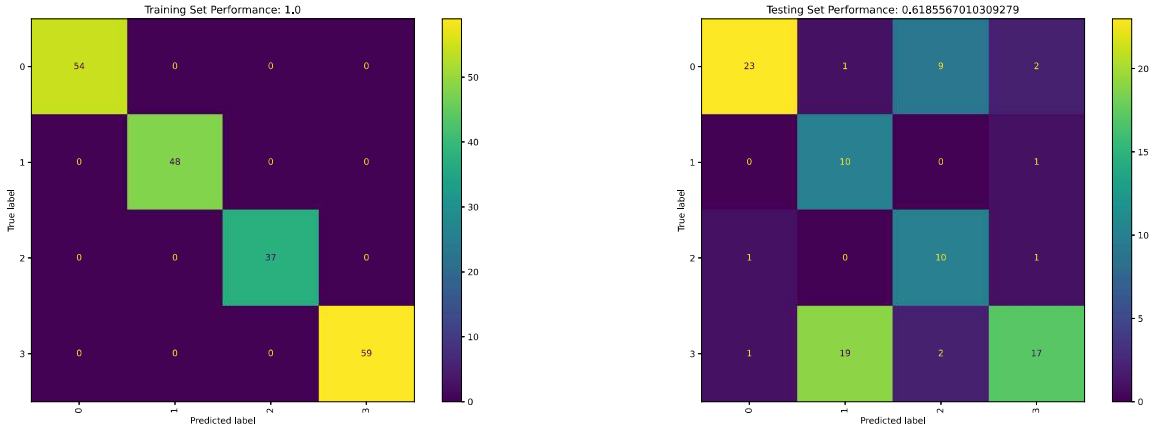


Default RFC

Code taken from Week 3 Example 2

In [22]:

```
rf = RandomForestClassifier()
rf.fit(X_train, Y_train)
eval_model(rf, X_train, Y_train, X_test, Y_test)
```



Find optimal values

In [23]:

```
rf = RandomForestClassifier()
params = { 'n_estimators' : [1,10,50,100,250],'max_depth' : list(range(1,21)),'class_we
ight': ['balanced', 'balanced_subsample', 'None']}
grid_search = GridSearchCV(rf, params)
grid_search.fit(X_val, Y_val)
grid_search.cv_results_
```

```
ocal\Programs\Python\Python38-32\lib\site-packages\sklearn\ensemble\_fores
t.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
```

```
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
```

```
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
```

```
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
```

```
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_validation.py:610: FitFailedWarning: Estimator f
it failed. The score on this train-test partition for these parameters wil
l be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\model_selection\_validation.py", line 593, in _fit_and_sco
re
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 331, in fit
    y, expanded_class_weight = self._validate_y_class_weight(y)
  File "C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-p
ackages\sklearn\ensemble\_forest.py", line 582, in _validate_y_class_weigh
```

```
t
    raise ValueError('Valid presets for class_weight include '
ValueError: Valid presets for class_weight include "balanced" and "balance
d_subsample".Given "None".

  warnings.warn("Estimator fit failed. The score on this train-test"
C:\Users\User\AppData\Local\Programs\Python\Python38-32\lib\site-packages
\sklearn\model_selection\_search.py:918: UserWarning: One or more of the t
est scores are non-finite: [0.36792271 0.59246377 0.60067633 0.6142029  0.
61855072 0.68859903
 0.7189372  0.76270531 0.74086957 0.74077295 0.71874396 0.74067633
 0.73623188 0.74067633 0.7363285  0.67942029 0.74512077 0.75816425
 0.75391304 0.74512077 0.70135266 0.7668599  0.75391304 0.76270531
 0.75826087 0.71468599 0.73642512 0.75371981 0.74956522 0.74512077
 0.70183575 0.74512077 0.75391304 0.75391304 0.75400966 0.66222222
 0.75855072 0.76289855 0.7494686  0.75835749 0.73227053 0.76714976
 0.77574879 0.76270531 0.74966184 0.65362319 0.73207729 0.74956522
 0.75845411 0.75835749 0.68405797 0.76280193 0.77584541 0.77169082
 0.74521739 0.6884058  0.75381643 0.75410628 0.74956522 0.75391304
 0.70599034 0.74067633 0.76724638 0.75826087 0.76705314 0.72792271
 0.78444444 0.75845411 0.74086957 0.75835749 0.69285024 0.74985507
 0.76299517 0.75400966 0.76270531 0.67043478 0.7494686  0.74086957
 0.74956522 0.76270531 0.71487923 0.75816425 0.74512077 0.74531401
 0.77149758 0.66676329 0.75826087 0.7715942  0.75381643 0.75400966
 0.69285024 0.73652174 0.75391304 0.77149758 0.75400966 0.69710145
 0.75391304 0.78463768 0.75835749 0.75855072 0.24570048 0.51410628
 0.60086957 0.60966184 0.62280193 0.6752657  0.73652174 0.74077295
 0.73188406 0.74512077 0.69700483 0.72753623 0.74502415 0.7363285
 0.74512077 0.692657   0.74956522 0.75826087 0.74956522 0.75381643
 0.6447343  0.74512077 0.75835749 0.7626087  0.7494686  0.68821256
 0.73623188 0.75371981 0.75400966 0.74956522 0.70183575 0.74067633
 0.76705314 0.74956522 0.75391304 0.6931401  0.74096618 0.75391304
 0.75391304 0.74086957 0.6752657  0.72338164 0.76270531 0.76289855
 0.75400966 0.6973913  0.71922705 0.75845411 0.76714976 0.75400966
 0.70570048 0.74956522 0.75381643 0.7715942  0.76705314 0.67120773
 0.74995169 0.74975845 0.74077295 0.74531401 0.67111111 0.78038647
 0.75835749 0.75845411 0.76714976 0.71062802 0.77120773 0.74937198
 0.75400966 0.74966184 0.66657005 0.76280193 0.74966184 0.76280193
 0.74966184 0.6705314  0.78028986 0.75835749 0.74956522 0.76289855
 0.68801932 0.74937198 0.74531401 0.75835749 0.75400966 0.69700483
 0.75391304 0.7626087  0.75410628 0.76705314 0.69700483 0.7410628
 0.75835749 0.7715942  0.75835749 0.692657   0.76270531 0.7668599
 0.76714976 0.74966184        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan]
  warnings.warn(
```

Out[23]:

```
0435, 0.91304348, 0.91304348,
        0.76086957, 0.91304348, 0.91304348, 0.91304348, 0.89130435,
        0.60869565, 0.89130435, 0.89130435, 0.89130435, 0.89130435,
        0.7173913 , 0.91304348, 0.89130435, 0.91304348, 0.89130435,
        0.80434783, 0.86956522, 0.89130435, 0.91304348, 0.89130435,
        0.84782609, 0.89130435, 0.91304348, 0.91304348, 0.91304348,
        0.84782609, 0.84782609, 0.91304348, 0.91304348, 0.91304348,
        0.7826087 , 0.89130435, 0.91304348, 0.91304348, 0.91304348,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan]),
 'split3_test_score': array([0.24444444, 0.57777778, 0.53333333, 0.5777777
8, 0.57777778,
        0.68888889, 0.66666667, 0.68888889, 0.68888889, 0.68888889,
        0.64444444, 0.64444444, 0.64444444, 0.66666667, 0.64444444,
        0.64444444, 0.62222222, 0.64444444, 0.64444444, 0.64444444,
        0.62222222, 0.62222222, 0.62222222, 0.64444444, 0.64444444,
        0.71111111, 0.66666667, 0.64444444, 0.66666667, 0.64444444,
        0.68888889, 0.62222222, 0.66666667, 0.66666667, 0.66666667,
        0.66666667, 0.73333333, 0.71111111, 0.64444444, 0.66666667,
        0.77777778, 0.71111111, 0.68888889, 0.66666667, 0.68888889,
        0.66666667, 0.64444444, 0.66666667, 0.71111111, 0.68888889,
        0.62222222, 0.73333333, 0.68888889, 0.71111111, 0.66666667,
        0.68888889, 0.68888889, 0.71111111, 0.66666667, 0.66666667,
        0.66666667, 0.62222222, 0.71111111, 0.64444444, 0.66666667,
        0.66666667, 0.68888889, 0.68888889, 0.68888889, 0.66666667,
        0.64444444, 0.71111111, 0.71111111, 0.66666667, 0.66666667,
        0.55555556, 0.64444444, 0.66666667, 0.64444444, 0.68888889,
        0.73333333, 0.68888889, 0.68888889, 0.66666667, 0.71111111,
        0.66666667, 0.62222222, 0.71111111, 0.64444444, 0.66666667,
        0.68888889, 0.68888889, 0.66666667, 0.68888889, 0.66666667,
        0.66666667, 0.68888889, 0.71111111, 0.71111111, 0.73333333,
        0.24444444, 0.53333333, 0.57777778, 0.57777778, 0.55555556,
        0.62222222, 0.68888889, 0.66666667, 0.64444444, 0.68888889,
        0.55555556, 0.64444444, 0.64444444, 0.66666667, 0.66666667,
        0.62222222, 0.66666667, 0.66666667, 0.66666667, 0.64444444,
        0.62222222, 0.66666667, 0.66666667, 0.64444444, 0.64444444,
        0.64444444, 0.64444444, 0.62222222, 0.66666667, 0.64444444,
        0.66666667, 0.62222222, 0.66666667, 0.66666667, 0.66666667,
        0.66666667, 0.68888889, 0.66666667, 0.64444444, 0.66666667,
        0.64444444, 0.64444444, 0.68888889, 0.71111111, 0.66666667,
        0.75555556, 0.71111111, 0.71111111, 0.66666667, 0.66666667,
```

```
        0.62222222, 0.68888889, 0.64444444, 0.71111111, 0.68888889,
        0.73333333, 0.71111111, 0.66666667, 0.64444444, 0.66666667,
        0.62222222, 0.73333333, 0.68888889, 0.68888889, 0.68888889,
        0.68888889, 0.66666667, 0.66666667, 0.68888889, 0.66666667,
        0.6       , 0.71111111, 0.66666667, 0.66666667, 0.68888889,
        0.53333333, 0.68888889, 0.66666667, 0.66666667, 0.71111111,
        0.53333333, 0.6       , 0.64444444, 0.66666667, 0.66666667,
        0.62222222, 0.64444444, 0.64444444, 0.68888889, 0.66666667,
        0.57777778, 0.68888889, 0.66666667, 0.71111111, 0.66666667,
        0.62222222, 0.64444444, 0.66666667, 0.71111111, 0.66666667,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan]),
  'split4_test_score': array([0.37777778, 0.68888889, 0.62222222, 0.6888888
9, 0.68888889,
        0.68888889, 0.68888889, 0.73333333, 0.71111111, 0.68888889,
        0.66666667, 0.71111111, 0.68888889, 0.68888889, 0.71111111,
        0.62222222, 0.75555556, 0.73333333, 0.75555556, 0.73333333,
        0.68888889, 0.75555556, 0.77777778, 0.77777778, 0.75555556,
        0.66666667, 0.71111111, 0.71111111, 0.73333333, 0.73333333,
        0.73333333, 0.75555556, 0.73333333, 0.73333333, 0.75555556,
        0.64444444, 0.73333333, 0.75555556, 0.73333333, 0.75555556,
        0.64444444, 0.73333333, 0.73333333, 0.75555556, 0.73333333,
        0.66666667, 0.73333333, 0.73333333, 0.73333333, 0.73333333,
        0.71111111, 0.71111111, 0.75555556, 0.77777778, 0.73333333,
        0.64444444, 0.68888889, 0.73333333, 0.73333333, 0.73333333,
        0.71111111, 0.73333333, 0.75555556, 0.75555556, 0.75555556,
        0.75555556, 0.73333333, 0.75555556, 0.71111111, 0.75555556,
        0.71111111, 0.75555556, 0.77777778, 0.75555556, 0.75555556,
        0.64444444, 0.73333333, 0.73333333, 0.75555556, 0.73333333,
        0.68888889, 0.68888889, 0.68888889, 0.75555556, 0.73333333,
        0.68888889, 0.77777778, 0.75555556, 0.73333333, 0.75555556,
        0.66666667, 0.71111111, 0.73333333, 0.75555556, 0.75555556,
        0.66666667, 0.71111111, 0.75555556, 0.71111111, 0.73333333,
        0.26666667, 0.71111111, 0.62222222, 0.64444444, 0.68888889,
        0.68888889, 0.71111111, 0.71111111, 0.68888889, 0.68888889,
        0.75555556, 0.68888889, 0.71111111, 0.68888889, 0.71111111,
        0.68888889, 0.73333333, 0.73333333, 0.73333333, 0.73333333,
        0.66666667, 0.71111111, 0.75555556, 0.75555556, 0.73333333,
        0.64444444, 0.68888889, 0.73333333, 0.75555556, 0.75555556,
        0.75555556, 0.73333333, 0.75555556, 0.73333333, 0.73333333,
        0.75555556, 0.73333333, 0.73333333, 0.75555556, 0.73333333,
        0.66666667, 0.73333333, 0.73333333, 0.75555556, 0.75555556,
        0.64444444, 0.71111111, 0.73333333, 0.77777778, 0.75555556,
```

```
        0.68888889, 0.71111111, 0.73333333, 0.75555556, 0.73333333,
        0.64444444, 0.77777778, 0.77777778, 0.73333333, 0.75555556,
        0.73333333, 0.75555556, 0.73333333, 0.75555556, 0.75555556,
        0.75555556, 0.71111111, 0.68888889, 0.73333333, 0.75555556,
        0.71111111, 0.73333333, 0.75555556, 0.77777778, 0.73333333,
        0.68888889, 0.77777778, 0.75555556, 0.73333333, 0.75555556,
        0.71111111, 0.75555556, 0.77777778, 0.75555556, 0.75555556,
        0.68888889, 0.75555556, 0.75555556, 0.75555556, 0.75555556,
        0.73333333, 0.75555556, 0.75555556, 0.75555556, 0.75555556,
        0.68888889, 0.77777778, 0.71111111, 0.73333333, 0.75555556,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan]),
 'mean_test_score': array([0.36792271, 0.59246377, 0.60067633, 0.6142029 ,
0.61855072,
        0.68859903, 0.7189372 , 0.76270531, 0.74086957, 0.74077295,
        0.71874396, 0.74067633, 0.73623188, 0.74067633, 0.7363285 ,
        0.67942029, 0.74512077, 0.75816425, 0.75391304, 0.74512077,
        0.70135266, 0.7668599 , 0.75391304, 0.76270531, 0.75826087,
        0.71468599, 0.73642512, 0.75371981, 0.74956522, 0.74512077,
        0.70183575, 0.74512077, 0.75391304, 0.75391304, 0.75400966,
        0.66222222, 0.75855072, 0.76289855, 0.7494686 , 0.75835749,
        0.73227053, 0.76714976, 0.77574879, 0.76270531, 0.74966184,
        0.65362319, 0.73207729, 0.74956522, 0.75845411, 0.75835749,
        0.68405797, 0.76280193, 0.77584541, 0.77169082, 0.74521739,
        0.6884058 , 0.75381643, 0.75410628, 0.74956522, 0.75391304,
        0.70599034, 0.74067633, 0.76724638, 0.75826087, 0.76705314,
        0.72792271, 0.78444444, 0.75845411, 0.74086957, 0.75835749,
        0.69285024, 0.74985507, 0.76299517, 0.75400966, 0.76270531,
        0.67043478, 0.7494686 , 0.74086957, 0.74956522, 0.76270531,
        0.71487923, 0.75816425, 0.74512077, 0.74531401, 0.77149758,
        0.66676329, 0.75826087, 0.7715942 , 0.75381643, 0.75400966,
        0.69285024, 0.73652174, 0.75391304, 0.77149758, 0.75400966,
        0.69710145, 0.75391304, 0.78463768, 0.75835749, 0.75855072,
        0.24570048, 0.51410628, 0.60086957, 0.60966184, 0.62280193,
        0.6752657 , 0.73652174, 0.74077295, 0.73188406, 0.74512077,
        0.69700483, 0.72753623, 0.74502415, 0.7363285 , 0.74512077,
        0.692657  , 0.74956522, 0.75826087, 0.74956522, 0.75381643,
        0.6447343 , 0.74512077, 0.75835749, 0.7626087 , 0.7494686 ,
        0.68821256, 0.73623188, 0.75371981, 0.75400966, 0.74956522,
        0.70183575, 0.74067633, 0.76705314, 0.74956522, 0.75391304,
        0.6931401 , 0.74096618, 0.75391304, 0.75391304, 0.74086957,
        0.6752657 , 0.72338164, 0.76270531, 0.76289855, 0.75400966,
        0.6973913 , 0.71922705, 0.75845411, 0.76714976, 0.75400966,
```

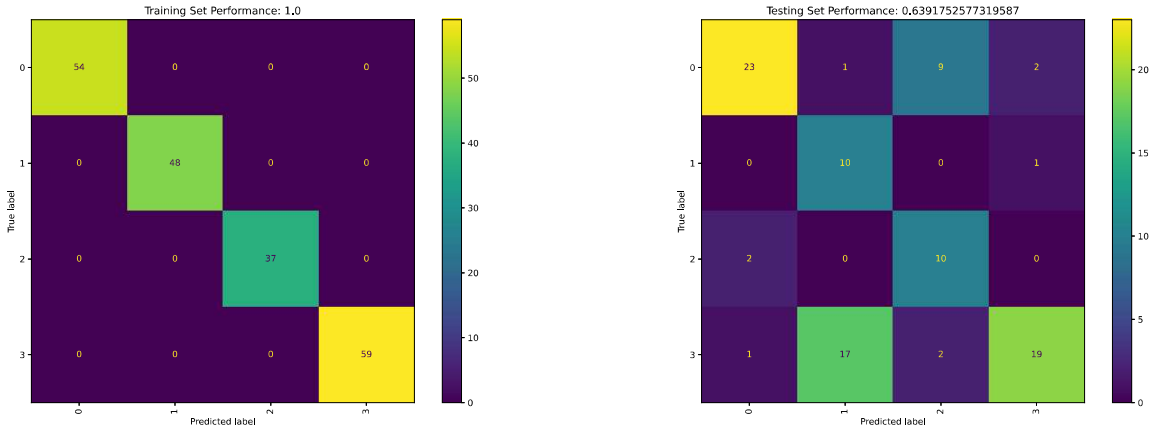
```
        0.70570048, 0.74956522, 0.75381643, 0.7715942 , 0.76705314,
        0.67120773, 0.74995169, 0.74975845, 0.74077295, 0.74531401,
        0.67111111, 0.78038647, 0.75835749, 0.75845411, 0.76714976,
        0.71062802, 0.77120773, 0.74937198, 0.75400966, 0.74966184,
        0.66657005, 0.76280193, 0.74966184, 0.76280193, 0.74966184,
        0.6705314 , 0.78028986, 0.75835749, 0.74956522, 0.76289855,
        0.68801932, 0.74937198, 0.74531401, 0.75835749, 0.75400966,
        0.69700483, 0.75391304, 0.7626087 , 0.75410628, 0.76705314,
        0.69700483, 0.7410628 , 0.75835749, 0.7715942 , 0.75835749,
        0.692657  , 0.76270531, 0.7668599 , 0.76714976, 0.74966184,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan]),
 'std_test_score': array([0.11513147, 0.10235556, 0.05561526, 0.0675732 ,
0.06194932,
        0.08287558, 0.07086968, 0.06560194, 0.06662997, 0.06987291,
        0.1004173 , 0.09456238, 0.08375814, 0.0792373 , 0.08337688,
        0.06747573, 0.0864724 , 0.08921469, 0.08816   , 0.08945451,
        0.07540838, 0.09543263, 0.10415244, 0.0983086 , 0.08677684,
        0.03821448, 0.0793969 , 0.08213833, 0.07532552, 0.09050495,
        0.03972577, 0.10610464, 0.07395793, 0.07522507, 0.07489101,
        0.05280651, 0.04677905, 0.06892039, 0.08852456, 0.07325201,
        0.06042812, 0.07494558, 0.07544799, 0.08122519, 0.08320092,
        0.01911726, 0.07500199, 0.07532552, 0.0699593 , 0.07928147,
        0.0914558 , 0.08295349, 0.08567003, 0.07442599, 0.07764871,
        0.04134995, 0.07679829, 0.07034852, 0.07532552, 0.08579123,
        0.0309776 , 0.10603987, 0.06933968, 0.08677684, 0.08042479,
        0.07799047, 0.07573696, 0.07008062, 0.09066571, 0.08293458,
        0.10604207, 0.04561615, 0.07391058, 0.07737399, 0.08238062,
        0.06966807, 0.06515302, 0.08084515, 0.08039936, 0.07749154,
        0.07789166, 0.07521204, 0.08926438, 0.08977889, 0.0627417 ,
        0.03374098, 0.08375012, 0.06231352, 0.08736864, 0.08549846,
        0.05032175, 0.08245436, 0.08355874, 0.06581476, 0.07489101,
        0.05372102, 0.06277339, 0.0682774 , 0.07802577, 0.06815971,
        0.02610055, 0.14240425, 0.0539217 , 0.05388239, 0.05951754,
        0.0836937 , 0.06730743, 0.0712724 , 0.07577984, 0.06756836,
        0.09871796, 0.07070391, 0.07609602, 0.0809731 , 0.07679342,
        0.08157897, 0.08477158, 0.08209809, 0.08477158, 0.08736864,
        0.04429359, 0.07679342, 0.09262541, 0.08625244, 0.08852456,
        0.06111109, 0.09336353, 0.09311845, 0.08549846, 0.08039936,
        0.06662703, 0.10954724, 0.07172792, 0.08911995, 0.08355874,
        0.07486857, 0.09035568, 0.07395793, 0.08030526, 0.08084515,
        0.05493533, 0.07366831, 0.06842285, 0.0779312 , 0.08438572,
        0.03779075, 0.09382205, 0.07885147, 0.08247439, 0.07489101,
```

```
        0.09743299, 0.05821094, 0.07943569, 0.07721639, 0.07665216,
        0.05073301, 0.04516596, 0.09072098, 0.08250189, 0.08977889,
        0.05672362, 0.06273099, 0.06908881, 0.081318  , 0.07753646,
        0.04160876, 0.08469116, 0.08768317, 0.08079825, 0.07747708,
        0.05492514, 0.06507517, 0.07747708, 0.07489325, 0.07355342,
        0.07727113, 0.07312765, 0.07453115, 0.08477158, 0.06612071,
        0.09153201, 0.0871851 , 0.08460227, 0.08293458, 0.07489101,
        0.07867452, 0.07911952, 0.08625244, 0.08280255, 0.08042479,
        0.09036033, 0.0582289 , 0.08293458, 0.07721639, 0.08518343,
        0.05534841, 0.07913238, 0.0834268 , 0.07494558, 0.08884288,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan]),
 'rank_test_score': array([199, 197, 196, 193, 192, 174, 154,  32, 129, 13
3, 155, 136, 145,
        139, 143, 179, 121,  61,  74, 121, 163,  24,  74,  35,  57, 157,
        142,  89,  99, 121, 162, 118,  74,  74,  66, 188,  41,  26, 110,
         47, 147,  15,   6,  35,  94, 189, 148,  99,  43,  47, 178,  29,
          5,   7, 117, 175,  85,  63,  99,  74, 159, 136,  14,  57,  19,
        150,   2,  43, 129,  47, 170,  92,  25,  66,  35, 185, 109, 129,
         99,  32, 156,  61, 118, 114,  11, 186,  57,   8,  86,  66, 170,
        140,  74,  12,  66, 165,  74,   1,  47,  41, 200, 198, 195, 194,
        191, 180, 140, 134, 149, 118, 166, 151, 126, 144, 121, 172,  99,
         57,  99,  86, 190, 121,  47,  39, 110, 176, 145,  89,  66,  99,
        161, 136,  19,  99,  74, 169, 128,  74,  74, 129, 180, 152,  32,
         26,  66, 164, 153,  43,  15,  66, 160,  99,  86,   8,  19, 182,
         91,  93, 134, 114, 183,   3,  47,  43,  15, 158,  13, 113,  65,
         94, 187,  29,  94,  31,  94, 184,   4,  47,  99,  26, 177, 112,
        114,  47,  66, 166,  74,  39,  63,  19, 166, 127,  47,   8,  47,
        172,  35,  23,  15,  94, 272, 271, 270, 269, 268, 267, 266, 265,
        264, 263, 262, 261, 260, 259, 258, 257, 256, 255, 254, 253, 252,
        273, 274, 275, 276, 298, 297, 296, 295, 294, 293, 292, 291, 290,
        289, 299, 288, 286, 285, 284, 283, 282, 281, 280, 279, 278, 277,
        251, 287, 250, 236, 221, 220, 219, 218, 217, 216, 215, 214, 213,
        212, 211, 210, 209, 208, 207, 206, 205, 204, 203, 202, 201, 222,
        223, 224, 225, 247, 246, 245, 244, 243, 242, 241, 240, 239, 238,
        248, 237, 235, 234, 233, 232, 231, 230, 229, 228, 227, 226, 249,
        300], dtype=int32)}
```

In [24]:

```
best_system = numpy.argmin(grid_search.cv_results_['rank_test_score'])
params = grid_search.cv_results_['params'][best_system]
print(params)
rf = RandomForestClassifier().set_params(**params)
rf.fit(X_train, Y_train)
eval_model(rf, X_train, Y_train, X_test, Y_test)
```

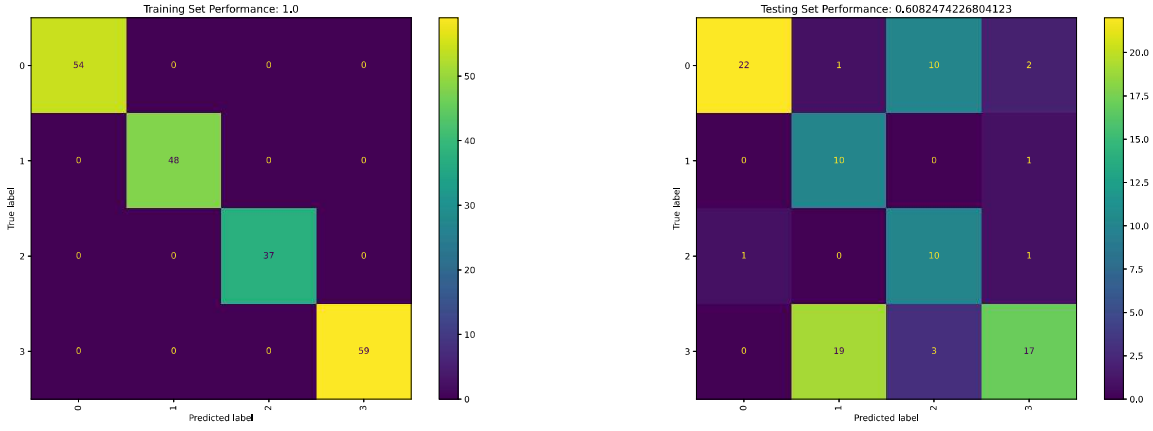
```
{'class_weight': 'balanced', 'max_depth': 20, 'n_estimators': 50}
```



verify using valiation

In [25]:

```
rf2 = RandomForestClassifier()
rf2.fit(X_train, Y_train)
eval_model(rf2, X_train, Y_train, X_test, Y_test)
```



In [ ]: