Question 1

In [45]:
```
import matplotlib.pyplot as plt
import numpy as np

# Import datasets, classifiers and performance metrics
from sklearn import datasets, metrics, svm
from sklearn.model_selection import train_test_split
```

Loading the dataset

In [33]:
```
dataset = datasets.load_digits()
X = dataset['images']
y = dataset['target']
```

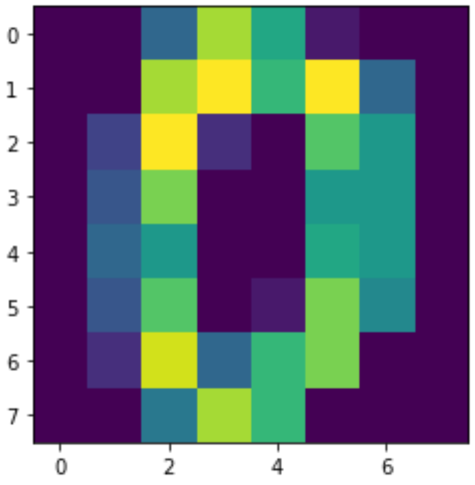
In [34]:
```
dataset
```

Out[34]:
```
{'data': array([[ 0.,  0.,  5., ...,  0.,  0.,  0.],
        [ 0.,  0.,  0., ..., 10.,  0.,  0.],
        [ 0.,  0.,  0., ..., 16.,  9.,  0.],
        ...,
        [ 0.,  0.,  1., ...,  6.,  0.,  0.],
        [ 0.,  0.,  2., ..., 12.,  0.,  0.],
        [ 0.,  0., 10., ..., 12.,  1.,  0.]]),
 'target': array([0, 1, 2, ..., 8, 9, 8]),
 'frame': None,
 'feature_names': ['pixel_0_0',
  'pixel_0_1',
  'pixel_0_2',
  'pixel_0_3',
  'pixel_0_4',
  'pixel_0_5',
  'pixel_0_6',
  'pixel_0_7',
  'pixel_1_0',
  'pixel_1_1',
  'pixel_1_2',
```

In [32]:
```
X[0]
```

Out[32]:
```
array([[ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.],
       [ 0.,  0., 13., 15., 10., 15.,  5.,  0.],
       [ 0.,  3., 15.,  2.,  0., 11.,  8.,  0.],
       [ 0.,  4., 12.,  0.,  0.,  8.,  8.,  0.],
       [ 0.,  5.,  8.,  0.,  0.,  9.,  8.,  0.],
       [ 0.,  4., 11.,  0.,  1., 12.,  7.,  0.],
       [ 0.,  2., 14.,  5., 10., 12.,  0.,  0.],
       [ 0.,  0.,  6., 13., 10.,  0.,  0.,  0.]])
```

```
In [38]: plt.imshow(X[0])

Out[38]: <matplotlib.image.AxesImage at 0x74ce8a573880>
```

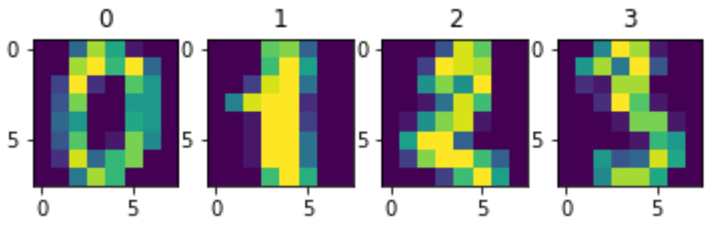


```
In [29]: len(X),len(y)

Out[29]: (1797, 1797)
```

```
In [30]: X.shape,y.shape

Out[30]: ((1797, 8, 8), (1797,))
```

Plotting and flattening of images

```
In [42]: plt.title("First 4 images")
         for i in range(1,5):
             plt.subplot(1,4,i)
             plt.title(y[i-1])
             plt.imshow(X[i-1])
```



```
In [47]: flattened_X = []
         for image in X:
             flattened_X.append(image.flatten())
         flattened_X = np.array(flattened_X)

Out[47]: (1797, 64)
```

Splitting and Training

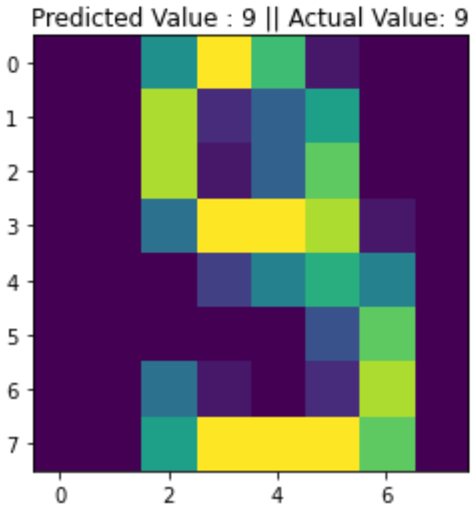
```
In [48]: X_train,X_test,y_train,y_test = train_test_split(flattened_X,y,test_s
In [60]: classifier = svm.SVC(gamma=0.001)
         classifier.fit(X_train,y_train)

Out[60]: SVC(gamma=0.001)
```

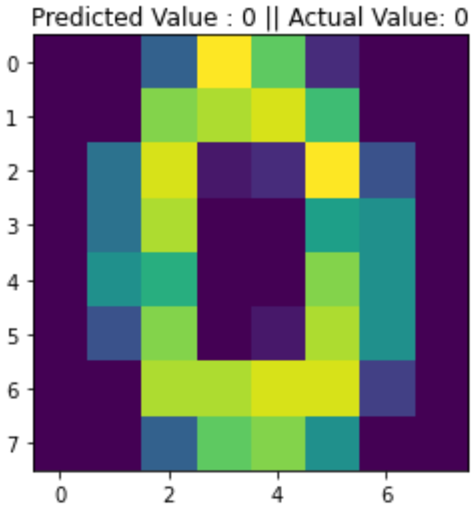
```
In [61]: y_pred = classifier.predict(X_test)
```

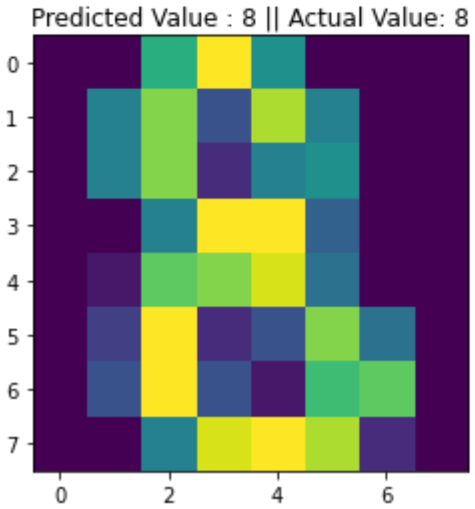
Evaluation

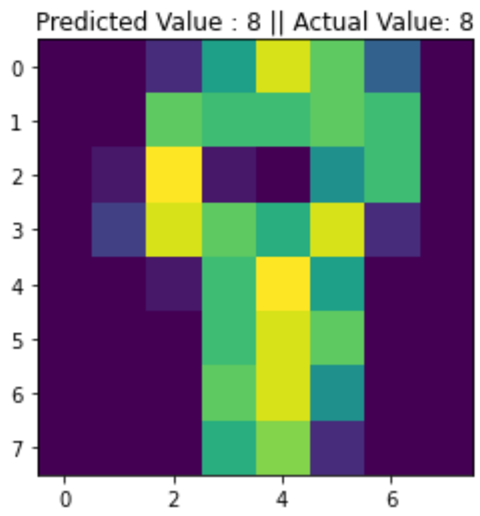
In [65]:

```
plt.title("First 4 images:")
for i in range(1,5):
    plt.title("Predicted Value : {} || Actual Value: {}".format(y_pre
    plt.imshow(X_test[i-1].reshape(8,8))
    plt.show()
```







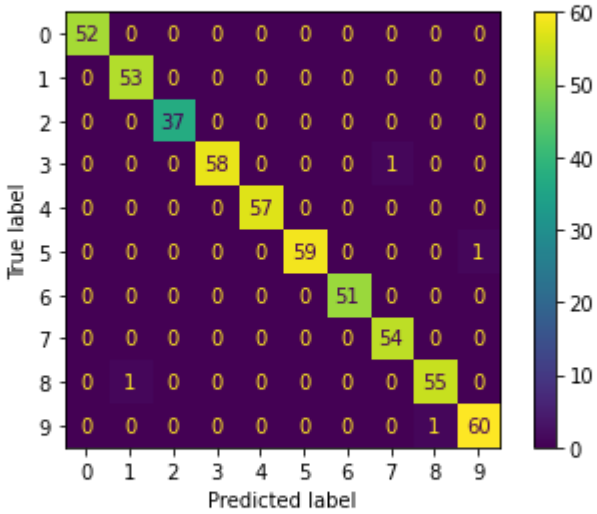


```
In [68]: print(metrics.classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        52
           1       0.98      1.00      0.99        53
           2       1.00      1.00      1.00        37
           3       1.00      0.98      0.99        59
           4       1.00      1.00      1.00        57
           5       1.00      0.98      0.99        60
           6       1.00      1.00      1.00        51
           7       0.98      1.00      0.99        54
           8       0.98      0.98      0.98        56
           9       0.98      0.98      0.98        61

    accuracy                           0.99       540
   macro avg       0.99      0.99      0.99       540
weighted avg       0.99      0.99      0.99       540
```

In [72]:
```
cm_plot = metrics.ConfusionMatrixDisplay(metrics.confusion_matrix(y_t
cm_plot.plot()

# this confusion matrix is a sparse matrix as it has a lot of zeroes.
```

Out[72]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0
x74ce8769fca0>
```



	0	1	2	3	4	5	6	7	8	9
0	52	0	0	0	0	0	0	0	0	0
1	0	53	0	0	0	0	0	0	0	0
2	0	0	37	0	0	0	0	0	0	0
3	0	0	0	58	0	0	0	1	0	0
4	0	0	0	0	57	0	0	0	0	0
5	0	0	0	0	0	59	0	0	0	1
6	0	0	0	0	0	0	51	0	0	0
7	0	0	0	0	0	0	0	54	0	0
8	0	1	0	0	0	0	0	0	55	0
9	0	0	0	0	0	0	0	0	1	60

Question 2

In [77]:
```
from sklearn.feature_selection import SelectKBest,chi2,f_classif,f_re
X, y = datasets.load_digits(return_X_y=True)
```

In [78]:
```
X_new = SelectKBest(chi2, k=20).fit_transform(X, y)
X_new.shape
```

Out[78]:
```
(1797, 20)
```

In [80]:
```
X_new = SelectKBest(f_classif, k=20).fit_transform(X, y)
X_new.shape
```

```
/opt/anaconda3/lib/python3.9/site-packages/sklearn/feature_selectio
n/_univariate_selection.py:114: UserWarning: Features [ 0 32 39] ar
e constant.
  warnings.warn("Features %s are constant." % constant_features_id
x,
/opt/anaconda3/lib/python3.9/site-packages/sklearn/feature_selectio
n/_univariate_selection.py:116: RuntimeWarning: invalid value encou
ntered in true_divide
  f = msb / msw
```

Out[80]:
```
(1797, 20)
```

```
In [81]: X_new = SelectKBest(f_regression, k=20).fit_transform(X, y)
         X_new.shape

         /opt/anaconda3/lib/python3.9/site-packages/sklearn/feature_selectio
         n/_univariate_selection.py:302: RuntimeWarning: invalid value encou
         ntered in true_divide
           corr /= X_norms

Out[81]: (1797, 20)
```

Question 3

```
In [83]: from sklearn.feature_selection import SelectPercentile, chi2 , f_clas

In [84]: X_new = SelectPercentile(chi2, percentile=10).fit_transform(X, y)

         X_new.shape

Out[84]: (1797, 7)

In [85]: X_new = SelectPercentile(f_classif, percentile=10).fit_transform(X, y
         X_new.shape

         /opt/anaconda3/lib/python3.9/site-packages/sklearn/feature_selectio
         n/_univariate_selection.py:114: UserWarning: Features [ 0 32 39] ar
         e constant.
           warnings.warn("Features %s are constant." % constant_features_id
         x,
         /opt/anaconda3/lib/python3.9/site-packages/sklearn/feature_selectio
         n/_univariate_selection.py:116: RuntimeWarning: invalid value encou
         ntered in true_divide
           f = msb / msw

Out[85]: (1797, 7)

In [86]: X_new = SelectPercentile(f_regression, percentile=10).fit_transform(X

         X_new.shape

         /opt/anaconda3/lib/python3.9/site-packages/sklearn/feature_selectio
         n/_univariate_selection.py:302: RuntimeWarning: invalid value encou
         ntered in true_divide
           corr /= X_norms

Out[86]: (1797, 7)
```