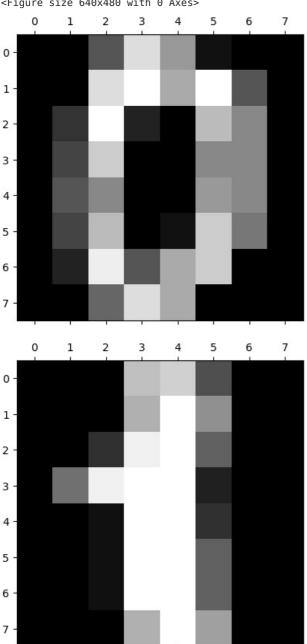
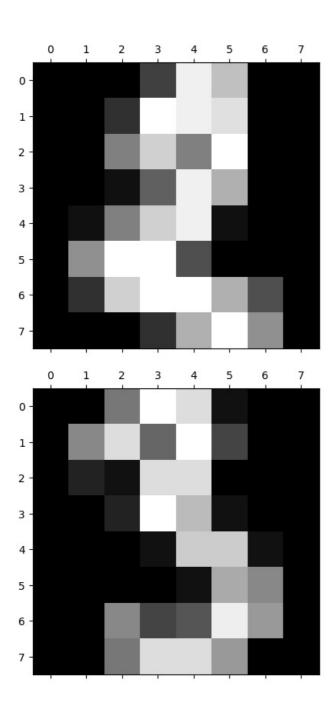
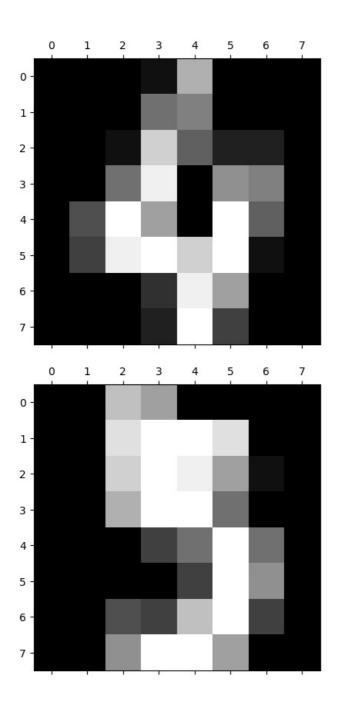
```
In [2]: import pandas as pd
    from sklearn.datasets import load_digits
           digits=load_digits()
```

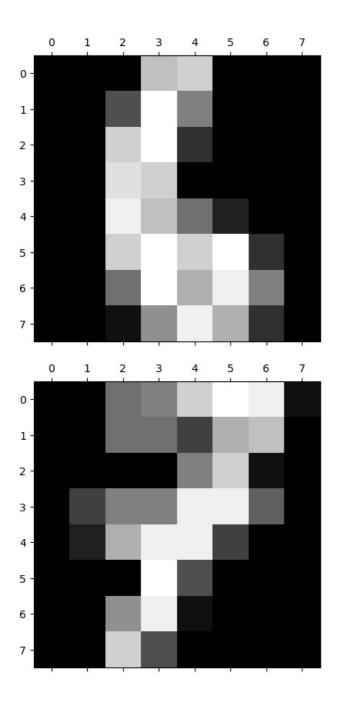
In [3]: plt.gray()
 for i in range(9):
 plt.matshow(digits.images[i])

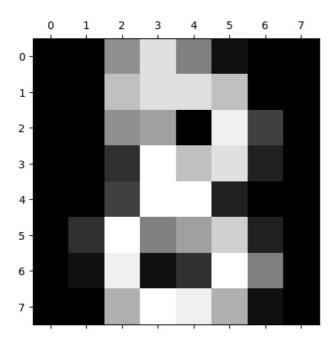
<Figure size 640x480 with 0 Axes>











In [6]: digits.feature\_names

```
Out[6]: ['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',
          'pixel_1_3',
          'pixel_1_4',
          'pixel_1_5',
'pixel_1_6',
          'pixel_1_7',
          'pixel_2_0',
          'pixel_2_1',
          'pixel_2_2',
          'pixel 2 3',
          'pixel_2_4',
           'pixel_2_5',
          'pixel_2_6',
          'pixel_2_7',
          'pixel_3_0',
           'pixel_3_1',
          'pixel_3_2',
          'pixel_3_3',
          'pixel_3_4',
           'pixel_3_5',
          'pixel_3_6',
          'pixel_3_7',
           'pixel_4_0',
           'pixel_4_1',
          'pixel_4_2',
          'pixel_4_3',
          'pixel_4_4',
'pixel_4_5',
          'pixel 4 6',
          'pixel_4_7',
          'pixel_5_0',
          'pixel_5_1',
          'pixel_5_2',
          'pixel_5_3',
           'pixel_5_4',
          'pixel_5_5',
          'pixel_5_6',
          'pixel_5_7',
           'pixel 6 0',
          'pixel_6_1',
          'pixel 6 2',
          'pixel_6_3',
           'pixel_6_4',
          'pixel_6_5',
          'pixel 6 6',
          'pixel_6_7',
          'pixel_7_0',
'pixel_7_1',
          'pixel 7 2',
          'pixel_7_3',
           'pixel_7_4',
          'pixel_7_5',
'pixel_7_6',
          'pixel_7_7']
In [7]: digits.target_names
Out[7]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
In [4]: df = pd.DataFrame(digits.data,columns=digits.feature names)
In [5]: df
```

	p	ixel_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	0 pixel_1_1		pixel_6	_6 pix
	0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0	0.0	0.0		C	0.0
	1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	0.0	0.0	0.0		C	0.0
	2	0.0	0.0	0.0	4.0	15.0		0.0	0.0	0.0				5.0
	3	0.0	0.0	7.0	15.0	13.0		0.0	0.0	0.0				0.0
	4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	0.0	0.0				0.0
	4700													
	1792 1793	0.0	0.0	4.0 6.0	10.0 16.0	13.0 13.0		0.0	0.0	0.0				.0
	1794	0.0	0.0	1.0	11.0	15.0		0.0	0.0	0.0				0.0
	1795	0.0	0.0	2.0	10.0	7.0		0.0	0.0	0.0				2.0
	1796	0.0	0.0	10.0	14.0	8.0		0.0	0.0	0.0				3.0
	1797 row	/s × 64 co	lumns											
	4													•
	df['tardf.head	-	digits.t	arget										
ut[8]:	pixel	_0_0 pix	cel_0_1 pi	xel_0_2 pix	cel_0_3 pi	xel_0_4 pi	xel_0_5 pi	xel_0_6 pi	cel_0_7 pi	xel_1_0 p	pixel_1_1	pix	kel_6_7	pixel_
	0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0	0.0	0.0		0.0	
	1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	0.0	0.0	0.0		0.0	
	2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	0.0	0.0	0.0		0.0	
	3	0.0	0.0	7.0	15.0	13.0	1.0	0.0	0.0	0.0	8.0		0.0	
	4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	0.0	0.0	0.0		0.0	
[10]: t[10]:			kel 0 1 pi	xel 0 2 pix	cel 0 3 pi	xel04 pi	xel 0 5 pi	xel 0 6 pi	kel07 pi	xel 10 r	pixel_1_1	cia	xel 7 0	pixel
	0	0.0	0.0	5.0	13.0	9.0				0.0			0.0	
	1	0.0				5.0	1.0	0.0	0.0	0.0	0.0			
	2		0.0	0.0	12.0	13.0	1.0 5.0	0.0	0.0	0.0	0.0		0.0	
	2	0.0	0.0	0.0	12.0 4.0									
	3	0.0				13.0	5.0	0.0	0.0	0.0	0.0		0.0	
	4		0.0	0.0	4.0	13.0 15.0	5.0 12.0	0.0	0.0	0.0	0.0 0.0		0.0	
Į.	4	0.0	0.0 0.0 0.0	0.0 7.0	4.0 15.0	13.0 15.0 13.0	5.0 12.0 1.0	0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 8.0		0.0 0.0 0.0	
	4	0.0	0.0 0.0 0.0	0.0 7.0	4.0 15.0	13.0 15.0 13.0	5.0 12.0 1.0	0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 8.0		0.0 0.0 0.0	
	4 5 rows ×	0.0 0.0 66 colum	0.0 0.0 0.0	0.0 7.0 0.0	4.0 15.0 1.0	13.0 15.0 13.0 11.0	5.0 12.0 1.0 0.0	0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 8.0		0.0 0.0 0.0	P
[11]:	4 5 rows × from sl	0.0 0.0 66 colum	0.0 0.0 0.0 ns	0.0 7.0	4.0 15.0 1.0 <b>ort</b> train	13.0 15.0 13.0 11.0	5.0 12.0 1.0 0.0	0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 8.0		0.0 0.0 0.0	<b> </b>
[11]: [12]: [13]:	from sl  x = df	0.0 0.0 66 colum xlearn.m	0.0 0.0 0.0 ins indel_sele	0.0 7.0 0.0	4.0 15.0 1.0 ort train e'],axis=	13.0 15.0 13.0 11.0	5.0 12.0 1.0 0.0	0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 8.0		0.0 0.0 0.0	Þ
[11]: [12]: [13]:	4 5 rows × from sl x = df x = df y = df	0.0 0.0 66 colum clearn.m drop(['	0.0 0.0 0.0 ins indel_sele target','	0.0 7.0 0.0 ction imp digit_nam	4.0 15.0 1.0 ort train e'],axis= e'],axis=	13.0 15.0 13.0 11.0 _test_spl 'columns'	5.0 12.0 1.0 0.0	0.0 0.0 0.0 0.0	0.0 0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 8.0		0.0 0.0 0.0	
[11]: [12]: [13]: [14]:	from sl x = df x = df x = df x_train	0.0 0.0 66 colum clearn.m drop([' drop([' target n, x_tes	0.0 0.0 0.0 ins indel_sele target','	0.0 7.0 0.0 ction imp digit_nam	4.0 15.0 1.0 ort train e'],axis= e'],axis=	13.0 15.0 13.0 11.0 _test_spl 'columns'	5.0 12.0 1.0 0.0	0.0 0.0 0.0 0.0	0.0 0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 8.0		0.0 0.0 0.0	
[11]: [ [12]: [ [13]: [ [14]: [ [15]: [	from sl x = df x = df x = df x_train len(x_1)	0.0 0.0 66 colum clearn.m drop([' drop([' target n, x_tes	0.0 0.0 0.0 ins indel_sele target','	0.0 7.0 0.0 ction imp digit_nam	4.0 15.0 1.0 ort train e'],axis= e'],axis=	13.0 15.0 13.0 11.0 _test_spl 'columns'	5.0 12.0 1.0 0.0	0.0 0.0 0.0 0.0	0.0 0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 8.0		0.0 0.0 0.0	
[11]: [12]: [13]: [14]: [15]: [t [15]:	4 5 rows ×  from sl  x = df  x = df  y = df  x_train  len(x_1  1437	0.0 0.0 66 colum clearn.m drop([' drop([' target n, x_tes	0.0 0.0 0.0 ins indel_sele target','	0.0 7.0 0.0 ction imp digit_nam	4.0 15.0 1.0 ort train e'],axis= e'],axis=	13.0 15.0 13.0 11.0 _test_spl 'columns'	5.0 12.0 1.0 0.0	0.0 0.0 0.0 0.0	0.0 0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 8.0		0.0 0.0 0.0	
[11]: [12]: [13]:	from sl x = df x = df y = df x_train len(x_1) len(x_1)	0.0 0.0 66 colum clearn.m drop([' drop([' target n, x_tes	0.0 0.0 0.0 ins indel_sele target','	0.0 7.0 0.0 ction imp digit_nam	4.0 15.0 1.0 ort train e'],axis= e'],axis=	13.0 15.0 13.0 11.0 _test_spl 'columns'	5.0 12.0 1.0 0.0	0.0 0.0 0.0 0.0	0.0 0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 8.0		0.0 0.0 0.0	
[11]: [12]: [13]: [14]: [15]: [16]: [16]:	from sl x = df x = df y = df x_train len(x_1) 1437 len(x_2) 360	0.0 0.0 66 colum clearn.m drop([' drop([' target n, x_tes train) test)	0.0 0.0 0.0 ns odel_sele target',' target','	0.0 7.0 0.0 ction imp digit_nam	4.0 15.0 1.0  ort train e'],axis= e'],axis= = train_	13.0 15.0 13.0 11.0  _test_spl 'columns' 'columns'	5.0 12.0 1.0 0.0	0.0 0.0 0.0 0.0	0.0 0.0 0.0 0.0	0.0 0.0 0.0	0.0 0.0 8.0		0.0 0.0 0.0	

```
KNeighborsClassifier(n_neighbors=3)
In [24]: knn.score(x_test, y_test)
Out[24]: 0.991666666666667
In [26]: from sklearn.metrics import confusion_matrix
         y_pred = knn.predict(x_test)
         cm = confusion_matrix(y_test, y_pred)
         cm
Out[26]: array([[33, 0, 0,
                              Θ,
                                  Θ,
                                      Θ,
                                          0,
                                              Θ,
                                                  Θ,
                                                      0],
                [ 0, 35, 0, 0,
                                  0,
                                      0,
                                          Θ,
                                              0,
                                                  0,
                                                      0],
                [ 0, 0, 27, 0,
                                  Θ,
                                      Θ,
                                          Θ,
                                              Θ,
                                                  Θ,
                                                      0],
                          0, 44,
                [ 0,
                      0,
                                  Ο,
                                      0,
                                          Ο,
                                              1,
                                                  0,
                                                      0],
                      0, 0, 0, 37,
                                      0, 0, 0,
                                                      0],
                [ 0,
                                                  Θ,
                [ 0, 0, 0, 0, 0, 33, 0, 0, 0,
                                                      1],
                                      0, 33, 0, 0,
                          Θ,
                                                      0],
                [ 0,
                      Θ,
                              Θ,
                                  Θ,
                                      Θ,
                                         0, 39, 0,
                [ 0,
                      Θ,
                          Θ,
                              Θ,
                                  Θ,
                                                     0],
                              Θ,
                                  Θ,
                      1,
                                      0, 0, 0, 39, 0],
                [ 0,
                          0,
                                         0, 0, 0, 37]], dtype=int64)
                [ 0,
                     Θ,
                         0, 0, 0,
                                     Θ,
In [27]: %matplotlib inline
         import matplotlib.pyplot as plt
         import seaborn as sn
         sn.heatmap(cm, annot=True)
         plt.xlabel('Predicted')
         plt.ylabel('Truth')
         plt.show()
                                                             0
                     0
                          0
                               0
                                         0
                                              0
                                                   0
                                                        0
                                                                      - 40
                0
                    35
                          0
                               0
                                    0
                                         0
                                              0
                                                   0
                                                        0
                                                             0
                                                                      - 35
                0
                     0
                               0
                                    0
                                         0
                                              0
                                                   0
                                                        0
                                                             0
           2
                                                                      - 30
                          0
                              44
                                    0
                0
                     0
                                         0
                                              0
                                                   1
                                                        0
                                                             0
           3
                                                                      25
                0
                     0
                          0
                               0
                                   37
                                         0
                                              0
                                                   0
                                                        0
                                                             0
           4
                0
                               0
                                    0
                                              0
                                                                      - 20
           2
                     0
                          0
                                                   0
                                                        0
                                                             1
                0
                               0
                                    0
                                         0
                                                   0
                     0
                          0
                                                        0
                                                             0
           9
                                                                      - 15
                0
                     0
                          0
                               0
                                    0
                                              0
                                                  39
                                                        0
                                                             0
                                         0
                                                                      - 10
                0
                     1
                          0
                               0
                                    0
                                         0
                                              0
                                                   0
                                                       39
                                                             0
           \infty
                                                                      - 5
                                    0
                0
                     0
                          0
                               0
                                         0
                                              0
                                                   0
                                                        0
                                                             37
          6
                          2
                                         5
                                                   7
                0
                     1
                               3
                                    4
                                              6
                                                             9
                                                        8
                                   Predicted
```

In [ ]:

Out[23]:

**KNeighborsClassifier** 

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js