Experiment 6: Implement Support Vector Machine by using Digit Dataset

In [2]:

```
import pandas as pd
from sklearn.datasets import load_digits
digits = load_digits()
```

In [4]:

```
dir(digits)
```

Out[4]:

```
['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_nam
es']
```

In [6]:

```
df = pd.DataFrame(digits.data, columns=digits.feature_names)
df.head()
```

Out[6]:

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5	pixel_0_6	pi
0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	
1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	
2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	
3	0.0	0.0	7.0	15.0	13.0	1.0	0.0	
4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	

5 rows × 64 columns

In [8]:

```
df['target'] = digits.target
df.head()
```

Out[8]:

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5	pixel_0_6	pi
0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	
1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	
2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	
3	0.0	0.0	7.0	15.0	13.0	1.0	0.0	
4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	

5 rows × 65 columns

In [11]:

```
digits.target_names
```

Out[11]:

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [13]:

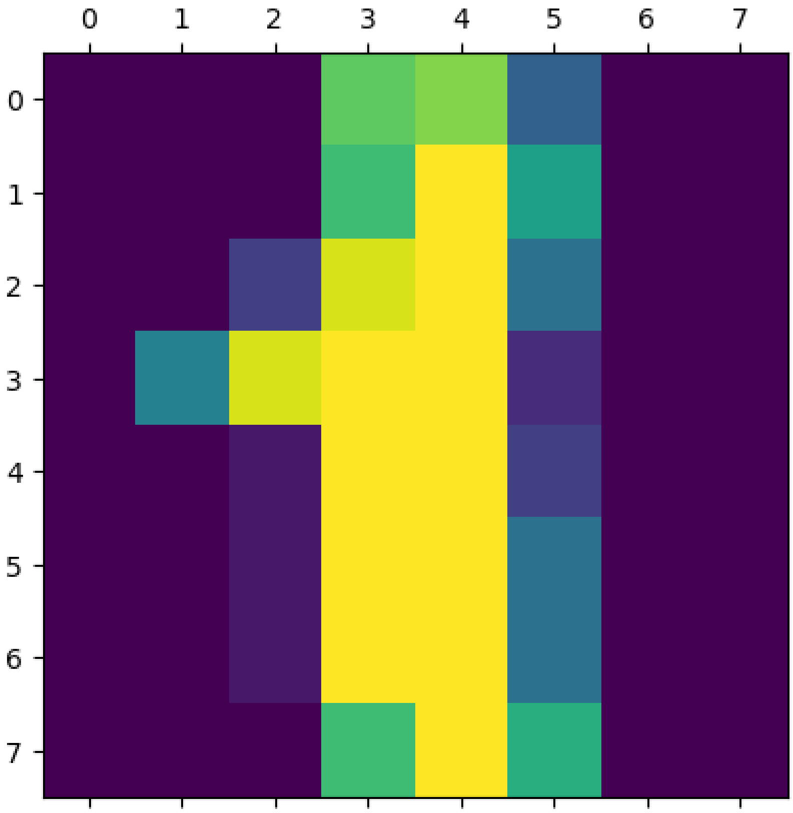
```
df.shape
```

```
Out[13]: (1797, 65)
```

```
In [17]: from matplotlib import pyplot as plt
         %matplotlib inline
```

```
In [31]: plt.matshow(digits.images[1])
```

```
Out[31]: <matplotlib.image.AxesImage at 0x1c14cc90e60>
```



```
In [19]: df0 = df[df.target==0]
         df1 = df[df.target==1]
         df2 = df[df.target==2]
         df3 = df[df.target==3]
         df4 = df[df.target==4]
         df5 = df[df.target==5]
         df6 = df[df.target==6]
         df7 = df[df.target==7]
         df8 = df[df.target==8]
         df9 = df[df.target==9]
```

```
In [29]: from sklearn.model_selection import train_test_split
         x = df.drop(['target'],axis='columns')
         x.head()
```

Out[29]:

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5	pixel_0_6	pi
0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	
1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	
2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	
3	0.0	0.0	7.0	15.0	13.0	1.0	0.0	
4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	

5 rows × 64 columns

In [33]:

```
y = df.target
```

In [37]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,shuffle=T
```

In [41]:

```
from sklearn.svm import SVC
model = SVC(kernel='poly', C = 3, gamma = 'scale')
model.fit(x_train, y_train)
model.score(x_test,y_test)
```

Out[41]:

```
0.9833333333333333
```

This notebook was converted with convert.ploomber.io