hwdr

August 27, 2023

[1]:
```
import numpy as np # linear algebra
from sklearn.datasets import load_digits
```

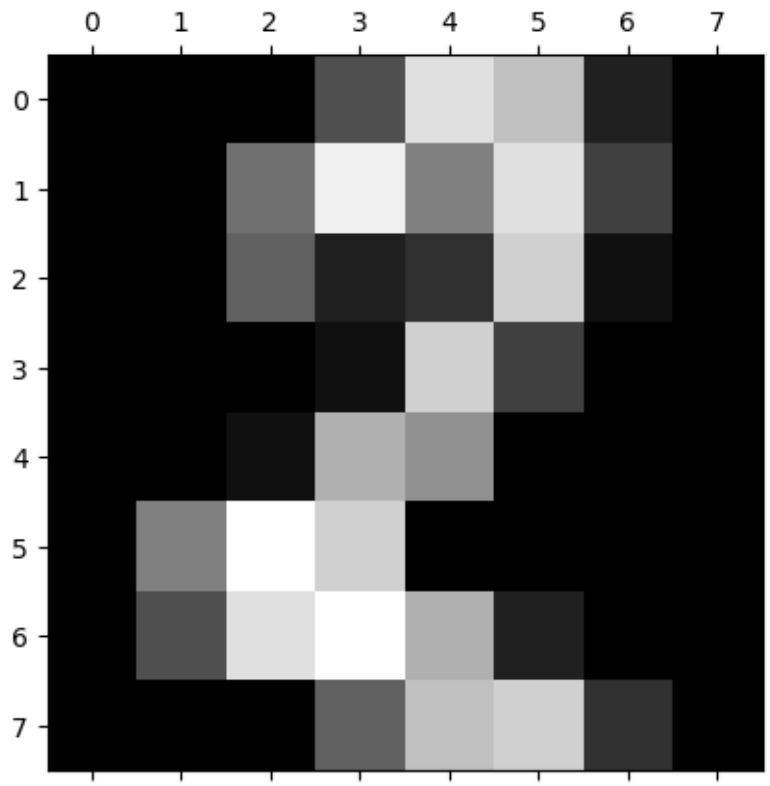
[2]:
```
dataset=load_digits()
```

[3]:
```
print(dataset.data)
print(dataset.target)
print(dataset.data.shape)
print(dataset.images.shape)
dataimageLength=len(dataset.images)
print(dataimageLength)
```

```
[[ 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.]]
[0 1 2 … 8 9 8]
(1797, 64)
(1797, 8, 8)
1797
```

[123]:
```
n=50
import matplotlib.pyplot as plt
plt.gray()
plt.matshow(dataset.images[n])
plt.show()
dataset.images[n]
```

```
<Figure size 640x480 with 0 Axes>
```



[123]:
```
array([[ 0.,  0.,  0.,  5., 14., 12.,  2.,  0.],
       [ 0.,  0.,  7., 15.,  8., 14.,  4.,  0.],
       [ 0.,  0.,  6.,  2.,  3., 13.,  1.,  0.],
       [ 0.,  0.,  0.,  1., 13.,  4.,  0.,  0.],
       [ 0.,  0.,  1., 11.,  9.,  0.,  0.,  0.],
       [ 0.,  8., 16., 13.,  0.,  0.,  0.,  0.],
       [ 0.,  5., 14., 16., 11.,  2.,  0.,  0.],
       [ 0.,  0.,  0.,  6., 12., 13.,  3.,  0.]])
```

[124]:
```
X=dataset.images.reshape((dataimageLength,-1))
X
```

[124]:
```
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.]])
```

[125]:
```
Y=dataset.target
Y
```

[125]:
```
array([0, 1, 2, …, 8, 9, 8])
```

[126]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.
  ↪25,random_state=0)
print(X_train.shape)
print(X_test.shape)
```

```
(1347, 64)
(450, 64)
```

[127]:
```
import matplotlib.pyplot as plt
from sklearn.svm import SVC
```

[128]:
```
model = SVC()  # Initialize the SVM model
model.fit(X_train, y_train)  # Fit the model to your training data
```

[128]:
```
SVC()
```

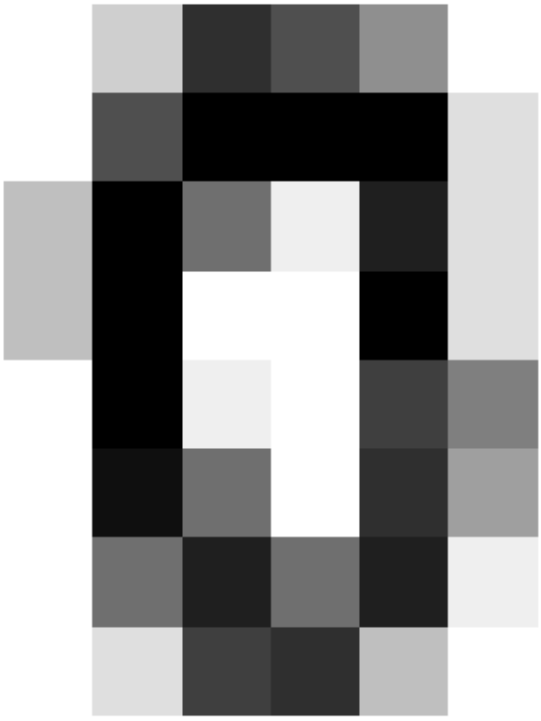
[129]:
```
result = model.predict(dataset.images[n].reshape((1, -1)))
```

[130]:
```
import matplotlib.pyplot as plt
from sklearn.svm import SVC
n=20
# Assuming you have X_train and y_train as your training data
model = SVC()  # Initialize the SVM model
model.fit(X_train, y_train)  # Fit the model to your training data

result = model.predict(dataset.images[n].reshape((1, -1)))

plt.imshow(dataset.images[n], cmap=plt.cm.gray_r, interpolation='nearest')
print(result)
print("/n")
plt.axis('off')
plt.show()
```

```
[0]
/n
```



[131]:
```
from sklearn import svm
model1=svm.SVC(kernel='linear')
model2=svm.SVC(kernel='rbf')
model3=svm.SVC(gamma=0.001)
model4=svm.SVC(gamma=0.001,C=0.8)
model1.fit(X_train,y_train)
model2.fit(X_train,y_train)
model3.fit(X_train,y_train)
model4.fit(X_train,y_train)
y_predModel1=model1.predict(X_test)
y_predModel2=model2.predict(X_test)
y_predModel3=model3.predict(X_test)
y_predModel4=model4.predict(X_test)
print("accuracy of the model1:{0}%".
  ↪format(accuracy_score(y_test,y_predModel1)*100))
print("accuracy of the model2:{0}%".
  ↪format(accuracy_score(y_test,y_predModel2)*100))
print("accuracy of the model3:{0}%".
  ↪format(accuracy_score(y_test,y_predModel3)*100))
print("accuracy of the model4:{0}%".
  ↪format(accuracy_score(y_test,y_predModel4)*100))
```

```
accuracy of the model1:97.11111111111111%
```

```
accuracy of the model2:99.11111111111111%
accuracy of the model3:99.55555555555556%
accuracy of the model4:99.55555555555556%
```

[132]:
```
#prediction for test data
y_pred=model.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1),y_test.
 ↪reshape(len(y_test),1)),1))
```

```
[[2 2]
 [8 8]
 [2 2]
 [6 6]
 [6 6]
 [7 7]
 [1 1]
 [9 9]
 [8 8]
 [5 5]
 [2 2]
 [8 8]
 [6 6]
 [6 6]
 [6 6]
 [6 6]
 [1 1]
 [0 0]
 [5 5]
 [8 8]
 [8 8]
 [7 7]
 [8 8]
 [4 4]
 [7 7]
 [5 5]
 [4 4]
 [9 9]
 [2 2]
 [9 9]
 [4 4]
 [7 7]
 [6 6]
 [8 8]
 [9 9]
 [4 4]
 [3 3]
 [1 1]
 [0 0]
```

```
[1 1]
[8 8]
[6 6]
[7 7]
[7 7]
[1 1]
[0 0]
[7 7]
[6 6]
[2 2]
[1 1]
[9 9]
[6 6]
[7 7]
[9 9]
[0 0]
[0 0]
[5 5]
[1 1]
[6 6]
[3 3]
[0 0]
[2 2]
[3 3]
[4 4]
[1 1]
[9 9]
[2 2]
[6 6]
[9 9]
[1 1]
[8 8]
[3 3]
[5 5]
[1 1]
[2 2]
[8 8]
[2 2]
[2 2]
[9 9]
[7 7]
[2 2]
[3 3]
[6 6]
[0 0]
[5 5]
[3 3]
[7 7]
```

```
[5 5]
[1 1]
[2 2]
[9 9]
[9 9]
[3 3]
[1 1]
[7 7]
[7 7]
[4 4]
[8 8]
[5 5]
[8 8]
[5 5]
[5 5]
[2 2]
[5 5]
[9 9]
[0 0]
[7 7]
[1 1]
[4 4]
[7 7]
[3 3]
[4 4]
[8 8]
[9 9]
[7 7]
[9 9]
[8 8]
[2 2]
[6 6]
[5 5]
[2 2]
[5 5]
[8 8]
[4 4]
[1 8]
[7 7]
[0 0]
[6 6]
[1 1]
[5 5]
[5 9]
[9 9]
[9 9]
[5 5]
[9 9]
```

```
[9 9]
[5 5]
[7 7]
[5 5]
[6 6]
[2 2]
[8 8]
[6 6]
[9 9]
[6 6]
[1 1]
[5 5]
[1 1]
[5 5]
[9 9]
[9 9]
[1 1]
[5 5]
[3 3]
[6 6]
[1 1]
[8 8]
[9 9]
[8 8]
[7 7]
[6 6]
[7 7]
[6 6]
[5 5]
[6 6]
[0 0]
[8 8]
[8 8]
[9 9]
[8 8]
[6 6]
[1 1]
[0 0]
[4 4]
[1 1]
[6 6]
[3 3]
[8 8]
[6 6]
[7 7]
[4 4]
[9 5]
[6 6]
```

[3 3]
[0 0]
[3 3]
[3 3]
[3 3]
[0 0]
[7 7]
[7 7]
[5 5]
[7 7]
[8 8]
[0 0]
[7 7]
[8 8]
[9 9]
[6 6]
[4 4]
[5 5]
[0 0]
[1 1]
[4 4]
[6 6]
[4 4]
[3 3]
[3 3]
[0 0]
[9 9]
[5 5]
[9 9]
[2 2]
[1 1]
[4 4]
[2 2]
[1 1]
[6 6]
[8 8]
[9 9]
[2 2]
[4 4]
[9 9]
[3 3]
[7 7]
[6 6]
[2 2]
[3 3]
[3 3]
[1 1]
[6 6]

```
[9 9]
[3 3]
[6 6]
[3 3]
[2 2]
[2 2]
[0 0]
[7 7]
[6 6]
[1 1]
[1 1]
[9 9]
[7 7]
[2 2]
[7 7]
[8 8]
[5 5]
[5 5]
[7 7]
[5 5]
[2 2]
[3 3]
[7 7]
[2 2]
[7 7]
[5 5]
[5 5]
[7 7]
[0 0]
[9 9]
[1 1]
[6 6]
[5 5]
[9 9]
[7 7]
[4 4]
[3 3]
[8 8]
[0 0]
[3 3]
[6 6]
[4 4]
[6 6]
[3 3]
[2 2]
[6 6]
[8 8]
[8 8]
```

[8 8]
[4 4]
[6 6]
[7 7]
[5 5]
[2 2]
[4 4]
[5 5]
[3 3]
[2 2]
[4 4]
[6 6]
[9 9]
[4 4]
[5 5]
[4 4]
[3 3]
[4 4]
[6 6]
[2 2]
[9 9]
[0 0]
[1 1]
[7 7]
[2 2]
[0 0]
[9 9]
[6 6]
[0 0]
[4 4]
[2 2]
[0 0]
[7 7]
[9 9]
[8 8]
[5 5]
[4 4]
[8 8]
[2 2]
[8 8]
[4 4]
[3 3]
[7 7]
[2 2]
[6 6]
[9 9]
[1 1]
[5 5]

[1 1]
[0 0]
[8 8]
[2 2]
[1 1]
[9 9]
[5 5]
[6 6]
[8 8]
[2 2]
[7 7]
[2 2]
[1 1]
[5 5]
[1 1]
[6 6]
[4 4]
[5 5]
[0 0]
[9 9]
[4 4]
[1 1]
[1 1]
[7 7]
[0 0]
[8 8]
[9 9]
[0 0]
[5 5]
[4 4]
[3 3]
[8 8]
[8 8]
[6 6]
[5 5]
[3 3]
[4 4]
[4 4]
[4 4]
[8 8]
[8 8]
[7 7]
[0 0]
[9 9]
[6 6]
[3 3]
[5 5]
[2 2]

[3 3]
[0 0]
[8 8]
[8 3]
[3 3]
[1 1]
[3 3]
[3 3]
[0 0]
[0 0]
[4 4]
[6 6]
[0 0]
[7 7]
[7 7]
[6 6]
[2 2]
[0 0]
[4 4]
[4 4]
[2 2]
[3 3]
[7 7]
[8 8]
[9 9]
[8 8]
[6 6]
[8 8]
[5 5]
[6 6]
[2 2]
[2 2]
[3 3]
[1 1]
[7 7]
[7 7]
[8 8]
[0 0]
[3 3]
[3 3]
[2 2]
[1 1]
[5 5]
[5 5]
[9 9]
[1 1]
[3 3]
[7 7]

```
[0 0]
[0 0]
[7 7]
[0 0]
[4 4]
[5 5]
[9 9]
[3 3]
[3 3]
[4 4]
[3 3]
[1 1]
[8 8]
[9 9]
[8 8]
[3 3]
[6 6]
[2 2]
[1 1]
[6 6]
[2 2]
[1 1]
[7 7]
[5 5]
[5 5]
[1 1]
[9 9]]
```

[133]:
```
#evaluate model-accuracy score
from sklearn.metrics import accuracy_score
print("Accuracy of the Model:{0}%".format(accuracy_score(y_test,y_pred)*100))
```

```
Accuracy of the Model:99.11111111111111%
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: