

✓ Import Library

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

✓ Import Data

```
from sklearn.datasets import load_digits
```

✓ Data Preprocessing

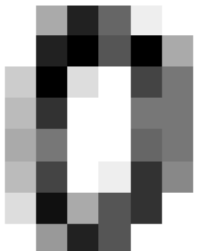
Flatten Image

```
df = load_digits()
```

```
_, axes = plt.subplots(nrows = 1, ncols = 4, figsize = (10, 3))  
for ax, image, label in zip(axes, df.images, df.target):  
    ax.set_axis_off()  
    ax.imshow(image, cmap=plt.cm.gray_r, interpolation = 'nearest')  
    ax.set_title('Training: %i' % label)
```



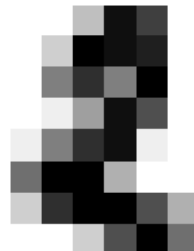
Training: 0



Training: 1



Training: 2




Training: 3




```
df.images.shape
```

 (1797, 8, 8)

df.images[0]


 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.]])

df.images[0].shape


 (8, 8)

```
n_samples = len(df.images)  
data = df.images.reshape((n_samples, -1))
```

data[0]

 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.]])

data[0].shape

 (64,)

data.shape


 (1797, 64)

✓ Scaling Data

data.min()

 0.0

data.max()

 16.0

```
data = data/16
```


```
data.min()
```

 0.0

```
data.max()
```

 1.0

```
data[0]
```




```
array([0.      , 0.      , 0.3125, 0.8125, 0.5625, 0.0625, 0.      , 0.      ,
        0.      , 0.      , 0.8125, 0.9375, 0.625 , 0.9375, 0.3125, 0.      ,
        0.      , 0.1875, 0.9375, 0.125 , 0.      , 0.6875, 0.5   , 0.      ,
        0.      , 0.25  , 0.75  , 0.      , 0.      , 0.5   , 0.5   , 0.      ,
        0.      , 0.3125, 0.5   , 0.      , 0.      , 0.5625, 0.5   , 0.      ,
        0.      , 0.25  , 0.6875, 0.      , 0.0625, 0.75  , 0.4375, 0.      ,
        0.      , 0.125 , 0.875 , 0.3125, 0.625 , 0.75  , 0.      , 0.      ,
        0.      , 0.      , 0.375 , 0.8125, 0.625 , 0.      , 0.      , 0.      ])
```

✓ Train test split Data

```
from sklearn.model_selection import train_test_split
```

```
xtrain, xtest, ytrain, ytest = train_test_split(data, df.target, test_size = 0.3)
```

```
xtrain.shape, xtest.shape, ytrain.shape, ytest.shape
```






```
((1257, 64), (540, 64), (1257,), (540,))
```

✓ Random Forest model

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf = RandomForestClassifier()
```


```
rf.fit(xtrain, ytrain)
```


 RandomForestClassifier  
 RandomForestClassifier()

✓ Predict Test Data

```
y_pred = rf.predict(xtest)
```

y_pred


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

✓ Model Accuracy

```
from sklearn.metrics import confusion_matrix, classification_report
```

```
confusion_matrix(ytest, y_pred)
```

```
array([[56, 0, 0, 0, 1, 0, 0, 0, 0, 0],
       [ 0, 52, 0, 0, 0, 0, 0, 0, 0, 0],
       [ 0, 0, 57, 0, 0, 0, 0, 0, 0, 0],
       [ 0, 0, 0, 60, 0, 0, 0, 0, 0, 0],
       [ 0, 0, 0, 0, 42, 0, 0, 0, 0, 0],
       [ 0, 0, 0, 0, 0, 60, 0, 0, 0, 0],
       [ 1, 0, 1, 0, 0, 0, 48, 0, 0, 0],
       [ 0, 0, 0, 0, 0, 0, 0, 44, 0, 1],
       [ 0, 2, 0, 1, 0, 0, 0, 1, 63, 1],
```

```
print(classification_report(ytest, y_pred))
```

```
precision    recall  f1-score   support

0           0.98      0.98      0.98        57
1           0.96      1.00      0.98        52
2           0.98      1.00      0.99        57
3           0.98      1.00      0.99        60
4           0.98      1.00      0.99        42
5           1.00      1.00      1.00        60
6           1.00      0.96      0.98        50
7           0.98      0.98      0.98        45
8           0.98      0.93      0.95        68
9           0.96      0.98      0.97        49

accuracy                0.98        540
macro avg              0.98      0.98      0.98        540
weighted avg           0.98      0.98      0.98        540
```