```python
import pandas as pd
```

```python
import numpy as np
```

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

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

```python
df = load_digits()
```

```python
_, 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)
```



```python
df.images.shape
```

```
(1797, 8, 8)
```

```python
df.images[0]
```

Show hidden output

```python
df.images[0].shape
```

```
(8, 8)
```

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

```python
data[0]
```

Show hidden output

```python
data.min()
```

Show hidden output

```python
data.max()
```

Show hidden output

```python
data = data/16
```

```python
data.min()
```

Show hidden output

```python
data.max()
```

Show hidden output

```python
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.    ])
```

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

```python
x_train,x_test , y_train , y_test = train_test_split(data , df.target, test_size = 0.3)
```

```python
x_train.shape , x_test.shape,y_train.shape , y_test.shape
```

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

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

```python
rf = RandomForestClassifier()
```

```python
rf.fit(x_train , y_train)
```

Show hidden output

```python
y_pred = rf.predict(x_test)
```

```python
y_pred
```

Show hidden output

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

```python
confusion_matrix(y_test, y_pred)
```

```
array([[44,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 54,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0, 59,  0,  0,  0,  0,  0,  0,  1],
       [ 0,  0,  0, 64,  0,  0,  0,  0,  1,  0],
       [ 0,  0,  0,  0, 40,  0,  0,  1,  0,  1],
       [ 0,  0,  0,  0,  1, 55,  0,  0,  0,  1],
       [ 1,  1,  0,  0,  1,  0, 55,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0, 45,  0,  0],
       [ 0,  5,  0,  0,  1,  1,  0,  0, 44,  0],
       [ 0,  0,  0,  0,  0,  1,  0,  0,  1, 62]])
```

```python
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        44
           1       0.90      1.00      0.95        54
           2       1.00      0.98      0.99        60
           3       1.00      0.98      0.99        65
           4       0.93      0.95      0.94        42
           5       0.96      0.96      0.96        57
           6       1.00      0.95      0.97        58
           7       0.98      1.00      0.99        45
           8       0.96      0.86      0.91        51
           9       0.95      0.97      0.96        64

    accuracy                           0.97       540
   macro avg       0.97      0.97      0.97       540
weighted avg       0.97      0.97      0.97       540
```