```
import numpy as np
import pandas as pd
```

```
import tensorflow
from tensorflow import keras
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, Flatten
```

```
(xtrain, ytrain), (xtest,ytest) = keras.datasets.mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

```
xtrain.shape
```

```
(60000, 28, 28)
```

```
xtrain.ndim
```

```
3
```

```
xtrain[1]
```

```
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  51, 238, 253,
        253, 190, 114, 253, 228,  47,  79, 255, 168,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,  48, 238, 252, 252,
        179,  12,  75, 121,  21,   0,   0, 253, 243,  50,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  38, 165, 253, 233, 208,
         84,   0,   0,   0,   0,   0,   0, 253, 252, 165,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   7, 178, 252, 240,  71,  19,
         28,   0,   0,   0,   0,   0,   0, 253, 252, 195,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  57, 252, 252,  63,   0,   0,
          0,   0,   0,   0,   0,   0,   0, 253, 252, 195,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0, 198, 253, 190,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0, 255, 253, 196,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,  76, 246, 252, 112,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0, 253, 252, 148,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,  85, 252, 230,  25,   0,   0,   0,
          0,   0,   0,   0,   0,   7, 135, 253, 186,  12,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,  85, 252, 223,   0,   0,   0,   0,
          0,   0,   0,   0,   7, 131, 252, 225,  71,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,  85, 252, 145,   0,   0,   0,   0,
          0,   0,   0,  48, 165, 252, 173,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,  86, 253, 225,   0,   0,   0,   0,
          0,   0, 114, 238, 253, 162,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,  85, 252, 249, 146,  48,  29,  85,
        178, 225, 253, 223, 167,  56,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,  85, 252, 252, 252, 229, 215, 252,
        252, 252, 196, 130,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,  28, 199, 252, 252, 253, 252, 252,
        233, 145,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  25, 128, 252, 253, 252, 141,
         37,   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,   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,   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,   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,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0]], dtype=uint8)
```

```
xtrain[0].shape
```

```
(28, 28)
```

```
xtrain
```

```
array([[[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, 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, 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, ..., 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, 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, 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],
        [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, 0],
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0]]], dtype=uint8)
```

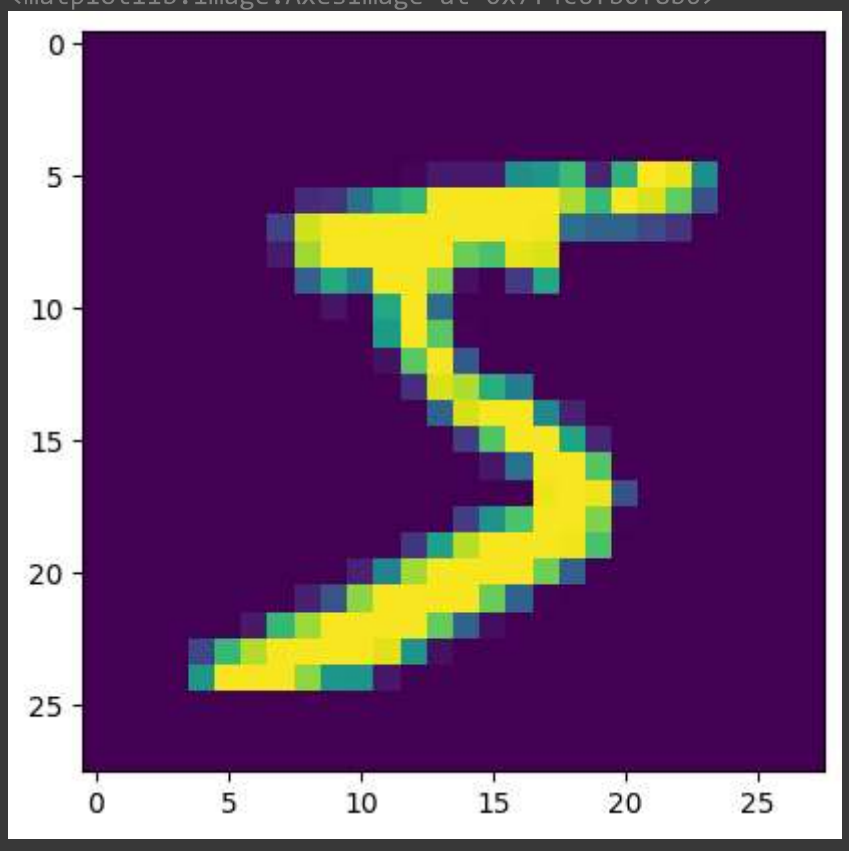
```
xtrain[0]
```

```
array([[  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,   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,   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,   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,   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,   0,   0,   0,   0,   0,   0,   0,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
        253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
        253, 198, 182, 247, 241,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  80, 156, 107, 253, 253,
        205,  11,   0,  43, 154,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,  14,   1, 154, 253,
         90,   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, 139, 253,
        190,   2,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  11, 190,
        253,  70,   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,  35,
   241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,   0,
     0,   0],
  [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
    81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,   0,   0,
     0,   0],
  [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
     0,  45, 186, 253, 253, 150,  27,   0,   0,   0,   0,   0,   0,
     0,   0],
  [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
     0,   0,  16,  93, 252, 253, 187,   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, 249, 253, 249,  64,   0,   0,   0,   0,   0,
     0,   0],
  [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
     0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,   0,
     0,   0],
  [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39,
```

```
import matplotlib.pyplot as plt
plt.imshow(xtrain[0])
```

```
<matplotlib.image.AxesImage at 0x7f4c6fb0f8b0>
```




```
xtest[0]
```

```
    0,   0],
  [  0,   0,   0,   0,   0,   0,  67, 114,  72, 114, 163, 227, 254,
   225, 254, 254, 254, 250, 229, 254, 254, 140,   0,   0,   0,   0,
     0,   0],
  [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  17,  66,
    14,  67,  67,  67,  59,  21, 236, 254, 106,   0,   0,   0,   0,
     0,   0],
  [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
     0,   0,   0,   0,  83, 253, 209,  18,   0,   0,   0,   0,   0,
     0,   0],
  [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
     0,   0,   0,   0,  22, 233, 255,  83,   0,   0,   0,   0,   0,
     0,   0],
  [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
     0,   0,   0,   0, 129, 254, 238,  44,   0,   0,   0,   0,   0,
```

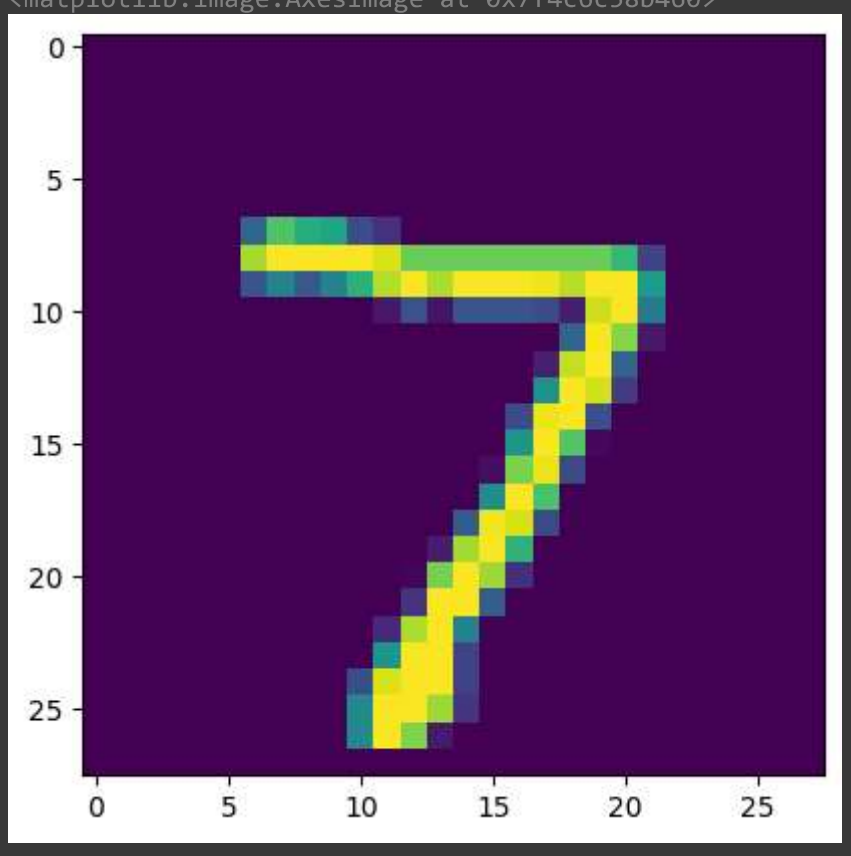
```
 254, 115,   1,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 133, 254,
 254,  52,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  61, 242, 254,
 254,  52,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 121, 254, 254,
 219,  40,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 121, 254, 207,
  18,   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,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0]], dtype=uint8)
```

```
xtest.shape
```

```
(10000, 28, 28)
```

```
plt.imshow(xtest[0])
```

```
<matplotlib.image.AxesImage at 0x7f4c6c58b460>
```




```
#### Scaling the value
xtrain = xtrain/255
xtest = xtest/255
```

```
xtrain
```

```
array([[[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., 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., 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., ..., 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., 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., 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.],
         [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., 0.],
         [0., 0., 0., ..., 0., 0., 0.],
         [0., 0., 0., ..., 0., 0., 0.]]])
```

```
xtrain[0]
```

```
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.97647059, 0.99215686, 0.97647059,
         0.25098039, 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.18039216,
         0.50980392, 0.71764706, 0.99215686, 0.99215686, 0.81176471,
         0.00784314, 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.15294118, 0.58039216, 0.89803922,
         0.99215686, 0.99215686, 0.99215686, 0.98039216, 0.71372549,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.09411765, 0.44705882, 0.86666667, 0.99215686, 0.99215686,
         0.99215686, 0.99215686, 0.78823529, 0.30588235, 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.09019608, 0.25882353,
         0.83529412, 0.99215686, 0.99215686, 0.99215686, 0.99215686,
         0.77647059, 0.31764706, 0.00784314, 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.07058824, 0.67058824, 0.85882353, 0.99215686,
         0.99215686, 0.99215686, 0.99215686, 0.76470588, 0.31372549,
         0.03529412, 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.21568627,
         0.6745098 , 0.88627451, 0.99215686, 0.99215686, 0.99215686,
         0.99215686, 0.95686275, 0.52156863, 0.04313725, 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.53333333,
         0.99215686, 0.99215686, 0.99215686, 0.83137255, 0.52941176,
         0.51764706, 0.0627451 , 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.        , 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.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
```

```
xtrain.shape
```

```
(60000, 28, 28)
```

```
xtest.shape
```

```
(10000, 28, 28)
```

```
##### Model Building

model = Sequential()
model.add(Flatten(input_shape = (28,28)))
model.add(Dense(units = 128,activation = 'relu' ))
```

```
model.add(Dense(units = 32, activation = 'relu'))

# add Output Layer
model.add(Dense(units = 10, activation = 'softmax'))
# activation will be set as 'softmax'

model.compile(loss = 'sparse_categorical_crossentropy', optimizer = 'adam', metrics

history = model.fit(xtrain,ytrain, epochs = 25, validation_split = 0.20)
```

```
Epoch 1/25
1500/1500 [==============================] - 10s 6ms/step - loss: 0.2886 - accuracy: 0.9158 - val_loss: 0.1429 - val_accuracy: 0.
Epoch 2/25
1500/1500 [==============================] - 7s 5ms/step - loss: 0.1220 - accuracy: 0.9630 - val_loss: 0.1183 - val_accuracy: 0.9
Epoch 3/25
1500/1500 [==============================] - 8s 6ms/step - loss: 0.0834 - accuracy: 0.9740 - val_loss: 0.1019 - val_accuracy: 0.9
Epoch 4/25
1500/1500 [==============================] - 7s 5ms/step - loss: 0.0609 - accuracy: 0.9810 - val_loss: 0.1024 - val_accuracy: 0.9
Epoch 5/25
1500/1500 [==============================] - 8s 6ms/step - loss: 0.0480 - accuracy: 0.9845 - val_loss: 0.1018 - val_accuracy: 0.9
Epoch 6/25
1500/1500 [==============================] - 10s 6ms/step - loss: 0.0391 - accuracy: 0.9872 - val_loss: 0.0927 - val_accuracy: 0.
Epoch 7/25
1500/1500 [==============================] - 7s 5ms/step - loss: 0.0304 - accuracy: 0.9900 - val_loss: 0.1020 - val_accuracy: 0.9
Epoch 8/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0249 - accuracy: 0.9914 - val_loss: 0.1081 - val_accuracy: 0.9
Epoch 9/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0216 - accuracy: 0.9927 - val_loss: 0.1078 - val_accuracy: 0.9
Epoch 10/25
1500/1500 [==============================] - 8s 5ms/step - loss: 0.0210 - accuracy: 0.9931 - val_loss: 0.1094 - val_accuracy: 0.9
Epoch 11/25
1500/1500 [==============================] - 8s 5ms/step - loss: 0.0164 - accuracy: 0.9946 - val_loss: 0.1094 - val_accuracy: 0.9
Epoch 12/25
1500/1500 [==============================] - 8s 5ms/step - loss: 0.0142 - accuracy: 0.9951 - val_loss: 0.1336 - val_accuracy: 0.9
Epoch 13/25
1500/1500 [==============================] - 7s 5ms/step - loss: 0.0152 - accuracy: 0.9948 - val_loss: 0.1305 - val_accuracy: 0.9
Epoch 14/25
1500/1500 [==============================] - 8s 6ms/step - loss: 0.0131 - accuracy: 0.9958 - val_loss: 0.1311 - val_accuracy: 0.9
Epoch 15/25
1500/1500 [==============================] - 7s 5ms/step - loss: 0.0115 - accuracy: 0.9960 - val_loss: 0.1272 - val_accuracy: 0.9
Epoch 16/25
1500/1500 [==============================] - 8s 5ms/step - loss: 0.0103 - accuracy: 0.9965 - val_loss: 0.1342 - val_accuracy: 0.9
Epoch 17/25
1500/1500 [==============================] - 8s 5ms/step - loss: 0.0089 - accuracy: 0.9967 - val_loss: 0.1274 - val_accuracy: 0.9
Epoch 18/25
1500/1500 [==============================] - 7s 5ms/step - loss: 0.0113 - accuracy: 0.9962 - val_loss: 0.1505 - val_accuracy: 0.9
Epoch 19/25
1500/1500 [==============================] - 8s 5ms/step - loss: 0.0076 - accuracy: 0.9975 - val_loss: 0.1577 - val_accuracy: 0.9
Epoch 20/25
1500/1500 [==============================] - 7s 5ms/step - loss: 0.0117 - accuracy: 0.9963 - val_loss: 0.1678 - val_accuracy: 0.9
Epoch 21/25
1500/1500 [==============================] - 8s 6ms/step - loss: 0.0101 - accuracy: 0.9969 - val_loss: 0.1316 - val_accuracy: 0.9
Epoch 22/25
1500/1500 [==============================] - 8s 6ms/step - loss: 0.0069 - accuracy: 0.9976 - val_loss: 0.1737 - val_accuracy: 0.9
Epoch 23/25
1500/1500 [==============================] - 7s 5ms/step - loss: 0.0086 - accuracy: 0.9970 - val_loss: 0.1446 - val_accuracy: 0.9
Epoch 24/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0089 - accuracy: 0.9969 - val_loss: 0.1493 - val_accuracy: 0.9
Epoch 25/25
1500/1500 [==============================] - 8s 5ms/step - loss: 0.0080 - accuracy: 0.9977 - val_loss: 0.1628 - val_accuracy: 0.9
```

```
model.history.history
```


```
{'loss': [0.2886253893375397,
  0.1220148503780365,
  0.08339332789182663,
  0.06090816110372543,
  0.04796852543950081,
  0.03906631842255592,
  0.030415311455726624,
  0.024853497743606567,
  0.021625030785799026,
  0.020998619496822357,
  0.016412358731031418,
  0.014219075441360474,
  0.01519712433218956,
  0.013069160282611847,
  0.011504829861223698,
  0.010303890332579613,
  0.008884144015610218,
  0.011348400264978409,
  0.007577407639473677,
  0.011731380596756935,
  0.01009281724691391,
  0.0068639121018350124,
  0.008550478145480156,
  0.008936578407883644,
  0.007958943024277687],
 'accuracy': [0.9158333539962769,
  0.9629999995231628,
  0.9740208387374878,
```

```
 0.9810208082199097,
 0.984499990940094,
 0.9872083067893982,
 0.9899791479110718,
 0.9914374947547913,
 0.9926666617393494,
 0.9931041598320007,
 0.9946041703224182,
 0.9951249957084656,
 0.9947916865348816,
 0.9958333373069763,
 0.9960416555404663,
 0.9965416789054871,
 0.9967291951179504,
 0.9962499737739563,
 0.9974583387374878,
 0.9962916374206543,
 0.9968958497047424,
 0.9976249933242798,
 0.9970208406448364,
 0.996916651725769,
 0.9977291822433472],
'val_loss': [0.14286759495735168,
 0.11828195303678513,
 0.1018834337592125,
 0.10236238688230515,
 0.10178138315677643,
 0.09271147847175598,
 0.10202585160732269,
 0.10806580632925034,
```

```
pd.DataFrame(model.history.history)
```

	loss	accuracy	val_loss	val_accuracy
0	0.288625	0.915833	0.142868	0.958500
1	0.122015	0.963000	0.118282	0.964917
2	0.083393	0.974021	0.101883	0.969250
3	0.060908	0.981021	0.102362	0.968250
4	0.047969	0.984500	0.101781	0.972167
5	0.039066	0.987208	0.092711	0.974667
6	0.030415	0.989979	0.102026	0.971917
7	0.024853	0.991437	0.108066	0.973583
8	0.021625	0.992667	0.107830	0.972417
9	0.020999	0.993104	0.109449	0.973833
10	0.016412	0.994604	0.109429	0.975667
11	0.014219	0.995125	0.133597	0.973417
12	0.015197	0.994792	0.130484	0.974583
13	0.013069	0.995833	0.131063	0.975000
14	0.011505	0.996042	0.127218	0.976167
15	0.010304	0.996542	0.134238	0.974917
16	0.008884	0.996729	0.127353	0.976583
17	0.011348	0.996250	0.150463	0.973417
18	0.007577	0.997458	0.157736	0.973917
19	0.011731	0.996292	0.167780	0.973417
20	0.010093	0.996896	0.131564	0.978500
21	0.006864	0.997625	0.173691	0.974250
22	0.008550	0.997021	0.144562	0.977000
23	0.008937	0.996917	0.149342	0.976750
24	0.007959	0.997729	0.162829	0.975500

```
xtest[0]
```

```
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.01176471, 0.79607843, 0.99607843,
 0.85882353, 0.1372549 , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.14901961, 0.99607843, 0.99607843,
 0.30196078, 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.12156863, 0.87843137, 0.99607843, 0.45098039,
 0.00392157, 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.52156863, 0.99607843, 0.99607843, 0.20392157,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.23921569, 0.94901961, 0.99607843, 0.99607843, 0.20392157,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.4745098 , 0.99607843, 0.99607843, 0.85882353, 0.15686275,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.4745098 , 0.99607843, 0.81176471, 0.07058824, 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.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ]])
```

```
xtest[0].shape
```

```
(28, 28)
```

```
xtest[0].ndim
```

```
2
```

```
yprob = model.predict(xtest)
yprob
# the probability for each image would be 10 because the image hold 0 to 9
```

```
313/313 [==============================] - 1s 3ms/step
array([[1.0783306e-07, 2.1212143e-11, 3.8766963e-08, ..., 9.9999732e-01,
        8.7933437e-08, 3.0503209e-08],
       [5.6865137e-24, 1.1572257e-17, 9.9999994e-01, ..., 1.1835296e-18,
        8.1543816e-17, 8.3996040e-31],
       [2.3440474e-11, 9.9999982e-01, 5.5395940e-08, ..., 6.1733488e-08,
        4.3816531e-08, 1.5845830e-11],
       ...,
       [2.4120450e-28, 7.1463390e-19, 2.1645240e-32, ..., 4.8999993e-19,
        2.0910201e-22, 2.0278223e-15],
       [5.7358659e-29, 1.6002102e-22, 1.6383892e-37, ..., 5.6463800e-29,
        2.0238061e-13, 2.7782945e-26],
       [2.3409878e-14, 4.8417624e-21, 2.6034682e-22, ..., 3.8172229e-30,
        1.9340978e-23, 5.2929224e-22]], dtype=float32)
```

```
yprob[0]
```

```
array([1.0783306e-07, 2.1212143e-11, 3.8766963e-08, 2.3773171e-06,
       6.7309278e-22, 1.8931308e-12, 6.4220836e-22, 9.9999732e-01,
       8.7933437e-08, 3.0503209e-08], dtype=float32)
```

```
ypred = yprob.argmax(axis = 1)
ypred
```

```
array([7, 2, 1, ..., 4, 5, 6])
```

```
ypred[0]
```

```
7
```

```
from sklearn.metrics import classification_report
print(classification_report(ytest,ypred))
```

```
              precision    recall  f1-score   support

           0       0.98      0.99      0.98       980
           1       0.99      0.99      0.99      1135
           2       0.97      0.99      0.98      1032
           3       0.96      0.98      0.97      1010
           4       0.98      0.98      0.98       982
           5       0.98      0.97      0.97       892
           6       0.98      0.98      0.98       958
           7       0.98      0.97      0.97      1028
           8       0.97      0.97      0.97       974
           9       0.98      0.97      0.97      1009

    accuracy                           0.98     10000
   macro avg       0.98      0.98      0.98     10000
weighted avg       0.98      0.98      0.98     10000
```

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
```

```
[<matplotlib.lines.Line2D at 0x7f4c3931ee30>]
```