Project Devlopment Phase

Sprint 1

MNIST Dataset pre_processing

Date	31 October 2022
Team ID	PNT2022TMID29769
Project Name	A Noval Method For Handwritten Digit Recognition System
Maximum Marks	4 Marks

Understanding the Data

#importing the required libraries

```
import numpy as np
import tensorflow
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.layers import Conv2D
from keras.optimizers import Adam
from keras.utils import np_utils
```

#loading data

Input:

```
(X_train,y_train) , (X_test,y_test)=mnist.load_data()
```

Output:

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

Analyzing the data

Understanding the Data

Input:

```
X_train[0]
```

Output:

```
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,
148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,
  0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,
 253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,
0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,
253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,
0,   0],
[  0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,
195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,   0],
[  0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244, 133,
11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,   0],
[  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,   0,
0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,   0],
[  0,   0,   0,   0,   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)
```

Input:

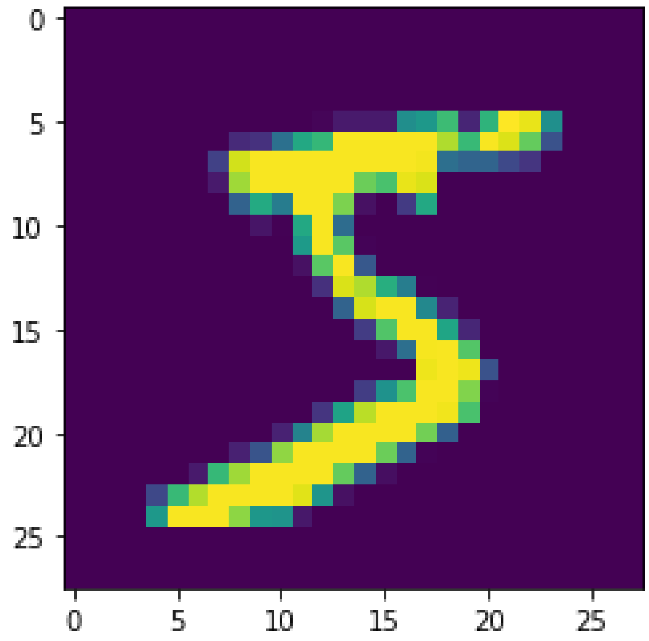
```
y_train[0]
```

Output:

```
5
```

Input:

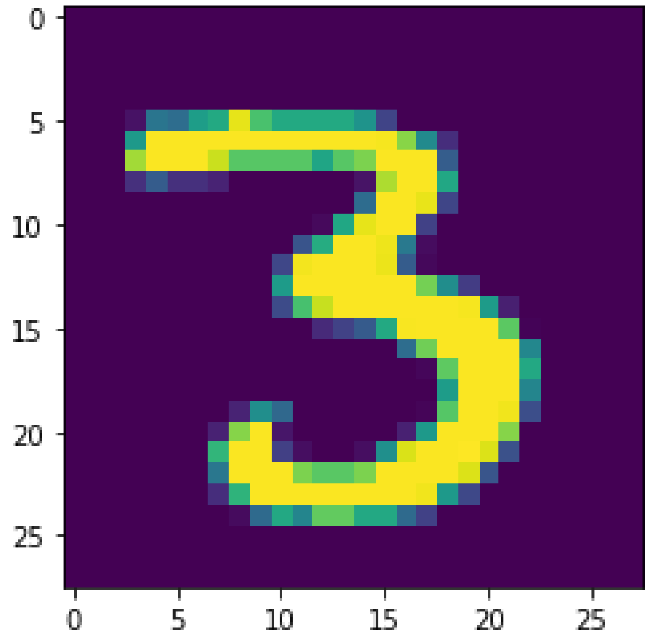
```
Import matplotlib.pyplot asplt
plt.imshow(X_train[0])
```



Input:

```
plt.imshow(X_train[12])
```

Output:



Reshaping the data

```
X_train=X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test=X_test.reshape(10000, 28, 28, 1).astype('float32')
```

Apply one-Hot Encoding

```
number_of_classes= 10
y_train=np_utils.to_categorical(y_train, number_of_classes)
y_test=np_utils.to_categorical(y_test,  number_of_classes)
```

Input:

```
y_train[0]
```

Output:

```
array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)
```