```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Dense, Flatten
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps
import numpy
```

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

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

```
print(X_train.shape)
print(X_test.shape)
```

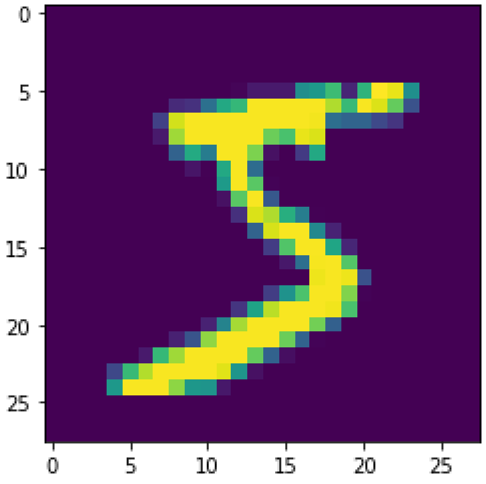
```
     (60000, 28, 28)
     (10000, 28, 28)
```

```
X_train[0]
```

```
y_train[0]
```

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

```
     <matplotlib.image.AxesImage at 0x7f8417b3b610>
```



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

```
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)
```

```
Y_train[0]
```

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

```
model = Sequential()
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation="relu"))
model.add(Conv2D(32, (3, 3), activation="relu"))
model.add(Flatten())
model.add(Dense(number_of_classes, activation="softmax"))
```

```
model.compile(loss='categorical_crossentropy', optimizer="Adam", metrics=["accuracy"])
```

```
model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test,Y_test))
```

```
     Epoch 1/5
     1875/1875 [==============================] - 184s 98ms/step - loss: 0.2590 - accuracy: 0
     Epoch 2/5
     1875/1875 [==============================] - 183s 98ms/step - loss: 0.0698 - accuracy: 0
     Epoch 3/5
     1875/1875 [==============================] - 182s 97ms/step - loss: 0.0485 - accuracy: 0
     Epoch 4/5
     1875/1875 [==============================] - 183s 98ms/step - loss: 0.0368 - accuracy: 0
     Epoch 5/5
     1875/1875 [==============================] - 182s 97ms/step - loss: 0.0296 - accuracy: 0
     <keras.callbacks.History at 0x7f8417cfda10>
```

```
metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)
```

```
     Metrics (Test Loss & Test Accuracy):
     [0.08822032809257507, 0.9757999777793884]
```

```
prediction = model.predict(X_test[:4])
print(prediction)
```

```
     1/1 [==============================] - 0s 99ms/step
     [[1.28290129e-10 1.12118045e-16 1.01993010e-06 1.30480888e-08
       9.70493717e-16 1.86878047e-13 6.70107648e-19 9.99998927e-01
       5.79429449e-10 4.57674076e-08]
      [1.75044264e-07 2.73469192e-10 9.99996662e-01 1.89776017e-09
       1.06790076e-12 1.99429283e-14 3.26794543e-06 8.39174379e-13
```

```
  1.74544734e-08 1.93928017e-13]
 [3.99311034e-07 9.96383667e-01 1.11743124e-04 2.49965755e-08
  9.09639402e-06 8.81441701e-06 1.99781243e-06 1.05764630e-05
  3.47366324e-03 1.45970069e-09]
 [9.99999881e-01 9.74408830e-15 5.95228045e-09 1.65517339e-12
  1.81564399e-13 6.26320176e-12 6.05326491e-08 4.98932076e-11
  2.95708311e-08 3.65778685e-10]]
```

```
print(numpy.argmax(prediction, axis=1))
print(Y_test[:4])
```

```
[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

```
model.save("model.h5")
```

```
model=load_model("model.h5")
```

```
from keras.datasets import mnist
from matplotlib import pyplot
(X_train,y_train),(X_test,y_test)=mnist.load_data()
print('X_train:' +str(X_train.shape))
print('y_train:' +str(y_train.shape))
print('X_test:' +str(X_test.shape))
print('y_test:' +str(y_test.shape))
from matplotlib import pyplot
for i in range(9):
  pyplot.subplot(330+1+i)
  pyplot.imshow(X_train[i],cmap=pyplot.get_cmap('gray'))
  pyplot.show()
```

```
X_train:(60000, 28, 28)
y_train:(60000,)
X_test:(10000, 28, 28)
y_test:(10000,)
```