```
import numpy as np
import pandas as pd
import random
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Flatten,
Conv2D,Dense,MaxPooling2D
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.datasets import mnist

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

print(X_train.shape)
X_train[0].min(),X_train[0].max()
X_train = (X_train - 0.0) / (255.0 - 0.0)
X_test = (X_test - 0.0) / (255.0 - 0.0)
X_train[0].min(), X_train[0].max()
(0.0, 1.0)

def plot_digit(image, digit, plt, i):
   plt.subplot(4, 5, i + 1)
   plt.imshow(image, cmap=plt.get_cmap('gray'))
   plt.title(f"Digit: {digit}")
   plt.xticks([])
   plt.yticks([])
plt.figure(figsize=(16, 10))
for i in range(20):
   plot_digit(X_train[i], y_train[i], plt, i)
plt.show()

X_tarin = X_train.reshape((X_train.shape+ (1,)))
X_test = X_test.reshape((X_test.shape+(1,)))

y_train[0:20]

model = Sequential([
   Conv2D(32,(3,3),
activation="relu",input_shape=(28,28,1)),
   MaxPooling2D((2,2)),
   Flatten(),
   Dense(100,activation="relu"),
   Dense(10,activation="softmax")
])

optimizer = SGD(learning_rate=0.01,
momentum=0.9)
model.compile(
   optimizer=optimizer,
   loss="sparse_categorical_crossentropy",
   metrics=["accuracy"]
)
model.summary()

Model_log = model.fit(X_train, y_train,
epochs=10, batch_size=15, verbose=1);


plt.figure(figsize=(16, 10))
for i in range(20):
   image = random.choice(X_test).squeeze()
   digit =
np.argmax(model.predict(image.reshape((1, 28,
28, 1)))[0], axis=-1)
   plot_digit(image, digit, plt, i)
plt.show()

predictions =
np.argmax(model.predict(X_test),axis=-1)
accuracy_score(y_test,predictions)

n = random.randint(0,9999)
plt.imshow(X_test[n])
plt.show()

predicted_value = model.predict(X_test)
print("Handwritten number in the image is = %d"
%np.argmax(predicted_value[n]))

score = model.evaluate(X_test,y_test,verbose=0)
print('Test loss:' , score[0])
print('Testaccuracy:',score[1])
```

Assignment3