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 / 255

X\_test = X\_test / 255

X\_train[0].min(), X\_train[0].max()

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\_train = X\_train.reshape((X\_train.shape ))

X\_test = X\_test.reshape((X\_test.shape ))

y\_train[0:20]

optimizer = SGD(learning\_rate=0.01, momentum=0.9)

model.compile(

optimizer=optimizer,

loss="sparse\_categorical\_crossentropy",

metrics=["accuracy"]

)

model.summary()

model.fit(X\_train, y\_train, epochs=10, batch\_size=32)

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()