import numpy as np

import keras

from tensorflow.keras. models import Sequential

from tensorflow.keras. layers import Dense,Dropout

from tensorflow.keras. optimizers import RMSprop

from tensorflow.keras .datasets import mnist

import matplotlib.pyplot as plt

from sklearn import metrics

from tensorflow.math import confusion\_matrix

(x\_train,y\_train),(x\_test,y\_test)=mnist.load\_data()

type(x\_train)

print(x\_train.shape,y\_train.shape,x\_test.shape,y\_test.shape)

print(x\_train[0])

print(x\_train[0].shape)

plt.imshow(x\_train[0])

plt.show()

print(np.unique(y\_train))

print(np.unique(y\_test))

x\_train=x\_train/255

x\_test=x\_test/255

print(x\_train[0])

model = keras.Sequential([

                          keras.layers.Flatten(input\_shape=(28,28)),

                          keras.layers.Dense(50, activation='relu'),

                          keras.layers.Dense(50, activation='relu'),

                          keras.layers.Dense(10, activation='sigmoid')

])

model.compile(optimizer='adam',

               loss='sparse\_categorical\_crossentropy',

              metrics=['accuracy'])

model.fit(x\_train,y\_train,epochs=10)

loss,accuracy=model.evaluate(x\_test,y\_test)

print(accuracy)

print(x\_test.shape)

plt.imshow(x\_test[0])

plt.show()

print(y\_test[0])

Y\_pred = model.predict(x\_test)

print(Y\_pred.shape)

print(Y\_pred[0])

label=np.argmax(Y\_pred[0])

print(label)

Y\_pred\_labels=[np.argmax(i) for i in Y\_pred]

print(Y\_pred\_labels)

conf\_mat=confusion\_matrix(y\_test,Y\_pred\_labels)

print(conf\_mat)