```
import numpy as np

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

(x_train, y_train), (x_test, y_test) = mnist.load_data()

plt.imshow(x_train[0], cmap='gray') // output 1 image

plt.show()

print(x_train[0]) // output 2  binary matrix

print("X_train shape", x_train.shape)

print("y_train shape", y_train.shape)

print("X_test shape", x_test.shape)

print("y_test shape", y_test.shape)   //output 3

x_train = x_train.reshape(60000, 784)

x_test = x_test.reshape(10000, 784)

x_train = x_train.astype('float32')

x_test = x_test.astype('float32')

x_train /= 255

x_test /= 255

num_classes = 10

y_train = np.eye(num_classes)[y_train]

y_test = np.eye(num_classes)[y_test]

model = Sequential()

model.add(Dense(512, activation='relu', input_shape=(784,)))

model.add(Dropout(0.2))

model.add(Dense(512, activation='relu'))

model.add(Dropout(0.2))

model.add(Dense(num_classes, activation='softmax'))

model.compile(loss='categorical_crossentropy',

          optimizer=RMSprop(),

          metrics=['accuracy'])

batch_size = 128

epochs = 20

history = model.fit(x_train, y_train,

               batch_size=batch_size,

               epochs=epochs,  verbose=1,

               validation_data=(x_test, y_test))

score = model.evaluate(x_test, y_test, verbose=0)

print('Test loss:', score[0])

print('Test accuracy:', score[1]) // output4
```