KERAS MNIST

E0320008
DHRISH

```
from keras. datasets import mnist
 (x_train, y_train), (x_test, y_test) =
                    mnist. load_data()

import matplotlib. pyplot as plt
x_train. shape
single_image = x_train [0]
plt. imshow (single_image)

from keras. util np_utils import to_
 categorical
y_train. shape
y_example = to_categorical (y_train)
y_example. shape

y_cat_test = to_categorical (y_test, 10)
y_cat_train = to_categorical (y_train,
                              10)

x_train = x_train/255
x_test = x_test/255

scaled_single = x_train[0]

# Now Reshaping :-

x_train = x_train. reshape (60000,
                    28, 28, 1)
```

```
x_test = x_test. reshape (10000,
                 28, 28, 1)


# Training

from keras. models import
      Sequential

from keras. layers import Dense,
 Conv2D, MaxPool20, Flatten.


model = sequential()
model. add( Conv2D( filters = 32,
kernel_size = (4,4), input_shape = (28,28,1),
activation = 'relu',))
model. add( MaxPool2D ( pool_size = (2,2)))
model. add (Flatten ())
model. add ( Dense (128, activation - 'softmax')

model. compile ( loss = 'categorical_
crossentropy', optimizes = 'rmsprop',
metrics = ['accuracy'])

model. fit (x_train, y_cat_train,
 epochs = 2)

# EVALUATION
model. evaluate (x_test, y_cat_test)

from sklearn. metrics import classification_
 report
```

```
predictions = model.predict_classes (x_test)
 y_cat_test.shape

predictions [0]

print ( classification_report (y_test,
 predictions))
```