ROLL NO M22AI612
NAME —RISHABH SACHAN

Question number 2——

This code uses a simple neural network to classify images of handwritten digits. The dataset consists of images of size 32x32 pixels, with each image representing a single digit from 0 to 9. The data has been split into a training set and a test set. The training set is used to train the neural network, while the test set is used to evaluate its performance.

The code loads the data from two folders: 'train' and 'val'. Each folder contains subfolders corresponding to each digit. The code reads in the images, resizes them to 32x32 pixels, and saves them as NumPy arrays. It then flattens the arrays, scales the pixel values to between 0 and 1, and uses them to train a simple neural network with a single dense layer of 10 neurons with the sigmoid activation function.

The neural network is then compiled and trained using the training set for 10 epochs. The test set is then used to evaluate the performance of the model. The code then creates a new neural network with two dense layers, the first of which has 1024 neurons with the ReLU activation function and the second has 10 neurons with the softmax activation function. The model is then trained for 10 epochs and evaluated on the test set.

The final evaluation of the model shows that it has an accuracy of approximately 91% on the test set. The confusion matrix can be used to analyze the performance of the model on individual digits.

Question number 3 —

The code is a Python script that trains and tests a convolutional neural network (CNN) model to classify different types of charts based on their

images. The code uses TensorFlow and Keras libraries for building the CNN model.
The data for training and testing the model is provided in the form of images saved in PNG format and their corresponding labels saved in a CSV file.

The first part of the code loads the images and their labels from the provided folders and CSV file, respectively. The images are resized to 128x128 pixels and converted to RGB color format. Then, the string labels are converted to numerical labels using the LabelEncoder from scikit-learn library. The converted images and labels are saved in NumPy format for later use.

After loading and processing the data, the code checks the shape and samples of the images to make sure the data was loaded correctly. It also defines the categories of charts as 'line', 'dot_line', 'hbar_categorical', 'vbar_categorical', and 'pie', and maps them to numerical labels.

the code normalizes the image data by dividing the pixel values by 255, to scale the pixel values between 0 and 1. Then, it defines a simple neural network architecture consisting of three dense layers with ReLU activation and a final dense layer with softmax activation for multiclass classification. The model is compiled with the stochastic gradient descent optimizer and sparse categorical cross-entropy loss. Finally, the model is trained on the training data for 10 epochs.

The code also splits the training data into training and validation sets using the train_test_split function from scikit-learn library.
Finally, the trained model is tested on the test data and evaluated using the confusion matrix and classification report from scikit-learn library.
Overall, the code successfully trains and tests a CNN model for chart classification based on their images.
However, the model can be improved by increasing the number of epochs, using more advanced CNN architectures, and optimizing the hyperparameters.