Start with large image of dataset of size 5000, and split into 5000 images

For 10 digits

Use first 250 as training data, other 250 as test data

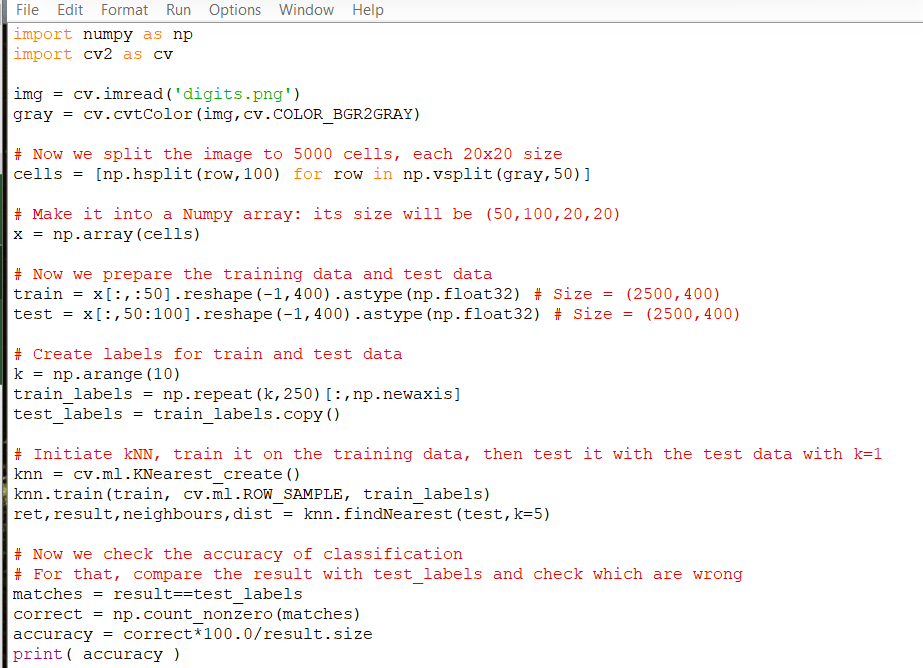


Image is split into 5000 cells, loaded into a numpy array, training data is first 50, test data is the last 50

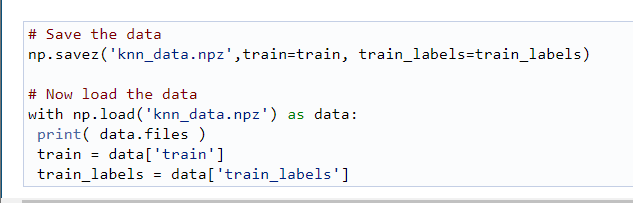
Labels are created for training and test data

Create knn object and train it on the training data

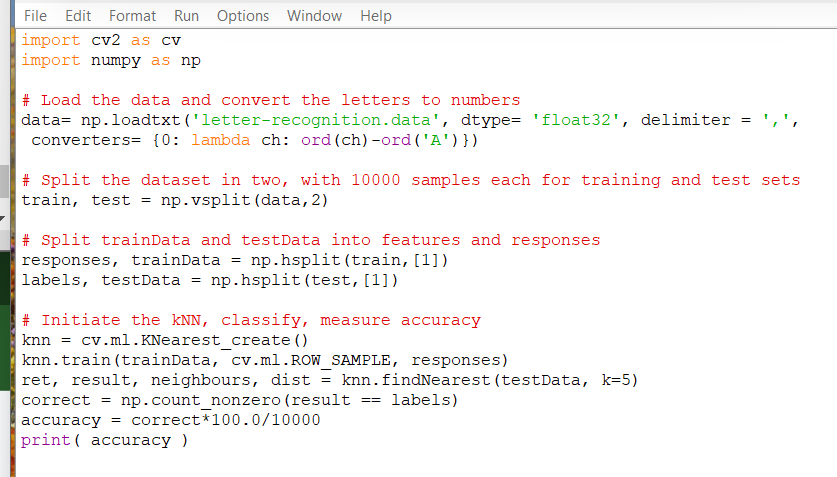
Test with k = 5, then use premade labels to test accuracy



91.76% accuracy



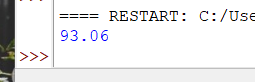
Numpy has functions np.savetxt np.savez and np.load to save and load data



Code using np.loadtxt from letter recognition data file

Trained using first half of data, tested using last half

Received 93.06 as accuracy



Using k = 20 I got 90.18 as accuracy

Using k = 3 i got 93.61 as accuracy

Using k = 4 i got 93.04 as accuracy

Using k = 2 i got 92.7 as accuracy

Accuracy seems to peak at k = 3