A classic problem in the field of pattern recognition is that of **handwritten digit recognition**. Suppose that you have images of handwritten digits ranging from 0-9 written by various people in boxes of a specific size - similar to the application forms in banks and universities.

The goal is to develop a model that can correctly identify the digit (between 0-9) written in an image.

**Objective**

You are required to develop a model using Support Vector Machine which should correctly classify the **handwritten digits from 0-9** based on the**pixel values given as features**. Thus, this is a 10-class classification problem.

**Data Description**

For this problem, we use the **MNIST data** which is a large database of handwritten digits. The 'pixel values' of each digit (image) comprise the features, and the actual number between 0-9 is the label.

Since each image is of 28 x 28 pixels, and each pixel forms a feature, there are 784 features. MNIST digit recognition is a well-studied problem in the ML community, and people have trained numerous models (Neural Networks, SVMs, boosted trees etc.) achieving error rates as low as 0.23% (i.e. accuracy = 99.77%, with a convolutional neural network).

Before the popularity of neural networks, though, models such as SVMs and boosted trees were the state-of-the-art in such problems.

[This webpage (from the original contributors of the dataset)](http://yann.lecun.com/exdb/mnist/) tabulates the accuracies achieved by various classification models.

In this assignment, try experimenting with various hyperparameters in SVMs and observe the highest accuracy you can get. With a sub-sample of 10-20% of the training data (see note below), you should expect to get an accuracy of more than 90%.

**Important Note:**

Since the training dataset is quite large (42,000 labelled images), it would take a lot of time for training an SVM on the full MNIST data, so you can sub-sample the data for training (10-20% of the data should be enough to achieve decent accuracy). Also, running a GridSearchCV() may take hours if you use a large value of k (fold-CV) such as 10 and a wide range of hyperparameters; k = 5 should be sufficient. Also, check the class-wise accuracy so as to make sure the results aren't skewed.