# Handwritten Digit Recognizer

This was done as part of a tutorial competition on Kaggle.com.
. http://www.kaggle.com/c/digit-recognizer

A set of handwritten single digit images were given by Kaggle. (MNIST dataset)

The code was developed in python.

I used numpy library (for Python) for efficiently handling the data. It provides basic mathematical operations like sum, power and subtraction along with easily handling large d-dimensional vector representations of data.

At first basic implementation of K-Nearest Neighbours algorithm was tried with a brute force approach. Later different approaches to finding the optimal k were tried and decided upon the k.

Later it was optimized using a k-dimensional tree approach. This involved use of Scikit-learn.

A k-d tree is basically a data structure which allows us to do space partitioning in an optimal way. It is particularly great for nearest neighbour searches. This search can be done efficiently by using the tree properties to quickly eliminate large portions of the search space.

The idea behind it is using the tree to navigate through space partitions while decreasing the size of each partition as you go through the tree. So by navigating through the tree we get a good approximation of the nearest points to the point you're searching for. But we cant end here as the point we are looking for may be at the edge of the partition. So we will need to re-check for the real nearest neighbour.