**MNIST Digit Recognition**

A **Euclidean Minimum Distance or Manhattan distance** criterion is used to find minimum distances and k-nearest neighbor classifier is used to classify the digits. A MNIST database is used for both training and testing the system

KNN APPROACH

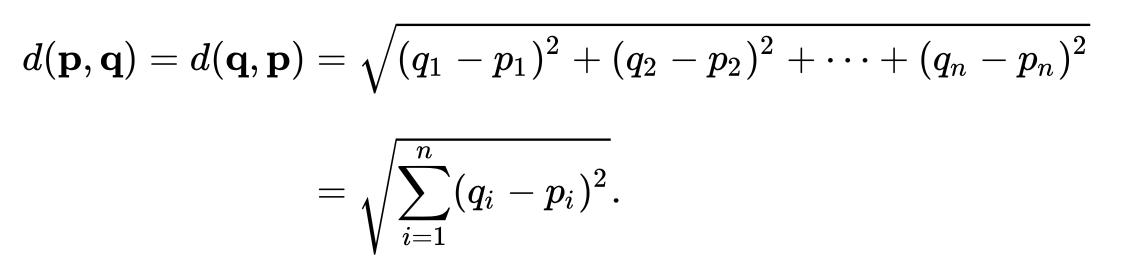
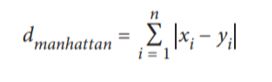
K-Nearest Neighbors (**KNN**) is one of the simplest algorithms used in **Machine Learning** for regression and classification problems. **KNN** algorithms use data and classify new data points based on similarity measures.

**Algorithm**

**Step 1 – Structuring our initial dataset**

**Step 2 – Splitting the dataset:** We’ll split dataset into training and training example set

**Step 3 – Comparisons and Calculations:** We compare between the images in training and testing data by calculating the distances for various values of K.

* We use Euclidean distance to calculate the distances between the pixels using the following formula.
* For manhattan distance it is:
* 
* The equation describes the Euclidean distance between two n-dimensional points p and q. In our case n = 784 because each image in MNIST is 784 dimensional

**Step 4 – Training our classification model:** After finding the distances ,sort and extract only k nearest label.

* Our k-NN classifier will be trained on the raw pixel intensities of the images in the training set.
* We’ll then store and return the majority votes label and determine the best value of *k* using the validation set.

**Step 5 – Evaluating our classifier:** We predict and print the result which has the maximum accuracy.

This is a time consuming process because we compare a single image with thousands of images in the training set and every time when we want to predict the written value in the test image.We can reduce the number of features by using methods like **calculating covariance** between images or using **eigenvectors** which will reduce the time.