**Machine learning model evaluation using the EMNIST dataset**

# Introduction

The project explores the efficiency and accuracy of my own implementations of K-Nearest Neighbor using two different distance metrics: Euclidian Distance and Manhattan (L1) Distance metrics. This project aims to find the most effective approach of classifying handwritten letters on the Extended Modified National Institute of Standards and Technology (EMNIST) dataset, which comprises of 26,000 images of handwritten letters and is used to train machine learning algorithms to recognise handwritten characters and words.

# Data and Preparation

The EMNIST dataset contains images of resolution 28 x 28. The data was prepared by randomly splitting it into a training and a testing subset, each containing half of the images. The data was stored as a 1 x 784 vector, which my code had to turn back into a 28 x 28 image using the “reshape” command. I used the “randperm” function to divide the data which ensured that the split was random, and running the code multiple times would yield different, although similar, results every time.

*Figure 1: sample data*



# Methodology

I chose the Manhattan model to compare with the Euclidean model to see how the results would compare for these two algorithms which are quite similar in theory; the primary difference between these metrics being that Euclidean distance is the direct path while Manhattan distance moves only in right angles to its destination. It is expected that Euclidean will outperform Manhattan distance for this reason. I chose to compare my KNN implementation with the built in KNN function to see if my solution was perhaps more efficient in the measures of time or accuracy. The ensemble model combines approaches a problem with uses multiple models. I chose it to see if this more diverse approach is effective. I evaluated time using tic and toc commands and I evaluated accuracy by using the formula shown below.

# Results

|  |  |  |
| --- | --- | --- |
|  | **Training time** | **Accuracy** |
| **Euclidean** | 464.7 seconds | 78.30% |
| **Manhattan** | 464.1 seconds | 76.08% |
| **fitcknn()** | 84.3 seconds | 77.62% |
| **fitcensemble()** | 118.9 seconds | 46.99% |

# Conclusion

The accuracy of the four models were not vastly different from one another, with the exception of the Ensembles algorithm. It is clear that it is the wrong approach to this EMNIST dataset. Now, looking at the remaining three options, we can see that the training time for Euclidean and Manhattan are almost identical, and that the fitcknn() time is much lower. The Euclidean accuracy is marginally higher than fitcknn(), but the drastically lower training time of fitcknn(), it is certainly the recommended option to be used with the EMNIST dataset and others like it.