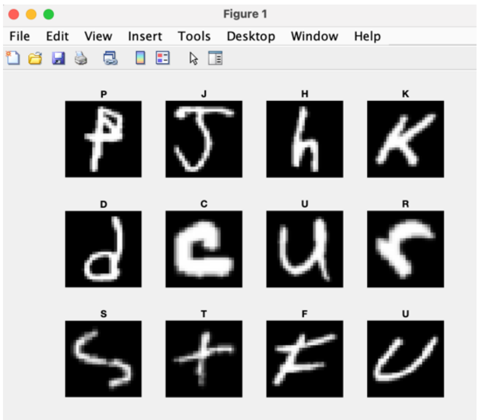
Testing 4 different AI models

Introduction

The problem we are solving is to figure out which model for machine learning is the best. This is important for potential users as they can figure out which model, they want to use based on these tests. We are aiming to test 4 different models for machine learning on a dataset of images and labels and time each model. Then we will compare the accuracy and time taken for each model against each other and see which model is the best.

Data and Preparation

The dataset comprises 26000 images depicting handwritten letters, each paired with corresponding labels indicating the specific letter they represent. Every image is a 28 by 28-pixel image which has been stored as a reshaped 1 x 784 vector. To prepare the data firstly I converted the images in the dataset into type double. Then to prepare the images to be shown like Figure 1 I used randi(26000) to give me a random number and then selected that image. I reshaped the image to a 28 x 28 vector instead of a 1 x728 vector. To randomly split the data in half for the testing and training data, I used randperm(26000) which creates 1 x 26000 vector with random non repeating numbers from 1 to 26000. The training data was allocated to the first half of 1 x 26000 vector, and the test data to the second half.



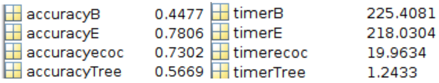
Methodology

The overall method for this problem is to train 4 models of machine learning on a set of training data. Then test the models against a set of test data and record each model's accuracy and time taken. The evaluation methods used are Euclidean Distance, Bray Curtis, fitcecoc and fitctree. I chose the Bray Curtis method as I expected it to have a low accuracy and it would be interesting to see how well it would do. I chose the fitcecoc method as it is designed for multi class problems, and I wanted to see how well it would do in a binary problem. Lastly, I chose the fitctree method as I wanted to test how well a decision tree would do as I expected it to have a low accuracy.

The Euclidean Distance model for knn works by identifying the k training data points with the shortest Euclidean distance to the test point. Then it predicts the class label based on the majority class among the k nearest neighbours. The Bray-Curtis method does the exact same but instead of identifying the shortest Euclidean distance it identifies the lowest Bray-Curtis dissimilarity values. The fitcecoc method uses a one-vs-all (OvA) strategy to train multiple binary classifiers, one for each class. These binary classifiers are combined to make multiclass predictions. This function takes two values which are the training features and training data. The fitctree method uses a decision tree model which uses the features provided during training to categorize the testing data into predetermined classes, relying on the decision rules acquired through the training process.This function takes two values which are the training features and training data.

Results

The Bray Curtis method it took 225.4081 seconds with an accuracy of 44.77%. Euclidian Distance took 218.0304 seconds with an accuracy of 78.06%. Fitcecoc took seconds 19.9634 and had an accuracy of 73.02%. Lastly fitctree took 1.2433 seconds and had an accuracy of 56.69%. Overall fitctree finished in the quickest time with 1.2433 seconds which is roughly 16 times quicker than the method that finished second. However, the Euclidian Distance method had the highest accuracy with 78.06% which is roughly 5% bigger than the second highest accuracy method (fitcecoc).



accuracyB	0.4477	timerB	225.4081
accuracyE	0.7806	timerE	218.0304
accuracyecoc	0.7302	timerecoc	19.9634
accuracyTree	0.5669	timerTree	1.2433

Conclusion

In conclusion I would recommend using the fitcecoc model. It has a balanced accuracy and time completion and has the second fastest time completion and accuracy out of the 4 models tested. However, the problems with fitcecoc are that even though it has the second highest accuracy the accuracy is still quite low. Some further research I could do is to test more methods like fitcensemble() or the co-sine distance between points as they might have a better performance than the 4 methods tested.