**Letter image Recognition**

In this assignment are given a large number of black-and-white rectangular pixel displays. I will use neural networks (MLP) and KNN in order to predict which of the 26 capital letters in the English alphabet each such display represents.

**Source information**

Creator: David J. Slate Odesta Corporation; 1890 Maple Ave; Suite 115; Evanston, IL 60201  
Donor: David J. Slate ([dave@math.nwu.edu](mailto:dave@math.nwu.edu)) (708) 491-3867  
Date: January, 1991

**Relevant information**

The character images is based on 20 different fonts and each letter within these 20 fonts has been randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15.

We typically train on the first 16000 items and then use the resulting model to predict the letter category for the remaining 4000 (**NOT in this assignment**). See the article cited below for more details:

**Approach**

In this assignment I will first conduct a simple EDA follow with MLP and KNN. The test results will then compare with the same datasets but with standardised values to see the comparison.

Notes: For full explanation and coding methods please have a look at the jupyter notebook for more details.

**Exploratory data Analysis(EDA)**

Splitting data for training and testing purpose:

* Use the following ratio for splitting the data:

Training set 80%: (16000, 17)

Validation set 10%: (2000,17)

Test set 10%: (2000, 17)

**Data Balance**: Print the percentage of class label letter (A-E) in each partition, and check that they are similar.

* The average of 26 letters distribution in 20000 possibilities is 3.85%

|  |  |  |  |
| --- | --- | --- | --- |
|  | Train Set | Validation Set | Test Set |
| A | 3.94% | 3.75% | 4.2% |
| B | 3.75% | 4.5% | 3.75% |
| C | 3.61% | 3.81% | 4.15% |
| D | 4.10 % | 3.45 % | 4.2% |
| E | 3.88 % | 4.2% | 3.15% |

Results:

* Looking at the percentage of each class from different dataset, the differences are at most + - 0.5% between different dataset
* Within the datasets each label is +- 0.5% from the average of 3.85%
* Therefore, we can safe to assume every label is evenly distributed in Train set, Validation set and Test set
* The datasets are balanced

Scaled Data set:

Standardisation: Standardizing the features around the centre and 0 with a standard deviation of 1

**Multilayer Perceptron (MLP) classifier**

Parameters:

* Random\_state = 0
* Max\_iter = 300

Unstandardized Train set:

* Test set accuracy: 0.9115
* Training set accuracy: 0.9486875

Standardized Train set:

* Test set accuracy: 0.949
* Training set accuracy: 0.9944375

**K-Nearest Neighbour (KNN)**

Train KNN models with different k values (1-10), and then report the best accuracy and its k value on unscaled training/test and scaled training/test data, respectively

Unstandardized Train set:

|  |  |  |
| --- | --- | --- |
| K | Train Accuracy (%) | Test Accuracy (%) |
| 1 | 1 | 0.956 |
| 2 | 0.975 | 0.949 |
| 3 | 0.981 | 0.957 |
| 4 | 0.975 | 0.96 |
| 5 | 0.979 | 0.955 |
| 6 | 0.971 | 0.953 |
| 7 | 0.971 | 0.95 |
| 8 | 0.968 | 0.948 |
| 9 | 0.965 | 0.943 |
| 10 | 0.963 | 0.9425 |

* Highest Accuracy:
  + Training set: when k = 1 Accuracy is 100%
  + Test set: when k = 4 Accuracy is 96%

Standardized Train set:

|  |  |  |
| --- | --- | --- |
| K | Train Accuracy (%) | Test Accuracy (%) |
| 1 | 1 | 0.956 |
| 2 | 0.976 | 0.947 |
| 3 | 0.977 | 0.951 |
| 4 | 0.971 | 0.9485 |
| 5 | 0.971 | 0.948 |
| 6 | 0.967 | 0.945 |
| 7 | 0.965 | 0.942 |
| 8 | 0.961 | 0.938 |
| 9 | 0.959 | 0.9365 |
| 10 | 0.957 | 0.933 |

* Highest Accuracy:
  + Scaled Training set: when k = 1 Accuracy is 100%
  + Scaled Test set: when k = 1 Accuracy is 95%

**Analysis and Evaluation of Results**

* From the table above we can see both training and Test accuracy are increasing as the number of Training set increases
* Therefore, this indicate the data is not too little or too big because there is a difference
* However, in my opinion, we have too much data because the increase in accuracy are small
* From the table above we can see test accuracy increase significantly from 10% to 30% about 10% accuracy, but from 30% to 90% only increase total of 6% accuracy
* In this data analysis is not a critical issue to some false prediction, so 86% accuracy in this scenario is a pretty good result
* In addition, the cost and time of collect and process extra data are not equivalent to the results, therefore in my opinion we should only have 30% of the training data