**ALPHANUMERIC RECOGNITION**

**PROJECT FOR LEARNING PURPOSE**

**SUBMITTED BY: PRATYUSH SINGH**

**SUBMITTED TO: MISS MALLIKA SRIVASTAVA**

**BATCH: 21DAT-063**

**CONTENT TABLE:**

|  |  |  |
| --- | --- | --- |
| S.NO. | TOPICS | PAGE |
| 1 | ABSTRACT | **3** |
| 2 | SUMMARY | **3** |
| 3 | OBJECTIVES | **4** |
| 4 | DETAILS OF PROCESS | **5 to 12** |
| 4.1 | **IMPORTING DATASET** | **5** |
| 4.2 | **DATA PRE-PROCESSING** | **7** |
| 4.3 | **INTERPRETATION OF DATA** | **8** |
| 4.4 | **BUILDING AND TRAINING MODEL:** | **9** |
| 4.5 | **TESTING AND EVALUATION** | **10** |
| 5 | REQUIREMENTS | **12** |
| 6 | CODE LINK | **12** |

**IMAGE AND TABLE CONTENT:**

**Figure1: (page 5 )**

**Figure2: (page 7 )**

**Figure3: (page 8)**

**Figure4: (page9 )**

**Figure5: (page 10)**

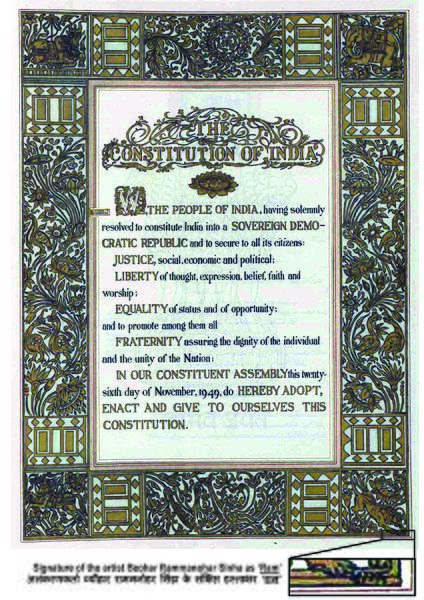
**Figure6: (page 11 )**

**Table1: (page 11)**

**ABSTRACT:** This project is small ML, which in intended to help others(on opensource ) to get the basic taste of ML related projects.

**PROJECT SUMMARY:** This project is based on Alphanumeric recognition capability using EMNIST byclass DATASET using Sklearn Logistic regression model using saga solver optimisation.

**OBJECTIVES:**

****

* Idea of making this project in my mind come, because I feel it is very difficult in ML, to get proper data and logics for preparing ML model.
* So to help people in opensource I decided to work on this small project, so that they can get a help while exploring the basics of the ML courses
* I have decided to upload the project in opensource(Github) that others can take help in understanding basics of python ML.

**Details of Process:**

1. **IMPORTING DATASET**: The Dataset used for this project is EMNIST byclass dataset, i.e these 4 give files:

For training purpose

* emnist-byclass-train-images-idx3-ubyte
* emnist-byclass-train-labels-idx1-ubyte
* emnist-byclass-test-images-idx3-ubyte

For testing purpose

* emnist-byclass-test-labels-idx1-ubyte

For reading these datasets, I used python-mnist package as shown in figure:

**FIGURE 1 IMPORTINF DATA**

Here, I saved stored all my 4 datasets in a folder and named it randomly with name ‘Pratyush\_Data’.

And stored the information in X\_train, y\_train, X\_test, y\_test respectively.

The numerical encoding of EMNIST dataset is as follows(here I am informing about the meaning of values written in y\_train and y\_test):

|  |  |
| --- | --- |
| NUMERICAL ENCODING | RESPECTIVE DIGITS/ALPHABETS |
| 0  1  2  3  .  .  9  10  11  12  13  .  .  35  36  37  38  39  .  .  60  61 | 0  1  2  3  .  .  9  A  B  C  D  .  .  Z  a  b  c  d  .  .  y  z |

**TABLE-1:** EXPAINING EMNIST BYCLASS DATASET ENCODING

1. **DATA PRE-PROCESSING (OPTIMISING DATA FOR MODEL):** The pre-processing required following steps:
2. Converting X\_train and X\_test (i.e images) to NumPy arrays.

1. Reshaping all images into 28\*28 matrix.
2. Taking Transpose of 2-D array for reversing and rotating all train and test images to proper orientation.

**FIGURE 2: DATA PRE-PROCESSING**

1. **INTERPRETATION OF DATA:** Now using matplotlib and NumPy ,we will check whether out data is properly pre-processed or not. We will do this as following:
2. Using matplotlib.pyplot.matshow, we will check our X\_train dataset
3. And using NumPy we will check our y\_train dataset.

**FIGURE 3: INTERPRETATION OF DATA**

And from the code-snippet (provided in the link) we can clearly see that out training data set is complete fine.

But we know that logistic regression take a 2-D array as its first parameter (that is our X\_train) and 1-D array as its second parameter(that is our y\_train).

But our X\_train is at present group of images that is a 3-D array. So first we have to convert 2-D image matrices to !-D array as shown:

**FIGURE 4: 2-D IMAGE MATRIX TO 1-D ARRAY CONVERSION**

Now our X\_train and Y\_train are perfectly ready for training model.

1. **BUILDING AND TRAINING MODEL:**

Here for defining data model, I have used Logistic regression. We know that Logistic Regression is the Supervised Learning Algorithm for solving classification problems. Hence can be used in this problem.  
  
In backend this regression solver has many parameters that train the model like Regularization parameter, hypothesis function, theta - fitting parameter etc.

**FIGURE 5: BUILDING AND TRAINING MODEL**

Even X\_train contains 697932 images(1-D arrays for model) but I used first 30000 images, because training with all the images was taking a lot of time.

Here solver is Algorithm used in the optimization problem, And for multiclass large data classification ‘saga’ is preferred.

1. **TESTING AND EVALUATION:**

**FIGURE 6: TESTING AND EVALUATION**

Here, I have used a list of alphanumeric to convert numbers of predictions and y\_test to their respective digits/alphabets.

And we can from both ‘Predictions’ and ‘Actual results’ that, model has done fairly nice. But It sometimes confuses between:

* O(capital o) ,o(small o),0(zero)
* G, C, c
* U, 0, O, u ,etc

It has a score of approximately 70% on test data, which is fairly nice.

**REQUIREMENTS:**

* Jupyter notebook or collab (PYTHON 3.8+ KERNEL)
* NumPy, matplotlib, mnist, sklearn library packages of python
* EMNIST byclass Dataset.

**MY CODE AND DATASET FOLDER LINK:**

[**https://drive.google.com/drive/folders/1e8Fo6lbo7gtZBo5FWwr1XDjluCBjVHWJ?usp=sharing**](https://drive.google.com/drive/folders/1e8Fo6lbo7gtZBo5FWwr1XDjluCBjVHWJ?usp=sharing)