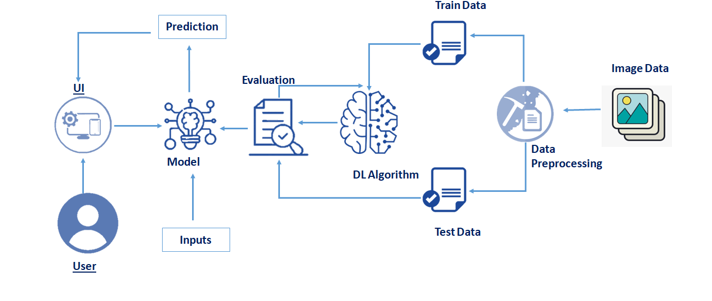
**A Novel Method For Handwritten Digit Recognition System**

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in the field of science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real-time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. this image is analyzed by the model and the detected result is returned on to UI

**Technical Architecture:**



**Project Objectives**

**By the end of this project you will:**

* Know fundamental concepts and techniques of the Artificial Neural Network and Convolution Neural Networks
* Gain a broad understanding of image data.
* Work with Sequential type of modeling
* Work with Keras capabilities
* Work with image processing techniques
* know how to build a web application using the Flask framework.

**Project Flow**

**Project Flow:**

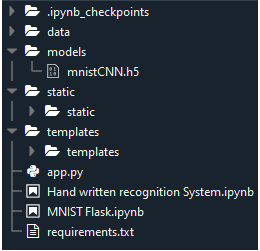
* The user interacts with the UI (User Interface) to upload the image as input
* The uploaded image is analyzed by the model which is integrated
* Once the model analyses the uploaded image, the prediction is showcased on the UI

**To accomplish this, we have to complete all the activities and tasks listed below**

* Understanding the data.
  + Importing the required libraries
  + Loading the data
  + Analyzing the data
  + Reshaping the data.
  + Applying One Hot Encoding
* Model Building
  + Creating the model and adding the input, hidden and output layers to it
  + Compiling the model
  + Training the model
  + Predicting the result
  + Testing the model by taking image inputs
  + Saving the model
* Application Building
  + Create an HTML file
  + Build Python Code

**Project Structure**

Create a Project folder which contains files as shown below



* We are building a Flask Application which needs  HTML pages stored in the templates folder and a python script app.py for server side scripting.
* The model is built in the notebook Hand written recognition system.ipynb
* We need the model which is saved and the saved model in this content is mnistCNN.h5
* The static folder will contain js and css files.
* The templates mainly used here are main.html and index6.html for showcasing the UI

**Prerequisites**

**To complete this project you should have the following software  and packages**

**Anaconda Navigator :**

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS. Conda is an open-source, cross-platform,  package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupiter notebook and spyder.

**Understanding The Data**

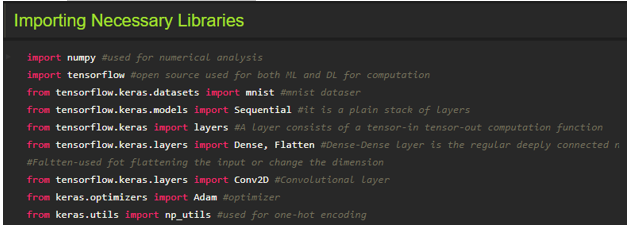
ML depends heavily on data, without data, it is impossible for a machine to learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions. TensorFlow already has MNist Data set so there is no need to explicitly download or create Dataset

The MNSIT dataset contains ten classes: Digits from 0-9. Each digit is taken as a class

In this activity, let's load the data and understand the features of the data.

**Importing The Required Libraries**

Lets first import the libraries



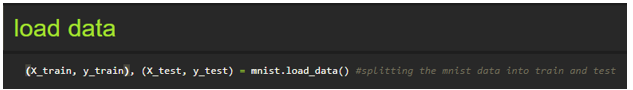
Importing the required libraries which are required for the model to run. The dataset for this model is imported from the Keras module.

The dataset contains ten classes: Digits from 0-9. Each digit is taken as a class

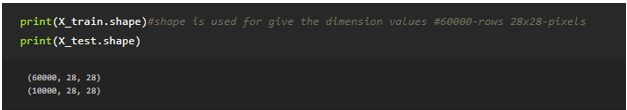
For a detail point of view on Keras and TensorFlow refer to the**link**[**here**](https://www.upgrad.com/blog/the-whats-what-of-keras-and-tensorflow/)**:**

**Loading The Data**

The dataset for this model is imported from the Keras module.



We split the data into train and test. Using the training dataset we train the model and the testing dataset is used to predict the results.

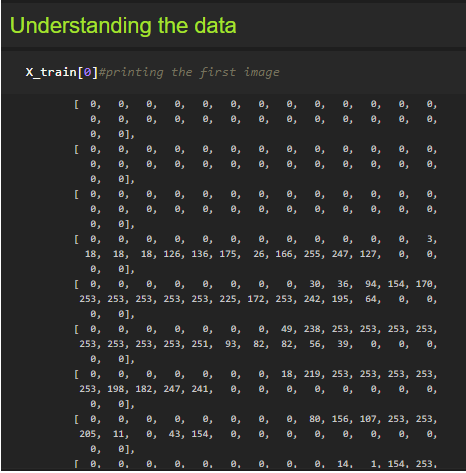


We are finding out the shape of X\_train and x\_test for better understanding. It lists out the dimensions of the data present in it.

in trainset, we have 60000 images, and in the test  set we have 10000 images

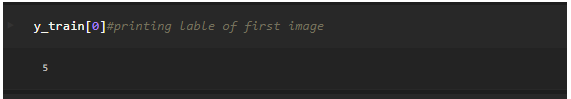
**Analyzing The Data**

Let's see the  Information of an image lying inside the x\_train variable



Basically, the pixel values range from 0-255. Here we are printing the first image pixel value which is index[0] of the training data. As you see it is displayed in the output.

With respect to this image, the  label of this image will e stored in y\_train let's see what is the label of this image by grabbing it from the y\_train variable



As we saw in the previous screenshot, we get to know that the pixel values are printed. Now here we are finding to which image the pixel values belong to. From the output displayed we get to know that the image is ‘5’.

Lets Plot the image on a graph using the Matplot library

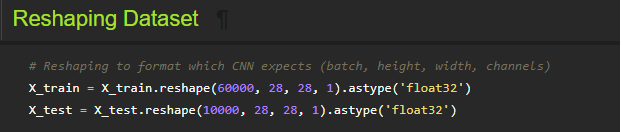


Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. By using the Matplotlib library we are displaying the number ‘5’ in the form of an image for proper understanding.

**Note:** You can see the results by replacing the index number till 59999 as the train set has 60K images

**Reshaping The Data**

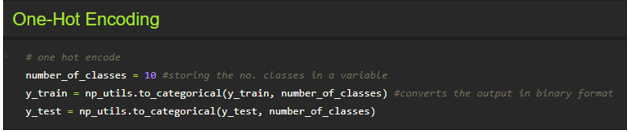
As we are using Deep learning neural network, the input for this network to get trained on should be of higher dimensional. Our dataset is having three-dimensional images so we have to reshape them too higher dimensions



We are reshaping the dataset because we are building the model using CNN. As CNN needs four attributes batch, height, width, and channels we reshape the data.

### Applying One Hot Encoding

If you see our y\_train variable contains Labels representing the images containing in x\_train. AS these are numbers usually they can be considered as numerical or continuous data, but with respect to this project these Numbers are representing a set of class so these are to be represented as categorical data, and we need to binaries these categorical data  that's why we are  applying One Hot encoding for y\_train set



One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. We apply One-Hot Encoding in order to convert the values into 0’s and 1’s. For a detailed point of view, look at this  [link](https://deeplizard.com/learn/video/v_4KWmkwmsU)

Now let's see how   our label 5 is index 0 of y\_train is converted



As we see the new the label is printed in the form of 0’s and 1’s and is of type float.