HANDWRITTEN DIGIT RECOGNITION USING MACHINE LEARNING

A thesis submitted in partial fulfillment of the
requirement for the award of the degree of

Bachelor of Technology
in
Electronics & Communication Engineering

by
Katamneni Keerthana (Y18EC072)
Konanki Devi sri (Y18EC079)
Kongara Harika (Y18EC082)
Malleboina Sumanth (Y18EC096)

Under the guidance of
Dr.Tummala Ranga Babu M.Tech., Ph.D
Professor of ECE

Department of Electronics & Communication Engineering
R.V.R. & J.C. COLLEGE OF ENGINEERING
(Autonomous)
Approved by AICTE :: Affiliated to Acharya Nagarjuna University
Chowdavaram, Guntur - 522019, Andhra Pradesh, India

2022

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the thesis report entitled **"HANDWRITTEN DIGIT RECOGNITION USING MACHINE LEARNING"** that is being submitted by **"Katamneni Keerthana (Y18EC072), Konanki Devi sri (Y18EC079), Kongara Harika (Y18EC082), Malleboina Sumanth (Y18EC096)"** in partial fullfillment for the award of the Degree of **Bachelor of Technology** in **Electronics & Communication Engineering** to R.V.R. & J.C.College of Engineering (Autonomous) is a record of bonafide work carried out by him under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma.

Date:

Signature of Guide

Dr.Tummala Ranga Babu M.Tech., Ph.D

Professor of ECE

Signature of HOD

Dr.T.Ranga Babu M.Tech., Ph.D

Professor & Head

Abstract

Handwriting is one of the mode available for communication. In today's world every domain like medical, business etc (for example-in storing the data) seems to be digitizing. But for currently available non digital form of data to be digitalized, usage of manual labour for this process will be quite expensive and tedious as well.

To solve the above problem, pattern recognition applications can be helpful. In the text that is available, both digits and alphabets will be present from which we choose to solve the problem of digit recognition (0-9). Handwritten digit recognition is a technique or technology for automatically recognizing and detecting handwritten data into machine editable text through a Machine Learning model.

The goal is to implement a pattern classification method to recognize the handwritten digits provided in the MNIST dataset containing images of handwritten digits based on the K Nearest Neighbors machine learning algorithm. Due to the varieties of different writing styles in structure and angle of orientation, recognition of digits becomes difficult. It has it's applications in numeric entries in the forms filled up by hand (for example-tax forms, application forms), online handwriting on computer tablets, processing bank cheques and so on.

Keywords: Handwritten digit recognition, Machine Learning, K Nearest Neighbors, MNIST dataset

Table of contents

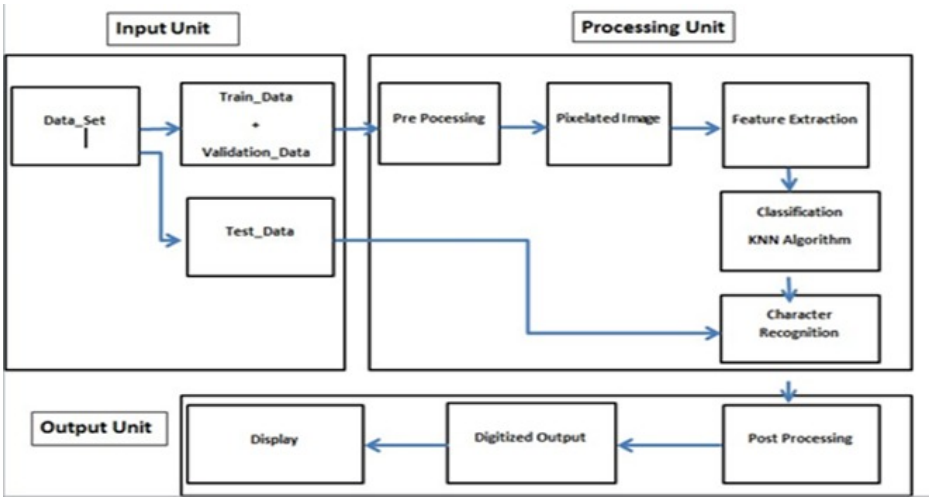
List of Figures

1

Introduction

Handwritten digit and letter recognition is a very important topic in the field of image processing and pattern recognition. Offline handwritten digit and letter recognition has numerous applications in different fields. Various algorithms using Machine Learning have been developed for classification and recognition of handwritten digits. There are four steps to be followed to solve a Machine Learning problem. They are conversion of domain problem to machine learning problem, data collection, training and deployment.

Conversion of the problem to ML problem involves steps like consulting the domain experts etc. In the data collection step, data is collected and prepared for training the model. Training step makes use of the data and prepares the model. Finally, in the deployment step, the model is made available to the end users.



1.1 TYPES OF ML

Machine Learning is divided into four types. They are Supervised Learning Unsupervised Learning Semi Supervised Learning Reinforcement Learning

1.1.1 Supervised Learning

This kind of learning uses a training set to teach models to yield the desired output. This training dataset includes inputs and correct outputs, which allow the model to learn over time. Supervised learning can be separated into two types of problems. They are Classification and Regression. Classification uses an algorithm to accurately assign test data into specific categories. It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labelled or defined. Regression is used to understand the relationship between dependent and independent variables. It is commonly used to make projections, such as for sales revenue for a given business. Examples of some of the Supervised Learning algorithms are Neural networks, Naive bayes, Linear regression, Support vector machine, K nearest neighbor, Random Forest etc.

The problem of digit recognition comes under Supervised Learning.

Problems and Issues in Supervised learning: Before we get started, we must know about how to pick a good machine learning algorithm for the given dataset. To intelligently pick an algorithm to use for a supervised learning task, we must consider the following factors.

1. Heterogeneity of Data: The algorithms that employ distance metrics are very sensitive to this, and hence if the data is heterogeneous, these methods should be the afterthought. Decision Trees can handle heterogeneous data very easily.

2. Redundancy of Data: If the data contains redundant information, i.e., contain highly correlated values, then it's useless to use distance-based methods because of numerical instability. In this case, some sort of Regularization can be employed to the data to prevent this situation.

3. Dependent Features: If there is some dependence between the feature vectors, then algorithms that monitor complex interactions like Neural Networks and Decision Trees fare better than other algorithms.

4. Curse of Dimensionality: If the problem has an input space that has a large number of dimensions, and the problem only depends on a subspace of the input space with small

dimensions, the machine learning algorithm can be confused by the huge number of dimensions and hence the variance of the algorithm can be high. In practice, if the data scientist can manually remove irrelevant features from the input data, this is likely to improve the accuracy of the learned function. In addition, there are many algorithms for feature selection that seek to identify the relevant features and discard the irrelevant ones, for instance Principle Component Analysis for unsupervised learning. This reduces the dimensionality.

5.Overfitting: The programmer should know that there is a possibility that the output values may constitute of an inherent noise which is the result of human or sensor errors. In this case, the algorithm must not attempt to infer the function that exactly matches all the data. Being too careful in fitting the data can cause overfitting, after which the model will answer perfectly for all training examples but will have a very high error for unseen samples. A practical way of preventing this is stopping the learning process prematurely, as well as applying filters to the data in the pre-learning phase to remove noises.

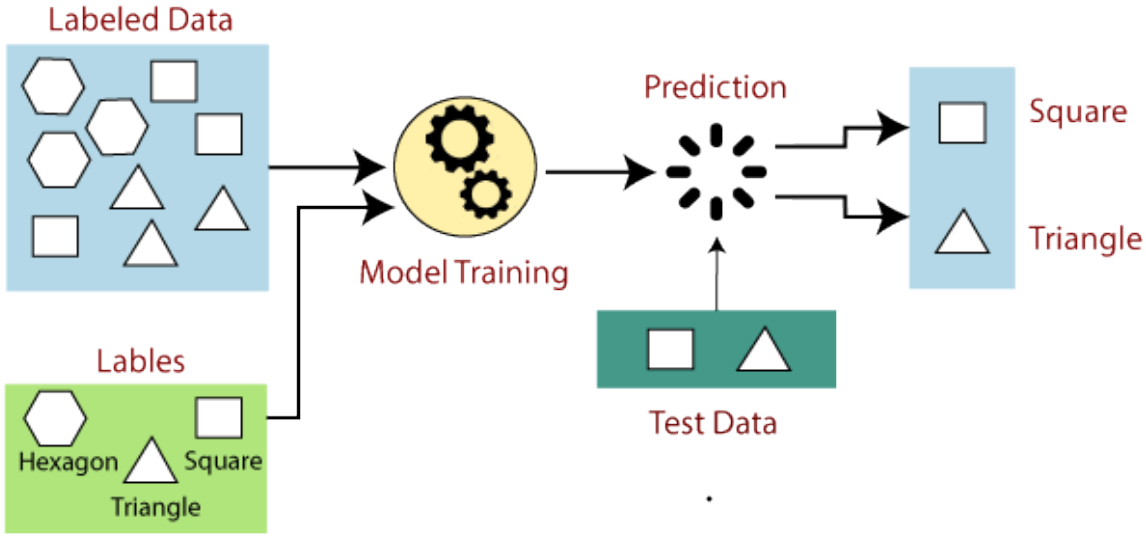


Figure 1.1: Supervised Machine Learning

1.1.2 Unsupervised Learning

As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things. It can be defined as:

Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.

This kind of learning uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition. Unsupervised learning models are utilized for three main tasks- clustering, association and dimensionality reduction. Examples of some of the Unsupervised machine learning algorithms include K-means clustering, Hierarchial clustering, Anomaly detection etc.

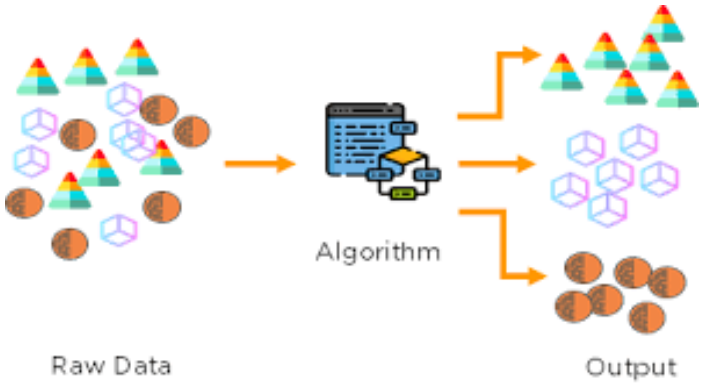


Figure 1.2: Unsupervised Machine Learning

1.1.3 Semi-supervised Learning

The basic disadvantage of supervised learning is that it requires hand-labeling by ML specialists or data scientists, and it also requires a high cost to process. Further unsupervised learning also has a limited spectrum for its applications. To overcome these drawbacks of supervised learning and unsupervised learning algorithms, the concept of Semi-supervised learning is introduced. In this algorithm, training data is a combination of both labeled and unlabeled data. However, labeled data exists with a very small amount while it consists of a huge amount of unlabeled data. Initially, similar data is clustered along with an unsupervised learning algorithm, and further, it helps to label the unlabeled data into labeled data. It is why label data is a comparatively, more expensive acquisition than unlabeled data.

This is another class of machine learning process and technique that also makes use of

unlabeled data for training (as unsupervised learning) but, typically, a small amount of labelled data with a large amount of unlabeled data is present and used by the model. This is usually referred to as partly labeled data.Semi-supervised learning programs do attempt to use certain standard assumptions to help them make use of unlabeled data. Semi-supervised learning schemes can be used for other kinds of semi-supervised learning, including natural language processing, classification, and regression on several services. Using the unlabeled data and applying unsupervised learning algorithms, grouping of data based on the similarities occur. This will help us find the most relevant samples in our data set. We can then label those and use them to train our model. Different types of machine learning algorithms include self-training, graph based, low density separation etc.

Some of the real world applications of Semisupervised learning includes Speech Analysis, Web content classification, Protein sequence classification, Text document classifier etc.

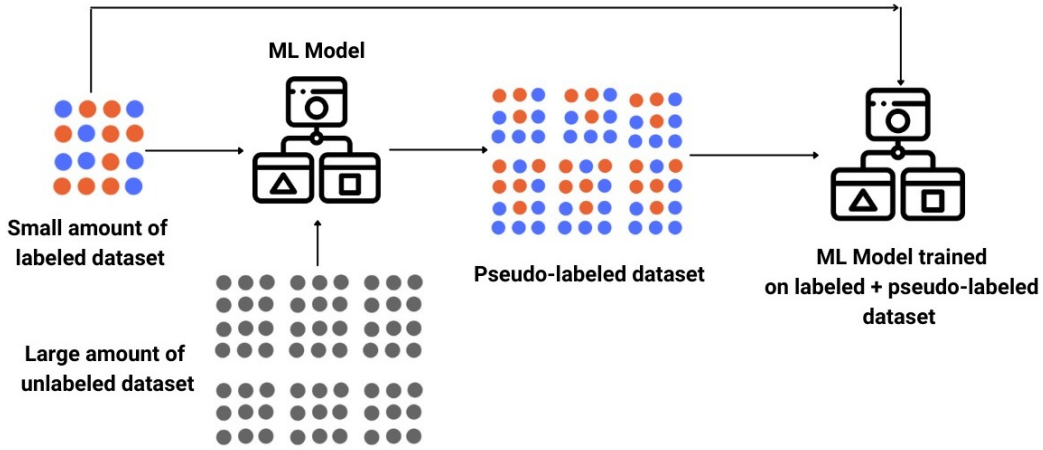


Figure 1.3: Semi-supervised Machine Learning

1.1.4 Reinforcement Learning

Reinforcement learning is the training of machine learning models to make a sequence of decisions. The agent learns to achieve a goal in an uncertain, potentially complex environment. In reinforcement learning, an artificial intelligence faces a game-like situation. The computer employs trial and error to come up with a solution to the problem. To get the machine to do what the programmer wants, the artificial intelligence gets either rewards or penalties for the actions it performs. Its goal is to maximize the total reward. Although the designer sets the reward policy–that is, the rules of the game–he gives the model no

hints or suggestions for how to solve the game. It's up to the model to figure out how to perform the task to maximize the reward, starting from totally random trials and finishing with sophisticated tactics and superhuman skills. By leveraging the power of search and many trials, reinforcement learning is currently the most effective way to hint machine's creativity. Examples include Q-learning, Deep Q network etc.

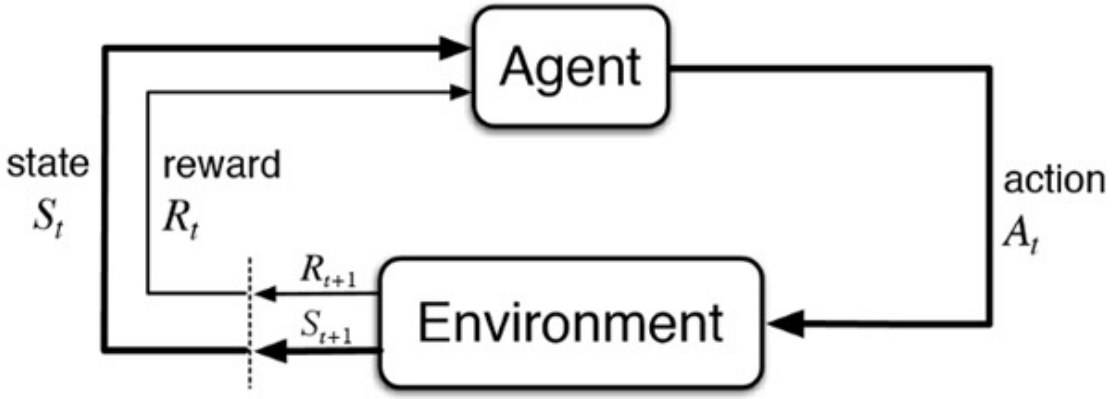


Figure 1.4: Reinforcement Machine Learning

There are basically two types of Reinforcement learning. They are Positive Reinforcement and Negative Reinforcement

The positive reinforcement learning means adding something to increase the tendency that expected behavior would occur again. It impacts positively on the behavior of the agent and increases the strength of the behavior.

This type of reinforcement can sustain the changes for a long time, but too much positive reinforcement may lead to an overload of states that can reduce the consequences.

The negative reinforcement learning is opposite to the positive reinforcement as it increases the tendency that the specific behavior will occur again by avoiding the negative condition. It can be more effective than the positive reinforcement depending on situation and behavior, but it provides reinforcement only to meet minimum behavior.

1.2 WORKFLOW

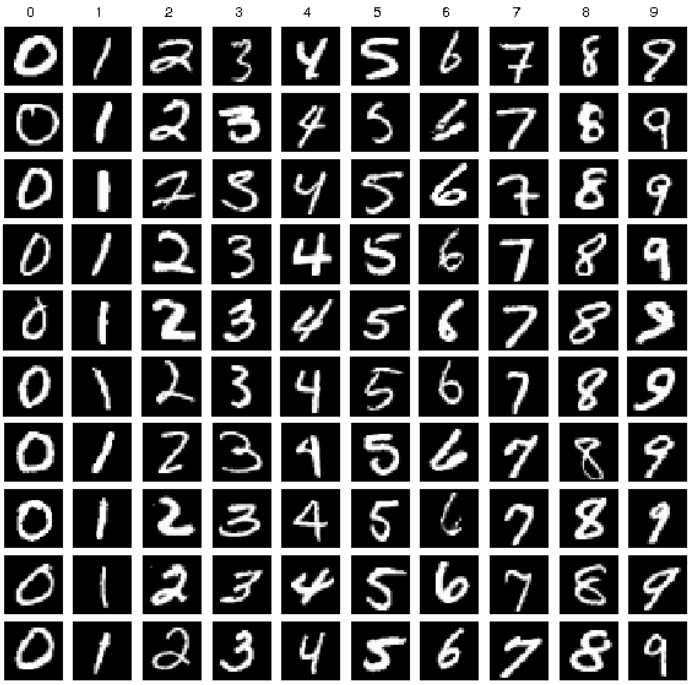
The first step deals with the extraction of data from the MNIST data set and the pre- processing of it. MNIST data set contains pictures of hand-written letters from different sources. The classification algorithm cannot work with the raw data which was extracted from the

site. Thus, we have to make few changes to the data sets. Thus, we change the raw data into a single one-dimensional array. This array will then be given to the final classifier that is trained.

The second step deals with the classification of the data sets which is the training dataset being provided to it. Along with the validation set which will be used to generalize the classification algorithm and make it ready to deal with real-world problems. Thus, after that the testdata, which works as the real-world problem is fed to the system and what the classification algorithm does is to classify the data which is given. The testdata is also a type of traindata, in other words, the format of the data is the same as the traindata except that the labels of the image will not be present. Finally, the classifier is trained and applied on the input. The output is then compared with the image using the matplot library. Due to the irregularities in the number of samples present in the dataset, the images given externally may predict different results which are not always correct. To avoid this, database is created from the digits written by us. These images are then converted by many steps to form a dataset. With this dataset, the classifier is trained. Thus, after training the classifier,the testdata or an external image which is similar to the real-world problem is fed to the algorithm and what the classification algorithm does is to classify the data which is given.

1.3 MNIST DATASET

MNIST is short for modified national institute of standard and technology database. The MNIST dataset is a large database of handwritten digits. It commonly used for training various image processing systems. This dataset is used for training models to recognize handwritten digits. This has an application in scanning for handwritten pin-codes on letters. MNIST contains a collection of 60,000, 28 x 28 images of handwritten digits from 0 to 9. The task is to classify a given image of a handwritten digit into one of 10 classes representing integer values from 0 to 9, inclusively. In the gray scale images, each pixel is encoded as an integer from 0 to 255.



It was created by ”re-mixing” the samples from NIST’s original datasets. The creators felt that since NIST’s training dataset was taken from American Census Bureau employees, while the testing dataset was taken from American high school students, it was not well-suited for machine learning experiments. Furthermore, the black and white images from NIST were normalized to fit into a 28x28 pixel bounding box and anti-aliased, which introduced grayscale levels.

The MNIST database contains 42,000 training images and 28,000 testing images. The digits have been size-normalized and centered in a fixed-size image. The original black and white (bilevel) images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. the images were centered in a 28x28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field.

This dataset is advantageous for the following reasons. MNIST dataset is publicly available. The data requires little to no processing before using. It is a voluminous dataset.

2

Literature Review

Yang Zong-chang, [1] In this study, to the main problem of establishing structure for the Artificial Neural Networks (ANN), from a microscopical perspective, two ideas called the fractal measurement of association multifaceted nature (FDCC) and the fractal measurement of the desire many-sided quality (FDEC) are presented. At that point a paradigm reference for setting up ANN structure taking into account the two proposed ideas is displayed that, the FDCC won't not be lower than its (FDEC), and when FDCC is equivalent or surmised to FDEC, the ANN structure may be an ideal one. The proposed measure is inspected with great results.

Selvi, P.P.; Meyyappan, T., [2] In the Study of the authors propose a method to recognize Arabic numerals using back propagation neural system. Arabic digit are the ten digits that were descended from the Indian numeral system. The recognition phase recognizes the numerals precisely. The prospect technique is implemented with Matlab coding. Model andwritten descriptions are tested with the proposed method and the results are plotted.

Sahu, N.; Raman, N.K., [3] In the Study of Character recognition systems for various languages and script has gain importance in recent decades and is the area of deep interest for a lot of researchers. Their growth is strongly integerated with Neural Networks.

Nguang Sing Ping; Yusoff, M.A., [4] Investigated on describes the application of 13-point feature of skeleton for an image-to-character credit. The representation can be a scanned handwritten character or drawn character from any graphic designing tool like Windows Paint clash. The representation is processed through conventional and 13-point feature of skeleton methods to extract the raw data.

Pradeep, J.; Srinivasan, E.; Himavathi, S., [5] In the Study of, an off-line handwritten English character recognition system using hybrid feature extraction technique and neural network classifiers are proposed. Neural Network (NN) topologies, namely, rear spread neural network and radial basis function network are built to classify the font. The k-nearest neighbour network is also built for evaluation. The nosh onward NN topology exhibits the highest recognition accuracy and is identified to be the most suitable classifier.

Budiwati, S.D.; Haryatno, J.; Dharma, E.M., [6] Investigated on Japanese language has complex writing systems, Kanji and Kana (Katakana and Hiragana). Each one has different

style of writing. One simple way to differentiate is Kanji have more strokes than Kana. Meanwhile, it needs a lot of effort to remember characters of Katakana and Hiragana, thus it will be very difficult to distinguish handwritten Katakana and Hiragana, since there are a lot of similar characters. This is the reason why we need pattern recognition.

3

Overview of the proposed method

Our task is to classify a given image of a handwritten digit into one of 10 classes representing integer values from 0 to 9, inclusively using a model that will be able to recognize the handwritten digits. So we will build an image classifier using scikit learn library on the MNIST dataset. Scikit learn short for sklearn the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

The steps involved in the proposed method are:

1. Importing the MNIST dataset

The digit images are separated into two sets: training and test. The train_x_df and test_x_df parts contain greyscale RGB codes (from 0 to 255). The train_y_df part contain labels from 0 to 9. We can see that the training set contains 42,000 images while the test set contains 28,000 images and that the images are indeed square with 28x28 pixels. They are to be loaded into the python file using pandas library. Incase of importing an image as an input, PIL library is used. Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

2. Reshaping the images in the dataset

We need to reshape the dataset to a single row format to be used by the trained model. This step is necessary while preparing own database. Beacuse, from seeing one of the inputs, we know that the dataset is 3-dimensional having red,blue and green coordinates, Since trainxdf represents (42000,784), where 42000 is the number of images in the train set and 784 represents the number of pixels since the size of each image is shaped to 28x28. So let's convert these to required format by the model. To reshape an external input image, opencv library is used. OpenCV-Python is a library of Python bindings designed to solve computer vision problems. OpenCV stands for Open Source Computer Vision Library, which is widely used for image recognition or identification. It was officially launched in

1999 by Intel. It was written in C/C++ in the early stage, but now it is commonly used in Python for the computer vision as well.

3. Normalizing the images in the database

Normalization is the process of changing the range of pixel intensity values to bring the image into a range that is more familiar or normal to the senses. We must normalize our data as it is always required in the trained model. This can be achieved by dividing the RGB codes by 255, which is the maximum RGB code minus the minimum RGB code (255 - 0). This step is not necessary for the database already available readily. But for the external input images, since the model is trained for the pixel values of images of size 28x28, general pixel values will not be useful because they contain red, blue and green values. So the image needs to be converted to a grey scale image and the pixel values needs to be normalized. The resized image from above is converted in the form of an array and divided by the corresponding value given.

4. Preparing the model

K-mean is a clustering technique which tries to split data points into K-clusters such that the points in each cluster tend to be near each other whereas K-nearest neighbor tries to determine the classification of a point, combines the classification of the K nearest points. As this is a multiclass classification problem with 10 possible outcomes (there are 10 digits overall from 0 to 9), the output layer will consist of 10 classes. Cross-validation involves randomly dividing the training set into groups, or folds, of approximately equal size. For instance, 90% data is used to train the model and remaining 10% to validate it. The hyperparameters like k value is then computed on the 10% validation data. This procedure repeats many times for obtaining the best performance of the model.

5. Compiling and fitting the model

Till now, we have created an empty kNN which is not optimized. So, we will now compile our model by providing some key parameters: the number of neighbours and the distance metric. There are multiple ways of evaluating models, but the most common one is the train-test split. When using a train-test split for model evaluation, you split the dataset into two parts.

Training data is used to fit the model. For kNN, this means that the training data will be used as neighbors.

Test data is used to evaluate the model. It means that you'll make predictions for the number of rings of each of the abalones in the test data and compare those results to the known true number of rings.

To fit a model from scikit-learn, you start by creating a model of the correct class. At this point, you also need to choose the values for your hyperparameters. For the kNN algorithm, you need to choose the value for k, which is called n_neighbors in the scikit-learn implementation.

You create an unfitted model with knn model. This model will use the nearest neighbors with the k value given to predict the value of a future data point. To get the data into the model, you can then fit the model on the training dataset using .fit(), you let the model learn from the data. At this point, knn model contains everything that's needed to make predictions on new data points. To find the best value for k, you're going to use a tool called GridSearchCV. This is a tool that is often used for tuning hyperparameters of machine learning models. In our case, it will help by automatically finding the best value of k for the dataset. GridSearchCV is available in scikit-learn, and it has the benefit of being used in almost the exact same way as the scikit-learn models.

6. Performance Evaluation

As the final step, we have evaluated the trained model with test set. The result label can be compared to the original value by plotting the image that is tested. For this purpose, matplot library is used. Matplotlib is a low level graph plotting library in python that serves as a visualization utility.Matplotlib was created by John D. Hunter.Matplotlib is open source and we can use it freely.

3.1 Choosing an algorithm

Determining which algorithm to use depends on many factors from the type of problem at hand to the type of output you are looking for.It depends on

Size of training- It is usually recommended to gather a good amount of data to get reliable

predictions. However, many a time, the availability of data is a constraint. If the training data is smaller, choose algorithms with high bias like Linear Regression, Naive Bayes etc. If the training data is large, we can go for low bias algorithms like KNN, Decision Trees etc.

Performance metrics- Metrics like accuracy means that the function predicts a response value for a given observation which is close to the true response value for that observation. To use which algorithm depends on the objective of the business problem.

Speed or training time- Higher accuracy means higher training time. Also, algorithms require more time to train on large training data. In real world, the choice of algorithm is driven by these two factors. Algorithms like Naïve Bayes, Linear regression are easy to implement and quick to run. Algorithms like SVM, random forests need a lot of time to train the data.

Number of features- The dataset may have large number of features that may not all be relevant and significant. For a certain type of data, the number of features can be large compared to to the number of data points.

3.2 Algorithm

K-nearest Neighbor is a non parametric and supervised machine learning algorithm used for both Classification and Regression.

Uses the phenomenon “similar things are near to each to each other”.

It predicts the class of the new data point by majority of votes of k nearest neighbors based on the similarity measure (i.e., distance functions). Varying the value of K, differs the prediction class. It is commonly used for its easy of interpretation and low calculation time. Knearest neighbor algorithm is simplest classification technique because the computations are simple. The classification of objects based on votes of its neighbors which represented by k. In K-nn object is classified to a particular class which has majority of votes.

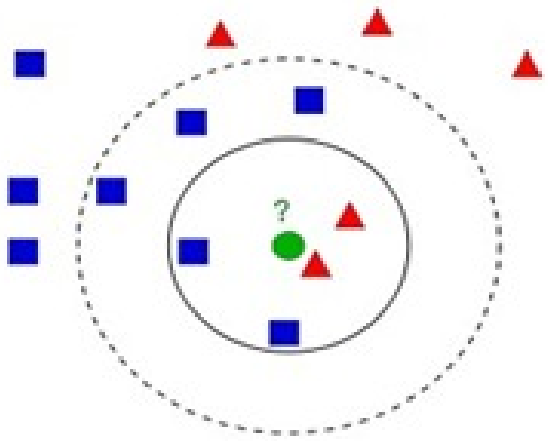


Figure 3.1: Classification using knn example

Then we find the nearest neighbours of new-comer. We can specify how many neighbours we want. It returns: The label given to new-comer depending upon the kNN theory we saw earlier. If you want Nearest Neighbour algorithm, just specify k=1 where k is the number of neighbours. The labels of k-Nearest Neighbours. Corresponding distances from new-comer to each nearest neighbour.

3.3 Working of the algorithm

• Compute the Euclidean distance between the test data point and all the training data.

• Sort the calculated distances in ascending order.

• Get the k nearest neighbors by taking top k rows from sorted array

• Find the majority class of these rows.

• Return predicted class.

Calculation of Euclidean distance:

Euclidean distance is the square root of the sum of squared distance between two points.

$$\sqrt{\sum[xi - yi]^2} \tag{3.1}$$

The Euclidean distance function calculates the difference between the squares of the points and finally the square root of the difference.

• Get the k nearest neighbors after sorting distance: In order to find the neighbors we need

to first sort the distance in ascending order to find the index of minimum distance. After that we will arrange the data according to the sorted index. Silcing the data according to the number of neighbors.

3.4 Performance Metrics

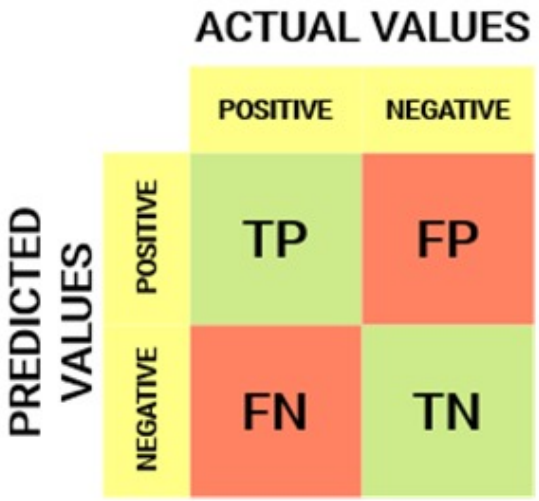


Figure 3.2: Confusion Matrix

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model.

• The target variable has two values: Positive or Negative

• The columns represent the actual values of the target variable

• The rows represent the predicted values of the target variable

True Positive (TP) -The predicted value matches the actual value.The actual value was positive and the model predicted a positive value True Negative (TN) -The predicted value matches the actual value.The actual value was negative and the model predicted a negative value False Positive (FP) –Type 1 error.The predicted value was falsely predicted.The actual value was negative but the model predicted a positive value. False Negative (FN) –Type 2 error.The predicted value was falsely predicted.The actual value was positive but the model predicted a negative value.

Accuracy is given by

$$\frac{TP + TN}{TP + FP + FN + TN} \tag{3.2}$$

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. Other performance metrics include Precision, Recall etc. Precision tells us how many of the correctly predicted cases actually turned out to be positive. Recall tells us how many of the actual positive cases we were able to predict correctly with our model.

In machine learning, we use the term parameters to refer to something that can be learned by the algorithm during training and hyperparameters to refer to something that is passed to the algorithm.For tuning of hyperparameters, GridSearchCV from scikitlearn library is used to find out the optimum value of k.GridSearchCV() determines the parameters like nneighbors and weights etc and the best set of hyperparameters to use for this data is evaluated and the score is compared.

4

Proposed method

4.1 Proposed Method

The process of recognizing handwritten digits using the datasets is as follows:

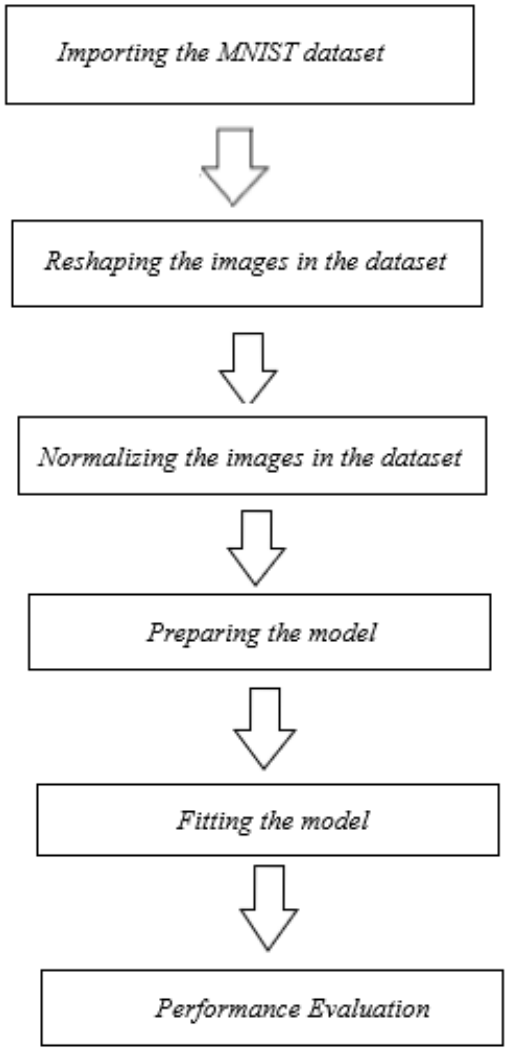


Figure 4.1: Steps in the proposed method

4.1.1 Importing the MNIST dataset

For importing the dataset, pandas library is used. Pandas is an open source library in Python. It provides ready to use high-performance data structures and data analysis tools. Pandas module runs on top of NumPy and it is popularly used for data science and data analytics. NumPy is a low-level data structure that supports multi-dimensional arrays and a wide

range of mathematical array operations. Pandas has a higher-level interface. It also provides streamlined alignment of tabular data and powerful time series functionality. DataFrame is the key data structure in Pandas. It allows us to store and manipulate tabular data as a 2-D data structure. Pandas provides a rich feature-set on the DataFrame. For example, data alignment, data statistics, slicing, grouping, merging, concatenating data, etc. There are 3 data structures provided by the Pandas module, which are as follows:

Series: It is a 1-D size-immutable array like structure having homogeneous data.
DataFrames: It is a 2-D size-mutable tabular structure with heterogeneously typed columns.
Panel: It is a 3-D, size-mutable array.

DataFrame is the most important and widely used data structure and is a standard way to store data. DataFrame has data aligned in rows and columns like the SQL table or a spreadsheet database. We can either hard code data into a DataFrame or import a CSV file, tsv file, Excel file, SQL table, etc. A CSV file is a text file with one record of data per line. The values within the record are separated using the "comma" character.

For analyzing data, we need to inspect data from huge volumes of datasets. Pandas provide many useful functions to inspect only the data we need. Some of the important functions are as follows:

We can use df.head(n) to get the first n rows or df.tail(n) to print the last n rows.We can get statistical summary (count, mean, standard deviation, min, max etc.) of the data using df.describe() function. Also, sorting can be done using df.sort_values() function where df contains data in the form of DataFrame.

Pandas provides a useful method, named read_csv() to read the contents of the CSV file into a DataFrame. This command is used to load the contents.

The drop() function is used to drop specified labels from rows or columns. Remove rows or columns by specifying label names and corresponding axis, or by specifying directly index or column names. When using a multi-index, labels on different levels can be removed by specifying the level.The syntax for this method is:

DataFrame.drop(self, labels=None, axis=0, index=None, columns=None, level=None, inplace=False, errors='raise')

The description of parameters are -

labels - Index or column labels to drop. axis - Whether to drop labels from the index (0 or 'index') or columns (1 or 'columns'). index - Alternative to specifying axis (labels, axis=0 is equivalent to index=labels). columns - Alternative to specifying axis (labels, axis=1 is equivalent to columns=labels). level - For MultiIndex, level from which the labels will be removed. inplace - If True, do operation inplace and return None. errors - If 'ignore', suppress error and only existing labels are dropped. The output of this function returns a dataframe without the removed index or column labels.It raises a KeyError if any of the labels is not found in the selected axis. The function is used to separate the labels that are present in the training set. Furthermore, the labels can be made a dataframe using slicing.

4.1.2 Reshaping the images in the dataset

To visualise the digits in the database, matplot library is used. Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Importing matplotlib can be done as

from matplotlib import pyplot as plt or import matplotlib.pyplot as plt

Firstly,to_numpy() function is used to return a NumPy ndarray representing the values in given Series or Index. This function will explain how we can convert the pandas Series to numpy Array. The reshape() function is used to give a new shape to an array without changing its data. The syntax is numpy.reshape(a, newshape, order='C'). a - Array to be reshaped. newshape - The new shape should be compatible with the original shape. If an integer, then the result will be a 1-D array of that length. One shape dimension can be -1. In this case, the value is inferred from the length of the array and remaining dimensions. order -¿ Read the elements of a using this index order, and place the elements into the reshaped array using this index order. 'C' means to read / write the elements using C-like

index order, with the last axis index changing fastest, back to the first axis index changing slowest. 'F' means to read / write the elements using Fortran-like index order, with the first index changing fastest, and the last index changing slowest. The order parameter is optional. Return value: reshaped_array : ndarray - This will be a new view object if possible; otherwise, it will be a copy. Note there is no guarantee of the memory layout (C- or Fortran- contiguous) of the returned array. The imshow() function in pyplot module of matplotlib library is used to display data as an image; i.e. on a 2D regular raster.

Syntax: matplotlib.pyplot.imshow(X, cmap=None, norm=None, aspect=None, interpolation=None, alpha=None, vmin=None, vmax=None, origin=None, extent=None, shape=, filternorm=1, filterrad=4.0, imlim=, resample=None, url=None, , data=None, kwargs)

Parameters: This method accept the following parameters that are described below: X: This parameter is the data of the image. cmap : This parameter is a colormap instance or registered colormap name. norm : This parameter is the Normalize instance scales the data values to the canonical colormap range [0, 1] for mapping to colors vmin, vmax : These parameter are optional in nature and they are colorbar range. alpha : This parameter is a intensity of the color. aspect : This parameter is used to controls the aspect ratio of the axes. interpolation : This parameter is the interpolation method which used to display an image. origin : This parameter is used to place the [0, 0] index of the array in the upper left or lower left corner of the axes. resample : This parameter is the method which is used for resembling. extent : This parameter is the bounding box in data coordinates. filternorm : This parameter is used for the antigrain image resize filter. filterrad : This parameter is the filter radius for filters that have a radius parameter. url : This parameter sets the url of the created AxesImage. Returns: This returns the following: image : This returns the AxesImage

matplotlib.pyplot.show(*, block=None) - Display all open figures.

The Parameters are blockbool, optional Whether to wait for all figures to be closed before returning. If True block and run the GUI main loop until all figure windows are closed. If False ensure that all figure windows are displayed and return immediately. In this case, you are responsible for ensuring that the event loop is running to have respon-

sive figures. Defaults to True in non-interactive mode and to False in interactive mode (see pyplot.isinteractive).

The visualised images are of the form:

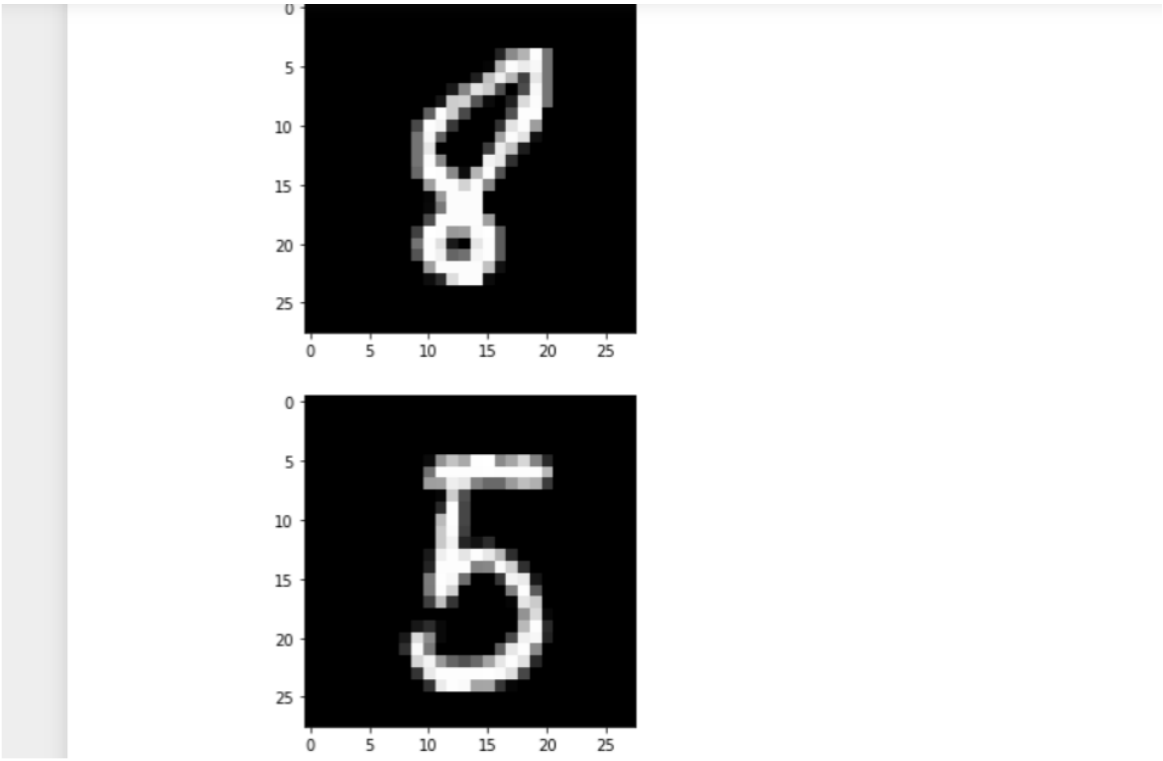


Figure 4.2: Visualizing images using matplot library

4.1.3 Normalizing the images in the database

This particular step is not required when working with the MNIST database since the data is already in the normalized form. We will discuss about the process later while creating own database.

4.1.4 Preparing the model

We use the scikit-learn library simply sklearn. Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. The function that is used here is

sklearn.model_selection.train_test_split(*arrays,test_size=None, train_size=None, random_state=None, shuffle=True, stratify=None)

This split arrays or matrices into random train and test subsets.

Parameters are given as

*arrays:sequence of indexables with same length / shape[0] Allowed inputs are lists, numpy arrays, scipy-sparse matrices or pandas dataframes. test_size:float or int, default=None If float, should be between 0.0 and 1.0 and represent the proportion of the dataset to include in the test split. If int, represents the absolute number of test samples. If None, the value is set to the complement of the train size. If train_size is also None, it will be set to 0.25. train_size:float or int, default=None If float, should be between 0.0 and 1.0 and represent the proportion of the dataset to include in the train split. If int, represents the absolute number of train samples. If None, the value is automatically set to the complement of the test size. random_state:int, RandomState instance or None, default=None Controls the shuffling applied to the data before applying the split. Pass an int for reproducible output across multiple function calls. shuffle:bool, default=True Whether or not to shuffle the data before splitting. If shuffle=False then stratify must be None. stratify:array-like, default=None If not None, data is split in a stratified fashion, using this as the class labels.

It returns splittinglist, length=2 * len(arrays) List containing train-test split of inputs.

To get the classifier implementing the k-nearest neighbors vote,KNeighborsClassifier function is used.It's syntax is-

sklearn.neighbors.KNeighborsClassifier(n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None)

Parameters are explained as n_neighbors:int, default=5 Number of neighbors to use by default for kneighbors queries. weights'uniform', 'distance' or callable, default='uniform' Weight function used in prediction. Possible values: 'uniform' : uniform weights. All points in each neighborhood are weighted equally. 'distance' : weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away. p:int, default=2 Power parameter for the Minkowski metric. When p = 1, this is equivalent to using manhattan_distance (l1),

and euclidean_distance (l2) for p = 2. For arbitrary p, minkowski_distance (l_p) is used. metric:str or callable, default='minkowski' The distance metric to use for the tree. The default metric is minkowski, and with p=2 is equivalent to the standard Euclidean metric. For a list of available metrics, see the documentation of DistanceMetric. If metric is "precomputed", X is assumed to be a distance matrix and must be square during fit. X may be a sparse graph, in which case only "nonzero" elements may be considered neighbors. metric_params:dict, default=None.Additional keyword arguments for the metric function. n_jobs:int, default=None.The number of parallel jobs to run for neighbors search. The resultant is the k-nearest neighbors classifier.

4.1.5 Fitting the model

For the specified parameter values, GridSearchCV function is used. This method performs exhaustive search over specified parameter values for an estimator.It's syntax is as follows: sklearn.model_selection.GridSearchCV(estimator,param_grid,*,scoring=None, n_jobs=None, refit=True, cv=None, verbose=0, pre_dispatch='2*n_jobs', error_score=nan, return_train_score=False)

To fit the classifier from above, fit(X,Y) - The parameters are specified as: X - array-like, sparse matrix of shape (n_samples, n_features) or (n_samples, n_samples) if metric='precomputed'.Training data. y - array-like, sparse matrix of shape (n_samples,) or (n_samples, n_outputs).Target values. The result is the fitted k-nearest neighbors classifier.

After fitting the model, the best parameters for the given number of neighbours and minkowski metric can be found by the best_params_method. This returns the parameter names mapped to their values.

The cv_results_attribute contains useful information for analysing the results of a search. Each row corresponds to a given parameter combination (a candidate) and a given iteration. These results can now be used to set the parameters efficiently after tuning the hyperparameters and cross validating with the validation set available earlier.

4.1.6 Performance Evaluation

Finally the predict function is called on the estimator with the best found parameters. It is

called as predict(X).

X : indexable, length n_samples. Must fulfill the input assumptions of the underlying estimator. This function returns an ndarray of shape (n_samples,) which is the predicted labels or values for X based on the estimator with the best found parameters. By calling this predict function, any test sample from the test_X_df (test dataset) the digit that corresponds to it will be returned as the result. The resultant can now be compared from visualising it by the help of matplot library.

4.2 Creating the database

In this process, digits that are handwritten are taken and scanned. Now these images are to be opened using PIL library. The PIL or Python Imaging Library is often confused with Pillow. Pillow is a fork of the PIL library in Python, that's why to install PIL we write "pip install Pillow", instead of "pip install PIL". Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. It is available for Windows, Mac OS X and Linux. The latest version of PIL is 1.1.7, was released in September 2009. Some of the file formats supported are PPM, PNG, JPEG, GIF, TIFF, and BMP. Some of the important features that this library offers you for image processing are:

• extensive file format support

• efficient internal representation

• creating thumbnails

• converting image files format

• applying filters to images

• also provides some powerful image processing capabilities

The module also provides a number of factory functions, including functions to load images from files, and to create new images. PIL.Image.open() - Opens and identifies the given image file. Further details about this command are -

Syntax: PIL.Image.open(fp, mode='r')

Parameters are - fp – A filename (string), pathlib.Path object or a file object. The file object

must implement read(), seek(), and tell() methods, and be opened in binary mode. mode – The mode. If given, this argument must be "r".The return type an image object.This command raises an IOError – If the file cannot be found, or the image cannot be opened and identified.

To convert this to grey scale image, the command convert is used. Image.convert() returns a converted copy of this image. For the "P" mode, this method translates pixels through the palette. If mode is omitted, a mode is chosen so that all information in the image and the palette can be represented without a palette. More detailed description is as follows: Syntax: Image.convert(mode=None, matrix=None, dither=None, palette=0, colors=256) Parameters are described as mode – The requested mode. See: Modes. matrix – An optional conversion matrix. If given, this should be 4- or 12-tuple containing floating point values. dither – Dithering method, used when converting from mode "RGB" to "P" or from "RGB" or "L" to "1". Available methods are NONE or FLOYDSTEINBERG (default). palette – Palette to use when converting from mode "RGB" to "P". Available palettes are WEB or ADAPTIVE. colors – Number of colors to use for the ADAPTIVE palette. Defaults to 256.The output returns an Image object.

The converted grey scale image can then be saved using the save function. Image.save() Saves this image under the given filename. If no format is specified, the format to use is determined from the filename extension, if possible. Syntax of this function is as follows- Image.save(fp, format=None, **params) Parameters are fp – A filename (string), pathlib.Path object or file object.format – Optional format override. If omitted, the format to use is determined from the filename extension. If a file object was used instead of a filename, this parameter should always be used. options – Extra parameters to the image writer. The command results in None Incase any exception occured it raises KeyError – if the output format could not be determined from the file name. Use the format option to solve this or IOError – if the file could not be written. The file may have been created, and may contain partial data.

To read an image from a file into an array, imread function from matplot library is used. The syntax is matplotlib.pyplot.imread (fname, format=None).

The parameters are fname : str or file-like The image file to read: a filename, a URL or a file-like object opened in read-binary mode.format : str, optional. The image file format assumed for reading the data. The image is loaded as a PNG file if format is set to ”png”, if fname is a path or opened file with a ”.png” extension. It returns a numpy.array which is the image data. The returned array has shape (M, N) for grayscale images. (M, N, 3) for RGB images. (M, N, 4) for RGBA images. PNG images are returned as float arrays (0-1). All other formats are returned as int arrays, with a bit depth determined by the file's contents. As specified above, the shape of the image is found using the shape method. The size (width, height) of the image can be obtained from the attribute shape.Not limited to OpenCV, the size of the image represented by ndarray, such as when an image file is read by Pillow and converted to ndarray, is obtained by shape. Since the classifier is prepared for 28x28 images as mentioned above, the image needs to be shaped to the same size. This is possible with the resize attribute. It's syntax is Image.resize(img,dimension,interpolation). The parameters are given by img - specifying the image file. Dimension can be made to the required size by reshaping the width and height of the image. The width and height of the image can be reshaped by slicing the shape attribute. Many times we need to resize the image i.e. either shrink it or scale up to meet the size requirements. OpenCV provides us several interpolation methods for resizing an image. The choices of interpolation are

cv2.INTER_AREA: This is used when we need to shrink an image.

cv2.INTER_CUBIC: This is slow but more efficient.

cv2.INTER_LINEAR: This is primarily used when zooming is required. This is the default interpolation technique in OpenCV.

Later the resultant resized image is converted to an array format to be used for future purposes. For such scenarios, numpy library is useful. NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. NumPy's main object is the homogeneous multidimensional array. It is more effective for array creation, array indexing, basic operations on arrays etc. The array attribute creates and returns an array.

numpy.array(object, dtype=None, *, copy=True, order='K', subok=False, ndmin=0,

like=None)

Parameters are object : array_like An array, any object exposing the array interface, an object whose array method returns an array, or any (nested) sequence. If object is a scalar, a 0-dimensional array containing object is returned. Other parameters listed in the syntax are given as dtype:data-type is optional, copy:bool is optional, order is optional, subok is of type bool and it is optional, ndmin of type int is optional and like of type array. Returns ndarray-An array object satisfying the specified requirements. The images as to support our model are normalized with the highest value - lowest value that is 255.

To view the resized image of shape 28x28, in the form of pixels the resultant image is to be converted to a dataframe because a Pandas DataFrame is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns. It is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns.

The syntax is pandas.DataFrame(data=None, index=None, columns=None, dtype=None, copy=None)

The parameters are specified as data:ndarray (structured or homogeneous), Iterable, dict, or DataFrame.Dict can contain Series, arrays, constants, dataclass or list-like objects. If data is a dict, column order follows insertion-order. If a dict contains Series which have an index defined, it is aligned by its index. index:Index or array-like.Index to use for resulting frame. Will default to RangeIndex if no indexing information part of input data and no index provided. columnsIndex or array-like.Column labels to use for resulting frame when data does not have them, defaulting to RangeIndex(0, 1, 2, . . . , n). If data contains column labels, will perform column selection instead. dtypedtype, default None.Data type to force. Only a single dtype is allowed. If None, infer. copybool or None, default None.Copy data from inputs. For dict data, the default of None behaves like copy=True. For DataFrame or 2d ndarray input, the default of None behaves like copy=False. The resultant will be of 28 rows and 28 columns specifying the intensity of each pixel.

Now, the next step is to convert the above obtained dataframe into a vector like containing single row with 784 columns. This can further be stored in the database. To achieve this, below steps are to be done.

Flattening is a technique that is used to convert multi-dimensional arrays into a 1-D array. In most cases, we will be dealing with a dataset which contains a large amount of images thus flattening helps in decreasing the memory as well as reducing the time to train the model.

The syntax is ndarray.flatten(order='C ').

The parameter order'C', 'F', 'A', 'K' is optional 'C' means to flatten in row-major (C-style) order. 'F' means to flatten in column-major (Fortran- style) order. 'A' means to flatten in column-major order if a is Fortran contiguous in memory, row-major order otherwise. 'K' means to flatten a in the order the elements occur in memory. The default is 'C'. The result is y: a ndarray.A copy of the input array, flattened to one dimension. This flattened array is converted to a dataframe using the attribute dataframe as mentioned above. This further converts to a single column dataframe. Further squeeze function is used.

Pandas Series.squeeze() function squeeze 1 dimensional axis objects into scalars. Series or DataFrames with a single element are squeezed to a scalar. DataFrames with a single column or a single row are squeezed to a Series. Otherwise the object is unchanged.

The syntax is Series.squeeze(axis=None)

Parameter - axis : A specific axis to squeeze. By default, all length-1 axes are squeezed. It returns a series projection after squeezing axis or all the axes. Finally, for reshaping the Pandas Series we are using reshape() method of Pandas Series object. Pandas.Series.values.reshape((dimension))

This function returns an ndarray with the values shape if the specified shape matches exactly the current shape, then return self. But the values need to be in the array format. So, after squeezing, the output needs to be converted to array using numpy.array function and later it is reshaped. The dimension needs to be in a format that it contains single row with multiple columns. Thus, the data returned can be given to the classifier or model that is prepared earlier.

Using the model trained above, this database is loaded and trained. In the same way of adding the samples into database, externally inputs will be given to predict for the model.

5

Conclusion

In this project, we learnt about the machine learning classifier — the k-Nearest Neighbor classifier, or simply k-NN for short. The k-NN algorithm classifies unknown data points by comparing the unknown data point to each data point in the training set. This comparison is done using a distance function or similarity metric. Then, from the k most similar examples in the training set, we accumulate the number of "votes" for each label. The category with the highest number of votes "wins" and is chosen as the overall classification. While simple and intuitive, the overall accuracy of 97.9% is achieved in the recognition process. Also, with the prepared database, external inputs are predicted and analysed. The work is carried out as an attempt, and the aim of the paper is to facilitate for pattern recognition. It is our future endeavour to modify this algorithm and design a still robust handwritten digit recognition for high recognition rate and also recognition of offline handwritten digit recognition.

6

References

Bibliography

[1] M. Sreeraj and S. M. Idicula,“k-NN Based On-Line Handwritten Character Recognition System,” First International Conference on Integrated Intelligent Computing, 2010, pp. 171-176

[2] Y. Lee,“Handwritten Digit Recognition Using K Nearest-Neighbor, Radial-Basis Function, and Backpropagation Neural Networks ”in Neural Computation, vol. 3, no. 3, pp. 440-449

[3] *Kaggle:*
`https://www.kaggle.com/dillsunnyb11/digit-recognizer`

[4] W. Liu, J. Wei and Q. Meng, “Comparisions on KNN, SVM, BP and the CNN for Handwritten Digit Recognition,” 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA), 2020, pp. 587-590

[5]
`https://www.researchgate.net/publication/286425352 Handwritten Digit Recognition Using K-Nearest Neighbour Classifier`

[6] Norhidayu binti Abdul Hamid, Nilam Nur Binti Amir Sharif “Handwritten Digit Recognition using Machine Learning Algorithms,”

[7] *Wikipedia*
`https : //en.wikipedia.org/wiki/Handwritingrecognition`

[8]

`https://scikit-learn.org/stable/modules/svm.html`

[9] Akanksha Gupta, Ravindra Pratap Narwaria, Madhav Singh, "Review on Deep Learning Handwritten Digit Recognition using Convolutional Neural Network,"

[10] Singh, P. K., Sarkar, R., Nasipuri, M, "A study of moment-based features on handwritten digit recognition. Applied Computational Intelligence and Soft Computing, "

[11] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," IEEE Signal Processing Magazine, vol. 29, pp. 141-142, "

[12] Liu, C., Fujisawa, H., "Handwritten digit recognition: Benchmarking of state-of-the-art techniques," Pattern Recognition, "

[13] M W Chen, M H Ng "Recognition of Unconstrained Handwritten Numerals using crossing features‖ International conference ISSPA 99 Brisbane, Australia, 22-25 August "

[14] U Pal and P.P.Roy, "Multi-oriented and curved text lines extraction from Indian documents", IEEE Trans on system, Man and Cybernetics-Part B, vol.34, pp.1667- 1684, "

[15] Y. LeCun, et al., "Comparison of learning algorithms for handwritten digit recognition, in: F. Fogelman-Souli e, P. Gallinari (Eds.), Proceedings of the International Conference on Artificial Neural Networks, Nanterre, France, pp. 53–60 "

[16] A.L.Koerich, R. Sabourin, C.Y.Suen, "Large off-line Handwritten Recognition: A survey", Pattern Analysis Application 6, 97-121, "

[17] Yann LeCun, "THE MNIST database of handwritten digits" Courant Institute, NYU Corinna Cortes, Google Labs, NewYork. http://www.research.att.com/yann/exdb/mnist/index.html "

[18] M. B. Abdulrazzaq and J. N. Saeed, “A Comparison of Three Classification Algorithms for Handwritten Digit Recognition,” 2019 International Conference on Advanced Science and Engineering (ICOASE), Zakho - Duhok, Iraq, 2019, pp. 58-63, doi: 10.1109/ICOASE.2019.8723702. ”

[19] Dr. R.Muralidharan “Object Recognition Using K-Nearest Neighbor Supported By Eigen Value Generated From the Features of an Image, ”

7

Appendix

7.1 Python 3.X

To get started working with Python 3, you'll need to have access to the Python interpreter. There are several common waysto accomplish this:

• Some operating systems, notably Linux, provide a package manager that can be run to Install Python.

• Python can be obtained from the Python Software Foundation website at python.org.Typically, that involves downloading the appropriate installer for your operating system and running it on your machine.

On macOS, the best way to install Python 3 involves installing a package manager called Homebrew. You'll see how to do this in the relevant section in the tutorial. On mobile operating systems like Android and iOS, you can install apps that provide a Python programming environment. This can be a great way to practice your coding skills on the go. Alternatively, there are several websites that allow you to access a Python interpreter online without installing anything on your computer at all. It is highly unlikely that your Windows system shipped with Python already installed. Windows systems typically do not. Fortunately, installing does not involve much more than downloading the Python installer from the python.org website and running it. If you are running Windows 10 Creators or Anniversary Update, you actually have another option for installing Python. These versions of Windows 10 include a feature called the Windows Subsystem for Linux, which allows you to run a Linux environment directly in Windows, unmodified and without the overhead of a virtual machine. For more information, see the Windows Subsystem for Linux Documentation article on the Microsoft website. For instructions on how to enable the subsystem in Windows 10 and install a Linux distribution, see the Windows 10 Installation Guide. You can also check out this presentation on YouTube by Sarah Cooley, one of the members of the WSL development team. Once you have installed the Linux distribution of your choice, you can install Python 3 from a Bash console window, just as you would if you were running that Linux distribution natively. Unlike most Unix systems and services, Windows does not include a system supported installation of Python. To make Python available, the CPython

team has compiled Windowsinstallers (MSI packages) with every release for many years. These installers are primarily intended to add a per-user installation of Python, with the core interpreter and library being used by a single user. The installer is also able to install for all users of a single machine, and a separate ZIP file is available for application-local distributions. As specified in PEP 11, a Python release only supports a Windows platform while Microsoft considers the platform under extended support. This means that Python 3.8 supports Windows Vista and newer. If you require Windows XP support, then please install Python 3.4. There are a number of different installers available for Windows, each with certain benefits and downsides. The full installer contains all components and is the best option for developers using Python for any kind of project. The Microsoft Store package is a simple installation of Python that is suitable for running scripts and packages, and using IDLE or other development environments. It requires Windows 10, but can be safely installed without corrupting other programs. It also provides many convenient commands for launching Python and its tools. The nuget.org packages are lightweight installations intended for continuous integration systems. It can be used to build Python packages or run scripts, but is not updateable and has no user interface tools. The embeddable package is a minimal package of Python suitable for embedding into a larger application.

Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. Jupyter Notebooks are popular development and training environment which have become the integrated development environment (IDE) for data science and machine learning. First, though: what is a “notebook”? A notebook integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media. This intuitive workflow promotes iterative and rapid development, making notebooks an increasingly popular choice at the heart of contemporary data science, analysis, and increasingly science at large. Best of all, as part of the open source Project Jupyter, they are completely free. The Jupyter project is

Figure 7.1: Jupyter Notebook logo

the successor to the earlier IPython Notebook, which was first published as a prototype in 2010. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. It contains Kernel. The kernel is a "computational engine" that executes the code written in the Notebook. It is similar to the back-end of the application. Pip, the package manager can be used to easily install Jupyter. On Windows, you can run Jupyter via the shortcut Anaconda adds to your start menu, which will open a new tab in your default web browser. This isn't a notebook just yet, but don't panic! There's not much to it. This is the Notebook Dashboard, specifically designed for managing your Jupyter Notebooks. Think of it as the launchpad for exploring, editing and creating your notebooks. Be aware that the dashboard will give you access only to the files and sub-folders contained within Jupyter's start-up directory; however, the start-up directory can be changed. It is also possible to start the dashboard on any system via the command prompt (or terminal on Unix systems) by entering the command jupyter notebook; in this case, the current working directory will be the start-up directory. The astute reader may have noticed that the URL for the dashboard is something like http://localhost:8888/tree. Localhost is not a website, but indicates that the content is being served from your local machine: your own computer. Jupyter's Notebooks and dashboard are web apps, and Jupyter starts up a local Python server to serve these apps to your web browser, making it essentially platform independent and opening the door to easier sharing on the web. The dashboard's interface

is mostly self-explanatory — though we will come back to it briefly later. So what are we waiting for? Browse to the folder in which you would like to create your first notebook, click the "New" drop-down button in the top-right and select "Python 3" (or the version of your choice). . The notebook consists of several cells. A cell is a textbox that allows us to edit and write code, with syntax highlighting, similar to that of a code editor or IDE. The kernel associated with the notebook takes care of executing the code written in the cell. Once the kernel finishes computing the results, they are retrieved and displayed in the notebook (below the cell that was executed) as the cell's output. We can also add some notes, headings using Markdown option.

Anaconda

Python is a high-level programming language devised by Guido van Rossum first released in 1991. It's the most popular coding language used by software developers to build, control, manage and for testing. In Python, compiled code is stored with the file extension .py. For example – hello.py It is also an interpreter which executes Python programs. The python interpreter is called python.exe on Windows. To execute hello.py program, type – python hello.py Together with a list of Python packages, tools like editors, Python distributions include the Python interpreter. Anaconda is one of several Python distributions. Anaconda is a new distribution of the Python and R data science package. It was formerly known as Continuum Analytics. Anaconda has more than 100 new packages. This work environment, Anaconda is used for scientific computing, data science, statistical analysis, and machine learning. The latest version of Anaconda 5.0.1 isreleased in October 2017.The released version 5.0.1 addresses some minor bugs and adds useful features, such as updated R language support. All of these features weren't available in the original 5.0.0 release. This package manager is also an environment manager, a Python distribution, and a collection of open source packages and contains more than 1000 R and Python Data Science Packages. There's no big reason to switch to Anaconda if you are completely happy with you regular python. But some people like data scientists who are not full-time developers, find anaconda much useful as it simplifies a lot of common problems a beginner runs into. Anaconda can help with –

• Installing Python on multiple platforms

• Separating out different environments

• Dealing with not having correct privileges and

• Getting up and running with specific packages and libraries

The free version of Anaconda distribution community edition can be downloaded directly from Anaconda's website. For the enterprise edition, one need professional support from Anaconda's sales team. Conda treats Python the same as any other package,so it is easy to manage and update multiple installations. Anaconda supports Python 2.7, 3.4, 3.5 and 3.6. The default is Python 2.7 or 3.6, depending on which installer you used:

• For the installers "Anaconda" and "Miniconda," the default is 2.7

• For the installers "Anaconda 3" or "Miniconda 3," the default is 3.6 Once downloaded the .exe file, run through the installer. Do accept the terms, and finish the installation. To check, close the browser and pull up the terminal. Once the installation is complete, it should have automatically added that to the path. To test this, go ahead and type 'python'. The version of python i.e. 3 will show and also the anaconda distribution will be seen. If you install the 4 versions of Anaconda, then all the packages, which are there in the packages, can also be imported easily. To check type import numpy or import matplotlib and run that.

IPYNB File

An IPYNB file is a notebook document created by Jupyter Notebook, an interactive computational environment that helps scientists manipulate and analyze data using Python. It contains all the content from a Jupyter Notebook web application session, including computation inputs and outputs, mathematical functions, images, and explanatory text. IPYNB files can be exported to the .HTML, .PDF, reStructuredText, and LaTeX formats. IPYNB notebooks are plain text files formatted using JSON, making them human-readable and easy to share with others. Oftentimes, scientists will host their IPYNB files online at publicly-accessible URLs. This allows other Jupyter Notebook users to either:

Download the files and open them in their own instance of Jupyter Notebook Paste the files' URLs into the online Jupyter Notebook Viewer, to view the notebook on the web without installing any software. You can open an IPYNB file in Jupyter Notebook (cross-platform),

Jupyter Notebook Viewer (Web), Cantor (Linux), or Google Colaboratory (Web). To open an IPYNB file in Jupyter Notebook Viewer, the file must be hosted online (via GitHub or another file hosting service).

Spyder Integrated Development Environment

It is always necessary to have interactive environments to create software applications and this fact becomes very important when you work in the fields of Data Science, engineering, and scientific research. The Python Spyder IDE has been created for the same purpose. In this article, you will be learning how to install and make use of Spyder or the Scientific Python and Development IDE. An integrated development environment (IDE) facilitates computer programmers by integrating fundamental tools (e.g., code editor, compiler, and debugger) into a single software package. Users do not need to install the language's compiler/interpreter on their machines; an IDE provides the environment itself. Spyder is a dedicated IDE for Python. It incorporates some useful features that make it a popular IDE. Spyder is an open-source cross-platform IDE. The Python Spyder IDE is written completely in Python. It is designed by scientists and is exclusively for scientists, data analysts, and engineers. It is also known as the Scientific Python Development IDE and has a huge set of remarkable features which are discussed below. Some of the remarkable features of Spyder are:

• Availability of breakpoints (debugging and conditional breakpoints)

• Interactive execution which allows you to run line, file, cell, etc.

• Can clear variables automatically (or enter debugging)

• Navigation through cells, functions, blocks, etc. can be achieved through the Outline Explorer

• It provides real-time code introspection (The ability to examine what functions, keywords, and classes are, what they are doing and what information they contain)

• Automatic colon insertion after if, while, etc.

• Inline display for graphics produced using Matplotlib.

It also provides features such as help, file explorer, find files, etc. Writing code in Spyder becomes very easy with its multi-language code editor and a number of powerful tools. As

mentioned earlier, the editor has features such as syntax highlighting, real-time analysis of code, style analysis, on-demand completion, etc. When you write your code, you will also notice that it gives a clear call stack for methods suggesting all the arguments that can be used along with that method. Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. Beyond its many built-in features, its abilities can be extended even further via its plugin system and API. Furthermore, Spyder can also be used as a PyQt5 extension library, allowing developers to build upon its functionality and embed its components, such as the interactive console, in their own PyQt software.

7.2 Libraries :

A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer. As we don't need to write the same code again and again for different programs. Python libraries play a very vital role in fields of Machine Learning, Data Science, Data Visualization, etc.

7.2.1 Working of Python Library

Python library is simply a collection of codes or modules of codes that we can use in a program for specific operations. We use libraries so that we don't need to write the code again in our program that is already available. But how it works. Actually, in the MS Windows environment, the library files have a DLL extension (Dynamic Load Libraries). When we link a library with our program and run that program, the linker automatically searches for that library. It extracts the functionalities of that library and interprets the program accordingly. The Python standard library consists of more than 200 core modules. All these works together to make Python a high-level programming language. Python Standard Library plays a very important role. Without it, the programmers can't have access to the functionalities of Python. But other than this, there are several other libraries in Python that make a programmer's life easier. Some of the libraries used in this project are:

a) Pandas

b) Matplotlib

c) Numpy

d) OpenCV

e) sklearn

7.2.1.1 Pandas

Python Pandas is defined as an open-source library that provides high-performance data manipulation in Python. It is used for data analysis in Python and developed by Wes McKinney in 2008. The name of Pandas is derived from the word Panel Data, which means an Econometrics from Multidimensional data. Data analysis requires lots of processing, such as restructuring, cleaning or merging, etc. There are different tools are available for fast data processing, such as NumPy, SciPy, Cython, and Panda. But we prefer Pandas because working with Pandas is fast, simple and more expressive than other tools.Pandas is built on top of the NumPy package. Before Pandas, Python was capable for data preparation, but it only provided limited support for data analysis. So, Pandas came into the picture and enhanced the capabilities of data analysis. It can perform five significant steps required for processing and analysis of data irrespective of the origin of the data, i.e., load, manipulate, prepare, model, and analyze.

Some of the important functions used for Machine Learning available in Pandas library are: read_csv() function helps read a comma-separated values (csv) file into a Pandas DataFrame. All you need to do is mention the path of the file you want it to read. It can also read files separated by delimiters other than comma

head(n) is used to return the first n rows of a dataset. By default, df.head() will return the first 5 rows of the DataFrame.

value_counts() returns a Pandas Series containing the counts of unique values.

7.2.1.2 Matplotlib

Matplotlib is one of the most popular and oldest plotting libraries in Python which is used in Machine Learning. In Machine learning, it helps to understand the huge amount of data through different visualisations.

Matplotlib is an open-source plotting library in Python introduced in the year 2003. It is a very comprehensive library and designed in such a way that most of the functions for plotting in MATLAB can be used in Python. It consists of several plots like the Line Plot, Bar Plot, Scatter Plot, Histogram etc. through which we can visualise various types of data.

It can be imported as

importing pyplot module from matplotlib

from matplotlib import pyplot as plt

or

import matplotlib.pyplot as plt

Some of the functions are

Histograms are created using plt.hist() function.

The plot() function in pyplot module of matplotlib library is used to make a 2D hexagonal binning plot of points x, y.

7.2.1.3 Numpy

NumPy stands for 'Numerical Python'. It is an open-source Python library used to perform various mathematical and scientific tasks. It contains multi-dimensional arrays and matrices, along with many high-level mathematical functions that operate on these arrays and matrices. An array is known as the central data structure of the NumPy library. Array in NumPy is called as NumPy Array.

NumPy can be import as follows:

import numpy as np

Some of the functions are

The basic ndarray can be created using the numpy.array() function in NumPy.

.shape() returns a tuple listing the length of the array along each dimension.

7.2.1.4 Pandas OpenCV (open-source computer vision) is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android.

It can process images and videos to identify objects, faces, or even the handwriting of a human. We can integrate it with various libraries, such as NumPy, so that whatever operations one can do in NumPy can be combined with OpenCV. The library has extensive collection of computer vision and machine learning algorithms.

Using OpenCV it becomes easy to do complex tasks such as identify and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D object models, generate 3D point clouds from stereo cameras, stitch images together to generate an entire scene with a high-resolution image and many more.

Functions of OpenCV:

• Read, write and display the image

• Rotate the images

• Image thresholding

• Edge detection

• Contour detection

• Face detection

Gary Bradsky invented OpenCV in 1999 and the OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications, part of a series of projects including real-time ray tracing and 3D display walls. The first alpha version of OpenCV was released to the public at the IEEE Conference on Computer Vision and Pattern Recognition in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. A version 1.1 "pre-release" was released in October 2008. The second major release of the OpenCV was in October 2009. OpenCV 2 includes major changes to the C++ interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core sys-

tems). Official releases now occur every six months and development are now done by an independent Russian team supported by commercial corporations. In August 2012, support for OpenCV was taken over by a non-profit foundation OpenCV.org, which maintains a developer and user site.

7.2.1.5 Scikitlearn

Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction. Components of scikit-learn:The library is focused on modelling data. Some popular groups of models provided by scikit-learn include:

• Clustering: For grouping unlabelled data such as K Means.

• Cross Validation: For estimating the performance of supervised models on unseen data.

• Datasets: For test datasets and for generating datasets with specific properties for investigating model behaviour.

• Dimensionality Reduction: For reducing the number of attributes in data for summarization, visualization and feature selection such as Principal component analysis.

• Feature extraction: For defining attributes in image and text data.

• Feature selection: For identifying meaningful attributes from which to create supervised models.

• Parameter Tuning: For getting the most out of supervised models.

• Supervised Models: A vast array not limited to generalized linear models, discriminate analysis, naive bayes, neural networks, support vector machines and decision trees.

This library contains methods like accuracy,Logistic Regression,KNeighbors Classifier,Linear Regression etc.