A Project Report On

Hand-Written Digit Recognizer

Submitted in partial fulfilment for the award of Bachelor's degree in Computer Application

Submitted by:

Suhail Ahmed Bhat	1040-IITM-2019
Gowhar Yousuf	1026-IITM-2019
Sayar Ahmed Rather	1034-IITM-2019

Under the guidance of:

Ms. Qurat Aamir

DEPARTMENT OF COMPUTER SCIENCE, IQBAL INSTITUTE OF TECHNOLOGY AND MANAGEMENT, AFFILIATED TO UNIVERSITY OF KASHMIR

Hazratbal Srinagar Kashmir-190006

DEPARTMENT OF COMPUTER SCIENCE, IQBAL INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Certificate

Certified that **Suhail Ahmed Bhat [1040-IITM-2019]**, **Gowhar Yousuf [1026-IITM-2019]** and **Sayar Ahmed Rather [1034-IITM-2019]** have carried out the project work presented in this report titled "Hand-Written Digit Recognizer" of the award of Bachelors of Computer Application from the University of Kashmir under my supervision. The report embodies result of work and studies carried out by students themselves and the content of the report do not form any basis for the award of any degree to the candidate or to any other.

Date: __/__/____

Principal	**H.O.D.**	**Ms. Qurat Aamir**
(IITM)	(Department of Computer Science)	(Lecturer Computer Science Dept.)

Abstract

Nowadays, the scope of machine learning and deep learning studies is increasing day by day. Handwriting recognition is one of the examples in daily life for this field of work. Data storage in digital media is a method that almost everyone is using nowadays. At the same time, it has become a necessity for people to store their notes in digital media and even take notes directly in the digital environment. As a solution to this need, applications have been developed that can recognize numbers, characters, and even text from handwriting using machine learning and deep learning algorithms. Moreover, these applications can recognize numbers, characters, and text from handwriting and convert them into visual characters. This project investigated the performance comparison of machine learning algorithms commonly used in handwriting recognition applications and which of them are more efficient. This study also developed a handwriting recognition system for numbers like these mentioned applications. In the Pattern Recognition field, growing interest has been shown in recent years for Multiple Classifier Systems and particularly for Bagging, Boosting and Random Subspaces. Those methods aim at inducing an ensemble of classifiers by producing diversity at different levels. Following this principle, Breiman has introduced in 2001 another family of methods called Random Forest. Our work aims at studying those methods in a strictly pragmatic approach, in order to provide rules on parameter settings for practitioners. For that purpose, we have experimented the Forest algorithm, considered as the Random Forest reference method, on the MNIST handwritten digits database. In this paper, we describe Random Forest principles and review some methods proposed in the literature. We present next our experimental protocol and results. We finally draw some conclusions on Random Forest global behaviour according to their parameter tuning. This report describes several techniques used for pre-processing the handwritten digits, as well as several ways in which neural networks were used for the recognition task. Whereas the main goal was a purely educational one, a moderate recognition rate of 97% was reached on a test.

Keywords:

Machine Learning, Handwriting Recognition, MNIST Dataset, Random Forest Model, K Nearest Neighbour Model, Logistic Regression Model, Perceptron Model, Gradio (User Interface), CSS Sheets).

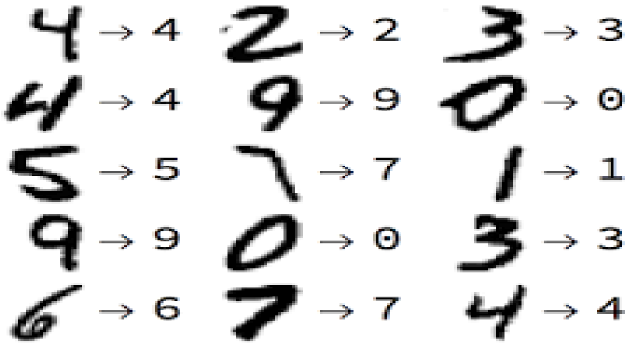


Table of Content

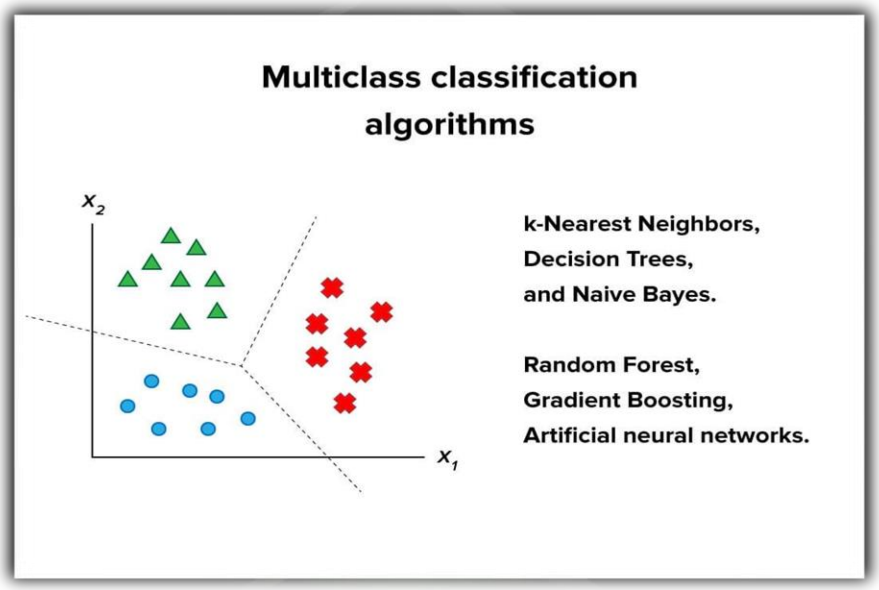
1. Introduction

Hand-written Digit Recognition is the project done using Machine Learning. This project helps computer or a model to train itself from the past available data and using the model we can predict the labels for different hand-written digits in future. Every person has a different handwriting, but this project has the great accuracy of predicting the label for every handwriting.

Machine Learning is now considered to be one of the biggest innovations in the field of Artificial Intelligence since the microchip. AI is on its way to becoming a crucial part of our daily routine in many people's lives. With the availability of so much data that is processed, it is now possible to build any predictive models that can analyse and study the complex data stored in the dataset to find useful insights and deliver more accurate results. Many companies build various Machine Learning models by using all the data received from people and use this in order to identify profitable opportunities and avoid risks.

Machine Learning is the study of computer algorithms that improves automatically through experience. It provides the ability to automatically learn and improve from experience without being explicitly programmed. The algorithm finds natural patterns in data that generate insight and help you make better diagnosis.

In this project we used matplotlib for plotting the images and we used sklearn machine learning library as our dataset. Handwritten Digit Recognition is the problem that belongs to the classification type of machine learning models. Classification is the supervised machine learning technique used to classify the new observations or in simple words, classification techniques are used to recognize the class to which the data point belongs.



2)Existing System And Our Solution:

2.1:Existing Systems:

There are many projects already done on Handwritten Digit Recognition. And most of the projects are done using CNN short form for Convolutional Neural Network or many other have used ANN (Artificial Neural Network. And even though the accuracy of these projects is very high, but CNN and ANN are very hard to understand and implement.

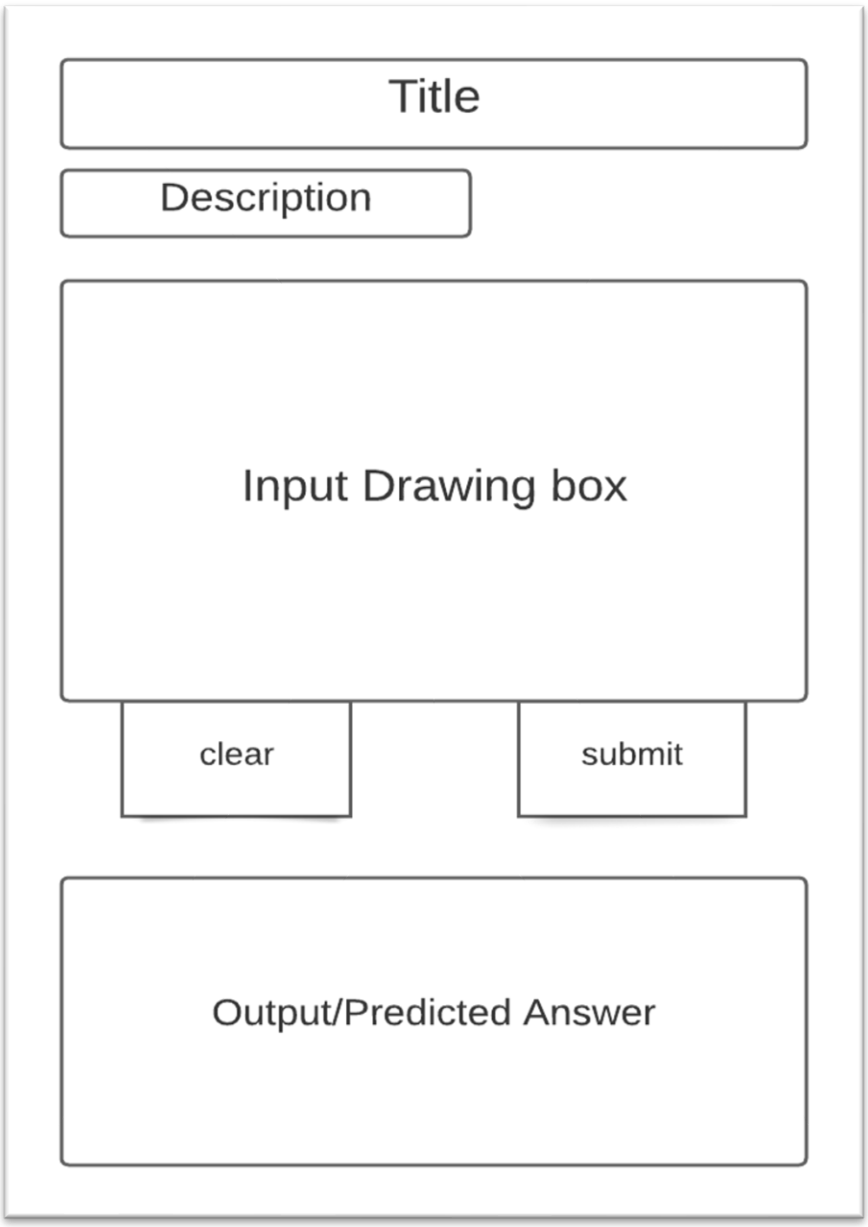
And other thing is that a naïve user can not test these applications as they don’t have provided any GUI for the user interaction. The testing is done by the developer itself using some evaluation matrices, but someone with no knowledge of AI, cannot understand or try to test these projects.

2.2:Our Solution:

In this project we have not used CNN or ANN but, we used Machine Learning which is comparatively easier to understand and implement. We have tried different Classification model to get the good accuracy, and finally we decided to use Random Forest Classifier which gives us the accuracy of 97%. We even tried Logistic Regression, SVM, Naïve Bayes but the better accuracy was shown by the Random Forest Classifier and that is why we choose this model.

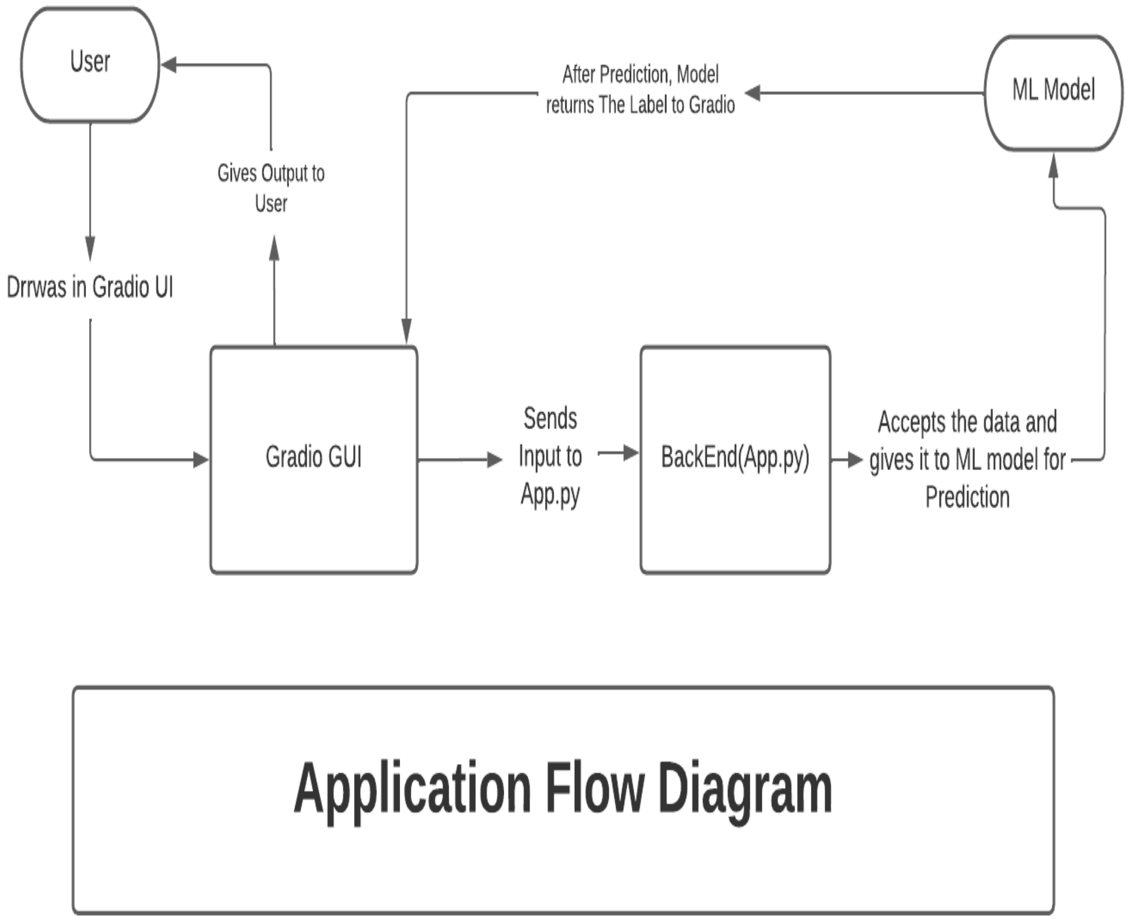
And Other issue was the Graphical User Interface, in this project we used Gradio which is a python library to build an interface for the users with zero knowledge of machine learning to try, test and have fun with our model.

3) WireFrame of the frontend Gradio UI



- There is the title present on the top of the page following with description.
- Then there is an input sketch pad, where user can draw handwritten digits.
- Then there are two buttons, clear and submit.
- And at the bottom there is output field where the predicted label from the model will be seen

4) Application Flow Diagram



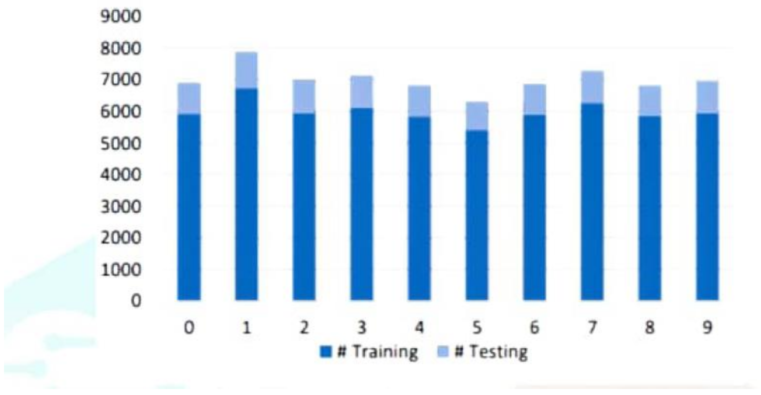
- The User Interacts with the Gradio User Interface.
- In the Gradio UI, the users Draws the Inputs in the sketch box and submits it to the backend (app.py).
- App.py accepts the data and gives it to the Machine Learning models which is already trained for the prediction.
- The model predicts the output and gives the output back to the Gradio UI.
- And the user can check the prediction in the webapp.

5) MNIST Dataset

MNIST is a widely used dataset of handwritten digits that contains 60,000 handwritten digits for training a machine learning model and 10,000 handwritten digits for testing the model. It was introduced in 1998 and has become a standard benchmark for classification tasks. It is also called the "Hello, World" dataset as it's very easy to use. MNIST was derived from an even larger dataset, the NIST Special Database19 which not only contains digits but also uppercase and lowercase handwritten letters.

In the MNIST dataset each digit is stored in a grayscale image with a size of 28x28 pixels. In the following you can see the first 10 digits from the training set:

Examples of samples and overall distribution MNIST dataset is given below:



Today, the dataset is considered as too simple for testing the modern, very complex deep learning models with up to billions of parameters.

6)Steps involved in training our Model (Random Forest)

Every Machine Learning model follows these steps:

Step 1: Getting the Data, Here MNIST dataset is used.

```
Train_data = pd.read_csv("./dataset/mnist_train.csv")

x_train = Train_data.iloc[:,1:].values
y_train = Train_data.label

Test_data = pd.read_csv("./dataset/mnist_test.csv")
x_test = Test_data.iloc[:,1:].values
y_test = Test_data.label
```

Step 2: Preparing the Data, Here Normalization and reshaping is done.

```
x_train = x_train.reshape(x_train.shape[0], 28*28)
x_test = x_test.reshape(x_test.shape[0], 28*28)

x_train = x_train/255
x_test = x_test/255
```

Step 3: Training the model

```
rfc = RandomForestClassifier()
rfc.fit(x_train, y_train)
```

Step 4: Evaluating the model

```
print(rfc.score(x_test, y_test))
```

Step 5: Predicting the label for new Observations.

```
rfc.predict(img.reshape(1,-1))[0]
```

7)Random Forest Model

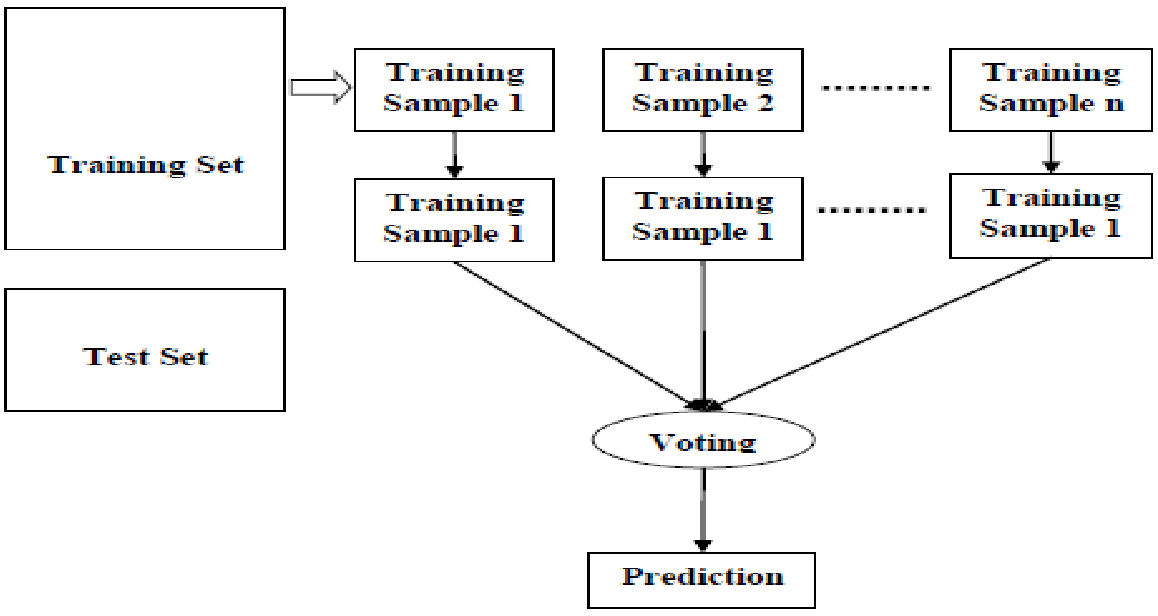
Random forest is a commonly used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fuelled its adoption, as it handles both classification and regression problems. Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Accuracy in our project : 0.97

Working of Random Forest Algorithm

We can understand the working of Random Forest algorithm with the help of following steps :

- **Step 1** − First, start with the selection of random samples from a given dataset.
- **Step 2** − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- **Step 3** − In this step, voting will be performed for every predicted result.
- **Step 4** − At last, select the most voted prediction result as the final prediction result.

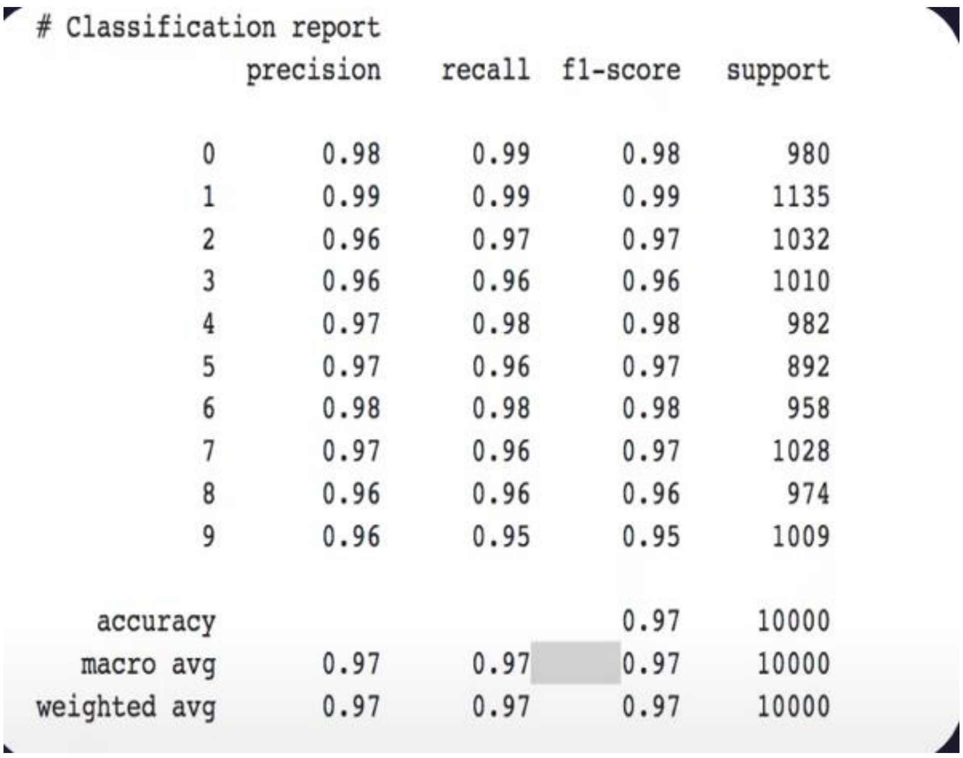


Benefits of random forest

There are a number of key advantages and challenges that the random forest algorithm presents when used for classification or regression problems. Some of them include:

Key Benefits

- **Reduced risk of overfitting:** Decision trees run the risk of overfitting as they tend to tightly fit all the samples within training data. However, when there's a robust number of decision trees in a random forest, the classifier won't overfit the model since the averaging of uncorrelated trees lowers the overall variance and prediction error.
- **Provides flexibility:** Since random forest can handle both regression and classification tasks with a high degree of accuracy, it is a popular method among data scientists. Feature bagging also makes the random forest classifier an effective tool for estimating missing values as it maintains accuracy when a portion of the data is missing.
- **Easy to determine feature importance:** Random Forest makes it easy to evaluate variable importance, or contribution, to the model. There are a few ways to evaluate feature importance. Gini importance and mean decrease in impurity (MDI) are usually used to measure how much the model's accuracy decreases when a given variable is excluded. However, permutation importance, also known as mean decrease accuracy (MDA), is another importance measure. MDA identifies the average decrease in accuracy by randomly permutating the feature values in samples.


```
# Classification report
              precision    recall  f1-score   support

           0       0.98      0.99      0.98       980
           1       0.99      0.99      0.99      1135
           2       0.96      0.97      0.97      1032
           3       0.96      0.96      0.96      1010
           4       0.97      0.98      0.98       982
           5       0.97      0.96      0.97       892
           6       0.98      0.98      0.98       958
           7       0.97      0.96      0.97      1028
           8       0.96      0.96      0.96       974
           9       0.96      0.95      0.95      1009

    accuracy                           0.97     10000
   macro avg       0.97      0.97      0.97     10000
weighted avg       0.97      0.97      0.97     10000
```

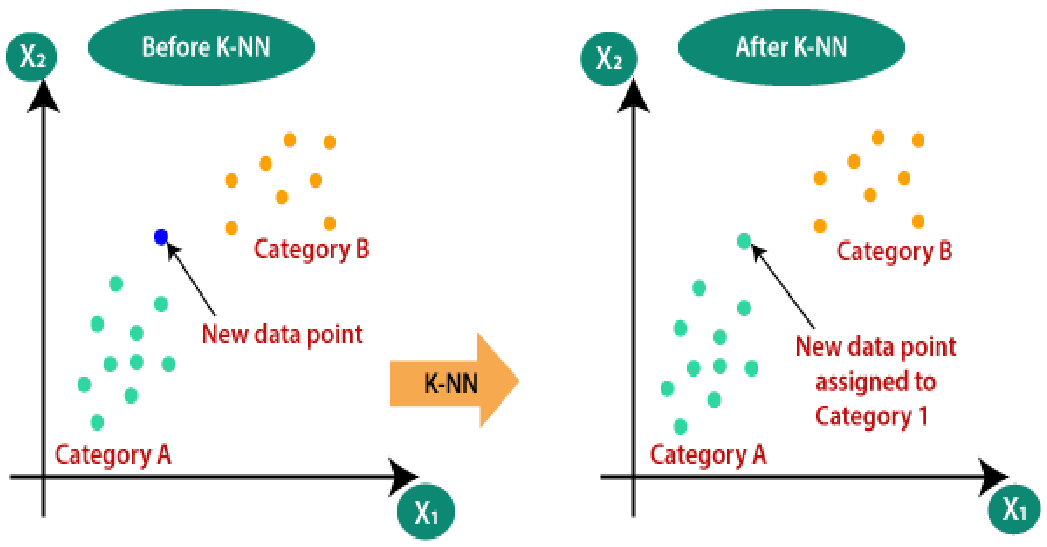
8) Machine Learning Models Tried In the Project

8.1: K Nearest Neighbour Model(KNN):

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much like the new data.

Accuracy in our project : 0.94

Example: Suppose, we have an image of a creature that looks like cat and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs' images and based on the most similar features it will put it in either cat or dog category.



8.2: Perceptron Model:

Perceptron is Machine Learning algorithm for supervised learning of various binary classification tasks. Further, Perceptron is also understood as an Artificial Neuron or neural network unit that helps to detect certain input data computations in business intelligence . Perceptron model is also treated as one of the best and simplest types of Artificial Neural networks. However, it is a supervised learning algorithm of binary classifiers.

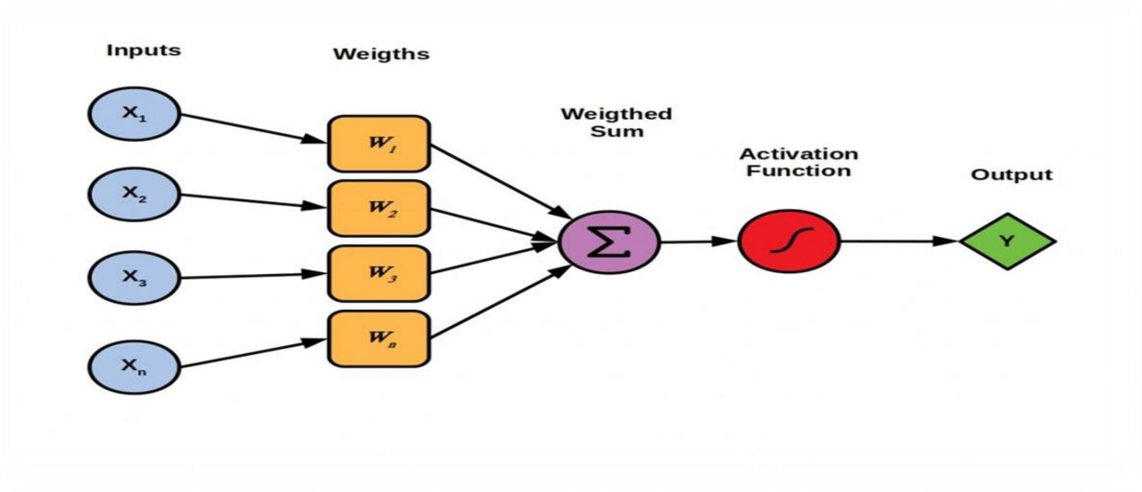
Accuracy in our project : 0.89

Basic Components of Perceptron Model:

Input Nodes or Input Layer: This is the primary component of Perceptron which accepts the initial data into the system for further processing. Each input node contains a real numerical value.

Wight and Bias: Weight parameter represents the strength of the connection between units. This is another most important parameter of Perceptron components. Weight is directly proportional to the strength of the associated input neuron in deciding the output. Further, Bias can be considered as the line of intercept in a linear equation.

Activation Function: These are the final and important components that help to determine whether the neuron will fire or not. Activation Function can be considered primarily as a step function. The data scientist uses the activation function to take a subjective decision based on various problem statements and forms the desired outputs. Activation function may differ (e.g., Sign, Step, and Sigmoid) in perceptron models by checking whether the learning process is slow or has vanishing or exploding gradients.

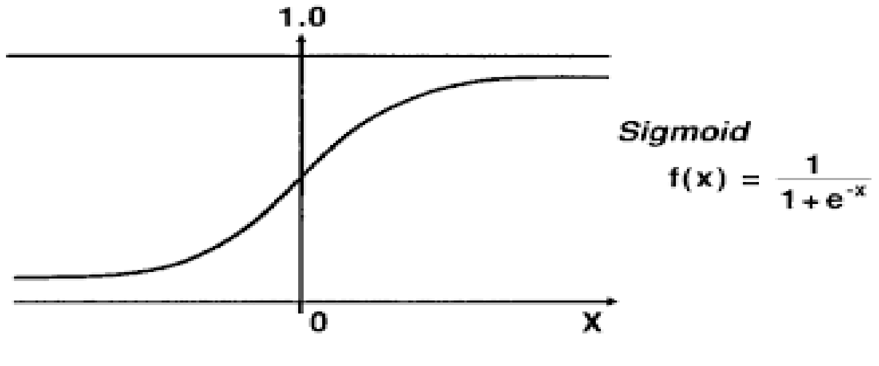


8.3: Logistic Regression Model:

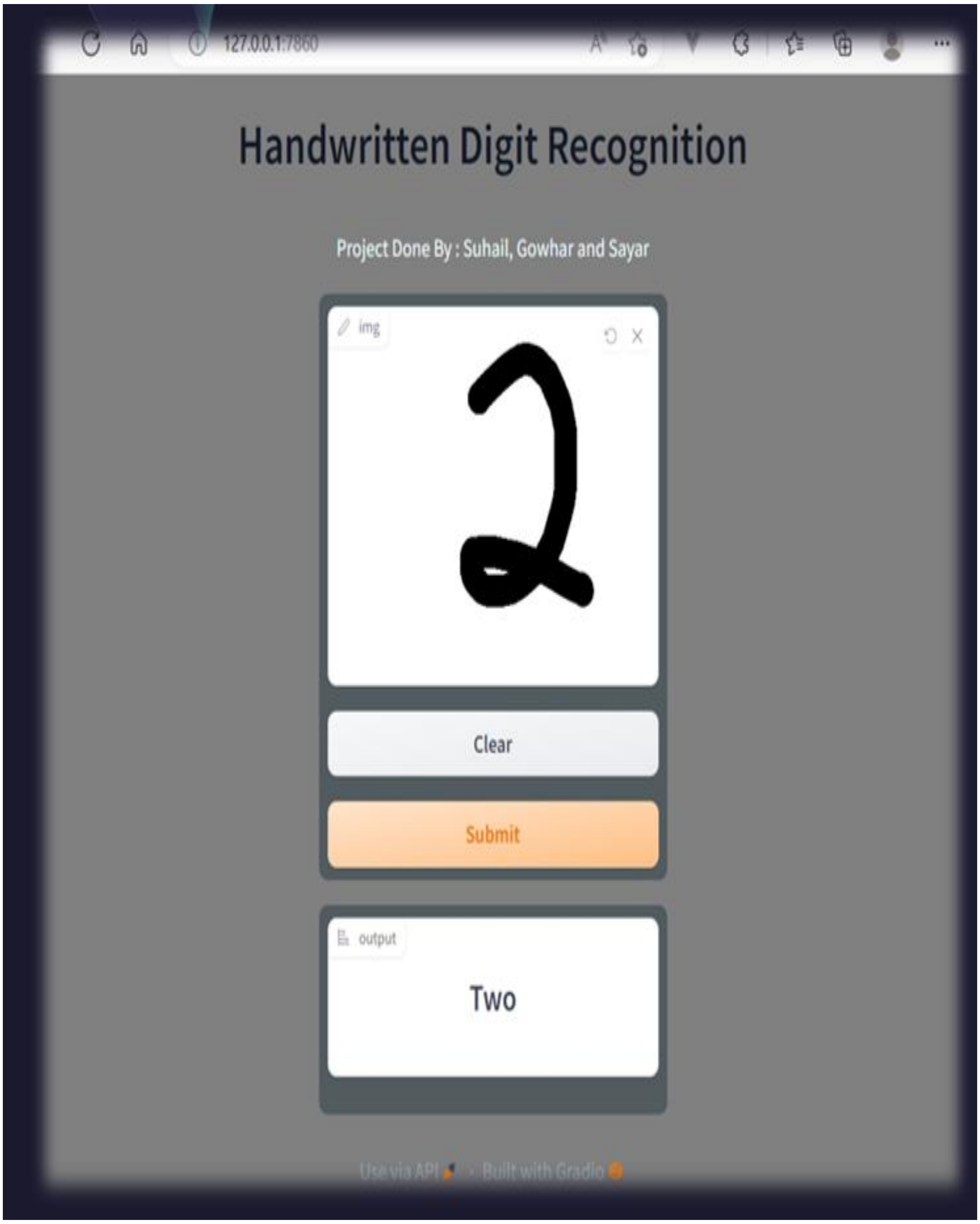
Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1 .Logistic Regression is much like the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm because it could provide probabilities and classify new data using continuous and discrete datasets

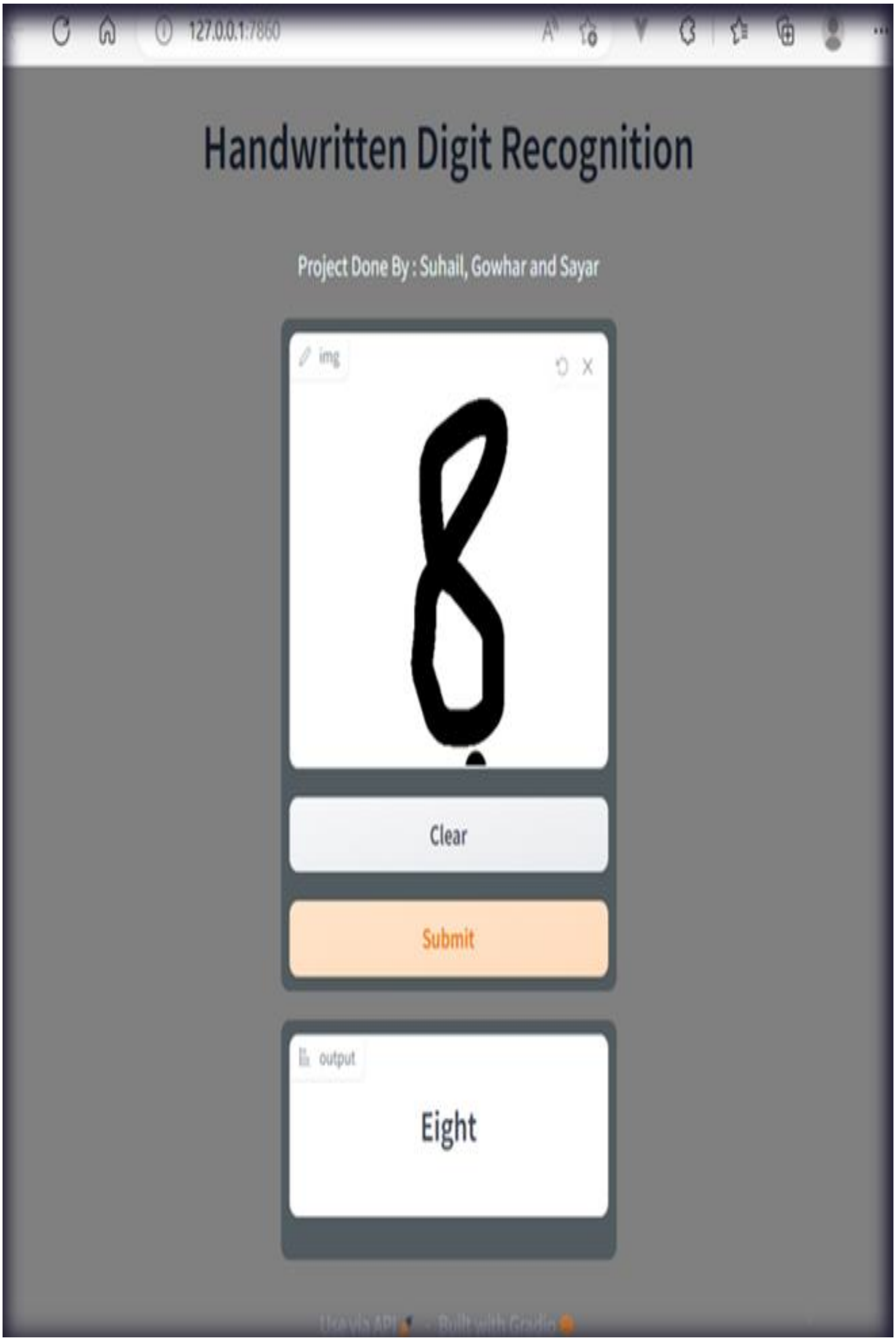
Accuracy in our project : 0.92

Sigmoid Function: The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1.The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function. In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.



9) Screenshots and examples of GUI:


127.0.0.1:7860
Handwritten Digit Recognition
Project Done By : Suhail, Gowhar and Sayar
img
Clear
Submit
output
Two
Use via API · Built with Gradio



Handwritten Digit Recognition

Project Done By : Suhail, Gowhar and Sayar

Eight

10) Software Required In the Project:

1. Python 3.5 +
2. Gradio Interface
3. Scikit-Learn for Machine Learning
4. NumPy
5. Matplotlib
6. Random Forest Classifier
7. Perceptron
8. Logistic Regression
9. KNN

11) References:

1. M. Wu and Z. Zhang, Handwritten Digit Classification using the MNIST Dataset, 2010.
2. Hamid, Norhidayu Abdul, and NilamBur Amir Sjarif,Handwritten recognition using SVM, KNN and neural network, arXiv preprint arXiv:1702.00723 (2017).
3. R.G.Mihalyi, Handwritten digit classification using support vector machines, 2011.
4. Gaurav Jain, Jason Ko, Handwritten DigitsRecognition, Project Report, University of Toronto, 11/21/2008.
5. Javatpoint – Classification Algorithms.
6. Machine Learning Techniques – IIT Madras.