Handwritten Digit Recognition Using the MNIST Dataset

PRESENTED BY: PADMA VASUDEVAN

27/01/2025

AGENDA

Introduction
Problem Statement
Solution Approach
Dataset Overview
Exploratory Data Analysis
Modeling & Performance

INTRODUCTION

Used the MNIST dataset to recognize handwritten digits in medical records because it serves as a benchmark dataset for digit classification tasks. MNIST dataset is clean, preprocessed and labeled making it easier to develop models.

Problem Statement

Objective:

The goal of this project is to create a smart system that can read and understand handwritten medical records, saving time and reducing mistakes caused by manual work. By using advanced computer algorithms like Naïve Bayes and K-Nearest Neighbours (KNN), the system will quickly and accurately turn handwritten data into digital format, making it easier for hospitals and clinics to manage large amounts of information and provide better care.

Solution Approach

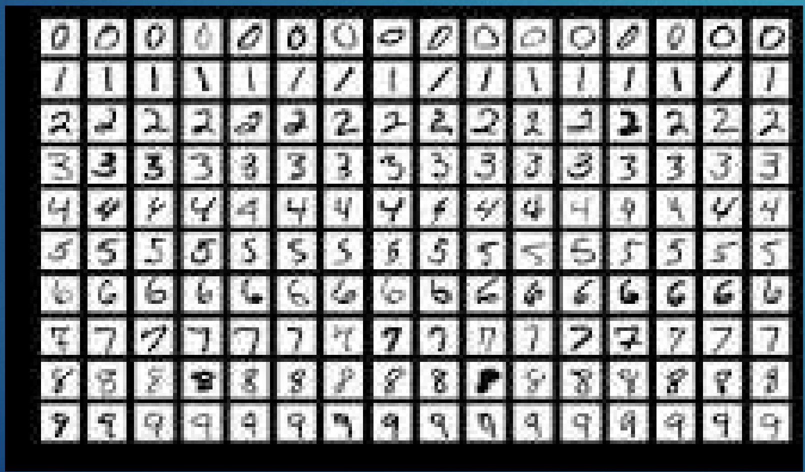
Intelligent Data-Driven Recognition: Leveraging machine learning for handwritten digit recognition.

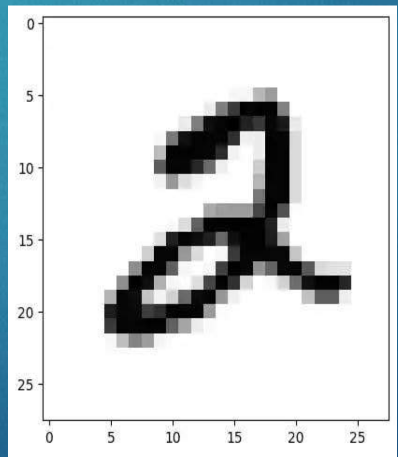
Key Algorithms:

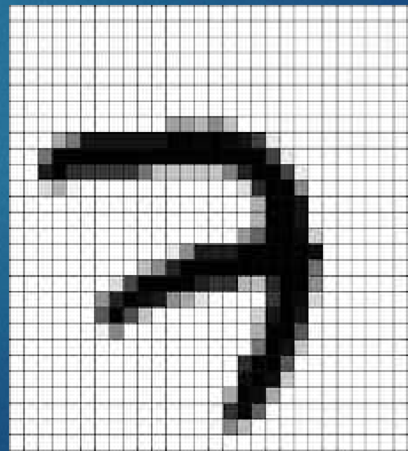
- Naïve Bayes Classifier
- Non-Naïve Bayes Classifier
- K-Nearest Neighbours (KNN) Classifier

Dataset Overview

- Description of the MNIST dataset:
  - Training dataset: 60,000 images with 784 pixels
  - Testing dataset: 10,000 images with 784 pixels
- Data format:
  - 28x28 pixel grayscale images (Intensity 0 to 255)
  - 10 classes (0 to 9)
- Visual examples (Intensity inverted images):

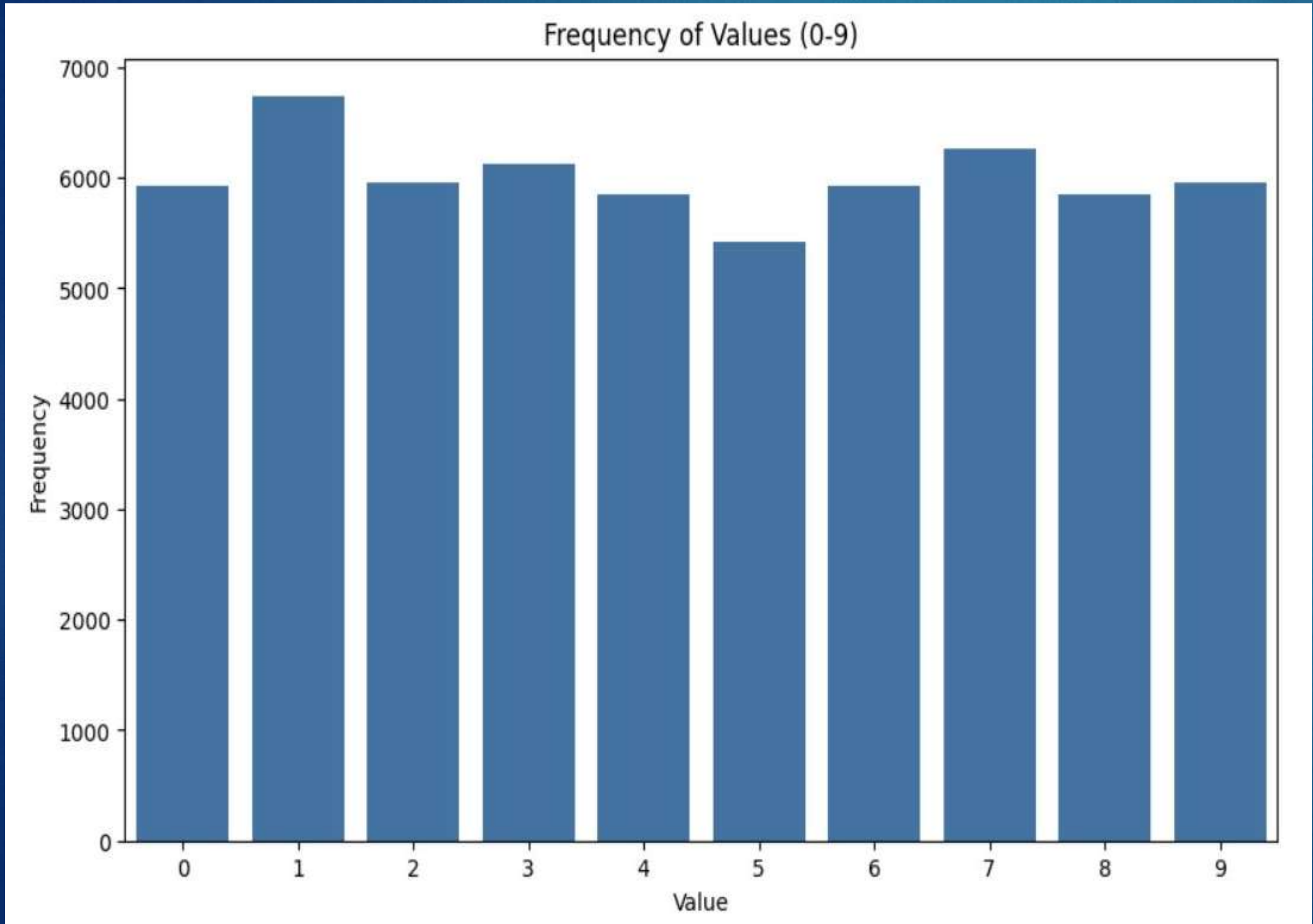






Exploratory data analysis

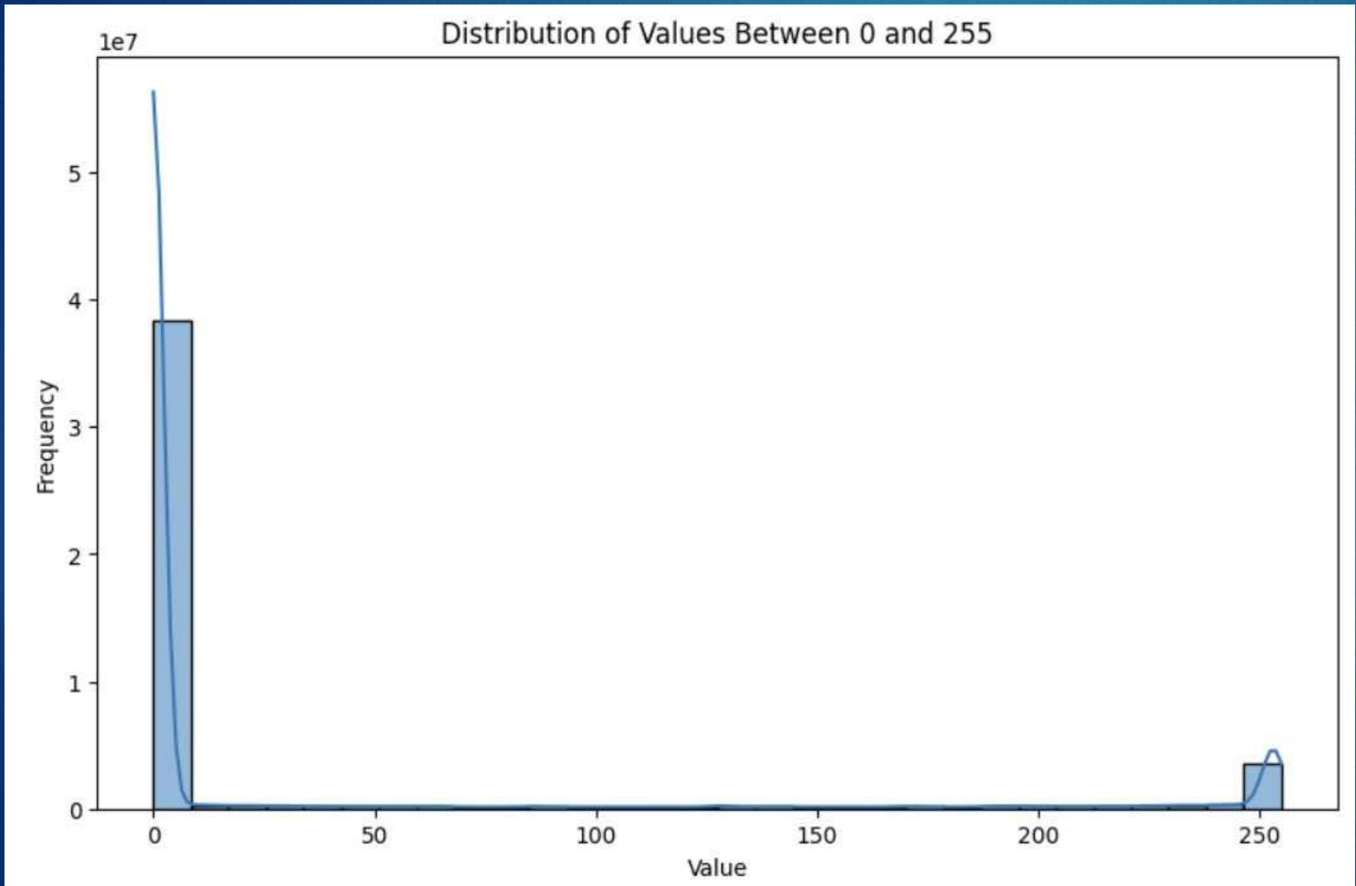
Distribution of digits from 0 to 9



- This bar plot shows the frequency distribution of handwritten digit classes (0–9).
- The dataset is **balanced**, with each digit having nearly the same number of samples. This ensures that the model is not biased toward any particular class.

Exploratory data analysis

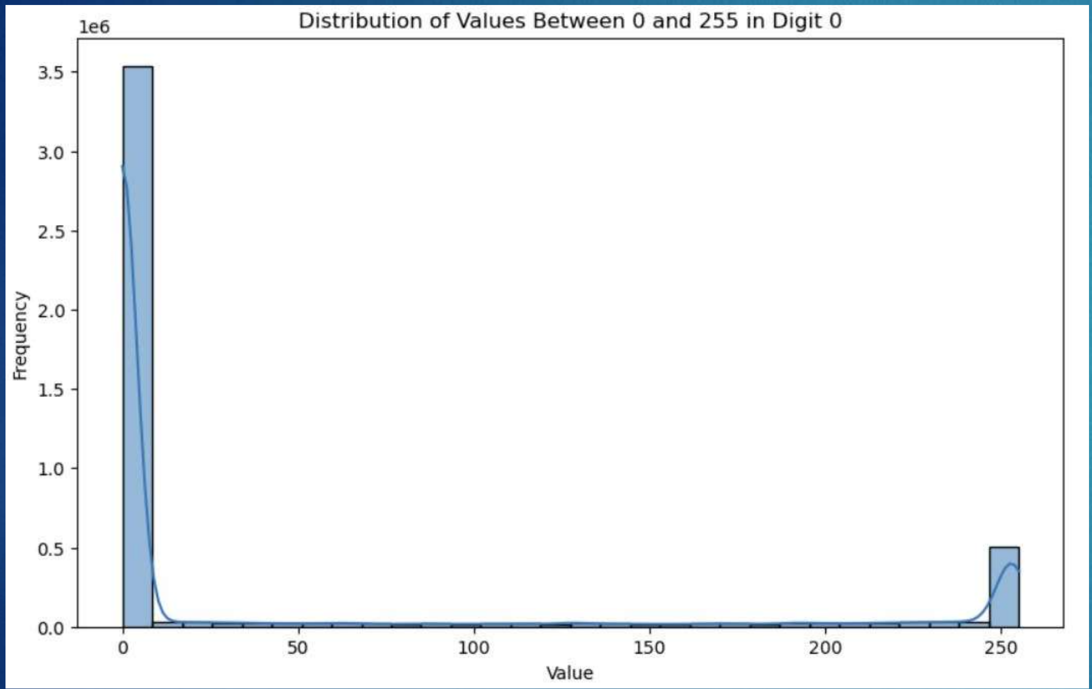
Distribution of pixel intensities from 0 to 255



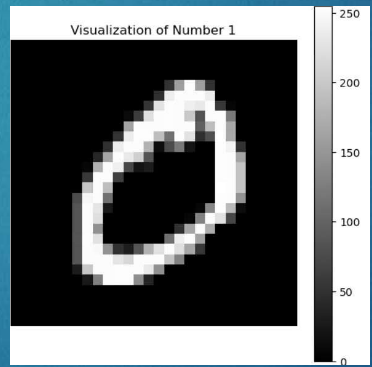
- This histogram shows the frequency distribution of pixel intensities.
- Majority of the pixels are dark or 0 intensity.
- Greater than 90% of intensity values are around 0 (black), and less than 10% intensity values are around 255 (white).

Exploratory data analysis

Distribution of pixel intensities from 0 to 255 in Digit 0



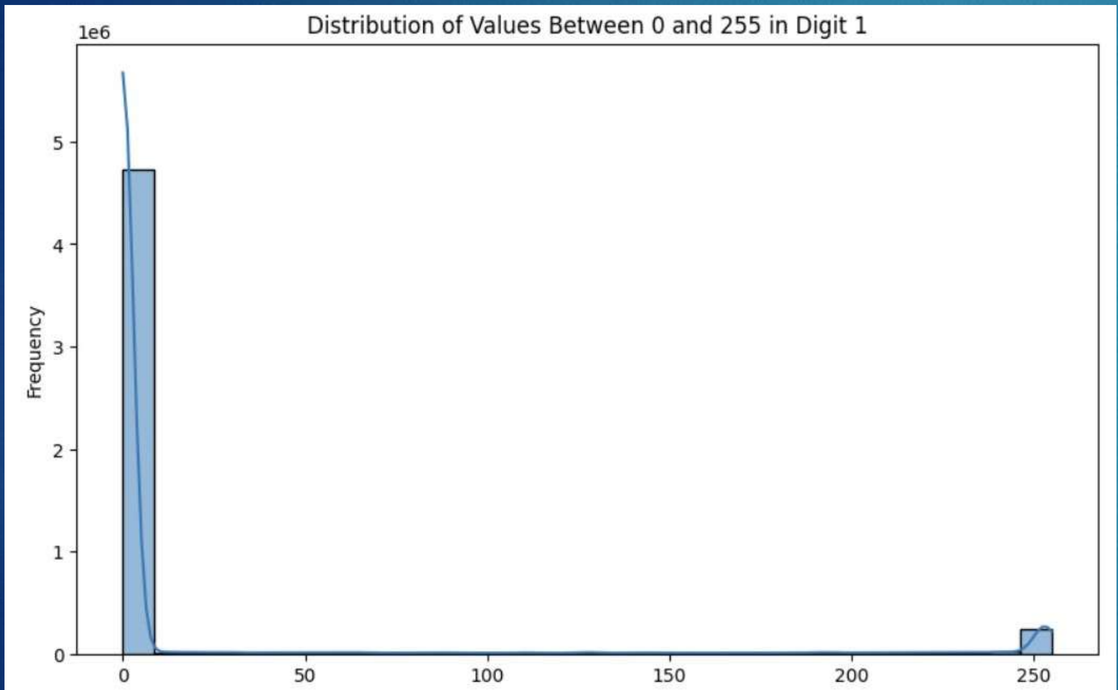
- This histogram shows the frequency distribution of pixel intensities in Digit 0.
- For all digit 0 records (images)
  No. of 0s : 3506565
  No. of 255s: 41752



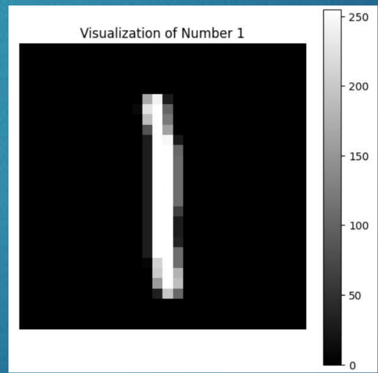
This image shows that the digit 0 occupies comparatively more white pixels in the grid (than Digit 1).

Exploratory data analysis

Distribution of pixel intensities from 0 to 255 in Digit 1



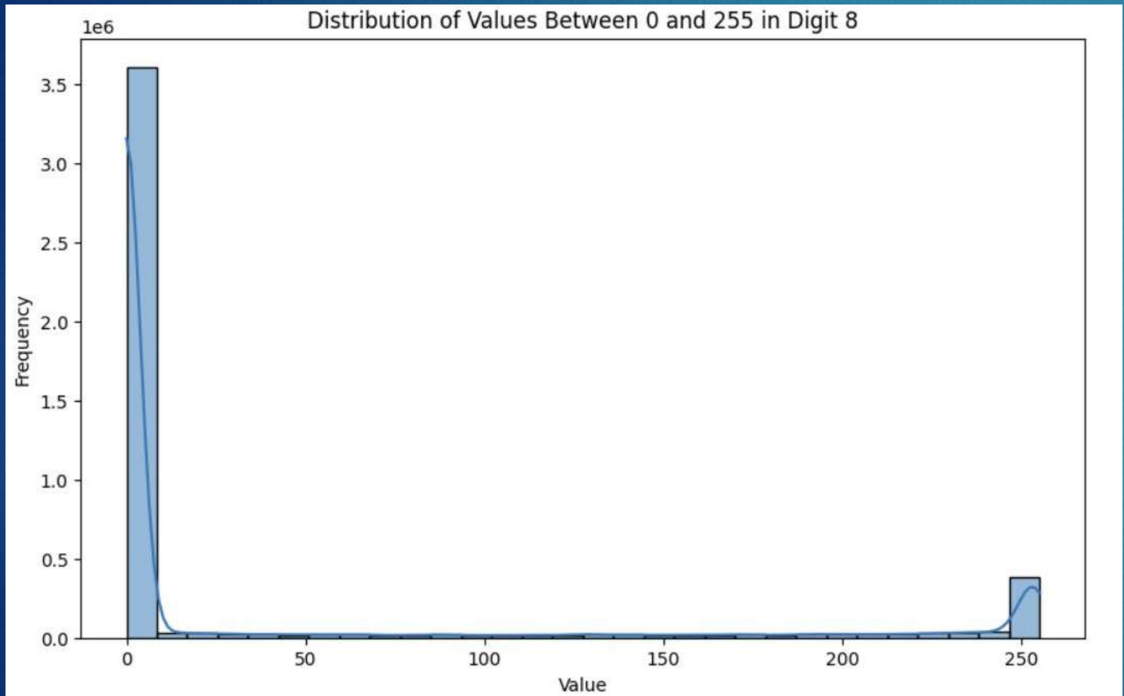
- This histogram shows the frequency distribution of pixel intensities in Digit 1.
- For all digit 1 records (images)
  No. of 0s : 4706954
  No. of 255s: 24019



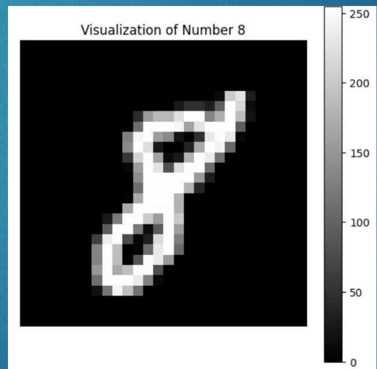
This image shows that the digit 1 occupies comparatively less white pixels in the grid (than Digit 0).

Exploratory data analysis

Distribution of pixel intensities from 0 to 255 in Digit 8



- This histogram shows the frequency distribution of pixel intensities in Digit 8.
- For all digit 8 records (images)
  No. of 0s : 357311
  No. of 255s: 32260



This image shows that the digit 8 occupies comparatively more white pixels in the grid (than Digit 1).

Naive Bayes Classifier

- Naive Bayes is a simple yet powerful classification algorithm based on Bayes' Theorem.
- It assumes that features are **independent** (hence "naive") and contribute equally to the outcome.
- **Bayes' Theorem**:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

P(A|B): Probability of class A given data B (posterior).
P(B|A): Probability of data B given class A (likelihood).
P(A): Prior probability of class A.
P(B): Evidence (overall probability of data B).

After tuning the hyperparameter (epsilon = 1e-2) and normalizing the X_train data:

- **Training Data Accuracy:** 80%
- **Test Data Accuracy:** 81%

Non-Naive Bayes Classifier

- **A Non-Naive Bayes algorithm removes the assumption of independence. It tries to model relationships between features (like how nearby pixels in an image are related) to make better predictions.**
- **Instead of assuming features are independent, Non-Naive Bayes considers how features interact with each other.**

After tuning the hyperparameter (epsilon = 5e-2) and normalizing the X_train data:

- **Training Data Accuracy:** 95%
- **Test Data Accuracy:** 94%

K-Nearest Neighbours Classifier

- K-Nearest Neighbours (KNN) is a simple yet powerful machine learning algorithm used for classification tasks. It works by assigning a class to a data point based on the majority class of its nearest neighbours in the feature space.
- The KNN algorithm does not explicitly "train" the model like other algorithms (e.g., decision trees or neural networks). Instead, it memorizes the training data.
- When a new test image is input for classification, KNN finds the "K" nearest neighbors of the test image from the training data.
- It calculates the distance between the test image and all training images using a distance metric, usually Euclidean distance.
- Formula for Euclidean distance:

$$distance = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2}$$

where x_i and y_i are the pixel values of the test image and training image, respectively.

Summary

For creating a smart system for recognizing the handwritten digits with large dataset, I choose Non-Naive Bayes as it is very simple and fast with proper tuning the epsilon value, I received 94% accuracy in test data.

Thank you!