**Phase-2 Submission Template**

**Student Name:** Monisha. P

**Register Number:** 613023104063

**Institution:** Vivekanandha College of Technology for Women

**Department:** BE - Computer Science and Engineering

**Date of Submission:** 01.05.2025

**Github Repository Link:** https://github.com/Monisha200627/Digit-AI.git

# 1. Problem Statement

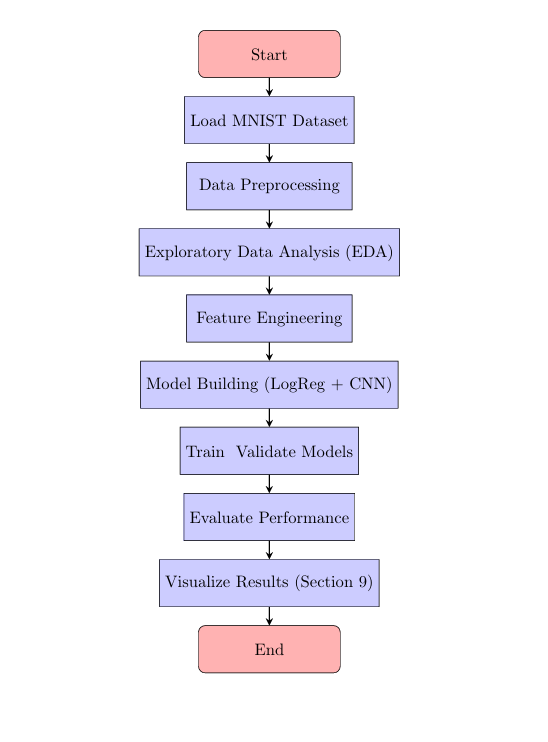
* Recognizing handwritten digits is a critical task in AI systems with real-world applications such as postal code reading, bank check processing, and digital form entry.

* Traditional methods fail due to handwriting variability..
* This project tackles the classification problem using deep learning, aiming for accurate, scalable, and robust recognition across various handwriting styles.

# 2. Project Objectives

* Build an AI model that can accurately classify digits (0–9) from handwritten images.
* Achieve high accuracy using deep learning models such as neural networks (e.g., CNNs).
* Develop a user-ready model applicable to real-world tasks like postal and banking automation.

**3. Flowchart of the Project Workflow**



# 4. Data Description

* Dataset&source: MNIST Handwritten Digits &kaggle
* Type of data: Image, structured (28x28 grayscale)
* Records: 70,000 total (60K train, 10K test)
* Static or dynamic: Static
* Target variable : Digit class (0 to 9)

# 5. Data Preprocessing

* Normalized pixel values to [0, 1] range.
* Converted labels to one-hot encoding (for classification).
* Checked for nulls (none found in MNIST).
* No duplicates due to standardized dataset.
* Image reshaped for CNN input where needed (e.g., 28x28x1).

# 6. Exploratory Data Analysis (EDA)

* Univariate Analysis:

* 1. Countplot of digit distribution (0-9) shows balanced data. Visual samples of each digit using matplotlib.
* Bivariate/Multivariate Analysis:

* 1. Mean and variance of pixel intensities analyzed.

○ Heatmaps used to show pixel intensity patterns across digits.

* Insights Summary:

Some digits have higher visual similarity (e.g., 3 and 5), which may cause misclassification.

○ Pixel intensity helps the model understand the shape of each digit.Edges and corners are important to tell digits apart, like 1 vs7.Middle part of the image is most useful since digits are usually centered.Unique patterns, like curves or straight lines, help the model learn differences (e.g., between 3 and 8).

# 7. Feature Engineering

* Used pixel values directly as features.

* Applied reshaping for CNN models.
* Tried PCA for dimensionality reduction as optional enhancement.

# 8. Model Building

* Models used:

1. Baseline: Logistic Regression
   1. Advanced: Convolutional Neural Network (CNN)

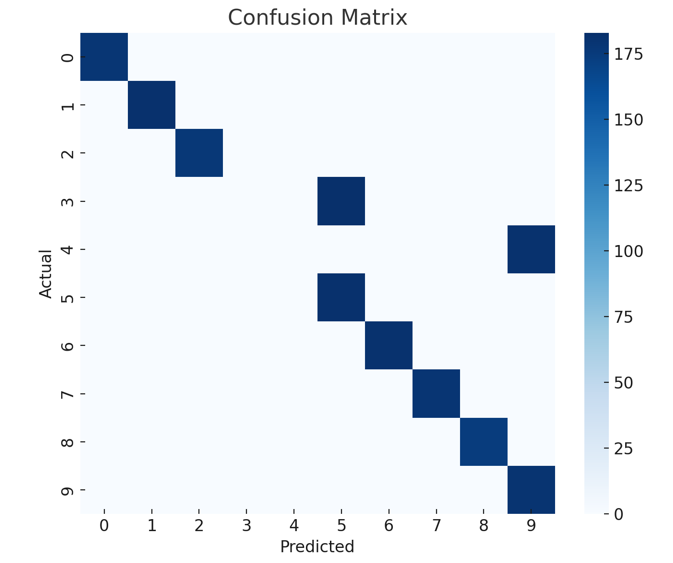
* Model Justification:

1. Logistic regression to establish a simple baseline.
2. CNNs for spatial feature extraction—ideal for image tasks.

* Evaluation Metrics: Accuracy, Precision, Recall, F1-score
* Best Accuracy Achieved: ~98% using CNN

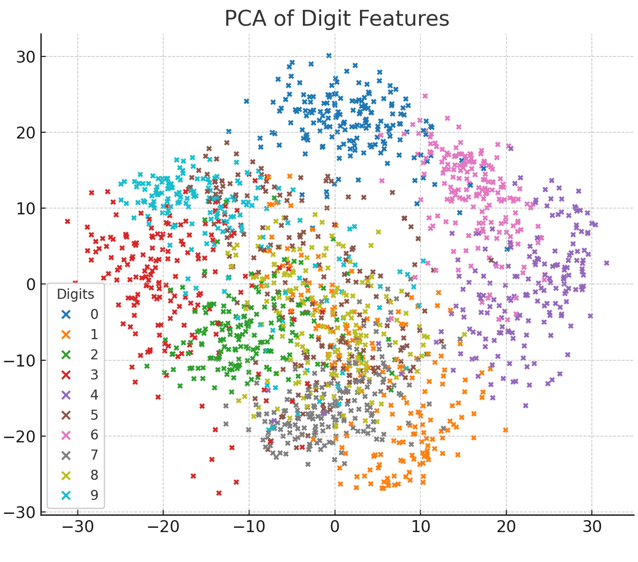
# 9. Visualization of Results & Model Insights

* Confusion Matrix: Showed misclassifications, mostly in 4/9, 3/5.



* Accuracy & Loss Curves: Used to monitor overfitting.

* Feature Importance (for non-CNN): Used PCA to visualize key components.



# 10. Tools and Technologies Used

* Programming Language: Python
* IDE : Google Colab
* Libraries: NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, TensorFlow/Keras
* Visualization Tools: Matplotlib, Seaborn

# 11. Team Members and Contributions

● Dharani.R

○ Data cleaning ( Load and clean data Normalize and split into train/test

sets)

* Karthika.G

○ EDA (Create and train the machine learning model (e.g., CNN)

Tune hyperparameters)

* Harshitha shree.A.S

○ Feature engineer(Check model accuracyPlot confusion matrix, accuracy

&loss graphs)

* Lavanya.R

○ Model development( Apply PCA & Visualize and analyse feature clusters)

* Monisha.P

○ Documentation and reporting( Write project report Prepare slides and visuals