Handwritten Digit Recognition: Mid-Progress Report

Group Members:

Srikar Adepu
Venkata Venu
Katta Srijith
Arjun Naik
Keshamoni Adwith
Kumavath Tanishq

1. Dataset and Problem Statement

The objective of this project is to develop a machine learning model for recognizing handwritten digits (0-9) using the MNIST dataset. The dataset contains 70,000 grayscale images, with 60,000 images for training and 10,000 for testing. Each image is 28x28 pixels, and our task is to train a model that can accurately classify each digit based on pixel values.

2. Progress So Far

Data Collection and Preprocessing

1. Images of digits (0-9) were collected and converted into an array format.
2. Pixel values were binarized: 0 for values between 0-100 and 1 for values between 100-255.
3. The dataset was stored in CSV format and shuffled for better training.

Model Training

1. Implemented **Linear Regression**, but it showed poor classification accuracy.
2. Evaluating other models like **K-Nearest Neighbors (KNN), Decision Tree, and Bayesian Classifier.**
3. Implemented **KNN**, which provided better accuracy compared to Linear Regression.

Live Prediction

1. A real-time digit recognition system was developed where users draw a digit, and the model predicts the number.

3. Challenges Encountered

1.Low Accuracy with Linear Regression – Struggles to capture complex patterns.

2.Data Noise – Requires better preprocessing techniques to improve clarity.

3.Overfitting in Decision Trees – Needs proper pruning to enhance generalization.

4.Computational Complexity of KNN – Requires optimization for large datasets.

5.High Memory Usage in KNN – Storing and computing distances for large datasets is resource-intensive.

6.Slow Prediction Time in KNN – Classification requires calculating distances for all training points, making real-time predictions slower.

7.Choice of Distance Metric in KNN – Different distance metrics (Euclidean, Manhattan, etc.) can impact classification accuracy.

8.Curse of Dimensionality in KNN – As the number of features increases, KNN performance can degrade due to sparse data distribution.

4. Remaining Tasks

1.Implement and Optimize **KNN, Bayesian Classifier, and Decision Tree Models** for better accuracy.

2.Enhance **Preprocessing** with noise reduction and data augmentation.

3.Fine-Tune **Hyperparameters** to improve model performance.

4.Evaluate Models using **accuracy, precision, recall, and F1-score.**

5.Develop a **User Interface** for real-time digit recognition and deployment.