

#KNN – Handwritten Digit Classification

##Project Title

Handwritten Digit Recognition using K-Nearest Neighbors (KNN)

##Project Description

This project demonstrates how the K-Nearest Neighbors (KNN) algorithm can be used to classify handwritten digits (0–9). The built-in Digits dataset from Scikit-learn is used, where each digit image is represented as pixel intensity values. The task highlights distance-based learning, feature scaling, and the impact of different K values on model accuracy.

##Objectives

- Load and explore the handwritten digits dataset
- Visualize digit images and their labels
- Split data into training and testing sets
- Apply feature scaling for distance-based learning
- Train a KNN classifier
- Tune the value of K to improve accuracy
- Evaluate the model using accuracy and confusion matrix

##Dataset Used

Digits Dataset (Scikit-learn)

- Images of handwritten digits (0–9)
- Each image is of size 8×8 pixels
- Each image is flattened into 64 numerical features

Target Variable:

- Digit label (0–9)

##Tools & Libraries Used

- Python
- NumPy
- Scikit-learn
- Matplotlib
- Seaborn
- Jupyter Notebook

##Steps Performed

1. Loaded the digits dataset using Scikit-learn
2. Explored the dataset structure and dimensions
3. Visualized sample digit images
4. Split the dataset into training and testing sets (80/20)
5. Applied feature scaling using StandardScaler
6. Trained a KNN classifier with initial K value

7. Tested multiple K values to find optimal performance
8. Evaluated model accuracy for different K values
9. Generated confusion matrix for prediction analysis
10. Visualized predicted results on test images

##Evaluation Metrics

- **Accuracy** – Percentage of correctly classified digits
- **Confusion Matrix** – Detailed view of correct and incorrect predictions

##Key Learnings

- KNN is a distance-based classification algorithm
- Feature scaling is essential for KNN performance
- Choosing the right value of K improves accuracy
- Smaller K may cause overfitting, larger K may underfit
- KNN works well for small to medium-sized datasets

##Files in this Repository

- KNN_Digit_Classification.ipynb – Jupyter Notebook
- README.md – Project documentation

##How to Run

1. Clone this repository
2. Open the Jupyter Notebook
3. Run all cells step by step
4. Observe accuracy, plots, and predictions

##Conclusion

This task demonstrates the practical application of the KNN algorithm for image classification. By tuning the value of K and applying feature scaling, the model achieves high accuracy in recognizing handwritten digits.