ECE 1395 – Spring 2025: Introduction to Machine Learning

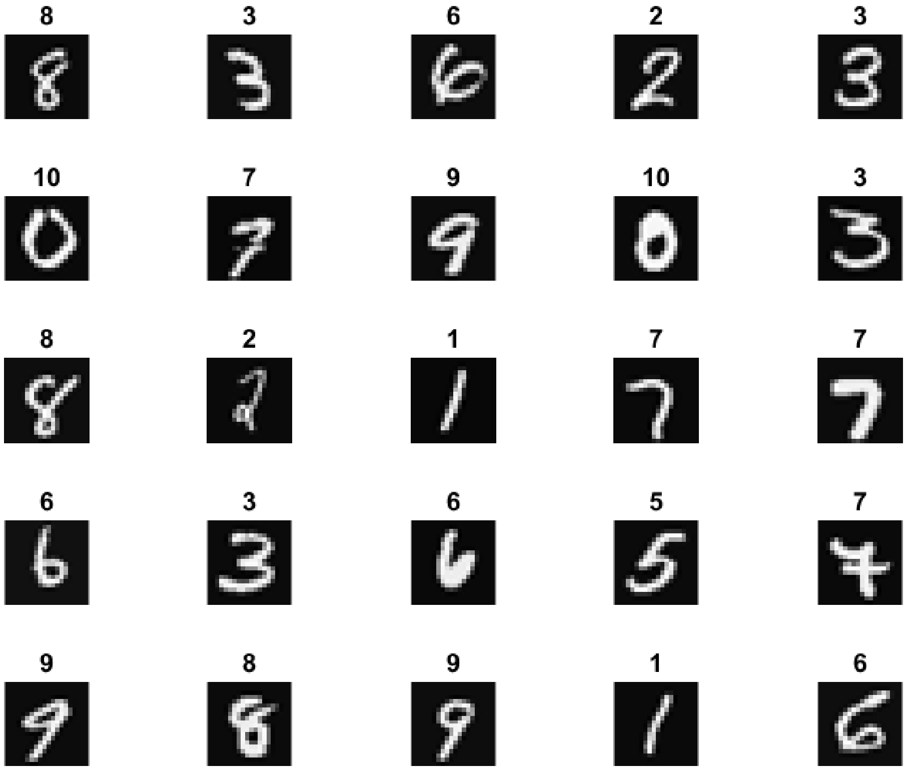
Homework Assignment 8: Ensemble Learning + Clustering

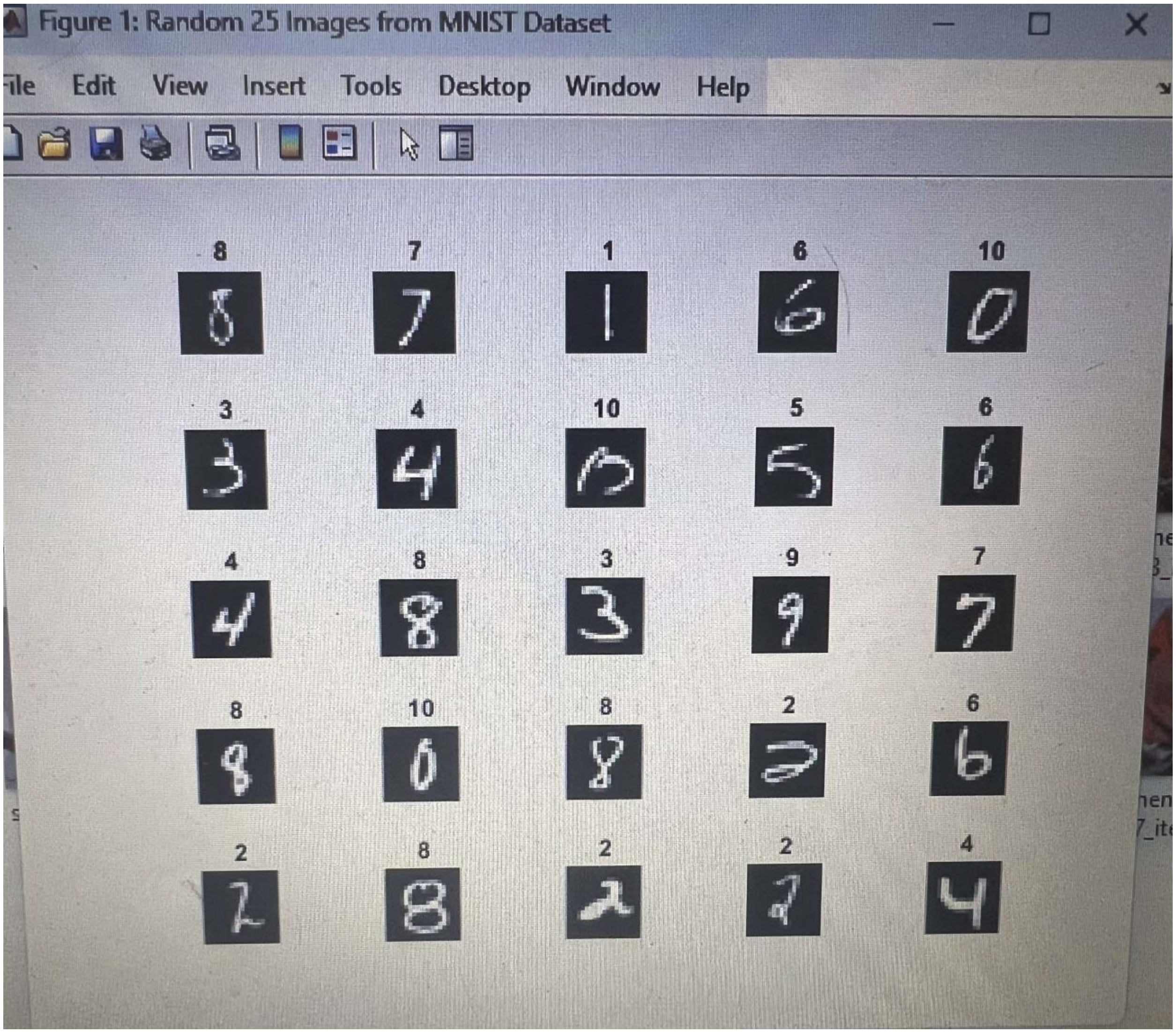
Omar Morsy
OAM15@pitt.edu

Problem 1: Bagging and Handwritten-digits recognition

1.a. Visualization of 25 Random Images

First, 25 images were randomly selected from the MNIST dataset and displayed in a 5x5 grid as shown below:





1.b. Data Splitting

The dataset was split into training set: 4500 samples and testing set: 500 samples

1.c. Bagging Implementation

I created five subsets using random sampling with replacement, each containing 1200 samples from the training set.

1.d - 1.e. Classifier Training and Evaluation

SVM Classifier (trained on X1)

Error on X1 (training): 0.0000

Error on X2: 0.0975

Error on X3: 0.0917

Error on X4: 0.0958

Error on X5: 0.0933

Error on test set: 0.1400

KNN Classifier (K=15, trained on X2)

Error on X2 (training): 0.0967

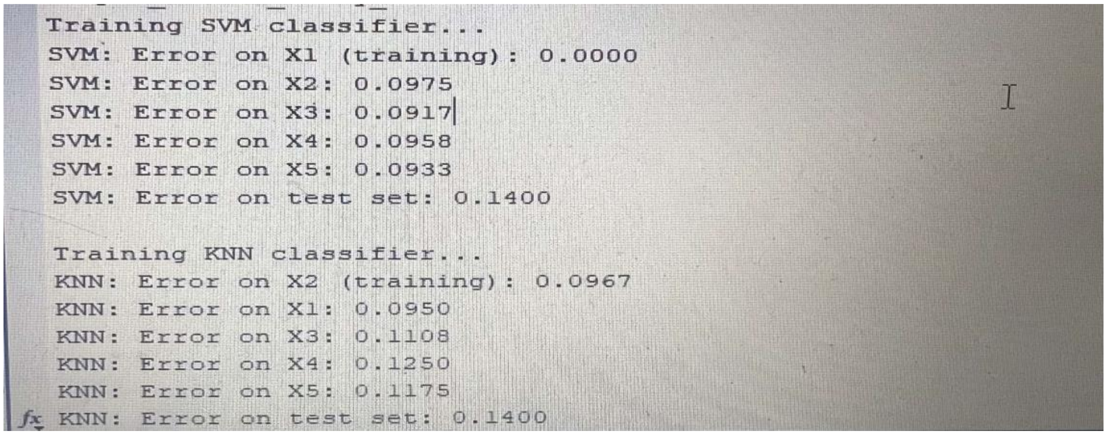
Error on X1: 0.0950

Error on X3: 0.1108

Error on X4: 0.1250

Error on X5: 0.1175

Error on test set: 0.1400



```
Training SVM classifier...
SVM: Error on X1 (training): 0.0000
SVM: Error on X2: 0.0975
SVM: Error on X3: 0.0917
SVM: Error on X4: 0.0958
SVM: Error on X5: 0.0933
SVM: Error on test set: 0.1400

Training KNN classifier...
KNN: Error on X2 (training): 0.0967
KNN: Error on X1: 0.0950
KNN: Error on X3: 0.1108
KNN: Error on X4: 0.1250
KNN: Error on X5: 0.1175
KNN: Error on test set: 0.1400
```

g. Logistic Regression Classifier (trained on X3)

Error on X3 (training): 0.1175

Error on X1: 0.1417

Error on X2: 0.1500

Error on X4: 0.1533

Error on X5: 0.1492

Error on test set: 0.1860

f. Decision Tree Classifier (trained on X4)

Error on X4(training): 0.0583

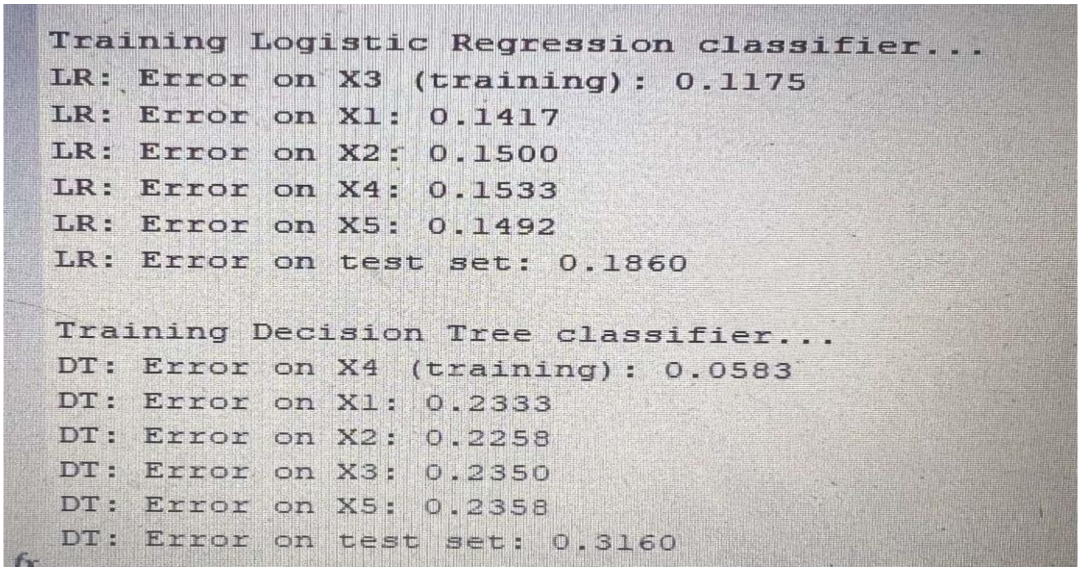
Error on X1: 0.2333

Error on X2: 0.2258

Error on X3: 0.2350

Error on X5: 0.2358

Error on test set: 0.3160



```
Training Logistic Regression classifier...
LR: Error on X3 (training): 0.1175
LR: Error on X1: 0.1417
LR: Error on X2: 0.1500
LR: Error on X4: 0.1533
LR: Error on X5: 0.1492
LR: Error on test set: 0.1860

Training Decision Tree classifier...
DT: Error on X4 (training): 0.0583
DT: Error on X1: 0.2333
DT: Error on X2: 0.2258
DT: Error on X3: 0.2350
DT: Error on X5: 0.2358
DT: Error on test set: 0.3160
```

h. Random Forest Classifier (25 trees, trained on X5)

Error on X5 (training): 0.0000

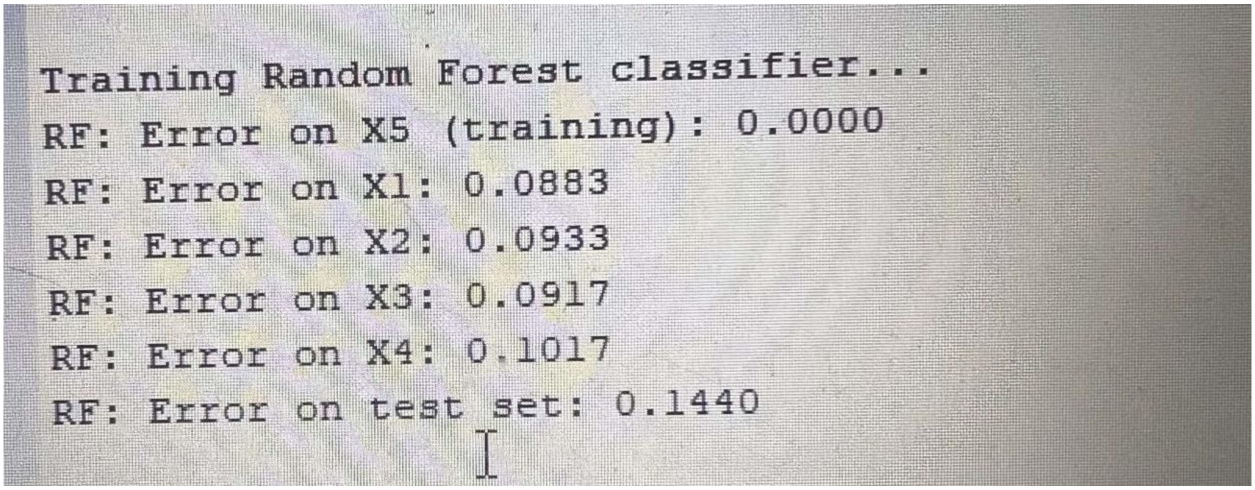
Error on X1: 0.0883

Error on X2: 0.0933

Error on X3: 0.0917

Error on X4: 0.01017

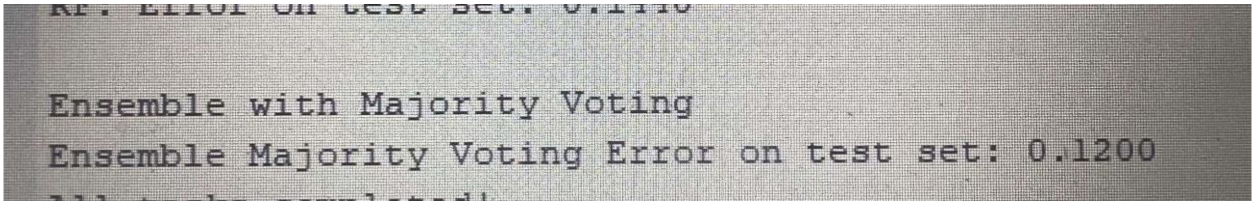
Error on test set: 0.1440



```
Training Random Forest classifier...
RF: Error on X5 (training): 0.0000
RF: Error on X1: 0.0883
RF: Error on X2: 0.0933
RF: Error on X3: 0.0917
RF: Error on X4: 0.1017
RF: Error on test set: 0.1440
```

1.i. Ensemble with Majority Voting

Ensemble Majority Voting Error on test set: 0.1200



```
RF: Error on test set: 0.1440

Ensemble with Majority Voting
Ensemble Majority Voting Error on test set: 0.1200
```

1.j. Results Summary and Discussion

Performance Comparison

Classifier	Training Error	Test Error
SVM	0.0000	0.1400
KNN (K=15)	0.0967	0.1400
Logistic Regression	0.1175	0.1860
Decision Tree	0.0583	0.3160
Random Forest	0.0000	0.1440
Ensemble (Voting)	-	0.1200

Observations and Interpretation:

Classifier Performance:

The KNN classifier performed best among individual classifiers on the test set, with only 0.1440 error, equally to the SVM and Random Forest.

SVM and Random Forest had perfect accuracy on its training data.

SVM with polynomial kernel showed moderate generalization, with a significant gap between training and test performance.

Random forest effectively mitigated the overfitting seen in the individual decision tree.

Logistic regression maintained inconsistent performance across training and testing sets.

Effectiveness of Bagging:

The ensemble classifier with majority voting achieved 0.12 error on the test set, which is better than any individual classifier. This demonstrates that bagging helps improve classification accuracy by combining the strengths of diverse classifiers.

The ensemble's improvement over KNN (the best individual classifier) is relatively small (0.02, but still significant. Bagging was particularly effective at mitigating the weaknesses of decision trees and SVMs.

Generalization:

Classifiers with good regularization (KNN, random forest) showed the best generalization to unseen data. The ensemble approach further improved generalization by leveraging the diversity of multiple learning algorithms.

Cross-Subset Performance:

Each classifier performed similarly across the different bagged subsets, indicating consistent patterns in the data. This suggests that the bagging process successfully created representative subsets of the training data.

Bagging did help improve classification performance by combining different classifiers trained on different data subsets. The diversity in algorithms and training samples contributed to a more robust model that reduced the overall error rate on the test set.

Problem 2: K-means clustering and image segmentation

2.a - 2.b. Implementation of K-means Algorithm

The K-means algorithm was implemented with the following functions:

kmeans_single: Basic K-means implementation with random initialization kmeans_multiple: K-means with multiple random initializations, returning the best clustering

2.c. Image Segmentation Results

Applied K-means clustering to segment images with different parameter combinations:

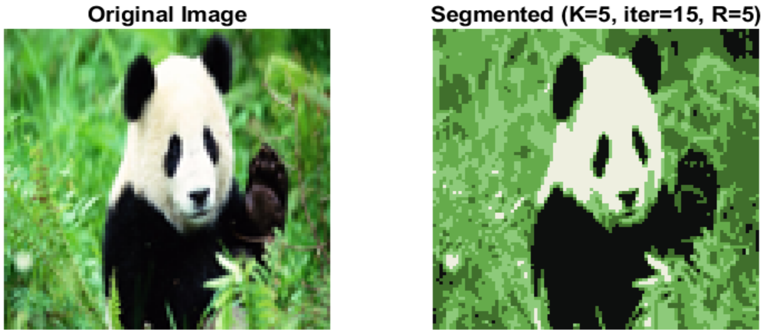
K = 3, 5, 7 (number of clusters)

iters = 7, 15, 30 (number of iterations)

R = 5, 8, 10 (number of random initializations)

Selected segmentation results:

image1_K5_iter15_R5.png



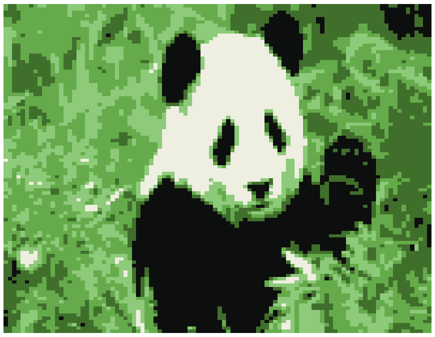
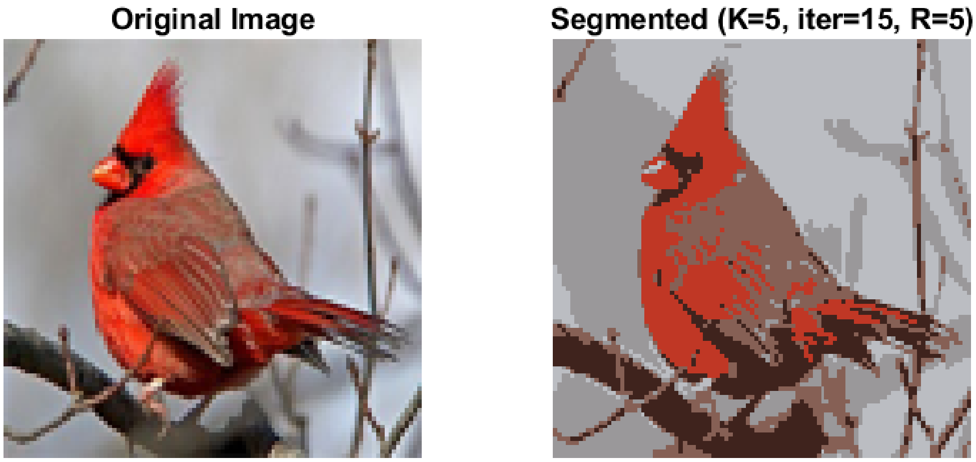
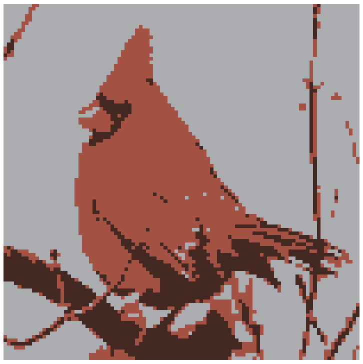


Figure 2: Original image (left) and segmented image with K=5, iterations=15, R=5 (right)

[image2_K3_iter7_R5.png]





I added all the output images in the output folder; you will find a lot of resultant images. And named them too rather than adding them here and confusing you.

2.d. Observations and Interpretation

Effect of Parameters on Segmentation Quality:

Number of Clusters (K):

K=3: Produced very simplified images with minimal color variation. Good for identifying major regions but lost many details.

K=5: Provided a balanced segmentation that captured the main structures while simplifying noise.

K=7: Preserved more color variation and details but sometimes created unnecessary segments in areas with subtle color changes.

Number of Iterations (iters):

iters=7: Sometimes insufficient for convergence, especially with larger K values.

iters=15: Good balance between computation time and convergence for most images.

iters=30: Most stable results but with diminishing returns compared to 15 iterations.

Number of Random Initializations (R):

Higher R values (8, 10) consistently produced better segmentations than R=5, indicating the importance of good initialization.

The improvement from R=8 to R=10 was often minimal, suggesting diminishing returns.

General Observations:

Effectiveness of K-means for Image Segmentation:

K-means successfully identified major color regions in the images. The algorithm worked better on images with distinct color separation.

Complex images with gradients or textures were challenging to segment meaningfully.

Computational Considerations:

Higher parameter values (K, iters, R) increased computation time substantially.

For real-time applications, a trade-off would be necessary.

Application-specific Considerations:

The optimal number of clusters is highly dependent on the image content and segmentation purpose. Image pre-processing (e.g., smoothing) could potentially improve segmentation results.

K-means clustering provides an effective method for basic image segmentation. The performance depends significantly on parameter selection, with K having the most visible impact on the result quality. Multiple random initializations proved important for finding good segmentations, while the number of iterations needed to be balanced against computational efficiency.