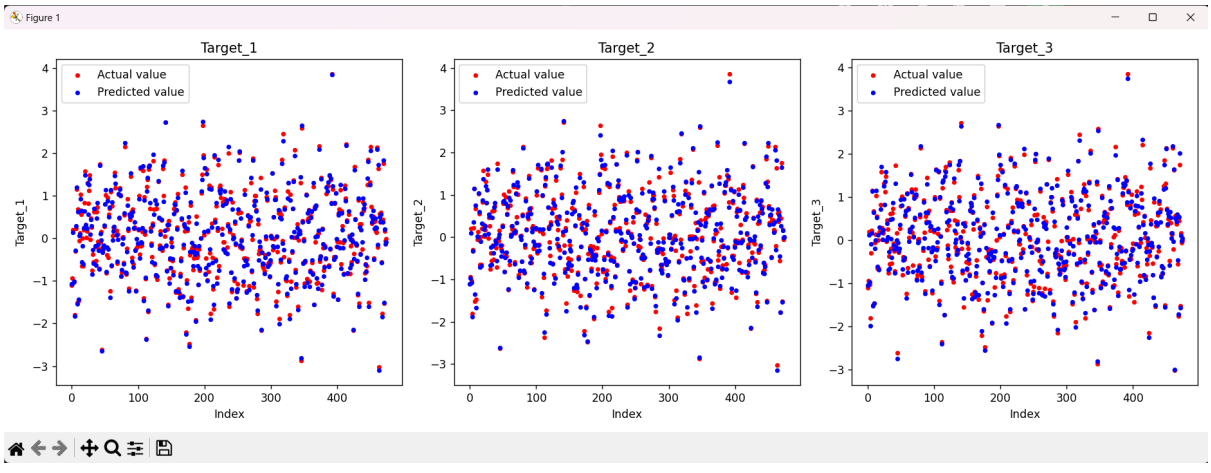
Task 1 and 2: Missing Value Imputation and Trainon Imputed Data

This study analyzed the impact of missing values on model performance using a synthetic regression dataset with five features and three target variables. Missing values were introduced in one feature and imputed using random and regression-based methods. A neural network was then trained and evaluated on the imputed datasets to assess the effects of imputation methods.

The performance, evaluated using Mean Squared Error (MSE), showed that the original dataset had the lowest test MSE (8.0632), while random imputation (1877.4565) and regression-based imputation (838.2659) resulted in higher errors. Contrary to expectations, regression-based imputation outperformed random imputation. This might be attributed to model-specific factors or data variability.

In conclusion, imputation methods significantly impact model accuracy, highlighting the need for careful handling of missing data during preprocessing.

```
Mean Squared Error (MSE): 5.204098554654681e-07
Datasets saved.
Mean Squared Errors (MSE):
Original Dataset: 8.0632
Random Imputed Dataset: 1877.4565
Regression Imputed Dataset: 838.2659
```

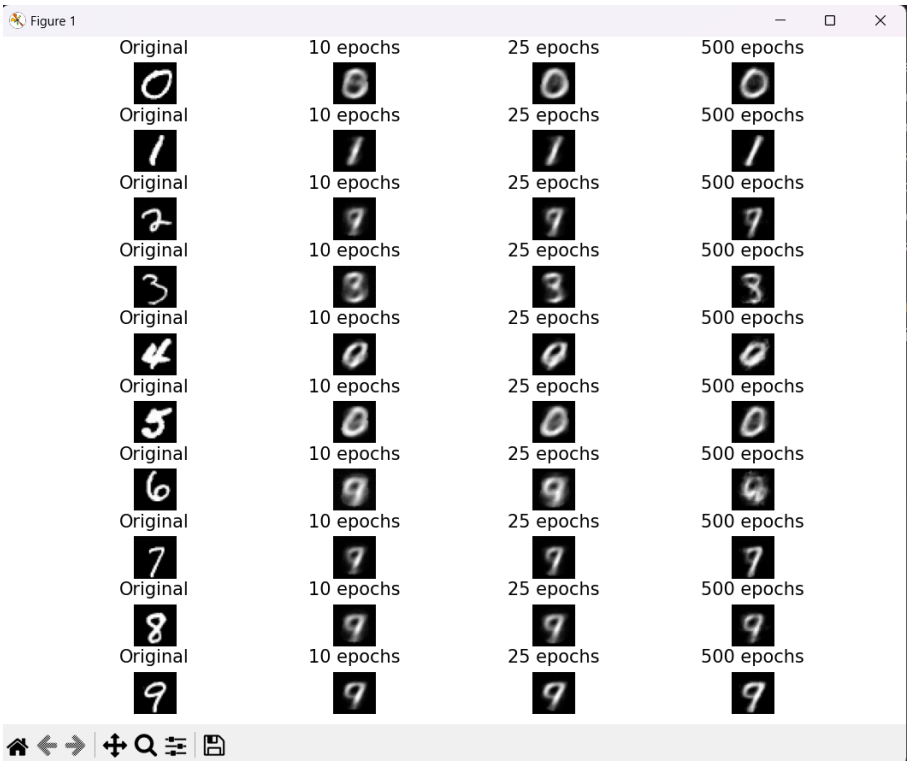


Task 3: Reconstruction of Images

In this study, a method for encoding and reconstructing missing data was applied using the MNIST dataset. Initially, the dataset was reduced to three dimensions using Principal Component Analysis (PCA), a dimensionality reduction technique. The reduced representation was then used to reconstruct the original data through a Multi-Layer Perceptron (MLP). The study evaluated the reconstruction performance of the model across different epoch values (10, 25, and 500).

In the first step of the code, a subset of the MNIST dataset was created, containing 1,000 samples with an equal number of examples from each digit class. This subset was reduced to three principal components using PCA, and an MLP model was used to reconstruct the original 784-dimensional data. The model consisted of two hidden layers, with the first layer containing 10 neurons and the second containing 50 neurons. The output layer, designed to match the original data dimensions, had 784 neurons and employed a sigmoid activation function. The model was trained using the Adam optimization algorithm to minimize the Mean Squared Error (MSE) loss.

The model was trained separately for 10, 25, and 500 epochs, and the reconstructed images were recorded after each training session. The outputs included visual comparisons of the original images and their reconstructions for each digit (0 to 9) at different epochs. Results showed that the reconstructed images were blurry with fewer epochs (10), indicating insufficient learning by the model. At 25 epochs, the clarity of the images improved, and at 500 epochs, the reconstructed images closely resembled the original ones. These findings demonstrate that as the model is trained for more epochs, it learns data representations better, leading to improved reconstruction performance.

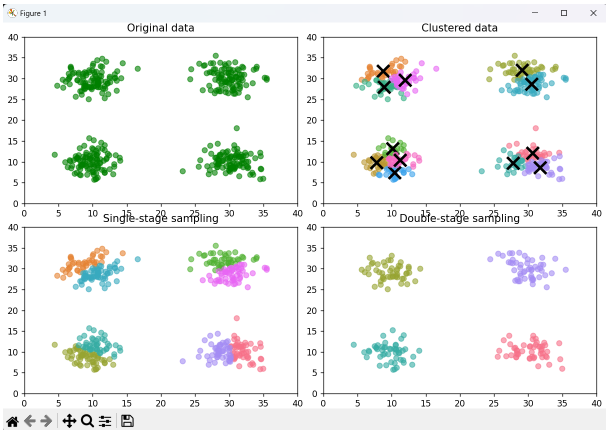


Task 4: Cluster Sampling

This study applied multi-stage clustering on a synthetic dataset using the KMeans algorithm to evaluate the impact of clustering stages on accuracy and processing time. The dataset, comprising 400 data points across four groups, underwent clustering in three stages: 12 clusters ("original multi-clustering"), 8 clusters ("single-stage sampling"), and 4 clusters ("two-stage sampling"). Clusters were visualized and analyzed at each stage.

Evaluation showed that reducing the number of clusters improved processing speed but affected accuracy. For 12 clusters, test accuracy was 10%, with a training time of 121.10 ms. At 8 clusters, accuracy remained 10%, and training time dropped to 9.67 ms. At 4 clusters, accuracy rose to 23%, with a training time of 9.48 ms. Two-stage sampling showed the best balance of accuracy and speed, emphasizing its effectiveness.

This study highlights the trade-off between cluster count, processing efficiency, and accuracy in clustering analysis.




```
(Original Data) Mean Testing Accuracy: 0.10, Training Time: 232.57 ms
(Single-stage Clustering) Mean Testing Accuracy: 0.10, Training Time: 10.77 ms
(Double-stage Clustering) Mean Testing Accuracy: 0.23, Training Time: 4.89 ms
```

Task 5: Novelty Detection

This study developed a text classification model to categorize SMS messages as spam or ham (non-spam). The dataset included two columns: labels (ham = 0, spam = 1) and messages. Messages were preprocessed by converting text to lowercase, cleaning, and removing grammatical inconsistencies.

Text data was transformed into numerical form using the Term Frequency-Inverse Document Frequency (TF-IDF) method with unigrams, filtering out English stop words. The dataset was split into training and test subsets, with the test set including all spam messages and 100 randomly selected ham messages. A Logistic Regression model was trained with a maximum of 1,000 iterations and fixed random state (42).

Performance evaluation on the test dataset yielded 594 True Positives, 100 True Negatives, 0 False Positives, and 153 False Negatives, resulting in an accuracy of 81.94%. While the model effectively identified spam, some spam messages were misclassified.

In conclusion, this model provides a robust approach to spam detection, with room for further optimization in text classification tasks.

```
Performance Metrics:
True Positives (TP): 594
True Negatives (TN): 100
False Positives (FP): 0
False Negatives (FN): 153
Accuracy: 0.8194
```