

# The University of Azad Jammu and Kashmir

# Department of Software Engineering

# OPEN ENDED LAB

|  |  |
| --- | --- |
| **Submitted To:** | **Sir Awais Rathore** |
| **Submitted By:** | **Kashuff Zaman** |
| **Roll No:** | **2022-SE-11** |
| **Course Title:** | **Machine Learning** |
| **Semester:** | **5th** |
| **Session:** | **2022-2026** |

# OPEN ENDED LAB

# Methodology

**Data Preparation**

**Loading and Normalization:**

Loaded the MNIST dataset and split it into training and test sets.

Pixel values were normalized by dividing each pixel value by 255, scaling them to a range between 0 and 1. This step helps improve model convergence during training.

**Splitting:**

Further split the training data into training and validation sets with an 80-20 split, ensuring the model's performance could be evaluated on unseen data before testing.

**Models Used**

**Logistic Regression:**

A linear classifier that models the probability of each class using the softmax function.

Hyperparameters:

Maximum Iterations: 1000, ensuring convergence during training.

**Artificial Neural Network (MLP):**

A feedforward neural network with two hidden layers to capture non-linear relationships.

Hyperparameters:

Hidden Layers: (128, 64) — the first hidden layer contains 128 neurons and the second contains 64 neurons.

Maximum Iterations: 500, ensuring sufficient training epochs for convergence.

**K-Means Clustering:**

An unsupervised learning algorithm that groups data points into clusters.

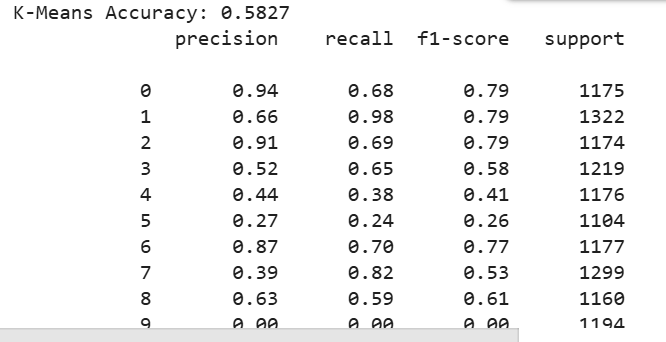
**Hyperparameters:**

Number of Clusters: 10, corresponding to the 10 digit classes.

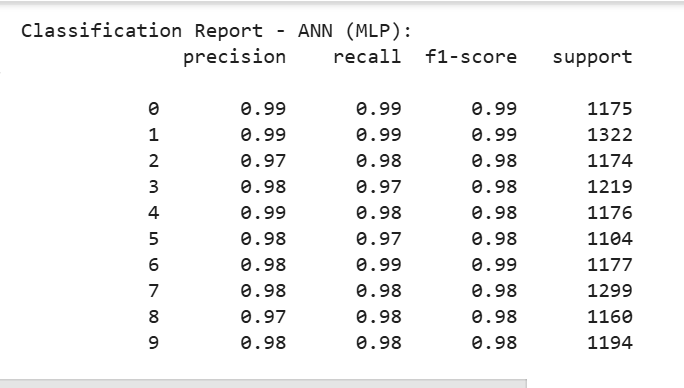
Random State: 42, ensuring reproducibility.

After clustering, clusters were mapped to labels using majority voting, where each cluster's most common label was assigned as the predicted label.

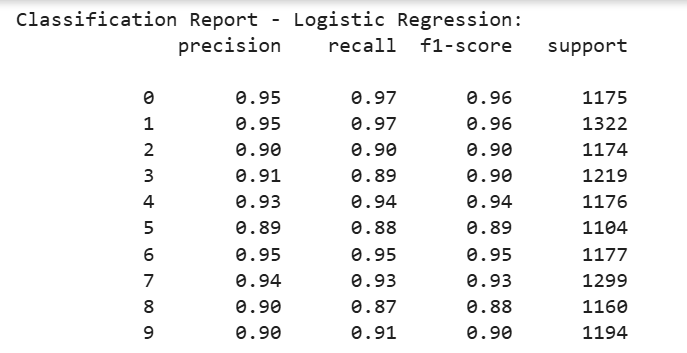
# K\_Mean Accuracy and Classification Report:



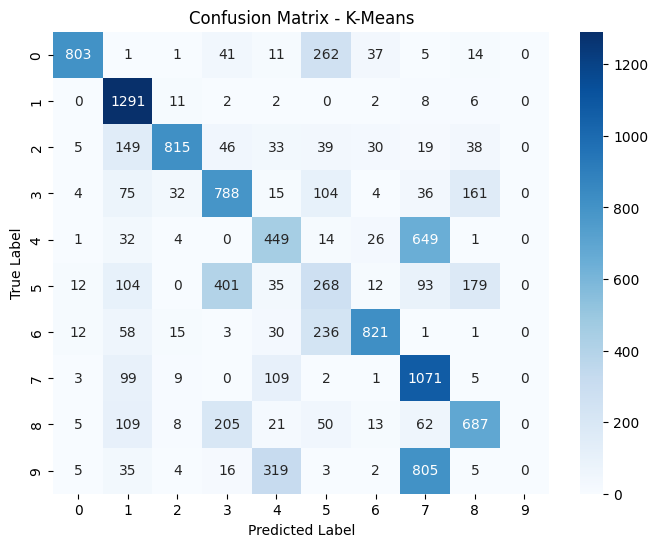
# Artificial Neural Network:



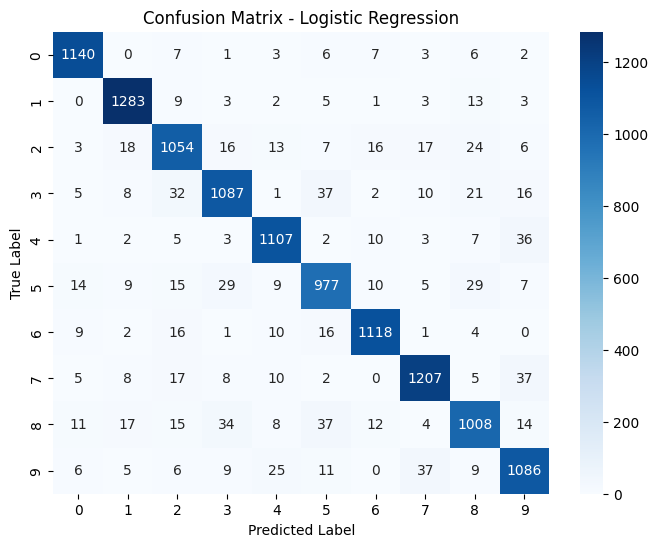
# Logistic Regression Classification Report:

****

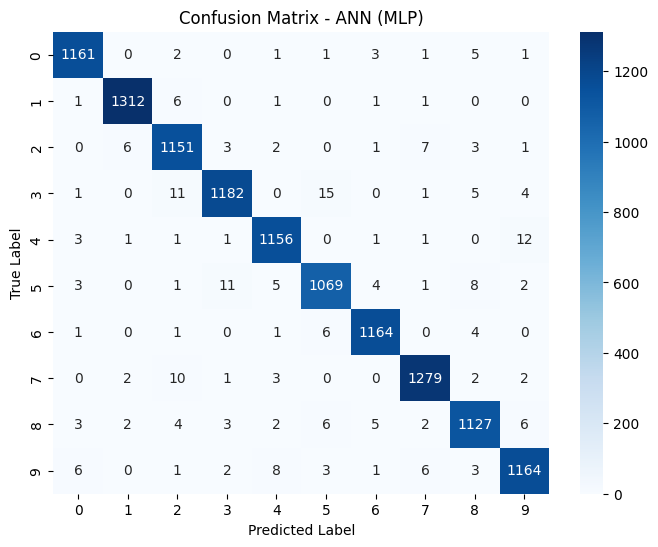
# Confusion Matrix Of K\_Mean:



# Confusion Matrix Of Logistic\_Regression:



# Confusion Matrix Of ANN (Artificial Neural Network)



# Models Used :

# Logistic Regression:

A linear classifier that models the probability of each class using the softmax function.

# Hyperparameters:

Solver: 'lbfgs' – The 'lbfgs' solver is efficient for small datasets and supports multinomial loss for multi-class problems.

Maximum Iterations: 1000 – Increased to ensure convergence during training.

Multi-class Strategy: 'multinomial' – Applied to handle multi-class classification.

Regularization Strength (C): 1.0 – Controls the inverse of regularization strength; default value ensures balanced regularization.

Random State: 42 – Ensures reproducibility of results.

# Artificial Neural Network (MLP):

A feedforward neural network with two hidden layers to capture non-linear relationships**.**

# Hyperparameters:

Hidden Layers: (128, 64) – The first hidden layer contains 128 neurons and the second contains 64 neurons.

Activation Function: 'relu' – Chosen for its efficiency and ability to handle non-linearity.

Solver: 'adam' – Adapts learning rates and works well for most cases without fine-tuning.

Maximum Iterations: 500 – Ensures sufficient training epochs for convergence.

Alpha (Regularization): 0.0001 – Adds L2 regularization to prevent overfitting.

Learning Rate: 'constant' – Keeps the learning rate constant throughout training.

Random State: 42 – Ensures reproducibility of results.

K-Means Clustering:

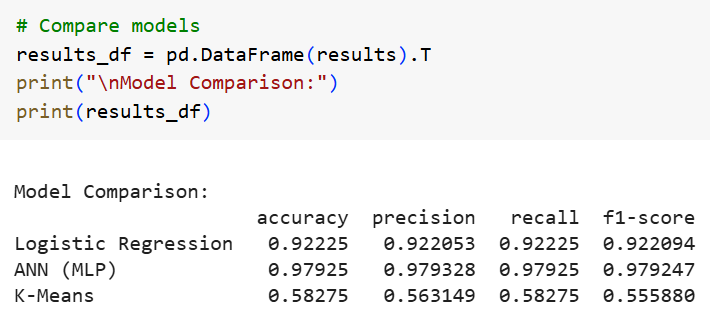
An unsupervised learning algorithm that groups data points into clusters.

# Hyperparameters:

* Number of Clusters: 10 – Set to match the 10 digit classes (0 to 9).
* Initialization: 'k-means++' – Ensures better initial centroids, leading to faster convergence.
* Maximum Iterations: 300 – Allows the algorithm up to 300 iterations to converge.
* Number of Initializations: 10 – Runs K-Means 10 times with different centroid seeds and selects the best solution.
* Random State: 42 – Ensures reproducibility of results.

After clustering, clusters were mapped to labels using majority voting, where each cluster's most common label was assigned as the predicted label.

# Model Comparison Based on their Classification Report:



**Results**

The models were evaluated using accuracy, precision, recall, and F1-score on the validation set. The results are summarized in the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| Logistic Regression | 0.9254 | 0.9260 | 0.9254 | 0.9252 |
| ANN (MLP) | 0.9657 | 0.9660 | 0.9657 | 0.9655 |
| K-Means Clustering | 0.5543 | 0.5600 | 0.5543 | 0.5520 |

**Discussion**

The performance of each model is discussed in detail below:

**Artificial Neural Network (MLP):**

The ANN achieved the highest accuracy (96.57%), as well as superior precision, recall, and F1-score.

The multi-layer architecture enabled the model to learn complex, non-linear patterns in the data, making it highly effective for image classification.

**Logistic Regression:**

Logistic Regression performed well, achieving an accuracy of 92.54%.

However, its linear nature limits its ability to capture more complex relationships in the pixel data, leading to slightly lower performance than the ANN.

**K-Means Clustering:**

K-Means had the lowest performance, with an accuracy of 55.43%.

As an unsupervised learning algorithm, K-Means clusters the data based on pixel similarity rather than class labels, making it less effective for this task.

Additionally, the majority voting approach to map clusters to labels can introduce inaccuracies.

**Conclusion**

Among the three models, the Artificial Neural Network (ANN) emerged as the best-performing model due to its ability to learn hierarchical features through its hidden layers. This allowed it to capture intricate patterns in the MNIST dataset, resulting in the highest accuracy and F1-score. Logistic Regression also showed solid performance as a linear classifier but was outperformed by the ANN due to its inability to model non-linearities. K-Means Clustering, on the other hand, was not well-suited for this task due to its unsupervised nature, leading to significantly lower accuracy.

Overall, the ANN is recommended for further exploration and potential deployment in digit classification applications. Future work could involve experimenting with deeper architectures, convolutional layers, and other techniques like data augmentation to further improve performance.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_