Name: Mubashir Ahmed
ID: CSC-21F-061

Performance Evaluation of Machine Learning Algorithms for MNIST Digit Recognition

Abstract

This study compares the performance of four machine learning algorithms - Logistic Regression, K-Nearest Neighbors (KNN), Random Forest, and Convolutional Neural Networks (CNN) - on the task of MNIST digit recognition. We evaluate these models based on key metrics such as accuracy, log loss, precision, recall, and F1 score. Through comprehensive analysis and visualization, we aim to identify the strengths and weaknesses of each model in handling this classification task.

Introduction:

MNIST digit recognition is a fundamental machine learning problem with applications in optical character recognition and digit classification. This study evaluates the effectiveness of four algorithms for classifying handwritten digits from the MNIST dataset: Logistic Regression, K-Nearest Neighbors (KNN), Random Forest, and Convolutional Neural Networks (CNN). Logistic Regression offers a simple, interpretable baseline. KNN uses a non-parametric approach based on data point proximity. Random Forest employs multiple decision trees for better accuracy. CNNs use deep learning to learn hierarchical representations from pixel values. We compare these models using accuracy, log loss, precision, recall, and F1 score to assess their performance.

Methodology:

Dataset:

The MNIST dataset, from the National Institute of Standards and Technology (NIST), contains 60,000 training images and 10,000 testing images of handwritten digits, each 28x28 pixels with 256 gray levels. The training set includes data from 250 people, evenly split between high school students and Census Bureau employees. Some examples are shown below:

1. Data Collection and Preprocessing

- **Data Source:** The MNIST dataset, consisting of 60,000 training images and 10,000 test images of handwritten digits from 0 to 9.
- **Preprocessing:** The images are normalized to have pixel values in the range [0, 1]. For CNN, images are reshaped to 28x28x1 arrays to fit the input shape.

2. Model Selection and Implementation

2.1. Logistic Regression (LR)

Logistic Regression is a fundamental classification technique that models the probability of a class based on input features. For multi-class tasks like digit recognition, it is extended using methods like softmax regression. It is simple, efficient, and serves as a strong baseline.

Parameters:

- **max_iter:** The maximum number of iterations taken for the solvers to converge.
- **solver:** The solver algorithm used for optimization. In this study, the **'lbfgs'** solver was chosen.
- **multi_class:** The method used to handle multinomial logistic regression. We used the **'multinomial'** option for multiclass classification.

2.2 . K-Nearest Neighbors (KNN)

KNN classifies data points based on the majority class of their nearest neighbors. It is non-parametric and straightforward.

Parameters:
- **n_neighbors:** The number of neighbors to consider for classification. We used a value of **3** in this study.

2.3. Random Forests

Random Forests is an ensemble method that constructs multiple decision trees and aggregates their predictions to improve accuracy and robustness.

Parameters:
- **n_estimators**: The number of trees in the forest. We utilized **100** decision trees in this study.

2.4. Convolutional Neural Networks (CNN)

CNNs are deep learning models tailored for image recognition, utilizing convolutional layers to automatically extract features from pixel data.

Parameters:
- **Architecture:** We employed a basic CNN architecture consisting of convolutional layers, pooling layers, and fully connected layers.
- **Optimizer:** The **'adam'** optimizer was used for gradient descent optimization.
- **Loss function:** We used **'categorical_crossentropy'** as the loss function for multiclass classification.
- **Number of epochs:** The model was trained for **10** epochs with a batch size of **200**.

3. Model Training and Evaluation

- **Training:** Each model is trained on the training set using appropriate training techniques (e.g., gradient descent for LR, backpropagation for CNN).
- **Evaluation Metrics:** The following evaluation metrics are calculated for each model on the test set:
  - **Accuracy:** The proportion of correctly classified instances.
  - **Log Loss:** The negative log-likelihood of the predicted probabilities.
  - **Precision:** The ratio of correctly predicted positive observations to the total predicted positives.
  - **Recall:** The ratio of correctly predicted positive observations to the all observations in actual class.
  - **F1 Score:** The harmonic mean of precision and recall.

4. Evaluation

- **Results Analysis:** The performance metrics of each model are compared and analyzed to determine their effectiveness in digit recognition.
- **Visualizations:** If feasible, visualizations such as confusion matrices, ROC curves, and sample predictions are provided to illustrate the performance of each model.

5. Interpretation and Conclusion

- **Interpretation:** The findings are interpreted to understand the strengths and weaknesses of each model in the context of digit recognition.
- **Conclusion:** A conclusion is drawn regarding the most effective model for MNIST digit recognition based on the evaluation results and analysis.

Literature Review:

An Ensemble of Simple Convolutional Neural Network Models for MNIST Digit Recognition:
This paper explores the use of multiple simple CNN models combined in an ensemble to improve the accuracy of MNIST digit recognition. The authors demonstrate that ensembling several CNNs can outperform individual models by reducing the variance and improving generalization.

Assessing Four Neural Networks on Handwritten Digit Recognition Dataset (MNIST):

The study compares the performance of four different neural network architectures, including CNNs, on the MNIST dataset. The findings highlight the strengths andweaknesses of each model, offering valuable insights into their performance metrics such as accuracy, trainingtime, and computational efficiency.

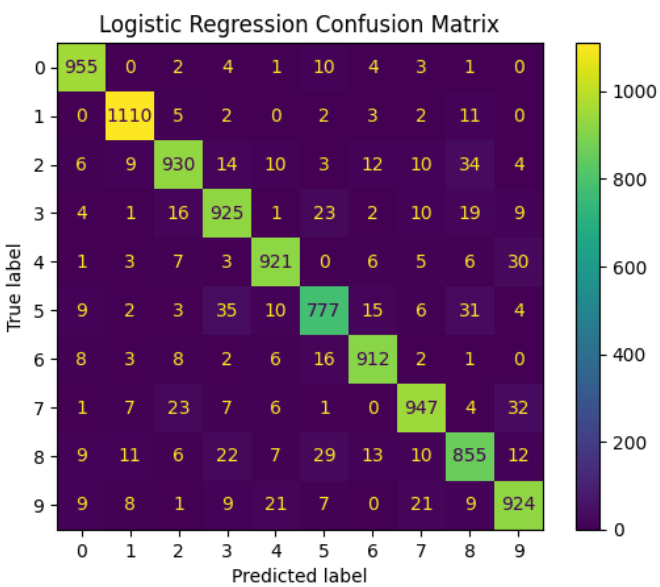
Evaluation:

Results

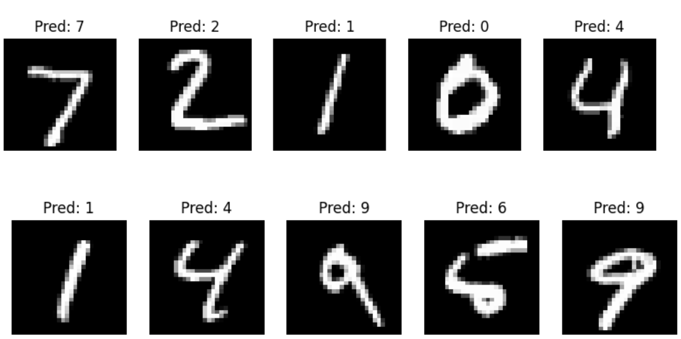
	Logistic Regression	KNN	Random Forest	CNN
Accuracy	**0.9256**	**0.9705**	**0.9693**	**0.9852**
Log Loss	**0.2712**	**0.4875**	**0.2451**	**0.0524**
Precision	**0.9254**	**0.9706**	**0.9692**	**0.9852**
Recall	**0.9256**	**0.9705**	**0.9693**	**0.9852**
F1 Score	**0.9254**	**0.9704**	**0.9692**	**0.9851**

Analysis:

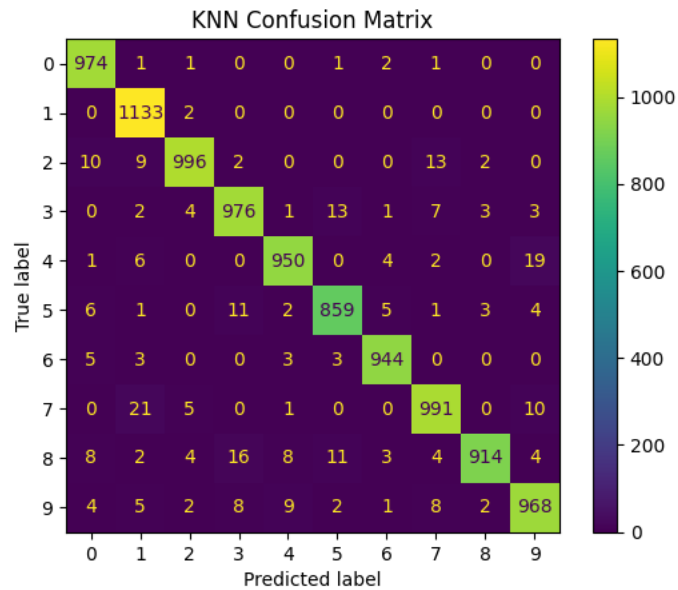
1. Logistic Regression



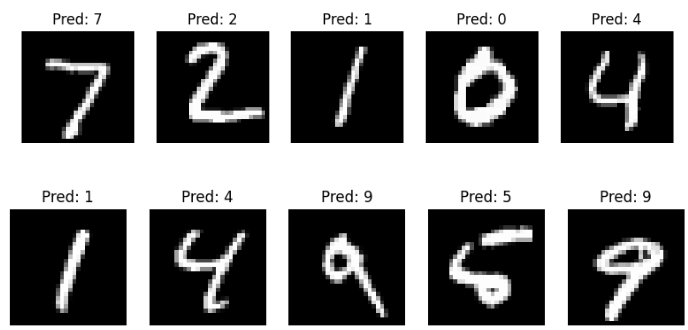
LR Sample Predictions:



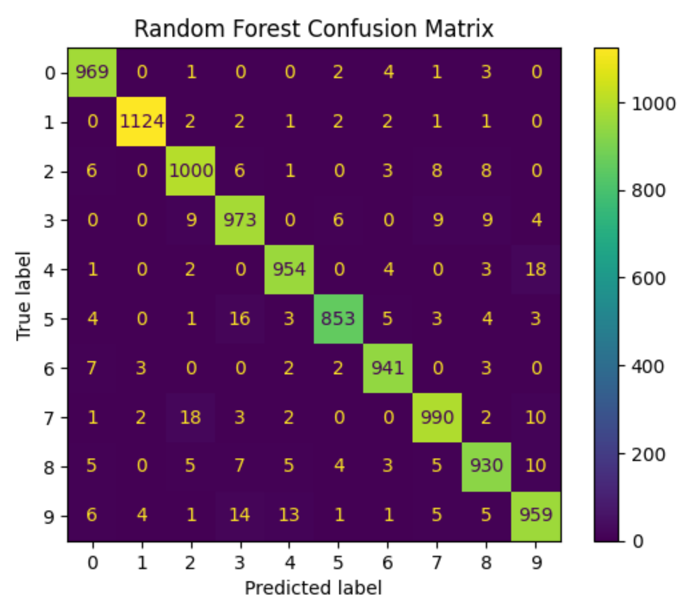
2. K-Nearest Neighbors



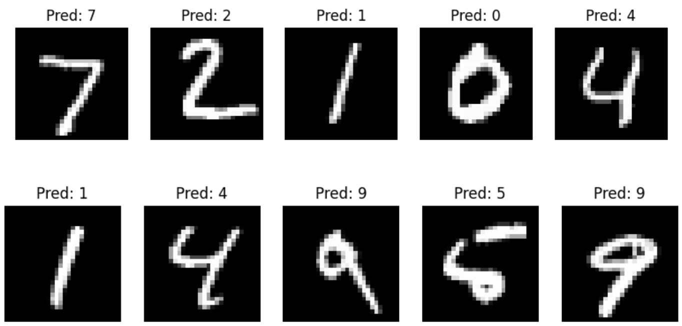
KNN Sample Predictions:



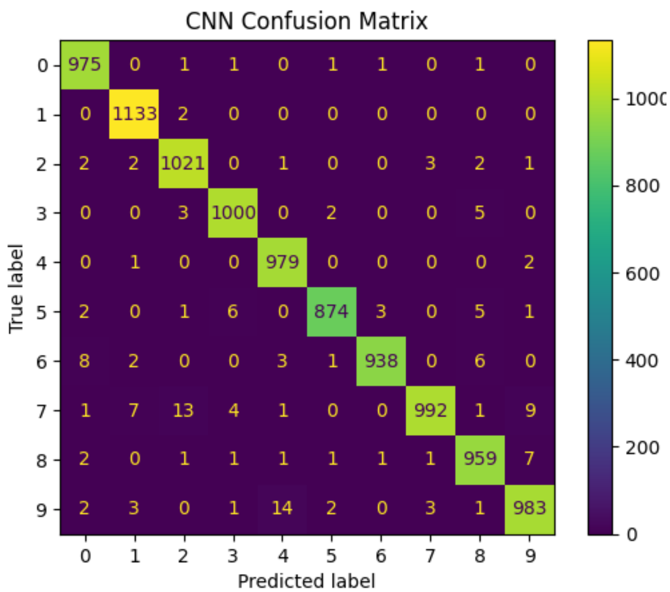
3. Random Forrest:



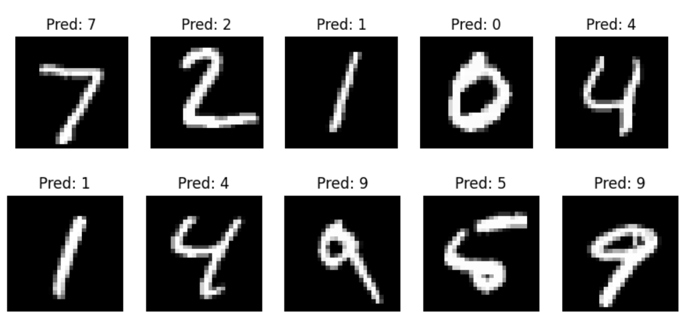
RF Sample Predictions:



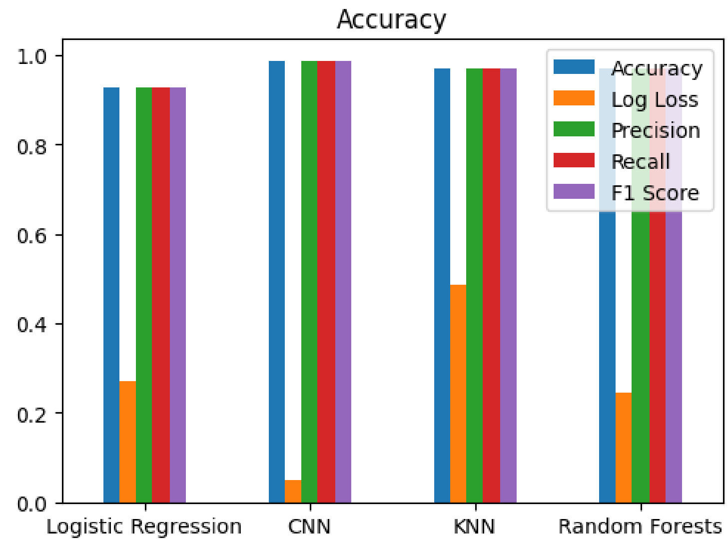
4. CNN:



CNN Sample Predictions:



Comparative Performance Visualization:



Interpretation:

The accuracy scores show that the CNN model performs the best with an accuracy of 98.52%, followed by Random Forest and KNN with accuracies of 96.93% and 97.05%, respectively. Logistic Regression lags slightly behind with an accuracy of 92.56%.

Log loss measures the uncertainty of the model's predictions, and lower values indicate better performance. The CNN model achieves the lowest log loss of 0.0524, indicating its superior predictive capability. Logistic Regression and Random Forest also perform reasonably well in this metric, while KNN exhibits a higher log loss.

Precision, recall, and F1 score are metrics used to evaluate classification models. All models achieve high precision, recall, and F1 scores, indicating robust performance across the board. The CNN model demonstrates the highest precision, recall, and F1 score, closely followed by KNN and Random Forest. Logistic Regression achieves slightly lower scores compared to the other models but still performs reasonably well.

Overall, the **CNN** model stands out as the top performer across most metrics, followed closely by **Random Forest** and **KNN**, while **Logistic Regression** performs slightly less effectively but still demonstrates respectable performance.

Conclusion

In conclusion, this study underscores the importance of selecting appropriate machine learning algorithms based on the task at hand, with CNNs standing out as a powerful tool for image classification tasks like MNIST digit recognition. While Logistic Regression and Random Forest demonstrate respectable performance, the superior accuracy and robustness of CNNs make them the preferred choice for MNIST digit recognition tasks. Future research could explore further optimization and fine-tuning of these models to improve performance and scalability in real-world applications.

References

1. ([AN ENSEMBLE OF SIMPLE CONVOLUTIONAL NEURAL NETWORK MODELS FOR MNIST DIGIT RECOGNITION](https://github.com/Mubashir-19/AI-Course-Work/blob/main/MNIST%20Digit%20Recognition%20Paper%202.pdf))

2. ([Assessing Four Neural Networks on Handwritten Digit Recognition Dataset (MNIST)](https://github.com/Mubashir-19/AI-Course-Work/blob/main/MNIST%20DIGIT%20Recognition%20Paper%201.pdf))

3. [MNIST Database](http://yann.lecun.com/exdb/mnist/)

4. [TensorFlow Documentation](https://www.tensorflow.org/)

5. [Keras Documentation](https://keras.io/)

6. chat.openai.com

7. gemini.google.com