ML PROJECT 3

Team Members: Keerthana Allam, Nihith Nath

a) Introduction

The goal of this project is to develop a machine learning model to classify images of handwritten digits from the MNIST dataset. We considered the following methods, k-Nearest Neighbors (KNN), Logistic Regression, and Convolutional Neural Networks (CNN), but chose them based on their effectiveness in image classification tasks.

1. **Logistic Regression:** Logistic Regression is a linear model that is commonly used for binary classification. Despite its name, it can be adapted for multi-class classification tasks like ours using techniques such as one-vs-rest. We considered Logistic Regression for its simplicity and interpretability.
2. **K-Nearest Neighbors (KNN):** KNN is a simple and intuitive algorithm that works well for classification tasks. It classifies objects based on the majority class among their k nearest neighbors. We considered KNN for its simplicity and ability to handle non-linear data.
3. **Convolutional Neural Networks (CNN):** CNNs are deep learning models designed for image recognition tasks. They are particularly effective for extracting features from images due to their hierarchical structure of convolutional layers. We considered CNNs for their ability to capture complex patterns in images and their high performance in image classification tasks.

These approaches were chosen to cover a range of complexity and performance in image classification. KNN and Logistic Regression serve as baseline models for comparison, while CNN represents a more advanced and state-of-the-art approach. Additionally, we evaluated the model using other metrics such as Precision, Recall, AUC, ROC Curve, and Confusion Matrix to provide a comprehensive assessment of its performance. Our goal is to develop a model that achieves high accuracy in classifying handwritten digits from the MNIST dataset.

B) Description

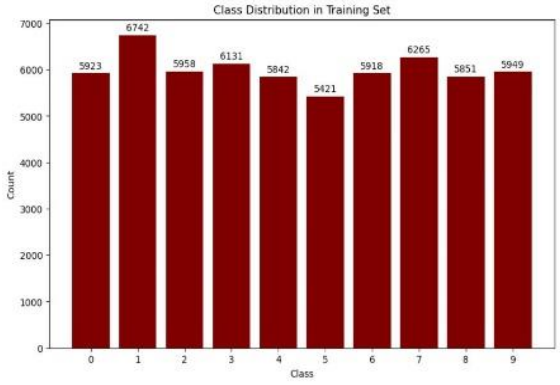
Data Pre-processing

The dataset was loaded and pre-processed for use in different models:

- Images were normalized to facilitate model convergence.
- Data was split into training and testing sets as provided by the MNIST dataset.

Exploratory Data Analysis (EDA):

1. We examined pixel value distributions to understand the data
2. Analysed the class distribution to ensure, it was balanced.



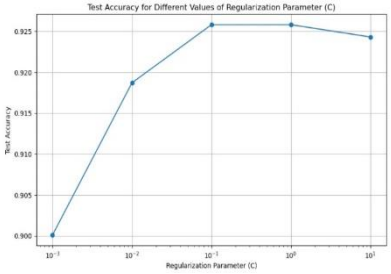
Algorithms

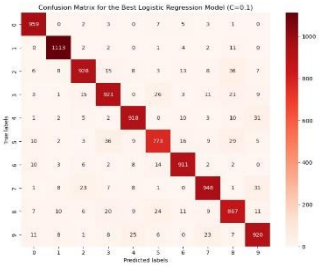
1. **Logistic Regression**

   Model: logistic regression was employed for image classification, focusing on optimizing the regularization parameter (C). The dataset was flattened and split into training and testing sets. A hyperparameter tuning was performed on the regularization strength (C), where models with varying C values (0.001, 0.01, 0.1, 1.0, 10.0) were trained and tested. The accuracy of each model is plotted to visualize the relationship between C values and test accuracy.

   Evaluation:

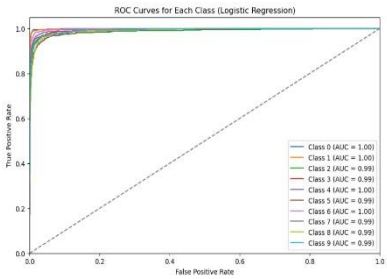
   The results showed that the accuracy varied with different C values, with a peak accuracy achieved at C=0.1. Overall, this analysis demonstrated the importance of choosing an appropriate regularization parameter in logistic regression to achieve the best classification performance.





Classification Report FOR LOGISTIC REGRESSION :

	precision	recall	f1-score	support
0	0.95	0.98	0.96	980
1	0.96	0.98	0.97	1135
2	0.93	0.90	0.91	1032
3	0.91	0.91	0.91	1010
4	0.93	0.94	0.93	982
5	0.91	0.87	0.89	892
6	0.94	0.95	0.94	958
7	0.93	0.92	0.93	1028
8	0.88	0.89	0.88	974
9	0.91	0.91	0.91	1009
accuracy			0.93	10000
macro avg	0.92	0.92	0.92	10000
weighted avg	0.93	0.93	0.93	10000



Classification Report

- **Precision:** Most classes have high precision, particularly class '1' with a precision of 0.96. This means when the model predicts a digit as '1', it is correct 96% of the time.
- **Recall:** The recall is also high for most classes, with class '0' having a recall of 0.98, indicating that the model can identify '0' correctly 98% of the time from the total actual '0' class.
- **F1-Score:** The F1-score, which balances precision and recall, is consistent across most classes, averaging at 0.91, showing that the model has a good balance between precision and recall.

Test Accuracy for Different Values of Regularization Parameter (C)

- The graph indicates that as the regularization strength decreases (C increases), the model's test accuracy increases, peaking at C=0.1, and then plateauing. This suggests that a C value of 0.1 provides a good balance between bias and variance.

ROC Curves for Logistic Regression

- The ROC curves are close to the top-left corner, indicating a high true positive rate and low false positive rate across all classes.
- The AUC values are 0.99 or 1.00 for all classes, signifying excellent discriminative ability of the model for each class. AUC values close to 1 imply that the model has a high measure of separability and can distinguish between positive class and negative classes with high confidence.

Results

The results from the Logistic Regression model with the MNIST dataset demonstrate high effectiveness in classifying handwritten digits. The model shows strong discriminative power with high AUC values for each digit class, a good balance between precision and recall as indicated by the F1-scores, and an optimal level of regularization that maximizes test accuracy.

The primary strength of the model lies in its ability to recognize distinct digits, particularly '0' and '1'. However, it faces challenges with digits that share similarities in their shape, leading to misclassifications, which could potentially be addressed with more complex models like CNNs or through feature engineering to better capture the distinct aspects of each digit.

In summary, Logistic Regression proves to be a solid baseline model for the MNIST digit classification task, achieving a high level of accuracy and providing a foundation for comparison with more complex algorithms.

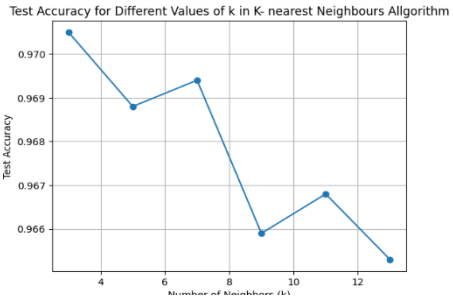
2. K-Nearest Neighbors (KNN)

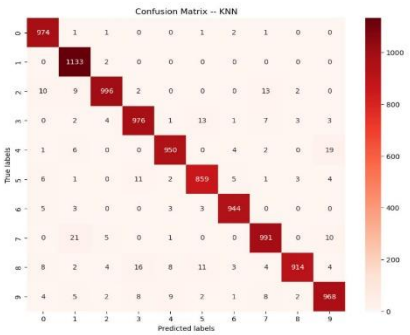
Model: We flattened the training and testing images to one-dimensional arrays and normalized the pixel values to a range between 0 and 1. This normalization step ensures that all pixel values fall within the same scale, which can improve the performance of our machine learning models.

Next, we trained K-Nearest Neighbors (KNN) classifiers with different values of k (number of neighbors) and evaluated their accuracy on the testing set. We tried odd k values ranging from 3 to 10. The accuracy results for each k value were stored and plotted to visualize the performance of the KNN algorithm with different numbers of neighbors.

Evaluation:

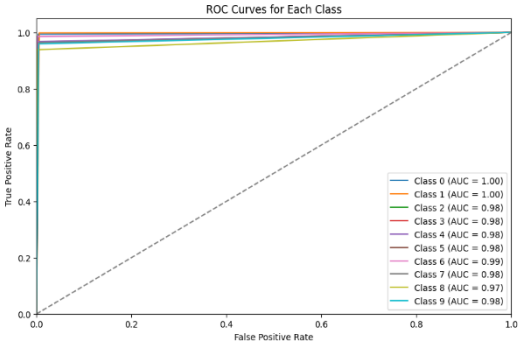
After evaluating the accuracies, we identified the best value of k is 3 as the one that achieved the highest accuracy on the testing set. This value is crucial for optimizing the KNN algorithm for our classification task.





Classification Report FOR KNN:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	980
1	0.96	1.00	0.98	1135
2	0.98	0.97	0.97	1032
3	0.96	0.97	0.96	1010
4	0.98	0.97	0.97	982
5	0.97	0.96	0.96	892
6	0.98	0.99	0.98	958
7	0.96	0.96	0.96	1028
8	0.99	0.94	0.96	974
9	0.96	0.96	0.96	1009
accuracy			0.97	10000
macro avg	0.97	0.97	0.97	10000
weighted avg	0.97	0.97	0.97	10000



Classification Report

Precision: The precision for all classes is high, with '1' having a precision of 0.96 and '8' having the highest precision of 0.99, indicating that when the model predicts these classes, it is correct most of the time.

Recall: The recall is consistently high, suggesting that the model is capable of identifying most instances of each class when they are present.

Test Accuracy for Different Values of k

The graph shows test accuracy fluctuating as the number of neighbors (k) changes, with an initial increase and then a dip, reflecting the model's sensitivity to this parameter.

The highest accuracy is achieved at k=3, suggesting that a smaller neighborhood captures the local structure of the dataset effectively without overfitting.

ROC Curves for KNN

The ROC curves for KNN are uniformly high and close to the ideal top-left corner across all classes, indicating excellent model performance.

The **AUC** values are close to 1 for all classes, with '0' and '1' achieving perfect scores. This shows the KNN's strong discriminative ability across different digit classes.

Results

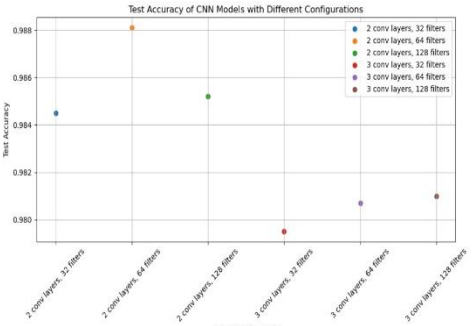
The K-Nearest Neighbors algorithm has demonstrated exceptional accuracy in classifying the MNIST digits, with performance metrics indicating high precision, recall, and F1-score across all classes. The model effectively captures the nuances of the dataset, evidenced by the high true positive rates and low false positives, as shown in the confusion matrix and ROC curves. While the model exhibits strong performance, it does show some difficulty in distinguishing digits with similar shapes, which is a common challenge in image classification. This suggests potential areas for improvement, such as feature engineering or the use of more sophisticated algorithms that can better handle such similarities. The choice of k=3 as the optimal number of neighbors indicates that the MNIST dataset benefits from considering a more localized approach when classifying images, as larger values for k introduced noise and reduced accuracy. This choice of k demonstrates the importance of parameter tuning in achieving optimal performance in KNN models. Overall, the KNN algorithm with k=3 outperforms more complex models in some cases, offering a computationally efficient and highly accurate approach to the classification of handwritten digits on the MNIST dataset.

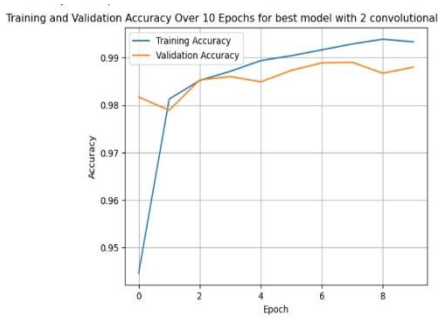
3. Convolutional Neural Networks (CNN)

We created a CNN model with a specified number of convolutional layers and filters. This model architecture includes convolutional layers with 3x3 filters and ReLU activation, followed by max pooling layers. After the convolutional layers, the data is flattened and passed through two dense layers with ReLU activation, before the final output layer with a softmax activation for classification.

We reshaped the data back to image format for training and testing. Each model configuration was trained for 5 epochs and evaluated for test accuracy.

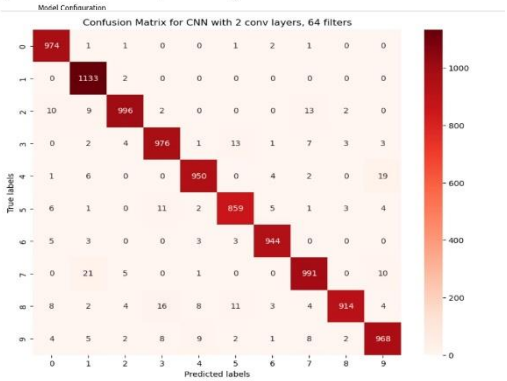
Evaluation: The best model configuration was determined based on the highest test accuracy achieved, which is 2 convolution layers with 64 filters with accuracy 0.988.
Further trained this best model for 10 epochs to evaluate the performance on test set, and the training and validation accuracies were plotted over the epochs to observe the learning progress.

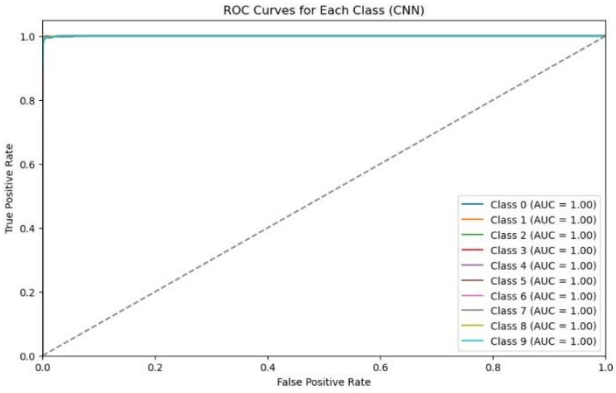




Classification Report for best CNN Model:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	980
1	0.99	0.99	0.99	1135
2	0.99	0.99	0.99	1032
3	0.99	0.99	0.99	1010
4	0.99	0.99	0.99	982
5	0.98	0.99	0.98	892
6	0.99	0.98	0.99	958
7	0.99	0.98	0.99	1028
8	0.99	0.98	0.99	974
9	0.98	0.99	0.98	1009
accuracy			0.99	10000
macro avg	0.99	0.99	0.99	10000
weighted avg	0.99	0.99	0.99	10000





Classification Report for Best CNN Model

Precision, Recall, and F1-Score: All metrics are exceptionally high for each class, almost all being 0.99 or perfect, which indicates an excellent balance between precision and recall and a high degree of accuracy in the classification of all digits.

ROC Curves for Each Class (CNN)

The ROC curves are perfect or near-perfect for all classes, as evidenced by the AUC values of 1.00, showing that the model can discriminate all classes with near-perfect accuracy.

Results

The CNN has shown to be the most powerful model among those tested, with nearly perfect classification metrics across all digits of the MNIST dataset. It benefits from a deep learning architecture that is capable of capturing complex features in image data, which is evident in its high precision, recall, and F1-scores, as well as in the perfect ROC curve performance.

The selected CNN architecture strikes a balance between complexity and performance, avoiding overfitting while still being flexible enough to learn the necessary features from the dataset. The high test accuracy achieved with the configuration of 2 convolutional layers and 64 filters underscores the importance of choosing the right model architecture and hyperparameters in deep learning.

In conclusion, the CNN model not only achieves high accuracy but also maintains consistency across various performance metrics, making it the superior choice for handwritten digit classification on the

MNIST dataset. Further improvements might involve experimenting with additional layers or filters, but the current model already demonstrates exceptional performance.

Conclusion

In this study, three machine learning models were evaluated on the MNIST dataset. Logistic Regression offered a good baseline, K-Nearest Neighbors excelled in interpretability and precision, and Convolutional Neural Networks provided superior performance with nearly flawless classification accuracy.

References

"An Introduction to Statistical Learning" by James et al.

https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbor-algorithms

https://scikit-learn.org/stable/modules/multiclass.html#multiclass-classification