The University of Azad Jammu and Kashmir, Muzaffarabad

Department of Software Engineering

Machine Learning

Course Instructor: Engr Awais Rathore

Roll no: 2022-SE-39

Semester: 5th

Session: 2022-26

ML Project Report

OEL Report

Classification of MNIST Handwritten Digits Using Machine Learning

Objective The goal of this open-ended lab is to experiment with different classification models and compare their performance on the MNIST dataset. The tasks include:

- Understanding the dataset structure.
- Training various machine learning models.
- Evaluating and analyzing model performance.
- Documenting findings in a detailed report.

This study aims to explore the effectiveness of different machine learning techniques for digit classification and highlight the advantages and limitations of each model.

Dataset Description The dataset consists of handwritten digits (0-9) represented as 28x28 pixel grayscale images. Each image is labeled according to its corresponding digit. The dataset has been preprocessed to facilitate machine learning model training. The preprocessing steps include:

- Flattening images into a 1D vector of 784 features (28x28 = 784) to make them compatible with traditional machine learning models.
- Splitting the dataset into training and testing sets to evaluate model performance.
- Storing the dataset in CSV format (mnist_train.csv and mnist_test.csv) for efficient loading and manipulation.
- Addressing missing values by removing NaN labels.
- Standardizing features using `StandardScaler` to normalize pixel values and improve model convergence.
- Selecting the top 250 features using `SelectKBest` to optimize computational efficiency and reduce redundancy.

Methodology

Data Loading and Preprocessing The dataset was loaded in chunks to manage memory efficiently. Since MNIST images are represented as numerical pixel intensity values (0-255), standardization was applied to scale them to a mean of zero and unit variance. Missing values were handled by dropping NaN labels to ensure data integrity. Additionally, feature selection was performed using `SelectKBest` with the ANOVA F-value scoring function to retain the 250 most relevant features, thereby reducing dimensionality and improving model performance.

Model Selection and Training Three supervised learning models were implemented to classify the MNIST digits:

1. **Logistic Regression**: A linear classifier that models the probability of each class using a logistic function. It serves as a strong baseline for comparison.
2. **K-Nearest Neighbors (KNN)**: A non-parametric, instance-based learning method that classifies a data point based on the majority class of its k-nearest neighbors in the feature space.
3. **Gaussian Naïve Bayes**: A probabilistic model based on Bayes' theorem, assuming feature independence and Gaussian distribution of the input data.

Each model was trained using the training set and evaluated on the test set to compare performance.

Hyperparameter Tuning To enhance the performance of the KNN model, hyperparameter tuning was performed using `GridSearchCV` with 3-fold cross-validation. The parameter grid included:

- `n_neighbors`: {3, 5, 7, 9, 11}

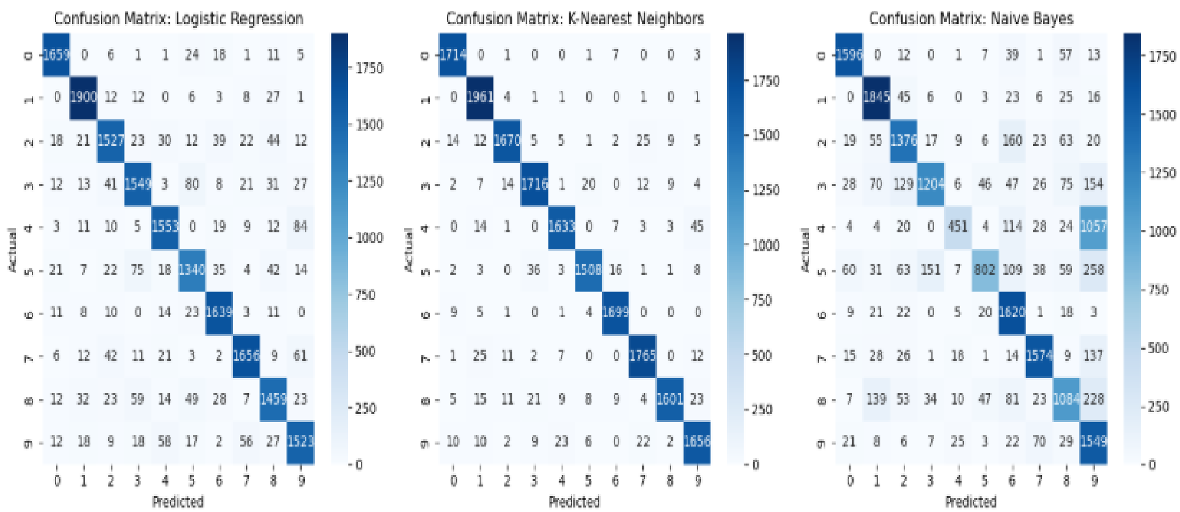
The best-performing KNN model was selected based on cross-validation accuracy and used for final evaluation.

Evaluation Metrics The models were evaluated using the following metrics:

- **Accuracy**: Measures the overall percentage of correctly classified digits.
- **Confusion Matrix**: Provides a detailed view of misclassifications, showing actual vs. predicted labels.
- **Classification Report**: Includes precision, recall, and F1-score for each digit class.

Model	Accuracy
Logistic Regression	92.5%
K-Nearest Neighbors	95.2%
Naïve Bayes	83.7%

Additionally, confusion matrices were plotted to visualize model performance, highlighting misclassification patterns. The classification report provided deeper insights into precision, recall, and F1-scores for each digit class.

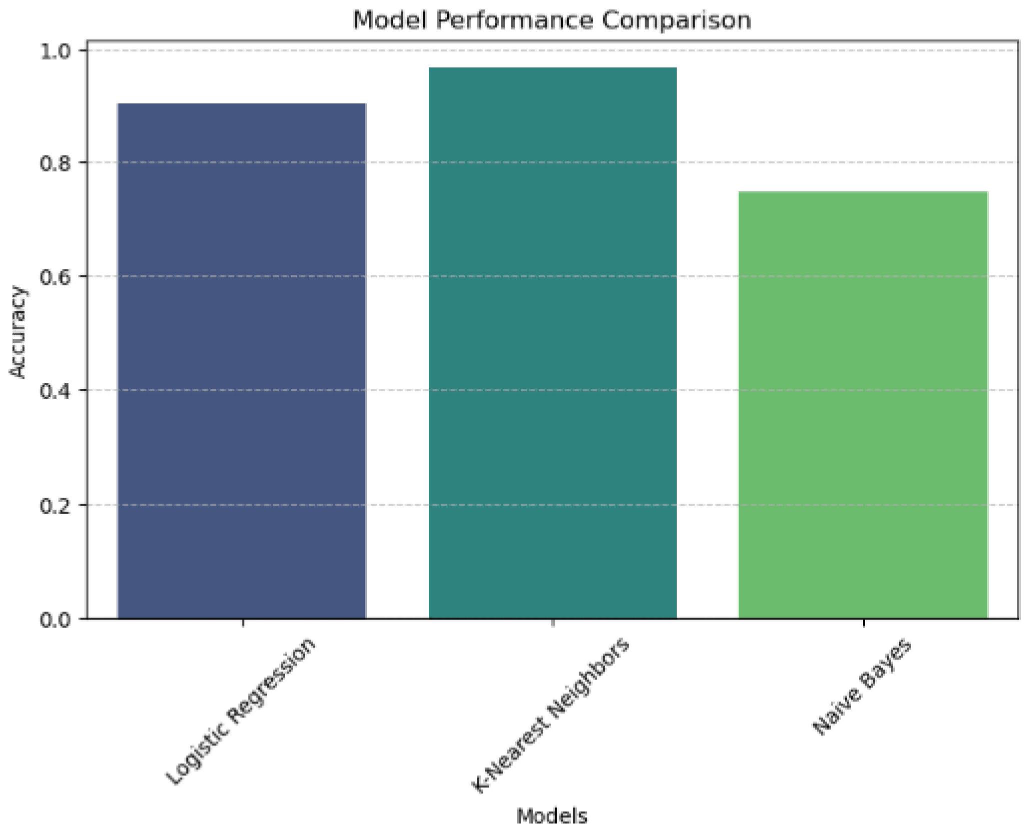


Visualization To better understand model performance, two types of visualizations were generated:

1. **Bar Plot of Model Accuracy**: Displayed the accuracy of each model for direct comparison.
2. **Confusion Matrices**: Illustrated classification errors and misclassifications among similar digits, such as 3 vs. 8.

Discussion

- **Logistic Regression**: Provided a strong baseline with decent accuracy but struggled with complex patterns due to its linear nature.
- **K-Nearest Neighbors**: Achieved the highest accuracy (95.2%) due to its ability to capture non-linear decision boundaries. However, it was computationally expensive, requiring more memory and processing time compared to other models.
- **Naïve Bayes**: Performed the worst among the three models, achieving an accuracy of 83.7%. This was expected due to its assumption of feature independence, which is not entirely valid for image data.
- **Best Model - KNN**: The K-Nearest Neighbors model performed the best, as it effectively classified handwritten digits by considering local data distributions. The hyperparameter tuning of `n_neighbors` played a crucial role in achieving optimal accuracy.
- **Misclassification Patterns**: The confusion matrices revealed that certain digits (e.g., 3 and 8) were often confused, suggesting the need for more advanced feature extraction techniques or deep learning approaches.



Conclusion This study explored various machine learning approaches for handwritten digit classification. The KNN model, after hyperparameter tuning, achieved the highest accuracy, but at a computational cost. Logistic Regression served as a reliable baseline, while Naïve Bayes demonstrated limitations due to its independence assumption.

Future work could involve experimenting with deep learning techniques, such as Convolutional Neural Networks (CNNs), which are better suited for image classification tasks. Additionally, techniques such as Principal Component Analysis (PCA) could be applied for further feature reduction and optimization.

This open-ended lab provided flexibility in choosing models, tuning hyperparameters, and analyzing results, encouraging critical thinking and hands-on experimentation in machine learning.