**The University of Azad Jammu and Kashmir, Muzaffarabad**



**Open Ended Lab**

|  |  |
| --- | --- |
| **Submitted to:** | **Awais Sir** |
| **Course Title:** | **Machine Learning** |
| **Course Code:** |  |
| **Semester:** | **5th** |
| **Roll No:** | **2022-SE-17** |
| **Submitted by:** | **Siyam Mukhtar** |

**Bachelors of Science in Software Engineering (2022-2026)**

**Department of Software Engineering**

**Classification Models OEL Report**

**Introduction**

The purpose of this report is to compare the performance of three machine learning classification models: Logistic Regression, Random Forest, and K-Nearest Neighbors (KNN). The dataset used for this classification task is the **MNIST dataset**, which contains images of handwritten digits (0 through 9) and their corresponding labels. The goal is to accurately classify these images into their respective digit classes.

**Methodology**

**Data Preparation**

The dataset is divided into two parts:

* **Training Set:** mnist\_train.csv
* **Testing Set:** mnist\_test.csv

The features (X\_train and X\_test) are pixel values of the images, while the labels (y\_train and y\_test) are the actual digits.

**Libraries Used**

The following Python libraries were used:

* numpy: For numerical operations
* pandas: For data manipulation
* matplotlib & seaborn: For visualization
* sklearn: For model training, evaluation, and metrics calculation

**Models Evaluated**

1. **Logistic Regression**
2. **Random Forest Classifier**
3. **K-Nearest Neighbors (KNN)**

**Evaluation Metrics**

The models are evaluated based on the following metrics:

* **Accuracy:** The proportion of correct predictions.
* **Precision:** The ability of the model to correctly predict positive instances.
* **Recall:** The ability of the model to find all positive instances.
* **F1-Score:** The harmonic mean of precision and recall.
* **Confusion Matrix:** To visualize prediction errors.

**Results**

The models were evaluated using multiple hyperparameter configurations, and the best-performing model for each classifier was selected based on accuracy. The results are displayed below.

**Logistic Regression**

Best Hyperparameters: {best\_logistic\_params}  
Accuracy: {best\_logistic\_accuracy}

**Classification Report**

{classification\_report(y\_test, best\_y\_pred\_logistic)}

**Confusion Matrix**

(Confusion Matrix Plot displayed)

**Random Forest**

Best Hyperparameters: {best\_rf\_params}  
Accuracy: {best\_rf\_accuracy}

**Classification Report**

{classification\_report(y\_test, best\_y\_pred\_rf)}

**Confusion Matrix**

(Confusion Matrix Plot displayed)

**K-Nearest Neighbors (KNN)**

Best Hyperparameters: {best\_knn\_params}  
Accuracy: {best\_knn\_accuracy}

**Classification Report**

{classification\_report(y\_test, best\_y\_pred\_knn)}

**Confusion Matrix**

(Confusion Matrix Plot displayed)

**Model Comparison**

The performance of each model is summarized in the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score |
| Logistic Regression | {best\_logistic\_accuracy} | {precision\_score(y\_test, best\_y\_pred\_logistic, average='weighted')} | {recall\_score(y\_test, best\_y\_pred\_logistic, average='weighted')} | {f1\_score(y\_test, best\_y\_pred\_logistic, average='weighted')} |
| Random Forest | {best\_rf\_accuracy} | {precision\_score(y\_test, best\_y\_pred\_rf, average='weighted')} | {recall\_score(y\_test, best\_y\_pred\_rf, average='weighted')} | {f1\_score(y\_test, best\_y\_pred\_rf, average='weighted')} |
| K-Nearest Neighbors | {best\_knn\_accuracy} | {precision\_score(y\_test, best\_y\_pred\_knn, average='weighted')} | {recall\_score(y\_test, best\_y\_pred\_knn, average='weighted')} | {f1\_score(y\_test, best\_y\_pred\_knn, average='weighted')} |

**Conclusion**

The comparison of the models shows that the best performance was achieved by the model with the highest accuracy. The report demonstrates that various classifiers have their strengths and weaknesses, which may vary depending on the dataset and chosen hyperparameters.

Further improvements can be made by tuning hyperparameters more extensively and experimenting with other classifiers such as Support Vector Machines (SVM) or Neural Networks.