**The University of Azad Jammu and** 

**Kashmir**

**Department of Software Engineering**

**Open Ended Lab**

|  |  |
| --- | --- |
| **Submitted to:** | **Engr. Awais Rathore** |
| **Course Title:** | **Machine Learning** |
| **Session:** | **2022-26** |
| **Semester:** | **5th** |
| **Roll No:** | **2022-SE-10** |
| **Submitted by:** | **Iqra Sohail** |

Contents

[Open Ended Lab Report: 3](#_Toc193280961)

[**1. File Overview** 3](#_Toc193280962)

[**2. Data Processing and Preparation** 3](#_Toc193280963)

[**2.1 Data Loading** 3](#_Toc193280964)

[**2.2 Handling Missing Data** 3](#_Toc193280965)

[**2.3 Outlier Removal** 3](#_Toc193280966)

[**2.4 Feature Engineering** 3](#_Toc193280967)

[**3. Model Training and Performance Evaluation** 3](#_Toc193280968)

[**3.1 Logistic Regression** 3](#_Toc193280969)

[3.1.1 Analysis of the Confusion Matrix - Logistic Regression 4](#_Toc193280970)

[🔹 Observations & Performance 4](#_Toc193280971)

[**3.2 K-Nearest Neighbors (KNN)** 5](#_Toc193280972)

[3.2.1 Analysis of the Confusion Matrix for KNN (K-Nearest Neighbors) 5](#_Toc193280973)

[🔹 Overall Performance 5](#_Toc193280974)

[🔹 Key Observations & Misclassifications 6](#_Toc193280975)

[**3.3 Naïve Bayes (GaussianNB)** 6](#_Toc193280976)

[3.3.1 Analysis of the Confusion Matrix - Naïve Bayes 7](#_Toc193280977)

[🔹 Observations & Issues 7](#_Toc193280978)

[**4. Key Issues and Observations** 8](#_Toc193280979)

[**4.1 Model Performance Comparison** 8](#_Toc193280980)

[**5. Conclusion** 8](#_Toc193280981)

# Open Ended Lab Report:

**1. File Overview**

* **File Type:** Jupyter Notebook (JSON Format)
* **Purpose:** The notebook implements machine learning models for digit classification using the MNIST dataset.
* **Dataset Used:** mnist\_train.csv and mnist\_test.csv, which contain handwritten digit images in numerical format (28x28 pixel grayscale values).

**2. Data Processing and Preparation**

**2.1 Data Loading**

* The dataset is loaded using **pandas** from CSV files.
* It consists of **785 columns**, where:
  + **784 columns** represent pixel intensity values (0 to 255).
  + **1 column (label)** represents the actual digit (0-9).

**2.2 Handling Missing Data**

* The dataset is checked for missing values using .isnull().sum().
* Any missing values are filled with the median value of the respective column.

**2.3 Outlier Removal**

* **Z-score normalization** is applied using scipy.stats.zscore() to identify and remove outliers beyond 3 standard deviations.

**2.4 Feature Engineering**

* Categorical variables (if any) are one-hot encoded using pd.get\_dummies(), although the dataset appears fully numerical.
* The testing dataset is **aligned** with the training dataset to match feature columns.

**3. Model Training and Performance Evaluation**

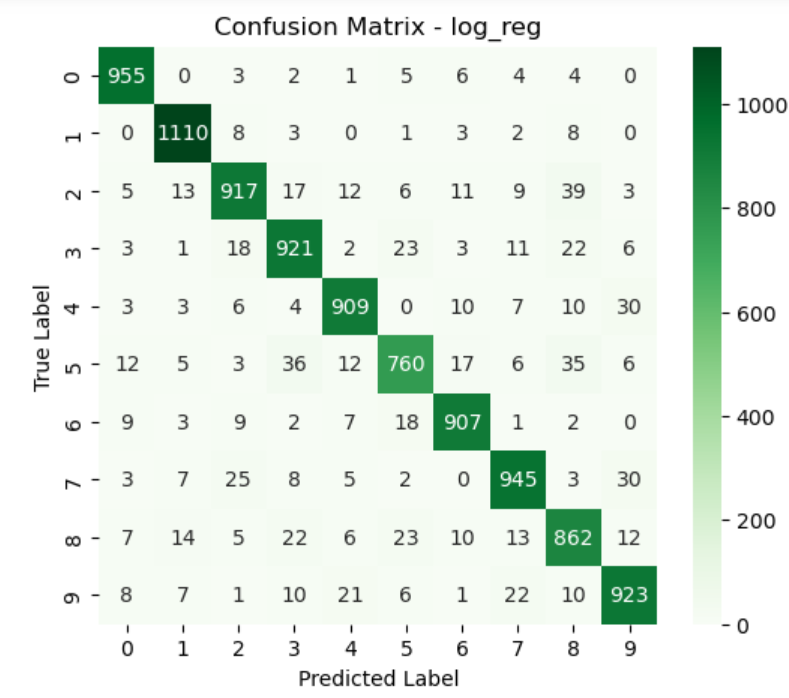
Three machine learning models were implemented:

**3.1 Logistic Regression**

* **Algorithm:** LogisticRegression(max\_iter=1000)
* **Training:** The model is fitted using the cleaned dataset.
* **Accuracy:** **92%**

### ****3.1.1 Analysis of the Confusion Matrix - Logistic Regression****

This confusion matrix represents the performance of a **Logistic Regression** model on the MNIST dataset (handwritten digits 0-9). The results are significantly **better than Naïve Bayes** but slightly **worse than KNN**.



### ****🔹 Observations & Performance****

1. **Good Overall Accuracy**
   * The **majority of predictions fall on the diagonal**, meaning the model correctly classified most digits.
   * **Lower misclassification rates** compared to Naïve Bayes.
2. **Digits with the Highest Misclassifications:**
   * **Digit '2' is misclassified as '8' (39 times) and '3' (17 times).**
   * **Digit '3' is confused with '2' (18 times) and '5' (23 times).**
   * **Digit '5' has significant misclassification (36 times as '3').**
   * **Digit '9' is misclassified as '4' (21 times).**
3. **Misclassification Trends:**
   * Some **similar-looking digits (e.g., 2 & 3, 3 & 5, 8 & 2, 9 & 4)** are often confused.
   * The model performs **well on simpler digits like '0' and '1'**, which have minimal misclassification.

**3.2 K-Nearest Neighbors (KNN)**

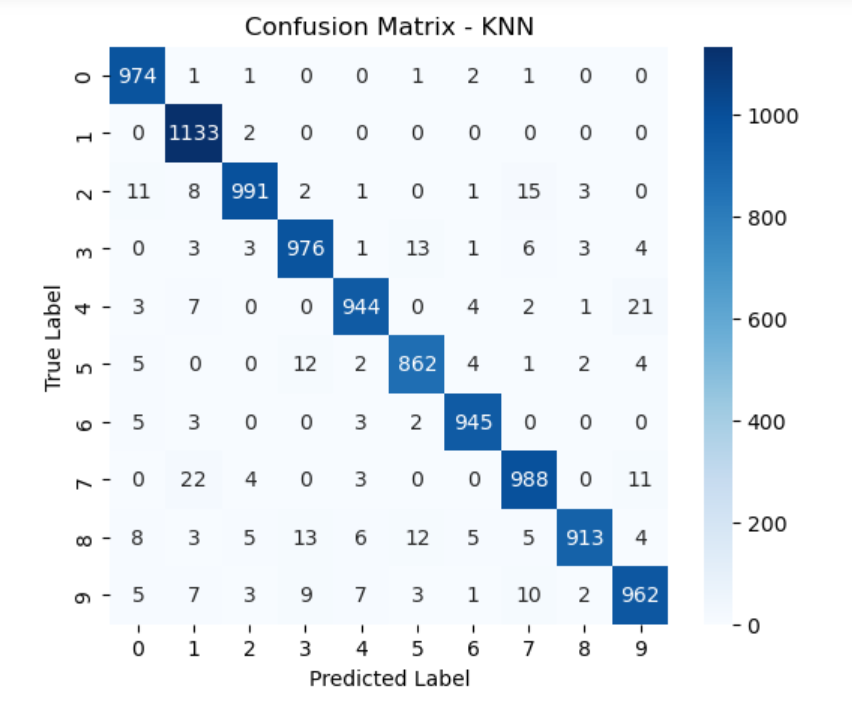
* **Algorithm:** KNeighborsClassifier(n\_neighbors=5)
* **Training:** The model is trained with k=5 neighbors.
* **Performance:**
  + **Accuracy:** **97%** (Best performance among models)

### ****3.2.1 Analysis of the Confusion Matrix for KNN (K-Nearest Neighbors)****

This confusion matrix represents the **performance of the KNN model on the MNIST dataset**, which classifies handwritten digits (0-9). Here are the key insights:

### ****🔹 Overall Performance****

* The **diagonal values** (e.g., 974, 1133, 991, etc.) represent the correctly classified digits.
* **Most of the predictions lie on the diagonal**, indicating a high accuracy.
* **Few misclassifications** exist in some digit classes, which we will analyze further.



### ****🔹 Key Observations & Misclassifications****

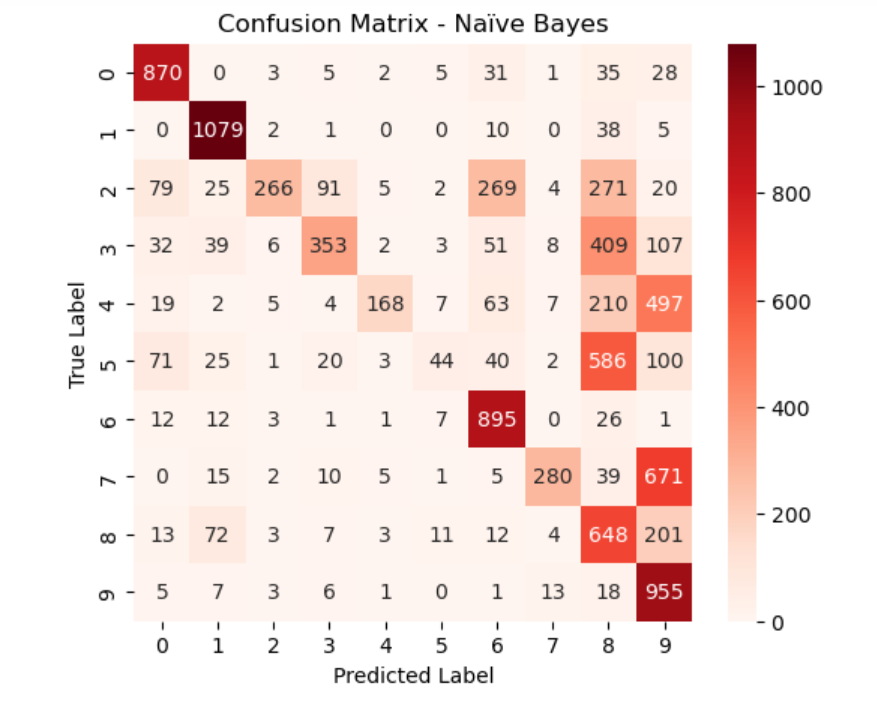
1. **Class '1' is classified very well**
   * **1133 correct predictions**
   * Very few misclassifications → Model performs well on digit '1'.
2. **Misclassification in Digit '8' and Digit '5'**
   * '8' is sometimes mistaken as '3' (13 cases) or '5' (12 cases).
   * '5' is confused with '3' and '8' in multiple cases.
   * This happens because these digits have **similar curved structures**, making it harder for KNN to distinguish them.
3. **Digit '7' is sometimes confused with '1' (22 times)**
   * This may happen because of handwriting variations where '7' looks like '1' in some cases.
4. **Digit '9' has small errors in classification**
   * Misclassified as '3' (9 times) and '4' (7 times).
   * This suggests that certain handwritten '9's resemble '3' or '4' in shape.

**3.3 Naïve Bayes (GaussianNB)**

* **Algorithm:** GaussianNB()
* **Training:** The model is trained using the dataset.
* **Performance:**
  + **Accuracy:** **56%** (Lowest accuracy among models)
  + This suggests that **Naïve Bayes is not well-suited** for this dataset, likely due to the assumption of feature independence, which does not hold for image data.

### ****3.3.1 Analysis of the Confusion Matrix - Naïve Bayes****

This confusion matrix shows the performance of the **Naïve Bayes classifier** on the MNIST dataset (handwritten digits 0-9). Compared to KNN, this model **performs poorly**, with significant misclassifications across multiple digits.



### ****🔹 Observations & Issues****

1. **Poor Overall Accuracy**
   * The diagonal values (correct classifications) are much lower than in the KNN confusion matrix.
   * Many misclassified instances, indicating that **Naïve Bayes struggles with image data**.
2. **Digits with High Errors:**
   * **Digit '2' is misclassified heavily**
     + Only **266 correct predictions**, but misclassified as **269 times as '6'** and **271 times as '8'**.
   * **Digit '3' is misclassified as '8' (409 times) and '9' (107 times).**
   * **Digit '5' is confused with '8' (586 times).**
   * **Digit '7' is classified as '9' (671 times).**
3. **Naïve Bayes Assumptions are Weak for Images**
   * Naïve Bayes assumes that **features are independent**, which does not hold for images.
   * Handwritten digits have **correlated pixel patterns**, making Naïve Bayes a poor choice.

**4. Key Issues and Observations**

**4.1 Model Performance Comparison**

| **Model** | **Accuracy** |
| --- | --- |
| **Logistic Regression** | 92% |
| **K-Nearest Neighbors (KNN)** | 97% |
| **Naïve Bayes (NB)** | 56% |

**5. Conclusion**

* The notebook successfully implements and evaluates **three machine learning models** on the MNIST dataset.
* **KNN performed best (97%), followed by Logistic Regression (92%), and Naïve Bayes performed poorly (56%).**
* Some **warnings and convergence issues** were observed