**The University of Azad Jammu and** 

**Kashmir**

**Department of Software Engineering**

**Open Ended Lab**

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

Contents

[Report on Jupyter Notebook (MNIST Classification) 2](#_Toc193056061)

[**1. File Overview** 2](#_Toc193056062)

[**2. Data Processing and Preparation** 2](#_Toc193056063)

[**2.1 Data Loading** 2](#_Toc193056064)

[**2.2 Handling Missing Data** 2](#_Toc193056065)

[**2.3 Outlier Removal** 2](#_Toc193056066)

[**2.4 Feature Engineering** 2](#_Toc193056067)

[**3. Model Training and Performance Evaluation** 2](#_Toc193056068)

[**3.1 Logistic Regression** 3](#_Toc193056069)

[**3.2 K-Nearest Neighbors (KNN)** 3](#_Toc193056070)

[**3.3 Naïve Bayes (GaussianNB)** 3](#_Toc193056071)

[**4. Key Issues and Observations** 3](#_Toc193056072)

[**4.1 Model Performance Comparison** 3](#_Toc193056073)

[**5. Conclusion** 3](#_Toc193056074)

# Report on Jupyter Notebook (MNIST Classification)

**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%**

**3.2 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)

**3.3 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.

**4. Key Issues and Observations**

**4.1 Model Performance Comparison**

| **Model** | **Accuracy** | **Issues/Warnings** |
| --- | --- | --- |
| **Logistic Regression** | 92% | Convergence warning (more iterations needed) |
| **K-Nearest Neighbors (KNN)** | 97% | Future SciPy mode behavior warning |
| **Naïve Bayes (NB)** | 56% | Poor accuracy due to feature independence assumption |

**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