Handwritten Digit Recognition for Legacy Utility Meters

By
Manoj

Date:03-Sep-2024

Introduction

- ABC Utility company that produce and deliver basic essential Services such as electricity, natural gas and water.

Problem Statement:

- ABC Utility company need to regularly read meter data for billing purposes, and some older meters display readings in handwritten digits.
- We have MNIST dataset to recognize handwritten digits ,which consists of 60,000 training images and 10,000 test images of digits (0-9).

Challenges of Manual Meter Reading

- Human error in reading and recording meter data.
- High operational costs due to labor-intensive processes.
- Inconsistent readings leading to billing inaccuracies.
- Difficulty in reading meters in remote or hard-to-access locations.

Action

A digit recognition model can be deployed .So that,it automatically read and record meter data, even from analog meters with handwritten or stylized digits.

Data Preprocessing:

- **Data Loading:** The MNIST dataset was loaded from CSV files containing pixel values (0-255) and corresponding labels.
- **Normalization:** The pixel values were normalized to the range of 0 to 1 to improve the efficiency of the algorithm.
- **Reshaping:** The images, originally represented as 784-dimensional vectors, were reshaped into 28*28 matrices for visualization purposes.

Classification Model:

- **The Classification model is an algorithm that is deployed in analyzing the complex set of known data points and appropriately classifying them.**

Algorithm Models:

- Naives Bayes
- KNN Algorithm

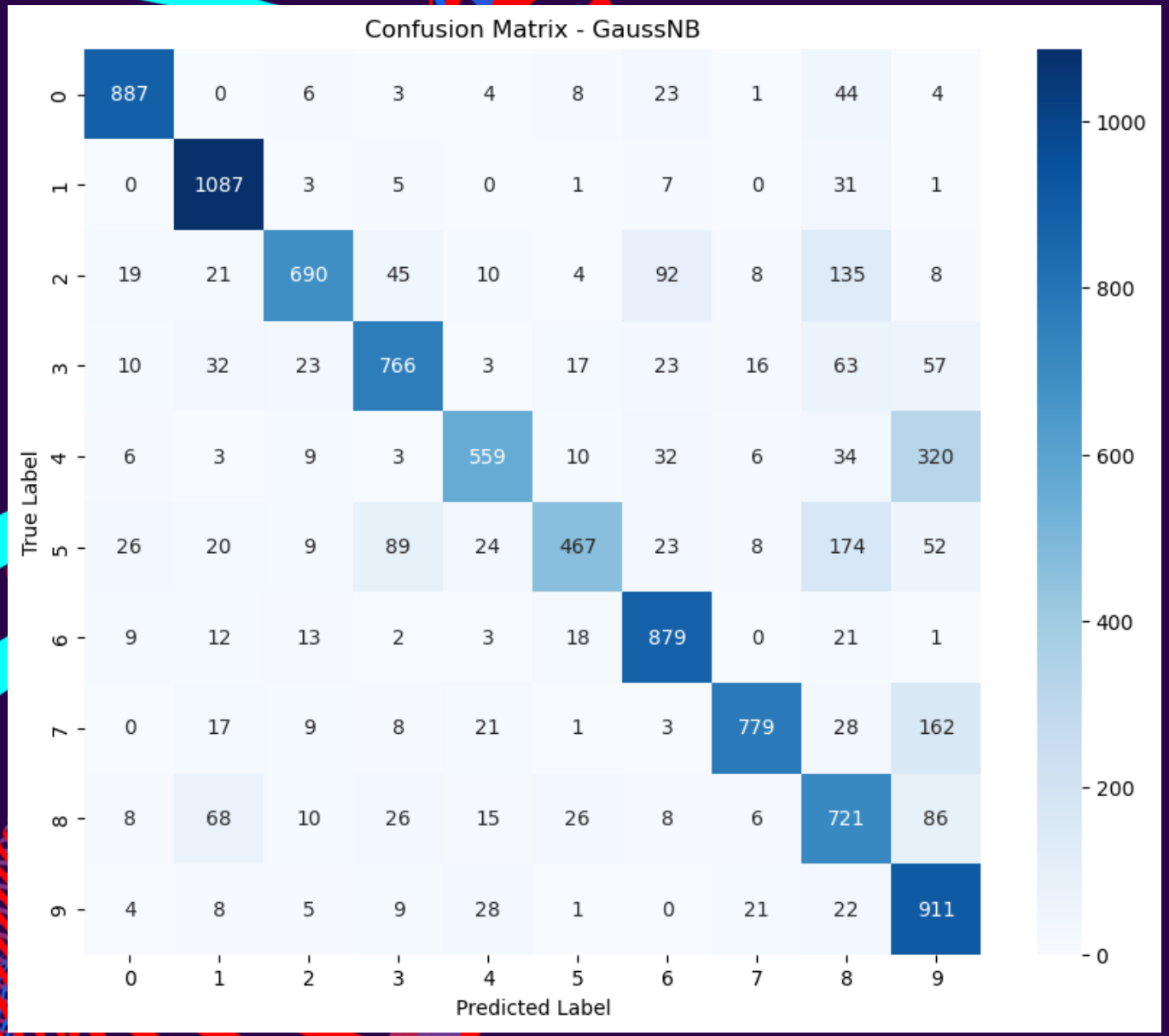
Naive Bayes:

- **The Naives Bayes Classifier will operate by returning the class,which has the maximum posterior probability out of a group of classes(i.e. “spam” or “not spam”) for a given e-mail.**

Implementation:

- **The Naives Bayes Classifier is implemented**
  Train Data Accuracy-76.82%
  Test Data Accuracy-77.46

Key_Visuals



Actionable Insights:

- **Model Performance:** The classifier performs well, but there are specific classes where misclassification is more frequent.
- **Potential Improvements:** Focus on improving the classification of classes with significant off-diagonal elements (e.g., class 5 and class 8) by possibly fine-tuning the model or incorporating additional features.
- Further tuning makes the model overfitting.

Gaussian Processes (GP):

- A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. It's used for defining a distribution over functions.
- For a set of input points, a GP predicts a distribution of possible functions that fit the data. The core idea is that any function can be modeled as a sample from a Gaussian process.

Implementation:

- **The Gaussian Classifier is implemented**
  Train Data Accuracy-93.07%
  Test Data Accuracy-91.07%

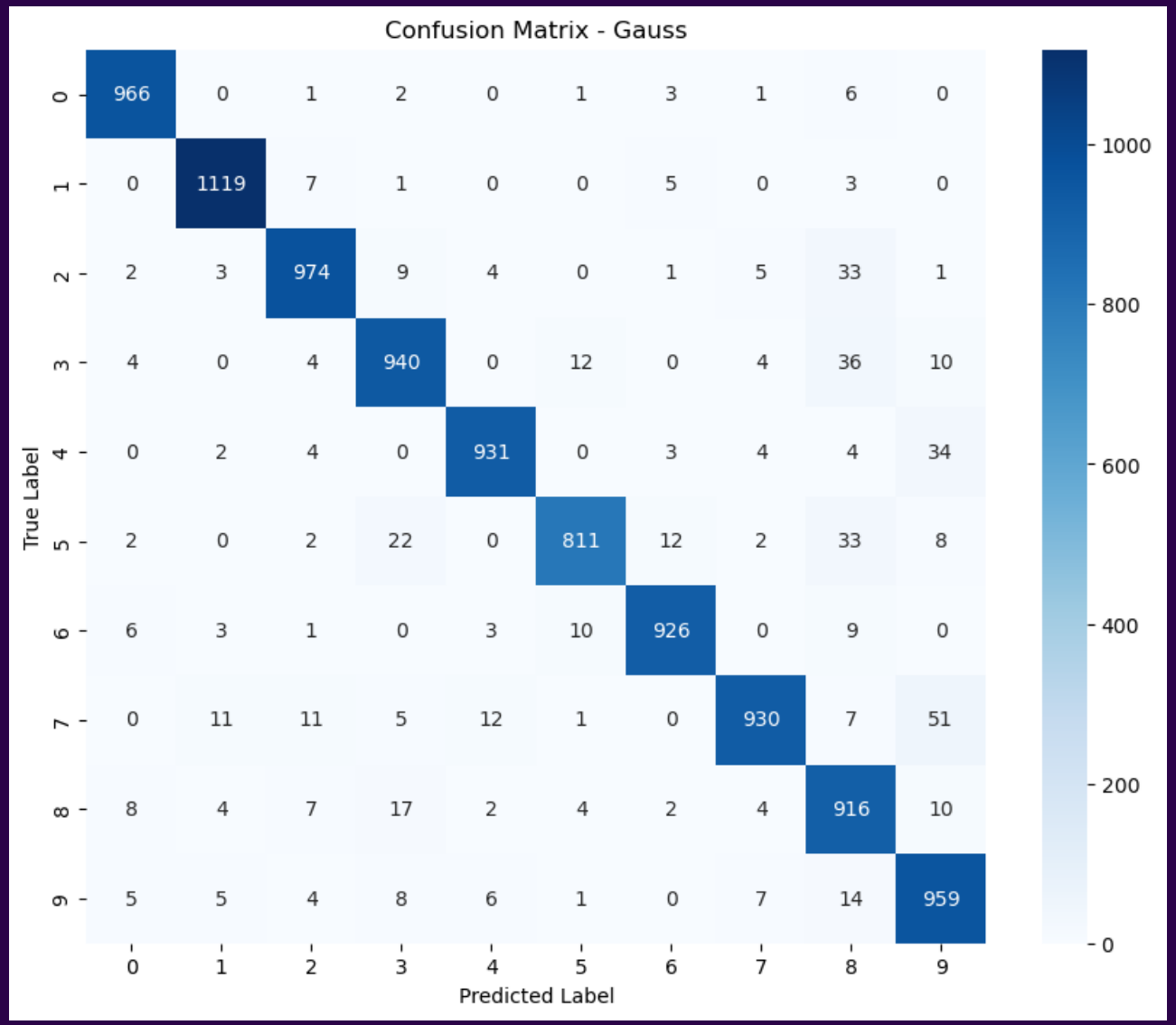
Hyperparameter Tuning (Epsilon):

- The epsilon parameter is used for variance smoothing. You can try different values of epsilon to see if it helps in improving accuracy.
- You could implement a simple grid search manually to find the best epsilon value.

Implementation:

- **The Gaussian Classifier is implemented**
  After Tuning:
  Train Data Accuracy- 95.67 %
  Test Data Accuracy-94.72%

Key_visuals



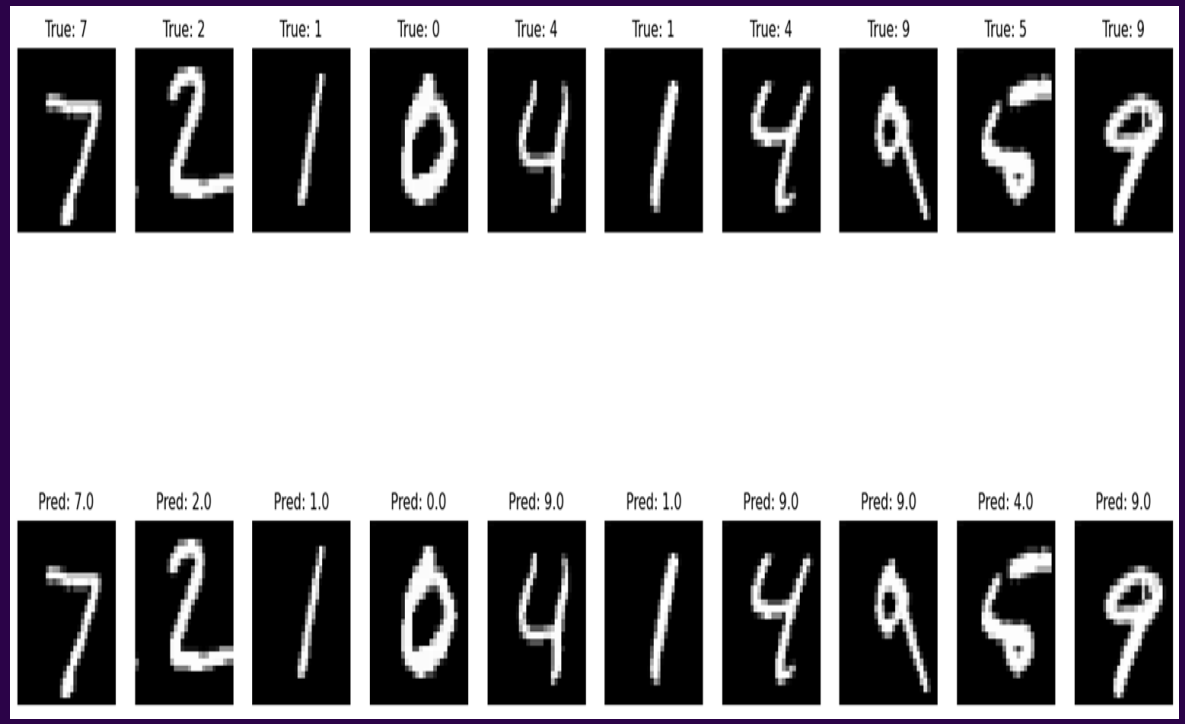
- This confusion matrix gives a detailed breakdown of the model's performance for each class, highlighting both its strengths and areas where it may need improvement. Overall, the model seems to perform well, with relatively few errors compared to the total number of instances.

K_Nearest Neighbours

- **The K-Nearest Neighbours(KNN) algorithm is a popular machine learning technique used for classification and regression tasks.it relies on the idea that similar data points tends to have similar labels or values.**

TImplementation

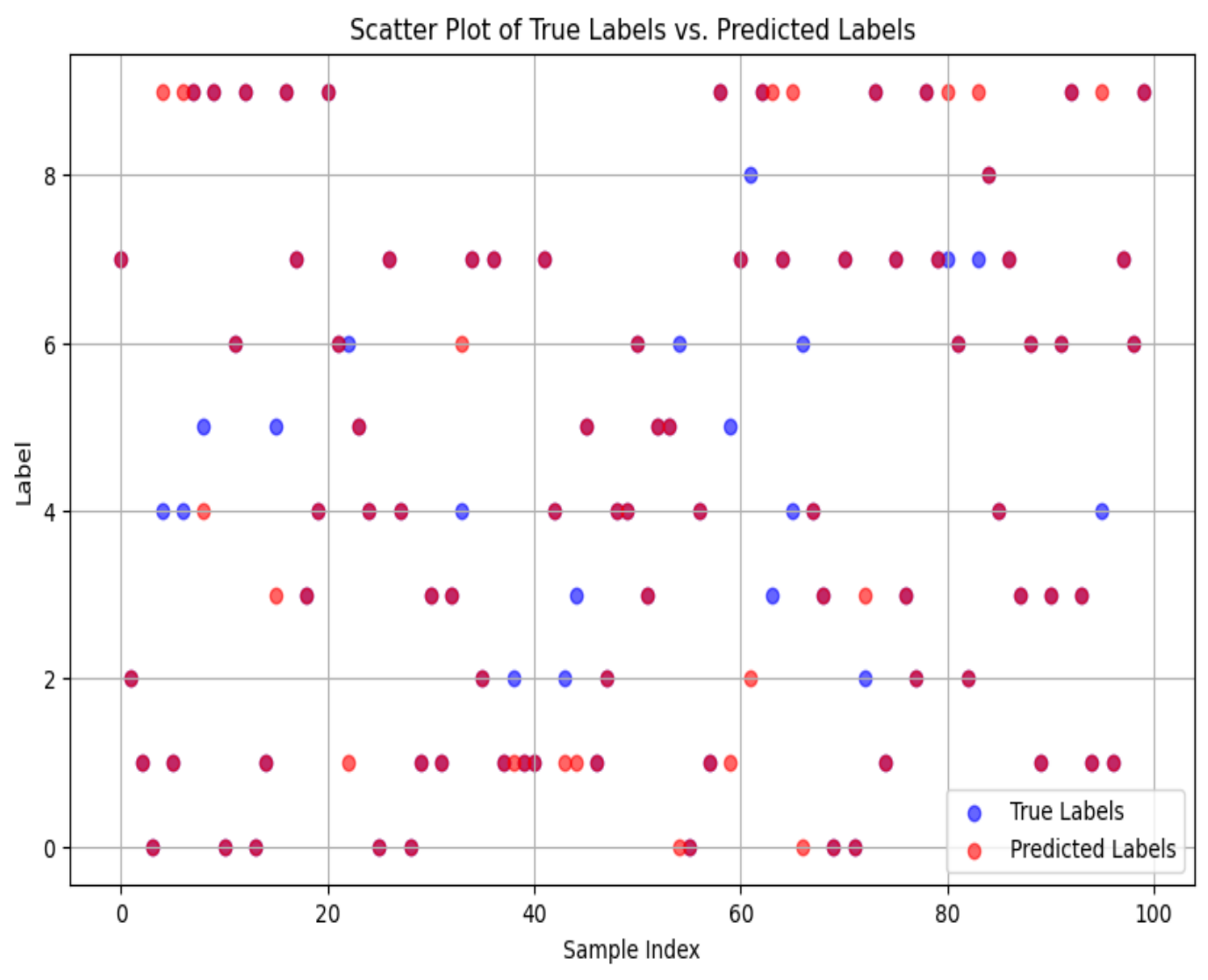
- First taking 100 data set and implement in the KNN algorithm
- The accuracy is 81.00%



Key points:

- More data More accuracy
- Tuning the Model

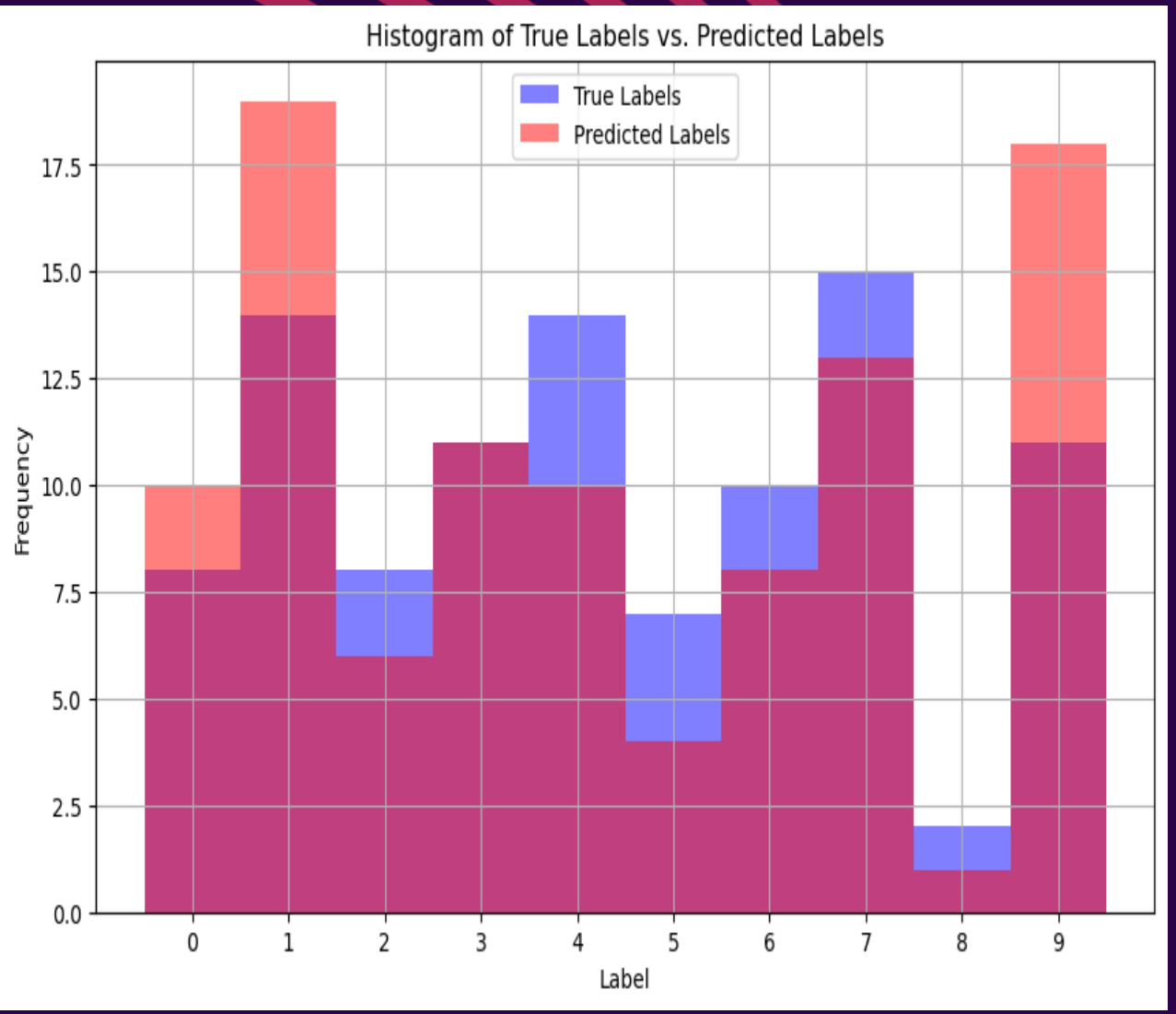
Key Visuals



Insights

- The extent to which the predicted labels (red dots) match the true labels (blue dots) directly correlates with the accuracy of the model.
- In cases where the dots are mostly overlapping, it suggests good performance, whereas significant non-overlap indicates areas where the model may need improvement.

Key Visuals



Insights

- This histogram provides a visual summary of the model's performance across different digit classes.
- The degree of overlap between true and predicted label frequencies helps diagnose where the model is performing well and where it might be making systematic errors, such as consistently over predicting certain digits or underpredicting others.

Consideration:

1. Tuning the Hyperparameter kkk:

- Try different values of kkk to find the one that gives the best performance. Generally, smaller values of kkk can lead to more complex models that may overfit, while larger values may over smooth the decision boundary.

2. Feature Scaling:

- Since k-NN is distance-based, it is crucial to scale the features so that each feature contributes equally to the distance computation.

3. Weighted k-NN:

- You can give more weight to closer neighbors when making predictions. This often improves performance, especially when the data points are not uniformly distributed.

. **Implementation:**

Weighted k-NN:

- The weighted option in the KNN class allows you to use inverse distance weighting. Neighbors closer to the query point have more influence on the prediction.

Hyperparameter Tuning:

- The loop at the end tests different values of kkk to find the one that gives the highest accuracy.

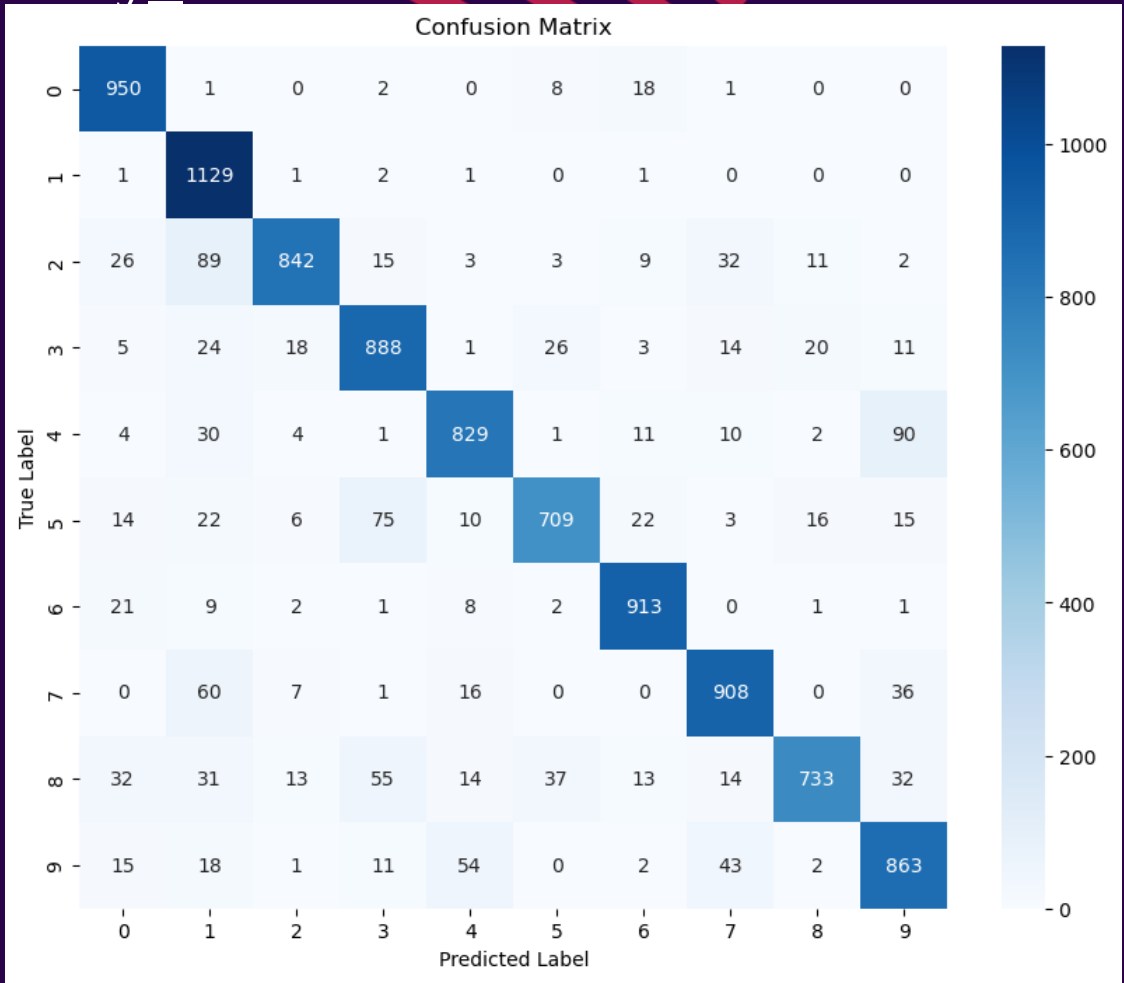
Result

- **The KNN Classifier is implemented**

Before Tuning:
Train Data Accuracy- 87%
Test Data Accuracy- 93%

After Tuning:
Train Data Accuracy-93.7%
Test Data Accuracy-95.78%

Key_visulas:



Summary:

- This confusion matrix indicates that the model generally performs well, with a high number of correct classifications (as seen in the high diagonal values).
- However, there are some areas of potential confusion where certain classes are misclassified as others.

Conclusion:

- The project successfully demonstrated handwritten digit recognition using the Naive Bayes ,Gauss Bayes and KNN algorithm.
- The manual implementation provided a deep understanding of how Naive Bayes,Gauss Bayes and KNN algorithm works, including its computational challenges with large datasets.
- Visualization of the results enabled a straightforward comparison between true and predicted labels, showcasing the effectiveness of the model and identifying areas for further optimization.
- Compared to all the algorithm Gauss bayes and KNN model giving good accuracy but the KNN is cost effective and takes a long time to run but it gives similar results.
- This implementation reduces the time spent on reading meters, bills can be generated and sent out more quickly and Improved Customer Satisfaction:
- Accurate and timely billing reduces customer complaints and improves trust in the service.
- This will improve 10% profit and reduce the labor cost upto 5 to 10 Percent.

Thank you……