Data Modeling: Star-Garage Invoices

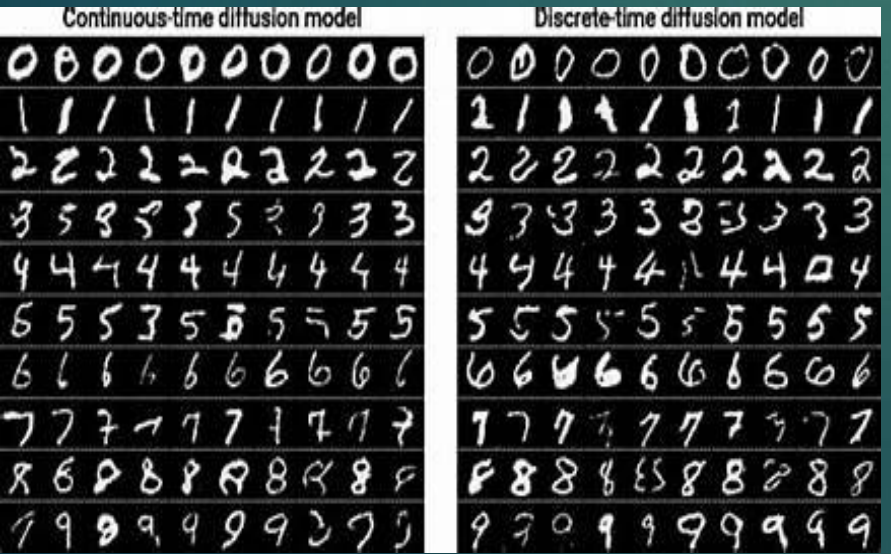
BY: SONAM MEHTA

CONTENTS:

Objective:

Develop and utilize a machine learning model trained on the **MNIST dataset** to accurately recognize handwritten digits from invoices. MNIST stands for Modified National Institute of Standards and Technology.

This will enable the automation of data extraction processes, ensuring efficiency and reducing errors in reading critical information such as invoice numbers, dates, amounts, and other numerical data.



Problem Statement:

In garages, handwritten data entry for critical details like invoice numbers, dates, payment details, product/service descriptions, VAT information, and total amounts often leads to challenges such as:

•**Misinterpretation of Handwriting**:
-Errors arise from unclear handwriting and variations in styles.

•**Financial Discrepancies**:
-Time-consuming manual entry can lead to errors in totals, taxes, or VAT, causing billing mistakes and financial losses.

These challenges emphasize the need for an effective data modeling solution to improve accuracy, reliability, and efficiency in managing garage operations.

Solution Strategies:

• Using digital systems for invoicing and pricing (e.g., Point of Sale systems).

• Employing OCR(Optical Character Recognition) tools or training machine learning models to recognize handwritten digits effectively.

Based on the identified solution ideas, I proceeded with training machine learning models to recognize handwritten digits effectively. The following steps will be undertaken:

1. Dataset Selection
2. Model Training
3. Model Evaluation

This approach addresses the challenges of misinterpretation and inefficiencies in handwritten data processing by leveraging automation through machine learning.

I

Introduction to MNIST Dataset

MNIST Dataset:

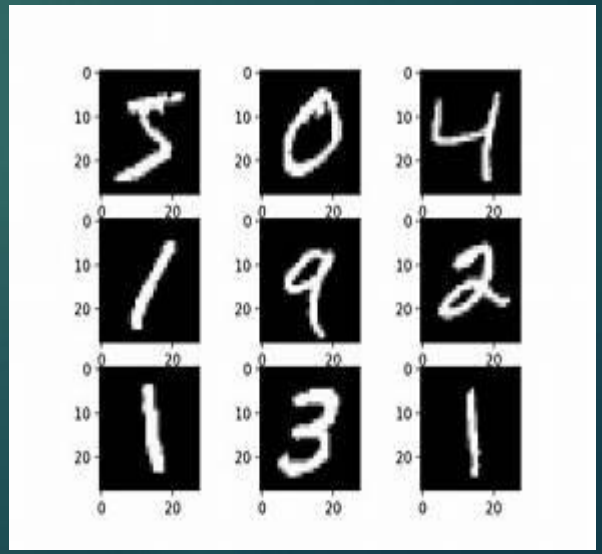
The MNIST dataset is a collection of handwritten digit images, widely used for training and evaluating machine learning models. MNIST is a large database of small, square 28×28-pixel grayscale images of handwritten single digits between 0 and 9.

Key Features:

- Each image is labelled with respective digit it represents.
- A total of 10 class of digits i.e.(from 0 to 9)

Relevance to Invoices:

Garage invoices often contain handwritten digits, making MNIST an ideal dataset for developing a model to recognize these digits.



While training a dataset with a machine learning model, in this scenario MNIST data is typically divided into two main sets: training set and testing set.

1.Training Set:

1. The training set is used to teach the model.
2. It consists of labeled data, where the model learns patterns, relationships, and structures within the input data to make predictions.
3. For example, in the MNIST dataset, the training set includes 60,000 images of handwritten digits with their respective labels (0–9).

2.Testing Set:

1. The testing set is a separate portion of the data used to evaluate the model's performance.
2. It ensures that the model generalizes well to unseen data and doesn't overfit to the training set.
3. In the MNIST dataset, the testing set includes 10,000 images that the model hasn't seen during training.

Training the Model

Training Data

The model is trained on a large subset of the MNIST dataset to learn patterns and associations.

Optimization

During training, the model's parameters are adjusted to minimize errors and improve accuracy.

Validation(Testing model)

Validation data is used to monitor model performance and prevent overfitting during training.

Model Development: Using Naïve Bayes Classifier

Data Preprocessing:

-Removing unwanted features.

-Normalizing the images to scale pixel values (e.g., [0, 1]).

- Hyperparameter Tuning

Model Architecture: Naïve Bayes Classifier

Accuracy:

During the initial training of the model on the dataset, an accuracy rate received was:

Training dataset→ **0.5938**

Testing dataset → **0.5878**

To improve the accuracy, **normalization** and **hyperparameter tuning** were applied. Results are:

Training dataset→ **0.8020**

Testing dataset→ **0.8146**

Confusion Matrix using Naïve Bayes:

This is the heatmap for test dataset.

-**Rows**: Represent the true labels (0 to 9).
-**Columns**: Represent the predicted labels (0 to 9).

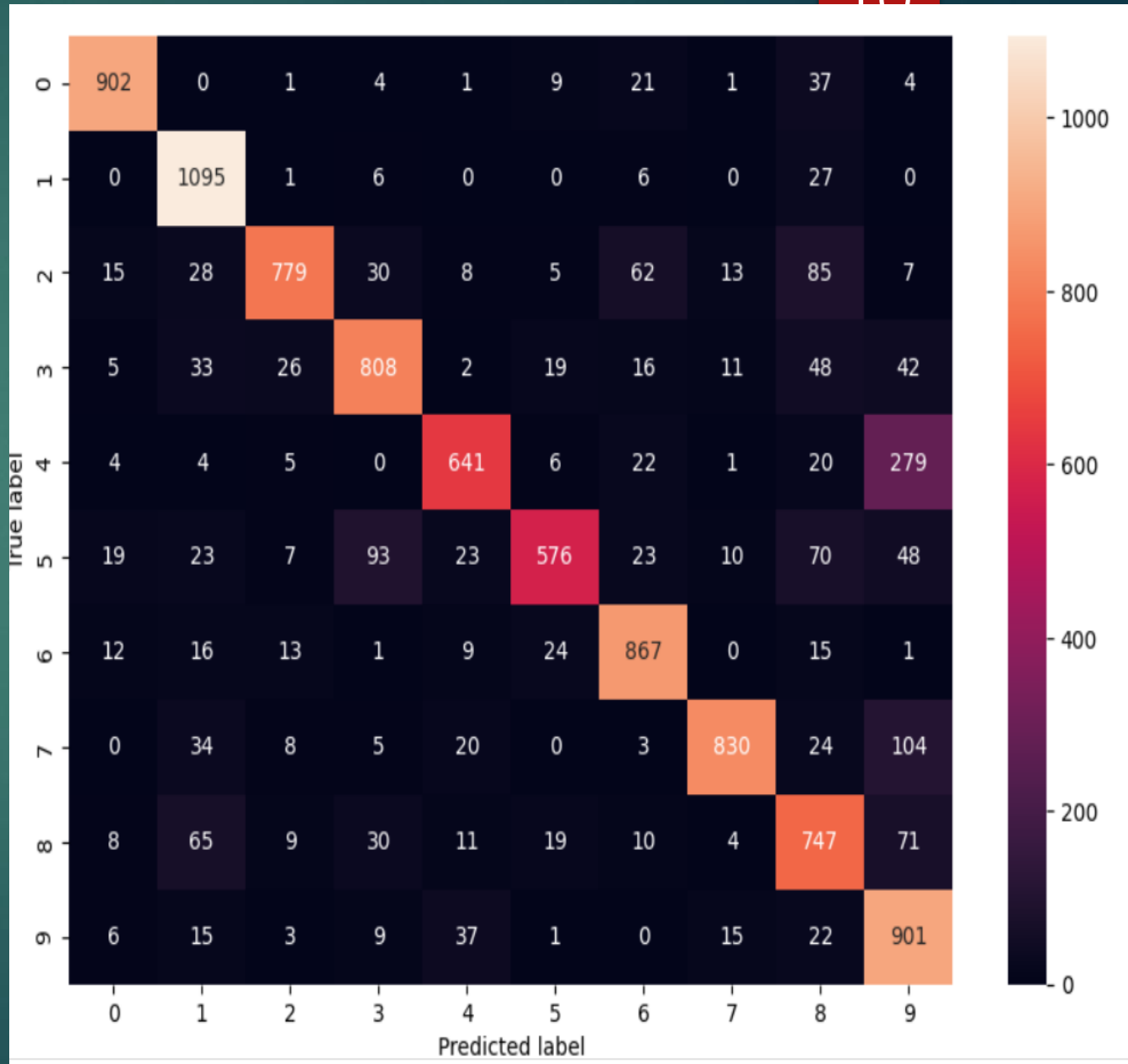
Key Observations:

1. For digits (i.e. 0,1,9) the diagonal values are higher than other values in their row, shows model perform well in identifying these objects.

For example:
0 → 902 correct predictions.
1 → 1095 correct predictions.

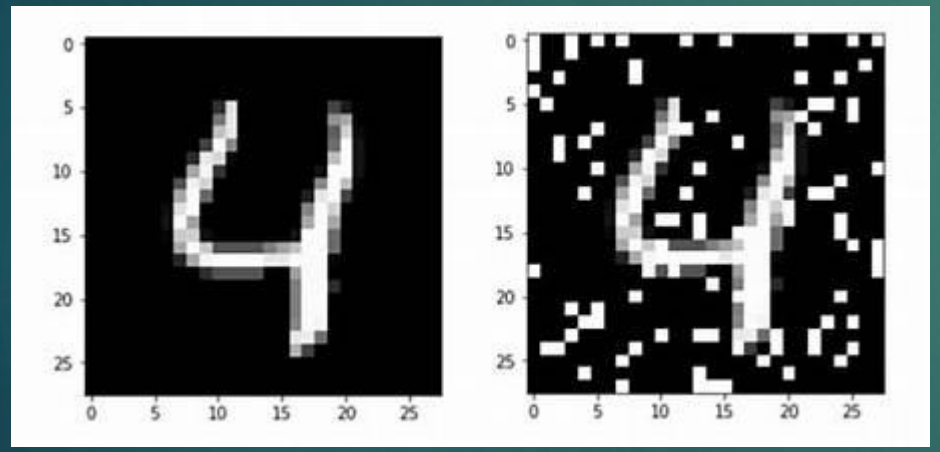
2. The off-diagonal values indicate where the model has incorrectly classified a digit.

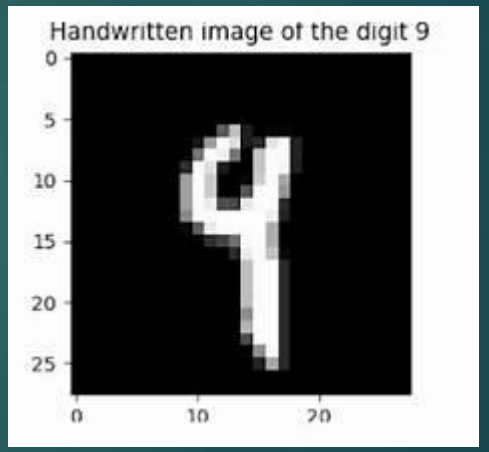
For Example:
-Digit 4 of true label is confused with digit 9 : 279 times.
-Digit 7 is confused as 9 :104 times.



Next Steps of Improvement:

- After analyzing the misclassified examples (e.g.,4 as 9, 7 as 9 etc.), it would be better to implement a more complex model might help. Next model to be tested is Non-Naïve Bayes Classifier.
- Focus on improving the model's ability to distinguish between visually similar digits, such as 4 as 9.





Model Development: Using Non-Naïve Bayes Classifier

Data Preprocessing:

-Removing unwanted features.

-Normalizing the images to scale pixel values (e.g., [0, 1]).

- Hyperparameter Tuning

Model Architecture: Non-Naïve Bayes Classifier

Accuracy:

During the initial training of the model on the dataset, an accuracy rate received was:

Training dataset→ **0.7856**

Testing dataset→ **0.7532**

To improve the accuracy, **normalization** and **hyperparameter tuning** were applied. Results are:

Training dataset→ **0.9561**

Testing dataset → **0.9485**

Confusion Matrix using Non-Naïve Bayes:

This is the heatmap for test dataset.

-**Rows**: Represent the true labels (0 to 9).
-**Columns**: Represent the predicted labels (0 to 9).

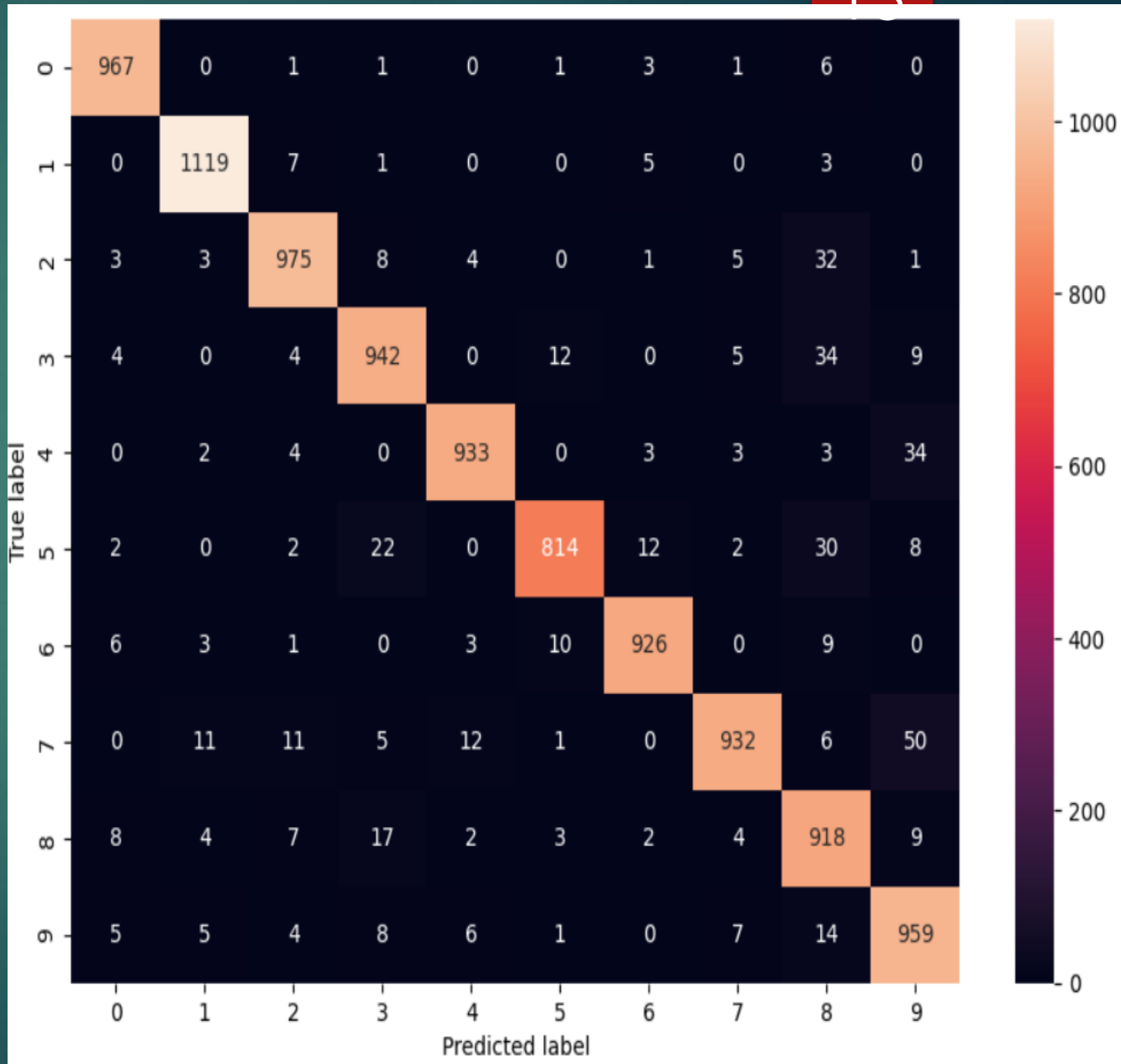
Key Observations:

1. For digits (i.e. 0,1,9) the diagonal values are higher than other values in their row, shows model perform well in identifying these objects.

For example:
0 → 967 correct predictions.
1 → 1119 correct predictions.

2. The off-diagonal values indicate where the model has incorrectly classified a digit.

For Example:
-Digit 4 of true label is confused with digit 9 : 34 times.
-Digit 7 is confused as 9 :50 times.



Comparison between Naïve and Non-Naïve Classifier based on Confusion Matrix Heatmap:

Naïve Classifier

1. Average Diagonal Dominance.
0 → 902 correct predictions.
1 → 1095 correct predictions.

2. More Misclassifications.
Digit 4 of true label is confused with digit 9 : 279 times.
-Digit 7 is confused as 9 :104 times.

3. Average Accuracy(**0.8146**).

Non-Naïve Classifier

1. Improved Diagonal Dominance.
0 → 967 correct predictions.
1 → 1119 correct predictions.

2. Reduced Off-Diagonal Values (Misclassifications)
Digit 4 of true label is confused with digit 9 : 34 times.
-Digit 7 is confused as 9 :50 times

3. Good Accuracy(**0.9485**) because more true labels align with predicted labels.

Conclusion:

Key Takeaway:

- While Naive Bayes is a good baseline, the non-Naive Bayes model (e.g., CNN or similar) is better suited for digit recognition tasks in real-world garage invoices due to its ability to handle complexities like noisy inputs and diverse handwriting styles.

Next Steps:

- Integrate the non-Naive Bayes model into an invoice processing pipeline.
- Collect and label additional real-world garage invoice samples for fine-tuning.
- Explore further optimization to balance accuracy and computational efficiency for large-scale deployments.

THANK YOU