May 2025

Handwritten Digit Recognition using Machine Learning

Dipok Deb
PhD Student, Big Data Analytics, UCF, di876893@ucf.edu

Handwritten Digit Recognition using Machine Learning

Dipok Deb
STA 6366
Statistics and Data Science Department
University of Central Florida

***Abstract*—Handwritten Digit Recognition (HDR) remains a fundamental benchmark in pattern recognition and machine learning due to its practical applications and inherent classification challenges posed by diverse handwriting styles. This study investigates and compares two classical statistical classifiers—Gaussian Naive Bayes (GNB) and Linear Discriminant Analysis (LDA)—to recognize the digits from the MNIST dataset. Both models assume underlying normality in feature distributions and offer computational efficiency, making them suitable for high-dimensional input such as image pixels. Using 60,000 training and 10,000 test samples, we evaluate model performance through accuracy, precision, recall, F1 score, and confusion matrices. The results reveal that while GNB achieves moderate accuracy (55.58%), LDA significantly outperforms it with an accuracy of 87.30%, demonstrating superior capability in distinguishing visually similar digits. Our analysis further highlights the limitations of GNB's independence assumption and underscores LDA's strength in capturing shared variance across classes. These findings reinforce the effectiveness of LDA as a robust baseline for HDR tasks, especially when interpretability and computational simplicity are desired.**

***Index Terms*—Machine Learning, Handwritten Digit Recognition, MNIST Dataset, Multiclass Classification, Image Classification.**

I. Introduction

Handwritten digit recognition (HDR) is a key problem in the domains of pattern recognition and machine learning [1]. It focuses on categorizing handwritten digits into one of ten classes (0-9) using pixel data from images. This task holds significant real-world value in applications such as postal mail sorting, automatic bank check processing, and digitizing historical records. Despite being conceptually straightforward, HDR presents several challenges due to the variability in handwriting styles, line thickness, orientation, and the presence of noise in the images.

Over time, the application of machine learning methods to handwritten digit recognition has advanced considerably. These methods seek to learn distinguishing features from image data and build accurate models for digit classification. From a statistical point of view, the goal of this project is to evaluate and analyze a machine learning classifier to accurately recognize handwritten digits from the MNIST dataset. Gaussian Naive Bayes (GNB) and Linear Discriminant Analysis (LDA) are two established, computationally efficient classifiers based on statistical principles. The MNIST dataset contains grayscale pixel data, which can be treated as continuous variables. Gaussian Naive Bayes, a variant of Naive Bayes, assumes that the features follow a Gaussian distribution, making it a suitable option for continuous data like pixel intensities. Similarly, LDA assumes a normal distribution for the data, making it appropriate for classifying digit images based on pixel values. This project investigates the performance of both Gaussian Naive Bayes and LDA for handwritten digit recognition, offering a detailed comparison of the two classifiers.

The rest of the project is structured as follows: Section II covers methodology, Section III outlines data analysis, and Section IV encompasses the conclusion of this project.

II. Methodology

A. Dataset

The most widely used dataset for handwritten digit recognition is the MNIST dataset [2]. The MNIST dataset is a collection of 70,000 handwritten digits (0-9), with each image being 28×28 pixels.

B. Methods

In this project, we will utilize Gaussian Naive Bayes (GNB) and Linear Discriminant Analysis (LDA) classifiers to identify all the digits in the MNIST dataset.

1) Gaussian Naive Bayes (GNB): The Gaussian Naive Bayes (GNB) classifier [3] is a probabilistic model based on Bayes' theorem, which assumes that the features $X_1, X_2, \ldots, X_n$ are conditionally independent given the class label Y, and that each feature follows a Gaussian distribution.

2) Linear Discriminant Analysis (LDA): Linear Discriminant Analysis (LDA) is a linear classifier that assumes that the features $X_1, X_2, \ldots, X_n$ are normally distributed for each class Y, and all classes share the same covariance matrix [4].

The details of these two methods has been described in Appendix 1.

C. Evaluation Metrics

To evaluate the performance of the model, we will use various evaluation metrics such as accuracy, precision, recall and f1 score. We will also utilize the confusion matrix to identify misclassified examples.

III. Data Analysis

A. Dataset Exploration

The MNIST dataset has become a standard for evaluating and comparing various machine learning algorithms, largely due to its simplicity and accessibility. This dataset features the following attributes:

- **Number of Instances**: 70,000 images
- **Number of Attributes**: 784 (28×28 pixel grid)
- **Target**: The column indicates the digit (0-9) associated with each handwritten image.
- **Pixel Values**: Pixels 1-784, where each pixel value (0-255) corresponds to the grayscale intensity of a pixel in the image.

The dataset is split into two key subsets:

1) **Training Set**: Includes 60,000 images with labels, typically used for training machine learning models.
2) **Test Set**: Comprises 10,000 images and their respective labels, used to assess the performance of trained models.

Fig. 1 illustrates sample images of handwritten digits from the MNIST dataset. The labels above each image indicate the corresponding digit (0 to 9). Also, Fig. 2 represents the distribution of the training data of MNIST dataset. The pie chart visually displays the number of images for each digit (0-9). The image count for each digit varies slightly, with digit 1 having the largest number of images (6742), and digit 5 having the smallest number of images (5421). From this distribution we can understand that the training dataset is almost balanced. Moreover, the pie chart in Fig. 3 provides a visual representation of 10,000 test images that are distributed across the 10 digits. Each slice is labeled with the corresponding digit and the number of images for that digit. The distribution is fairly uniform, with digit 1 having the highest number of images (1134) and digit 5 having the lowest number of images (891).

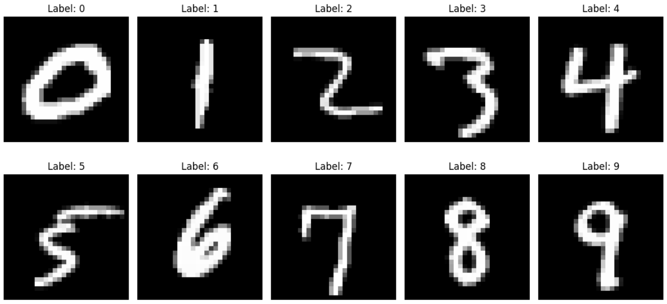


Fig. 1. Sample Image of digits in MNIST Dataset.

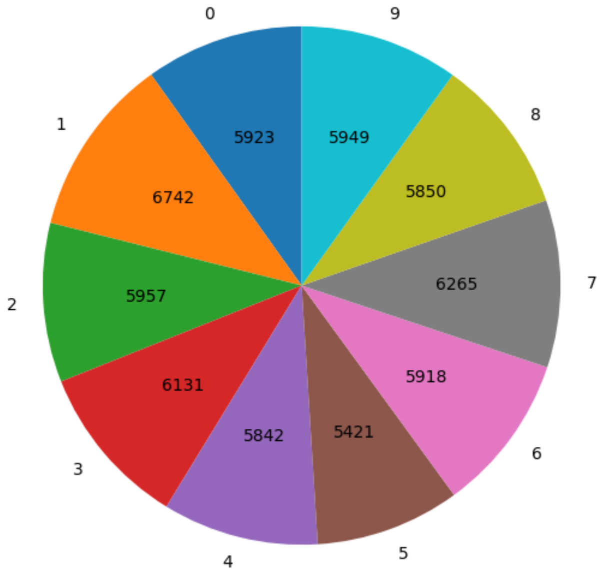


Fig. 2. Training Set Distribution (Image Counts).

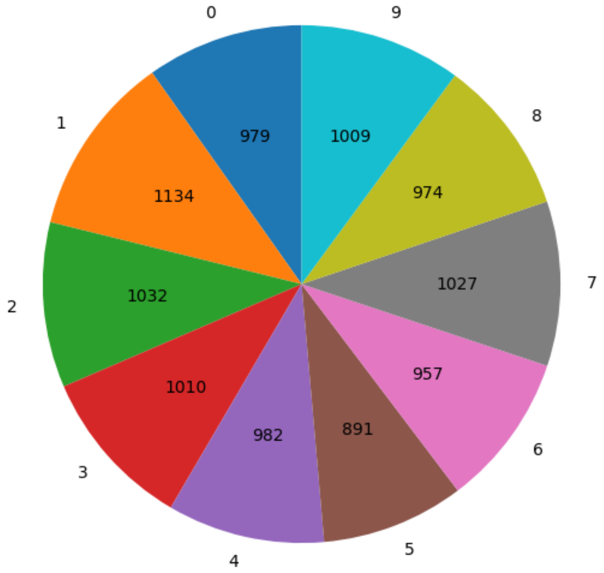


Fig. 3. Test Set Distribution (Image Counts).

B. Model Training

We trained the Gaussian Naive Bayes (GNB) and Linear Discriminant Analysis (LDA) classifiers using 60,000 images from the MNIST dataset, referred to as the training data. For each image in the this dataset, the input consists of 784 features (the pixel values of the 28x28 image), denoted as $X_1, X_2, \ldots, X_{784}$, and the class Y represents all class of 10 digit labels (0-9).

Gaussian Naive Bayes: For Gaussian Naive Bayes assuming that this input features are conditionally independent given the class label, and that they follow a Gaussian distribution. The posterior probability of a class Y given the input features $X_1, X_2, \ldots, X_{784}$ is given by Bayes' theorem:

$$P(Y|X_1, \ldots, X_{784}) = \frac{P(X_1, \ldots, X_{784}|Y)P(Y)}{P(X_1, \ldots, X_{784})} \quad (1)$$

Since we assume a Gaussian distribution for each pixel given the class, the likelihood for each feature (pixel) is:

$$P(X_i|Y) = \frac{1}{\sqrt{2\pi\sigma_Y^2}} \exp\left(-\frac{(X_i - \mu_Y)^2}{2\sigma_Y^2}\right) \quad (2)$$

The overall likelihood is computed by multiplying the individual likelihoods of all features:

$$P(X_1, \ldots, X_{784}|Y) = \prod_{i=1}^{784} P(X_i|Y) \quad (3)$$

The model predicts the class $\hat{Y}$ that maximizes the posterior probability using the following equation:

$$\hat{Y} = \arg\max_{Y} P(Y|X_1, \ldots, X_{784}) \quad (4)$$

In this project, we have used `GaussianNB()` as the Gaussian Naive Bayes (GNB) classifier from the Scikit-learn library. The default hyper-parameters include a uniform prior distribution for all classes (`priors=None`), meaning that each class is assumed to be equally likely. Additionally, the `var_smoothing` parameter, which controls the amount of variance added to prevent division by zero during likelihood calculations, is set to a default value of $1e{-}9$. These default settings ensure numerical stability while maintaining simplicity in model assumptions.

Linear Discriminant Analysis (LDA): For LDA, we assume that the input features are normally distributed for each class and aims to find a linear combination of features that best separates the classes. Also assumes that all classes share the same covariance matrix. LDA models the conditional distribution of the feature vector X given the class Y as a multivariate Gaussian:

$$\begin{aligned} P(X|Y=k) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \\ &\times \exp\left(-\frac{1}{2}(X-\mu_k)^T\Sigma^{-1}(X-\mu_k)\right) \end{aligned} \quad (5)$$

Where:

- X is the 784-dimensional vector representing the pixel values.
- n is the number of features (784 in this case).

LDA classifies a new instance by computing the discriminant function using equation

$$\delta_k(X) = X^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log P(Y=k) \quad (6)$$

for each class and selecting the class with the highest value utilizing equation

$$\hat{Y} = \arg\max_{k} \delta_k(X) \quad (7)$$

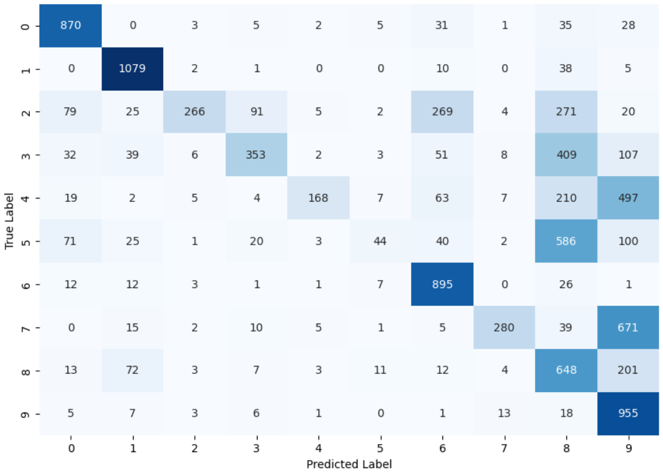
In this project, we utilized the **LinearDiscriminantAnalysis()** classifier from the Scikit-learn library. The default hyperparameters for LDA include the `solver`, which is set to `svd` by default. This solver is based on Singular Value Decomposition and is efficient for large datasets as it does not compute the covariance matrix. The `shrinkage` parameter is set to `None`, meaning no regularization is applied unless the `lsqr` or `eigen` solvers are used. Additionally, the `priors` parameter is also set to `None`, implying that the class priors are inferred from the training data. By default, `n_components` is set to `None`, meaning that the dimensionality reduction retains all components up to the number of classes minus one. The `store_covariance` parameter is `False`, which indicates that the covariance matrix is not stored. Lastly, the `tol` parameter, which controls the tolerance for rank estimation in the `svd` solver, is set to `1e-4`. These default settings make LDA suitable for efficient performance, particularly with large datasets, while still allowing flexibility for tuning the model.

C. Result and Discussion

In this project, we utilized a Gaussian Naive Bayes (GNB) and Linear Discriminant Analysis (LDA) classifier for the task of multi-class classification, specifically focusing on the recognition of handwritten digits from the MNIST dataset. In this section, we will describe the outcomes of both models and also will analyse their performance.

The confusion matrix in the Fig. 4 presents a detailed view of the Gaussian Naive Bayes model's performance, showing the number of correct and incorrect predictions for each digit (0-9). The true class labels are placed along the vertical axis, and the predicted class labels are along the horizontal axis. The diagonal values represent the correct predictions, while the off-diagonal values indicate misclassifications. The model shows high accuracy for digits such as 0, 1, 6, and 9, with a large number of correct predictions, as indicated by the dark diagonal cells (e.g., 870 correct predictions for 0 and 1079 for 1). However, the model struggles with several digits, particularly 2, 5, and 8, where significant misclassifications occur. For instance, 271 instances of 2 are misclassified as 8, and 586 instances of 5 are predicted as 8, reflecting the difficulty in distinguishing between these digits. Similarly, 7 is often confused with 9, with 671 instances of misclassification. These off-diagonal values suggest that the classifier is likely challenged by the similar shapes of these digits. The overall pattern in the confusion matrix indicates that while the Gaussian Naive Bayes model is effective for certain digits, it faces difficulties in recognizing others due to overlapping features, especially with digits that share similar structures or pixel patterns.

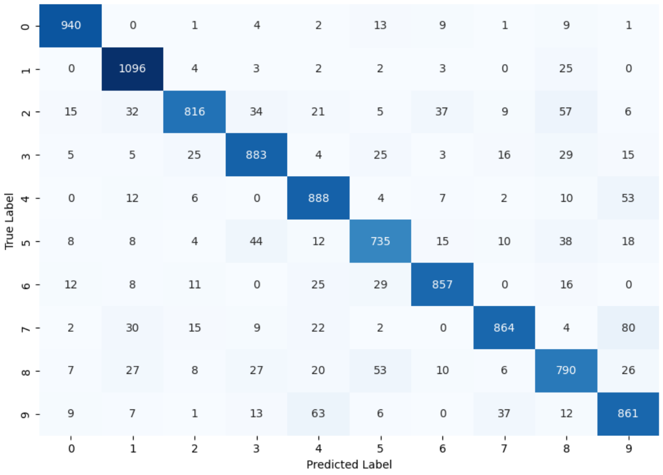
On the other hand, the confusion matrix in Fig. 5 from Linear Discriminant Analysis (LDA) classifier shows high accuracy for several digits, especially 0, 1, and 4, with 940, 1096, and 888 correct predictions, respectively. This is evidenced by the strong diagonal elements representing correct classifications. However, the matrix also highlights some misclassification issues. For example, the digit 2 is often



True \ Predicted	0	1	2	3	4	5	6	7	8	9
0	870	0	3	5	2	5	31	1	35	28
1	0	1079	2	1	0	0	10	0	38	5
2	79	25	266	91	5	2	269	4	271	20
3	32	39	6	353	2	3	51	8	409	107
4	19	2	5	4	168	7	63	7	210	497
5	71	25	1	20	3	44	40	2	586	100
6	12	12	3	1	1	7	895	0	26	1
7	0	15	2	10	5	1	5	280	39	671
8	13	72	3	7	3	11	12	4	648	201
9	5	7	3	6	1	0	1	13	18	955

Fig. 4. Confusion Matrix of Gaussian Naive Bayes.

confused with 1 (32 instances) and 8 (57 instances), while 8 is misclassified as 3 (27 instances) and 5 (53 instances). Another notable confusion occurs between 9 and 4, with 63 instances of 4 being classified as 9. These misclassifications suggest that certain digits share similar shapes or pixel distributions, making them difficult for the LDA model to distinguish. Overall, LDA performs well than Gaussian Naive Bayes for many digits.



True \ Predicted	0	1	2	3	4	5	6	7	8	9
0	940	0	1	4	2	13	9	1	9	1
1	0	1096	4	3	2	2	3	0	25	0
2	15	32	816	34	21	5	37	9	57	6
3	5	5	25	883	4	25	3	16	29	15
4	0	12	6	0	888	4	7	2	10	53
5	8	8	4	44	12	735	15	10	38	18
6	12	8	11	0	25	29	857	0	16	0
7	2	30	15	9	22	2	0	864	4	80
8	7	27	8	27	20	53	10	6	790	26
9	9	7	1	13	63	6	0	37	12	861

Fig. 5. Confusion Matrix of Linear Discriminant.

Table I provides a detailed comparison between the Gaussian Naive Bayes and LDA models across overall Accuracy, Precision, Recall, and F1 Score. The results clearly show that the LDA model surpasses Gaussian Naive Bayes in all metrics. LDA achieves a significantly higher accuracy of 0.8730 compared to 0.5558 for Gaussian Naive Bayes, indicating that it makes more correct predictions overall. Additionally, LDA shows superior precision (0.8743), meaning it is better at reducing false positives compared to Gaussian Naive Bayes (0.6917). In terms of recall, LDA also excels with a score of 0.8730, demonstrating its effectiveness in correctly identifying positive instances, whereas Gaussian Naive Bayes lags at 0.5558. Finally, the F1 score for LDA is 0.8727, highlighting its better balance between precision and recall, as opposed to Gaussian Naive Bayes, which scores 0.5170. These results suggest that LDA is a more robust model for this classification task, delivering more accurate and balanced performance compared to Gaussian Naive Bayes.

The accuracy, precision, recall, and F1 score for both models regarding each digit are shown in Appendix 1.

TABLE I
OVERALL COMPARISON OF METRICS FOR GAUSSIAN NAIVE BAYES (GNB) AND LINEAR DISCRIMINANT ANALYSIS (LDA)

Metric	GNB	LDA
Accuracy	0.5558	0.8730
Precision	0.6917	0.8743
Recall	0.5558	0.8730
F1 Score	0.5170	0.8727

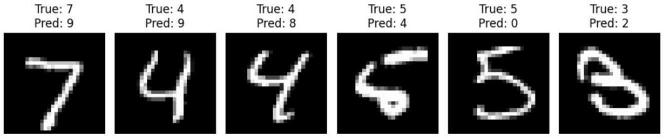


Fig. 6. Some Misclassified Images from Gaussian Naive Bayes Classifier.

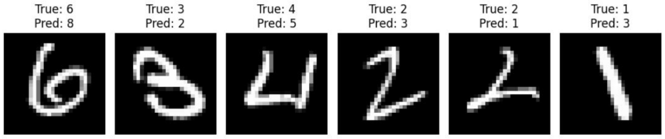


Fig. 7. Some misclassified images from Linear Discriminant Classifier.

Fig. 6 and Fig. 7 provide examples of misclassified digits by the Gaussian Naive Bayes and LDA classifiers, respectively. Both classifiers are using pixel intensity values as input features, and their misclassifications suggest that they may struggle to capture subtle variations in pixel patterns, particularly when digits have similar shapes or overlapping features. For instance, in Fig. 6, the Gaussian Naive Bayes classifier frequently misclassifies digits that have similar pixel structures, such as 7 being confused with 9, and 5 being misclassified as 0. This indicates that the Gaussian Naive Bayes model, which assumes conditional independence of pixels, may not be effectively capturing the relationships between neighboring pixels, leading to confusion when two digits share similar pixel patterns. Moreover, In Fig. 7, the LDA classifier also shows misclassifications, such as 6 being predicted as 8 and 2 being confused with 1. This suggests that while LDA can better capture the overall variance and relationships in pixel values compared to Gaussian Naive Bayes, it still struggles when the pixel intensities of two digits overlap or when slight variations in pixel patterns occur.

IV. Conclusion

In this project, we explored the effectiveness of two machine learning classifiers, Gaussian Naive Bayes (GNB) and Linear Discriminant Analysis (LDA), for the task of handwritten digit recognition using the MNIST dataset. Our findings indicate that while both models can classify certain digits with high accuracy, LDA consistently performs better than Gaussian Naive Bayes across all key evaluation metrics, including accuracy, precision, recall, and F1 score. Overall, this study concludes that while Gaussian Naive Bayes is a simpler model, Linear Discriminant Analysis is better suited for the handwritten digit recognition task due to its ability to capture more complex relationships between pixel features.

References

[1] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," Pattern Recognition, vol. 36, no. 10, pp. 2271-2285, 2003.

[2] Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits," 2012. [Online]. Available: http://yann.lecun.com/exdb/mnist/.

[3] G. GeeksforGeeks, "Naive Bayes Classifiers", GeeksforGeeks, 2020. [Online]. Available: https://www.geeksforgeeks.org/naive-bayes-classifiers/. [Accessed: 2024-10-07].

[4] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," AI Communications, vol. 30, no. 2, pp. 169-190, 2017. [Online]. Available: https://www.iospress.nl/journal/ai-communications.

Appendix 1:

Gaussian Naive Bayes Model: The Gaussian Naive Bayes (GNB) classifier is a probabilistic model based on Bayes' theorem, which assumes that the features $X_1, X_2, \ldots, X_n$ are conditionally independent given the class label Y, and that each feature follows a Gaussian distribution.

Bayes' Theorem: For a classification problem, Bayes' Theorem can be written as:

$$P(Y|X_1, X_2, \ldots, X_n) = \frac{P(X_1, X_2, \ldots, X_n|Y)P(Y)}{P(X_1, X_2, \ldots, X_n)} \tag{8}$$

Where:

- $P(Y|X_1, X_2, \ldots, X_n)$ is the posterior probability of class Y given the feature vector $X_1, X_2, \ldots, X_n$.
- $P(X_1, X_2, \ldots, X_n|Y)$ is the likelihood, which is the probability of observing the features given the class Y.
- $P(Y)$ is the prior probability of class Y.
- $P(X_1, X_2, \ldots, X_n)$ is the evidence, normalizing the probabilities.

Gaussian Likelihood: The likelihood $P(X_i|Y)$ for each feature X_i given the class Y is modeled as a Gaussian distribution:

$$P(X_i|Y) = \frac{1}{\sqrt{2\pi\sigma_Y^2}} \exp\left(-\frac{(X_i - \mu_Y)^2}{2\sigma_Y^2}\right) \tag{9}$$

Where:

- X_i is the value of the i-th feature.
- μ_Y and σ_Y are the mean and standard deviation of the i-th feature for class Y.

The overall likelihood $P(X_1, X_2, \ldots, X_n|Y)$ is the product of the likelihoods for all features (assuming independence):

$$P(X_1, X_2, \ldots, X_n|Y) = \prod_{i=1}^{n} P(X_i|Y) \tag{10}$$

Prediction: The model predicts the class Y that maximizes the posterior probability:

$$\hat{Y} = \arg\max_{Y} P(Y|X_1, X_2, \ldots, X_n) \tag{11}$$

Linear Discriminant Analysis (LDA): Linear Discriminant Analysis (LDA) is a linear classifier that assumes that the features $X_1, X_2, \ldots, X_n$ are normally distributed for each class Y, and all classes share the same covariance matrix.

LDA models the conditional probability $P(X_1, X_2, \ldots, X_n|Y)$ of the features given the class Y as a multivariate Gaussian distribution:

$$\begin{aligned} P(X|Y=k) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \\ &\times \exp\left(-\frac{1}{2}(X-\mu_k)^T\Sigma^{-1}(X-\mu_k)\right) \end{aligned} \tag{12}$$

Where:

- $X = [X_1, X_2, \ldots, X_n]$ is the feature vector.
- μ_k is the mean vector for class k.
- Σ is the shared covariance matrix for all classes.

- $|\Sigma|$ is the determinant of the covariance matrix.
- n is the number of features.

Discriminant Function: For each class $Y = k$, the discriminant function is:

$$\delta_k(X) = X^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log P(Y = k) \quad (13)$$

Where $\log P(Y = k)$ is the prior probability of class $Y = k$.

Prediction: The predicted class Y is the one that maximizes the discriminant function:

$$\hat{Y} = \arg\max_k \delta_k(X) \quad (14)$$

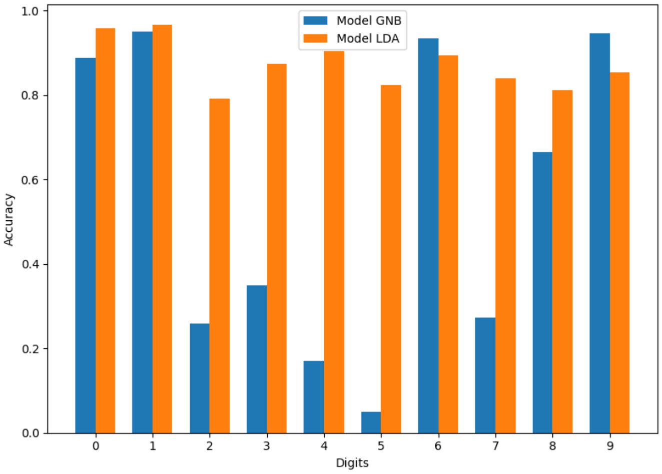


Fig. 8. Accuracy Comparison for each Digits recognition (Model GNB vs Model LDA).

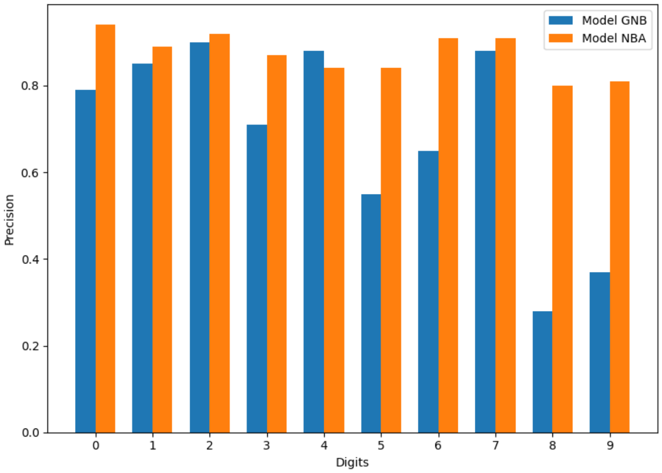


Fig. 9. Precision Comparison for each Digits recognition (Model GNB vs Model LDA).

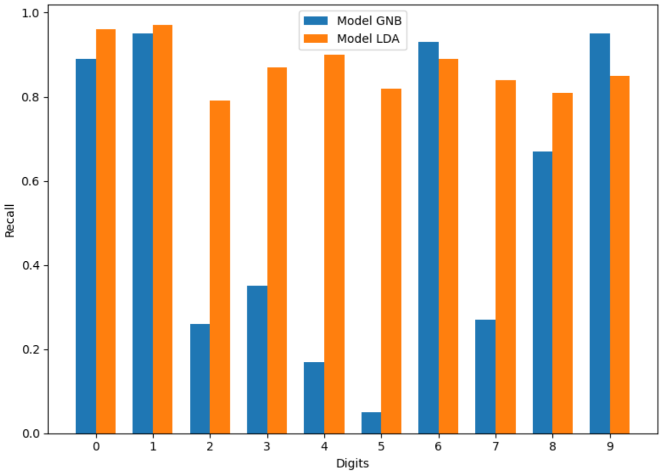


Fig. 10. Recall Comparison for each Digits recognition (Model GNB vs Model LDA).

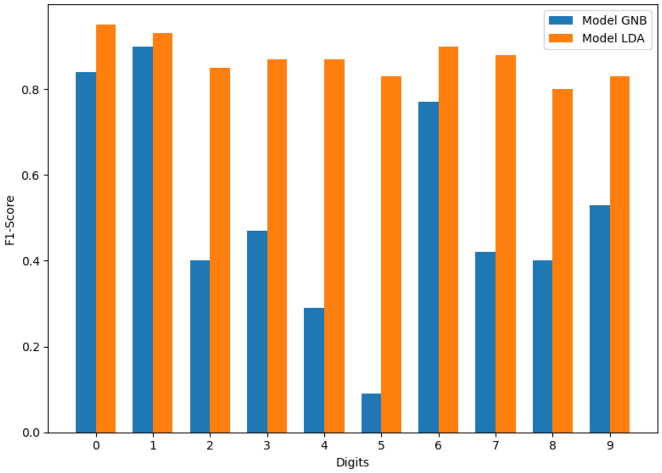


Fig. 11. F1 score Comparison for each Digits recognition (Model GNB vs Model LDA).