Report

The MNIST dataset classification

The MNIST dataset consists of 60,000 training images and 10,000 testing images, each of which is a 28x28 gray-scale image of a handwritten digit (0-9). The goal of the task is to train a model that can accurately classify the digits in the testing set. The pixel values in the MNIST dataset range from 0 to 255, where 0 represents a white pixel and 255 represent a black pixel. The gray-scale values in between represent different shades of gray.

Feature Extraction methods we employed

- **Raw Pixel values**

The MNIST is actually a huge dataset and we had to increase the use of the numpy library since the computations took about hours in a single iteration let alone the required minimum accuracy ranges! (We have tried using nested loops on the array multiple times and the result was just a non-stop computation, even we can’t compute the sigmoid function using math library)

We used the raw pixel values as a feature extraction method by treating the pixel values of an image as features. In this approach, each pixel value is treated as a separate feature, and the entire image is represented as a vector of pixel values.

  - It turned out to be computationally expensive.
  - We got a maximum accuracy of 81.49% using Naïve Bayes and 83.01% using Logistic regression.

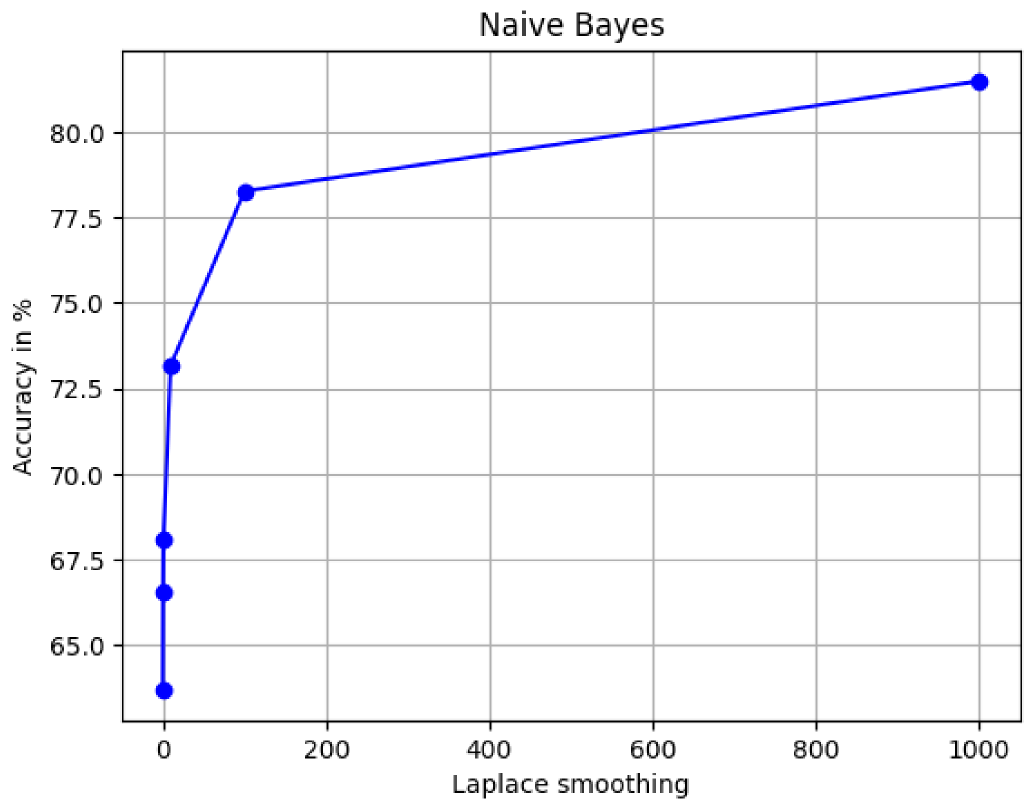


Figure1: Naïve Bayes Accuracy for different values of Laplace Smoothing (raw pixels feature)

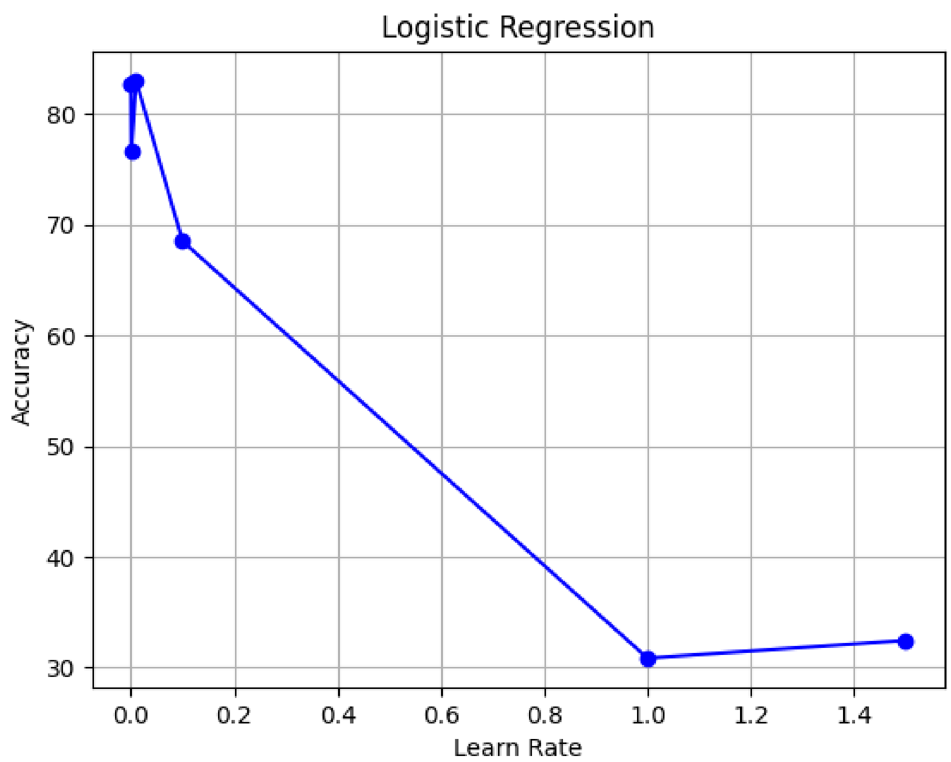


Figure 2: Logistic regression accuracy values for different values of learning rates (raw pixels feature, 100 iterations)

- **Normalized pixel intensity**

This involved normalizing the pixel values of an image to a common scale. This can be useful to reduce the impact of variations in lighting and contrast on the image, and to ensure that the features have similar ranges and distributions.

This is implemented by dividing each pixel value by the maximum pixel value in the image (which is 255). This results in pixel values that range from 0 to 1, with 0 representing the minimum intensity and 1 representing the maximum intensity.

- This was able to reduce the computational complexity
- Reduced sensitivity to lighting and contrast: Normalizing pixel intensity reduced the impact of variations in lighting and contrast on the image.
- We got a maximum accuracy of 77.92% using Naïve Bayes and 88.55% using Logistic regression.

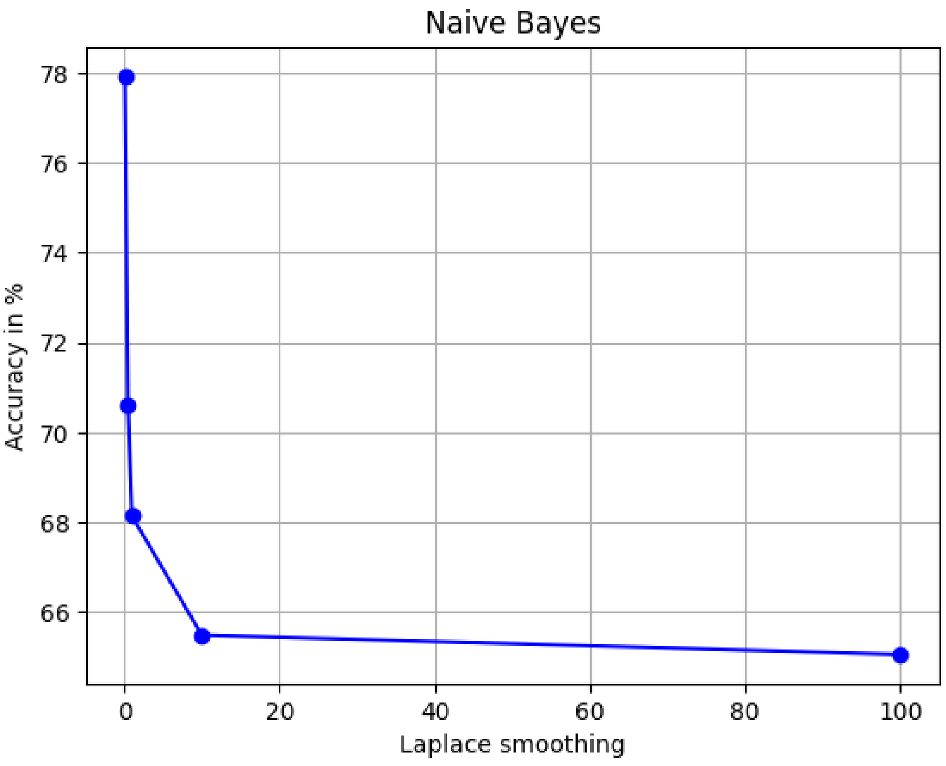


Figure 3: Naïve Bayes Accuracy for different values of Laplace Smoothing (normalized pixel intensity feature)

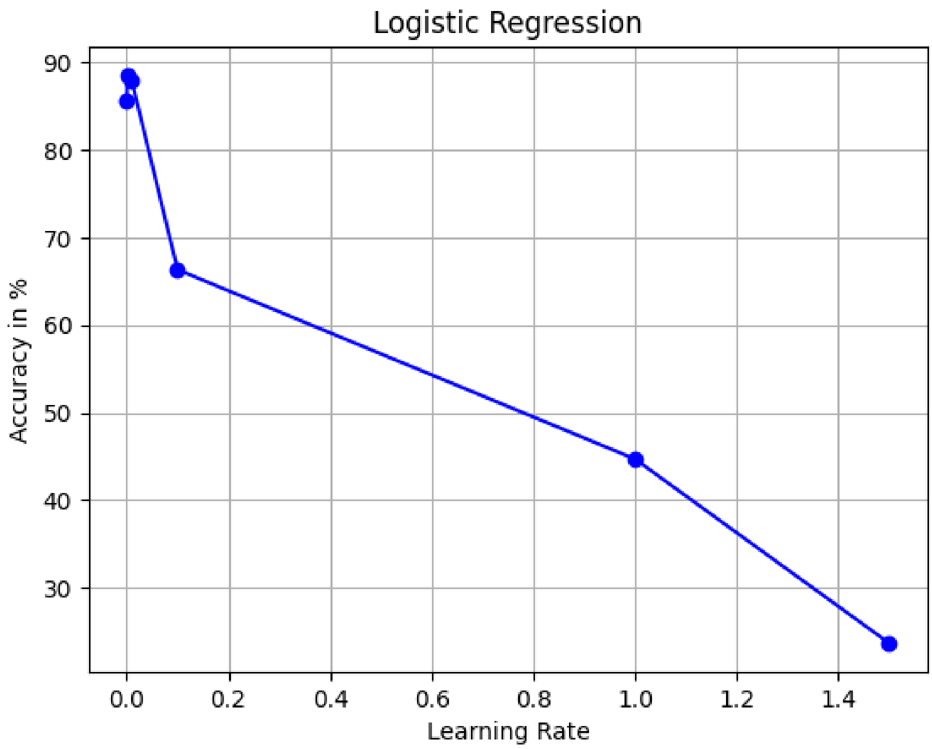
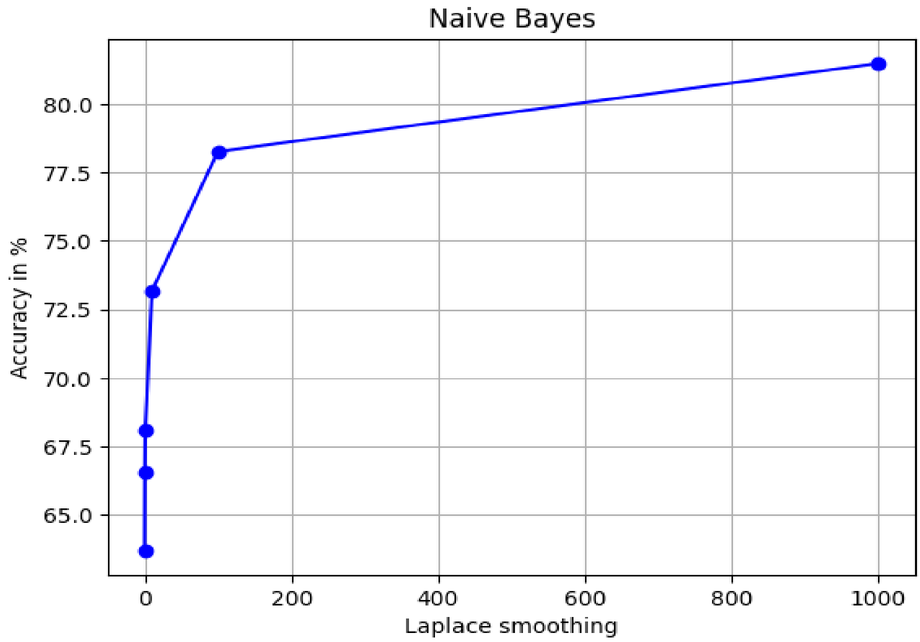


Figure 4: Logistic regression accuracy values for different values of learning rates (normalized pixel intensity feature, 100 iterations)

- **Average Pixel Intensity**

This involved computing the average pixel intensity of an image as a feature. It is calculated by finding the mean of the pixel values in an image.

- This is similar to the former two.
- Average Pixel Intensity produces a low-dimensional feature space (i.e., a single feature value per image), which was expected to reduce the dimensionality of the data and avoid over fitting.
- We got a maximum accuracy of 81.49% using Naïve Bayes and 83.01% using Logistic regression.



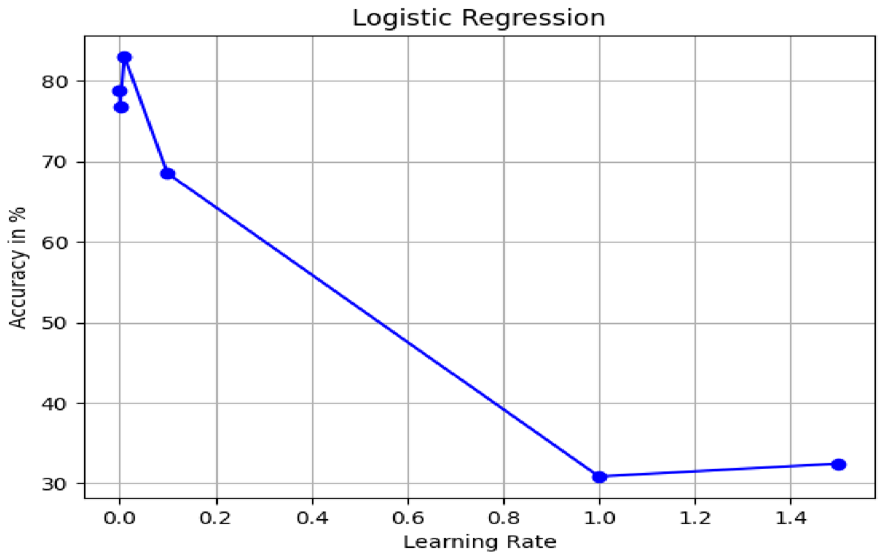


Figure 5 & 6: Naïve Bayes Accuracy for different values of Laplace Smoothing and Logistic regression accuracy values for different values of learning rates (average pixel intensity feature, 100 iterations)