Pattern Recognition Assignment

1. MNIST Digits Classification

Introduction:

MNIST ("Modified National Institute of Standards and Technology") is the classic dataset of handwritten images and it has served as the basis for benchmarking classification algorithms.It has a training set of 60,000 examples, and a test set of 10,000 examples.MNIST is a subset of the larger set available from NIST.(Dataset link: http://yann.lecun.com/exdb/mnist/)

The original black and white (bilevel) images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. the images were centered in a 28x28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field.

Objective:

The given problem statement requires to build a Bayes classifier for 2 tasks:

i) distinguish between '0' and '1' digits

ii) distinguish between '3' and '8' digits

Next step is to calculate the classification accuracy of the model and plot the ROC curves between GAR (Genuine Acceptance Rate) and FAR(False Acceptance Rate)

Theory:

Naive Bayes Classifier assumes that probability of each pixel is a Gaussian distribution and the probability of each digit is equal.Each image in MNIST dataset is a 28x28 pixels but for our purpose we are converting the images in a single flat array of 784 pixels. Knowing that each pixel from the array can take values between 0–255, it appears like continuous data. To ease the computation, we use Gaussian to get the probabilities of each pixel given a class.

The equation used is

$$P(x|c) = \frac{1}{\sqrt{(2\pi)^D\ |\Sigma|}} \exp\left(\frac{-1}{2}[(x -|\mu)^T \Sigma^{-1} (x - \mu)]\right)$$

The number of dimensions being very large, the probabilities obtained are very small to overcome this we take the log likelihood.

$$\text{Log } P(x|c) = -\frac{D}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(x-\mu)^T\varepsilon^{-1}(x - \mu)$$

$\mu - \mathit{Sample\ mean}$

$\sum$ - Co-variance

Each pixel in data is assumed to have a Gaussian distribution, the code uses Scikit Learn modules Gaussian Naive Bayes classifier, each class is assigned with equal probability. Mean and standard deviation is calculated to summarize the distribution of the data, for each class.

Mean(x) = 1/n * sum(x),
where n = number of times the unique value is repeated for an input variable x.
Standard deviation (x) = √ (1/n* Σ(xi-mean(x)^2)),
it is the root of squared distance of input value x from mean value of x where n is the number of times the unique value is repeated.

To calculate the probability of new x value the Gaussian probability density function is used. The Gaussian PDF provides an estimate value of probability that x belongs to a certain class.

Procedure:

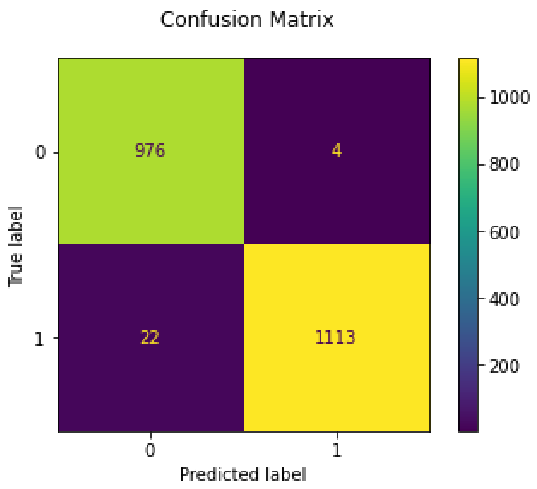
1. At first, we downloaded the dataset which comprised 4 files , 2 each for training and testing samples.Each dataset is a binary file in index type format which contains images for digits in bw 0-9 and their corresponding labels.
2. Next we extracted the input from these binary files into arrays using some datatype conversions in Python and stored them in the form of arrays.
3. Then we filtered out all images which had labels ‘0’,’1’,’3’ and ‘8’ as this is the required subset for the classification task.
4. Then we trained the Gaussian Naive Bayes classifier from scikit learn on training dataset and performed the classification on testing dataset.
5. Lastly, the accuracy was computed for both the classification problems and their respective ROC curves were plotted for GAR against FAR.

Results:

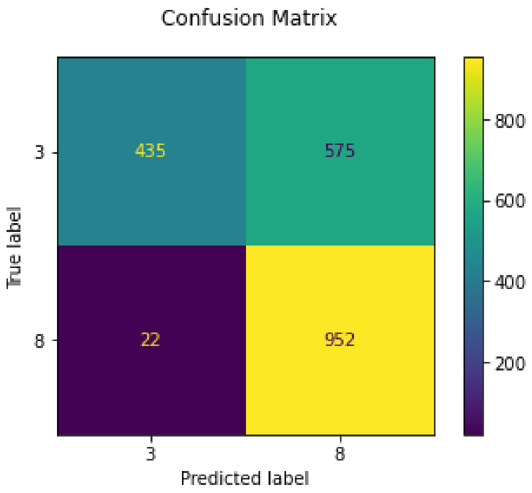
After executing the code, the following results were obtained.

1. For 0-1 digit classification problem , an accuracy score of 98.77% was observed.

2. The model was trained on 12,665 samples and tested on 2,115 images.Out of these, only 26 images were wrongly classified.
3. The ROC score was calculated as 98.83, which is reasonably good for classification tasks.
4. The accuracy for distinguishing 3-8 digits came out to be 69.90%.
5. This classifier was trained on 11,981 samples and tested on 1,984 images.The results obtained in this case were not that impressive.Only 435 images labelled as 3 were correctly classified and rest 575 was misclassified as 8.
6. ROC score came out to be 70.41 , which suggests that Bayes classifier failed to recognize minute differences between the digits.



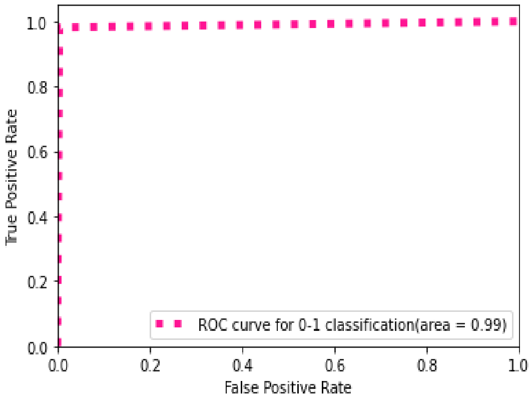
0-1 classification

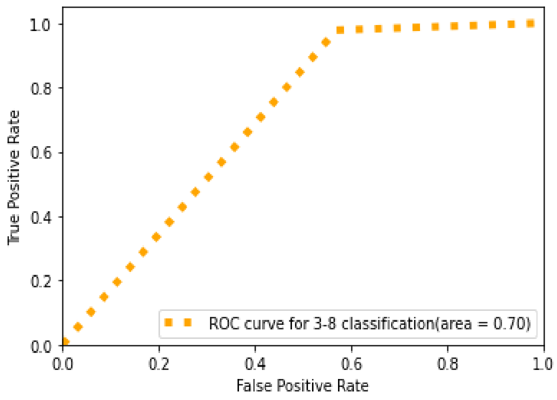


3-8 classification

Comparison

Classifier 1 performs better than classifier 2 because of more similarities in digits 3 and 8 .


True Positive Rate
False Positive Rate
ROC curve for 0-1 classification(area = 0.99)


True Positive Rate
False Positive Rate
ROC curve for 3-8 classification(area = 0.70)