ML-Assignment-2

Name: Arsh Pratap
Roll No : 2018-IMT-021
Course: Machine Learning
Course-Code : 4107
Project Name: Classification with MNIST
Git Repo : https://github.com/arshPratap/ML-Assignment-2

Problem : This project involves the implementation of an efficient and effective Naive Bayes classifier on MNIST data set. The MNIST data comprises of digital images of several digits ranging from 0 to 9. Each image is 28 x 28 pixels. Thus, the data set has 10 levels of classes.

Prerequisites : Before moving forward with the project code it is important to change the download data from the site to a csv file.

The reader can find the conversion code here : https://github.com/arshPratap/ML-Assignment-2/blob/main/MNIST-to-CSV.ipynb

I implemented a Naive Bayes classifier and presented the testing error for each digit in a table.

Below are the implementation details:

Assuming that the probability model for each pixel is Gaussian and that the probability of each class — i.e., digit—is equal. That is, P(c = 0) = P(c = 1) = : : : = P(c = 9), thus, there is no need to model the prior probabilities.

$$P(c|X) \approx \prod_{i=1}^{784} P(x_i|c)$$

In order to manage underflow error, the function below will be modeled:

$Thus,\ \mu_i|c\ and\ \sigma_i^2|c\ were\ determined\ using\ the\ training\ set.$

In order to avoid zero variance while computing the parameters above, smoothing will be applied. This involves adding a reasonable value to all the variances. The maximum variance given a class is about 13083, based on this, smoothing values ranging from 500 to 2000 with an increment of 100 will be tested through 5 fold cross validation by using the training set. The best smoothing value will then be applied to the entire training to get the model parameters. Finally, the parameters will be used for predicting the class label of the testing set. Also, we also tried to apply zero - one normalization, but it gave a poor accuracy rate. Thus, feature scaling is not applied.

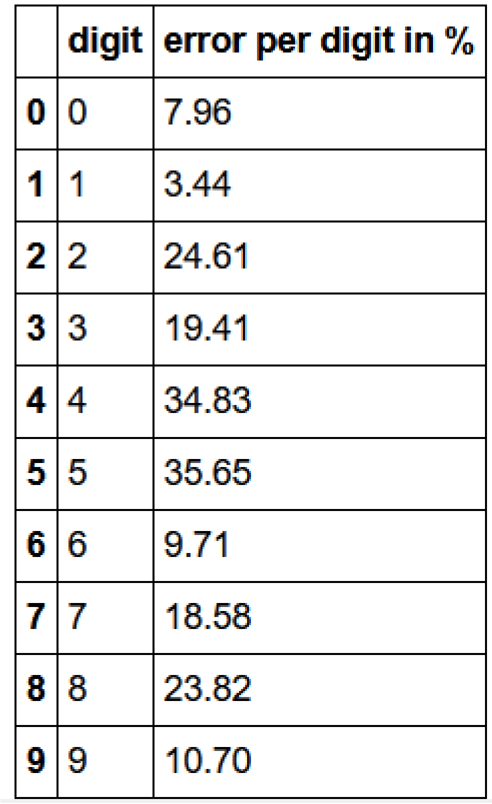
a.) Without Smoothing :
https://github.com/arshPratap/ML-Assignment-2/blob/main/NBwithSmooth.ipynb

b.) With Smoothing :
https://github.com/arshPratap/ML-Assignment-2/blob/main/NBwithSmooth.ipynb

Conclusion :
The best value of smoothing is 1000 with an error of 19.86%

smoothing = 1000 will be used to model the entire training set and prediction on the test set.

Digit-wise error



	digit	error per digit in %
0	0	7.96
1	1	3.44
2	2	24.61
3	3	19.41
4	4	34.83
5	5	35.65
6	6	9.71
7	7	18.58
8	8	23.82
9	9	10.70

Avg error : 18.51%

Implementation :

Digits 4 & 5 have the highest error rates, visualizing some of them show that they have high variations. Digit 4 is mostly misclassified as 9, some people tend to write 4 in a way that it looks like 9. Also, digit 2 is mostly misclassified as 8.

Digit 9 is mostly misclassified as either 4 or 8, likewise, digits 4 and 8 are also occasionally misclassified as 9.

Generally, Naive Bayes did poorly on the MNIST dataset, this could be attributed to the independent assumption which is likely not to be correct. Query time is faster compared to KNN, however, KNN provided better performance on the MNIST dataset. Naive Bayes doesn't perform well when there are repeated attributes or when attributes are not equally important, which is the case in the MNIST dataset.