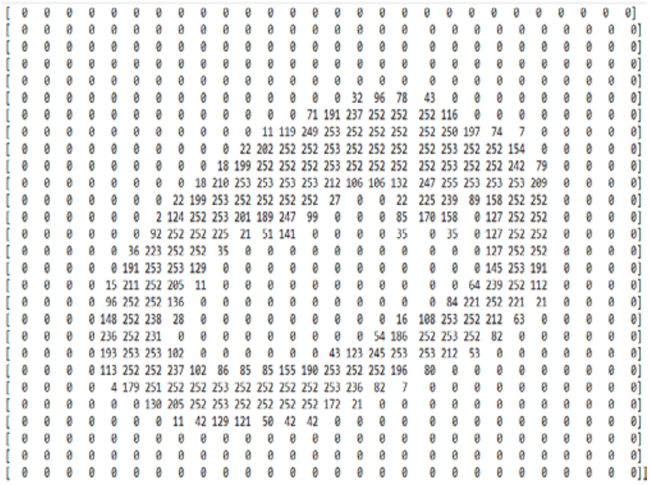
Total 4 sets of data have been downloaded below. Two sets for training, two sets for testing.

Number of samples in the training set: "0": 5000;"1": 5000
Number of samples in the testing set: "0": 980; "1": 1135

Each sample in set has 28 x 28 matrix to represent digit 0 or digit 1. Value of 0 in matrix means white, value of 255 is black. Color is grayscale and can be represented in between 0 and 255.



```
[   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  32  96  78  43   0   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   0   0   0   0   0   0   0   0  71 191 237 252 252 252 116   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   0   0   0   0   0   0  11 119 249 253 252 252 252 252 250 197  74   7   0   0   0   0   0]
[   0   0   0   0   0   0   0   0   0   0  22 202 252 252 253 252 252 252 252 253 252 252 154   0   0   0   0   0]
[   0   0   0   0   0   0   0   0   0  18 199 252 252 252 253 252 252 252 252 253 252 252 242  79   0   0   0   0]
[   0   0   0   0   0   0   0   0  18 210 253 253 253 253 212 106 106 132 247 255 253 253 253 209   0   0   0   0]
[   0   0   0   0   0   0   0  22 199 253 252 252 252 252  27   0   0  22 225 239  89 158 252 252   0   0   0   0]
[   0   0   0   0   0   0   2 124 252 253 201 189 247  99   0   0   0  85 170 158   0 127 252 252   0   0   0   0]
[   0   0   0   0   0   0  92 252 252 225  21  51 141   0   0   0   0  35   0  35   0 127 252 252   0   0   0   0]
[   0   0   0   0   0  36 223 252 252  35   0   0   0   0   0   0   0   0   0   0   0 127 252 252   0   0   0   0]
[   0   0   0   0   0 191 253 253 129   0   0   0   0   0   0   0   0   0   0   0   0 145 253 191   0   0   0   0]
[   0   0   0   0  15 211 252 205  11   0   0   0   0   0   0   0   0   0   0   0  64 239 252 112   0   0   0   0]
[   0   0   0   0  96 252 252 136   0   0   0   0   0   0   0   0   0   0   0  84 221 252 221  21   0   0   0   0]
[   0   0   0   0 148 252 238  28   0   0   0   0   0   0   0   0   0  16 108 253 252 212  63   0   0   0   0   0]
[   0   0   0   0 236 252 231   0   0   0   0   0   0   0   0   0  54 186 252 253 252  82   0   0   0   0   0   0]
[   0   0   0   0 193 253 253 102   0   0   0   0   0   0  43 123 245 253 253 212  53   0   0   0   0   0   0   0]
[   0   0   0   0 113 252 252 237 102  86  85  85 155 190 253 252 252 196  80   0   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   4 179 251 252 252 253 252 252 252 252 253 236  82   7   0   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   0 130 205 252 253 252 252 252 252 172  21   0   0   0   0   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   0   0  11  42 129 121  50  42  42   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]]
```

I need to extract two features for each image. Figure 1 is average all pixel brightness values within a whole image array. Feature 2 standard deviation of all pixel brightness values within a whole image array. Since I am going to use the Gaussian probability distribution function (GPDF) to classify digit 0 or digit 1, we need mean and standard deviation. I need to calculate 8 the parameters for the two-class (digit 0 and 1) naive bayes classifiers with GPDF.

Y=class	Figure 1(mean)	Figure 1(variance)	Figure 2(mean)	Figure 2(variance)
0	44.2838607143	114.410875544	87.5115913698	100.356180612
1	19.335694898	31.346498039	61.2965618747	82.046224968

Now, I will need calculate probability to differentiate digit 0 from digit 1. If probability of digit 0 is greater than probability of digit 1, I classify as digit 0 by using Naïve Bayse Classifier.
We assume that these two features are independent for Naïve Bayse Classifier.

Here are formulas which I used for Naïve Bayse Classifier
P(y|x1,x2) = P(y)P(x1|y) P(x2|y) / P(x1)P(x2), ignore P(x1)P(x2)
P(0|f1,f2) = P(0)P(f1|0) P(f2|0)
P(1|f1,f2) = P(1)P(f1|1) P(f2|1)
,where P(0)=P(1)=0.5
I need to calculate P(f1|0) P(f2|0) and P(f1|1) P(f2|1) by using GPDF, where x = testset sample, mean and variance for 8 parameters.

P(0|f1,f2) > P(1|f1,f2), then digit = 0
P(0|f1,f2) < P(1|f1,f2), then digit = 1

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Gaussian probability distribution function

```
# Calculate the Gaussian probability distribution function for x
def cal_gauss_prob(x, mean, var):
    return (1 / (sqrt(2 * pi * var)) * exp(-((x-mean)**2 / (2 * var))))
```

Python code for Gaussian probability distribution function

```
# predicting label for test 0 data
i=0 #predicting label digit 0 in test data set
t=0 #total number of test data set
for test in test0_features:
    #print(test[0], test[1])
    prob_0 = 0.5 * cal_gauss_prob(test[0], train0_means[0], train0_vars[0]) * cal_gauss_prob(test[1], train0_means[1], train0_vars
    prob_1 = 0.5 * cal_gauss_prob(test[0], train1_means[0], train1_vars[0]) * cal_gauss_prob(test[1], train1_means[1], train1_vars
    #print(prob_0, prob_1)
    if prob_0 > prob_1:
        i += 1
    t += 1
#print(i, t)
print('Accuracy_for_digit0testset = ', i/t)
```

Python code to calculate the accuracy of your predictions for testset Accuracy (digit 0 and 1).

After comparing probability of digit 0 with probability of digit 1, I picked up the label digit which has higher probability and count it in. Then, I am able to calculate accuracy by using Counts divided by total number as you could see at python code.

Here are all results of Project 1

Mean_of_feature1_for_digit0= 44.2838607143
Variance_of_feature1_for_digit0 = 114.410875544
Mean_of_feature2_for_digit0 = 87.5115913698
Variance_of_feature2_for_digit0 = 100.356180612
Mean_of_feature1_for_digit1 = 19.335694898
Variance_of_feature1_for_digit1 = 31.346498039
Mean_of_feature2_for_digit1 = 61.2965618747
Variance_of_feature2_for_digit1 = 82.046224968
Accuracy_for_digit0testset = 0.9173469387755102
Accuracy_for_digit1testset = 0.9233480176211454

[44.2838607143, 114.410875544, 87.5115913698, 100.356180612, 19.335694898, 31.346498039, 61.2965618747, 82.046224968, 0.9173469387755102, 0.9233480176211454]