Problem: classifying given image data set using the naïve Bayes' classifier in Python.

Solution approach:

1. In the beginning, we need to import the image and label the data set. This is utilized using the pandas library.
2. The imported data set should be divided into a training set and a test set. Training data set is used to train our model, and test data set is used to evaluate the results.
3. In this step, we need to calculate the mean parameters $\widehat{\mu_c}$:

$$\widehat{\mu_c} = \frac{\sum_{i=1}^{N} x_i 1(y_i=c)}{\sum_{i=1}^{N} 1(y_i=c)}$$

The standard deviation parameters $\widehat{\sigma_c^2}$:

$$\widehat{\sigma_c^2} = \frac{\sum_{i=1}^{N}(x_i - \widehat{\mu_c})^2 \mathbb{1}(y_i = c)}{\sum_{i=1}^{N} \mathbb{1}(y_i = c)}$$

And the prior probabilities $\hat{P}(y_i = c)$:

$$\hat{P}(y_i = c) = \frac{\sum_{i=1}^{N} \mathbb{1}(y_i = c)}{N}$$

4. With all of the above-mentioned parameters, we can now calculate the confusion matrix for the data points in our training set using the parametric classification rule. Since Bernoulli naïve Bayes' classifier (mentioned in the textbook) is derived for binary input features, it is not possible to use it for our data set. Instead, we need to use Gaussian Naive Bayes, which can be implemented for continuous data.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

5. Using the sklearn library confusion matrix is built from the list calculated in the previous section. Afterward, we can print the results as below:

```
y_truth         1         2         3         4         5
y_pred
          1  3685        49         4       679         6
          2  1430      5667      1140      1380       532
          3   508       208      4670      2948       893
          4   234        60       123       687       180
          5   143        16        63       306      4389

y_truth         1         2         3         4         5
y_pred
          1   597         6         0       114         1
          2   237       955       188       267        81
          3    92        25       785       462       167
          4    34        11        16       109        29
          5    40         3        11        48       722

Process finished with exit code 0
```