```python
"""AI&ML Lab-6:
Write a program to implement the naive Bayesian Classifier for
a sample training data set stored as .csv file.
Compute the accuracy of the classifier, considering few test data sets."""

import csv
import math
import random
import statistics


def cal_probability(x, mean, stdev):
    exponent = math.exp(-(math.pow(x - mean, 2) / (2 * math.pow(stdev, 2))))
    return (1 / (math.sqrt(2 * math.pi) * stdev)) * exponent


dataset = []
dataset_size = 0

with open('lab6.csv') as csvfile:
    lines = csv.reader(csvfile)
    for row in lines:
        dataset.append([float(attr) for attr in row])
dataset_size = len(dataset)
print("size of dataset is: ", dataset_size)
train_size = int(0.7 * dataset_size)
print(train_size)
x_train = []
x_test = dataset.copy()
training_indexes = random.sample(range(dataset_size), train_size)
for i in training_indexes:
    x_train.append(dataset[i])
    x_test.remove(dataset[i])
```

```
34
35 classes = {}
36
37 for samples in x_train:
38     last = int(samples[-1])
39     if last not in classes:
40         classes[last] = []
41     classes[last].append(samples)
42
43 print(classes)
44 summaries = {}
45 for classValue, training_data in classes.items():
46     summary = [(statistics.mean(attribute), statistics.stdev(attribute)) for attribute in zip(*
   training_data)]
47     del summary[-1]
48     summaries[classValue] = summary
49 print(summaries)
50 x_prediction = []
51 for i in x_test:
52     probabilities = {}
53     for classValue, classSummary in summaries.items():
54         probabilities[classValue] = 1
55         for index, attr in enumerate(classSummary):
56             probabilities[classValue] *= cal_probability(i[index], attr[0], attr[1])
57     best_label, best_prob = None, -1
58     for classValue, probability in probabilities.items():
59         if best_label is None or probability > best_prob:
60             best_prob = probability
61             best_label = classValue
62     x_prediction.append(best_label)
63 correct = 0
64
65 for index, key in enumerate(x_test):
```

```
    if x_test[index][-1] == x_prediction[index]:
        correct += 1
print("Accuracy:", correct / (float(len(x_test))) * 100)

"""
Output:
size of dataset is: 768
537
{1: [[0.0, 113.0, 76.0, 0.0, 0.0, 33.3, 0.278, 23.0, 1.0], [11.0, 120.0, 80.0, 37.0, 150.0, 42.3, 0.785,
  48.0, 1.0],..... 2, 0.893, 33.0, 1.0], [8.0, 167.0, 106.0, 46.0, 231.0, 37.6, 0.165, 43.0, 1.0]],
0: [[1.0, 116.0, 78.0, 29.0, 180.0, 36.1, 0.496, 25.0, 0.0], [0.0, 165.0, 76.0, 43.0, 255.0, 47.9, 0.259,
  26.0, 0.0],....., 28.6, 0.692, 21.0, 0.0],  [2.0, 90.0, 80.0, 14.0, 55.0, 24.4, 0.249, 24.0, 0.0]]}
{1: [(5.035897435897436, 3.8854398561841514), (140.96410256410257, 28.168560078510275),
 .
 .
 .....(31.172514619883042, 11.617413065169245)]}
accuracy: 73.16017316017316
"""
```