# Practical No: 5

# Naive Bayes and Gaussian Classification

**AIM: Write a program to implement the Naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.**

## Description:

**Naïve Bayesian classifier:**

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in *text classification* that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

**Gaussian Classifier :**

A Gaussian classifier, often known as a Gaussian Naive Bayes classifier, is a method of classification that uses this distribution to predict results by assuming that the features have a Gaussian (normal) distribution. This approach is frequently employed in situations requiring continuous numerical data.

## Code with output

```
import numpy as np
import pandas as pd
import sklearn
#Import dataset
from sklearn import datasets
wine = datasets.load_wine()
print("Features: ", wine.feature_names)
print("Labels: ", wine.target_names)

X=pd.DataFrame(wine['data'])
print(X.head())
print(wine.data.shape)
y=print(wine.target)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(wine.data, wine.target,
test_size=0.30,random_state=10)
#import gaussian naive bayes model.
from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()
gnb.fit(X_train,y_train)

#predict the response for test dataset
y_pred = gnb.predict(X_test)
print(y_pred)

from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

from sklearn.metrics import confusion_matrix
cm=np.array(confusion_matrix(y_test,y_pred))
cm
```

# OUTPUT

```
Features:  ['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium', 'total_phenols', 'flavanoids', 'nonflavanoid_phenols',
Labels:  ['class_0' 'class_1' 'class_2']
       0     1     2     3      4     5     6     7     8     9    10    11  \
0  14.23  1.71  2.43  15.6  127.0  2.80  3.06  0.28  2.29  5.64  1.04  3.92
1  13.20  1.78  2.14  11.2  100.0  2.65  2.76  0.26  1.28  4.38  1.05  3.40
2  13.16  2.36  2.67  18.6  101.0  2.80  3.24  0.30  2.81  5.68  1.03  3.17
3  14.37  1.95  2.50  16.8  113.0  3.85  3.49  0.24  2.18  7.80  0.86  3.45
4  13.24  2.59  2.87  21.0  118.0  2.80  2.69  0.39  1.82  4.32  1.04  2.93

       12
0  1065.0
1  1050.0
2  1185.0
3  1480.0
4   735.0
(178, 13)
[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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
```

[3]

```
GaussianNB
GaussianNB()
```

[4]

```
#predict the response for test dataset
y_pred = gnb.predict(X_test)
print(y_pred)
```

```
[1 1 0 1 0 1 1 0 2 0 0 0 0 2 1 1 2 1 1 2 0 2 0 0 1 2 1 2 1 1 2 2 2 2 1 0 0
 1 0 1 1 0 2 1 0 1 2 1 1 1 2 1 2 0]
```

```
5    cm
```

```
Accuracy: 0.8888888888888888
array([[14,  1,  0],
       [ 2, 22,  3],
       [ 0,  0, 12]])
```

### Analysis of Confusion Matrix

- Row 1 (True Class 1):

  14 instances of Class 1 were correctly predicted as Class 1 (True Positives).

  1 instance of Class 1 was incorrectly predicted as Class 2 (False Negative).

  0 instances of Class 1 were incorrectly predicted as Class 3 (False Negative).

- Row 2 (True Class 2):

  2 instances of Class 2 were incorrectly predicted as Class 1 (False Positive).

22 instances of Class 2 were correctly predicted as Class 2 (True Positives).

3 instances of Class 2 were incorrectly predicted as Class 3 (False Negative).

- Row 3 (True Class 3):

  0 instances of Class 3 were incorrectly predicted as Class 1 (False Positive).

  0 instances of Class 3 were incorrectly predicted as Class 2 (False Positive).

  12 instances of Class 3 were correctly predicted as Class 3 (True Positives).

## Learnings

1. It loads the "wine" dataset, which is a standard dataset available in scikit-learn containing information about different types of wines.
2. It splits the dataset into training and testing sets.
3. It trains a Gaussian Naive Bayes classifier on the training data.
4. The classifier is used to make predictions on the test data.
5. The code calculates and prints the accuracy of the classifier's predictions.
6. It also computes and displays the confusion matrix, which provides information about how well the classifier performed in terms of correctly classifying different wine types.