In [1]:
```
import numpy as np

import pandas as pd
```

In [3]:
```
from sklearn import datasets
wine =datasets.load_wine()
```

In [6]:
```
df=wine
```

In [8]:
```
print(df)
```

```
{'data': array([[1.423e+01, 1.710e+00, 2.430e+00, ..., 1.040e+00, 3.920e+00,
        1.065e+03],
       [1.320e+01, 1.780e+00, 2.140e+00, ..., 1.050e+00, 3.400e+00,
        1.050e+03],
       [1.316e+01, 2.360e+00, 2.670e+00, ..., 1.030e+00, 3.170e+00,
        1.185e+03],
       ...,
       [1.327e+01, 4.280e+00, 2.260e+00, ..., 5.900e-01, 1.560e+00,
        8.350e+02],
       [1.317e+01, 2.590e+00, 2.370e+00, ..., 6.000e-01, 1.620e+00,
        8.400e+02],
       [1.413e+01, 4.100e+00, 2.740e+00, ..., 6.100e-01, 1.600e+00,
        5.600e+02]]), 'target': array([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, 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, 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]), 'frame': None, 'target_names': array(['class_0', 'class_1', 'c
lass_2'], dtype='<U7'), 'DESCR': '.. _wine_dataset:\n\nWine recognition datas
et\n-----------------------\n\n**Data Set Characteristics:**\n\n    :Number
of Instances: 178 (50 in each of three classes)\n    :Number of Attributes: 1
3 numeric, predictive attributes and the class\n    :Attribute Information:\n
\t\t- Alcohol\n \t\t- Malic acid\n \t\t- Ash\n\t\t- Alcalinity of ash  \n \t
\t- Magnesium\n\t\t- Total phenols\n \t\t- Flavanoids\n \t\t- Nonflavanoid ph
enols\n \t\t- Proanthocyanins\n\t\t- Color intensity\n \t\t- Hue\n \t\t- OD28
0/OD315 of diluted wines\n \t\t- Proline\n\n    - class:\n            - class
_0\n            - class_1\n            - class_2\n\t\t\n    :Summary Statisti
cs:\n    \n    ============================== ==== ===== ======= =====\n
Min   Max   Mean     SD\n    ============================== ==== ===== =======
=====\n    Alcohol:                      11.0  14.8    13.0   0.8\n    Malic
Acid:                   0.74  5.80    2.34  1.12\n    Ash:
1.36  3.23    2.36  0.27\n    Alcalinity of Ash:            10.6  30.0    19.
5   3.3\n    Magnesium:                    70.0 162.0    99.7  14.3\n    Tota
l Phenols:                0.98  3.88    2.29  0.63\n    Flavanoids:
0.34  5.08    2.03  1.00\n    Nonflavanoid Phenols:         0.13  0.66    0.3
6  0.12\n    Proanthocyanins:              0.41  3.58    1.59  0.57\n    Colo
ur Intensity:              1.3  13.0     5.1   2.3\n    Hue:
0.48  1.71    0.96  0.23\n    OD280/OD315 of diluted wines: 1.27  4.00    2.6
1  0.71\n    Proline:                       278  1680     746   315\n    ====
========================== ==== ===== ======= =====\n\n    :Missing Attribute
Values: None\n    :Class Distribution: class_0 (59), class_1 (71), class_2 (4
8)\n    :Creator: R.A. Fisher\n    :Donor: Michael Marshall (MARSHALL%PLU@io.
arc.nasa.gov)\n    :Date: July, 1988\n\nThis is a copy of UCI ML Wine recogni
tion datasets.\nhttps://archive.ics.uci.edu/ml/machine-learning-databases/win
e/wine.data\n\nThe data is the results of a chemical analysis of wines grown
in the same\nregion in Italy by three different cultivators. There are thirte
en different\nmeasurements taken for different constituents found in the thre
e types of\nwine.\n\nOriginal Owners: \n\nForina, M. et al, PARVUS - \nAn Ext
endible Package for Data Exploration, Classification and Correlation. \nInsti
tute of Pharmaceutical and Food Analysis and Technologies,\nVia Brigata Saler
```

```
no, 16147 Genoa, Italy.\n\nCitation:\n\nLichman, M. (2013). UCI Machine Learn
ing Repository\n[https://archive.ics.uci.edu/ml]. Irvine, CA: University of C
alifornia,\nSchool of Information and Computer Science. \n\n.. topic:: Refere
nces\n\n  (1) S. Aeberhard, D. Coomans and O. de Vel, \n  Comparison of Class
ifiers in High Dimensional Settings, \n  Tech. Rep. no. 92-02, (1992), Dept.
of Computer Science and Dept. of  \n  Mathematics and Statistics, James Cook
University of North Queensland. \n  (Also submitted to Technometrics). \n\n
The data was used with many others for comparing various \n  classifiers. The
classes are separable, though only RDA \n  has achieved 100% correct classifi
cation. \n  (RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed dat
a)) \n  (All results using the leave-one-out technique) \n\n  (2) S. Aeberhar
d, D. Coomans and O. de Vel, \n  "THE CLASSIFICATION PERFORMANCE OF RDA" \n
Tech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. of \n  Math
ematics and Statistics, James Cook University of North Queensland. \n  (Also
submitted to Journal of Chemometrics).\n', 'feature_names': ['alcohol', 'mali
c_acid', 'ash', 'alcalinity_of_ash', 'magnesium', 'total_phenols', 'flavanoid
s', 'nonflavanoid_phenols', 'proanthocyanins', 'color_intensity', 'hue', 'od2
80/od315_of_diluted_wines', 'proline']}
```

In [9]:
```
print(wine.feature_names)
```

```
['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium', 'total_pheno
ls', 'flavanoids', 'nonflavanoid_phenols', 'proanthocyanins', 'color_intensit
y', 'hue', 'od280/od315_of_diluted_wines', 'proline']
```

In [10]:
```
print(wine.target_names)
```

```
['class_0' 'class_1' 'class_2']
```

In [14]:
```
x=pd.DataFrame(wine['data'])
print(x.head())
```

```
      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
```

In [16]:
```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 13 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   0       178 non-null    float64
 1   1       178 non-null    float64
 2   2       178 non-null    float64
 3   3       178 non-null    float64
 4   4       178 non-null    float64
 5   5       178 non-null    float64
 6   6       178 non-null    float64
 7   7       178 non-null    float64
 8   8       178 non-null    float64
 9   9       178 non-null    float64
 10  10      178 non-null    float64
 11  11      178 non-null    float64
 12  12      178 non-null    float64
dtypes: float64(13)
memory usage: 18.2 KB
```

In [18]:
```
x.shape
```

Out[18]:
```
(178, 13)
```

In [19]:
```
y=print(wine.target)
```

```
[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 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]
```

In [35]:
```
from sklearn.model_selection import train_test_split
```

In [57]:
```
X_train, X_test, y_train, y_test = train_test_split(wine.data, wine.target, test_
```

In [58]:
```
from sklearn.naive_bayes import GaussianNB
gnb= GaussianNB()
```

In [59]:
```
gnb.fit(X_train,y_train)
y_pred=gnb.predict(X_test)
print(y_pred)
```

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

In [ ]:

In [60]:
```
from sklearn import metrics
print(metrics.accuracy_score(y_test,y_pred))
```

```
0.9074074074074074
```

In [ ]: