```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
from sklearn.datasets import load_wine
```

```
df=load_wine()
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, 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]),
 'frame': None,
 'target_names': array(['class_0', 'class_1', 'class_2'], dtype='<U7'),
 'DESCR': '.. _wine_dataset:\n\nWine recognition dataset\n------------------------\n\n**Data Set Characteristics:**\n\n:Number
of Instances: 178\n:Number of Attributes: 13 numeric, predictive attributes and the class\n:Attribute Information:\n    -
Alcohol\n    - Malic acid\n    - Ash\n    - Alcalinity of ash\n    - Magnesium\n    - Total phenols\n    - Flavanoids\n    -
Nonflavanoid phenols\n    - Proanthocyanins\n    - Color intensity\n    - Hue\n    - OD280/OD315 of diluted wines\n    -
Proline\n    - class:\n        - class_0\n        - class_1\n        - class_2\n\n:Summary
Statistics:\n\n============================ ==== ===== ======= =====\n                              Min   Max   Mean
SD\n============================ ==== ===== ======= =====\nAlcohol:                          11.0  14.8    13.0   0.8\nMalic Acid:
0.74  5.80    2.34  1.12\nAsh:                         1.36  3.23    2.36  0.27\nAlcalinity of Ash:           10.6  30.0
19.5   3.3\nMagnesium:                    70.0 162.0    99.7  14.3\nTotal Phenols:               0.98  3.88    2.29
0.63\nFlavanoids:                  0.34  5.08    2.03  1.00\nNonflavanoid Phenols:         0.13  0.66    0.36
0.12\nProanthocyanins:             0.41  3.58    1.59  0.57\nColour Intensity:             1.3  13.0     5.1   2.3\nHue:
0.48  1.71    0.96  0.23\nOD280/OD315 of diluted wines: 1.27  4.00    2.61  0.71\nProline:                      278  1680
746   315\n============================ ==== ===== ======= =====\n\n:Missing Attribute Values: None\n:Class Distribution:
class_0 (59), class_1 (71), class_2 (48)\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 recognition datasets.\nhttps://archive.ics.uci.edu/ml/machine-learning-
databases/wine/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 thirteen different\nmeasurements taken for different constituents found in the three types
of\nwine.\n\nOriginal Owners:\n\nForina, M. et al, PARVUS -\nAn Extendible Package for Data Exploration, Classification and
Correlation.\nInstitute of Pharmaceutical and Food Analysis and Technologies,\nVia Brigata Salerno, 16147 Genoa,
Italy.\n\nCitation:\n\nLichman, M. (2013). UCI Machine Learning Repository\n[https://archive.ics.uci.edu/ml]. Irvine, CA:
University of California,\nSchool of Information and Computer Science.\n\n.. dropdown:: References\n\n    (1) S. Aeberhard, D.
Coomans and O. de Vel,\n    Comparison of Classifiers 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 classification.\n    (RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed
data))\n    (All results using the leave-one-out technique)\n\n    (2) S. Aeberhard, 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    Mathematics
and Statistics, James Cook University of North Queensland.\n    (Also submitted to Journal of Chemometrics).\n',
 'feature_names': ['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
```


```
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]])
```

```
df.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, 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])
```

```
df.feature_names
```

```
['alcohol',
 'malic_acid',
 'ash',
 'alcalinity_of_ash',
 'magnesium',
 'total_phenols',
 'flavanoids',
 'nonflavanoid_phenols',
 'proanthocyanins',
 'color_intensity',
 'hue',
 'od280/od315_of_diluted_wines',
 'proline']
```

```
new=pd.DataFrame(data=df.data,columns=df.feature_names)
```

```
new.head(4)
```

| | alcohol | malic_acid | ash | alcalinity_of_ash | magnesium | total_phenols | flavanoids | nonflavanoid_phenols | proanthocyanins | color_i |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 14.23 | 1.71 | 2.43 | 15.6 | 127.0 | 2.80 | 3.06 | 0.28 | 2.29 | |
| **1** | 13.20 | 1.78 | 2.14 | 11.2 | 100.0 | 2.65 | 2.76 | 0.26 | 1.28 | |
| **2** | 13.16 | 2.36 | 2.67 | 18.6 | 101.0 | 2.80 | 3.24 | 0.30 | 2.81 | |
| **3** | 14.37 | 1.95 | 2.50 | 16.8 | 113.0 | 3.85 | 3.49 | 0.24 | 2.18 | |

Next steps: Generate code with new | View recommended plots | New interactive sheet

```
#independent and dependent features
X=new
y=df.target
```

```
X.head(4)
```

| | alcohol | malic_acid | ash | alcalinity_of_ash | magnesium | total_phenols | flavanoids | nonflavanoid_phenols | proanthocyanins | color_i |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 14.23 | 1.71 | 2.43 | 15.6 | 127.0 | 2.80 | 3.06 | 0.28 | 2.29 | |
| **1** | 13.20 | 1.78 | 2.14 | 11.2 | 100.0 | 2.65 | 2.76 | 0.26 | 1.28 | |
| **2** | 13.16 | 2.36 | 2.67 | 18.6 | 101.0 | 2.80 | 3.24 | 0.30 | 2.81 | |
| **3** | 14.37 | 1.95 | 2.50 | 16.8 | 113.0 | 3.85 | 3.49 | 0.24 | 2.18 | |

Next steps: Generate code with X | View recommended plots | New interactive sheet

```
y
```

```
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, 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])
```

```
#standardization
from sklearn.preprocessing import StandardScaler
```

```
scaler=StandardScaler()
feature_scale=scaler.fit_transform(X)
```

```
feature_scale
```

```
array([[ 1.51861254, -0.5622498 ,  0.23205254, ...,  0.36217728,
         1.84791957,  1.01300893],
       [ 0.24628963, -0.49941338, -0.82799632, ...,  0.40605066,
         1.1134493 ,  0.96524152],
       [ 0.19687903,  0.02123125,  1.10933436, ...,  0.31830389,
         0.78858745,  1.39514818],
       ...,
       [ 0.33275817,  1.74474449, -0.38935541, ..., -1.61212515,
        -1.48544548,  0.28057537],
       [ 0.20923168,  0.22769377,  0.01273209, ..., -1.56825176,
        -1.40069891,  0.29649784],
       [ 1.39508604,  1.58316512,  1.36520822, ..., -1.52437837,
        -1.42894777, -0.59516041]])
```

```
#train_test_split
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.30,random_state=42)
```

```
X_train.head(4)
```

| | alcohol | malic_acid | ash | alcalinity_of_ash | magnesium | total_phenols | flavanoids | nonflavanoid_phenols | proanthocyanins | color |
|---|---|---|---|---|---|---|---|---|---|---|
| **138** | 13.49 | 3.59 | 2.19 | 19.5 | 88.0 | 1.62 | 0.48 | 0.58 | 0.88 | |
| **104** | 12.51 | 1.73 | 1.98 | 20.5 | 85.0 | 2.20 | 1.92 | 0.32 | 1.48 | |
| **78** | 12.33 | 0.99 | 1.95 | 14.8 | 136.0 | 1.90 | 1.85 | 0.35 | 2.76 | |
| **36** | 13.28 | 1.64 | 2.84 | 15.5 | 110.0 | 2.60 | 2.68 | 0.34 | 1.36 | |

Next steps: Generate code with X_train View recommended plots New interactive sheet

```
#naive_bayes algorithm
from sklearn.naive_bayes import GaussianNB
```

```
nb=GaussianNB()
nb.fit(X_train,y_train)
```

```
▾ GaussianNB
GaussianNB()
```

```
y_pred=nb.predict(X_test)
y_pred
```

```
array([0, 0, 2, 0, 1, 0, 1, 2, 1, 2, 0, 2, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1,
       1, 2, 2, 2, 1, 1, 1, 0, 0, 1, 2, 0, 0, 0, 2, 2, 1, 2, 0, 1, 1, 1,
       2, 0, 1, 1, 2, 0, 1, 0, 0, 2])
```

```
#evaluation
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
print(accuracy_score(y_pred,y_test))
print('\n')
print(confusion_matrix(y_pred,y_test))
print('\n')
print(classification_report(y_pred,y_test))
```

```
1.0


[[19  0  0]
 [ 0 21  0]
 [ 0  0 14]]


              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       1.00      1.00      1.00        21
           2       1.00      1.00      1.00        14

    accuracy                           1.00        54
   macro avg       1.00      1.00      1.00        54
weighted avg       1.00      1.00      1.00        54
```

```
nb.predict_proba(X_test)
```

```
array([[9.99994880e-01, 5.11985627e-06, 6.42760330e-33],
       [9.99999563e-01, 4.37161281e-07, 1.89052628e-26],
       [8.30191002e-18, 1.79036458e-03, 9.98209635e-01],
       [1.00000000e+00, 4.99340759e-10, 3.17066405e-41],
       [6.56464948e-07, 9.99999344e-01, 7.04895557e-23],
       [1.00000000e+00, 1.34177096e-12, 1.39939533e-35],
       [2.00966282e-10, 1.00000000e+00, 1.65899118e-14],
       [3.46879621e-20, 2.84799447e-12, 1.00000000e+00],
       [1.48349034e-04, 9.99851651e-01, 5.78491291e-34],
       [6.28418391e-15, 3.28566984e-04, 9.99671433e-01],
       [9.94593314e-01, 5.40668636e-03, 5.78156665e-34],
       [4.19230151e-18, 1.37204597e-12, 1.00000000e+00],
       [9.91430689e-01, 8.56931112e-03, 8.10476270e-21],
       [2.86352527e-15, 9.75916343e-01, 2.40836573e-02],
       [1.00000000e+00, 1.64055127e-13, 5.07621055e-36],
       [2.26900821e-07, 9.99999773e-01, 4.93022081e-16],
       [9.57901494e-11, 1.00000000e+00, 1.51234803e-13],
       [5.74326465e-12, 1.00000000e+00, 9.41060938e-15],
       [1.00000000e+00, 6.44749311e-12, 3.55759492e-42],
       [1.73865294e-08, 9.99999983e-01, 2.47238594e-20],
       [1.00000000e+00, 7.13027513e-23, 5.55460023e-61],
       [4.45119181e-01, 5.54880819e-01, 1.33378226e-39],
       [1.51318881e-15, 9.99989483e-01, 1.05167831e-05],
       [2.07202346e-18, 2.21944514e-13, 1.00000000e+00],
       [7.89459892e-28, 1.49000610e-18, 1.00000000e+00],
       [5.18077736e-23, 4.58986793e-19, 1.00000000e+00],
       [1.16070941e-07, 9.99999884e-01, 3.04045563e-17],
       [1.77503914e-02, 9.82249609e-01, 3.83509549e-19],
       [2.49501189e-13, 9.99999897e-01, 1.02972602e-07],
       [9.99999998e-01, 2.10958560e-09, 6.84816740e-34],
       [9.99999489e-01, 5.11271683e-07, 5.97376401e-33],
       [1.42900979e-08, 9.99999986e-01, 2.29251398e-18],
       [4.63892912e-21, 4.70879205e-06, 9.99995291e-01],
       [1.00000000e+00, 1.67615853e-15, 7.18032431e-43],
       [1.00000000e+00, 1.45901913e-10, 2.64064033e-33],
       [9.99999998e-01, 1.80228611e-09, 7.82125807e-49],
       [2.20350157e-09, 3.39394861e-22, 9.99999998e-01],
       [6.40298724e-28, 5.28157451e-07, 9.99999472e-01],
       [7.90060188e-02, 9.20993981e-01, 2.85125256e-41],
       [1.36922237e-14, 3.37732038e-20, 1.00000000e+00],
       [9.90869286e-01, 9.13071422e-03, 4.01288406e-23],
       [2.91807775e-08, 9.99999971e-01, 1.96237669e-20],
       [6.64401634e-06, 9.99993356e-01, 1.97154722e-20],
       [1.64529805e-08, 9.99567049e-01, 4.32934516e-04],
       [1.48189827e-16, 7.88099796e-05, 9.99921190e-01],
       [9.99999992e-01, 8.31682578e-09, 3.96521534e-44],
       [2.58682457e-06, 9.99997413e-01, 2.04300546e-23],
       [1.90225495e-10, 1.00000000e+00, 6.40334519e-17],
       [2.91990068e-23, 2.44691591e-10, 1.00000000e+00],
       [1.00000000e+00, 2.49715177e-10, 2.59660510e-43],
       [2.63525606e-06, 9.99997365e-01, 2.14094128e-16],
       [1.00000000e+00, 1.07688014e-10, 1.54787583e-38],
       [9.99999929e-01, 7.05451274e-08, 7.63926524e-34],
       [6.40845164e-24, 6.19690847e-21, 1.00000000e+00]])
```

Start coding or generate with AI.