```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
import numpy as np
```

```
from sklearn.datasets import load_wine
data=load_wine()
```

```
data
```


```
{'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,
        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 \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 phenols\n \t\t- Proanthocyanins\n\t\t- Color intensity\n \t\t- Hue\n \t\t- OD280/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 Statistics:\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    Total 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    0.36  0.12\n
Proanthocyanins:            0.41  3.58    1.59  0.57\n    Colour 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    2.61  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 (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.. topic:: 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',
```


```
data.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']
```

```
data.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, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 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])
```

```
data.target_names
```

```
array(['class_0', 'class_1', 'class_2'], dtype='<U7')
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(data.data,data.target,test_size=0.2,random_state=15)
# print(x_train)
# print(x_test)
# print(y_test)
# print(y_train)
```

```
from sklearn.neighbors import KNeighborsClassifier
clf=KNeighborsClassifier(n_neighbors=7,leaf_size=20,metric="euclidean")
clf.fit(x_train,y_train)
```

```
KNeighborsClassifier
KNeighborsClassifier(leaf_size=20, metric='euclidean', n_neighbors=7)
```

```
print(clf.predict(x_test))
y_pred=clf.predict(x_test)
print(y_test)
```

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

```
diff=y_pred-y_test
print(diff)
print(sum(abs(diff)))
```

```
[ 0  0  0  0  0  0  1  0 -1  0  0  0  0 -1  0  1 -1  0 -1  0  0  0 -2 -2
 -1  0  0  0  0  0  0  0  0  0  0  0]
11
```

```
print(clf.score(x_test,y_test))
```

```
0.75
```

```
from sklearn.metrics import recall_score
re=recall_score(y_test,y_pred,average="macro")
re
```

```
0.75
```

```
from sklearn.metrics import confusion_matrix,classification_report
cm=confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[12  0  0]
 [ 2  8  2]
 [ 2  3  7]]
```

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import precision_score
```

```
pr=precision_score(y_test,y_pred,average='macro')
pr
```

```
0.7516835016835017
```

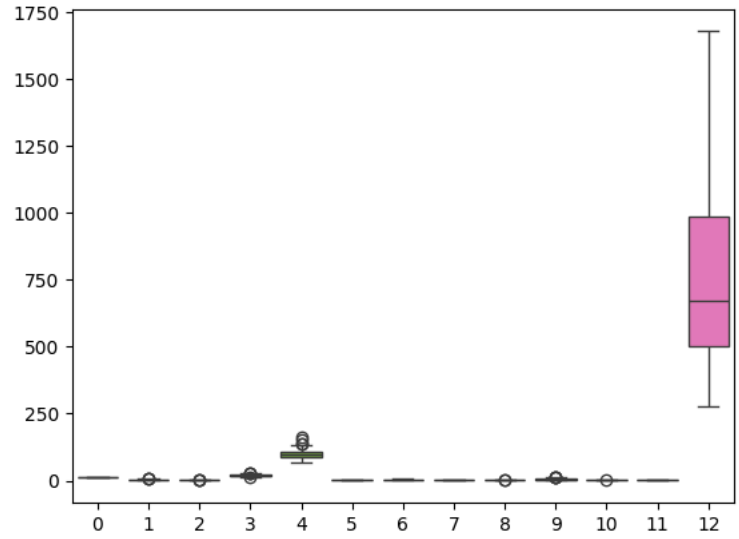
```
f1=f1_score(y_test,y_pred,average='macro')
f1
```

```
0.7398205659075224
```

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
sns.boxplot(data.data)
```

```
<Axes: >
```



Start coding or generate with AI.