

```

import pandas as pd
import numpy as np
from math import sqrt

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

iris = load_iris() #we are load the dataset here
#iris


print("IRIS FEATURES :",iris.feature_names)
print("IRIS TARGET NAMES:",iris.target_names)

    IRIS FEATURES : ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
    IRIS TARGET NAMES: ['setosa' 'versicolor' 'virginica']

X = iris.data
y = iris.target

data = np.insert(X, 4, y, axis =1)
data

```



```

array([[5.1, 3.5, 1.4, 0.2, 0. ],
       [4.9, 3. , 1.4, 0.2, 0. ],
       [4.7, 3.2, 1.3, 0.2, 0. ],
       [4.6, 3.1, 1.5, 0.2, 0. ],
       [5. , 3.6, 1.4, 0.2, 0. ],
       [5.4, 3.9, 1.7, 0.4, 0. ],
       [4.6, 3.4, 1.4, 0.3, 0. ],
       [5. , 3.4, 1.5, 0.2, 0. ],
       [4.4, 2.9, 1.4, 0.2, 0. ],
       [4.9, 3.1, 1.5, 0.1, 0. ],
       [5.4, 3.7, 1.5, 0.2, 0. ],
       [4.8, 3.4, 1.6, 0.2, 0. ],
       [4.8, 3. , 1.4, 0.1, 0. ],
       [4.3, 3. , 1.1, 0.1, 0. ],
       [5.8, 4. , 1.2, 0.2, 0. ],
       [5.7, 4.4, 1.5, 0.4, 0. ],
       [5.4, 3.9, 1.3, 0.4, 0. ],
       [5.1, 3.5, 1.4, 0.3, 0. ],
       [5.7, 3.8, 1.7, 0.3, 0. ],
       [5.1, 3.8, 1.5, 0.3, 0. ],
       [5.4, 3.4, 1.7, 0.2, 0. ],
       [5.1, 3.7, 1.5, 0.4, 0. ],
       [4.6, 3.6, 1. , 0.2, 0. ],
       [5.1, 3.3, 1.7, 0.5, 0. ],
       [4.8, 3.4, 1.9, 0.2, 0. ],
       [5. , 3. , 1.6, 0.2, 0. ],
       [5. , 3.4, 1.6, 0.4, 0. ],
       [5.2, 3.5, 1.5, 0.2, 0. ],
       [5.2, 3.4, 1.4, 0.2, 0. ],
       [4.7, 3.2, 1.6, 0.2, 0. ],
       [4.8, 3.1, 1.6, 0.2, 0. ],
       [5.4, 3.4, 1.5, 0.4, 0. ],
       [5.2, 4.1, 1.5, 0.1, 0. ],
       [5.5, 4.2, 1.4, 0.2, 0. ],
       [4.9, 3.1, 1.5, 0.2, 0. ],
       [5. , 3.2, 1.2, 0.2, 0. ],
       [5.5, 3.5, 1.3, 0.2, 0. ],
       [4.9, 3.6, 1.4, 0.1, 0. ],
       [4.4, 3. , 1.3, 0.2, 0. ],
       [5.1, 3.4, 1.5, 0.2, 0. ],
       [5. , 3.5, 1.3, 0.3, 0. ],
       [4.5, 2.3, 1.3, 0.3, 0. ],
       [4.4, 3.2, 1.3, 0.2, 0. ],
       [5. , 3.5, 1.6, 0.6, 0. ],
       [5.1, 3.8, 1.9, 0.4, 0. ],
       [4.8, 3. , 1.4, 0.3, 0. ],
       [5.1, 3.8, 1.6, 0.2, 0. ],
       [4.6, 3.2, 1.4, 0.2, 0. ],
       [5.3, 3.7, 1.5, 0.2, 0. ],
       [5. , 3.3, 1.4, 0.2, 0. ],
       [7. , 3.2, 4.7, 1.4, 1. ],
       [6.4, 3.2, 4.5, 1.5, 1. ],
       [6.9, 3.1, 4.9, 1.5, 1. ],
       [5.5, 2.3, 4. , 1.3, 1. ],
       [6.5, 2.8, 4.6, 1.5, 1. ],
       [5.7, 2.8, 4.5, 1.3, 1. ],

```

```

[6.3, 3.3, 4.7, 1.6, 1. ],
[4.9, 2.4, 3.3, 1. , 1. ],

from sklearn.model_selection import train_test_split
train, test = train_test_split(data, test_size = 0.2)

#https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3)#The default metric is minkowski
#knn = KNeighborsClassifier(n_neighbors=10,metric='euclidean')
x_train = train[:, :-1]
y_train = train[:, -1]
knn.fit(x_train, y_train)

KNeighborsClassifier(n_neighbors=3)

x_test = test[:, :-1]
y_test = test[:, -1]
y_pred = knn.predict(x_test)

from sklearn import metrics
print("KNN model accuracy:", metrics.accuracy_score(y_test, y_pred))

kNN model accuracy: 1.0

#https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)

array([[10,  0,  0],
       [ 0, 10,  0],
       [ 0,  0, 10]])

y_test

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

y_test = np.asarray(y_test, dtype = 'int')
y_pred = np.asarray(y_pred, dtype = 'int')

iris.target_names

array(['setosa', 'versicolor', 'virginica'], dtype='<U10')

# printing output of entire test_dataset
actual=[iris.target_names[p] for p in y_test]
predicted = [iris.target_names[p] for p in y_pred]

print(" ", " ", 'ACTUAL', " ", 'PREDICTED')
for i in range(len(x_test)):
    print(i, " ", actual[i], " ", predicted[i])

    ACTUAL  PREDICTED
0  virginica  virginica
1  versicolor  versicolor
2  setosa  setosa
3  versicolor  versicolor
4  versicolor  versicolor
5  virginica  virginica
6  setosa  setosa
7  virginica  virginica
8  virginica  virginica
9  setosa  setosa
10  setosa  setosa
11  setosa  setosa
12  virginica  virginica
13  setosa  setosa
14  virginica  virginica
15  setosa  setosa
16  virginica  virginica
17  setosa  setosa
18  virginica  virginica
19  versicolor  versicolor

```

```
20 virginica virginica
21 setosa setosa
22 setosa setosa
23 versicolor versicolor
24 versicolor versicolor
25 versicolor versicolor
26 versicolor versicolor
27 versicolor versicolor
28 versicolor versicolor
29 virginica virginica
```

[Colab paid products](#) - [Cancel contracts here](#)

