In [10]:
```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

In [17]:
```
df=pd.read_csv("f:/dataset/classification/fruits.csv")
df
```

Out[17]:

	diameter	weight	FruitName
0	3.0	30	Banana
1	6.0	100	Apple
2	6.1	95	Apple
3	3.2	35	Banana
4	5.5	80	Apple
5	7.1	120	Banana
6	2.5	60	Banana
7	2.3	100	Banana
8	4.8	70	Apple
9	4.8	79	Apple
10	5.8	120	Apple
11	2.6	85	Banana
12	6.0	110	Apple
13	6.3	95	Apple
14	3.0	40	Banana
15	3.5	25	Banana
16	5.5	100	Apple
17	7.5	120	Apple
18	2.5	50	Banana
19	2.7	40	Banana
20	4.8	90	Apple
21	5.8	90	Apple

In [15]:
```
model=KNeighborsClassifier()
X=df.iloc[:,:-1].values
y=df.iloc[:,-1].values
model.fit(X,y)
pred=model.predict(X)
accuracy_score(y,pred)
```

Out[15]:
```
0.8636363636363636
```

In [18]:
```
from sklearn.model_selection import train_test_split
```

In [19]:
```
train_test_split([1,2,3,4,5]) #.75 &.25
```

```
[[5, 2, 4], [3, 1]]
```

Out[19]:

In [20]:
```
train_test_split([1,2,3,4,5]) #.75 &.25
```

Out[20]:
```
[[1, 4, 5], [2, 3]]
```

In [21]:
```
train_test_split([1,2,3,4,5,6,7,8,9,10])
```

Out[21]:
```
[[1, 10, 9, 5, 8, 4, 3], [7, 2, 6]]
```

In [22]:
```
train_test_split([1,2,3,4,5,6,7,8,9,10],train_size=.6)
```

Out[22]:
```
[[4, 5, 7, 3, 8, 6], [2, 10, 1, 9]]
```

In [23]:
```
train_test_split([1,2,3,4,5,6,7,8,9,10],train_size=8)
```

Out[23]:
```
[[9, 5, 4, 8, 2, 3, 1, 10], [6, 7]]
```

In [24]:
```
train_test_split([1,2,3,4,5,6,7,8,9,10],test_size=.3)
```

Out[24]:
```
[[4, 2, 7, 8, 3, 6, 5], [9, 1, 10]]
```

In [25]:
```
train_test_split([1,2,3,4,5,6,7,8,9,10],train_size=.6,test_size=.4)
```

Out[25]:
```
[[6, 2, 10, 8, 1, 7], [4, 9, 3, 5]]
```

In [26]:
```
train_test_split([1,2,3,4,5,6,7,8,9,10],train_size=.6,test_size=.5)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[26], line 1
----> 1 train_test_split([1,2,3,4,5,6,7,8,9,10],train_size=.6,test_size=.5)

File ~\anaconda3\Lib\site-packages\sklearn\model_selection\_split.py:2562, in train_test
_split(test_size, train_size, random_state, shuffle, stratify, *arrays)
   2559 arrays = indexable(*arrays)
   2561 n_samples = _num_samples(arrays[0])
-> 2562 n_train, n_test = _validate_shuffle_split(
   2563     n_samples, test_size, train_size, default_test_size=0.25
   2564 )
   2566 if shuffle is False:
   2567     if stratify is not None:

File ~\anaconda3\Lib\site-packages\sklearn\model_selection\_split.py:2205, in _validate_
shuffle_split(n_samples, test_size, train_size, default_test_size)
   2202     raise ValueError("Invalid value for test_size: {}".format(test_size))
   2204 if train_size_type == "f" and test_size_type == "f" and train_size + test_size >
 1:
-> 2205     raise ValueError(
   2206         "The sum of test_size and train_size = {}, should be in the (0, 1)"
   2207         " range. Reduce test_size and/or train_size.".format(train_size + test_s
ize)
   2208     )
   2210 if test_size_type == "f":
   2211     n_test = ceil(test_size * n_samples)

ValueError: The sum of test_size and train_size = 1.1, should be in the (0, 1) range. Re
duce test_size and/or train_size.
```

In [27]:
```
train_test_split([1,2,3,4,5,6,7,8,9,10],random_state=1)
```

```
[[5, 1, 4, 2, 8, 9, 6], [3, 10, 7]]
```

Out[27]:

In [28]:
```
train_test_split([1,2,3,4,5,6,7,8,9,10],random_state=1)
```

Out[28]:
```
[[5, 1, 4, 2, 8, 9, 6], [3, 10, 7]]
```

In [29]:
```
train_test_split([1,2,3,4,5],[6,7,8,9,10],random_state=1)
```

Out[29]:
```
[[5, 1, 4], [3, 2], [10, 6, 9], [8, 7]]
```

In [30]:
```
model=KNeighborsClassifier()
X=df.iloc[:,:-1].values
y=df.iloc[:,-1].values

X_train,X_test,y_train,y_test=train_test_split(X,y,random_state=1)
```

In [31]:
```
X_train
```

Out[31]:
```
array([[  5.8, 120. ],
       [  5.5,  80. ],
       [  6.1,  95. ],
       [  7.5, 120. ],
       [  2.5,  60. ],
       [  2.3, 100. ],
       [  6. , 100. ],
       [  3. ,  40. ],
       [  3. ,  30. ],
       [  5.8,  90. ],
       [  4.8,  90. ],
       [  4.8,  79. ],
       [  4.8,  70. ],
       [  6. , 110. ],
       [  2.6,  85. ],
       [  7.1, 120. ]])
```

In [32]:
```
y_train
```

Out[32]:
```
array(['Apple', 'Apple', 'Apple', 'Apple', 'Banana', 'Banana', 'Apple',
       'Banana', 'Banana', 'Apple', 'Apple', 'Apple', 'Apple', 'Apple',
       'Banana', 'Banana'], dtype=object)
```

In [33]:
```
X_test
```

Out[33]:
```
array([[  2.7,  40. ],
       [  5.5, 100. ],
       [  3.2,  35. ],
       [  6.3,  95. ],
       [  2.5,  50. ],
       [  3.5,  25. ]])
```

In [34]:
```
y_test
```

Out[34]:
```
array(['Banana', 'Apple', 'Banana', 'Apple', 'Banana', 'Banana'],
      dtype=object)
```

In [35]:
```
model=KNeighborsClassifier()
model.fit(X_train,y_train)
```

Out[35]:
▾ KNeighborsClassifier

KNeighborsClassifier()

In [36]:
```
pred_train=model.predict(X_train)
```

```
pred_test=model.predict(X_test)
```

In [37]:
```
accuracy_score(y_train,pred_train)
```

Out[37]:
```
0.75
```

In [38]:
```
accuracy_score(y_test,pred_test)
```

Out[38]:
```
1.0
```

In [ ]:
```

```