In [3]:

```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
```

In [4]:

```
df=pd.read_csv("f:/dataset/classification/fruits.csv")
X=df.iloc[:,:-1].values
y=df.iloc[:,-1].values
model=KNeighborsClassifier()
model.fit(X,y)
```

Out[4]:

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

In [20]:

```
model.predict([[3.5,80]])
```

Out[20]:

```
array(['Apple'], dtype=object)
```

In [23]:

```
model.predict([[5.5,20]])
```

Out[23]:

```
array(['Banana'], dtype=object)
```

Feature Scaling

- making all fearures in same range
- sothat all features will contibute equally in prediction

Note:if we do not perform scaling then a feature with big values will impact the prediction.

Techniques:

- MinMaxScaler
- MaxAbsScaler
- StandardScalar
- Binarizer

etc.

In [24]:

```
import numpy as np
X=np.array([[2,20],[1,18],[2,15],[4,50],[3,30],[4,60]])
print(X)
```

```
[[ 2 20]
 [ 1 18]
 [ 2 15]
 [ 4 50]
 [ 3 30]
 [ 4 60]]
```

In [25]:

```
from sklearn.model_selection import train_test_split
```

In [29]:

```
X_train,X_test=train_test_split(X,random_state=1)
```

In [30]:

```
X_train
```

Out[30]:

```
array([[ 3, 30],
       [ 2, 20],
       [ 4, 50],
       [ 4, 60]])
```

In [31]:

```
X_test
```

Out[31]:

```
array([[ 2, 15],
       [ 1, 18]])
```

In [33]:

```
from sklearn.preprocessing import MinMaxScaler
```

In [34]:

```
sc=MinMaxScaler(feature_range=(0,1))
sc.fit(X_train)
X_train_new=sc.transform(X_train)
```

In [35]:

```
X_train_new
```

Out[35]:

```
array([[0.5 , 0.25],
       [0.  , 0.  ],
       [1.  , 0.75],
       [1.  , 1.  ]])
```

In [36]:

```
X_test_new=sc.transform(X_test)
```

In [37]:

```
X_test_new
```

Out[37]:

```
array([[ 0.   , -0.125],
       [-0.5  , -0.05 ]])
```

In [38]:

```
X_train
```

Out[38]:

```
array([[ 3, 30],
       [ 2, 20],
       [ 4, 50],
       [ 4, 60]])
```

In [39]:

```
sc=MinMaxScaler(feature_range=(0,1))
sc.fit(X_train) #find (learn) values of xmin,xmax for all features
sc.transform(X_train)#apply actual formula
```

Out[39]:

```
array([[0.5 , 0.25],
       [0.  , 0.  ],
       [1.  , 0.75],
       [1.  , 1.  ]])
```

In [40]:

```
sc.transform(X_test)
```

Out[40]:

```
array([[ 0.   , -0.125],
       [-0.5  , -0.05 ]])
```

In [41]:

```
sc=MinMaxScaler(feature_range=(0,1))
sc.fit_transform(X_train) #find (learn) values of xmin,xmax as well as apply formula
```

Out[41]:

```
array([[0.5 , 0.25],
       [0.  , 0.  ],
       [1.  , 0.75],
       [1.  , 1.  ]])
```

In [42]:

```
sc.transform(X_test)
```

Out[42]:

```
array([[ 0.   , -0.125],
       [-0.5  , -0.05 ]])
```

In [ ]: