1.Read the data
2.Check the data or preprocess the data
3.Seperate input labels and target or output labels
4.Seperate the train and test data(opt)
5.Apply the algorithm
6.Train the model by giving input and output values
7.test the data
8.Evalute the model

In [1]:

```
#1.read the data
import pandas as ps
data = ps.read_csv("shirtsize.csv")
print(data)
```

...

In [3]:

```
#2.Check the data
data.isnull().sum()
```

Out[3]:

```
Height    0
Weight    0
Size      0
dtype: int64
```

In [4]:

```
data.columns
```

Out[4]:

```
Index(['Height', 'Weight', 'Size'], dtype='object')
```

In [5]:

```
#3.Seperate the input and output
input_data = data[['Height','Weight']]
input_data
```

...

In [6]:

```
output_data = data['Size']
output_data
```

...

In target columns must the values in int or float format only

In [8]:

```
tran_output = ps.get_dummies(data['Size'],drop_first=True)
tran_output
```

...

In [9]:

```
#5.
data.shape
```

Out[9]:

```
(18, 3)
```

In [10]:

```
#5.Split the train and test optional
#6.Apply the Algorithm
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=)
```

In [11]:

```
#7.train the model by using fit method
knn.fit(input_data,tran_output)
```

...

In [12]:

```
# test the model or algorithm by using predict method
pred_values = knn.predict(input_data)
```

In [13]:

```
from sklearn.metrics import accuracy_score
accuracy_score(tran_output,pred_values)
```

Out[13]:

```
0.8333333333333334
```

In [14]:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(tran_output,pred_values)
```

Out[14]:

```
array([[10,  1],
       [ 2,  5]], dtype=int64)
```

In [15]:

```
knn.predict([[158,120]])
```

Out[15]:

```
array([0], dtype=uint8)
```

In [ ]:

```
1
```