In [76]:

```
#Loading the dataset
import pandas as pd
import numpy as np
dataset=pd.read_csv('Data1.csv')
dataset
```

Out[76]:

	Country	Age	Salary	Purchased
0	France	44.0	72000.0	No
1	Spain	27.0	48000.0	Yes
2	Germany	30.0	54000.0	No
3	Spain	38.0	61000.0	No
4	Germany	40.0	NaN	Yes
5	France	35.0	58000.0	Yes
6	Spain	NaN	52000.0	No
7	France	48.0	79000.0	Yes
8	Germany	50.0	83000.0	No
9	France	37.0	67000.0	Yes

In [77]:

```
#Identifying the missing values
from sklearn.impute import SimpleImputer
imputer=SimpleImputer(missing_values=np.NaN,strategy='mean')
imputer=imputer.fit(dataset[['Age']])
dataset['Age']=imputer.transform(dataset[['Age']])
imputer=imputer.fit(dataset[['Salary']])
dataset['Salary']=imputer.transform(dataset[['Salary']])
dataset
```

Out[77]:

	Country	Age	Salary	Purchased
0	France	44.000000	72000.000000	No
1	Spain	27.000000	48000.000000	Yes
2	Germany	30.000000	54000.000000	No
3	Spain	38.000000	61000.000000	No
4	Germany	40.000000	63777.777778	Yes
5	France	35.000000	58000.000000	Yes
6	Spain	38.777778	52000.000000	No
7	France	48.000000	79000.000000	Yes
8	Germany	50.000000	83000.000000	No
9	France	37.000000	67000.000000	Yes

In [78]:

```
#filling missing values
dataset.fillna({'Age':'young'})
```

Out[78]:

	Country	Age	Salary	Purchased
0	France	44.000000	72000.000000	No
1	Spain	27.000000	48000.000000	Yes
2	Germany	30.000000	54000.000000	No
3	Spain	38.000000	61000.000000	No
4	Germany	40.000000	63777.777778	Yes
5	France	35.000000	58000.000000	Yes
6	Spain	38.777778	52000.000000	No
7	France	48.000000	79000.000000	Yes
8	Germany	50.000000	83000.000000	No
9	France	37.000000	67000.000000	Yes

In [79]:

```
dataset.fillna({'Age':'25.0','Salary':'50000'})
```

Out[79]:

	Country	Age	Salary	Purchased
0	France	44.000000	72000.000000	No
1	Spain	27.000000	48000.000000	Yes
2	Germany	30.000000	54000.000000	No
3	Spain	38.000000	61000.000000	No
4	Germany	40.000000	63777.777778	Yes
5	France	35.000000	58000.000000	Yes
6	Spain	38.777778	52000.000000	No
7	France	48.000000	79000.000000	Yes
8	Germany	50.000000	83000.000000	No
9	France	37.000000	67000.000000	Yes

In [80]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Country    10 non-null     object 
 1   Age        10 non-null     float64
 2   Salary     10 non-null     float64
 3   Purchased  10 non-null     object 
dtypes: float64(2), object(2)
memory usage: 448.0+ bytes
```

In [81]:

```
#using LabelEncoder
from sklearn.preprocessing import LabelEncoder
x=dataset.iloc[:,:-1].values
label=LabelEncoder()
x[:,0]=label.fit_transform(x[:,0])
print(x)
```

```
[[0 44.0 72000.0]
 [2 27.0 48000.0]
 [1 30.0 54000.0]
 [2 38.0 61000.0]
 [1 40.0 63777.77777777778]
 [0 35.0 58000.0]
 [2 38.77777777777778 52000.0]
 [0 48.0 79000.0]
 [1 50.0 83000.0]
 [0 37.0 67000.0]]
```

In [82]:

```
#using OneHotEncoder
from sklearn.preprocessing import OneHotEncoder
dummy=pd.get_dummies(dataset['Country'])
dummy
```

Out[82]:

	France	Germany	Spain
0	1	0	0
1	0	0	1
2	0	1	0
3	0	0	1
4	0	1	0
5	1	0	0
6	0	0	1
7	1	0	0
8	0	1	0
9	1	0	0

In [83]:

```
from sklearn.preprocessing import OneHotEncoder
dummy=pd.get_dummies(dataset['Purchased'])
dummy
```

Out[83]:

	No	Yes
0	1	0
1	0	1
2	1	0
3	1	0
4	0	1
5	0	1
6	1	0
7	0	1
8	1	0
9	0	1

In [84]:

```
from sklearn.preprocessing import OneHotEncoder
onehot=OneHotEncoder()
onehot.fit_transform(dataset.Country.values.reshape(-1,1)).toarray()
```

Out[84]:

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

In [85]:

```
dataset
```

Out[85]:

	Country	Age	Salary	Purchased
0	France	44.000000	72000.000000	No
1	Spain	27.000000	48000.000000	Yes
2	Germany	30.000000	54000.000000	No
3	Spain	38.000000	61000.000000	No
4	Germany	40.000000	63777.777778	Yes
5	France	35.000000	58000.000000	Yes
6	Spain	38.777778	52000.000000	No
7	France	48.000000	79000.000000	Yes
8	Germany	50.000000	83000.000000	No
9	France	37.000000	67000.000000	Yes

In [86]:

```
#Training and Testing
from sklearn.model_selection import train_test_split
x_train,x_text=train_test_split(x,test_size=0.2,random_state=0)
x_train
```

Out[86]:

```
array([[1, 40.0, 63777.77777777778],
       [0, 37.0, 67000.0],
       [2, 27.0, 48000.0],
       [2, 38.77777777777778, 52000.0],
       [0, 48.0, 79000.0],
       [2, 38.0, 61000.0],
       [0, 44.0, 72000.0],
       [0, 35.0, 58000.0]], dtype=object)
```

In [87]:

```
from sklearn.model_selection import train_test_split
y=dataset.iloc[:,-1:1].values
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=
x_train
```

Out[87]:

```
array([[0, 37.0, 67000.0],
       [2, 27.0, 48000.0],
       [2, 38.77777777777778, 52000.0],
       [0, 48.0, 79000.0],
       [2, 38.0, 61000.0],
       [0, 44.0, 72000.0],
       [0, 35.0, 58000.0]], dtype=object)
```

In [94]:

```
#Using standardscaler
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
print(x_train)
print("test")
print(x_test)
```

```
[[-0.8660254  -0.2029809   0.44897083]
 [ 1.15470054 -1.82168936 -1.41706417]
 [ 1.15470054  0.08478949 -1.0242147 ]
 [-0.8660254   1.5775984   1.62751925]
 [ 1.15470054 -0.04111006 -0.14030338]
 [-0.8660254   0.93011502  0.94003267]
 [-0.8660254  -0.52672259 -0.43494049]]
test
[[ 0.         -1.22474487 -1.07298811]
 [ 0.          1.22474487  1.33431759]
 [ 0.          0.         -0.26132948]]
```