In [1]:

```
#importing the Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

```
#importing the dataset
dataset  = pd.read_csv("C://Users//pritespa//Data_Preprocessing//Data.csv")
print(dataset)
X = dataset.iloc[:,:-1].values
print(X)
y = dataset.iloc[:,3].values
print(y)
```

```
   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
[['France' 44.0 72000.0]
 ['Spain' 27.0 48000.0]
 ['Germany' 30.0 54000.0]
 ['Spain' 38.0 61000.0]
 ['Germany' 40.0 nan]
 ['France' 35.0 58000.0]
 ['Spain' nan 52000.0]
 ['France' 48.0 79000.0]
 ['Germany' 50.0 83000.0]
 ['France' 37.0 67000.0]]
['No' 'Yes' 'No' 'No' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes']
```

In [3]:

```
#Taking care of Missing Data
from sklearn.preprocessing import Imputer
imputer = Imputer(missing_values= 'NaN', strategy='mean', axis=0)
imputer = imputer.fit(X[:, 1:3])
X[:, 1:3] = imputer.transform(X[:, 1:3])
print(X)
```

```
[['France' 44.0 72000.0]
 ['Spain' 27.0 48000.0]
 ['Germany' 30.0 54000.0]
 ['Spain' 38.0 61000.0]
 ['Germany' 40.0 63777.77777777778]
 ['France' 35.0 58000.0]
 ['Spain' 38.77777777777778 52000.0]
 ['France' 48.0 79000.0]
 ['Germany' 50.0 83000.0]
 ['France' 37.0 67000.0]]
```

In [4]:

```
#Encoding categorial data
from sklearn.preprocessing import LabelEncoder , OneHotEncoder
```

In [5]:

```
#1st Country column of the matrix is encoded
labelencoder_X = LabelEncoder()
X[:,0] = labelencoder_X.fit_transform(X[:, 0])
print(X)
onehotencoder = OneHotEncoder(categorical_features = [0])
X = onehotencoder.fit_transform(X).toarray()
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]]
[[  1.00000000e+00   0.00000000e+00   0.00000000e+00   4.40000000e+01
    7.20000000e+04]
 [  0.00000000e+00   0.00000000e+00   1.00000000e+00   2.70000000e+01
    4.80000000e+04]
 [  0.00000000e+00   1.00000000e+00   0.00000000e+00   3.00000000e+01
    5.40000000e+04]
 [  0.00000000e+00   0.00000000e+00   1.00000000e+00   3.80000000e+01
    6.10000000e+04]
 [  0.00000000e+00   1.00000000e+00   0.00000000e+00   4.00000000e+01
    6.37777778e+04]
 [  1.00000000e+00   0.00000000e+00   0.00000000e+00   3.50000000e+01
    5.80000000e+04]
 [  0.00000000e+00   0.00000000e+00   1.00000000e+00   3.87777778e+01
    5.20000000e+04]
 [  1.00000000e+00   0.00000000e+00   0.00000000e+00   4.80000000e+01
    7.90000000e+04]
 [  0.00000000e+00   1.00000000e+00   0.00000000e+00   5.00000000e+01
    8.30000000e+04]
 [  1.00000000e+00   0.00000000e+00   0.00000000e+00   3.70000000e+01
    6.70000000e+04]]
```

In [6]:

```
#encode Purchased column of the matrix
labelencoder_y = LabelEncoder()
y = labelencoder_y.fit_transform(y)
print(y)
```

```
[0 1 0 0 1 1 0 1 0 1]
```

In [7]:
```
#splitting the dataset into Training and Test set
from sklearn.cross_validation import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size=0.2,random_s
tate=0)
print(X_train, X_test, Y_train, Y_test)
```

```
[[  0.00000000e+00   1.00000000e+00   0.00000000e+00   4.00000000e+01
    6.37777778e+04]
 [  1.00000000e+00   0.00000000e+00   0.00000000e+00   3.70000000e+01
    6.70000000e+04]
 [  0.00000000e+00   0.00000000e+00   1.00000000e+00   2.70000000e+01
    4.80000000e+04]
 [  0.00000000e+00   0.00000000e+00   1.00000000e+00   3.87777778e+01
    5.20000000e+04]
 [  1.00000000e+00   0.00000000e+00   0.00000000e+00   4.80000000e+01
    7.90000000e+04]
 [  0.00000000e+00   0.00000000e+00   1.00000000e+00   3.80000000e+01
    6.10000000e+04]
 [  1.00000000e+00   0.00000000e+00   0.00000000e+00   4.40000000e+01
    7.20000000e+04]
 [  1.00000000e+00   0.00000000e+00   0.00000000e+00   3.50000000e+01
    5.80000000e+04]] [[  0.00000000e+00   1.00000000e+00   0.00000000e+00
3.00000000e+01
    5.40000000e+04]
 [  0.00000000e+00   1.00000000e+00   0.00000000e+00   5.00000000e+01
    8.30000000e+04]] [1 1 1 0 1 0 0 1] [0 0]

D:\Anaconda\lib\site-packages\sklearn\cross_validation.py:41: DeprecationWarn
ing: This module was deprecated in version 0.18 in favor of the model_selecti
on module into which all the refactored classes and functions are moved. Also
note that the interface of the new CV iterators are different from that of th
is module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
```

In [11]:
```
#feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
X_train
```

Out[11]:
```
array([[-1.        ,  2.64575131, -0.77459667,  0.26306757,  0.12381479],
       [ 1.        , -0.37796447, -0.77459667, -0.25350148,  0.46175632],
       [-1.        , -0.37796447,  1.29099445, -1.97539832, -1.53093341],
       [-1.        , -0.37796447,  1.29099445,  0.05261351, -1.11141978],
       [ 1.        , -0.37796447, -0.77459667,  1.64058505,  1.7202972 ],
       [-1.        , -0.37796447,  1.29099445, -0.0813118 , -0.16751412],
       [ 1.        , -0.37796447, -0.77459667,  0.95182631,  0.98614835],
       [ 1.        , -0.37796447, -0.77459667, -0.59788085, -0.48214934]])
```

In [12]:
```
X_test
```

Out[12]:
```
array([[-1.        ,  2.64575131, -0.77459667, -1.45882927, -0.90166297],
       [-1.        ,  2.64575131, -0.77459667,  1.98496442,  2.13981082]])
```