In [2]:
```
##  Data Preprocessing using Pandas
```

In [8]:
```
import pandas as pd
import numpy as np
```

In [9]:
```
df = pd.read_csv('C:/Users/SW20407278/Desktop/Final AI/Hands-On/Data Preprocessing/pre
```

In [10]:
```
df
```

Out[10]:

	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	NaN	50.0	83000.0	No
9	France	37.0	67000.0	Yes

In [11]:
```
### Data Preprocessing includes
### Handling of Missing data
### Handling of Categorical data
### Feature Scaling
```

In [12]:
```
df.info()
```

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

In [13]:
```
###  Handling Missing values with Statistics

# If the column is numerical continuous, replace the NaN value by the mean of that col
# If the column is numerical discrete, replace the NaN value by the median of that col
# If the column is non-numerical column, replace the NaN value by the mode of that col
```

In [14]:
```
## Country is non-numerical column, replace NaN value with the mode of that column.
df.Country.mode()[0]
```

Out[14]:
```
'France'
```

In [15]:
```
df.Country.fillna( df.Country.mode()[0] , inplace=True )
```

In [14]:
```
### Age is continuous variable but here considered as discrete so replacing NaN value
df.Age.fillna( df.Age.median() , inplace=True )
```

In [15]:
```
### Salary is continuous variable, replace NaN with mean of that column.
df.Salary.fillna( round( df.Salary.mean() ) , inplace=True )
```

In [ ]:
```
### Handling of categorical data (Creation of dummy variables)
```

In [16]:
```
pd.get_dummies(df.Country)
```

Out[16]:

	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	1	0	0
9	1	0	0

In [18]:
```
updated_dataset = pd.concat([ pd.get_dummies(df.Country),df.iloc[:,[1,2,3]]],axis=1)
```

In [19]:
```
updated_dataset
```

Out[19]:

	France	Germany	Spain	Age	Salary	Purchased
0	1	0	0	44.0	72000.0	No
1	0	0	1	27.0	48000.0	Yes
2	0	1	0	30.0	54000.0	No
3	0	0	1	38.0	61000.0	No
4	0	1	0	40.0	63778.0	Yes
5	1	0	0	35.0	58000.0	Yes
6	0	0	1	38.0	52000.0	No
7	1	0	0	48.0	79000.0	Yes
8	1	0	0	50.0	83000.0	No
9	1	0	0	37.0	67000.0	Yes

In [20]:
```
updated_dataset.Purchased.replace(['No','Yes'],[0,1] , inplace=True)
```

In [21]:
```
updated_dataset
```

Out[21]:

	France	Germany	Spain	Age	Salary	Purchased
0	1	0	0	44.0	72000.0	0
1	0	0	1	27.0	48000.0	1
2	0	1	0	30.0	54000.0	0
3	0	0	1	38.0	61000.0	0
4	0	1	0	40.0	63778.0	1
5	1	0	0	35.0	58000.0	1
6	0	0	1	38.0	52000.0	0
7	1	0	0	48.0	79000.0	1
8	1	0	0	50.0	83000.0	0
9	1	0	0	37.0	67000.0	1

In [ ]:
```

```