DATA PRE-PROCESSING IN ML

DATA PRE-PROCESSING

- Data pre-processing is one of the most important steps when we create any ML model.
- There are different data pre-processing tools and we use them depending upon the data set we have.
- Our first implementation will be with all the tools.

	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

IMPORTING LIBRARIES

- These are the generic libraries which we are going to use in all the ML models.

Importing the libraries

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

IMPORTING DATA SET

- The next step in data pre-processing is to import you data set.
- All the data set contains features and the dependent variable thus it is divided into the matrix of independent variable(features) and a vector of dependent variable.

```
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 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]]
```

```
print(y)
```

```
['No' 'Yes' 'No' 'No' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes']
```

TAKING CARE OF MISSING DATA

- The missing data in your data set can produce unwanted results therefore it is very important to take care of the missing data.
- There are two ways to do it:
  1. to ignore the missing data
  2. To replace the missing the data

```
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]]
```

ENCODING CATEGORIAL DATA(1/2)

- Most of the time in our data set we have categorial data which is to encoded and converted into numerical values.
- This will prevent the ML model from creating unnecessary co-relation between the features and the dependent variable.

	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

ENCODING CATEGORIAL DATA(2/2)

```
print(X)
```

```
[[1.0 0.0 0.0 44.0 72000.0]
 [0.0 0.0 1.0 27.0 48000.0]
 [0.0 1.0 0.0 30.0 54000.0]
 [0.0 0.0 1.0 38.0 61000.0]
 [0.0 1.0 0.0 40.0 63777.77777777778]
 [1.0 0.0 0.0 35.0 58000.0]
 [0.0 0.0 1.0 38.77777777777778 52000.0]
 [1.0 0.0 0.0 48.0 79000.0]
 [0.0 1.0 0.0 50.0 83000.0]
 [1.0 0.0 0.0 37.0 67000.0]]
```

```
print(y)
```

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

SPLITTING THE DATA SET INTO TRAINING SET AND TEST SET

- The data set from the previous steps is further divided into training set and test set.
- The training set is the set on which you will train your ML model.
- The test set is assumed as the future set on which you are going to test your model.

```
print(X_train)
```

```
[[0.0 1.0 0.0 40.0 63777.77777777778]
 [1.0 0.0 0.0 37.0 67000.0]
 [0.0 0.0 1.0 27.0 48000.0]
 [0.0 0.0 1.0 38.77777777777778 52000.0]
 [1.0 0.0 0.0 48.0 79000.0]
 [0.0 0.0 1.0 38.0 61000.0]
 [1.0 0.0 0.0 44.0 72000.0]
 [1.0 0.0 0.0 35.0 58000.0]]
```

```
print(X_test)
```

```
[[0.0 1.0 0.0 30.0 54000.0]
 [0.0 1.0 0.0 50.0 83000.0]]
```

```
print(y_train)
```

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

```
print(y_test)
```

```
[0 0]
```

FEATURE SCALING

- Feature scaling is the process of scaling of our variables/features to take the same scale, and we do this to prevent one feature dominating the other.

```
print(X)
```

```
[[ 1.22474487 -0.65465367 -0.65465367  0.75887436  0.74947325]
 [-0.81649658 -0.65465367  1.52752523 -1.71150388 -1.43817841]
 [-0.81649658  1.52752523 -0.65465367 -1.27555478 -0.89126549]
 [-0.81649658 -0.65465367  1.52752523 -0.11302384 -0.25320042]
 [-0.81649658  1.52752523 -0.65465367  0.17760889  0.        ]
 [ 1.22474487 -0.65465367 -0.65465367 -0.54897294 -0.52665688]
 [-0.81649658 -0.65465367  1.52752523  0.         -1.0735698 ]
 [ 1.22474487 -0.65465367 -0.65465367  1.34013983  1.38753832]
 [-0.81649658  1.52752523 -0.65465367  1.63077256  1.75214693]
 [ 1.22474487 -0.65465367 -0.65465367 -0.25834021  0.29371249]]
```

THANK YOU