Data Pre-Processing

In [ ]:

```
import numpy as np
import pandas as pd
```

In [ ]:

```
data=pd.read_csv("Data.csv")
data
```

Out[ ]:

	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 [ ]:

```
x=data.iloc[:,:-1].values
y=data.iloc[:,-1].values
print(x)
print()
print(y)
```

```
[['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 [ ]:

```
from sklearn.impute import SimpleImputer
imputa = SimpleImputer(missing_values = np.nan, strategy = 'mean')
imputa.fit(x[:, 1:3])
x[:, 1:3] = imputa.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 [ ]:

```
data2=pd.read_csv("dataset12.csv")
data2
```

Out[ ]:

	Ageyr	Weight	BMI	Healty
0	10yr	15kg	15-25	yes
1	15yr	25kg	18-23	yes
2	22yr	50kg	4-5	no
3	19yr	53kg	9-10	no

In [ ]:

```
a=data2.iloc[:,:-1]
b=data2.iloc[:,-1]
print(a)
print()
print(b)
```

```
  Ageyr Weight    BMI
0  10yr   15kg  15-25
1  15yr   25kg  18-23
2  22yr   50kg    4-5
3  19yr   53kg   9-10

0    yes
1    yes
2     no
3     no
Name: Healty, dtype: object
```

In [ ]:

```
import re
unit="kg"
for i in data2[:]:
    res = [sub.replace(unit, "").strip() for sub in data2[::]]
print(str(res))
```

```
['Ageyr', 'Weight', 'BMI', 'Healty']
```

In [ ]:

```
# for i,rows in data2.iterrows():
#     print(i,rows)
```

In [ ]:

```
def dataclean(data2):
    re=[]
    unit="kg"
    for i in data2:
        data2[i]=data2[i].replace(r'\D',r'',regex=True)
    print(data2)
    return data2
a=dataclean(a)
```

```
  Ageyr Weight   BMI
0    10     15  1525
1    15     25  1823
```

```
2    22     50    45
3    19     53   910
```

```
C:\Users\asus\AppData\Local\Temp\ipykernel_12364\4083331859.py:5: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  data2[i]=data2[i].replace(r'\D',r'',regex=True)
```

In [ ]:

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder=
x = np.array(ct.fit_transform(x))
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]]
```

Standardization

In [ ]:

```
from sklearn.preprocessing import StandardScaler
Sc=StandardScaler()
X_train=Sc.fit_transform(x)
print(X_train)
```

```
[[ 1.22474487e+00 -6.54653671e-01 -6.54653671e-01  7.58874362e-01
   7.49473254e-01]
 [-8.16496581e-01 -6.54653671e-01  1.52752523e+00 -1.71150388e+00
  -1.43817841e+00]
 [-8.16496581e-01  1.52752523e+00 -6.54653671e-01 -1.27555478e+00
  -8.91265492e-01]
 [-8.16496581e-01 -6.54653671e-01  1.52752523e+00 -1.13023841e-01
  -2.53200424e-01]
 [-8.16496581e-01  1.52752523e+00 -6.54653671e-01  1.77608893e-01
   6.63219199e-16]
 [ 1.22474487e+00 -6.54653671e-01 -6.54653671e-01 -5.48972942e-01
  -5.26656882e-01]
 [-8.16496581e-01 -6.54653671e-01  1.52752523e+00  0.00000000e+00
  -1.07356980e+00]
 [ 1.22474487e+00 -6.54653671e-01 -6.54653671e-01  1.34013983e+00
   1.38753832e+00]
 [-8.16496581e-01  1.52752523e+00 -6.54653671e-01  1.63077256e+00
   1.75214693e+00]
 [ 1.22474487e+00 -6.54653671e-01 -6.54653671e-01 -2.58340208e-01
   2.93712492e-01]]
```

In [ ]: