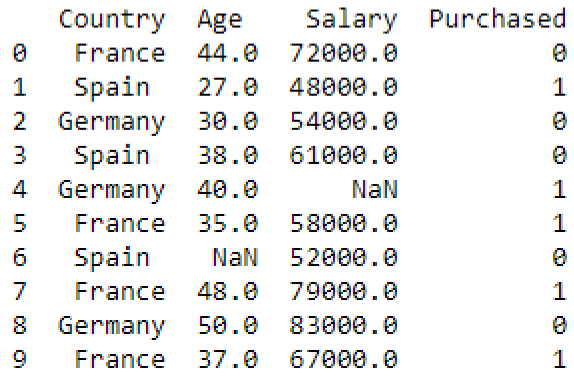
Data preprocessing steps in machine learning

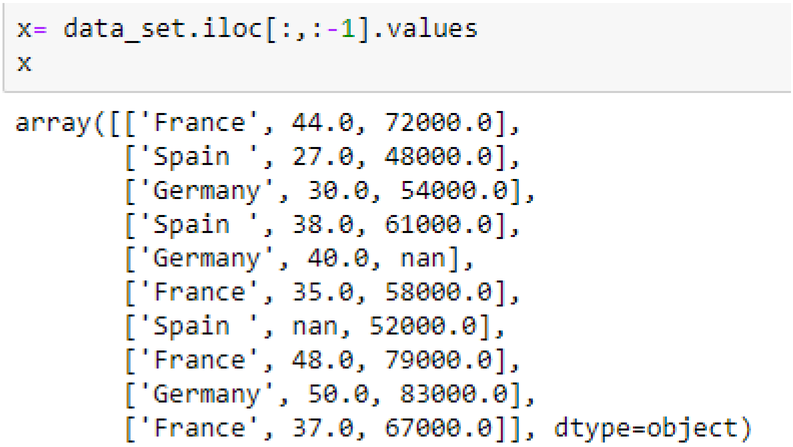
Import libraries and the dataset

```
import pandas as pd

import numpy as np

dataset = pd.read_csv('Datasets.csv')

print (data_set)
```



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

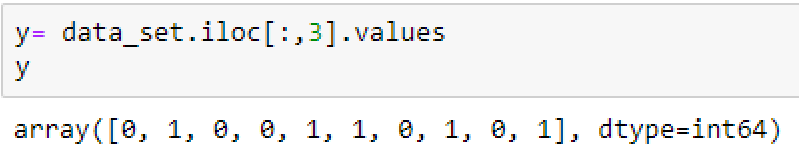
Extracting independent variable:



```
x= data_set.iloc[:,:-1].values
x
```

```
array([['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]], dtype=object)
```

Extracting dependent variable:

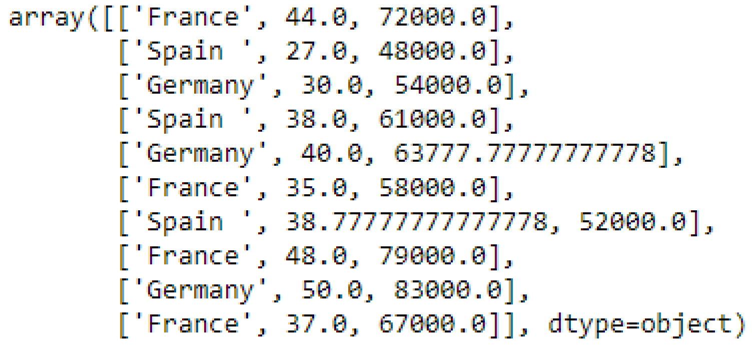


```
y= data_set.iloc[:,3].values
y
```

```
array([0, 1, 0, 0, 1, 1, 0, 1, 0, 1], dtype=int64)
```

Filling the dataset with the mean value of the attribute

```
from sklearn.preprocessing import Imputer

imputer= Imputer(missing_values ='NaN', strategy='mean', axis = 0)

imputerimputer= imputer.fit(x[:, 1:3])

x[:, 1:3]= imputer.transform(x[:, 1:3])

x
```

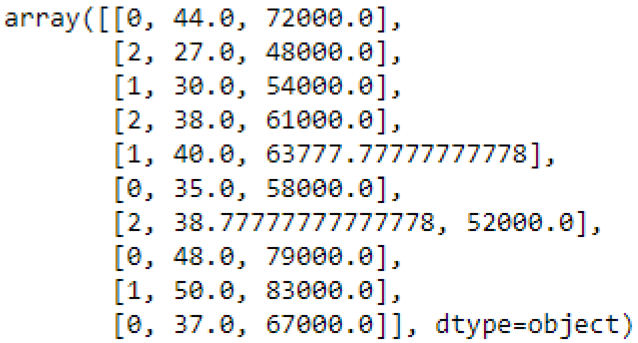


```
array([['France', 44.0, 72000.0],
       ['Spain ', 27.0, 48000.0],
       ['Germany', 30.0, 54000.0],
       ['Spain ', 38.0, 61000.0],
       ['Germany', 40.0, 63777.77777777778],
       ['France', 35.0, 58000.0],
       ['Spain ', 38.77777777777778, 52000.0],
       ['France', 48.0, 79000.0],
       ['Germany', 50.0, 83000.0],
       ['France', 37.0, 67000.0]], dtype=object)
```

Encoding the country variable

The machine learning models use mathematical equations. So categorical data is not accepted so we convert it into numerical form.

```
from sklearn.preprocessing import LabelEncoder

label_encoder_x= LabelEncoder()

x[:, 0]= label_encoder_x.fit_transform(x[:, 0])
```



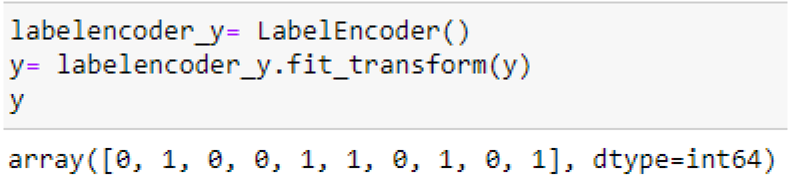
```
array([[0, 44.0, 72000.0],
       [2, 27.0, 48000.0],
       [1, 30.0, 54000.0],
       [2, 38.0, 61000.0],
       [1, 40.0, 63777.77777777778],
       [0, 35.0, 58000.0],
       [2, 38.77777777777778, 52000.0],
       [0, 48.0, 79000.0],
       [1, 50.0, 83000.0],
       [0, 37.0, 67000.0]], dtype=object)
```

Dummy encoding

These dummy variables replace the categorical data as 0 and 1 in the absence or the presence of the specific categorical data.

Encoding for Purchased variable

```
labelencoder_y= LabelEncoder()

y= labelencoder_y.fit_transform(y)
```



```
labelencoder_y= LabelEncoder()
y= labelencoder_y.fit_transform(y)
y
```

```
array([0, 1, 0, 0, 1, 1, 0, 1, 0, 1], dtype=int64)
```

Splitting the dataset into training and test set:

```
 from sklearn.model_selection import train_test_split
```

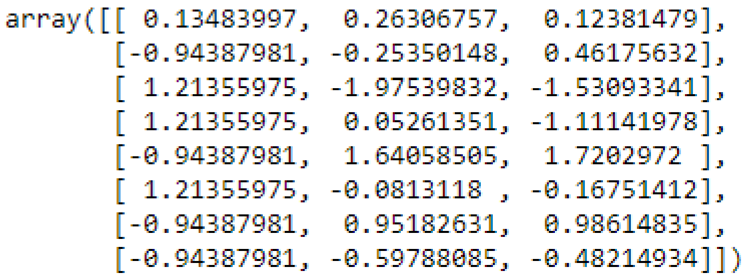
```
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2,
random_state=0)
```

Feature Scaling

```
from sklearn.preprocessing import StandardScaler
```

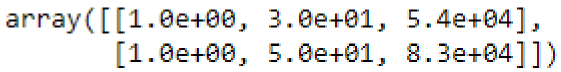
```
st_x= StandardScaler()
```

```
x_train= st_x.fit_transform(x_train)
```



```
array([[ 0.13483997,  0.26306757,  0.12381479],
       [-0.94387981, -0.25350148,  0.46175632],
       [ 1.21355975, -1.97539832, -1.53093341],
       [ 1.21355975,  0.05261351, -1.11141978],
       [-0.94387981,  1.64058505,  1.7202972 ],
       [ 1.21355975, -0.0813118 , -0.16751412],
       [-0.94387981,  0.95182631,  0.98614835],
       [-0.94387981, -0.59788085, -0.48214934]])
```

```
x_test= st_x.transform(x_test)
```



```
array([[1.0e+00, 3.0e+01, 5.4e+04],
       [1.0e+00, 5.0e+01, 8.3e+04]])
```