# Data Preprocessing Tools / Feature

Engineering

# Importing the libraries

In [1] : N

1nport numpy as np

import matplotlib.pyplot as plt import pandas as pd

**Importing the** dataset g

In [6]: fl

dataset = pd.read\_csv('Data.csv') dataset.head()

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Out[6]: |  | | | | |
|  |  | **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 |

|  |  |  |  |
| --- | --- | --- | --- |
| In | [2]: | H | X = dataset.iloc[:, :-1].values y = dataset.iloc[:, -1].values |
|  |  |  |  |
| In | [3]: | N | 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]]

In [4] : N

print (y)

[ No ' ' Yes ' ' No ' No ' Yes ' Yes No’ Yes ' No ' Yes ' ]

# Taking care of missing data

|  |  |  |  |
| --- | --- | --- | --- |
| In | [5]: | N | from sklearn.impute import SimpleImputer  imputer SimpleImputer(missing\_values=np.nan, strategy='mean') imputer.fit(X[:, 1:3])  X[:, 1:3] = imputer.transform(X[:, 1:3]) |
|  |  |  |  |
| In | [0]: | N | 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 categorical data

## Encoding the Independent Variable

In [0]: §

from sklearn.compose import ColumnTransformer from sklearn.preprocessing import OneHotEncoder

ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], r X = np.array(ct.fit\_transform(X))

In [0]: N

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  [1.0 | 0.0  0.0 | 1.0  0.0 | 38.77777777777778 52000.  48.0 79000.0] | |
| [0.0 | 1.0 | 0.0 | 50.0 | 83000.0] |
| [1.0 | 0.0 | 0.0 | 37.0 | 67000.0]] |

0]

## Encoding the Dependent Variable

In [0]: H

from sklearn.preprocessing import LabelEncoder le = LabelEncoder()

y = le.fit\_transform(y)

In [0]: N

print(y)

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

# Splitting the dataset into the Training set and Test set



|  |  |  |  |
| --- | --- | --- | --- |
| In | [0]: | N | 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, |
|  |  |  |  |
| In | [0]: | § | print(X\_train) |
|  |  |  | [[0.0 0.0 1.0 38.77777777777778 52000.0] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [0.0 | 1.0 | 0.0 | 40.0 | 63777.77777777778] |
| [1.0 | 0.0 | 0.0 | 44.0 | 72000.0] |
| [0.0 | 0.0 | 1.0 | 38.0 | 61000.0] |
| [0.0 | 0.0 | 1.0 | 27.0 | 48000.0] |
| [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 | 35.0 | 58000.0]] |

In [0]: N

print(X\_test)

|  |  |  |
| --- | --- | --- |
| [[0.0 1.0 0.0 | 30.0 | 54000.0] |
| [1.0 0.0 0.0 | 37.0 | 67000.0]] |

|  |  |  |  |
| --- | --- | --- | --- |
| In | [0]: | 8 | print(y\_train) |
|  |  |  | [0 1 0 0 1 1 0 1] |
| In | [0]: | N | print(y\_test) |
|  |  |  | [0 1] |

# Feature Scaling

In [0]: N

from sklearn.preprocessing import StandardScaler sc = StandardScaler()

X\_train[:, 3:] = sc.fit\_transform(X\_train[:, 3:])

X\_test[:, 3:] = sc.transform(X\_test[:, 3:])

In [0]: N

print(X\_train)

|  |  |  |  |
| --- | --- | --- | --- |
| [[0.0 | 0.0 | 1.0 | -0.19159184384578545 -1.0781259408412425] |
| [0.0 | 1.0 | 0.0 | -0.014117293757057777 -0.07013167641635372] |
| [1.0 | 0.0 | 0.0 | 0.566708506533324 0.633562432710455] |
| [0.0 | 0.0 | 1.0 | -0.30453019390224867 -0.30786617274297867] |
| [0.0 | 0.0 | 1.0 | -1.9018011447007988 -1.420463615551582] |
| [1.0 | 0.0 | 0.0 | 1.1475343068237058 1.232653363453549] |
| [0.0 | 1.0 | 0.0 | 1.4379472069688968 1.5749910381638885] |
| [1.0 | 0.0 | 0.0 | -0.7401495441200351 -0.5646194287757332]] |

In [0]: 8

print(X\_test)

[[0.0 1.0 0.0 -1.4661817944830124 -0.9069571034860727]

[1.0 0.0 0.0 -0.44973664397484414 0.2056403393225306]]