

▼ Data Preprocessing Tools

▼ Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

▼ Importing the dataset

```
dataset = pd.read_csv('Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

```
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

```
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])
```

```
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

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder='passthrough')
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 63777.777777777778]
 [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]]
```

▼ Encoding the Dependent Variable

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

```
print(y)
```

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

▼ Splitting the dataset into the Training set 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 = 1)
```

```
print(X_train)
```

```
[[0.0 0.0 1.0 38.77777777777778 52000.0]
 [0.0 1.0 0.0 40.0 63777.777777777778]
 [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]]
```

```
print(X_test)
```

```
[[0.0 1.0 0.0 30.0 54000.0]
 [1.0 0.0 0.0 37.0 67000.0]]
```

```
print(y_train)
```

```
print(y_test)
```

▼ Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
print(X_train)
```

```
print(X_test)
```

