Data Preprocessing Template

Importing the libraries

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

Importing the dataset

```python
dataset = pd.read_csv('/content/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]]
```

Taking care of the missing data

```python
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', 41, 0, 72000, 0]

Encoding Categorical Data :

['Germany', 40, 0, 63777.77777777778]

Encoding Independent Variable

['France', 48, 0, 79000, 0]

```
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))
```

Encoding 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 theTraining 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)
```

Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train[:, 3:]=sc.fit_transform(X_train[:, 3:])
X_test[:, 3:]=sc.transform(X_test[:, 3:])

print(X)
```

⤷

```
       [[1.0 0.0 0.0 44.0 72000.0]
print(y)

  [→   [0 1 0 0 1 1 0 1 0 1]

       [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]]
```