

EXPERIMENT 5

5. Experiment to understand feature scaling.

Aim:

To conduct an experiment to understand feature scaling of a dataset

Description:

Understand the importance of feature scaling

Algorithm:

Step 1: Identify Numerical Features

Step 2: Choose a Scaling Technique (e.g., Min-Max Standardization)

Step 3: Apply the Scaling Method

Step 4: Validate and Compare Scaled Output

Code With Output:

```
import numpy as np
import pandas as pd

df = pd.read_csv(r'D:\REC 2nd Year\Data Science\Data Sets\
Pre_Process_Data.csv')
df
```

	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

```
df.head()
```

	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

```
df['Country '].fillna(df['Country '].mode()[0], inplace=True)
features = df.iloc[:, :-1].values
```



```
label=df.iloc[:, -1].values
```

```
from sklearn.impute import SimpleImputer
age=SimpleImputer(strategy="mean",missing_values=np.nan)
Salary=SimpleImputer(strategy="mean",missing_values=np.nan)
age.fit(features[:, [1]])
```

▼ SimpleImputer ⓘ ?



► Parameters

```
Salary.fit(features[:, [2]])
```

▼ SimpleImputer  

► Parameters

SimpleImputer()

▼ SimpleImputer  

► Parameters

```
features[:,[1]]=age.transform(features[:,[1]])
features[:,[2]]=Salary.transform(features[:,[2]])
features
```

```
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)
```

```
from sklearn.preprocessing import OneHotEncoder
oh = OneHotEncoder(sparse_output=False)
Country=oh.fit_transform(features[:,[0]])
Country
```

```
array([[1., 0., 0.],
       [0., 0., 1.],
       [0., 1., 0.],
       [0., 0., 1.],
       [0., 1., 0.],
       [1., 0., 0.],
       [0., 0., 1.],
       [1., 0., 0.],
       [0., 1., 0.],
       [1., 0., 0.]])
```

Result:

Thus python program to understand feature scaling was conducted successfully.

