

Exercise5

November 2, 2025

```
[9]: import pandas as pd  
import numpy as np  
df=pd.read_csv('pre_process_datasample_outlayers.csv')
```

```
[10]: df.head()
```

```
[10]:   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
```

```
[11]: df['Country'] = df['Country'].fillna(df['Country'].mode()[0])
```

```
[12]: val=df.iloc[:, :-1].values  
val1=df.iloc[:, -1].values  
from sklearn.impute import SimpleImputer  
n=SimpleImputer(strategy="mean", missing_values=np.nan)  
sa=SimpleImputer(strategy="mean", missing_values=np.nan)  
n.fit(val[:, [1]])
```

```
[12]: SimpleImputer()
```

```
[13]: sa.fit(val[:, [2]])
```

```
[13]: SimpleImputer()
```

```
[15]: val[:, [1]]=n.transform(val[:, [1]])  
val[:, [2]]=sa.transform(val[:, [2]])  
val
```

```
[15]: 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)
```

```
[16]: from sklearn.preprocessing import OneHotEncoder  
m = OneHotEncoder(sparse_output=False)  
m
```

```
[16]: OneHotEncoder(sparse_output=False)
```

```
[17]: c=m.fit_transform(val[:,[0]])  
c
```

```
[17]: 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.]])
```

```
[18]: set_final=np.concatenate((c,val[:,[1,2]]),axis=1)
```

```
[19]: from sklearn.preprocessing import StandardScaler
```

```
[20]: sc=StandardScaler()  
sc.fit(set_final)  
feat_standard_scaler=sc.transform(set_final)  
feat_standard_scaler
```

```
[20]: array([[ 1.22474487e+00, -6.54653671e-01, -6.54653671e-01,  
            7.58874362e-01,  7.49473254e-01],  
           [-8.16496581e-01, -6.54653671e-01,  1.52752523e+00,  
            -1.71150388e+00, -1.43817841e+00],  
           [-8.16496581e-01,  1.52752523e+00, -6.54653671e-01,  
            -1.27555478e+00, -8.91265492e-01],  
           [-8.16496581e-01, -6.54653671e-01,  1.52752523e+00,  
            -1.13023841e-01, -2.53200424e-01],  
           [-8.16496581e-01,  1.52752523e+00, -6.54653671e-01,  
            1.77608893e-01,  6.63219199e-16],  
           [ 1.22474487e+00, -6.54653671e-01, -6.54653671e-01,  
            -5.48972942e-01, -5.26656882e-01],  
           [-8.16496581e-01, -6.54653671e-01,  1.52752523e+00,  
            0.00000000e+00, -1.07356980e+00],
```

```
[ 1.22474487e+00, -6.54653671e-01, -6.54653671e-01,
  1.34013983e+00,  1.38753832e+00],
[-8.16496581e-01,  1.52752523e+00, -6.54653671e-01,
  1.63077256e+00,  1.75214693e+00],
[ 1.22474487e+00, -6.54653671e-01, -6.54653671e-01,
 -2.58340208e-01,  2.93712492e-01]])
```

```
[21]: from sklearn.preprocessing import MinMaxScaler
mn1=MinMaxScaler(feature_range=(0,1))
mn1.fit(set_final)
f_min=mn1.transform(set_final)
f_min
```

```
[21]: array([[1.        , 0.        , 0.        , 0.73913043, 0.68571429],
 [0.        , 0.        , 1.        , 0.        , 0.        ],
 [0.        , 1.        , 0.        , 0.13043478, 0.17142857],
 [0.        , 0.        , 1.        , 0.47826087, 0.37142857],
 [0.        , 1.        , 0.        , 0.56521739, 0.45079365],
 [1.        , 0.        , 0.        , 0.34782609, 0.28571429],
 [0.        , 0.        , 1.        , 0.51207729, 0.11428571],
 [1.        , 0.        , 0.        , 0.91304348, 0.88571429],
 [0.        , 1.        , 0.        , 1.        , 1.        ],
 [1.        , 0.        , 0.        , 0.43478261, 0.54285714]]))
```

```
[ ]:
```