

DataTransformation

July 7, 2019

Steps for Data Transformation

1. Import Libraries
2. Load Data
3. Seprate Input and Output attributes
4. Perform scaling (Min-Max Normalization)
5. Perform Standardization

```
In [1]: # Step 1: Import Libraries
```

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler
```

```
# Step 2: Load Data
```

```
datasets = pd.read_csv('Data_for_Transformation.csv')
print("\nData :\n", datasets)
#print("\nData statistics\n", datasets.describe())
```

```
# Step 3: Seprate Input and Output attributes
```

```
# All rows, all columns except last
X = datasets.iloc[:, :-1].values
```

```
# Only last column
Y = datasets.iloc[:, -1].values
```

```
#print("\n\nInput : \n", X)
#print("\n\nOutput: \n", Y)
```

```
X_new = datasets.iloc[:, 1:3].values
print("\n\nX for transformation : \n", X_new)
```

Data :

	Country	Age	Salary	Purchased
0	France	44	72000	No

1	Spain	27	48000	Yes
2	Germany	30	54000	No
3	Spain	38	61000	No
4	Germany	40	68000	Yes
5	France	35	58000	Yes
6	Spain	39	52000	No
7	France	48	79000	Yes
8	Germany	50	83000	No
9	France	37	67000	Yes
10	Spain	45	55000	No

X for transformation :

```
[[ 44 72000]
 [ 27 48000]
 [ 30 54000]
 [ 38 61000]
 [ 40 68000]
 [ 35 58000]
 [ 39 52000]
 [ 48 79000]
 [ 50 83000]
 [ 37 67000]
 [ 45 55000]]
```

In [2]: # Step 4 : Perform scaling on age and salary

```
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X_new)
print("\n\nScaled X : \n", X_scaled)
```

Scaled X :

```
[[0.73913043 0.68571429]
 [0.         0.         ]
 [0.13043478 0.17142857]
 [0.47826087 0.37142857]
 [0.56521739 0.57142857]
 [0.34782609 0.28571429]
 [0.52173913 0.11428571]
 [0.91304348 0.88571429]
 [1.         1.         ]
 [0.43478261 0.54285714]
 [0.7826087  0.2         ]]
```

In [3]: # Step 5 : Perform standardization on age and salary

```
std = StandardScaler()
X_std = std.fit_transform(X_new)
print("\n\nStandardized X : \n", X_std)
```

Standardized X :

```
[[ 0.68188156  0.79548755]
 [-1.81835082 -1.41513049]
 [-1.37713334 -0.86247598]
 [-0.2005534  -0.21771238]
 [ 0.09359159  0.42705121]
 [-0.64177088 -0.49403964]
 [-0.05348091 -1.04669415]
 [ 1.27017153  1.44025115]
 [ 1.56431652  1.80868749]
 [-0.34762589  0.33494213]
 [ 0.82895405 -0.77036689]]
```